TOSHIBA

TM TCmini

TCmini

Programming manual

Chapter 2 Instruction

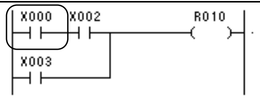
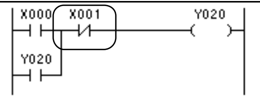
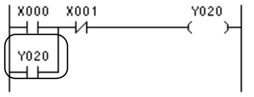
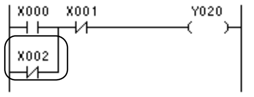
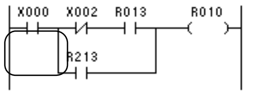
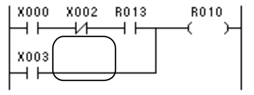
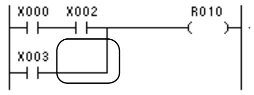
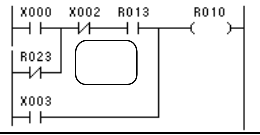
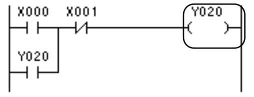
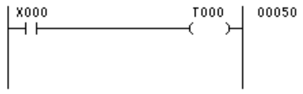
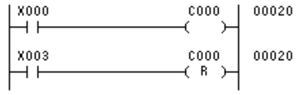
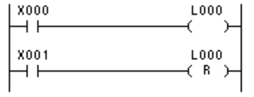
Contents

1. Instruction list

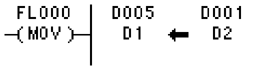
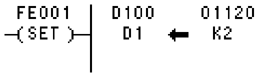
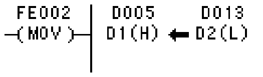
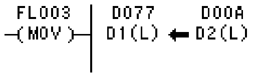
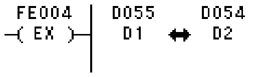
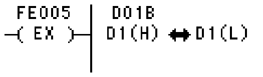
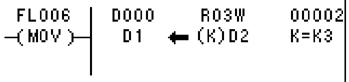
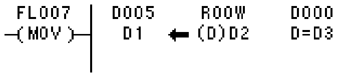
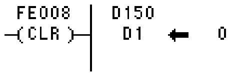
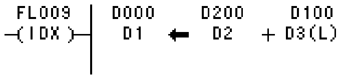
1.1 Basic instruction

TC-mini adopts the ladder symbolic method. Therefore, if the ladder diagram (circuit) is completed, the programming can be done. The order of the logical operation need not be considered.

A basic instruction words is shown as follows.

Instruction	Illustration	Outline	Words	Pages
“a”contact (Normally open)		AND operation of bit X000 is executed.	1	
“b”contact (Normally close)		NOT AND operation of bit X001is executed.	1	
“a”contact branch		OR operation of bit Y020 is executed.	1	
“b”contact branch		NOT OR operation of bit X002 is executed.	1	
Vertical connection		An unconditional connection with the line above is done.	1	
Horizontal connection		An unconditional connection with the previous row is done.	1	
Vertical & horizontal connection		An unconditional connection with a line and previous row above is done.	1	
Blank		Nothing is done.	1	
Coil		When X000 ON & X001 OFF, Y020 is turn on.	1	11
Timer		When it Keeps turning on X000 for 5 seconds, T000 is turned on.	1	13
Counter		When X000 repeats ON/OFF 20 times when X003 is turned off, C000 is turned on.	1	14
Latch		When X000 is turned on, L000 is turned on. When X001 is turned on, L000 is turned off.	1	12

1.2 Data transfer instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
000	Transfer data		The content of register D001 is transmitted to register D005.	4	15
001	Set constant		The BIN constant (1120) is set in D100.	4	16
002	Transfer high-order 8bit data.		The low-order 8 bits data of register D013 is transmitted to high-order 8 bits of register D005.	4	17
003	Transfer low-order 8bit data		The low-order 8 bits data of register D00A is transmitted to low-order 8 bits of register D077.	4	18
004	Exchange data		The data of register D054 is exchanged for the data of register D055.	4	19
005	Exchange data		The high-order 8 bits data of register D01B is exchanged for the low-order 8 bits data.	3	19
006	Transfer blocks （Constant specification）	 FL006 (MOV) D000 D1 R03W (K)D2 00002 K=K3	The 2 words data (R03W and R04W) whose register R03W is a head is transmitted to the area following from register D000 (D000 and D001) .	5	20
007	Transfer blocks variable length (Register specification)	 FL007 (MOV) D005 D1 R00W (D)D2 D000 D=D3	The data（R00W～R02W） that is a number of words (3words) that register D000 shows, and data to make register R00W a head, is transmitted to the area following from register D005(D000～D002).	5	21
008	Clear data		The value of register D150 is Clear to 0.	3	22
009	Extraction& distribution data	 FL009 (IDX) D000 D1 D200 D2 D100 + D3(L)	When the value of D100 is"0204H", D202 (1 word) that is the data of the fourth byte from D200 is forwarded to D001 in the second byte from D000.	5	23

1.3 Data conversion instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
010	BIN→BCD conversion (Unsigned)	FL010 Y03W D121 ─(BCD)─ D1 ← D2 BIN	The value of register D121 is converted into the BCD data, and is stored in register Y03W.	4	24
011	BCD→BIN conversion (Unsigned)	FL011 D120 X10W ─(BIN)─ D1 ← D2 BCD	The BCD data of register X10W is converted into the BIN data, and is stored in register D120.	4	25
012	BIN→BCD conversion (Signed)	FL012 D055 D070 ─(BCD)─ D1+1 ← D2 BIN	The signed BIN data of register D070 is converted into the signed BCD data. The data of low-order 4 digits is stored in register D055. High-order 1 digit and the sign are stored in D056.	4	26
013	BCD→BIN conversion (Signed)	FL013 D010 D200 ─(BIN)─ D1 ← D2+1 BCD	The singed BCD data of register D200 and D201 are converted into the signed BIN data, and is stored in register D010.	4	27
015	Bit reversing	FE015 D007 ─(CPL)─ D1	The bit of the value of register D007 is reversed, and is stored in register D007.	3	28
016	One's complement	FE016 D00F ─(NEG)─ D1	The one's complement of the BIN data of register D00F is operated, and is stored in register D00F.	3	28
096	Floating point number conversion	FE096 D000 D002 ─(B.F)─ D1 ← D2	The double length BIN data of register D003 and D002 is converted into the floating-point number data, and is stored in register D001 and D000.	4	70
097	Floating point number inversion	FE097 D010 D012 ─(F.B)─ D1 ← D2	The floating point number data of register D013 and D012 is converted into double length BIN data, and is stored in register D011 and D010.	4	71

Note） FE00:Execute once by the following scanning that the input condition stood up.
FL00:When the input condition is turned on, it is always executed.

1.4 Arithmetic operation instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
020	BIN addition	FL020 D000 D001 D002 —(BIN)— D1 ← D2 + D3	The sum of the BIN data of register D001 and D002 is requested, and is stored in register D000.	5	29
021	BIN addition (With carry)	FL021 D110 D00F D120 —(BIN)— D1 ← D2 + D3+C	The sum of data of register D00F and carry flag and the BIN data of register D120 is requested, and is stored in register D110.	5	30
022	BIN subtraction	FL022 D030 D112 D020 —(BIN)— D1 ← D2 − D3	The BIN data of register D020 is subtracted from the data of register D112. The difference is stored in register D030.	5	31
023	BIN subtraction (With borrow)	FL023 D003 D051 D044 —(BIN)— D1 ← D2 − D3−C	The BIN data and the carry flag of register D044 are subtracted from the data of register D051. The difference is stored in register D003.	5	32
024	BIN multiplication (Unsigned)	FL024 D105 D04F D01B —(BIN)— D1 ← D2 * D3	The product of the BIN data of register D04F and D01B is requested, and is stored in 2 words of register D106 and D105.	5	33
025	BIN division (Unsigned)	FL025 D005 D101 D05F —(BIN)— D1 ← D2 / D3	The 2 words BIN data of register D102 and D101 is divided by the BIN data of register D05F. The quotient is stored in register D005, and the remainder is stored in D006.	5	34
026	BCD addition	FL026 D010 D011 D012 —(BCD)— D1 ← D2 + D3	The sum of the BCD data of register D011 and D012 is requested, and is stored in register D010.	5	35
027	BCD addition (With carry)	FL027 D01A D02B D03F —(BCD)— D1 ← D2 + D3+C	The sum of the BCD data of register D03F and register D02B and carry flag is requested, and is stored in register D01A.	5	36
028	BCD subtraction	FL028 D033 D02F D031 —(BCD)— D1 ← D2 − D3	The BCD data of register D031 is subtracted from the data of register D02F. The difference is stored in register D033.	5	37
029	BCD subtraction (With Carry)	FL029 D054 D043 D077 —(BCD)— D1 ← D2 − D3−C	The BCD data and the carry flag of register D077 are subtracted from the data of register D043. The difference is stored in register D054.	5	38
030	BCD multiplication	FL030 D055 D042 D040 —(BCD)— D1 ← D2 * D3	The product of the data of register D042 and the BCD data of register D040 is requested, and is stored in 2 words of register D056 and D055.	5	39
031	BCD division	FL031 D060 D035 D02F —(BCD)— D1 ← D2 / D3	The 2 words data of register D036 and D035 is divided by the BCD data of register D02F. The quotient of the operation result is stored in register D060, and the remainder is stored in D061.	5	40

1.4 Arithmetic operation instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
035	Increment	FE035 ┃ D015 ─(INC)─┤ D1 ← D1 + 1	The data of register D015 is increased (+1).	5	44
036	Decrement	FE036 ┃ D070 ─(DEC)─┤ D1 ← D1 − 1	The data of register D070 is decreased (-1).	5	44
092	Double length BIN addition	FE092 ┃ D000 D002 D004 ─(W.B)─┤ D1 ← D2 + D3	The sum of double length BIN data of register D003 and D002 and D005 and D004 is requested, and is stored in register D001 and D000.	5	66
093	Double length BIN subtraction	FE093 ┃ D010 D012 D014 ─(W.B)─┤ D1 ← D2 − D3	The double length BIN data of register D015 and D014 is subtracted from register D013 and D012. The difference is stored in register D011 and D010.	5	67
094	Double length BIN multiplication	FE094 ┃ D020 D024 D026 ─(W.B)─┤ D1 ← D2 * D3	The double length BIN data of register D025 and D024 is multiplied to register D027 and D026, and is stored in register D023 and D022 and D021 and D020.	5	68
095	Double length BIN division	FE095 ┃ D030 D034 D038 ─(W.B)─┤ D1 ← D2 / D3	The BIN data four times length stored in register D037 and D036 and D035 and D034 is divided in double length BIN data of register D039 and D038. The quotient is stored in register D031 and D030, and the remainder is stored in D033 and D034.	5	69
098	Floating point number addition	FL098 ┃ D000 D002 D004 ─(FLW)─┤ D1 ← D2 + D3	The sum of the floating point number data of register D003 and D002 and register D005 and D004 is requested, and is stored in register D001 and D000.	5	72
099	Floating point number subtraction	FE099 ┃ D010 D012 D014 ─(FLW)─┤ D1 ← D2 − D3	The floating-point number data of register D015 and D014 is subtracted from register D013 and D012. The difference is stored in register D011 and D010.	5	73
100	Floating point number multiplication	FE100 ┃ D020 D024 D026 ─(FLW)─┤ D1 ← D2 * D3	The floating-point number data of register D025 and D024 is multiplied to register D027 and D026 and is stored in register D021 and D020.	5	74
101	Floating point number division	FE101 ┃ D030 D034 D038 ─(FLW)─┤ D1 ← D2 / D3	The data of register D035 and D034 is divided in the floating point number data of register D039 and D038. The quotient is stored in register D031 and D030.	5	75
103	BIN multiplication (Signed)	FL103 ┃ D105 D04F D01B ─(BIN)─┤ D1 ← D2 * D3(s)	The product of the BIN data of register D04F and register D01B is Requested, and is stored in 2 words of register D106 and D105.	5	77
104	BIN division (Signed)	FL104 ┃ D005 D101 D05F ─(BIN)─┤ D1 ← D2 / D3(s)	The 2 words data of register D101 and D100 is divided by the BIN data of register D05F. The quotient is stored in register D005, and the remainder is stored in D006.	5	78

1.5 Comparison instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
037	Comparison (Unsigned)	FL037 D100 D005 R050 -(CMP)- D1 ⇔ D2 ➡ B3	The BIN data of register D100 is compared with the BIN data of register D005. The result is stored in R050 and R051.	5	45
038	Comparison (Signed)	FL038 D000 D001 R000 -(CMP)- D1 ⇔ D2 S ➡ B3	The bin data of register D000 is compared with the BIN data of register D001. The result is stored in R000 and R001.	5	46
066	Comparison between data & table	FL066 D000 D010 R00L -(CPD)- D1 ⇔ D2 ➡ D3	The data of register D000 is compared with the data of register D010～D017.If it is agreement, the correspondence bit of register R00L is turned on. If it is a disagreement, R00L is turned off.	5	54
067	Comparison between table & table	FL067 D000 D010 R20L -(CPT)- D1 ⇔ D2 ➡ D3	The data of register D000～D007 is compared with the data of register D010 ～ D017.If it is agreement, the correspondence bit of register R20L is turned on. If it is disagreement, R20L is turned off.	5	55
068	Range comparison	FL068 D000 D030 R000 -(CPR)- D1 ⬌ D2 ➡ B3	The data of register D030 is compared in the section with the data of register D000 and D001. The result is stored Register R000 and R001.	5	56
102	Double length data comparison	FL102 D030 D040 R000 -(CPL)- D1 ⇔ D2 ➡ B3	The double length BIN data of register D031 and D030 and register D041 and D040 data are compared. The result is stored in R000 and R001.	5	76

1.6 Logic operation instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
032	Logical product	FL032 D000 D001 D002 -(AND)- D1 ⬅ D2 & D3	The logical product of the data of register D001 and the data of register D002 is requested, and is stored in register D000.	5	41
033	Logical add	FL033 D10F D110 D112 -(OR)- D1 ⬅ D2 ¦ D3	The logical add of the data of register D110 and the data of register D112 is requested, and is stored in register D10F.	5	42
034	Exclusive logical add	FL034 D10D D103 D102 -(XOR)- D1 ⬅ D2 ^ D3	The exclusive logical add of the data of register D103 and the data of register D102 is requested, and is stored in register D10D.	5	43

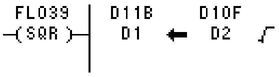
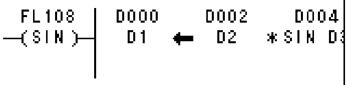
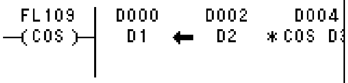
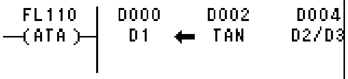
1.7 Shift instruction

FUN No.	Instruction	Illustration	Outline	Words	Pages
040	Left shift	FE040 | R02W -(SLA)- | D1 C « 0	The value of MSB of register R02W is set in A000, and after the left shifts of 1 bit, 0 is set in LSB.	3	48
041	Left rotate	FE041 | R01W -(RL)- | D1 C « C	The value of MSB of register R02W is set in A000, and after the left shifts of 1 bit, the value of A000 before operation is set in LSB.	3	49
042	Right shift	FE042 | R05W -(SRL)- | D1 0 » C	The value of LSB of register R05W is set in A000, and after a right shift 1 bit, 0 is set in MSB.	3	50
043	Right rotate	FE043 | R10W -(RR)- | D1 C » C	The value of LSB of register R10W is set in A000, and after a right shift 1 bit, the value of A000 before operation is set in MSB.	3	51
044	Arithmetic right shift	FE044 | D005 -(SRA)- | D1 » C	The value of LSB of register D005 is set in A000, and after a right shift 1 bit, the value of MSB before operation is set in MSB.	3	52

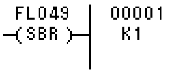
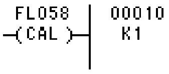
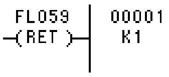
1.8 Data processing order

FUN No.	Instruction	Illustration	Outline	Words	Pages
045	4→16 decipherment	FL045 (DCD) R01W D005 D1 ⬅ D2	The data of register D005 is deciphered, and is stored in register R01W.	4	53
046	16→4 encode	FL046 (ECD) D010 X01W D1 ⬅ D2	The data of register X01W is encoded, and is stored in register D010.	4	54
047	Bit test (Constant specification)	FL047 (TST) D022 00004 R015 D1BIT K2 ➡ B3	The state of bit 4 of the data of register D022 is reflected in relay R015.	5	55
048	Bit test (Register specification)	FL048 (TST) D011 D035 R02B D1BIT D2 ➡ B3	The state of the bit at the position that the data of register D035 shows is reflected in relay R02B by the data of register D011.	5	56
063	1 scan ON	FE063 (EDG) E010 B1	Relay E010 is turned on. After this instruction of the following scanning is executed, E010 is turned off.	3	58
069	FIFO push	FE069 (PUS) D000 D140 00016 TB(D1) K3 ⬅ D2	The data of register D140 is transmitted to the FIFO register of 16 words of register D000 ～ D00F, and the pointer is increased (+1).	5	61
070	FIFO pop	FE070 (POP) D100 D000 00016 D1 ⬅TB(D2) K3	Pop does data from the FIFO register of 16 words of register D000～D00F. After it is forwarded to register D100, the pointer is decreased.	5	61
073	Set coil	FE073 (SET) R100 B1	Relay R100 is turned on.	3	64
074	Reset coil	FE074 (RES) R100 B1	Relay R100 is turned off.	3	64
085	Multi counter	FE085 (MCT) R00W D010 D000 D1 D2 ⇔ D3	The value of register D000 is increased. The value is compared with the data of register D010 ～ D017. The correspondence bit of register R00W is turned on in case of the agreement.	5	65
111	Reference to table	FL111 (CPD) D000 D010 R00W D1 ⬌ D2 ➡ D3	The data of register D000 is compared with the data of register D010～D017. The correspondence bit of register R01W is turned on in case of the agreement. If it is a disagreement, it is assumed as invariable.	5	82

1.9 Function operation instruction

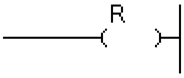
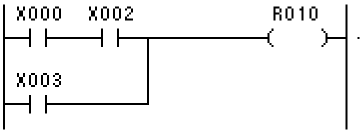
FUN No.	Instruction	Illustration	Outline	Words	Pages
039	Square root (Unsigned)		The square root of the BIN data of register D10F is requested. The result is stored in register D11B.	5	47
108	SIN multiplication (Unsigned)		The double length BIN data of register D005 and D004 is assumed to be angle data in every 0.001 degrees. Sin is operated by using it, and the double length BIN data of register D003 and D002 is multiplied. The result is stored in register D001 and D000.	5	79
109	COS multiplication (Unsigned)		The double length BIN data of register D005 and D004 is assumed to be angle data in every 0.001 degrees. COS is operated by using it, and the double length BIN data of register D003 and D002 is multiplied. The result is stored in register D001 and D000.	5	80
110	ATAN 演算		The double length BIN data of register D005 and D004 data is assumed to be COS data, the double length BIN data of register D003 and D002 data is assumed to be SIN data, and ATAN is operated by using them. The result is stored in register D001 and D000.	5	81

1.10 Program control instruction

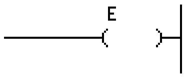
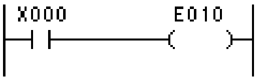
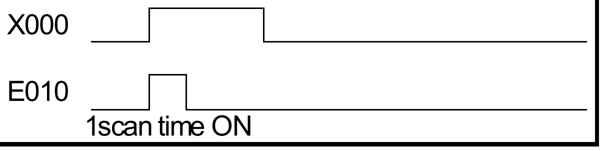
FUN No.	Instruction	Illustration	Outline	Words	Pages
049	Subroutine start		The head of subroutine program 01 is shown.	3	57
058	Subroutine call		Subroutine program 10 is executed.	3	57
059	Subroutine Return		The end of subroutine program 01 is shown.	3	57

2. Basic instruction

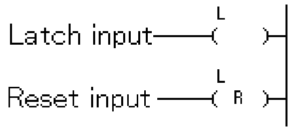
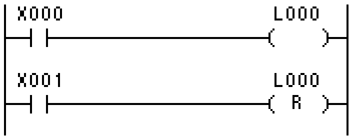
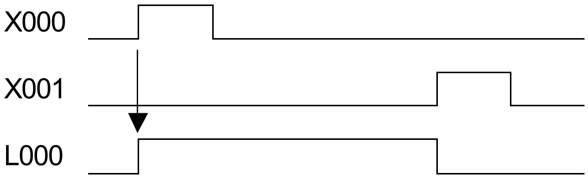
Coil Relay

Symbol		 When both X000 and X002 are turned on or X003 is turned on, R010 is turned on.
Function	When the input condition is turned on, a specified register is turned on.	
Execution condition	When input condition is turned on	
Range of use of register	Y020～Y02B, Y140～Y14F Z200～ZF7F:1792 points R000～R77F:1024points A009, A00A:2 points	

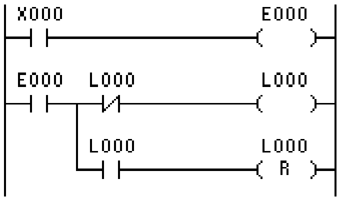
Coil Edge relay

Symbol		 When X000 stands up for turning on from turning off, E010 is turned on only by 1 scanning. 
Function	When the input condition stands up for turning on from turning off, a specified register is turned on only by 1 scanning.	
Execution condition	When the input condition stands up for turning on	
Range of use of register	E000～E17F:256 points	

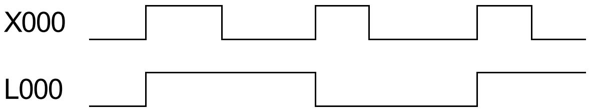
Coil Latch relay

Symbol		 X001 turns on and when X000 is turned on while turned off, L000 is turned on. When X001 is turned on, L000 is turned off. 
Function	When the latch input is turned on, a specified register is turned on. When the reset input is turned on, a specified register is turned off.	
Execution condition	When latch input is turned on	
Range of use of register	L000～L07F:128 points	

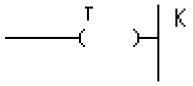
Alternation circuit

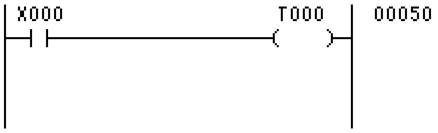


Whenever X000 is turned on, L000 repeats ON/OFF.

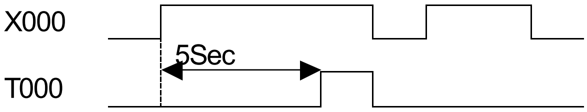


Coil Timer（On delay）

Symbol	
Function	When the time provided in constant K passes, relay "T" is turned on.
Execution condition	When input condition is turned on
Range of use of T	T000～T17F(256Points) 0.1 second Timer (0.1～6553.5sec) T200～T27F(128Points) 0.01sec Timer (0.01～655.35sec)
Range of constant K	1～65535（BIN Data）
Timer setting value register	V000～V27F:384W It is common to the counter.
Timer present value register	P000～P27F:384W It is common to the counter.

