

제 1 장 개요

1.1. IEC 61131-3	1-1
1.2.	1-1

제 2 장 소프트웨어 구조

2.1.	2-1
2.2. (Project)	2-1
2.3. (Configuration)	2-1
2.3.1. (Resource)	2-2
2.3.1.1. (Program)	2-2
2.3.1.2. (Resource Global Variable)	2-2
2.3.1.3. (Task)	2-3
2.3.2. (Configuration Global Variable)	2-4
2.3.3.	2-4

제 3 장 공통 요소

3.1.	3-1
3.1.1. (Identifiers)	3-1
3.1.2.	3-1
3.1.2.1. (Numeric Literals)	3-2
3.1.2.2. (Character String)	3-2
3.1.2.3. (Time Literals)	3-2
3.1.2.3.1. (Duration)	3-2
3.1.2.3.2. (Time Of Day And Date)	3-3
3.2.	3-4
3.2.1.	3-4
3.2.2.	3-5
3.2.3.	3-5
3.2.4.	3-6
3.3.	3-8
3.3.1.	3-8
3.3.2.	3-9
3.3.3.	3-11

목 차

3.4.	3-16
3.5.	3-17
3.5.1.	3-17
3.5.2.	3-18
3.5.3.	3-19

제 4 장 SFC(Sequential Function Chart)

4.1.	4-1
4.2. SFC	4-1
4.2.1.	4-1
4.2.2.	4-2
4.2.3.	4-2
4.2.4.	(Action Qualifier)	4-3
4.3.	4-8
4.3.1.	4-8
4.3.2.	4-8
4.3.3.	4-9
4.3.4.	4-9

제 5 장 IL(Instruction List)

5.1.	5-1
5.2.	(Current Result : CR)	5-1
5.3.	5-2
5.3.1.	5-2
5.3.2.	5-2
5.3.3.	5-3
5.3.3.1.	5-5
5.4.	5-24

제 6 장 LD(Ladder Diagram)

6.1.	6-1
6.2.	6-1
6.3.	6-2
6.4.	6-3
6.5.	6-4
6.6.	6-5

제 7 장 평선과 평선 블록

7.1.	7-1
7.1.1.	7-1
7.1.2.	7-10
7.1.2.1.	7-10
7.1.2.2.	7-10
7.1.3.	7-11
7.1.3.1.	7-11
7.1.3.2.	7-11
7.1.4.	7-11
7.1.5.	7-12
7.1.6.	7-12
7.1.7.	7-13
7.1.8.	7-13
7.2.	7-14
7.3.	7-14
7.3.1.	7-14
7.3.2.	7-14
7.3.3.	7-14
7.3.4.	7-14
7.4.	7-15

제 8 장 기본 평션/ 평션 블록 라이브러리

8.1	8-1
	ABS	8-2
	ACOS	8-3
	ADD	8-4
	ADD_TIME	8-5
	AND	8-6
	ARY_TO_STRING	8-7
	ASIN	8-8
	ATAN	8-9
	BCD_TO_***	8-10
	BOOL_TO_***	8-11
	BYTE_TO_***	8-12
	CONCAT	8-13
	CONCAT_TIME	8-14
	COS	8-15
	DATE_TO_***	8-16
	DELETE	8-17
	DI	8-18
	DINT_TO_***	8-20
	DIREC_IN	8-22
	DIREC_O	8-25
	DIV	8-27
	DIV_TIME	8-28
	DT_TO_***	8-29
	DWORD_TO_***	8-30
	EI	8-32
	EQ	8-33
	ESTOP	8-34
	EXP	8-35
	EXPT	8-36
	FIND	8-37
	GE	8-38
	GT	8-39
	INSERT	8-40
	INT_TO_***	8-41
	LE	8-43
	LEFT	8-44
	LEN	8-45
	LIMIT	8-46

LINT_TO_***	8-47
LN	8-49
LOG	8-50
LREAL_TO_***	8-51
LT	8-53
LWORD_TO_***	8-54
MAX	8-56
MID	8-57
MIN	8-58
MOD	8-59
MOVE	8-60
MUL	8-61
MUL_TIME	8-62
MUX	8-63
NE	8-64
NOT	8-65
NUM_TO_STRING	8-66
OR	8-67
REAL_TO_***	8-68
REPLACE	8-70
RIGHT	8-72
ROL	8-73
ROR	8-74
SEL	8-75
SHL	8-76
SHR	8-77
SIN	8-78
SINT_TO_***	8-79
SQRT	8-81
STOP	8-82
STRING_TO_***	8-83
STRING_TO_ARY	8-85
SUB	8-86
SUB_DATE	8-87
SUB_DT	8-88
SUB_TIME	8-89
SUB_TOD	8-90
TAN	8-91
TIME_TO_***	8-92
TOD_TO_***	8-93
TRUNC	8-94
UDINT_TO_***	8-95

목 차

UINT_TO_***	8-97
ULINT_TO_***	8-99
USINT_TO_***	8-101
WDT_RST	8-103
WORD_TO_***	8-105
XOR	8-106
8.2	8-108
ARY_ASC_TO_BCD	8-109
ARY_ASC_TO_BYTE	8-111
ARY_AVE_***	8-113
ARY_BCD_TO_ASC	8-115
ARY_BYTE_TO_ASC	8-117
ARY_CMP_***	8-119
ARY_FLL_***	8-121
ARY_MOVE	8-123
ARY_ROT_C_***	8-125
ARY_SCH_***	8-127
ARY_SFT_C_***	8-129
ARY_SWAP_***	8-131
ASC_TO_BCD	8-133
ASC_TO_BYTE	8-134
BCD_TO_ASC	8-135
BIT_BYTE	8-136
BMOV_B,W,D,L	8-137
BSUM_B,W,D,L	8-139
BYTE_BIT	8-140
BYTE_TO_ASC	8-141
BYTE_WORD	8-142
DEC_B,W,D,L	8-143
DECO_B,W,D,L	8-144
DEG_***	8-145
DIS_***	8-146
DWORD_LWORD	8-148
DWORD_WORD	8-149
ENCO_B,W,D,L	8-150
GET_CHAR	8-151
INC_B,W,D,L	8-152
LWORD_DWORD	8-153
MCS	8-154
MCSCLR	8-156
MEQ	8-157
PUT_CHAR	8-159

	RAD_***	8-160
	ROTATE_A_***	8-161
	ROTATE_C_***	8-164
	RTC_SET	8-166
	SEG	8-169
	SHIFT_A_***	8-171
	SHIFT_C_***	8-174
	SWAP_***	8-176
	UNI_***	8-177
	WORD_BYTE	8-179
	WORD_DWORD	8-180
	XCHG_***	8-181
8.3		8-183
	CTD	8-184
	CTU	8-186
	CTUD	8-188
	F_TRIG	8-190
	RS	8-191
	R_TRIG	8-192
	SEMA	8-193
	SR	8-196
	TOF	8-197
	TON	8-199
	TP	8-201
8.4		8-203
	CTR	8-204
	DUTY	8-206
	FIFO_***	8-208
	LIFO_***	8-212
	SCON	8-216
	TMR	8-218
	TMR_FRK	8-220
	TMR_UINT	8-222
	TOF_RST	8-224
	TOF_UINT	8-226
	TON_UINT	8-228
	TP_RST	8-230
	TP_UINT	8-232
	TRTG	8-234
	TRTG_UINT	8-236

1.