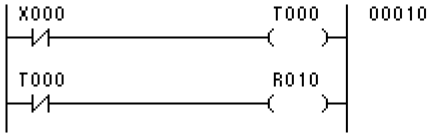


When it Keeps turning on X000, and 5.0 seconds pass, T000 is turned on.

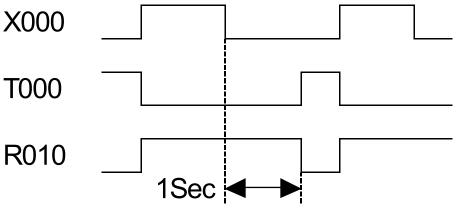


A set value of the timer is stored in the timer setting value register of the same number as the timer address. The set value register of T000 is V000.
This timer is a subtraction type timer. The value of the set value register is forwarded to the present value register now when the input condition stands up for turning on. The value register is subtracted with the passage of time now. When the value becomes 0 now, the timer is turned on.

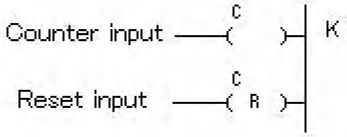
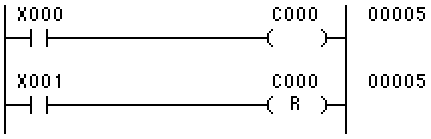
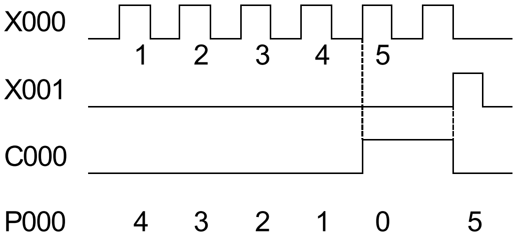
Example of off delay timer circuit



R010 is turned on when X000 is turned on, and after X000 is turned off, R010 will be turned off in 1.0 seconds in set time of T000.



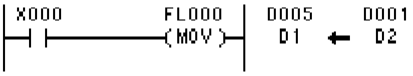
Coil Counter

Symbol		
Function	When the counter input rises from turning off in turning on only as for the frequency provided in constant K, relay “C” is turned on.	C000 counts the frequency from turning off X000 to turning on. C000 is turned on when counting 5 times. When X001 is turned on, C000 is turned off.
Execution condition	When the input condition rises from turning off in turning on	
Range of use of C	C000～C27F:(384Points)	
Range of use of K	1～65535（BIN data）	
Counter setting value register	V000～V27F:384W It is common to the timer.	A set value of the counter is stored in the counter setting value register of the same number as the counter address. The set value register of C000 is V000.
Value register of counter present	P000～P27F:384W It is common to the timer.	This counter is a subtraction type. The value of the set value register is forwarded to the value register when the reset condition is turned off. When the value subtracts with the count, and becomes 0 now the value, C000 is turned on.

3. Data transfer instruction

F∗000 Data transfer

Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
MOV		F∗000	**D1** **d1**	**D2** **d2**	–
Function		The data of the register shown in Ag.2 is transmitted to the register shown in Ag.1.			
Content of operation		MOV (D1 d1) ← (D2 d2)			
Range of use of Ag.1		D1 (Direct register) :All register d1 (Indirect register) :All register			
Range of use of Ag.2		D2 (Direct register) :All register d2 (Indirect register) :All register			
After operation	Contents of Ag.1	Data value that Ag.2 indicates			
	Contents of Ag.2	There is no change.			
	Contents of Ag.3	There is no change.			
	Flag	There is no change.			



When input X000 is turned on, the content of register D001 is transmitted to register D005.

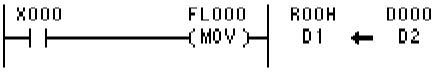
D001:(500) 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0

MSB ↓ LSB

D005:(500) 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0

Both word register and byte register can be forwarded.

Example.1 When register D1 is byte register



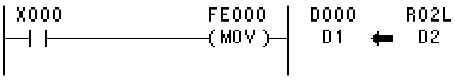
When X000 is turned on, low-order 8 bits of D000 are forwarded to R00H. The content of R00L is not influenced.

D000:(14H) 0 0 0 1 0 1 0 0

↓

R00H:(14H) 0 0 0 1 0 1 0 0

Example.2 When register D2 is byte register



When you turn on X000 from turning off, the 8 bits data of R02L is transmitted to low-order 8 bits of D000.

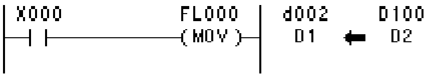
The high-order 8 bits of D000 are cleared to 0.

R02L:(14H) 0 0 0 1 0 1 0 0

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

D000:(14H) 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0

Example.3 When you use indirect register

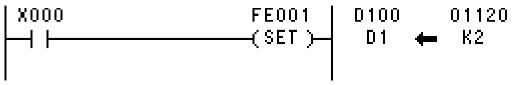


When X000 turn on, the content of D100 is stored at the address that the data value of D002 shows.

When 20H (32) is set to D002, the content of D100 is transmitted to D010.

F∗001 Constant set

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
SET	F∗001	**D1** **d1**	**K2**	—
Function	The constant K2 shown in Ag.2 is stored in the register shown in Ag.1.			
Content of operation	D1 d1 ← K2			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	K2 Constant:-32768～32767			
After operation Contents of Ag.1	Value of K2			
Contents of Ag.2	There is no change.			
Contents of Ag.3	There is no change.			
Flag	There is no change.			



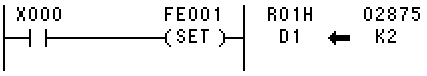
When input X000 stands up for turning on from turning off, the BIN constant (1120) is stored in register D100.

Constant:(1120) 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0

D100:(1120) 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0

When D1 is byte register, the content of the constant of low-order 8 bits is stored in D1. High-order 8 bits are disregarded.

Example.1 When register D1 is byte register

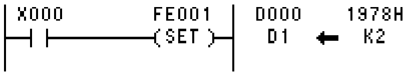


When X000 stands up for turning on from turning off, low-order 8 bits of the constant is stored in register R01H.

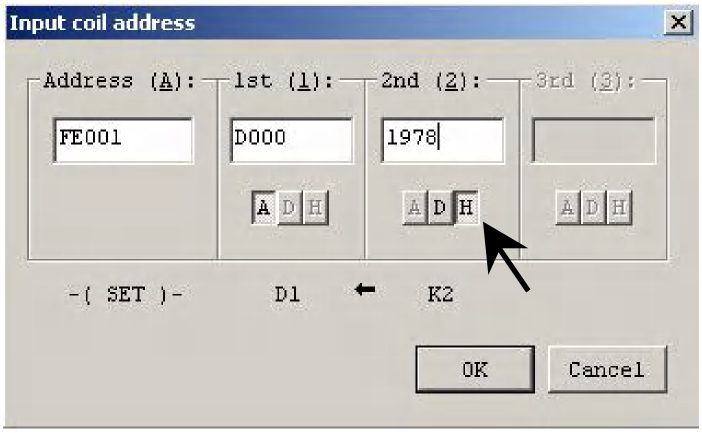
Constant:(2875) 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 1

R01H:(59) 0 0 1 1 1 0 1 1

Example.2 When you set the BCD constant



When X000 stands up for turning on from turning off, the BCD constant (1978H) is stored in register D000.


Input coil address

Address (A): FE001

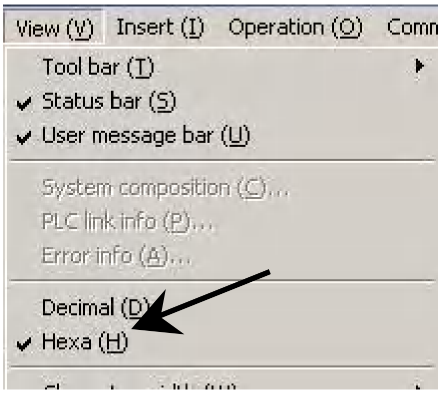
1st (1): D000

2nd (2): 1978

3rd (3):

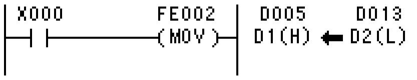
When the BCD constant is input, the input of the hexadecimal number is specified.

To see the numerical value of the BCD constant input, the display is assumed to be a hexadecimal number display.

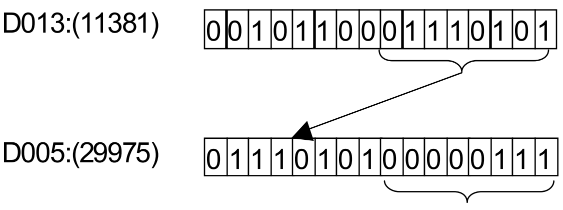


F∗002 High-order 8 bits data transfer

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
MOV	F∗002	**D1** **d1**	**D2** **d2**	—
Function	The data of low-order 8 bits of register where Ag.2 shows is transferred to high-order 8 bits of the register where Ag.1 shows.			
Content of operation	$\begin{pmatrix}D1\\d1\\(H)\end{pmatrix} \leftarrow \begin{pmatrix}D2\\d2\\(L)\end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
After operation	Contents of Ag.1	The content of high-order 8 bits becomes the content of Ag.2 of low-order 8 bits.		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Flag	There is no change.		

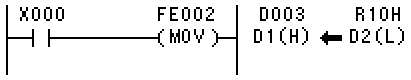


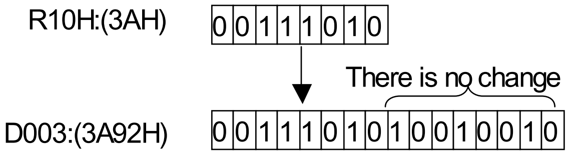
When input X000 stands up for turning on from turning off, the low-order 8 bits data of register D013 is transmitted to high-order 8 bits of register D005.



After operation, low-order 8 bits of D005 do not change.

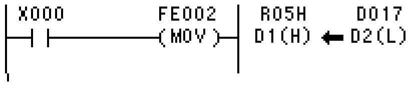
Example.1 When register D2 is byte register

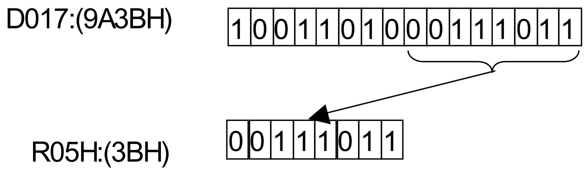




When X000 stands up for turning on from turning off, the content of register R10H is transmitted to 8 high-order bits of register D003.

Example.2 When register D1 is byte register



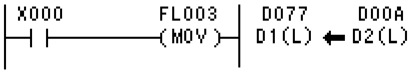


The low-order 8 bits of D017 are transmitted to R05H.

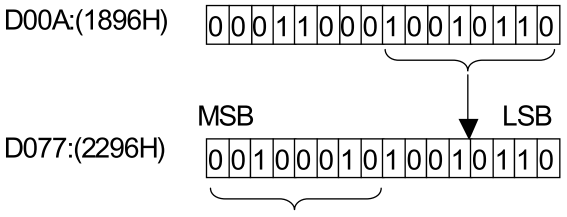
Note) It becomes the same operation as FE000.

F∗003 Low-order 8 bits data transfer

Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
MOV		F∗003	**D1** **d1**	**D2** **d2**	
Function		The low-order 8 bits data of the register where Ag.1 shows is transmitted to low-order 8 bits of the register that Ag.2 shows.			
Content of operation		$\begin{pmatrix}D1\\d1\\(L)\end{pmatrix} \leftarrow \begin{pmatrix}D2\\d2\\(L)\end{pmatrix}$			
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register			
After operation	Contents of Ag.1	The content of low-order 8 bits becomes the content of Ag.2 of low-order 8 bits.			
	Contents of Ag.2	There is no change.			
	Flag	There is no change.			

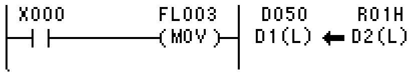


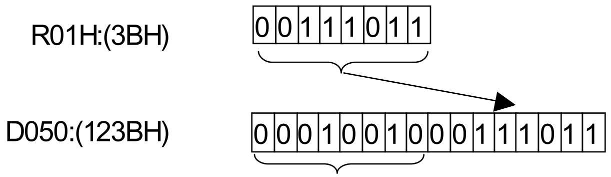
When input X000 is turned on, the data of low-order 8 bits of register D00A is transmitted to low-order 8 bits of register D077.



High-order 8bit bits of D077 do not change.

Example.1 When register D2 is byte register

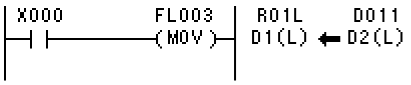


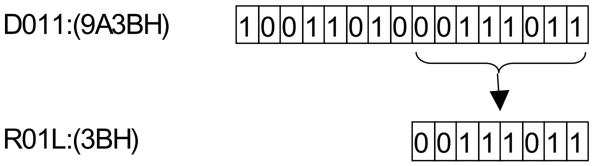


High-order 8 bits do not change.

When X000 stands up for turning on from turning off, the content of register R01H is transmitted to low-order 8 bits of register D050.

Example.2 When register D1 is byte register



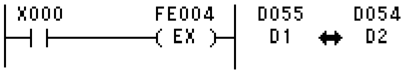


The low-order 8 bits of D011 are transmitted to R05L.

It becomes the same operation as FE000.
The value of R01H doesn't change.

F∗004 Data exchange

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
EX	F∗004	**D1** **d1**	**D2** **d2**	
Function	The data of the register shown in Ag.1 is exchanged for the data of the register shown in Ag.2.			
Content of operation	D1 d1 ⇔ D2 d2			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
After operation	Contents of Ag.1	Data value of register that Ag.2 shows.		
	Contents of Ag.2	Data value of register that Ag.1 shows.		
	Flag	There is no change.		



When input X000 stands up for turning on from turning off, the data of register D054 is exchanged for the data of register D055.

Before operation

D054:(1234H) 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0

D055:(5555H) 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

↓

After operation

D054: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

D055: 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0

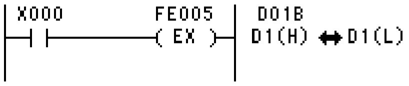
The data exchange of byte register can be done.

When the data exchange is executed between byte register and the word register, the data of the word register of low-order 8 bits is exchanged for the data of byte register. High-order 8 bits of the word register are cleared.

F∗005 Data exchange

（high-order 8 bits⇔low-order 8 bits）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
EX	F∗005	**D1** **d1**		
Function	The data of high-order 8 bits of the register that Ag.1 shows is exchanged for the data of low-order 8 bits.			
Content of operation	D1 d1 (H) ⇔ D1 d1 (L)			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
After operation	Contents of Ag.1	Operation result		
	Flag	There is no change.		



When input X000 stands up for turning on from turning off, the value of high-order 8 bits of register D01B is exchanged for the value of low-order 8 bits.

Before operation

D01B:(1234H) 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0

↓

After operation

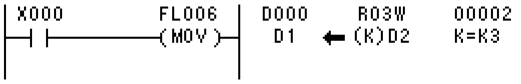
D01B:(3412H) 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0

When byte register is specified, the value of high-order 4 bits is exchanged for the value of low-order 4 bits.

F∗006 Block transfer

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
MOV	F∗006	**D1** **d1**	**D2** **d2**	**K3**

Function		The block data of number of constant K3 words whose register that Ag.2 shows is head is transmitted behind the register that Ag.1 shows.
Content of operation		D2 d2 | D2+1 d2+1 …… …d2+K3-1 D2+K3-1 ↓ D1 d1 | D1+1 d1+1 ……… D1+K3-1 d1+K3-1
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
Range of use of Ag.3		K3 Constant:0～255
After operation	Contents of Ag.1	Data value of register that Ag.2 shows
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Flag	There is no change.



When input X000 is turned on, the data of 2 words whose register R03W is head(R03W · R04W) is transmitted behind register D000 (D000 and D001).

R03W:(1) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

R04W:(2) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

（2 words）

↓

D000:(1) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

D001:(2) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

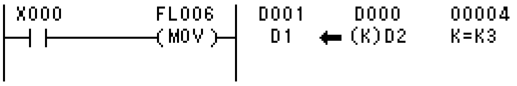
（2 words）

When the transfer destination comes off from the area of the register, it is not executed.

Even if byte register is specified for the register of the destination of the transfer origin or transfer, it is forwarded as a word register.

Example.1 When the register of the forwarding destination and the forwarding origin comes in succession

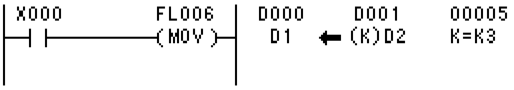
(1)Transfer destination:D1 ＞ Transfer origin:D2



After executing the operation, D000 ～D004 reaches the same value.

Transfer origin:D2		Transfer destination:D1	
D000	Data1	D000	Data1
D001	Data2	D001	Data1
D002	Data3	D002	Data1
D003	Data4	D003	Data1
D004	Data5	D004	Data1
D005	Data6	D005	Data6

(2)Transfer destination:D1 ＜ Transfer origin:D2



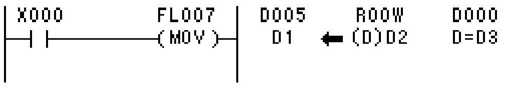
1 data moves to the register in this side after executing the operation.

Transfer origin:D2		Transfer destination:D1	
D000	Data1	D000	Data2
D001	Data2	D001	Data3
D002	Data3	D002	Data4
D003	Data4	D003	Data5
D004	Data5	D004	Data6
D005	Data6	D005	Data6

F＊007 Variable length block transfer

Symbol	Code	Argument Ag.1	 Ag.2	 Ag.3
MOV	F＊007	**D1** **d1**	**D2** **d2**	**D3**

Function		The block data that is number of words of content of register shown in Ag.3 and that makes the register shown in Ag.2 a head is transmitted since the register that Ag.1 shows.
Content of operation		D2 / d2, D2+1 / d2+1 …… D2+(D3-1) / d2+(D3-1) ↓ D1 / d1, D1+1 / d1+1 …… D1+(D3-1) / d1+(D3-1)
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
Range of use of Ag.3		D3（Direct register）:All register BIN（0～255）
After operation	Contents of Ag.1	Data value of register that Ag.2 shows.
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Flag	There is no change.



When input X000 is turned on, the data whose register R00W is head and that is number of words (three words) shown in register D000(R00W～R02W) is transmitted since register D005 (D000～D002).

R00W:(1) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

R01W:(2) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

R02W:(3) (3Words) 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

↓

D005:(1) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

D006:(2) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

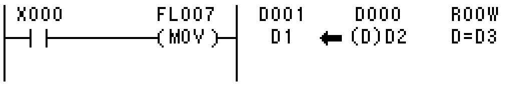
D007:(3) (3Words) 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

When the transfer destination comes off from the area of the register, it is not executed.

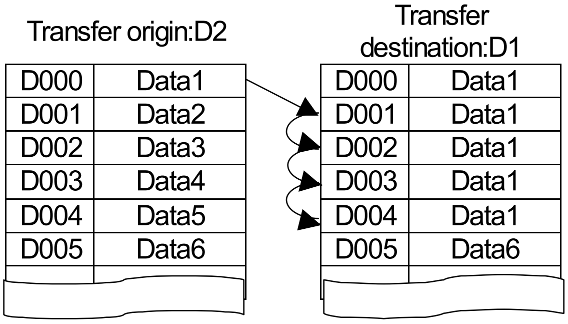
Even if byte register is specified for the register at the transfer former register and transfer destination, it is forwarded as word register.

Example.1 When the register of the forwarding destination and the forwarding origin comes in succession

① Transfer destination:D1 ＞ Transfer origin:D2



After executing the operation, D000～D004 reaches the same value.



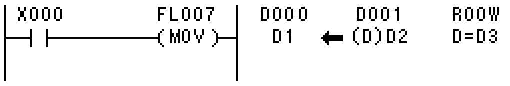
Transfer origin:D2

D000	Data1
D001	Data2
D002	Data3
D003	Data4
D004	Data5
D005	Data6

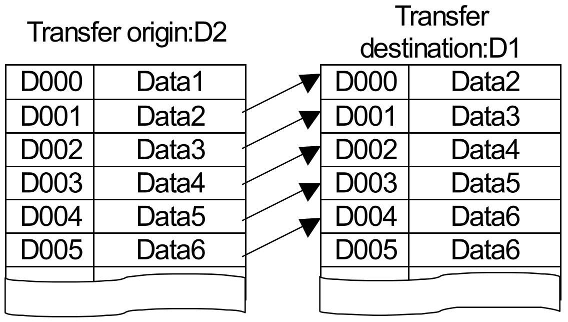
Transfer destination:D1

D000	Data1
D001	Data1
D002	Data1
D003	Data1
D004	Data1
D005	Data6

② Transfer destination:D1 ＜ Transfer origin:D2



1 data moves to the register in this side after executing the operation.



Transfer origin:D2

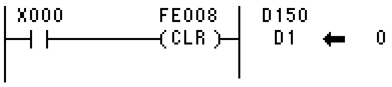
D000	Data1
D001	Data2
D002	Data3
D003	Data4
D004	Data5
D005	Data6

Transfer destination:D1

D000	Data2
D001	Data3
D002	Data4
D003	Data5
D004	Data6
D005	Data6

F＊008 Data clear

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
CLR	F＊008	**D1** **d1**		
Function	The data of register that Ag.1 shows is cleared (set to 0).			
Content of operation	D1 d1 ← 0			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
After Contents of Ag.1	Set to 0.			
After Flag	There is no change.			

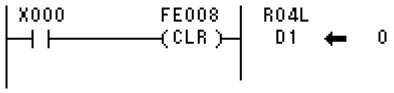


When input X000 stands up for turning on from turning off, the value of register D150 is Cleared.

Before operation D150:(2222H)	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
After operation D150:(0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

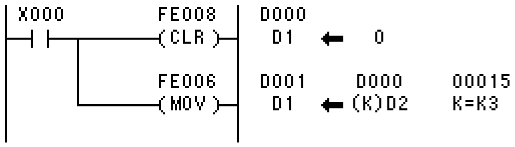
When byte register is specified, 0 clearing of 1 byte.

Example.1 When register D1 is byte register



When X000 stands up for turning on from turning off, the value of register R04L is cleared. R04H is not changed.

Example.2 When you want to clear 16 Words to register D000～D00F0

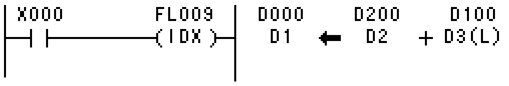


Register D000 is cleared.

The data of 15Words is forwarded from D000 since D001.

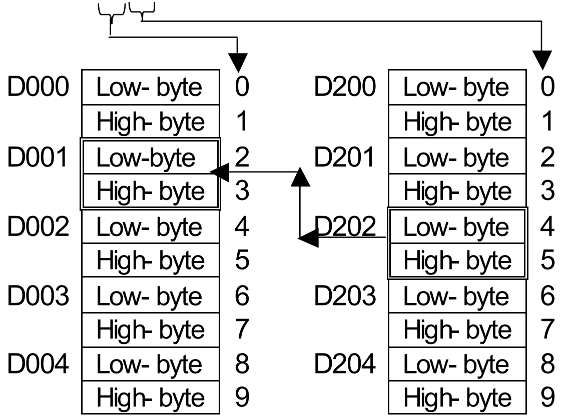
F＊009 Data extraction & distribution

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
MOV	F＊009	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The data value indicated in the low-order byte of Ag.3 is adjusted to the offset value of each byte of the address, the data is extracted from the table that makes the register shown in Ag.2 a head, and the data value that a high-order byte of Ag.3 indicates is forwarded to the table that makes the register shown in Ag.1 a head as an offset value of each byte of the address.			
Content of operation	(D1(D3(H)) d1(d3(H))) ← (D2(D3(L)) d2(d3(L)))			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Flag	There is no change.		



When the value of D100 is (0204H), the data D202 (1 word) in the fourth byte from D200 is forwarded to D001 in the second byte from D000.

D100:（0204H）

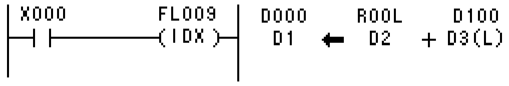


D000	Low- byte	0	D200	Low- byte	0
	High- byte	1		High- byte	1
D001	Low-byte	2	D201	Low- byte	2
	High- byte	3		High- byte	3
D002	Low- byte	4	D202	Low- byte	4
	High- byte	5		High- byte	5
D003	Low- byte	6	D203	Low- byte	6
	High- byte	7		High- byte	7
D004	Low- byte	8	D204	Low- byte	8
	High- byte	9		High- byte	9

The amount of transfer data is 2 bytes (1 word).

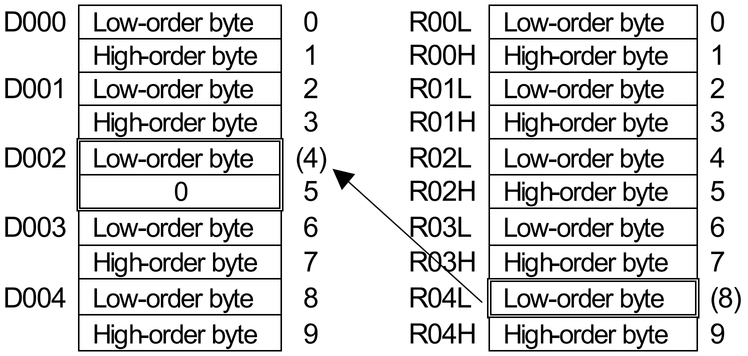
Example.1

When the addressing at the transfer origin and transfer destination are the byte register and the word register



When Ag.2 is specified by R00L and byte register, a high-order byte is forwarded as 0.

D100:（0408H）



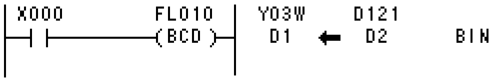
D000	Low-order byte	0	R00L	Low-order byte	0
	High-order byte	1	R00H	High-order byte	1
D001	Low-order byte	2	R01L	Low-order byte	2
	High-order byte	3	R01H	High-order byte	3
D002	Low-order byte	(4)	R02L	Low-order byte	4
	0	5	R02H	High-order byte	5
D003	Low-order byte	6	R03L	Low-order byte	6
	High-order byte	7	R03H	High-order byte	7
D004	Low-order byte	8	R04L	Low-order byte	(8)
	High-order byte	9	R04H	High-order byte	9

4. Data conversion instruction

F＊010 BIN→BCD conversion（Unsigned）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BCD	F＊010	**D1** **d1**	**D2** **d2**	

Function		The unsigned BIN data of the register that Ag.2 shows is converted into BCD data, and is stored in the register that Ag.1 shows.
Content of operation		$BCD\begin{bmatrix}D1\\d1\end{bmatrix} \leftarrow BIN\begin{bmatrix}D2\\d2\end{bmatrix}$
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	A002 flag	It turns it on when BIN Data of D2 is larger than that of 9999.

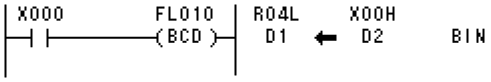


When input X000 is turned on, the value of register D121 is converted into BCD data and is stored in register Y03W.

D121:(648)	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0
Y03W:(648H)	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0
					6				4				8			

When the BIN data of register D2 is larger than that of 9999, the codes other than BCD are stored in the BCD highest-order digit of register D1. It doesn't become normal conversion.

Example.1 When register D1, D2 are byte register



The BIN data of X00H in 1 byte is converted into BCD data, and is stored in R04L.

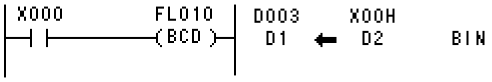
Note)
When the value of X00H exceeds 99, it is disregarded more than the treble.

X00H							
78							
0	1	0	0	1	1	1	0

→

R04L							
7				8			
0	1	1	1	1	0	0	0

Example.2 When register D1 is word register, and register D2 is byte register



The BIN data of X00H in 1 byte is converted into BCD data and is stored in D003.

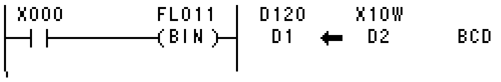
X00H							
209							
1	1	0	1	0	0	0	1

→

D003															
0				2				0				9			
0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1

F＊011 BCD→BIN conversion（Unsigned）

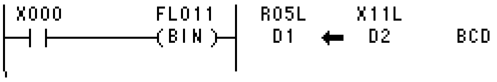
<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">BIN</td><td>F＊011</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The unsigned BCD data of the register that Ag.2 shows is converted into BIN data, and is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">BIN (D1 / d1) ← BCD (D2 / d2)</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td rowspan="3">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 is turned on, the BCD data of register X10W is converted into BIN data, and is stored in register D120.

X10W:(256H)	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0
D120:(256)	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Example.1 When register D1, D2 are byte register



The BCD data of X11L in 1 byte is converted into BIN data, and is stored in R05L.

X11L

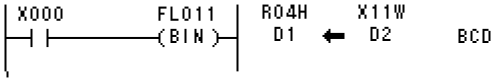
6				3			
0	1	1	0	0	0	1	1

→

R05L

63							
0	0	1	1	1	1	1	1

Example.2 When register D1 is byte register, and register D2 is word register



The BIN data of X11W is converted into BCD data, and 1 low-order byte is stored in R04H.

X11W

0				2				5				4			
0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0

→

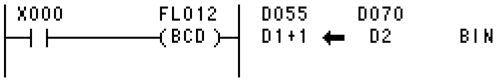
R04H

254							
1	1	1	1	1	1	1	0

Note)
When the value of X11W is 256 or more, 1 low-order byte of the result of conversion into BIN data is stored in R04H.

F∗012 BIN→BCD conversion（Signed）

<table>
<tr><td rowspan="2" colspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">BCD</td><td>F∗012</td><td>D1
d1</td><td>D2
d2</td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The signed BIN data of the register that Ag.2 shows is converted into BCD data, and is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">BCD (D1+1,D1
d1+1,d1) ← BIN (D2
d2)</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td rowspan="3">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 is turned on, the signed BIN data of register D070 is converted into signed BCD data. Low-order 4 digits are stored in register D055, and high-order 1 digit and the sign are stored in D056.

D070:(-25089)

1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1

↓

D055:(5089H)

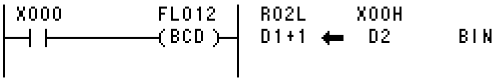
0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	1
5				0				8				9			

D056:(-002H)

1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
—				0				0				2			

The negative value is shown when high-order 4 bits of register D1+1 are "1101", and the positive number is shown at "0000".

Example.1 When register D1, D2 are byte register



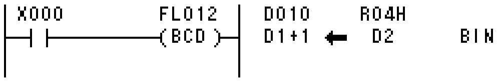
Even if byte register is specified, register D1 is stored as data in 4 bytes.

X00H →

X00H							
195							
1	1	0	0	0	0	1	1

R03H								R03L								R02H								R02L							
0				0				0				0				0				1				9				5			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1

Example.2 When register D1 is word register, and register D2 is byte register



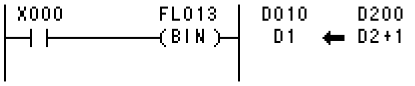
R04H →

R04H							
195							
1	1	0	0	0	0	1	1

D011																D010															
0				0				0				0				0				1				9				5			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1

F∗013 BCD→BIN conversion（Signed）

<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">BIN</td><td>F∗013</td><td>D1
d1</td><td>D2
d2</td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The signed BCD data of the register that Ag.2 shows is converted into BIN data, and is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">BIN （D1
d1）← BCD （D2,D2+1
d2,d2+1）</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td rowspan="3">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 is turned on, the signed BCD data of register D200 and D201 is converted into signed BIN data and is stored in register D010.

D200:(8432H)

<table>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">8</td><td colspan="4">4</td><td colspan="4">3</td><td colspan="4">2</td></tr>
</table>

D201:(-001H)

<table>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="4">—</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">1</td></tr>
</table>

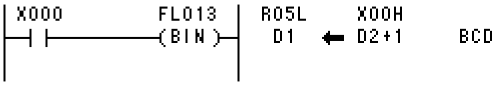
↓

D010:(-18432)

<table>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

When the BCD data stored in register D2 and D2+1 is not a range of -32768～32767, conversion is not correct.

Example.1 When register D1, D2 are byte register



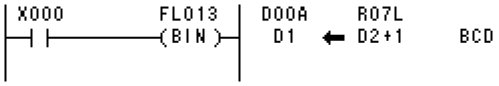
Even if byte register is specified, register D2 is treated as data in 4 bytes.

<table>
<tr><th colspan="8">X02L</th><th colspan="8">X01H</th><th colspan="8">X01L</th><th colspan="8">X00H</th></tr>
<tr><td colspan="4">—</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">1</td><td colspan="4">0</td><td colspan="4">3</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table>

↓

<table>
<tr><th colspan="8">R05L</th></tr>
<tr><td colspan="8">-103</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
</table>

Example.2 When register D1 is word register, and register D2 is byte register



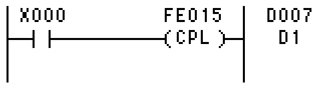
<table>
<tr><th colspan="8">R10H</th><th colspan="8">R10L</th><th colspan="8">R07H</th><th colspan="8">R07L</th></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">7</td><td colspan="4">0</td><td colspan="4">9</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
</table>

↓

<table>
<tr><th colspan="16">D00A</th></tr>
<tr><td colspan="16">709</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
</table>

F∗015 Bit reversing

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
CPL	F∗015	**D1** **d1**		
Function	The data of the register that Ag.1 shows reverses biting, and is stored in register that Ag.1 shows.			
Content of operation	$\begin{pmatrix}D1\\d1\end{pmatrix}\xrightarrow[INV]{}\begin{pmatrix}D1\\d1\end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register） :All register			
After operation — Contents of Ag.1	Data after bit reverses			
After operation — Flag	There is no change.			



When input X000 stands up for turning on from turning off, the data of register D007 is reversed.

Before operation
D007:(9B63H)

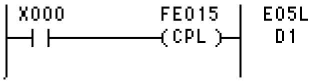
1	0	0	1	1	0	1	1	0	1	1	0	0	0	1	1

After operation
D007:(649CH)

0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0

Byte register can be operated.

Example.1 When register D1 is byte register



E05L

A				A			
1	0	1	0	1	0	1	0

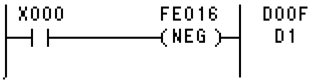
→

E05L

5				5			
0	1	0	1	0	1	0	1

F∗016 One's complement

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
NEG	F∗016	**D1** **d1**		
Function	The one's complement of data of the register that Ag.1 shows is operated, and is stored in the register that Ag.1 shows.			
Content of operation	$\begin{pmatrix}D1\\d1\end{pmatrix}\xrightarrow[NEG]{}\begin{pmatrix}D1\\d1\end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register） :All register			
After — Contents of Ag.1	One's complement value			
After — Flag	There is no change.			



When input X000 stands up for turning on from turning off, the one's complement of BIN data of register D00F is operated and is stored in register D00F.

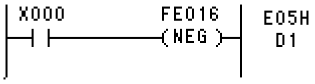
Before operation
D00F:(5192)

0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0

After operation
D00F:(-5192)

1	1	1	0	1	0	1	1	1	0	1	1	1	0	0	0

Example.1 When register D1 is byte register



E05H

170							
1	0	1	0	1	0	1	0

→

E05H

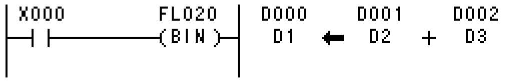
86							
0	1	0	1	0	1	1	0

5. Arithmetic operation instruction

F＊020 BIN Addition

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊020	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**

Function		The BIN data of the register that Ag.2 shows and BIN data of the register that Ag.3 shows are added, and are stored in the register that Ag.1 shows.
Content of operation		$\mathrm{BIN}\begin{Bmatrix}\mathrm{D1}\\ \mathrm{d1}\\ \end{Bmatrix} \leftarrow \begin{Bmatrix}\mathrm{D2}\\ \mathrm{d2}\\ \mathrm{K2}\end{Bmatrix} + \begin{Bmatrix}\mathrm{D3}\\ \mathrm{d3}\\ \mathrm{K3}\end{Bmatrix}$
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Carry Flag Zero Flag Sign Flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.

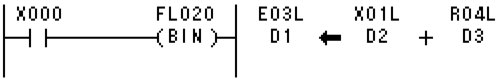


When input X000 is turned on, the data of register D001 and the BIN data of register D002 is added. The result is stored in register D000.

	MSB															LSB
D001:(5641)	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1
＋																
D002:(1158)	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
↓																
D000:(6799)	0	0	0	1	1	0	1	0	1	0	0	0	1	1	1	1
A000:																0

When byte register is used for register D1, sum in 1 low-order byte is stored in D1. 1 high-order byte is disregarded.

Example.1 When register D1, D2 are byte register



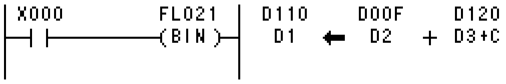
BIN data of R04L and X01L is added, and 1 low-order byte is stored in E03L.

	X01L		R04L	→	E03L
	198	＋	211		153
	1 1 0 0 0 1 1 0		1 1 0 1 0 0 1 1		1 0 0 1 1 0 0 1

Even if carry is generated (Operation result is 256 or more) for all byte register, A000 is not turned on.
When register D2 and D3 are byte registers, this instruction operates the register as word register whose 1 high-order byte is 0.

F＊021 BIN addition（with carry）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊021	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The BIN data of the register that Ag.2 shows, and BIN data of the register that Ag.3 shows, and carry flag (A000) are added in BIN, and is stored in the register that Ag.1 shows.			
Content of operation	$\text{BIN}\begin{Bmatrix}D1\\d1\end{Bmatrix}\leftarrow\begin{Bmatrix}D2\\d2\\K2\end{Bmatrix}+\begin{Bmatrix}D3\\d3\\K3\end{Bmatrix}+C$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.		

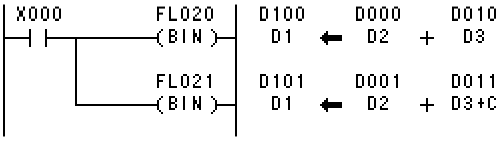


When input X000 is turned on, the data of register D120, the BIN data of register D00F, and carry flag are added. The result is stored in register D110.

D00F:(11282)	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0
								＋								
D120:(4932)	0	0	0	1	0	0	1	1	0	1	0	0	0	1	0	0
								＋								
A000:																1
								↓								
	MSB															LSB
D110:(16215)	0	0	1	1	1	1	1	1	0	1	0	1	0	1	1	1
A000:																0

When byte register is used for register D1, the sum in low-order 1 byte is stored in D1. High-order 1 byte is disregarded.

Example.1 When you add BIN data of 1 word or more

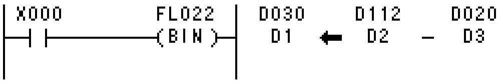


Register (D001･D000) and register (D011･D010) is added by BIN. The result is stored in register (D101･D100).

Computing range: 0～4294967295

F∗022 BIN subtraction

Symbol	Code	Argument Ag.1	Argument Ag.2	Argument Ag.3
BIN	F∗022	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The BIN data of the register that Ag.3 shows from BIN data of the register that Ag.2 shows is subtracted, and is stored in the register that Ag.1 shows.			
Content of operation	$\text{BIN}\begin{pmatrix}D1\\d1\end{pmatrix} \leftarrow \begin{pmatrix}D2\\d2\\K2\end{pmatrix} - \begin{pmatrix}D3\\d3\\K3\end{pmatrix} \quad \begin{pmatrix}\ \\ \ \end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation: Contents of Ag.1	Operation result			
After operation: Contents of Ag.2	There is no change.			
After operation: Contents of Ag.3	There is no change.			
Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.			



When input X000 is turned on, the BIN data of register D020 is subtracted from the data of register D112. The difference is stored in register D030.

D112:(25140) 0 1 1 0 0 0 1 0 0 0 1 1 0 1 0 0

—

D020:(12449) 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1

↓

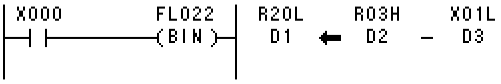
MSB LSB

D030:(12691) 0 0 1 1 0 0 0 1 1 0 0 1 0 0 1 1

A000: 0

The sum in 1 low-order byte is stored in D1 when byte register is used for register D1, and 1 high-order byte is disregarded.

Example.1 When register D1, D2 are byte register

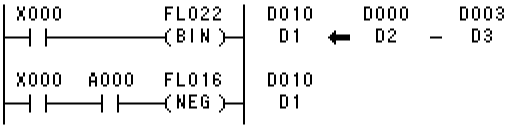


The BIN data of X01L is subtracted from R03H.
The difference is stored in R20L.

R03H		X01L		R20L
57	—	108	→	250
0 0 1 1 1 0 0 1		0 1 1 0 1 1 0 0		1 1 0 0 1 1 0 1

Carry A000 is turned on when the digit fall (0 or less as a result) is generated in case of all byte registers.

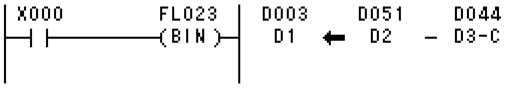
Example.2 When you want result by the absolute value



The one's complement of the result is taken when borrow is generated, and it is made to the positive number.

F＊023 BIN subtraction（with borrow）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊023	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The BIN data and the carry flag of the register shown in Ag.3 are subtracted from the BIN data of the register shown in Ag.2. The result is stored in the register that Ag.1 shows.			
Content of operation	$\mathrm{BIN}\begin{Bmatrix}D1\\d1\\ \end{Bmatrix}\leftarrow\begin{Bmatrix}D2\\d2\\K2\end{Bmatrix}-\begin{Bmatrix}D3\\d3\\K3\end{Bmatrix}-C$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.		



When input X000 is turned on, the BIN data of register D044 and the carry flag are subtracted from data of register D051.
The difference is stored in register D003.

D051:(22312) 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0

—

D044:(8994) 0 0 1 0 0 0 1 1 0 0 1 0 0 0 1 0

—

A000: 1

↓

MSB LSB

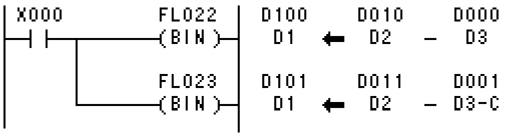
D003:(13317) 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 1

A000: 0

When byte register is used for register D1, low-order 1 byte of the result is stored in D1 and high-order 1 byte is disregarded.

Example.1 When you do the BIN subtraction by the data length of 2 words

The value of register (D001 and D000) is subtracted from the value of register (D011 and D010). The difference is stored in register (D101 and D100).



Data of low-order 1 word is subtracted.

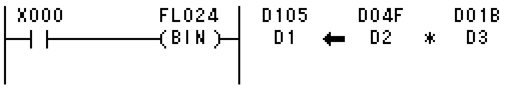
Data of high-order 1 word is subtracted.

Note) When borrow is generated by operating FL23, operation result is not correct.

F＊024 BIN multiplication（Unsigned）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊024	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**

Function		The BIN data of the register shown in Ag.2 is multiplied to the BIN data of the register shown in Ag.3. The result is stored in register (2Words) that Ag.1 shows.
Content of operation		BIN $\begin{Bmatrix} \text{D1,D1+1} \\ \text{d1,d1+1} \end{Bmatrix} \leftarrow \begin{Bmatrix} \text{D2} \\ \text{d2} \\ \text{K2} \end{Bmatrix} \times \begin{Bmatrix} \text{D3} \\ \text{d3} \\ \text{K3} \end{Bmatrix}$
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:0～65535
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:0～65535
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Flag	There is no change.



When input X000 is turned on, the product of BIN data of register D01B and data of register D04F is requested. The result is stored in 2Words of register D106 and D105.

D04F:(5130) 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0

×

D01B:(44) 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0

↓

MSB LSB

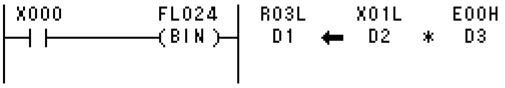
D106 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

D105:(225720) 0 1 1 1 0 0 0 1 1 0 1 1 1 0 0 0

（D106:上位Words、D105:下位Words）

Even when byte register is specified for register D1, data is stored as 4 bytes longs.

Example.1 When registers D1, D2, D3 are byte register



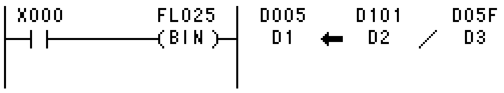
X01L（100）× E00H（50）→ R04H and R04L, R03H and R03L（5000）

This instruction cannot BIN multiplication with the sign.
Please execute the BIN multiplication with the signed by the F＊103 instruction.

F＊025 BIN division（Unsigned）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊025	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**

Function		The 2 words length BIN data of the register that Ag.2 shows is divided with BIN data of the register that Ag.3 shows. The quotient and the remainder are stored in the register that Ag.1 shows.
Content of operation		BIN (D1 / d1) ← (D2,D2+1 / d2,d2+1 / K2) ／ (D3 / d3 / K3) D1+1（remainder） d1+1
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:0～65535
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:0～65535
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Flag	A002:When the quotient exceeds 63555, it turns on.



When input X000 is turned on, the 2Words data of register D101 and D100 is divided with the BIN data of register D05F. The quotient of the result is stored in register D005. The remainder is stored in D006.

D102（High-order）	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
D101:(6362145)（Low-order）	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1

／

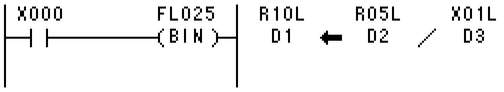
D05F:(22018)	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	0

↓

	MSB															LSB
D005:(288)（Quotient）	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
D006:(20961)（Remainder）	0	1	0	1	0	0	0	1	1	1	1	0	0	0	0	1

Even if byte register is specified, register D2 is divided as 2Words data.