GLOFA PLC GM1 GM7
GLOFA PLC IEC(International Electrotechnical Commission -)

1.1. IEC 61131-3

IEC 가
가 , PLC 가
가
가

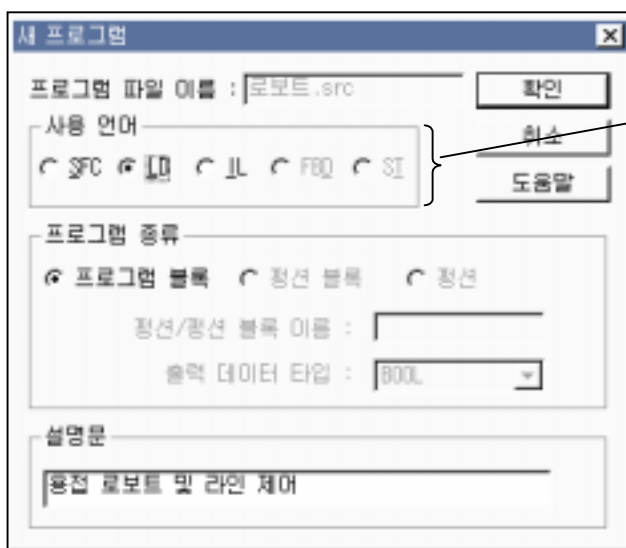
1.2.

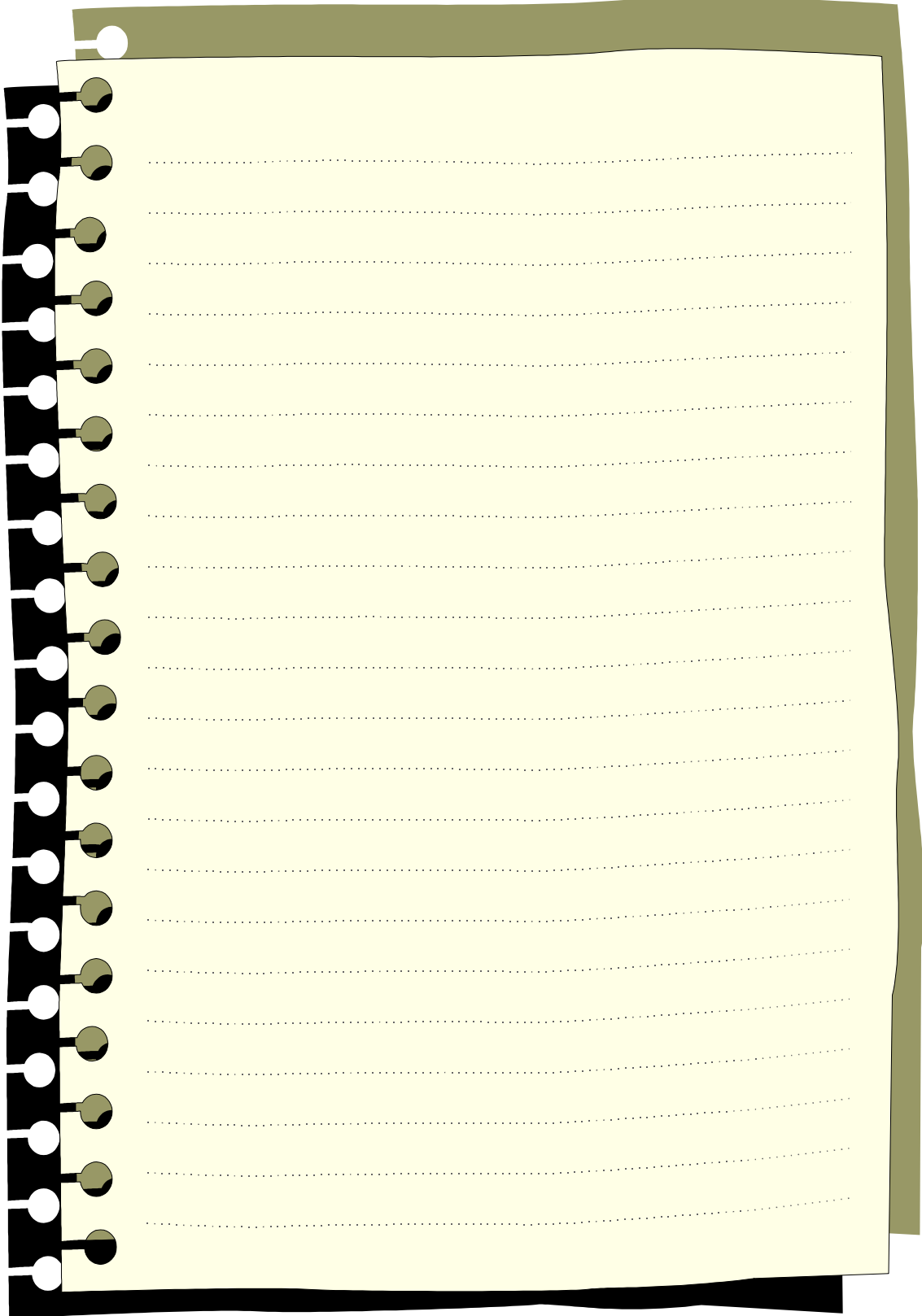
IEC PLC , SFC

- a) LD(Ladder Diagram) :
- b) FBD(Function Block Diagram) :

- a) IL(Instruction List) :
- b) ST(Structured Text) :
- SFC(Sequential Function Chart)

GLOFA PLC IL LD, SFC

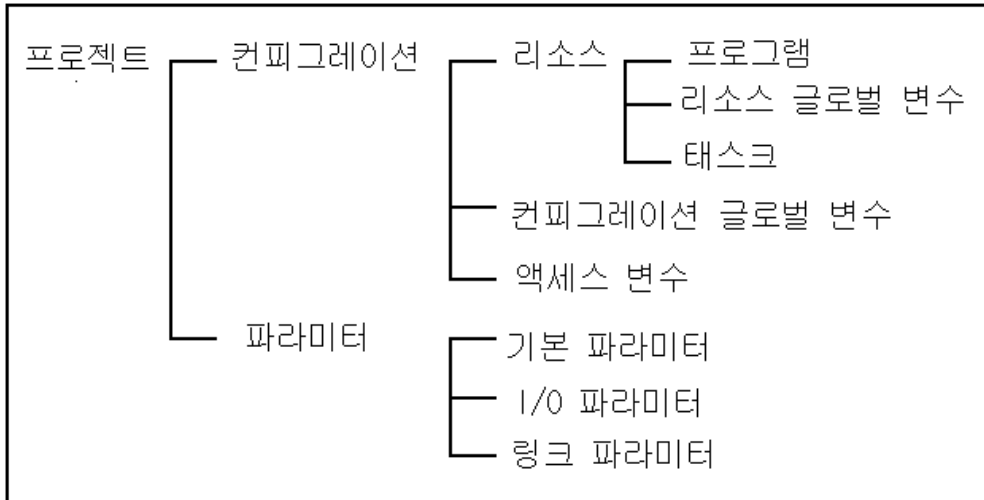




2.

2.1.

PLC
 GLOFA PLC PLC
 PLC



2.2. (Project)

GLOFA PLC
 PLC
 ,가 (PLC)
 , I/O
 PLC 가 , , PLC
 가
 GMWIN

2.3. (Configuration)

PLC PLC CPU
 I/O , PLC CPU
 가 GM1 PLC 4 CPU
 PLC 가
 PLC PLC PLC 가
 8 3.1.1.
 , 가

2.3.1. (Resource)

CPU . GM1 4
. GM2 GM7 1
가
8 , 3.1.1.
가

2.3.1.1. (Program)

PLC . GLOFA PLC
A , B 1
, C
()

—

:

가

—

3.5.2.

2.3.1.2. (Resource Global Variable)

VAR_EXTERNAL

—

3.3.2.

2.3.1.3. (Task)

3가 가

- 1) (Single) : BOOL
- 2) (Interval) : 3.1.2.3.1.
- 3) (Interrupt) : 0n

	%IX0.0.1	%IX0.0.1 0n 1
	T#1S	1
	4	4가 0n 1

0 7 가 가 , 가

_ERR_SYS, _H_INIT _INIT 가
 _ERR_SYS : (GM1,2 .)
 _H_INIT :
 _INIT : /

2.