Example.1 When registers D1, D2, D3 are byte register

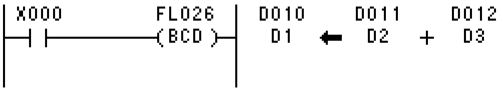


R06H and R06L, R05H and R05L ／ X01L → R10H and R10L（Quotient）
R11H and R11L（Remainder）

F∗026 BCD addition

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BCD	F∗026	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**

Function		The BCD data of the register that Ag.2 shows, the BCD data of the register that Ag.3 shows are added and are stored in the register that Ag.1 shows.
Content of operation		BCD (D1 d1) ← (D2 d2 K2) + (D3 d3 K3)
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:0～9999
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:0～9999
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.



When input X000 is turned on, the sum of the data of register D011 and the BCD data of register D012 is requested. The result is stored in register D010.

D011:(3130H)	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0

+

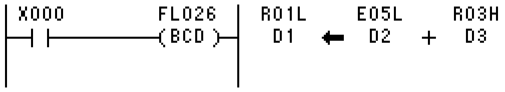
D012:(8375H)	1	0	0	0	0	0	1	1	0	1	1	1	0	1	0	1

↓

	MSB															LSB
D010:(1505H)	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	1
A000:																1

When byte register is used for register D1, the sum in low-order 1 byte is stored in D1. High-order 1 byte is disregarded.

Example.1 When register D1, D2 are byte register



The BCD data of E05L and R03H is added, and low-order 1 byte is stored in R01L.

E05L							
5				9			
0	1	0	1	1	0	0	1

＋

R03H							
8				1			
1	0	0	0	0	0	0	1

→

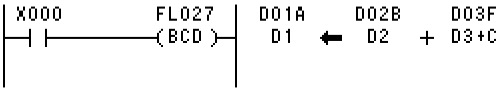
R01L							
4				0			
0	1	0	0	0	0	0	0

A000:OFF

Even if carry is generated (The operation result is 100 or more) when all registers are byte registers, A000 is not turned on.
When register D2 or D3 is byte register, it is operated as word register that high rank 1 byte is 0.

F∗027 BCD addition（with carry）

<table>
<tr><th rowspan="2" colspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">BCD</td><td>F＊027</td><td>D1
d1</td><td>D2
d2
K2</td><td>D3
d3
K3</td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The BCD data of the register that Ag.2 shows, the BCD data of the register that Ag.3 shows, and carry flag (A000) are added in BCD, and is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$BCD\begin{Bmatrix}D1\\d1\end{Bmatrix}\leftarrow\begin{Bmatrix}D2\\d2\\K2\end{Bmatrix}+\begin{Bmatrix}D3\\d3\\K3\end{Bmatrix}+C$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register
K2 Constant:0～9999</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register
K3 Constant:0～9999</td></tr>
<tr><td rowspan="6">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Carry flag</td><td colspan="4">A000:When carrying is caused in the result, it turns on.</td></tr>
<tr><td>Zero flag</td><td colspan="4">A006:When result is 0, it turns on.</td></tr>
<tr><td>Sign flag</td><td colspan="4">A007:When MSB of the result is 1, it turns on.</td></tr>
</table>



When input X000 is turned on, the data of register D02B, the BCD data of register D03F, and carry flag are added. The result is stored in register D01A.

D02B:(1274H) | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

＋

D03F:(3745H) | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

＋

A000: | 1 |

↓

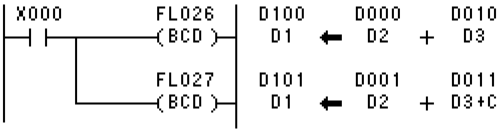
MSB LSB

D01A:(5020H) | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

A000: | 0 |

When byte register is used for register D1, the sum in low-order 1 byte is stored in D1. High-order 1 byte is disregarded.

Example.1 When you add BCD more than 4 shapes



Register (D001 and D000) and register (D011 and D010) is added. The result is stored in register (D101 and D100).

Computing range: 0～99999999

Addition of high-order 4 digits（FL027）

	D001	0037H
	D011	0542H
+)	A000	1
	D101	0580H
	A000	0

Addition of low-order 4 digits（FL026）

	D000	8320H
	D010	7495H
+)		
	D100	5815H
	A000	1

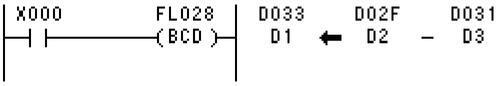
	378320H
	5427495H
+)	
	5805815H

F∗028 BCD subtraction

Symbol	Code	Argument Ag.1	Argument Ag.2	Argument Ag.3
BCD	F∗028	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**

Function	The BCD data of the register shown in Ag.3 is subtracted from the BCD data of the register shown in Ag.2. The result is stored in the register that Ag.1 shows.
Content of operation	$BCD\begin{pmatrix}D1\\d1\\ \end{pmatrix} \leftarrow \begin{pmatrix}D2\\d2\\K2\end{pmatrix} - \begin{pmatrix}D3\\d3\\K3\end{pmatrix}$
Range of use of Ag.1	D1 (Direct register) :All register d1 (Indirect register) :All register
Range of use of Ag.2	D2 (Direct register) :All register d2 (Indirect register) :All register K2 Constant:0～9999
Range of use of Ag.3	D3 (Direct register) :All register d3 (Indirect register) :All register K3 Constant:0～9999

After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.

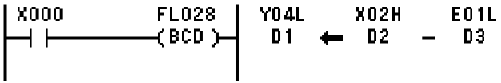


When input X000 is turned on, the BCD data of register D031 is subtracted from data of register D02F. The difference is stored in register D033.

D02F:(7814H)	0	1	1	1	1	0	0	0	0	0	0	1	0	1	0	0
—																
D031:(3972H)	0	0	1	1	1	0	0	1	0	1	1	1	0	0	1	0
↓	MSB															LSB
D033:(3842H)	0	0	1	1	1	0	0	0	0	1	0	0	0	0	1	0
A000:																0

When byte register is used for register D1, the sum in low-order 1 byte is stored in D1. High-order 1 byte is disregarded.

Example.1 When register D1, D2 are byte register



The BCD data of X02H is subtracted from E01L.
The difference is stored in Y04L.

X02H — E01L → Y04L

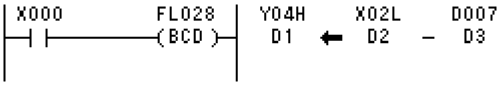
3				8			
0	0	1	1	1	0	0	0

5				6			
0	1	0	1	0	1	1	0

8				2			
1	0	0	0	0	0	1	0

When all registers are byte registers, even if the digit fall is generated, carry A000 is not turned on.

Example.2 When the Words register exists together to byte register



A000:OFF

High-order 2 digits are disregarded.

X02L — D007 → Y04H

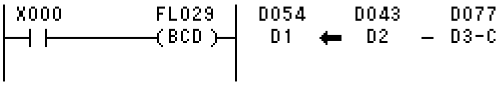
5				1			
0	1	0	1	0	0	0	1

1				0				3				2			
0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0

1				9			
0	0	0	1	1	0	0	1

F∗029 BCD subtraction（with borrow）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BCD	F∗029	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The BCD data and carry flag of the register that Ag.3 shows are subtracted from the BCD data of the register that Ag.2 shows. The result is stored in the register that Ag.1 shows.			
Content of operation	$\mathrm{BCD}\begin{Bmatrix}D1\\d1\\ \end{Bmatrix} \leftarrow \begin{Bmatrix}D2\\d2\\K2\end{Bmatrix} - \begin{Bmatrix}D3\\d3\\K3\end{Bmatrix} - C$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:0～9999			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:0～9999			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.		



When input X000 is turned on, the BCD data and carry flag of register D077 are subtracted from the data of register D043. The difference is stored in register D054.

D043:(6435H) 0 1 1 0 0 1 0 0 0 0 1 1 0 1 0 1

−

D077:(1268H) 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0

−

A000: 1

↓

MSB LSB

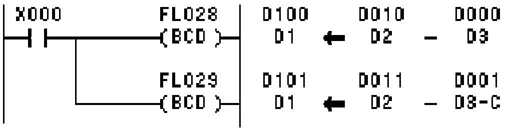
D054:(5166H) 0 1 0 1 0 0 0 1 0 1 1 0 0 1 1 0

A000: 0

When byte register is used for register D1, low-order 1 byte of result is stored in D1. High-order 1 byte is disregarded.

Example.1 When you subtract BCD of 4 digits or more

The value of register (D001 and D000) is subtracted from the value of register (D011 and D010), and the difference is stored in register (D101 and D100).



The data of low-order 4 digits is subtracted.

The data of high-order 4 digits is subtracted.

Subtraction of four high-order digits (FL029) Subtraction of four low-order digits (FL028)

	D011	0075H
	D001	0031H
−)	A000	1
	D101	0043H
	A000	0

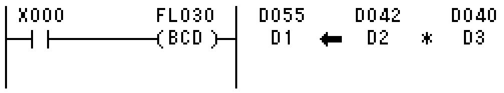
	D010	1375H
	D000	8143H
−)		
	D100	3232H
	A000	1

	751375H
	318143H
−)	
	433232H

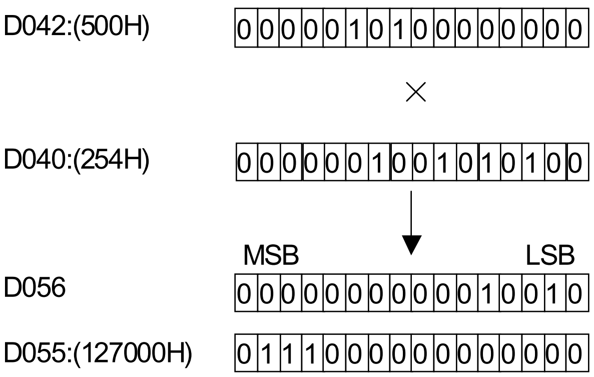
Note) When borrow is generated by operating FL29, operation result is not correct.

F∗030 BCD multiplication

<table>
<tr><th rowspan="2" colspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">BCD</td><td>F∗030</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b>
<b>K2</b></td><td><b>D3</b>
<b>d3</b>
<b>K3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The BCD data of the register shown in Ag.2 is multiplied to the BCD data of the register shown in Ag.3.The result is stored in register (2Words) that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\mathrm{BCD}\begin{pmatrix}\mathrm{D1,D1+1}\\ \mathrm{d1,d1+1}\end{pmatrix} \leftarrow \begin{pmatrix}\mathrm{D2}\\ \mathrm{d2}\\ \mathrm{K2}\end{pmatrix} \times \begin{pmatrix}\mathrm{D3}\\ \mathrm{d3}\\ \mathrm{K3}\end{pmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register
K2 Constant:0～9999</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register
K3 Constant:0～9999</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



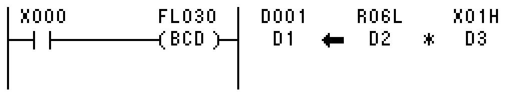
When input X000 is turned on, the product of BCD data of register D042 and register D040 is requested. The result is stored in 2Words of register D056 and D055.



D056:High-order word
D055:Low-order word

Even if byte register is specified for register D1, it stores it as 4 bytes longs Data.

Example.1 When registers D1, D2, D3 are byte register

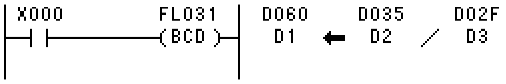


R06L（12H）× X01H（34H）→ D002 and D001（408H）

F＊031 BCD division

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BCD	F＊031	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The 2words length BCD data of register that Ag.2 shows is divided with the BCD data of the register that Ag.3 shows. The quotient and the remainder are stored in the register that Ag.1 shows.			
Content of operation	BCD (D1 d1) ← (D2,D2+1 d2,d2+1 K2) ／ (D3 d3 K3) D1+1 (Remainder) d1+1			
Range of use of Ag.1	D1 (Direct register) :All register d1 (Indirect register) :All register			
Range of use of Ag.2	D2 (Direct register) :All register d2 (Indirect register) :All register K2 Constant:0～99999			
Range of use of Ag.3	D3 (Direct register) :All register d3 (Indirect register) :All register K3 Constant:0～9999			

After operation		
	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Flag	A002: When quotient exceeds 9999, it turns it on.

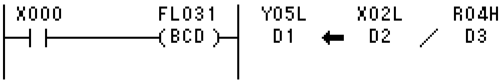


When input X000 is turned on, the 2Words data of register D036 and D035 is divided with the BCD data of register D02F. The quotient is stored in register D060 and the remainder is stored in D061.

D036 (High-order)	0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0
D035:(645071H) (Low-order)	0 1 0 1 0 0 0 0 0 1 1 1 0 0 0 1
	／
D02F:(375H)	0 0 0 0 0 0 1 1 0 1 1 1 0 1 0 1
	↓
	MSB LSB
D060:(1720H) (Quotient)	0 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0
D061:(71H) (Remainder)	0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1

Even if byte register is specified, register D2 is divided as 2Words data.

Example.1 When registers D1, D2, D3 is byte register.

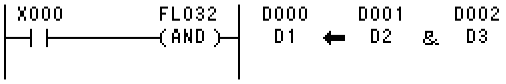


X02H and X02L ／ R04H → Y05H and Y05L (Quotient)
Y06H and Y06L (Remainder)

6. Logical operation instruction

F∗032 Logical product

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
AND	F∗032	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The logical product of data of the register that Ag.2 shows and data of the register that Ag.3 shows is requested. The result is stored in the register that Ag.1 shows.			
Content of operation	$\begin{pmatrix} D1 \\ d1 \\ \ \end{pmatrix} \leftarrow \begin{pmatrix} D2 \\ d2 \\ K2 \end{pmatrix} \wedge \begin{pmatrix} D3 \\ d3 \\ K3 \end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation: Contents of Ag.1	Operation result			
After operation: Contents of Ag.2	There is no change.			
After operation: Contents of Ag.3	There is no change.			
After operation: Flag	There is no change.			

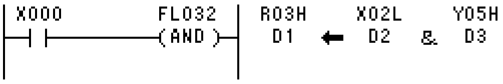


When input X000 is turned on, the logical product of data of register D001 and data of register D002 is requested. The result is stored in register D000.

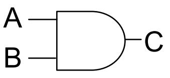
D001:(1564H)	0	0	0	1	0	1	0	1	0	1	1	0	0	1	0	0
										∧						
D002:(00FFH)	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
										↓						
D000:(0064H)	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0

When byte register is used for register D1, low-order 1 byte of operation result is stored in D1. High-order 1 byte is disregarded.

Example.1 When registers D1, D2, D3 are byte register

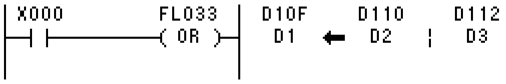


X02L								∧	Y05H								→	R03H							
79									198									70							
0	1	0	0	1	1	1	1		1	1	0	0	0	1	1	0		0	1	0	0	0	1	1	0

Logical product	Truth value	A	B	C
		0	0	0
		1	0	0
		0	1	0
		1	1	1

F∗033 Logical addition

Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
OR		F∗033	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function		The logical add of data of the register that Ag.2 shows and data of the register that Ag.3 shows is requested. The result is stored in the register that Ag.1 shows.			
Content of operation		$\begin{pmatrix} D1 \\ d1 \\ \end{pmatrix} \leftarrow \begin{pmatrix} D2 \\ d2 \\ K2 \end{pmatrix} \vee \begin{pmatrix} D3 \\ d3 \\ K3 \end{pmatrix}$			
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation	Contents of Ag.1	Operation result			
	Contents of Ag.2	There is no change.			
	Contents of Ag.3	There is no change.			
	Flag	There is no change.			

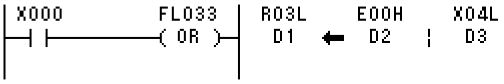


When input X000 is turned on, the logical add of the data of register D110 and the data of register D112 is requested. The result is stored in register D10F.

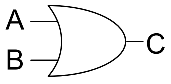
D110:(3564H)	0	0	1	1	0	1	0	1	0	1	1	0	0	1	0	0
	∨															
D112:(80A5H)	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1
	↓															
D10F:(B5E5H)	1	0	1	1	0	1	0	1	1	1	1	0	0	1	0	1

When byte register is used for register D1, low-order 1 byte of result is stored in D1.High-order 1 byte is disregarded.

Example.1 When registers D1, D2, D3 are byte register

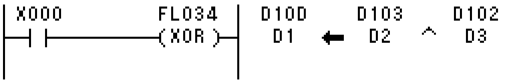


E00H		X04L		R03L
146	∨	21	→	151
1 0 0 1 0 0 1 0		0 0 0 1 0 1 0 1		1 0 0 1 0 1 1 1

Logical add	Truth value	A	B	C
		0	0	0
		1	0	1
		0	1	1
		1	1	1

F∗034 Exclusive logical add

<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">AND</td><td>F∗034</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b>
<b>K2</b></td><td><b>D3</b>
<b>d3</b>
<b>K3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The exclusive logical add of the data of the register that Ag.2 shows and the register that Ag.3 shows is requested. The result is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\left(\begin{matrix}D1\\d1\end{matrix}\right) \leftarrow \left(\begin{matrix}D2\\d2\\K2\end{matrix}\right) \oplus \left(\begin{matrix}D3\\d3\\K3\end{matrix}\right)$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register
K2 Constant:-32768～32767</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register
K3 Constant:-32768～32767</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>

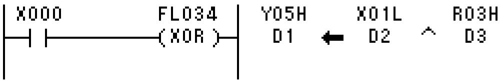


When input X000 is turned on, the exclusive logical add of the data of register D103 and data of register D102 is requested. The result is stored in register D10D.

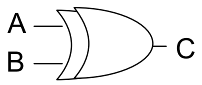
D103:(1564H)	0	0	0	1	0	1	0	1	0	1	1	0	0	1	0	0
⊕																
D102:(80A5H)	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1
↓																
D10D:(95C1H)	1	0	0	1	0	1	0	1	1	1	0	0	0	0	0	1

When byte register is used for register D1, low-order 1 byte of result is stored in D1. High-order byte is disregarded.

Example.1 When registers D1, D2, D3 are byte register



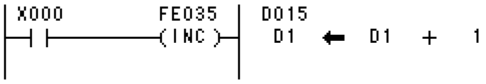
X01L	⊕	R03H	→	Y05H
79		198		70
0 1 0 0 1 1 1 1		1 1 0 0 0 1 1 0		1 0 0 0 1 0 0 1

Exclusive logical add	Truth value	A	B	C
		0	0	0
		1	0	1
		0	1	1
		1	1	0

7. Operation instruction

F∗035 Increment

<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">INC</td><td>F∗035</td><td>D1
d1</td><td></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">1 is added to the value of register that Ag.1 shows. The result is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\begin{pmatrix} D1 \\ d1 \end{pmatrix} + 1 \rightarrow \begin{pmatrix} D1 \\ d1 \end{pmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1 （Direct register） :All register
d1 （Indirect register） :All register</td></tr>
<tr><td rowspan="3">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 stands up for turning on from turning off, 1 is added to the value of D015.

Before operation
D015:(2329)

0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1

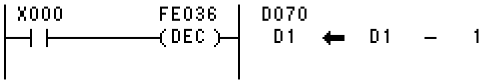
After operation
D015:(2330)

0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0

The words register is a range of (-32768～32767).
Byte register is operated within the range of (0～255).

F∗036 Decrement

<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">DEC</td><td>F∗036</td><td>D1
d1</td><td></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">1 is subtracted from the value of the register that Ag.1 shows. The result is stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\begin{pmatrix} D1 \\ d1 \end{pmatrix} - 1 \rightarrow \begin{pmatrix} D1 \\ d1 \end{pmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1 （Direct register） :All register
d1 （Indirect register） :All register</td></tr>
<tr><td rowspan="3">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 stands up for turning on from turning off, 1 is subtracted from register D070.

Before operation
D070:(3140)

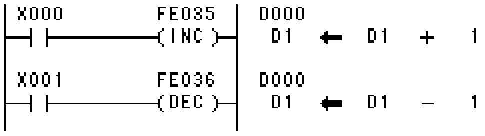
0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0

After operation
D070:(3139)

0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1

The Words register is a range of (-32768～32767).
Byte register is operated within the range of (0～255).

Example.1 Addition and subtraction counter

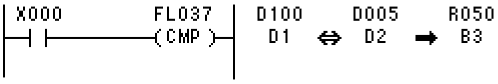


When X000 stands up for turning on from turning off, 1 is added to value of register D000.
When X001 stands up for turning on from turning off, 1 is subtracted from register D000.

8. Comparison instruction

F＊037 Comparison（Unsigned）

<table>
<tr><th rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td>CMP</td><td>F＊037</td><td><b>D1</b>
<b>d1</b>
<b>K1</b></td><td><b>D2</b>
<b>d2</b>
<b>K2</b></td><td><b>B3</b></td></tr>
<tr><td>Function</td><td colspan="4">The BIN data of the register that Ag.1 shows is compared with the BIN data of the register that Ag.2 shows. The result is stored in the relay that Ag.3 shows.</td></tr>
<tr><td>Content of operation</td><td colspan="4">$\begin{Bmatrix}D1\\d1\\K1\end{Bmatrix} \Leftrightarrow \begin{Bmatrix}D2\\d2\\K2\end{Bmatrix} \rightarrow \begin{Bmatrix}B3\end{Bmatrix}$</td></tr>
<tr><td>Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register
K1 Constant:0～65535</td></tr>
<tr><td>Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register
K2 Constant:0～65535</td></tr>
<tr><td>Range of use of Ag.3</td><td colspan="4">B3 Relay
:All relay（Even number address）</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="3">There is no change.</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="3">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="3"><table><tr><th>Relay</th><th>ON</th><th>OFF</th></tr><tr><td>B3</td><td>Ag.1≦Ag.2</td><td>Ag.1＞Ag.2</td></tr><tr><td>B3+1</td><td>Ag.1＝Ag.2</td><td>Ag.1≠Ag.2</td></tr></table></td></tr>
<tr><td>Flag</td><td colspan="3">There is no change.</td></tr>
</table>



When input X000 is turned on, the BIN data of register D100 is compared with the BIN data of register D005.The result is stored in R050 and R051.

D100:(2339) 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 1

MSB LSB

D005:(9752) 0 0 1 0 0 1 1 0 0 0 0 1 1 0 0 0

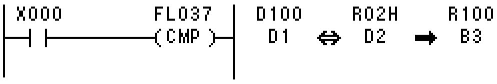
↓

R050:ON、R051:OFF

When the BCD data is compared, D1 and D2 are considered the BIN data and compared.

When byte register is used, registers are compared as a word register in which high-order 1 byte is assumed to be 0.

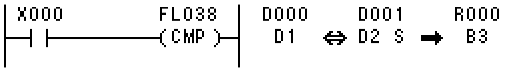
Example.1 When register D2 is byte register



D100	R02H	Result	R100	R101
72 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0	202 1 1 0 0 1 0 1 0	＜	ON	OFF
202 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0	202 1 1 0 0 1 0 1 0	＝	ON	ON
2304 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0	174 1 0 1 0 1 1 1 0	＞	OFF	OFF

F＊038 Comparison（Signed）

<table>
<tr><td rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td>CMP</td><td>F＊038</td><td><b>D1</b>
<b>d1</b>
<b>K1</b></td><td><b>D2</b>
<b>d2</b>
<b>K2</b></td><td><b>B3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="3">The BIN data of the register that Ag.1 shows is compared with the BIN data of the register that Ag.2 shows. The result is stored in the relay that Ag.3 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="3">$\begin{pmatrix}D1\\d1\\K1\end{pmatrix}_S \Leftrightarrow \begin{pmatrix}D2\\d2\\K2\end{pmatrix}_S \rightarrow \begin{pmatrix}B3\end{pmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="3">D1（Direct register）:All register
d1（Indirect register）:All register
K1 Constant:-32768～32767</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="3">D2（Direct register）:All register
d2（Indirect register）:All register
K2 Constant:-32768～32767</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="3">B3 Relay
:All relay（Even number address）</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="3">There is no change.</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="3">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="3"><table><tr><td>Relay</td><td>ON</td><td>OFF</td></tr><tr><td>B3</td><td>Ag.1≦Ag.2</td><td>Ag.1＞Ag.2</td></tr><tr><td>B3+1</td><td>Ag.1＝Ag.2</td><td>Ag.1≠Ag.2</td></tr></table></td></tr>
<tr><td>Flag</td><td colspan="3">There is no change.</td></tr>
</table>



When input X000 is turned on, the BIN data of register D000 is compared with the BIN data of register D001.
The result is stored in R000 and R001.

D000:(3140) 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0

MSB LSB

D001:(6241) 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1

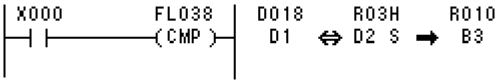
↓

R000:ON、R001:OFF

When byte register is used, registers are compared as a word register in which high-order 1 byte is assumed to be 0.

Please do not execute the comparison of BCD data.

Example.1 When register D2 is bite register

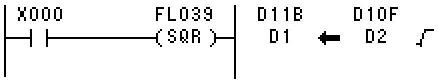


D018	R03H	Result	R010	R011
-73 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1	153 1 0 0 1 1 0 0 1	＜	ON	OFF
-174 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 0	184 1 0 1 1 1 0 0 0	＜	ON	OFF
-3140 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 0	83 0 1 0 1 0 0 1 1	＜	ON	OFF
206 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1 0	206 1 1 0 0 1 1 1 0	＝	ON	ON
456 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0	226 1 0 1 0 1 1 1 0	＞	OFF	OFF

9. Operation instruction

F＊039 Square root（Unsigned）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
SQR	F＊039	**D1** **d1**	**D2** **d2**	
Function	Square root of the BIN data of the register that Ag.2 shows is requested. The result is stored in the register that Ag.1 shows.			
Content of operation	$\begin{pmatrix} D1 \\ d1 \end{pmatrix} \leftarrow \sqrt{\begin{pmatrix} D2 \\ d2 \end{pmatrix}}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
After operation: Contents of Ag.1	Operation result			
After operation: Contents of Ag.2	There is no change.			
After operation: Flag	There is no change.			

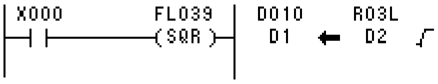


When input X000 is turned on, the square root of the BIN data of register D10F is requested. The result is stored in register D11B.

D10F:(6561)	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1
D11B:(81)	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1

The round down is done to the numerical value below the decimal point of the square root.

Example.1 When register D2 is byte register

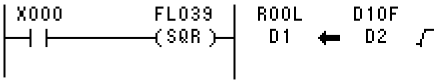


R03L → D010

R03L							
217							
1	1	0	1	1	0	0	1

D010															
14															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Example.2 When register D1 is byte register



D10F → R00L

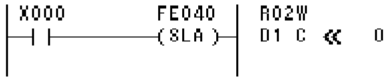
D10F															
6562															
0	0	0	1	1	0	0	1	1	0	1	0	0	0	1	0

R00L							
81							
0	1	0	1	0	0	0	1

10. Shift instruction

F∗040 Arithmetic left shift

Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
SLA		F∗040	**D1** **d1**		
Function		The value of the most significant bit (MSB) of the register that Ag.1 shows is set in carry flag, and the left shifts by 1 bit, and 0 sets of the least significant bit (LSB).			
Content of operation		$C \leftarrow \begin{pmatrix} D1 \\ d1 \end{pmatrix} \leftarrow 0$			
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register			
After operation	Contents of Ag.1	Data shifted to the left of 1 bit			
	Flag	A000:MSB of Ag.1 before it shifts A006 and A007:There is no change.			



When input X000 stands up for turning on from turning off, the value of MSB of register R02W is set in A000, and after the left shifts of 1 bit, 0 is set in LSB.

Before operation

	A000		MSB															LSB
R02W:	0	←	1	0	0	1	0	1	0	1	0	1	1	0	0	1	0	1

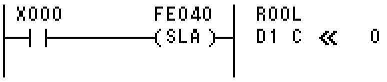
↓

After operation

	A000		MSB															LSB
R02W:	1		0	0	1	0	1	0	1	0	1	1	0	0	1	0	1	0

When byte register is used for register D1, left shift is done as 1 word data assumed that a high-order byte is 0.

Example.1 When register D1 is byte register



Data of R00L shifts left, and LSB is adjusted to 0.

A000: 0

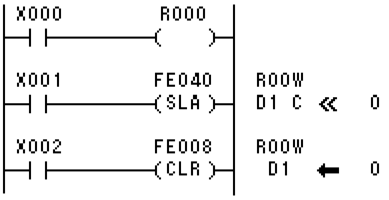
R00L							
1	0	1	0	0	1	1	1

⟶

A000: 0

R00L							
70							
0	1	0	0	1	1	1	0

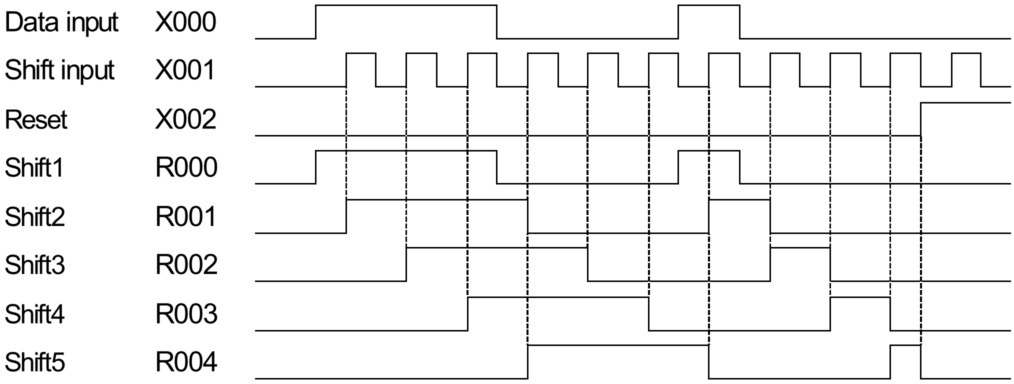
Example.2 Example as 1Words shift register



Data input

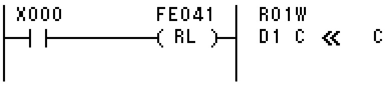
Shift input

Reset input

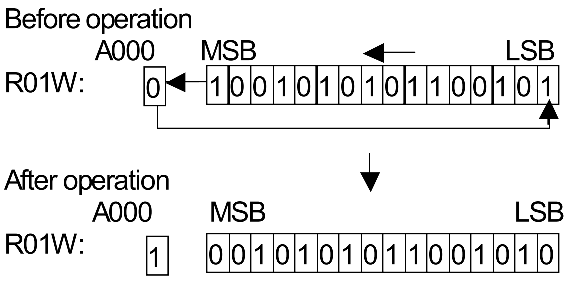


F∗041 Left rotate

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
RL	F∗041	**D1** **d1**		
Function	The value of the MSB of the register that Ag.1 shows is set in the carry flag. After the left shifts of 1 bit, the value of the carry flag before it operates is set in the LSB.			
Content of operation	C ← (D1 d1) ← C			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
After operation Contents of Ag.1	Data shifted to the left of 1 bit			
Flag	A000:MSB of Ag.1 before it shifts A006 and A007:There is no change.			

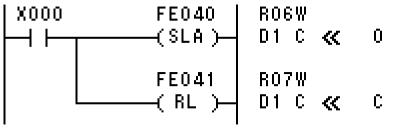


When input X000 stands up for turning on from turning off, the value of MSB of register R02W is set in A000. Data is set and after the left shifts of one bit, the value of A000 before it operates it is set in LSB.



When byte register is used for register D1, a left shift is done as 1 word data assumed that a high-order byte is 0.

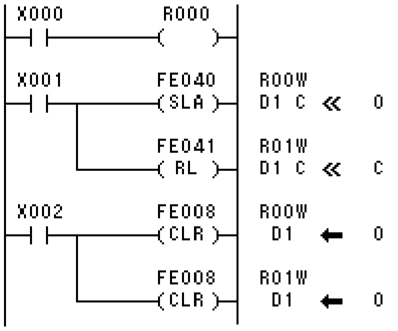
Example.1 When it shifts to the left more than one word



Low-order 1 Word is shifted to the left.

High-order 1 Word is rotate to the left

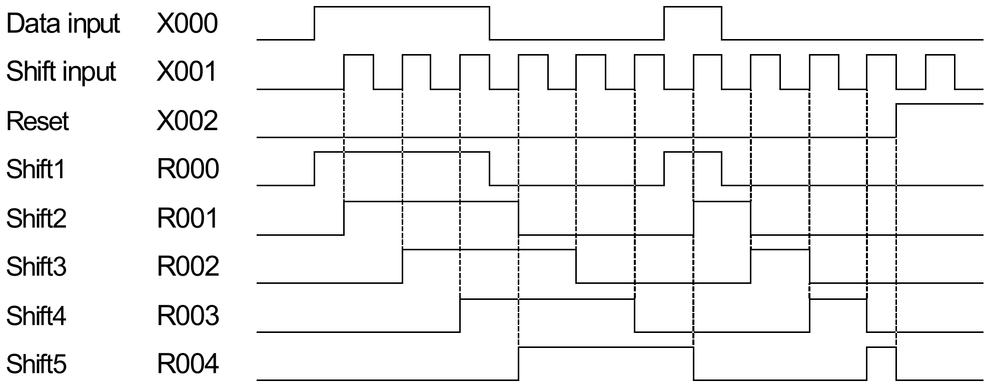
Example, 2 Example as 2 Words shift register



Data input

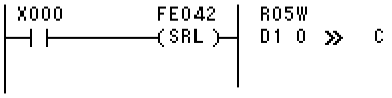
Shift input

Reset input



F∗042 Right shift

Symbol	Code	Argument Ag.1	Argument Ag.2	Argument Ag.3
SRL	F∗042	**D1** **d1**		
Function	The value of the LSB of the register that Ag.1 shows is set in carry flag, and after a right shift 1 bit, 0 is set in MSB.			
Content of operation	0 → (D1 / d1) → C			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
After operation: Contents of Ag.1	Data shifted to the right of 1 bit			
After operation: Flag	A000:LSB of Ag.1 before it shifts A006 and A007:There is no change.			



When input X000 stands up for turning on from turning off, The value of LSB of register R05W is set in A000, and after a right shift 1 bit, 0 is set in MSB.

Before operation

MSB → LSB A000

R05W: 0 0 1 0 1 0 1 0 1 1 0 0 1 0 1 0 → 1

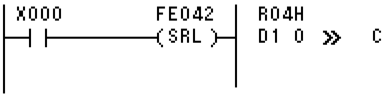
After operation

MSB LSB A000

R05W: 0 0 0 1 0 1 0 1 0 1 1 0 0 1 0 1 0

When byte register is used for register D1, a right shift is done as 1 word data of which 0 is a high-order byte.

Example.1 When register D1 is byte register



Data of R04H is shifted to the right, and MSB is adjusted to 0.

A000

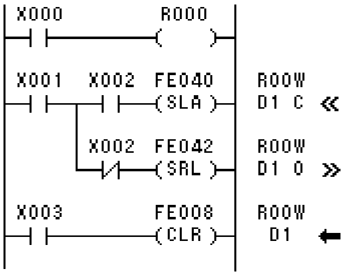
R04H
70
0 1 0 0 1 1 1 0 0

→

R04H
0 0 1 0 0 1 1 1

A000
0

Example.2 Example as 1 Word interactive shift registers

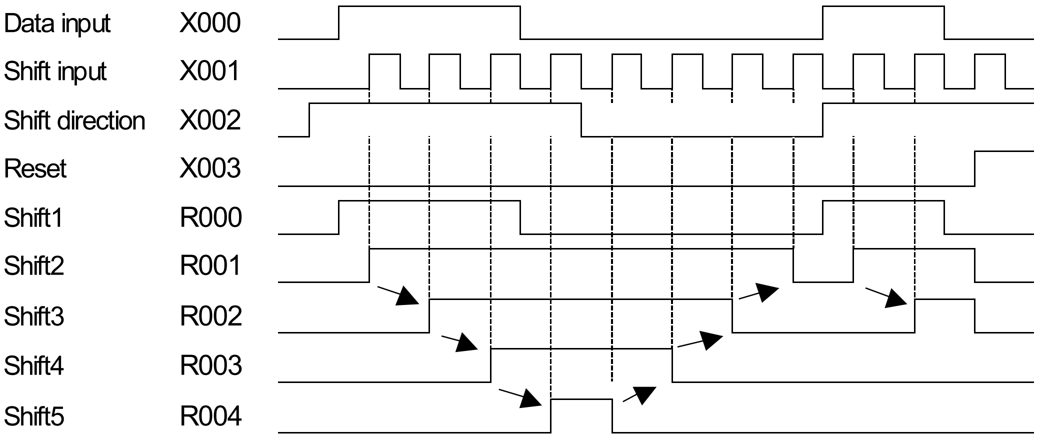


Data input

Left shift input

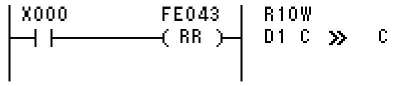
Right shift input

Reset input

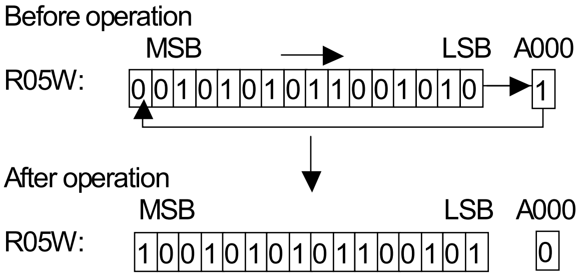


F∗043 Right rotate

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
RR	F∗043	**D1** **d1**		
Function	The value of the LSB of the register that Ag.1 shows is set in the carry flag. After data is shifted to the right of 1 bit, the carry flag before it shifts to the MSB is set.			
Content of operation	C → D1 d1 → C			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
After operation: Contents of Ag.1	Data shifted to the right of 1 bit			
After operation: Flag	A000:LSB of Ag.1 before it shifts A006 and A007:There is no change.			

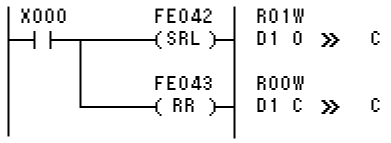


When input X000 stands up for turning on from turning off, the value of LSB of register R10W is set in A000. Data is set and after a right shift 1 bit, the value of A000 before it shifts is set in MSB.



When byte register is used, right rotate is done as 1 word data assumed that a high-order byte is 0.

Example.1 When you want to do a right shift of 1 Word or more

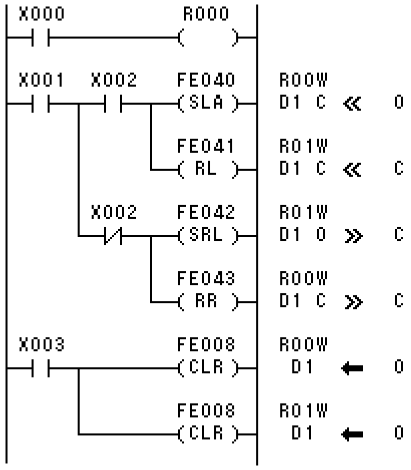


High-order 1Words is shifted to the right.

Low-order 1Words is shifted to the rotate.

Note) To achieve the digit lowering by carry, a right shift previously does the high rank.

Example.2 Example as 1Words interactive shift registers

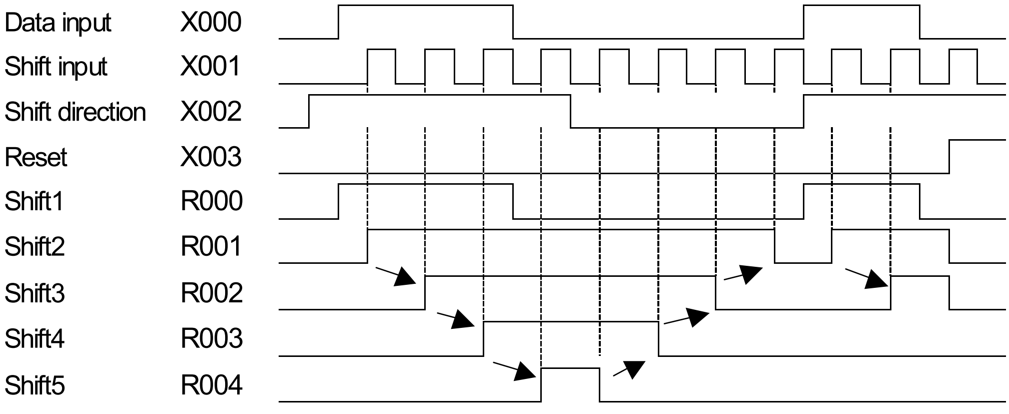


Data input

Left shift input

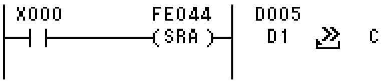
Right shift input

Reset input



F＊044 Arithmetic right shift

<table>
<tr><th rowspan="2" colspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">SRA</td><td>F＊044</td><td><b>D1</b>
<b>d1</b></td><td></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The value of the LSB of the register that Ag.1 shows is set in the carry flag. After a right shift 1 bit, the value of the MSB before it shifts is set in the MSB again.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">D1
d1 → C</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）
:All register</td></tr>
<tr><td rowspan="2">After operation</td><td>Contents of Ag.1</td><td colspan="4">Data shifted to the right of 1 bit</td></tr>
<tr><td>Flag</td><td colspan="4">A000:LSB of Ag.1 before it shifts
A006 and A007:There is no change.</td></tr>
</table>



When input X000 stands up for turning on from turning off, the value of LSB of register D005 is set in A000. Data is set and after a right shift 1 bit, the value of MSB before it shifts is set in MSB.

Before operation

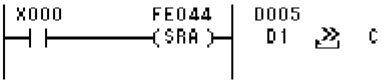
	MSB															LSB		A000
D005:	1	0	1	0	1	0	1	0	1	1	0	0	1	0	1	0	→	1

After operation

	MSB															LSB	A000
D005:	1	1	0	1	0	1	0	1	0	1	1	0	0	1	0	1	0

When byte register is used, a right shift is done as 1 word data of which 0 is a high-order byte.

Example.1　The arithmetic right shift of BIN data



If the arithmetic right shift is done to BIN data, the value becomes 1/2.

D005 → D005

-1120															
1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0

-560															
1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0

D005 → D005

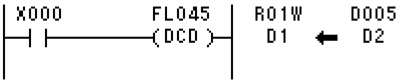
6562															
0	0	0	1	1	0	0	1	1	0	1	0	0	0	1	0

3281															
0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	1

11. Data processing instruction

F∗045 4→16 Decipherment

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
DCD	F∗045	**D1** **d1**	**D2** **d2**	
Function	Low-order 4 bits of the BIN data of the register that Ag.2 shows are deciphered to 16 bits. The result is stored in the register that Ag.1 shows.			
Content of operation	DCD $\begin{pmatrix} D1 \\ d1 \end{pmatrix} \leftarrow \begin{pmatrix} D2 \\ d2 \end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
After Contents of Ag.1	Operation result			
Contents of Ag.2	There is no change.			
Flag	There is no change.			



When input X000 is turned on, the data of register D005 is deciphered, and is stored in register R01W.

D005:(7) 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

They are disregarded.

MSB LSB

R05W:(128) 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

15……………8⑦ 6 5 4 3 2 1 0

When byte register is used for register D1, low-order 8 bits of the decipherment result are stored in D1.

Example.1

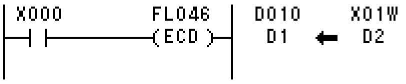
Content of decipherment

o.	Origin Data →	Conversion ahead Data
0	0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1	1 0 0 0 0 0 0 0 1	2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
2	2 0 0 0 0 0 0 1 0	4 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
3	3 0 0 0 0 0 0 1 1	8 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
4	4 0 0 0 0 0 1 0 0	16 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
5	5 0 0 0 0 0 1 0 1	32 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
6	6 0 0 0 0 0 1 1 0	64 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
7	7 0 0 0 0 0 1 1 1	128 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
8	8 0 0 0 0 1 0 0 0	256 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
9	9 0 0 0 0 1 0 0 1	512 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
10	10 0 0 0 0 1 0 1 0	1024 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
11	11 0 0 0 0 1 0 1 1	2048 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
12	12 0 0 0 0 1 1 0 0	4096 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
13	13 0 0 0 0 1 1 0 1	8192 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
14	14 0 0 0 0 1 1 1 0	16384 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	15 0 0 0 0 1 1 1 1	-32768 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

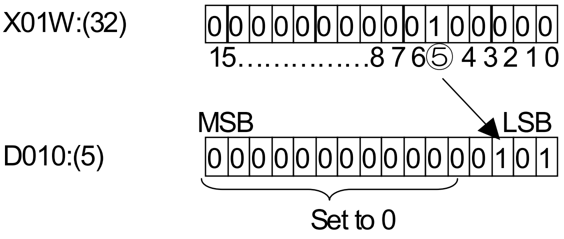
F∗046 16→4 encode

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
ECD	F∗046	**D1** **d1**	**D2** **d2**	

Function		The 16 bits data of the register that Ag.2 shows are encoded to BIN data and is stored in the register that Ag.1 shows.
Content of operation		ECD (D1 d1) ← (D2 d2)
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Flag	There is no change.



When input X000 is turned on, data of register X01W is encoded. The result is stored in register D010.



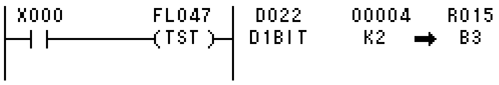
When there are two "1" or more in the content of register D2, encode is done to first “1” counted from MSB. 0 is stored when “1” is missing.

Example.1　Decode contents

No.	Origin Data →	Conversion ahead Data
0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 0
1	2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	1 0 0 0 0 0 0 0 1
2	4 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	2 0 0 0 0 0 0 1 0
3	8 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	3 0 0 0 0 0 0 1 1
4	16 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	4 0 0 0 0 0 1 0 0
5	32 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	5 0 0 0 0 0 1 0 1
6	64 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	6 0 0 0 0 0 1 1 0
7	128 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	7 0 0 0 0 0 1 1 1
8	256 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	8 0 0 0 0 1 0 0 0
9	512 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	9 0 0 0 0 1 0 0 1
10	1024 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	10 0 0 0 0 1 0 1 0
11	2048 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	11 0 0 0 0 1 0 1 1
12	4096 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	12 0 0 0 0 1 1 0 0
13	8192 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	13 0 0 0 0 1 1 0 1
14	16384 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	14 0 0 0 0 1 1 1 0
15	-32768 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15 0 0 0 0 1 1 1 1

F∗047 Bit test（Constant specification）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
TST	F∗047	**D1** **d1**	**K2**	**B3**
Function	The state of the bit at the position of the constant data shown in Ag.2 is reflected in the relay that Ag.3 shows by the data of the register shown in Ag.1.			
Content of operation	D1 d1 → B3 K2			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	K2 Constant:0～15			
Range of use of Ag.3	B3 Relay:All relay			
After operation	Contents of Ag.1	There is no change.		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	State of specified bit		
	Flag	There is no change.		