2.3.2. (Configuration Global Variable)

VAR_EXTERNAL

3.3.2.

가

GM1

2.3.3.

PLC

()

3.

GLOFA PLC (, ,) IL, LD, SFC 가 .

3.1.

3.1.1. (Identifiers)

(_) , , 가

(Space)

16 ,

8

	IW210, IW215Z, QX75, IDENT
,	LIM_SW_2, LIMSW5, ABCD, AB_CD
	_MAIN, _12V7, _ABCD

3.1.2.

GLOFA PLC (Numeric Literals) (Character String),
(Time Literals) .

	-12, 0, 123_456, +986
	-12.0, 0.0, 0.456, 3.14159_26
	-1.34E-12, 1.0E+6, 1.234E6
2	2#1111_1111, 2#1110_0000
8	8#377(255) 8#340(224)
16	16#FF(255) 16#E0(224)
BOOL	0, 1, TRUE, FALSE

3.1.2.1. (Numeric Literals)

(Integer Literals) (Real Literals)가
 ()가
 (Exponent) +, - 가 'E'
 가 가
) 12E-5 (×) 12.0E-5 ()
 2,8,16 가 , #
 16 0 - 9, A - F a - f
 (+, -)가
 BOOL (Boolean Data) 0 1

3.1.2.2. (Character String)

(') 가
 16 , 30
 'CONVEYER'

3.1.2.3. (Time Literals)

(Control Event) (Elapsed Time)
 (Duration) ,
 (Time Of Day And Date)

3.1.2.3.1. (Duration)

'T#' 't#'
 (d), (h), (m), (s), ms
 ms
 ()
 (Overflow) , ms
 가 T#49d17h2m47s295ms
 (ms 32)
 (s) 3
 ms
 가

(Underline)	T#14ms, T#14.7s, T#14.7m, T#14.7h t#14.7d, t#25h15m, t#5d14h12m18s356ms

3.1.2.3.2. (Time Of Day And Date)

, , 3 가 가 .

	D#
	TOD#
	DT#

1984 1 1 .

가 . (1ms)

(Overflow)가 .

	D#1984-06-25 d#1984-06-25
	TOD#15:36:55.36 tod#15:36:55.369
	DT#1984-06-25-15:36:55.36 dt#1984-06-25-15:36:55.369

3.

3.2.

가

3.2.1.

GLOFA PLC

			()	
1	SINT	Short Integer	8	-128 127
2	INT	Integer	16	-32768 32767
3	DINT	Double Integer	32	-2147483648 2147483647
4	LINT	Long Integer	64	-2^{63} $2^{63}-1$
5	USINT	Unsigned Short Integer	8	0 255
6	UINT	Unsigned Integer	16	0 65535
7	UDINT	Unsigned Double Integer	32	0 4294967295
8	ULINT	Unsigned Long Integer	64	0 $2^{64}-1$
9	REAL	Real Numbers	32	-3.402823E38 -1.401298E-45 1.401298E-45 3.402823E38
10	LREAL	Long Reals	64	-1.7976931E308 -4.9406564E-324 4.9406564E-324 1.7976931E308
11	TIME	Duration	32	T#0S T#49D17H2M47S295MS
12	DATE	Date	16	D#1984-01-01 D#2163-6-6
13	TIME_OF_DAY	Time Of Day	32	TOD#00:00:00 TOD#23:59:59.999
14	DATE_AND_TIME	Date And Time Of Day	64	DT#1984-01-01-00:00:00 DT#2163-12-31-23:59:59.999
15	STRING	Character String	30*8	-
16	BOOL	Boolean	1	0,1
17	BYTE	Bit String Of Length 8	8	16#0 16#FF
18	WORD	Bit String Of Length 16	16	16#0 16#FFFF
19	DWORD	Bit String Of Length 32	32	16#0 16#FFFFFFFF
20	LWORD	Bit String Of Length 64	64	16#0 16#FFFFFFFFFFFFFFFF


LINT, ULINT, REAL, LREAL, LWORD GM1, GM2

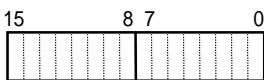
3.

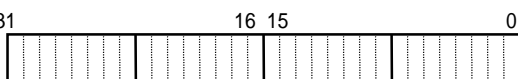
3.2.4.

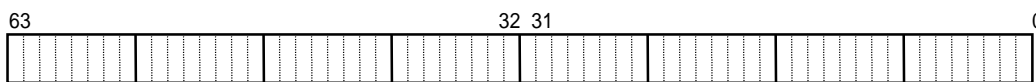
Bit String

BOOL  1 , : 0, 1

BYTE  8 , : 2#0000_0000 2#1111_1111, 16#00 16#FF

WORD  16 , : 2#0000_0000_0000_0000 2#1111_1111_1111_1111
16#0000 16#FFFF

DWORD  32 , : 2#0000_...000 2#1111_...111
16#00000000 16#FFFFFFFF

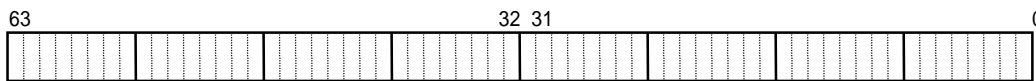
LWORD  64 , : 2#0000_...000 2#1111_...111, 16#0000000000000000 16#FFFFFFFFFFFFFFFF

Unsigned Integer

USINT  8 , : 0 255

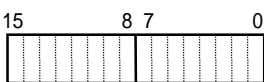
UINT  16 , : 0 65,535

UDINT  32 , : 0 4,294,967,295

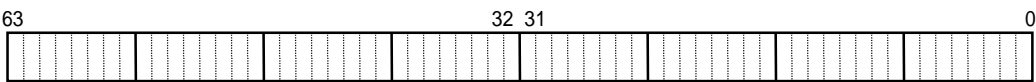
ULINT  64 , : 0 $2^{64}-1$

Integer (2' Complement)

SINT  8 , : -128 127

INT  16 , : -32,768 32,767

DINT  32 , : -2,147,483,648 2,147,483,647

LINT  64 , : -2^{63} $2^{63}-1$

3.

3.3.

가 . PLC ,
가 .

3.3.1.

2 가 가 () , PLC . () . (가) (%)

1	I	(Input Location)
2	Q	(Output Location)
3	M	(Memory Location)

1	X	1
2	None	1
3	B	1 (8)
4	W	1 (16)
5	D	1 (32)
6	L	1 (64)

%[] [] n1.n2.n3

	I, Q	
n1	(0)	[] n3 (0)
n2	(0)	n1 n2 (0) : 가
n3	[] n3 (0)	

—						
%QX3.1.4	%Q3.1.4	3	1	4	(1)
%IW2.4.1		2	4		1	(16)
%MD48		48				
%MW40.3		40			3	
		(,)

가

1

3.3.2.

(, ,)
가

1) : 가

VAR	
VAR_RETAIN	
VAR_CONSTANT	
VAR_EXTERNAL	VAR_GLOBAL

—
VAR_GLOBAL_RETAIN, VAR_GLOBAL_CONSTANT VAR_EXTERNAL VAR_GLOBAL,

2) :

3) : 가

---- 가 ()
(AT) ---- 가 ()

가 VAL1 BOOL
가
가
가
가 %I %Q
가 가
(Initial Value) : 3.2.3.

VAR_EXTERNAL

%I %Q

PLC

VAR_RETAIN

가

(Retention)

1)

2)

가

VAR_RETAIN

가

%I %Q

VAR_RETAIN, VAR_CONSTANT

STRING
(Scope),

가

VAR_EXTERNAL

가

I_VAL	VAR	INT	1234	
BIPOLAR	VAR_RETAIN	REAL		
LIMIT_SW	VAR	BOOL		%IX1.0.2
GLO_SW	VAR_EXTERNAL	DWORD		
READ_BUF	VAR	ARRAY OF INT[10]		

3.3.3.

가

‘CPU

1)

_ERR	BOOL	
_LER	BOOL	
_T20MS	BOOL	20 ms
_T100MS	BOOL	100 ms
_T200MS	BOOL	200 ms
_T1S	BOOL	1
_T2S	BOOL	2
_T10S	BOOL	10
_T20S	BOOL	20
_T60S	BOOL	60
_ON	BOOL	On
_OFF	BOOL	Off
_1ON	BOOL	1 On
_1OFF	BOOL	1 Off
_STOG	BOOL	
_INIT_DONE	BOOL	
_RTC_DATE	DATE	RTC
_RTC_TOD	TOD	RTC
_RTC_WEEK	UINT	RTC

3.

2)

_CNF_ER	WORD	()
_CPU_ER	BOOL	CPU
_IO_TYER	BOOL	
_IO_DEER	BOOL	
_FUSE_ER	BOOL	Fuse
_IO_RWER	BOOL	/ ()
_SP_IFER	BOOL	/ ()
_ANNUN_ER	BOOL	
_WD_ER	BOOL	Scan Watch-Dog
_CODE_ER	BOOL	
_STACK_ER	BOOL	Stack Overflow
_P_BCK_ER	BOOL	

3)

_CNF_ER_M	BYTE	()
_IO_DEER_M	BOOL	
_FUSE_ER_M	BOOL	
_IO_RWER_M	BOOL	/
_SP_IFER_M	BOOL	/
_ANNUN_ER_M	BOOL	

4)

_CNF_WAR	WORD	()
_RTC_ERR	BOOL	RTC
_D_BCK_ER	BOOL	
_H_BCK_ER	BOOL	가
_AB_SD_ER	BOOL	(Abnormal Shutdown)
_TASK_ERR	BOOL	(Task) (,)
_BAT_ERR	BOOL	
_ANNUN_WR	BOOL	

_HSPMT1_ER	BOOL	1
_HSPMT2_ER	BOOL	2
_HSPMT3_ER	BOOL	3
_HSPMT4_ER	BOOL	4

5)

_IO_TYER_N	UINT	
_IO_TYERR	ARRAY OF BYTE	
_IO_DEER_N	UINT	
_IO_DEERR	ARRAY OF BYTE	
_FUSE_ER_N	UINT	Fuse
_FUSE_ERR	ARRAY OF BYTE	Fuse
_IO_RWER_N	UINT	/
_IO_RWERR	ARRAY OF BYTE	/
_IP_IFER_N	UINT	/
_IP_IFERR	ARRAY OF BYTE	/
_ANC_ERR	ARRAY OF UINT	
_ANC_WAR	ARRAY OF UINT	
_ANC_WB	ARRAY OF BIT	Map
_TC_BMAP	ARRAY OF BYTE	
_TC_CNT	UINT	
_BAT_ER_TM	DT	
_AC_F_CNT	UINT	
_AC_F_TM	ARRAY OF DT	

6)

_CPU_TYPE	UINT	
_VER_NUM	UINT	PLC O/S
_MEM_TYPE	UINT	
_SYS_STATE	WORD	PLC
_GMWIN_CNF	BYTE	PADT
_RST_TY	BYTE	
_INIT_RUN	BIT	
_SCAN_MAX	UINT	(ms)
_SCAN_MIN	UINT	(ms)
_SCAN_CUR	UINT	(ms)
_STSK_NUM	UINT	
_STSK_MAX	UINT	(ms)
_STSK_MIN	UINT	(ms)
_STSK_CUR	UINT	(ms)
_RTC_TIME	ARRAY OF BYTE	
_SYS_ERR	UINT	

7)

[n

(n = 0 - 7)]

_CnVERNO	UINT	No.
_CnSTNOH _CnSTNOL	UINT	
_CnTXECNT	UINT	
_CnRXECNT	UINT	
_CnSVCFCNT	UINT	
_CnSCANMX	UINT	(1ms)
_CnSCANAV	UINT	(1ms)
_CnSCANMN	UINT	(1ms)
_CnLINF	UINT	
_CnCRDER	BOOL	(= 1)
_CnSVBSY	BOOL	RAM (=1)
_CnIFERR	BOOL	(= 1)
_CnINRING	BOOL	(IN_RING = 1)

8) I/O [m (m = 0 - 7)]

_FSMm_reset	BOOL(Write 가)	I/O	(=1)
_FSMm_io_reset	BOOL(Write 가)	I/O	(=1)
_FSMm_st_no	USINT(Write 가)	I/O	

9) [m (m = 1,2,3,4)]

_HSmRLINK	BIT	RUN_LINK
_HSmLTRBL	BIT	(Link Trouble)
_HSmSTATE	ARRAY OF BIT	k
_HSmMOD	ARRAY OF BIT	k (Run = 1, = 0)
_HSmTRX	ARRAY OF BIT	k (= 1, = 0)
_HSmERR	ARRAY OF BIT	k (= 0, = 1)

3.

3.4.

ACTION ... END_ACTION
ARRAY ... OF
AT
CASE ... OF ... ELSE ... END_CASE CONFIGURATION ... END_CONFIGURATION
DATE#, D# DATE_AND_TIME#, DT#
EXIT
FOR ... TO ... BY ... DO ... END_FOR
FUNCTION ... END_FUNCTION
FUNCTION_BLOCK ... END_FUNCTION_BLOCK
IF ... THEN ... ELSIF ... ELSE ... END_IF
OK
(IL) (ST)
PROGRAM
PROGRAM ... END_PROGRAM
REPEAT ... UNTIL ... END_REPEAT
RESOURCE ... END_RESOURCE
RETAIN
RETURN
STEP ... END_STEP
STRUCTURE ... END_STRUCTURE
T#
TASK ... WITH
TIME_OF_DAY#, TOD#
TRANSITION ... FROM... TO ... END_TRANSITION
TYPE ... END_TYPE
VAR ... END_VAR VAR_INPUT ... END_VAR VAR_OUTPUT ... END_VAR VAR_IN_OUT ... END_VAR VAR_EXTERNAL ... END_VAR
VAR_ACCESS ... END_VAR
VAR_GLOBAL ... END_VAR
WHILE ... DO ... END_WHILE
WITH

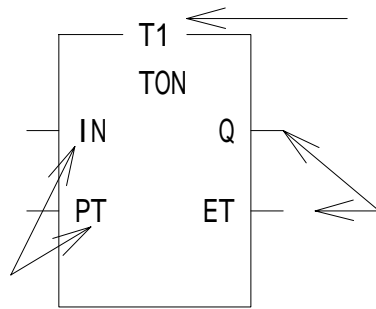
3.5.

.()

3.5.1.

—
A
IN1 IN2 100
1 <= IN1 + IN2 + 100
2 <= IN1 + IN2 * 100
1 2, 2

—
B
1 <= IN1 + IN2 + Val
Val <= 1 (Val)
Val 가
1 IN1 IN2
Val A A
IN1 20, IN2 가 30 1 150
가
VAR_EXTERNAL



가
TON T1 T2 T1 T2
가
(.)
Q
T1.Q, T2.Q , ET
T1.ET, T2.ET
(Return)

(AT) VAR_EXTERNAL

3.5.3.