When input X000 is turned on, the state of bit 4 of the data of register D022 is reflected in relay R015.

D022: 0|0|0|1|0|1|0|1|0|1|1|0|0|1|0|0

15…………..8 7 6 5④ 3 2 1 0

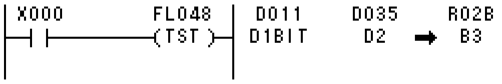
K2 = 4

R015:OFF ←

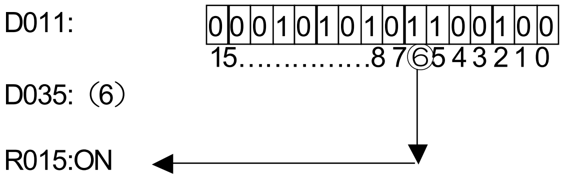
When byte register is used for register D1, high-order 8 bits test the bit as all 0.

F＊048 Bit test（Register specification）

Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
TST		F＊048	**D1** **d1**	**D2** **d2**	**B3**
Function		The state of the bit at the position of the constant data shown in Ag.2 is reflected in the relay that Ag.3 shows according to the data of the register shown in Ag.1.			
Content of operation		(D1, d1) → B3 (D2,d2)			
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3		B3 Relay:All relay			
After operation	Contents of Ag.1	There is no change.			
	Contents of Ag.2	There is no change.			
	Contents of Ag.3	State of specified bit			
	Flag	There is no change.			



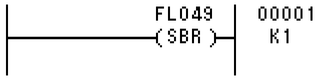
When input X000 is turning on, the state of the bit at the position that the data of register D035 shows is reflected in relay R02B by the data of register D011.



When byte register is used for register D1, high-order 8 bits are tested as all 0.
Only information in low-order 4 bits is effective to the BIN data of register D2.

12. Subroutine instruction

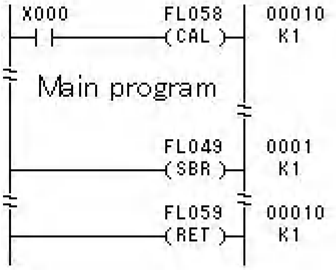
FL049 Subroutine start

<table>
<tr><td colspan="2" rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td><td rowspan="7">

The head of subroutine program 01 is shown.
The subroutine program is not cared about wherever it arranges it while programming it.</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">SBR</td><td>FL049</td><td>K1</td><td></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="3">The head of the subroutine program specified for the constant shown in Ag.1 is defined.</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="3">K1:0～31</td></tr>
<tr><td rowspan="2">After operation</td><td>Contents of Ag.1</td><td colspan="3">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="3">There is no change.</td></tr>
</table>

Note） Please put subroutine start instruction (FL049) on the head of the subroutine program.
The end must end by subroutine return instruction (FL059).

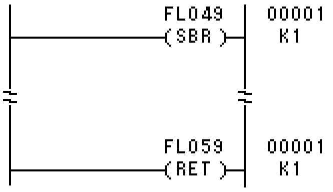
FL058 Subroutine call

<table>
<tr><td colspan="2" rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td><td rowspan="7">

X000
FL058
(CAL)
00010
K1
Main program
FL049
(SBR)
0001
K1
FL059
(RET)
00010
K1

When input X000 is turned on, subroutine program 10 is executed.</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">CAL</td><td>FL058</td><td>K1</td><td></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="3">The subroutine program specified for the constant that Ag.1 shown be executed.</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="3">K1:0～31</td></tr>
<tr><td rowspan="2">After operation</td><td>Contents of Ag.1</td><td colspan="3">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="3">There is no change.</td></tr>
</table>

FL059 Subroutine return

<table>
<tr><td colspan="2" rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td><td rowspan="7">

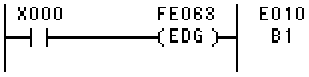
FL049
(SBR)
00001
K1
FL059
(RET)
00001
K1

The end of subroutine program 01 is shown.
The subroutine program is not cared about wherever it arranges it while programming it.</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">RET</td><td>FL059</td><td>K1</td><td></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="3">The end of the subroutine program specified for the constant shown in Ag.1 is defined.</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="3">K1:0～31</td></tr>
<tr><td rowspan="2">After operation</td><td>Contents of Ag.1</td><td colspan="3">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="3">There is no change.</td></tr>
</table>

13. Data processing instruction

FE063 1 scan ON

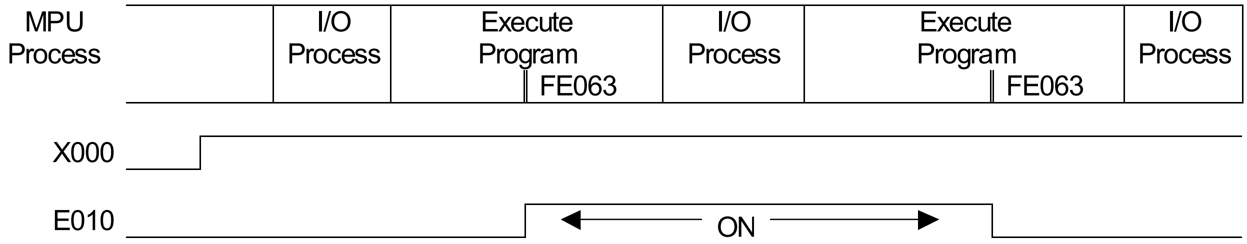
Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
EDG		FE063	B1		
Function		The relay shown in Ag.1 is turned on for 1 scanning.			
Range of use of Ag.1		B1 Relay:All relay			
After operation	Contents of Ag.1	It turns it on only by 1 scanning.			
	Flag	There is no change.			



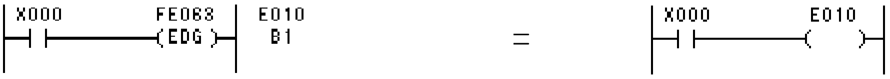
When input X000 stands up for turning on from turning off, relay E010 is turned on. After this instruction is executed by the following scanning, E010 is turned off.

FL063 cannot be used.

Operation timing diagram



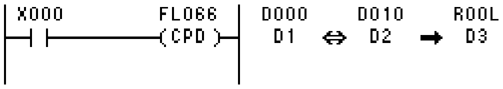
The circuit that uses the differentiation relay for this instruction and the coil becomes the same operation.



At the processing speed of MPU, the circuit that uses the differentiation relay for the coil becomes early.
We will recommend the use of the above -mentioned right circuit.

F＊066 Comparison between table & data

<table>
<tr><th rowspan="2" colspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">CPD</td><td>F＊066</td><td>D1
d1
K1</td><td>D2
d2</td><td>D3
d3</td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The data of the register shown in Ag.1 is compared with the content of the table of 8 words whose register shown in Ag.2 is head. The agreement and the disagreement result are bitting stored in the register that Ag.3 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">D1
d1 ⇔ D2,D2+1..D2+7 → D3
K1 d2,d2+1..d2+7 → d3</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register
K1 Constant:-32768～32767</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">Comparison result</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



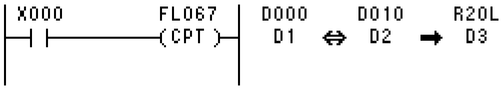
When input X000 is turned on, the data of register D000 is compared with the data of register D010～D017. The correspondence bit of register R00L is turned on in case of the agreement. If it is a disagreement, it is turned off.

			Value		R00L	
D000	⇔	D010	300	→	R000	ON
300		D011	200	→	R001	OFF
		D012	250	→	R002	OFF
		D013	300	→	R003	ON
		D014	1000	→	R004	OFF
		D015	2000	→	R005	OFF
		D016	2300	→	R006	OFF
		D107	10000	→	R007	OFF

Even if byte register is specified, registers D1 and D2 are compared as Word data.

F＊067 Comparison between table & table

<table>
<tr><th rowspan="2" colspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">CPT</td><td>F＊067</td><td><b>D1
d1
K1</b></td><td><b>D2
d2</b></td><td><b>D3
d3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The data shown in Ag.1 is compared with the content of the table of 8 words whose register shown in Ag.2 is head. The agreement and the disagreement result are biting stored in the register that Ag.3 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">(D1,D1+1..D1+7
d1,d1+1..d1+7)
CMP. → (D3 d3)
(D2,D2+1..D2+7
d2,d2+1..d2+7)</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">Comparison result</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 is turned on, data of register D000～D007 is compared with data of register D010～D017. If the comparison result is corresponding, the correspondence bit of register R20L is turned on. If the result is a disagreement, it is turned off.

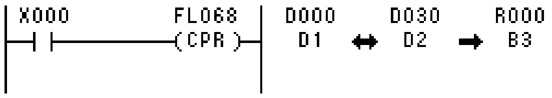
	Value			Value		R20L	
D000	300	⇔	D010	300	→	R200	ON
D001	-500		D011	200	→	R201	OFF
D002	300		D012	250	→	R202	OFF
D003	300		D013	300	→	R203	ON
D004	-400		D014	1000	→	R204	OFF
D005	800		D015	2000	→	R205	OFF
D005	2300		D016	2300	→	R206	ON
D007	20000		D107	10000	→	R207	OFF

Even if byte register is specified, registers D1 and D2 are compared as Word data.

F＊068 Range comparison（Signed）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
CPR	F＊068	**D1** **d1**	**D2** **d2**	**B3**

Function		The data of the register shown in Ag.1 is assumed to be section comparison data, and is compared with the BIN data of the register shown in Ag.2. The comparison result is stored in the relay that Ag.3 shows.
Content of operation		$\begin{pmatrix} D1 \\ d1 \end{pmatrix} \leqq \begin{pmatrix} D2 \\ d2 \end{pmatrix} \leqq \begin{pmatrix} D1+1 \\ d1+1 \end{pmatrix} \rightarrow B3$
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
Range of use of Ag.3		B3 Relay :All relay（Even number address）
After operation	Contents of Ag.1	There is no change.
	Contents of Ag.2	There is no change.
	Contents of Ag.3	Comparison result →
	Flag	There is no change.



When input X000 is turned on, data of register D030 is compared in the section with data of register D000 and D001. The result is stored in R000 and R001.

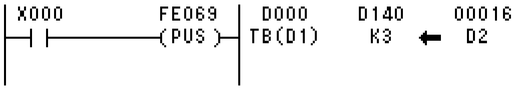
D000	D030	D001		R000	R001
100	300	1000	→	ON	ON
-2000	4000	2000	→	ON	OFF
1000	100	5000	→	OFF	OFF
0	3000	10000	→	ON	ON

Even if byte register is specified for register D1, the comparison is done as word data.

D2	D2＜D1	D1≦D2≦D1+1	D1+1≦D2
B3	OFF	ON	ON
B3+1	OFF	ON	OFF

F∗069 FIFO push

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
PUS	F∗069	**D1** **d1**	**D2** **d2**	**K3**
Function	The data of the register shown in Ag.1 is assumed to be section comparison data, and it is compared with the BIN data of the register shown in Ag.2. The comparison result is stored in the relay that Ag.3 shows.			
Content of operation	$\begin{Bmatrix} D1,D1+1..D1+K3 \\ d1,d1+1..d1+K3 \end{Bmatrix} \leftarrow \begin{Bmatrix} D2 \\ d2 \end{Bmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	K3 Constant:1～255			
After operation: Contents of Ag.1	Ag.2 is forwarded to the FIFO register.			
After operation: Contents of Ag.2	There is no change.			
After operation: Contents of Ag.3	There is no change.			
After operation: Flag	There is no change.			



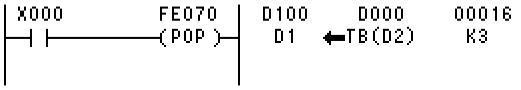
When input X000 stands up for turning on from turning off, the data of register D140 is transmitted to the FIFO register of D000～D00F of 16 words. 1 increases the value of the pointer.

		Before	After
D140			
2255H	D000	3370H	3370H
	D001	4468H	4468H
	D002	0	2255H
	D003	0	0
	D004	0	0
Pointer	D00F	2	3

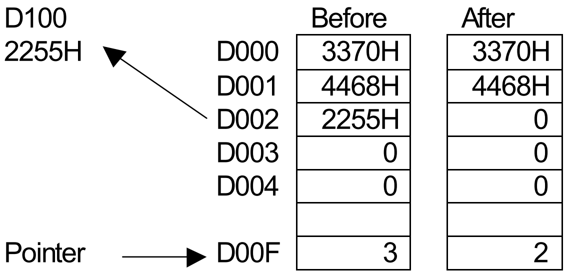
The last register of FIFO shows the pointer (storage number).

F∗070 FIFO pop

<table>
<tr><td colspan="2" rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">POP</td><td>F∗070</td><td>**D1**
d1</td><td>**D2**
d2</td><td>**K3**</td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The data of the register shown in Ag.1 is assumed to be section comparison data, and is compared with the BIN data of the register shown in Ag.2. The comparison result is stored in the relay that Ag.3 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\begin{Bmatrix}D1\\d1\end{Bmatrix} \leftarrow \begin{Bmatrix}D2,D2+1..D2+K3\\d2,d2+1..d2+K3\end{Bmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">K3 Constant:1～255</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Content of FIFO register that does pop</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">Pop does the FIFO register.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 stands up for turning on from turning off, the data is done from the FIFO register of D000～D00F of 16 words and pop is done. After this is forwarded to D100, the pointer is decreased.

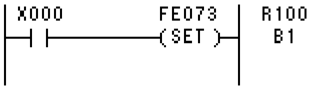


	Before	After
D000	3370H	3370H
D001	4468H	4468H
D002	2255H	0
D003	0	0
D004	0	0
D00F	3	2

The last register of FIFO shows the pointer (storage number).

F∗073 Set coil

Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
SET		F∗073	B1		
Function		The relay shown in Ag.1 is turned on.			
Range of use of Ag.1		B1 Relay:All relay			
After operation	Contents of Ag.1	It is turned on.			
	Flag	There is no change.			

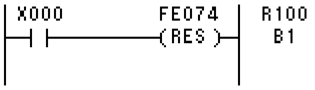


When input X000 stands up for turning on from turning off, relay R100 is turned on.

Even if X000 is turned off, it Keeps turning on R100.

F∗074 Reset coil

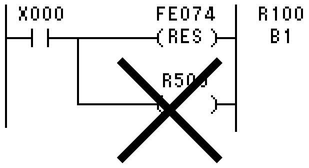
Symbol		Code	Argument		
			Ag.1	Ag.2	Ag.3
RES		F∗074	B1		
Function		The relay shown in Ag.1 is turned off.			
Range of use of Ag.1		B1 Relay:All relay			
After operation	Contents of Ag.1	It is turned off.			
	Flag	There is no change.			



When input X000 stands up for turning on from turning off, relay R100 is turned off.
Even if X000 is turned off, R100 is turning off.

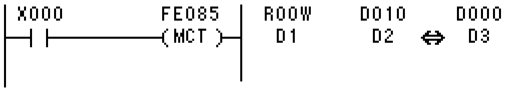
Prohibition

Please make neither F*073 nor F*074 a multi output as shown in the figure below.



F＊085 Multi counter（Up counter）

<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">MCT</td><td>F＊085</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b></td><td><b>D3</b>
<b>d3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">It is a counter in which the data of the register shown in Ag.3 is assumed to be a present value, and the table of 8 words that makes register shown in Ag.2 head is assumed to be a set value. The result is biting stored in the register that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">INC (D3 / d3) ⇔ (D2,D2+1..D2+7 / d2,d2+1..d2+7) → (D1 / d1)</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Result of count</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">Increment result</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



Whenever input X000 rises from turning off in turning on, 1 increases the value of register D000. The value is compared with the data of register D010～D017. The correspondence bit of register R00W is turned on in case of the agreement.

Present Value	Present Value			R00W	
D000 ⇔	D010	300	→	R000	ON
300	D011	400	→	R001	OFF
	D012	450	→	R002	OFF
	D013	600	→	R003	OFF
	D014	1000	→	R004	OFF
	D015	2000	→	R005	OFF
	D016	2300	→	R006	OFF
	D017	10000	→	R007	OFF

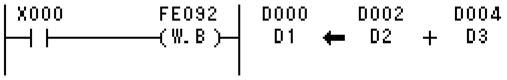
R008:Reset flag（A present value is cleared）
（R000～R007 is turned off.）
R00A:Stop count（It stops to turn on the count.）

Even if byte register is specified, registers D1 and D2 are compared as Word data.

14. Double length BIN operation instruction

F＊092 Double length BIN addition

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
W and B	F＊092	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The BIN data of 2 consecutive registers shown in Ag.3 and the BIN data of 2 consecutive registers shown in Ag.2 is added by double length. The result is stored in 2 registers that Ag.1 shows.			
Content of operation	$\begin{pmatrix}D1+1,D1\\ d1+1,d1\end{pmatrix} \leftarrow \begin{pmatrix}D2+1,D2\\ d2+1,d2\end{pmatrix} + \begin{pmatrix}D3+1,D3\\ d3+1,d3\end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.		

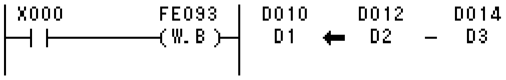


When input X000 stands up for turning on from turning off, the sum of the double length BIN data of register D003 and D002, and the double length BIN data of register D005 and D004 is requested. The result is stored in register D001 and D000.

Even if byte register is specified, it is operated as double length data (4 bytes).

F＊093 Double length BIN subtraction

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
W and B	F＊093	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The BIN data of 2 consecutive registers shown in Ag.2 is subtracted from the BIN data of 2 consecutive registers shown in Ag.3 in double length. The result is stored in 2 registers that Ag.1 shows.			
Content of operation	$\left[\begin{matrix}D1+1,D1\\ d1+1,d1\end{matrix}\right] \leftarrow \left[\begin{matrix}D2+1,D2\\ d2+1,d2\end{matrix}\right] - \left[\begin{matrix}D3+1,D3\\ d3+1,d3\end{matrix}\right]$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Carry flag Zero flag Sign flag	A000:When carrying is caused in the result, it turns on. A006:When result is 0, it turns on. A007:When MSB of the result is 1, it turns on.		

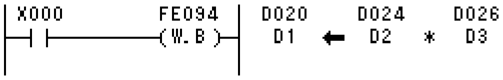


When input X000 stands up for turning on from turning off, the double length BIN data of register D015 and D014 is subtracted from the double length BIN data of register D013 and D012. The difference is stored in register D011 and D010.

Even if byte register is specified, it operates it as double length data (4 bytes).

F＊094 Double length BIN multiplication

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
W and B	F＊094	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The BIN data of 2 consecutive registers shown in Ag.2 is multiplied to the BIN data of 2 consecutive registers shown in Ag.3 in double length. The result is stored in 4 consecutive registers that Ag.1 shows.			
Content of operation	D1+3,D1+2,D1+1,D1 d1+3,d1+2,d1+1,d1 ← D2+1,D2 × D3+1,D3 d2+1,d2 d3+1,d3			
Range of use of Ag.1	D1 (Direct register) :All register d1 (Indirect register) :All register			
Range of use of Ag.2	D2 (Direct register) :All register d2 (Indirect register) :All register			
Range of use of Ag.3	D3 (Direct register) :All register d3 (Indirect register) :All register			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Flag	There is no change.		

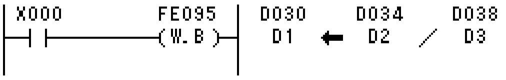


When input X000 stands up for turning on from turning off, the double length BIN data of register D025 and D024 is multiplied to the double length BIN data of register D027 and D026 data. The result is stored in register D023, D022, D021 and D020.

Even if byte register is specified, it is operated as double length data (4 bytes). Register D1 becomes the data of 4Words (8 bytes).

F＊095 Double length BIN division

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
W and B	F＊095	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The BIN data of consecutive 4 registers shown Ag.2 is divided in the consecutive BIN data of 2 consecutive registers shown in Ag.3. The quotient and the remainder are stored in 4 consecutive registers shown in Ag.1.			
Content of operation	Remainder Quotient D1+3,D1+2 D1+1,D1 d1+3,d1+2, d1+1,d1 ← D2+3,D2+2,D2+1,D2 / D3+1,D3 d1+3,d1+2,d2+1,d2 d3+1,d3			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register			
After operation: Contents of Ag.1	Operation result			
Contents of Ag.2	There is no change.			
Contents of Ag.3	There is no change.			
Flag	There is no change.			



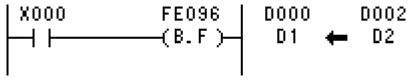
When input X000 stands up for turning on from turning off, the BIN data of 4 times length register D037, D036, D035 and D034 is divided in the BIN data of double length register D039 and D038. The quotient is stored in register D031 and D030 and the remainder is stored in D033 and D034.

Even if byte register is specified, it operates it as double length data (4bytes).
Register D2 becomes the data of 4Words (8 bytes).

15. Floating point number operation instruction

F＊096 Floating point number conversion

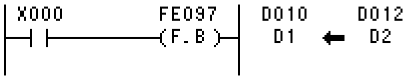
<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">B and F</td><td>F＊096</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b></td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The double length BIN data of 2 consecutive registers shown in Ag.2 is converted into the floating-point number data. The result is stored in 2 consecutive registers that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\mathrm{FLT}\left(\begin{matrix}D1+1,D1\\ d1+1,d1\end{matrix}\right)\leftarrow\left(\begin{matrix}D2+1,D2\\ d2+1,d2\end{matrix}\right)$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



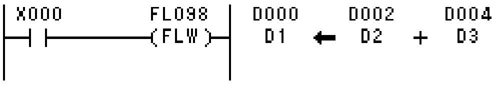
When input X000 stands up for turning on from turning off, the double length BIN data of register D003 and D002 is converted into floating point number data. The result is stored in register D001 and D000.

Even if byte register is specified, it operates it as floating point number data (4bytes).

F＊097 Floating point number inversion

<table>
<tr><td colspan="2" rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td><td rowspan="11">
When input X000 stands up for turning on from turning off, the floating points number data of register D013 and D012 is converted in double length BIN data.
The result is stored in register D011 and D010.
Even if byte register is specified, it is operated as floating point number data (4bytes).</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">F and B</td><td>F＊097</td><td>**D1**
d1</td><td>**D2**
d2</td><td></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The floating-point number data of 2 consecutive registers shown in Ag.2 is converted into double length BIN data. The result is stored in two consecutive registers that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\mathrm{BIN}\begin{pmatrix}D1+1,D1\\d1+1,d1\end{pmatrix} \leftarrow \begin{pmatrix}D2+1,D2\\d2+1,d2\end{pmatrix}_{\mathrm{FLT}}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>

F＊098 Floating point number addition

<table>
<tr><td colspan="2" rowspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td><td rowspan="12">

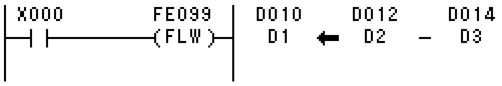
X000
FL098
(FLW)
D000
D1
D002
D2
+
D004
D3

When input X000 is turned on, the sum of the floating point number data of register D003 and D002 and the floating point number data of register D005 and D004 is requested. The result is stored in register D001 and D000.