/ 가

4. SFC(Sequential Function Chart)

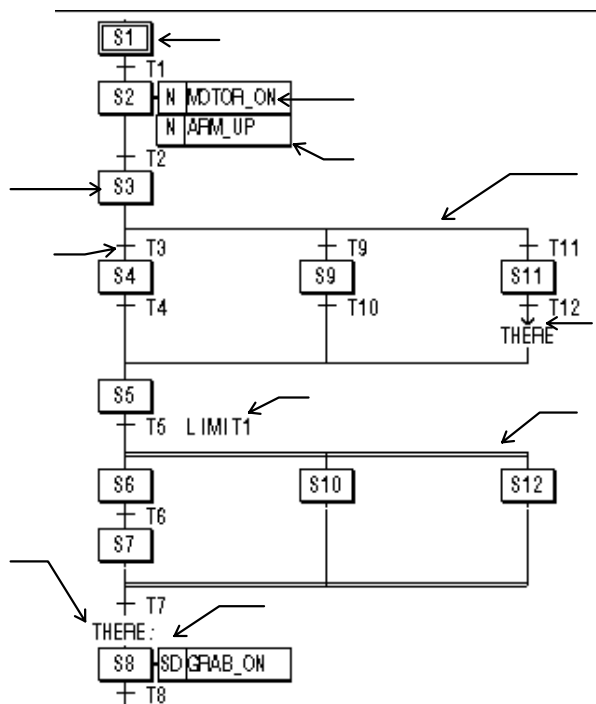
4.1.

SFC PLC

SFC

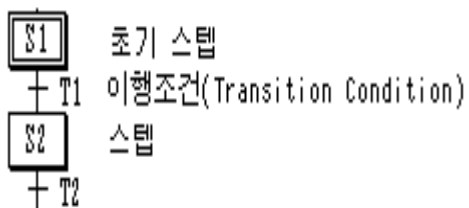
SFC 가

SFC



4.2. SFC

4.2.1.



(S1)

(Transition Condition)

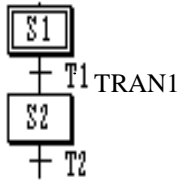
(S1)

(S2)

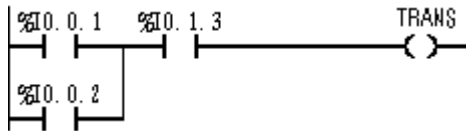
4. SFC

4.2.2.

PLC IL LD
 BOOL
 TRANS 가
 가 1

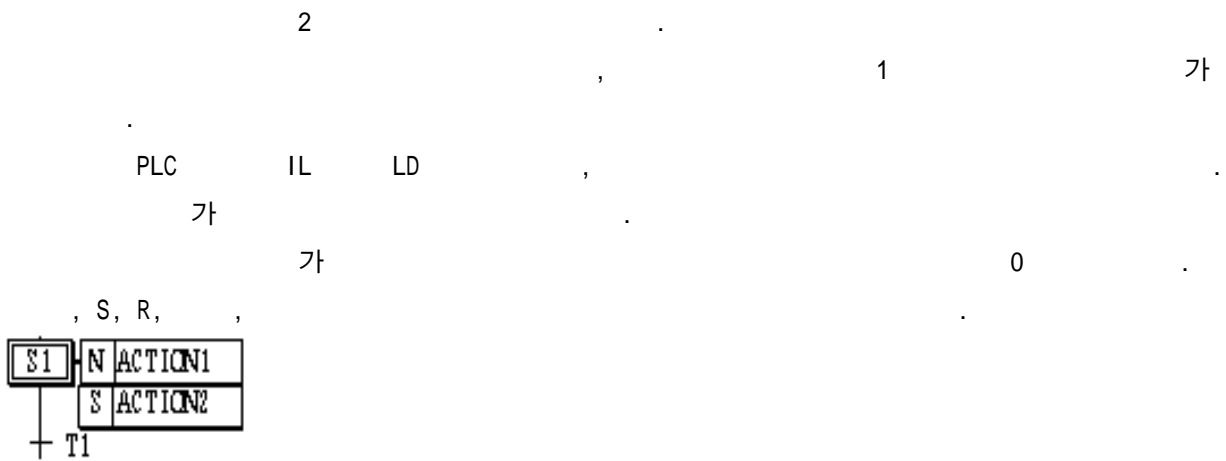


TRAN1

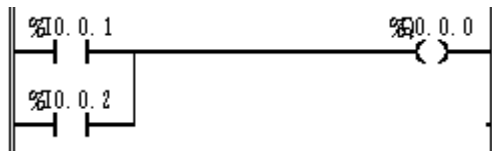


TRANS 가 On S1 S2 가 가
 TRANS
 TRANS

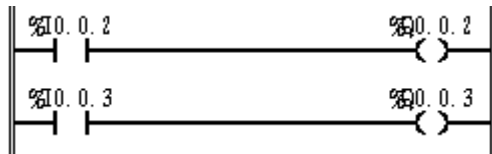
4.2.3.



ACTION1



ACTION2



- ACTION1 S1

- ACTION2 S1 R

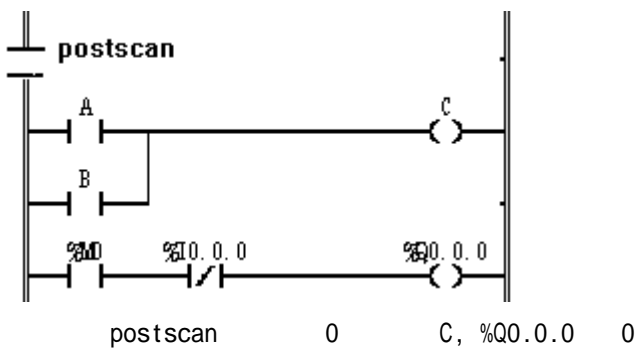
S1

- (Post Scan)

(0)

0

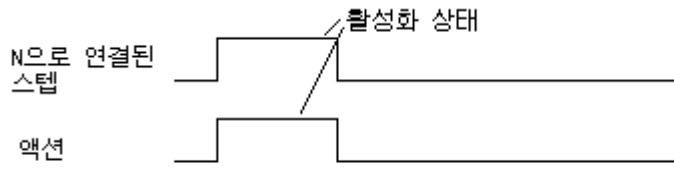
, S, R



4.2.4. (Action Qualifier)

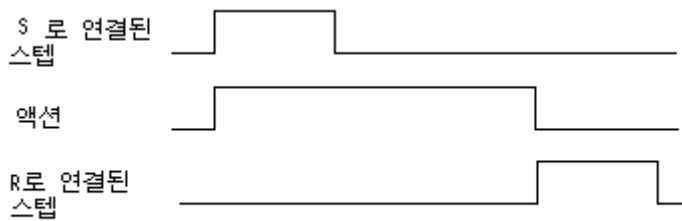
가

1) N(Non-Stored)



2) S(Set)

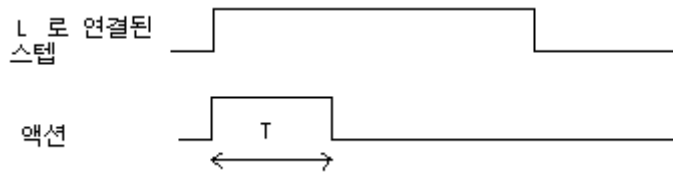
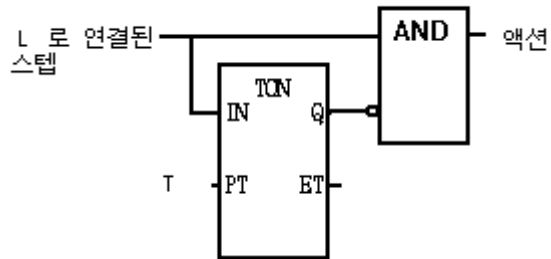
R 가



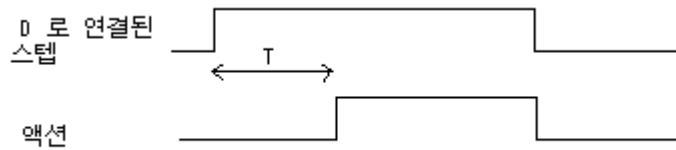
3) R(Overriding Reset)