Even if byte register is specified, it is operated as floating point number data (4 bytes).</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">FLW</td><td>F＊098</td><td>**D1**
d1</td><td>**D2**
d2</td><td>**D3**
d3</td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The floating-point number data of 2 consecutive registers shown in Ag.2 and the floating point number data of 2 consecutive registers shown in Ag.3 are added. The result is stored in 2 registers that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\mathrm{FLT}\begin{pmatrix}D1+1,D1\\ d1+1,d1\end{pmatrix} \leftarrow \begin{pmatrix}D2+1,D2\\ d2+1,d2\end{pmatrix} + \begin{pmatrix}D3+1,D3\\ d3+1,d3\end{pmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>

F＊099 Floating point number subtraction

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
FLW	F＊099	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The floating-point number data of 2 consecutive registers shown in Ag.3 is subtracted from floating point number data of 2 consecutive registers shown in Ag.2. The result is stored in 2 registers that Ag.1 shows.			
Content of operation	$\begin{pmatrix}D1+1,D1\\d1+1,d1\end{pmatrix} \leftarrow \begin{pmatrix}D2+1,D2\\d2+1,d2\end{pmatrix} - \begin{pmatrix}D3+1,D3\\d3+1,d3\end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Flag	There is no change.		

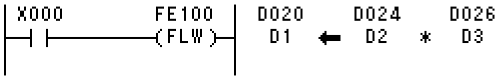


When input X000 stands up for turning on from turning off, the floating point number data of register D015 and D014 is subtracted from the floating point number data of register D013 and D012. The difference is stored in register D011 and D010.

Even if byte register is specified, it is operated as floating point number data (4 bytes).

F∗100 Floating point number multiplication

<table>
<tr><td rowspan="2" colspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">FLW</td><td>F∗100</td><td><b>D1
d1</b></td><td><b>D2
d2</b></td><td><b>D3
d3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The floating-point number data of 2 consecutive registers shown in Ag.2 is multiplied to the floating-point number data of 2 consecutive registers shown in Ag.3. The result is stored in 2 consecutive registers that Ag.1 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\begin{pmatrix} D1+1,D1 \\ d1+1,d1 \end{pmatrix} \leftarrow \begin{pmatrix} D2+1,D2 \\ d2+1,d2 \end{pmatrix} \times \begin{pmatrix} D3+1,D3 \\ d3+1,d3 \end{pmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



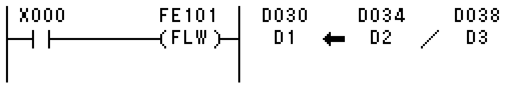
When input X000 stands up for turning on from turning off, the floating point number data of register D025 and D024 is multiplied to the floating point number data of register D027 and D026. The result is stored in register D021 and D020.

Even if byte register is specified, it is operated as floating point number data (4 bytes).

F＊101 Floating point number division

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
FLW	F＊101	**D1** **d1**	**D2** **d2**	**D3** **d3**

Function		The floating-point number data of 2 consecutive registers shown in Ag.2 is divided to the floating-point number data of 2 consecutive registers shown in Ag.3. The quotient is stored in 2 consecutive registers that Ag.1 shows.
Content of operation		$\begin{bmatrix} D1+1,D1 \\ d1+1,d1 \end{bmatrix} \leftarrow \begin{bmatrix} D2+1,D2 \\ d2+1,d2 \end{bmatrix} / \begin{bmatrix} D3+1,D3 \\ d3+1,d3 \end{bmatrix}$
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register
After operation	Contents of Ag.1	Operation result
	Contents of Ag.2	There is no change.
	Contents of Ag.3	There is no change.
	Flag	There is no change.



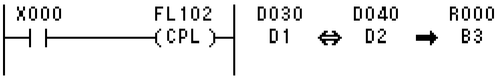
When input X000 stands up for turning on from turning off, the floating point number data of register D035 and D034 is divided in the floating point number data of register D039 and D038. The quotient is stored in register D031 and D030.

Even if byte register is specified, it is operated as floating point number data (4 bytes).

16. BIN operation instruction

F＊102 Double length data comparison（Unsigned）

<table>
<tr><td rowspan="2" colspan="2">Symbol</td><td rowspan="2">Code</td><td colspan="3">Argument</td></tr>
<tr><td>Ag.1</td><td>Ag.2</td><td>Ag.3</td></tr>
<tr><td colspan="2">CPL</td><td>F＊102</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b></td><td>B3</td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The double length BIN data of register shown in Ag.1 is compared with the double length BIN data of register shown in Ag.2. The result is stored in the relay where Ag.3 shows.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\begin{pmatrix}D1+1,D1\\ d1+1,d1\end{pmatrix} \Leftrightarrow \begin{pmatrix}D2+1,D2\\ d2+1,d2\end{pmatrix} \rightarrow B3$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">B3 Relay
:All relay（Even number address）</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4"><table><tr><td>Relay</td><td>ON</td><td>OFF</td></tr><tr><td>B3</td><td>Ag.1≦Ag.2</td><td>Ag.1＞Ag.2</td></tr><tr><td>B3+1</td><td>Ag.1＝Ag.2</td><td>Ag.1≠Ag.2</td></tr></table></td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 is turned on, the double length BIN data of register D031 and D030 is compared with the double length BIN data of register D041 and D040. The result is stored in R000 and R001.

	MSB															LSB
D031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D030:(3140)	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0
D041	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D040:(6241)	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1

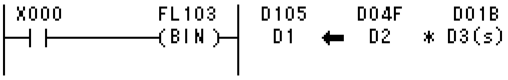
↓

R000:ON、R001:OFF

When byte register is used, it is compared as a value of double length registers whose high-order 3 bytes are 0.

F＊103 BIN multiplication（Signed）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊103	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The product of the BIN data of the register shown Ag.2 and the BIN data of the register shown in Ag.3 is requested. The result is stored in register (2 words) that Ag.1 shows.			
Content of operation	$\mathrm{BIN}\left\{\begin{matrix}D1,D1+1\\ d1,d1+1\end{matrix}\right\} \leftarrow \left\{\begin{matrix}D2\\ d2\\ K2\end{matrix}\right\} \times \left\{\begin{matrix}D3\\ d3\\ K3\end{matrix}\right\}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Flag	There is no change.		



When input X000 is turned on, the product of the BIN data of register D01B and the BIN data of register D04F is requested. The result is stored in 2 words of register D106 and D105.

D04F:(5130) 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0

×

D01B:(44) 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0

↓

MSB LSB

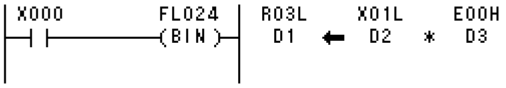
D106 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

D105:(225720) 0 1 1 1 0 0 0 1 1 0 1 1 1 0 0 0

（D106:High-order Words、D105:Low-order Words）

Even if byte register is specified for register D1, it is stored as 4 bytes long data.

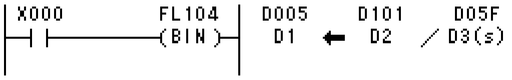
Example.1 When register D1, D2, D3 are byte register



X01L（100）× E00H（50）→ R04H and R04L, R03H and R03L（5000）

F＊104 BIN subtraction（Signed）

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
BIN	F＊104	**D1** **d1**	**D2** **d2** **K2**	**D3** **d3** **K3**
Function	The 2 words length BIN data of the register shown in Ag.2 is divided with BIN data of the register shown in Ag.3. The quotient and remainder is stored in the register that Ag.1 shows.			
Content of operation	BIN (D1 / d1) ← (D2,D2+1 / d2,d2+1 / K2) ／ (D3 / d3 / K3) D1+1（Reminder） d1+1			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register K2 Constant:-32768～32767			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register K3 Constant:-32768～32767			
After operation	Contents of Ag.1	Operation result		
	Contents of Ag.2	There is no change.		
	Contents of Ag.3	There is no change.		
	Flag	There is no change.		



When input X000 is turned on, the 2 words data of register D101 and D100 is divided by the BIN data of register D05F. The quotient is stored in register D005 and the reminder is stored in D006.

D102（High-order） 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1
D101:(6362145)（Low-order） 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1

／

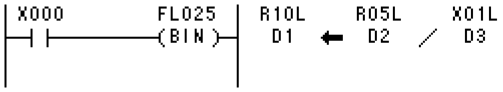
D05F:(22018) 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1 0

↓

MSB LSB
D005:(288)（Quotient） 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
D006:(20961)（Reminder） 0 1 0 1 0 0 0 1 1 1 1 0 0 0 0 1

Even if byte register is specified, register D2 is divided as 2 words data.

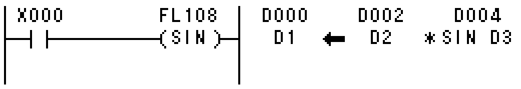
Example.1 When register D1, D2, D3 are byte registers



R06H and R06L, R05H and R05L ／ X01L → R10H and R10L（Quotient）
R11H and R11L（Reminder）

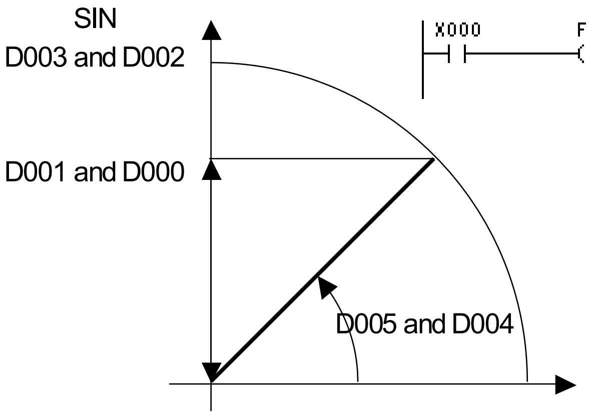
F∗108 SIN multiplication(Unsigned)

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
SIN	F∗108	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The BIN data of consecutive 2 registers shown in Ag.3 is assumed to be angle data in every 0.001 degrees, and the SIN value is requested. It is multiplied to the BIN data of 2 consecutive registers shown in Ag.2 by double length. The result is stored in 2 consecutive registers that Ag.1 shows.			
Content of operation	(D1+1,D1 / d1+1,d1) ← (D2+1,D2 / d2+1,d2) × SIN (D3+1,D3 / d3+1,d3)			
Range of use of Ag.1	D1 (Direct register) :All register d1 (Indirect register) :All register			
Range of use of Ag.2	D2 (Direct register) :All register d2 (Indirect register) :All register			
Range of use of Ag.3	D3 (Direct register) :All register d3 (Indirect register) :All register The range of the numerical value is 0～89999.			
After: Contents of Ag.1	Operation result			
After: Contents of Ag.2	There is no change.			
After: Contents of Ag.3	There is no change.			
After: Flag	There is no change.			



When input X000 is turned on, the double length BIN data of register D005 and D004 is assumed to be angle data in every 0.001 degrees and Sin is operated.
The double length BIN data of register D003 and D002 is multiplied to it.
The result is stored in register D001 and D000.

The content of the Sin operation: The Sin operation of 0～90 degree is executed.

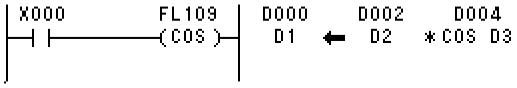


```
X000          FL108    D000    D002    D004
-| |-----------(SIN)-   D1  ←   D2    ∗SIN D3
```

Address	Value	Content
D001 and D000	5000	Sin(30)=0.5
D003 and D002	10000	
D005 and D004	30000	30 degrees

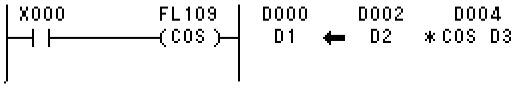
F∗109 COS multiplication(Unsigned)

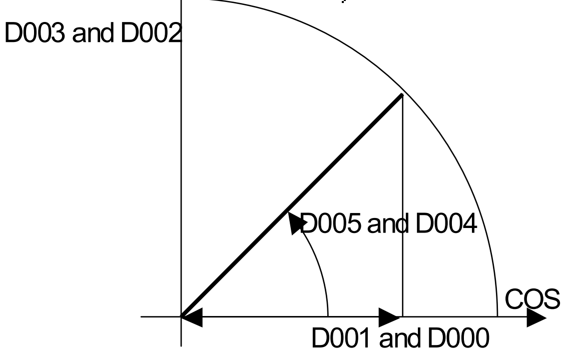
Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
COS	F∗109	**D1** **d1**	**D2** **d2**	**D3** **d3**
Function	The BIN data of consecutive 2 registers shown in Ag.3 is assumed to be angle data in every 0.001 degrees and the COS value is requested. The result is multiplied to the double length BIN data of consecutive 2 registers shown in Ag.2 by double length. The result is stored in 2 consecutive registers that Ag.1 shows.			
Content of operation	$\begin{pmatrix}D1+1,D1\\d1+1,d1\end{pmatrix} \leftarrow \begin{pmatrix}D2+1,D2\\d2+1,d2\end{pmatrix} \times COS \begin{pmatrix}D3+1,D3\\d3+1,d3\end{pmatrix}$			
Range of use of Ag.1	D1（Direct register）:All register d1（Indirect register）:All register			
Range of use of Ag.2	D2（Direct register）:All register d2（Indirect register）:All register			
Range of use of Ag.3	D3（Direct register）:All register d3（Indirect register）:All register The range of the numerical value is 0～89999.			
After operation	Contents of Ag.1	Operation result		
After operation	Contents of Ag.2	There is no change.		
After operation	Contents of Ag.3	There is no change.		
After operation	Flag	There is no change.		



When input X000 is turned on, the double length BIN data of register D005 and D004 is assumed to be angle data in every 0.001 degrees and COS is operated. The double length BIN data of register D003 and D002 is multiplied to it.
The result is stored in register D001 and D000.

The content of the COS operation: The COS operation of 0～90 degree is executed.

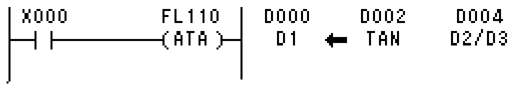




Address	Value	Content
D001 and D000	5000	COS(60)=0.5
D003 and D002	10000	
D005 and D004	60000	60degrees

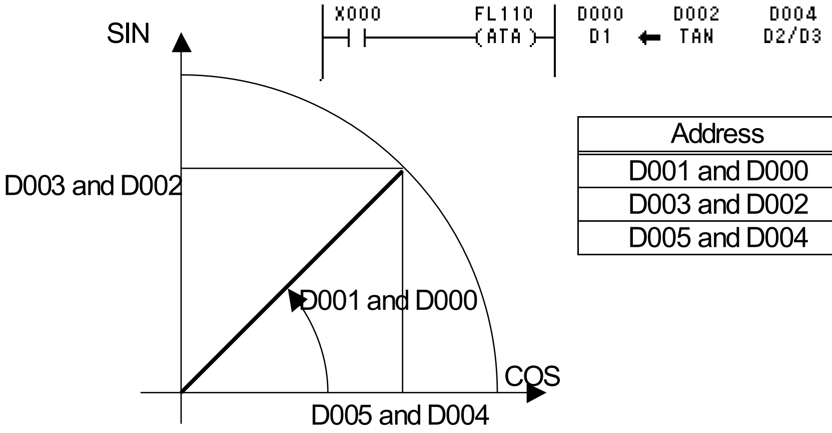
F∗110 ATAN operation

<table>
<tr><th colspan="2" rowspan="2">Symbol</th><th rowspan="2">Code</th><th colspan="3">Argument</th></tr>
<tr><th>Ag.1</th><th>Ag.2</th><th>Ag.3</th></tr>
<tr><td colspan="2">ATA</td><td>F∗110</td><td><b>D1</b>
<b>d1</b></td><td><b>D2</b>
<b>d2</b></td><td><b>D3</b>
<b>d3</b></td></tr>
<tr><td colspan="2">Function</td><td colspan="4">The BIN data of consecutive 2 registers shown in Ag.2 is assumed to be SIN value, and the BIN data of consecutive 2 registers shown in Ag.3 is assumed to be COS value and ATAN is operated. The result is stored 2 consecutive register shown in Ag.1 as an angle data in every 0.001 degrees.</td></tr>
<tr><td colspan="2">Content of operation</td><td colspan="4">$\begin{bmatrix} D1+1,D1 \\ d1+1,d1 \end{bmatrix} \leftarrow$ TAN-1 $\begin{bmatrix} D2+1,D2 \\ d2+1,d2 \end{bmatrix} ／ \begin{bmatrix} D3+1,D3 \\ d3+1,d3 \end{bmatrix}$</td></tr>
<tr><td colspan="2">Range of use of Ag.1</td><td colspan="4">D1（Direct register）:All register
d1（Indirect register）:All register
The range of the numerical value is 0～89999.</td></tr>
<tr><td colspan="2">Range of use of Ag.2</td><td colspan="4">D2（Direct register）:All register
d2（Indirect register）:All register</td></tr>
<tr><td colspan="2">Range of use of Ag.3</td><td colspan="4">D3（Direct register）:All register
d3（Indirect register）:All register</td></tr>
<tr><td rowspan="4">After operation</td><td>Contents of Ag.1</td><td colspan="4">Operation result</td></tr>
<tr><td>Contents of Ag.2</td><td colspan="4">There is no change.</td></tr>
<tr><td>Contents of Ag.3</td><td colspan="4">There is no change.</td></tr>
<tr><td>Flag</td><td colspan="4">There is no change.</td></tr>
</table>



When input X000 is turned on, the double length BIN data of register D005 and D004 is assumed to be COS data, the double length BIN data of register D003 and D002 is assumed to be SIN data, and ATAN is operated.
The result is stored in register D001 and D000.

The content of the ATAN operation: The ATAN operation of 0～90 degree is executed.

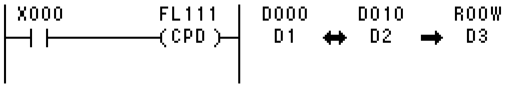


Address	Value	Content
D001 and D000	45000	45 degrees
D003 and D002	10000	
D005 and D004	10000	

F∗111 Refer to the table

Symbol	Code	Argument		
		Ag.1	Ag.2	Ag.3
CPD	F∗111	**D1** **d1** **K1**	**D2** **d2**	**D3** **d3**

Function		The data of the register shown in Ag.1 is compared with the content of the table whose register shown in Ag.2 is head. The comparison result is biting stored in the register shown in Ag.3. The size of the table can be specified up to 256W or less according to the value of Ag.3.
Content of operation		(D1, d1, K1) ⇔ (D2,D2+1..D2+n, d2,d2+1..d2+n) → (D3+1, d3+1) n =D3 d3
Range of use of Ag.1		D1（Direct register）:All register d1（Indirect register）:All register K1 定数:-32768～32767
Range of use of Ag.2		D2（Direct register）:All register d2（Indirect register）:All register
Range of use of Ag.3		D3（Direct register）:All register d3（Indirect register）:All register
After operation	Contents of Ag.1	There is no change.
	Contents of Ag.2	There is no change.
	Contents of Ag.3	Comparison result
	Flag	There is no change.



When input X000 is turned on, the data of register D000 is compared with the data of register D010～D017. The correspondence bit of register R01W is turned on in case of the agreement. If it is a disagreement, it is invariable.
(The size of the table is 8 words, and is specified from R00W.)

			Value		R01W	
D000 300	⇔	D010	300	→	R010	ON
		D011	200	→	R011	invariable
		D012	250	→	R012	invariable
		D013	300	→	R013	ON
		D014	1000	→	R014	invariable
		D015	2000	→	R015	invariable
		D016	2300	→	R016	invariable
		D107	10000	→	R017	invariable

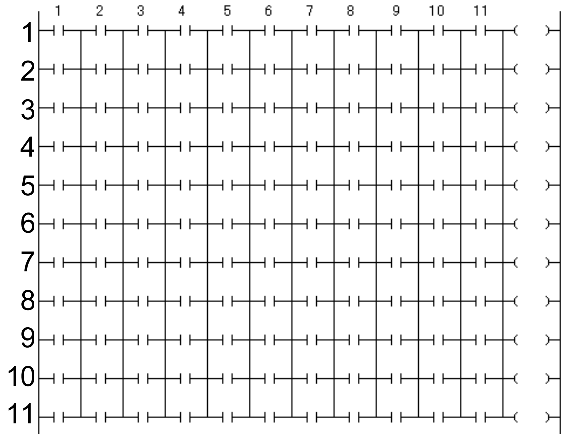
R018:Disagreement bit:0N
R00W:Size of table:8

17. Programming

In Tc-mini, each ladder circuit is read from the memory once, and is executed. Therefore, operation might be different from a general relay circuit. Moreover, there is a limitation such as being possible to connect it by return because there is a diode characteristic that flows from the left of the circuit only to the right.
Here, we will introduce the restriction item of those circuit designs. Please correctly understand this content, and use it for an efficient circuit design.

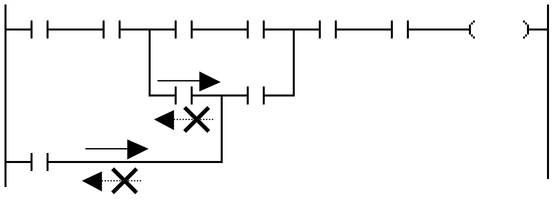
17.1 Restriction of size of circuit

In TC-mini, the maximum size of the circuit is restricted (11 rows + 1Coil) ×11 lines.



17.2 Diode characteristic of circuit

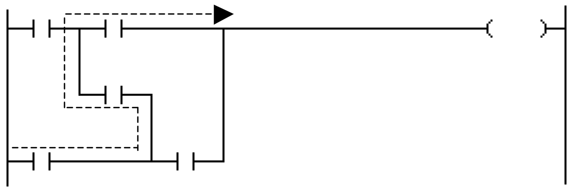
The circuit has the characteristic where the current flows only in the direction where it faces right coil from the left.



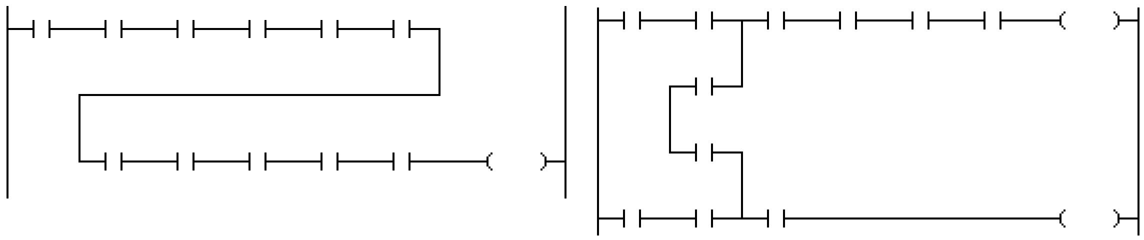
→:The current flows in this direction.