S, SD, DS, SL

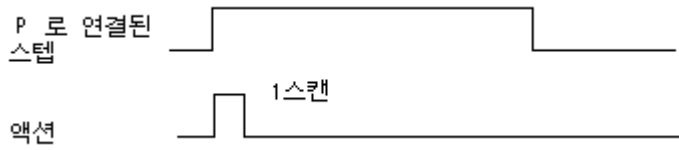
4) L(Time Limited)



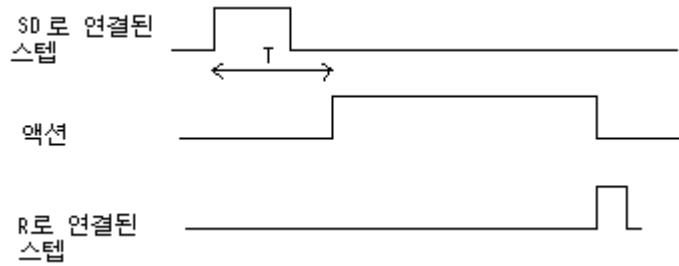
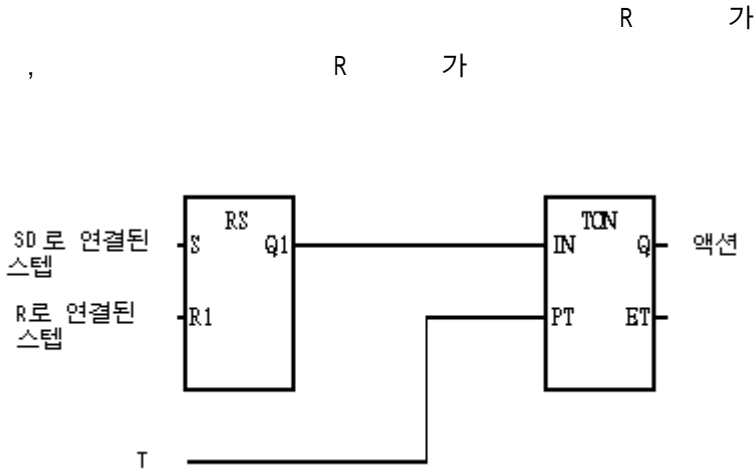
5) D(Time Delayed)



6) P(Pulse)

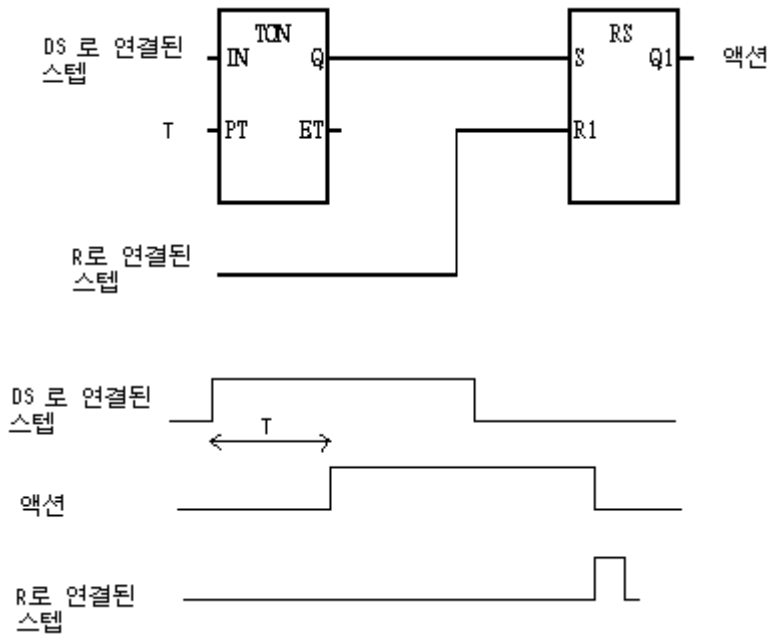


7) SD(Stored & Time Delayed)



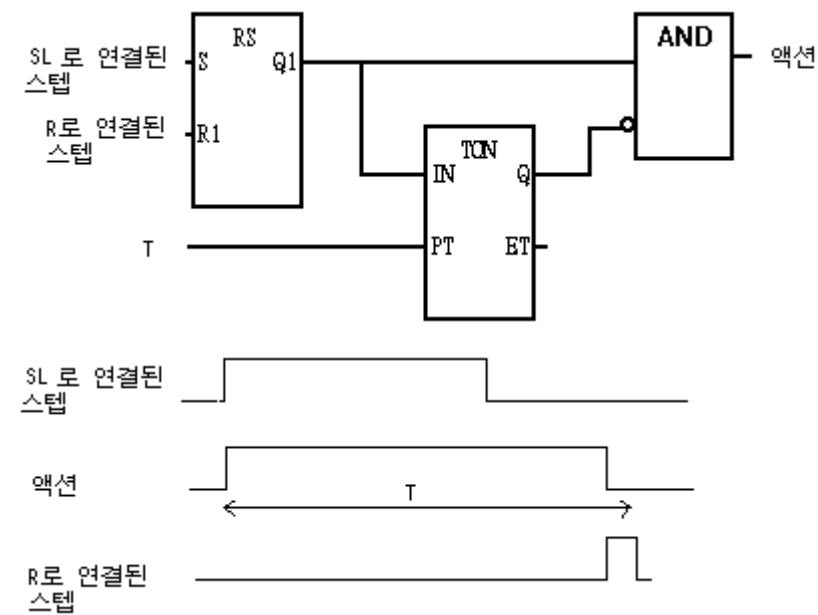
8) DS(Delayed & Stored)

R 가
R 가



9) SL(Stored & Timed Limited)

R 가



4.3.

4.3.1.

2
2



[] []

1

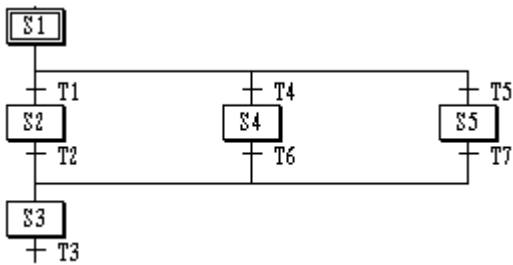
가

4.3.2.

2

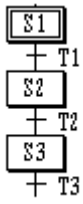
1

—

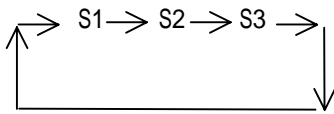


- * T1 1
S1 -> S2 -> S3 가 .
- * T4 1
S1 -> S4 -> S3 가 .
- * T5 1
S1 -> S5 -> S3 가 .
- 1 가 .
- * T1, T4 1
S1 -> S2 -> S3 가 .
- * T4, T5 1
S1 -> S4 -> S3 가 .

—



*

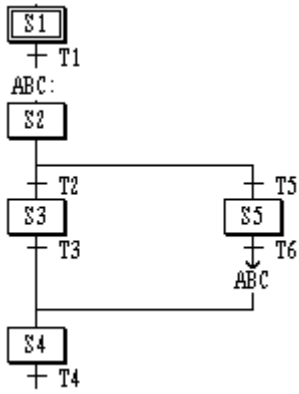


SFC

가

—

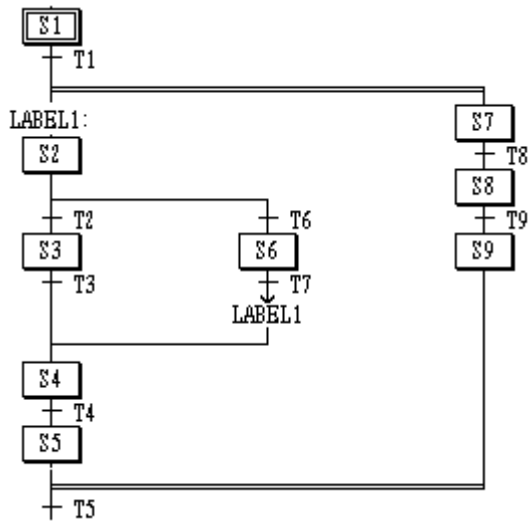
1)



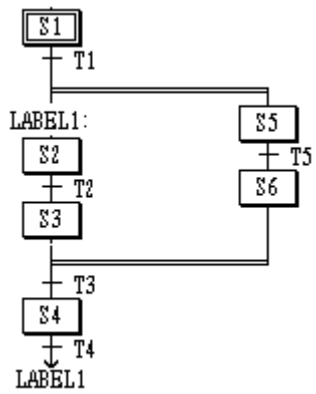
- S5

S2

2)

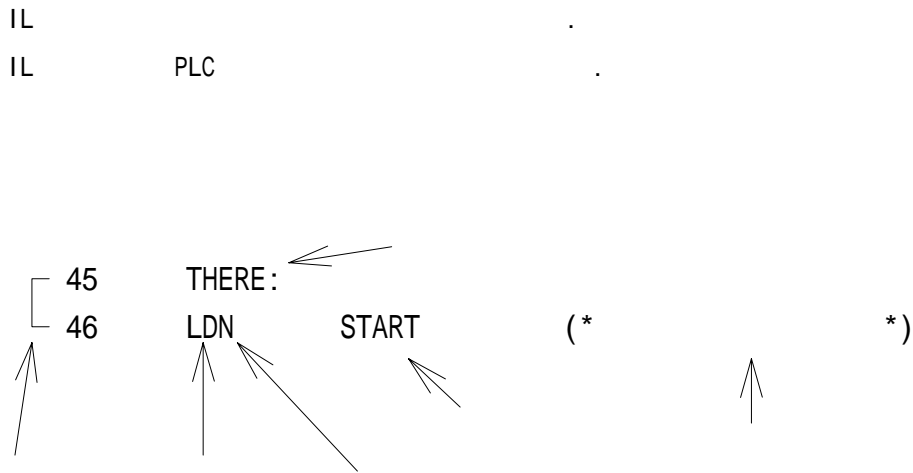


3)

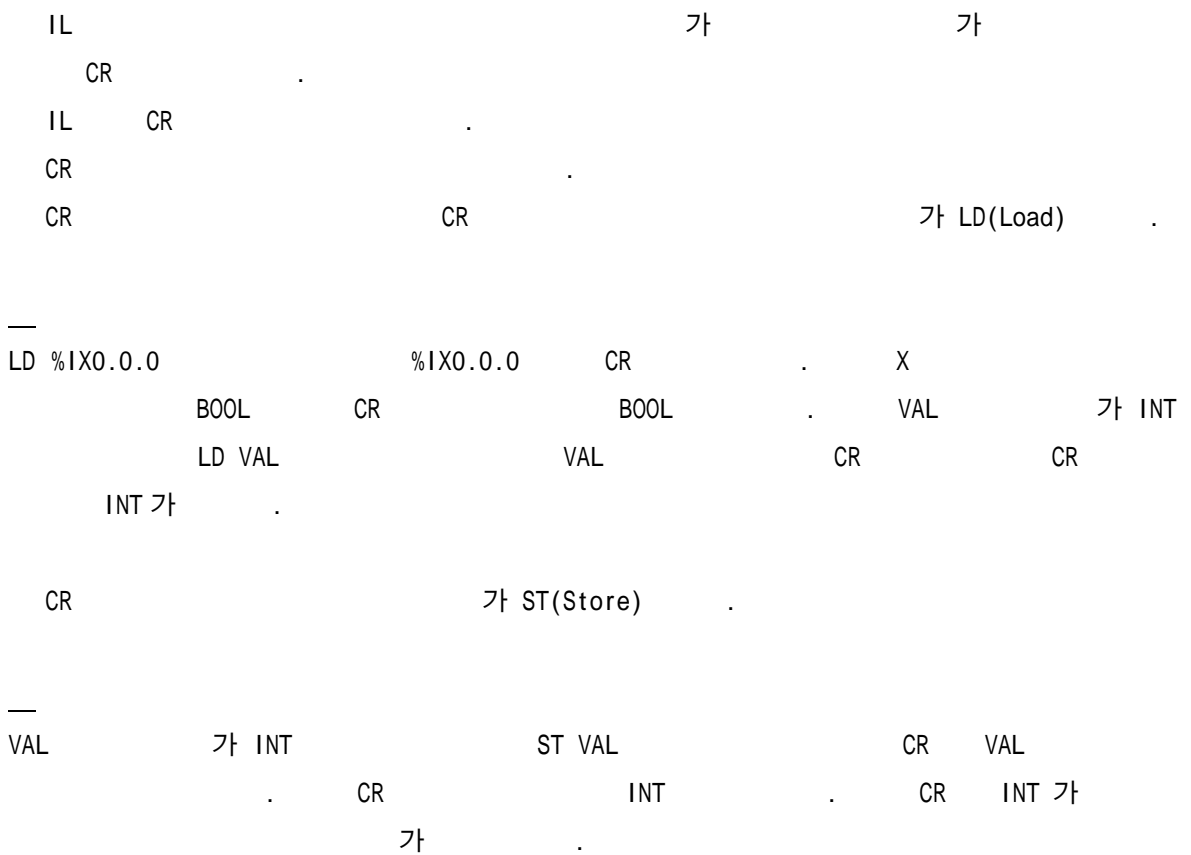


5. IL(Instruction List)

5.1.



5.2. (Current Result : CR)



5. IL

```
LD %IX0.0.0
ST VAL ( VAL INT 가 )
          CR BOOL          CR INT
          가
```

```
LD %IX0.0.0
ST START
LD 20
ST VAL ( START BOOL , VAL INT 가 )
          CR
```

5.3.

IL
가

5.3.1.

(:)

5.3.2.

N, (, C 가
" N " BOOL (Boolean Negation)

```
ANDN %IX2.0.0
      CR <= CR AND NOT %IX2.0.0
N JMP CAL, RET          CR BOOL 0
```

" (" ") "
IL CR CR 가 ,
CR

(가
)		

```

—
AND( %IX1.0.0
OR  %IX2.0.0
)
CR <= CR AND ( %IX1.0.0 OR %IX2.0.0 )
AND          )가
%IX1.0.0 OR %IX2.0.0
" C "          CR  BOOL 1

```

```

—
JMPC  THERE
CR  BOOL 1  THERE

```

5.3.3.

1	LD	N		(CR)
2	ST	N		
3	S		BOOL	BOOL 1 BOOL 1
	R		BOOL	BOOL 1 BOOL 0
4	AND	N, (AND
5	OR	N, (OR
6	XOR	N, (XOR
7	ADD	(
8	SUB	(
9	MUL	(
10	DIV	(
11	GT	(: > ()
12	GE	(: >= ()
13	EQ	(: = ()
14	NE	(: <> ()
15	LE	(: <= ()
16	LT	(: < ()
17	JMP	C,N		
18	CAL	C,N		
19	RET	C,N		
20)			‘(’

4 16

CR <== CR

, CR

CR

—
AND %IX1.0.0

CR <= CR AND %IX1.0.0

CR

BOOL

CR

—
GT %MW10 CR

10

CR

BOOL 1

,

0

CR

CR

—
LD VAL (a)

EQ GROSS (b)

AND %IX0.0.0 (c)

ST START (d)

(START BOOL , VAL GROSS INT 가)

(a) VAL INT CR

(b) CR GROSS INT

BOOL 1 , BOOL 0 CR CR

INT BOOL (c) (d)

5.3.3.1.