◄✕:The current doesn't flow in this direction.

The logic of this------▶ route is not approved in the following circuit.



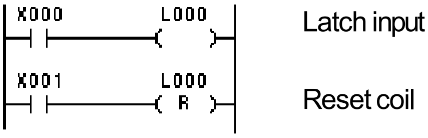
The following circuit doesn't operate from the above -mentioned content.



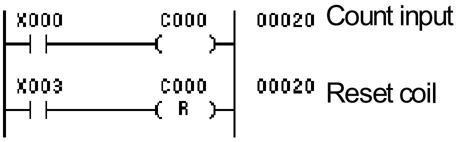
17.3 Pair coil instruction

There is an instruction treated as one circuit including the reset coil like the latch circuit and the counter circuit, etc.

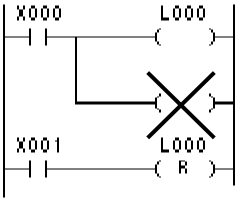
Latch instruction



Counter instruction



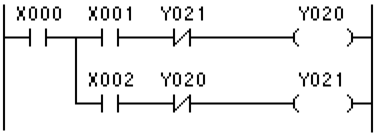
A usual coil instruction cannot be input in pair coil.



17.4 The order of processing program

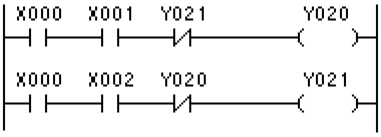
TC-mini operates each circuit from the head of the program to the end, and repeats this process. Moreover, TC-mini will sequentially operate the coil row from the first row from the left to the right once on the road. Please note that operation different from the interlock of the electric circuit might be done.

[Bad example]



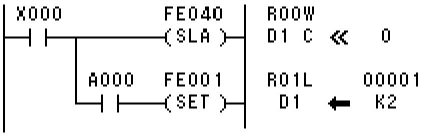
In the above mentioned circuit, when X001 and X002 are turned on at the same time, Y020 and Y021 repeat ON/OFF.

[Good example]



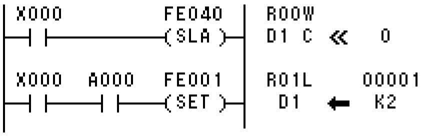
In the above mentioned circuit, Y020 gives priority when X001 and X002 are turned on at the same time.

[Bad example]



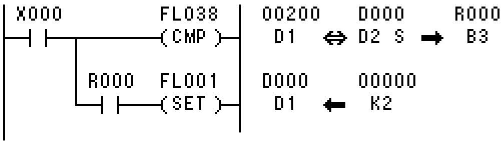
The above -mentioned circuit does R00W to set 1 in R01L when R00F shifts when the left shifts while turned on. However, it doesn't operate correctly.

[Good example]



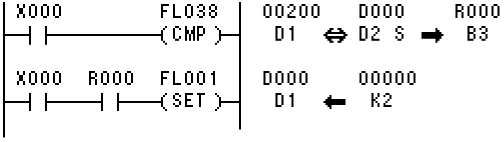
A000 (carry) in FE040 is reflected on the next Circuit.

[Bad example]



Comparison result R000 is reflected delaying 1 scanning.

[Good example]



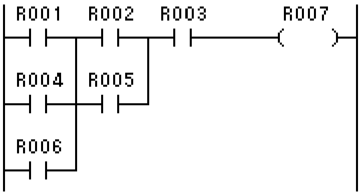
The result of R000 is reflected in the next road, and is treatable at once for the comparison result.

17.5 Efficiency of program

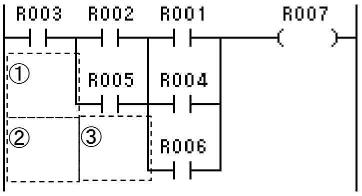
In TCmini, the circuit element such as a length connection, horizontal connections, and blanks is counted with all 1 word.
Therefore, the program capacity can be saved by collecting the point of contact elements left.

A left circuit becomes small program capacity though operation is the same as the following circuit.

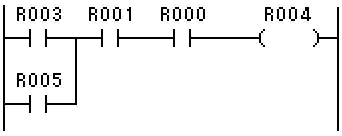
Circuit by 7 words



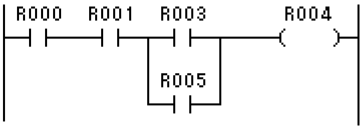
Circuit by 10 words:①, ②, ③ are counted



Circuit by 5 words

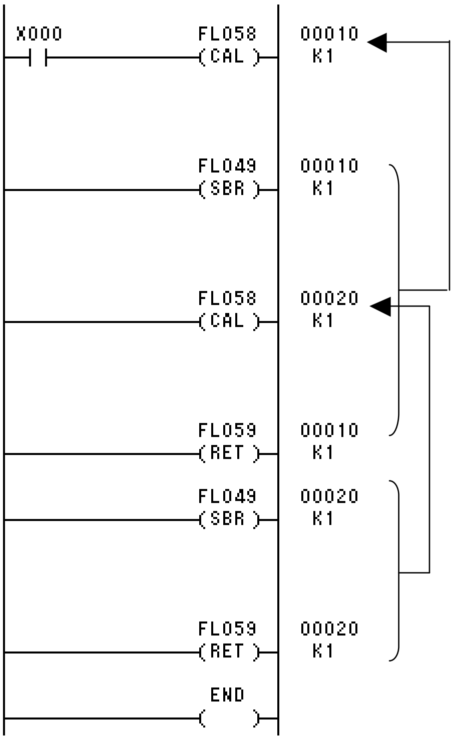


Circuit by 7 words



17.6 Subroutine program

In TC-mini, the program of 32 subroutines or less can be made. There is no limitation in the frequency of nesting (Another subroutine is called from among the subroutine). However, please do not call the same subroutine. It causes the malfunction.
The subroutine program can arrange at a circuit position arbitrary from the head of the main program to END, and is executed by call instruction (F*58).



When X000 is turned on, subroutine 10 is executed.

Subroutine 10

Nesting (1 step)

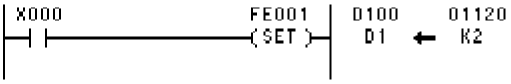
Subroutine 20

17.7 Edge instruction (standing up execution instruction)

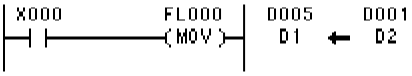
In the application instruction, when the instruction is described FL**, this instruction is always executed among turning on the input. However, when the input condition stands up for turning on from turning off, it executes it only once when describing FE**. This is called an application instruction the edge execution type. TC02V can only execute the edge execution types up to 4096.

The over function error occurs when TC02V begins executing if it programs it exceeding 4096 pieces, and it stops.

Example for Application instruction edge execution type



Example for Application instruction always execution type



17.8 Expression of numerical value

TC-mini can handle the following Kind of numerical values.

① Binary Code number: BIN
② Hexadecimal number: HEX
③ Binary Code Decimal number:BCD
④ Floating point number

① Binary Code number:BIN

In the binary number, the treated range of the numerical value changes in the signed or unsigned, length of Byte, length of Word, and length of double Word.

	Byte length	Word length	Double Word length（Double length）
Unsigned	0~255	0～65535	—
Signed	—	-32768～32767	-2147483648～2147483647

TC-mini has the register of the length of Byte and the length of Word.

Example of Byte length register:R00L, R00H, X01L, L02H, A03H, etc

Example of Word length register:D000, R00W. Y02W, etc

The data of the length of double word is treated by the register of consecutive length of Word.

For a register consecutive like D001·D000, D001 becomes high-order data, and D000 becomes low-order data.

② Hexadecimal number:HEX

In the hexadecimal number, the alphabet is used for the expression such as 0～9, A, B, C, D, E, and F. When displaying it, "H" is put on the end to distinguish from other numerical values. （Example:1AH）

	Byte length	Word length	Double word length（Double length）
Range of value	0~FFH	0～FFFFH	0～FFFFFFFFH

③ Binary Code Decimal number:BCD

BCD is a binary number with ten carried expressions as well as the decimal number. It uses it for a numeric input from Sam rotary SW and a numeric output to seven-segment indicator.
It is monitored by the hexadecimal number display.

	Byte length	Word length	Double word length（Double length）
Range of value	0~99H	0～9999H	-32768～99999999H (Note

Note）

When you convert a negative BIN value with F*012 (BIN→BCD conversion with the sign), “DH” is stored in the highest-order digit, and a minus expression is done.
Even F*013 (BCD→BIN conversion with the sign) is similar.

18.Appendex

18.1 Indirect addressing table of data register

The indirect addressing of the data register is shown.

Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register
D000	0	D100	256	D200	512	D300	768	D400	1024	D500	1280	D600	1536	D700	1792
D001	2	D101	258	D201	514	D301	770	D401	1026	D501	1282	D601	1538	D701	1794
D002	4	D102	260	D202	516	D302	772	D402	1028	D502	1284	D602	1540	D702	1796
D003	6	D103	262	D203	518	D303	774	D403	1030	D503	1286	D603	1542	D703	1798
D004	8	D104	264	D204	520	D304	776	D404	1032	D504	1288	D604	1544	D704	1800
D005	10	D105	266	D205	522	D305	778	D405	1034	D505	1290	D605	1546	D705	1802
D006	12	D106	268	D206	524	D306	780	D406	1036	D506	1292	D606	1548	D706	1804
D007	14	D107	270	D207	526	D307	782	D407	1038	D507	1294	D607	1550	D707	1806
D008	16	D108	272	D208	528	D308	784	D408	1040	D508	1296	D608	1552	D708	1808
D009	18	D109	274	D209	530	D309	786	D409	1042	D509	1298	D609	1554	D709	1810
D00A	20	D10A	276	D20A	532	D30A	788	D40A	1044	D50A	1300	D60A	1556	D70A	1812
D00B	22	D10B	278	D20B	534	D30B	790	D40B	1046	D50B	1302	D60B	1558	D70B	1814
D00C	24	D10C	280	D20C	536	D30C	792	D40C	1048	D50C	1304	D60C	1560	D70C	1816
D00D	26	D10D	282	D20D	538	D30D	794	D40D	1050	D50D	1306	D60D	1562	D70D	1818
D00E	28	D10E	284	D20E	540	D30E	796	D40E	1052	D50E	1308	D60E	1564	D70E	1820
D00F	30	D10F	286	D20F	542	D30F	798	D40F	1054	D50F	1310	D60F	1566	D70F	1822
D010	32	D110	288	D210	544	D310	800	D410	1056	D510	1312	D610	1568	D710	1824
D011	34	D111	290	D211	546	D311	802	D411	1058	D511	1314	D611	1570	D711	1826
D012	36	D112	292	D212	548	D312	804	D412	1060	D512	1316	D612	1572	D712	1828
D013	38	D113	294	D213	550	D313	806	D413	1062	D513	1318	D613	1574	D713	1830
D014	40	D114	296	D214	552	D314	808	D414	1064	D514	1320	D614	1576	D714	1832
D015	42	D115	298	D215	554	D315	810	D415	1066	D515	1322	D615	1578	D715	1834
D016	44	D116	300	D216	556	D316	812	D416	1068	D516	1324	D616	1580	D716	1836
D017	46	D117	302	D217	558	D317	814	D417	1070	D517	1326	D617	1582	D717	1838
D018	48	D118	304	D218	560	D318	816	D418	1072	D518	1328	D618	1584	D718	1840
D019	50	D119	306	D219	562	D319	818	D419	1074	D519	1330	D619	1586	D719	1842
D01A	52	D11A	308	D21A	564	D31A	820	D41A	1076	D51A	1332	D61A	1588	D71A	1844
D01B	54	D11B	310	D21B	566	D31B	822	D41B	1078	D51B	1334	D61B	1590	D71B	1846
D01C	56	D11C	312	D21C	568	D31C	824	D41C	1080	D51C	1336	D61C	1592	D71C	1848
D01D	58	D11D	314	D21D	570	D31D	826	D41D	1082	D51D	1338	D61D	1594	D71D	1850
D01E	60	D11E	316	D21E	572	D31E	828	D41E	1084	D51E	1340	D61E	1596	D71E	1852
D01F	62	D11F	318	D21F	574	D31F	830	D41F	1086	D51F	1342	D61F	1598	D71F	1854
D020	64	D120	320	D220	576	D320	832	D420	1088	D520	1344	D620	1600	D720	1856
D021	66	D121	322	D221	578	D321	834	D421	1090	D521	1346	D621	1602	D721	1858
D022	68	D122	324	D222	580	D322	836	D422	1092	D522	1348	D622	1604	D722	1860
D023	70	D123	326	D223	582	D323	838	D423	1094	D523	1350	D623	1606	D723	1862
D024	72	D124	328	D224	584	D324	840	D424	1096	D524	1352	D624	1608	D724	1864
D025	74	D125	330	D225	586	D325	842	D425	1098	D525	1354	D625	1610	D725	1866
D026	76	D126	332	D226	588	D326	844	D426	1100	D526	1356	D626	1612	D726	1868
D027	78	D127	334	D227	590	D327	846	D427	1102	D527	1358	D627	1614	D727	1870
D028	80	D128	336	D228	592	D328	848	D428	1104	D528	1360	D628	1616	D728	1872
D029	82	D129	338	D229	594	D329	850	D429	1106	D529	1362	D629	1618	D729	1874
D02A	84	D12A	340	D22A	596	D32A	852	D42A	1108	D52A	1364	D62A	1620	D72A	1876
D02B	86	D12B	342	D22B	598	D32B	854	D42B	1110	D52B	1366	D62B	1622	D72B	1878
D02C	88	D12C	344	D22C	600	D32C	856	D42C	1112	D52C	1368	D62C	1624	D72C	1880
D02D	90	D12D	346	D22D	602	D32D	858	D42D	1114	D52D	1370	D62D	1626	D72D	1882
D02E	92	D12E	348	D22E	604	D32E	860	D42E	1116	D52E	1372	D62E	1628	D72E	1884
D02F	94	D12F	350	D22F	606	D32F	862	D42F	1118	D52F	1374	D62F	1630	D72F	1886

Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register
D030	96	D130	352	D230	608	D330	864	D430	1120	D530	1376	D630	1632	D730	1888
D031	98	D131	354	D231	610	D331	866	D431	1122	D531	1378	D631	1634	D731	1890
D032	100	D132	356	D232	612	D332	868	D432	1124	D532	1380	D632	1636	D732	1892
D033	102	D133	358	D233	614	D333	870	D433	1126	D533	1382	D633	1638	D733	1894
D034	104	D134	360	D234	616	D334	872	D434	1128	D534	1384	D634	1640	D734	1896
D035	106	D135	362	D235	618	D335	874	D435	1130	D535	1386	D635	1642	D735	1898
D036	108	D136	364	D236	620	D336	876	D436	1132	D536	1388	D636	1644	D736	1900
D037	110	D137	366	D237	622	D337	878	D437	1134	D537	1390	D637	1646	D737	1902
D038	112	D138	368	D238	624	D338	880	D438	1136	D538	1392	D638	1648	D738	1904
D039	114	D139	370	D239	626	D339	882	D439	1138	D539	1394	D639	1650	D739	1906
D03A	116	D13A	372	D23A	628	D33A	884	D43A	1140	D53A	1396	D63A	1652	D73A	1908
D03B	118	D13B	374	D23B	630	D33B	886	D43B	1142	D53B	1398	D63B	1654	D73B	1910
D03C	120	D13C	376	D23C	632	D33C	888	D43C	1144	D53C	1400	D63C	1656	D73C	1912
D03D	122	D13D	378	D23D	634	D33D	890	D43D	1146	D53D	1402	D63D	1658	D73D	1914
D03E	124	D13E	380	D23E	636	D33E	892	D43E	1148	D53E	1404	D63E	1660	D73E	1916
D03F	126	D13F	382	D23F	638	D33F	894	D43F	1150	D53F	1406	D63F	1662	D73F	1918
D040	128	D140	384	D240	640	D340	896	D440	1152	D540	1408	D640	1664	D740	1920
D041	130	D141	386	D241	642	D341	898	D441	1154	D541	1410	D641	1666	D741	1922
D042	132	D142	388	D242	644	D342	900	D442	1156	D542	1412	D642	1668	D742	1924
D043	134	D143	390	D243	646	D343	902	D443	1158	D543	1414	D643	1670	D743	1926
D044	136	D144	392	D244	648	D344	904	D444	1160	D544	1416	D644	1672	D744	1928
D045	138	D145	394	D245	650	D345	906	D445	1162	D545	1418	D645	1674	D745	1930
D046	140	D146	396	D246	652	D346	908	D446	1164	D546	1420	D646	1676	D746	1932
D047	142	D147	398	D247	654	D347	910	D447	1166	D547	1422	D647	1678	D747	1934
D048	144	D148	400	D248	656	D348	912	D448	1168	D548	1424	D648	1680	D748	1936
D049	146	D149	402	D249	658	D349	914	D449	1170	D549	1426	D649	1682	D749	1938
D04A	148	D14A	404	D24A	660	D34A	916	D44A	1172	D54A	1428	D64A	1684	D74A	1940
D04B	150	D14B	406	D24B	662	D34B	918	D44B	1174	D54B	1430	D64B	1686	D74B	1942
D04C	152	D14C	408	D24C	664	D34C	920	D44C	1176	D54C	1432	D64C	1688	D74C	1944
D04D	154	D14D	410	D24D	666	D34D	922	D44D	1178	D54D	1434	D64D	1690	D74D	1946
D04E	156	D14E	412	D24E	668	D34E	924	D44E	1180	D54E	1436	D64E	1692	D74E	1948
D04F	158	D14F	414	D24F	670	D34F	926	D44F	1182	D54F	1438	D64F	1694	D74F	1950
D050	160	D150	416	D250	672	D350	928	D450	1184	D550	1440	D650	1696	D750	1952
D051	162	D151	418	D251	674	D351	930	D451	1186	D551	1442	D651	1698	D751	1954
D052	164	D152	420	D252	676	D352	932	D452	1188	D552	1444	D652	1700	D752	1956
D053	166	D153	422	D253	678	D353	934	D453	1190	D553	1446	D653	1702	D753	1958
D054	168	D154	424	D254	680	D354	936	D454	1192	D554	1448	D654	1704	D754	1960
D055	170	D155	426	D255	682	D355	938	D455	1194	D555	1450	D655	1706	D755	1962
D056	172	D156	428	D256	684	D356	940	D456	1196	D556	1452	D656	1708	D756	1964
D057	174	D157	430	D257	686	D357	942	D457	1198	D557	1454	D657	1710	D757	1966
D058	176	D158	432	D258	688	D358	944	D458	1200	D558	1456	D658	1712	D758	1968
D059	178	D159	434	D259	690	D359	946	D459	1202	D559	1458	D659	1714	D759	1970
D05A	180	D15A	436	D25A	692	D35A	948	D45A	1204	D55A	1460	D65A	1716	D75A	1972
D05B	182	D15B	438	D25B	694	D35B	950	D45B	1206	D55B	1462	D65B	1718	D75B	1974
D05C	184	D15C	440	D25C	696	D35C	952	D45C	1208	D55C	1464	D65C	1720	D75C	1976
D05D	186	D15D	442	D25D	698	D35D	954	D45D	1210	D55D	1466	D65D	1722	D75D	1978
D05E	188	D15E	444	D25E	700	D35E	956	D45E	1212	D55E	1468	D65E	1724	D75E	1980
D05F	190	D15F	446	D25F	702	D35F	958	D45F	1214	D55F	1470	D65F	1726	D75F	1982

Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register	Register	Indirect register
D060	192	D160	448	D260	704	D360	960	D460	1216	D560	1472	D660	1728	D760	1984
D061	194	D161	450	D261	706	D361	962	D461	1218	D561	1474	D661	1730	D761	1986
D062	196	D162	452	D262	708	D362	964	D462	1220	D562	1476	D662	1732	D762	1988
D063	198	D163	454	D263	710	D363	966	D463	1222	D563	1478	D663	1734	D763	1990
D064	200	D164	456	D264	712	D364	968	D464	1224	D564	1480	D664	1736	D764	1992
D065	202	D165	458	D265	714	D365	970	D465	1226	D565	1482	D665	1738	D765	1994
D066	204	D166	460	D266	716	D366	972	D466	1228	D566	1484	D666	1740	D766	1996
D067	206	D167	462	D267	718	D367	974	D467	1230	D567	1486	D667	1742	D767	1998
D068	208	D168	464	D268	720	D368	976	D468	1232	D568	1488	D668	1744	D768	2000
D069	210	D169	466	D269	722	D369	978	D469	1234	D569	1490	D669	1746	D769	2002
D06A	212	D16A	468	D26A	724	D36A	980	D46A	1236	D56A	1492	D66A	1748	D76A	2004
D06B	214	D16B	470	D26B	726	D36B	982	D46B	1238	D56B	1494	D66B	1750	D76B	2006
D06C	216	D16C	472	D26C	728	D36C	984	D46C	1240	D56C	1496	D66C	1752	D76C	2008
D06D	218	D16D	474	D26D	730	D36D	986	D46D	1242	D56D	1498	D66D	1754	D76D	2010
D06E	220	D16E	476	D26E	732	D36E	988	D46E	1244	D56E	1500	D66E	1756	D76E	2012
D06F	222	D16F	478	D26F	734	D36F	990	D46F	1246	D56F	1502	D66F	1758	D76F	2014
D070	224	D170	480	D270	736	D370	992	D470	1248	D570	1504	D670	1760	D770	2016
D071	226	D171	482	D271	738	D371	994	D471	1250	D571	1506	D671	1762	D771	2018
D072	228	D172	484	D272	740	D372	996	D472	1252	D572	1508	D672	1764	D772	2020
D073	230	D173	486	D273	742	D373	998	D473	1254	D573	1510	D673	1766	D773	2022
D074	232	D174	488	D274	744	D374	1000	D474	1256	D574	1512	D674	1768	D774	2024
D075	234	D175	490	D275	746	D375	1002	D475	1258	D575	1514	D675	1770	D775	2026
D076	236	D176	492	D276	748	D376	1004	D476	1260	D576	1516	D676	1772	D776	2028
D077	238	D177	494	D277	750	D377	1006	D477	1262	D577	1518	D677	1774	D777	2030
D078	240	D178	496	D278	752	D378	1008	D478	1264	D578	1520	D678	1776	D778	2032
D079	242	D179	498	D279	754	D379	1010	D479	1266	D579	1522	D679	1778	D779	2034
D07A	244	D17A	500	D27A	756	D37A	1012	D47A	1268	D57A	1524	D67A	1780	D77A	2036
D07B	246	D17B	502	D27B	758	D37B	1014	D47B	1270	D57B	1526	D67B	1782	D77B	2038
D07C	248	D17C	504	D27C	760	D37C	1016	D47C	1272	D57C	1528	D67C	1784	D77C	2040
D07D	250	D17D	506	D27D	762	D37D	1018	D47D	1274	D57D	1530	D67D	1786	D77D	2042
D07E	252	D17E	508	D27E	764	D37E	1020	D47E	1276	D57E	1532	D67E	1788	D77E	2044
D07F	254	D17F	510	D27F	766	D37F	1022	D47F	1278	D57F	1534	D67F	1790	D77F	2046