(1) LD

	CR	CR
	N : 가 BOOL	CR
	가	가
LD TRUE	BOOL 1	CR CR
LD INT_VALUE	INT INT_VALUE	CR
LD T#1S	T#1S	CR
LDN B_VALUE	BOOL B_VALUE	CR
	CR	BOOL

(2) ST

	CR	CR
	N : CR	BOOL CR
	가	가
	CR	
LD FALSE	BOOL 0	CR CR
ST B_VALUE1	CR 0	BOOL B_VALUE1
STN B_VALUE2	CR (1)	BOOL B_VALUE2
LD INT_VALUE	INT INT_VALUE	CR
ST I_VALUE1	CR	INT I_VALUE1
LD D#1995-12-25	D#1995-12-25	CR
ST D_VALUE1	CR	DATE D_VALUE1

(3) S(Set)

	CR	BOOL 1	BOOL	1
	CR	BOOL 0		
	CR			
		BOOL	가	
	LD	FALSE	BOOL 0	CR CR
	S	B_VALUE1	CR 0	
			B_VALUE1	
	LD	TRUE	BOOL 1	CR CR
			BOOL	
	S	B_VALUE2	CR 1	BOOL B_VALUE2
			1	

(4) R(Reset)

	CR	BOOL 1	BOOL	0
	CR	BOOL 0		
	CR			
		BOOL	가	
	LD	FALSE	BOOL 0	CR CR
	R	B_VALUE1	CR 0	
			B_VALUE1	
	LD	TRUE	BOOL 1	CR CR
			BOOL	
	R	B_VALUE2	CR 1	BOOL B_VALUE2
			0	CR
	ST	B_VALUE3	CR (1)	BOOL B_VALUE3

(5) AND

	CR	AND	CR
	CR		
	N :	BOOL	CR
	(:	BOOL	CR
		CR	()
	BOOL, BYTE, WORD, DWORD, LWORD 가		
	가		
LD	B_VALUE1	BOOL B_VALUE1	CR
		CR	BOOL
AND	B_VALUE2	CR	BOOL B_VALUE2 AND
		CR	
ANDN	B_VALUE3	CR	BOOL B_VALUE3 AND
		CR	
ST	B_VALUE4	CR	BOOL B_VALUE4
		B_VALUE4 <== B_VALUE1 AND B_VALUE2 AND NOT(B_VALUE3)	
LD	W_VALUE1	WORD W_VALUE1	CR
		CR	WORD
AND	W_VALUE2	CR	WORD W_VALUE2 AND
		CR	
ST	W_VALUE3	CR	WORD W_VALUE3
		W_VALUE3 <== W_VALUE1 AND W_VALUE2	
LD	B_VALUE1	BOOL B_VALUE1	CR
		CR	BOOL
AND(B_VALUE2	CR	BOOL B_VALUE2
		CR	
OR	B_VALUE3	CR	BOOL B_VALUE3 OR
		CR	
)		CR	CR AND
		CR	
ST	B_VALUE4	CR	BOOL B_VALUE4
		B_VALUE4 <== B_VALUE1 AND (B_VALUE2 OR B_VALUE3)	

(6) OR

	CR	OR	CR
	CR		
	N :	BOOL	CR
	(:	BOOL	CR
		CR	()
	BOOL, BYTE, WORD, DWORD, LWORD	가	
	가		
LD	B_VALUE1	BOOL	B_VALUE1 CR
		CR	BOOL
OR	B_VALUE2	CR	BOOL B_VALUE2 OR
		CR	
ORN	B_VALUE3	CR	BOOL B_VALUE3
		OR	CR
ST	B_VALUE4	CR	BOOL B_VALUE4
			B_VALUE4 <== B_VALUE1 OR B_VALUE2 OR NOT(B_VALUE3)
LD	W_VALUE1	WORD	W_VALUE1 CR
		CR	WORD
OR	W_VALUE2	CR	WORD W_VALUE2 OR
		CR	
ST	W_VALUE3	CR	WORD W_VALUE3
			W_VALUE3 <== W_VALUE1 OR W_VALUE2
LD	B_VALUE1	BOOL	B_VALUE1 CR
		CR	BOOL
OR(B_VALUE2	CR	BOOL B_VALUE2
		CR	
AND	B_VALUE3	CR	BOOL B_VALUE3 AND
		CR	
)		CR	CR OR
		CR	
ST	B_VALUE4	CR	BOOL B_VALUE4
			B_VALUE4 <== B_VALUE1 OR (B_VALUE2 AND B_VALUE3)

(7) XOR

	CR	XOR	CR
	CR		
	N :	BOOL	CR
	(:	BOOL	CR
		CR	()
	BOOL, BYTE, WORD, DWORD, LWORD 가		
	가		
LD	B_VALUE1	BOOL	B_VALUE1 CR
		CR	BOOL
XOR	B_VALUE2	CR	BOOL B_VALUE2 XOR
		CR	
XORN	B_VALUE3	CR	BOOL B_VALUE3 XOR
		CR	
ST	B_VALUE4	CR	BOOL B_VALUE4
		B_VALUE4 <== B_VALUE1 XOR B_VALUE2 XOR NOT(B_VALUE3)	
LD	W_VALUE1	WORD	W_VALUE1 CR
		CR	WORD
XOR	W_VALUE2	CR	WORD W_VALUE2 XOR
		CR	
ST	W_VALUE3	CR	WORD W_VALUE3
		W_VALUE3 <== W_VALUE1 XOR W_VALUE2	
LD	B_VALUE1	BOOL	B_VALUE1 CR
		CR	BOOL
XOR(B_VALUE2	CR	BOOL B_VALUE2
		CR	
AND	B_VALUE3	CR	BOOL B_VALUE3 AND
		CR	
)		CR	CR XOR
		CR	
ST	B_VALUE4	CR	BOOL B_VALUE4
		B_VALUE4 <== B_VALUE1 XOR (B_VALUE2 AND B_VALUE3)	

(8) ADD

	CR	CR
	CR	CR
	(: CR)	CR . (
	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL 가 가	
LD I_VALUE1	INT I_VALUE1	CR
ADD I_VALUE2	CR INT I_VALUE2	CR
ST I_VALUE3	CR INT I_VALUE3	I_VALUE3 <== I_VALUE1 + I_VALUE2
LD D_VALUE1	DINT D_VALUE1	CR
ADD(D_VALUE2	CR DINT	DINT
DIV D_VALUE3	CR DINT D_VALUE3	CR
)	CR	CR
ST D_VALUE4	CR DINT B_VALUE4	D_VALUE4 <== D_VALUE1 + (D_VALUE2 / D_VALUE3)

(9) SUB

	CR	CR
	CR	CR
	(: CR	CR .(
)	
	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL	
	가	가
LD	I_VALUE1	INT I_VALUE1 CR
		CR INT
SUB	I_VALUE2	CR INT I_VALUE2
		CR
ST	I_VALUE3	CR INT I_VALUE3
		I_VALUE3 <== I_VALUE1 - I_VALUE2
LD	D_VALUE1	DINT D_VALUE1 CR
		CR DINT
SUB(D_VALUE2	CR DINT
		D_VALUE2 CR
MUL	D_VALUE3	CR DINT D_VALUE3
		CR
)		CR CR
		CR
ST	D_VALUE4	CR DINT B_VALUE4
		D_VALUE4 <== D_VALUE1 - (D_VALUE2 * D_VALUE3)

(10) MUL

	CR	CR	.
	CR		.
	(: CR	CR	.(
)		
	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL 가 . 가 .		
LD	I_VALUE1	INT I_VALUE1	CR .
		CR INT	.
MUL	I_VALUE2	CR INT I_VALUE2	
		CR	.
ST	I_VALUE3	CR INT I_VALUE3	.
		I_VALUE3 <== I_VALUE1 * I_VALUE2	
LD	D_VALUE1	DINT D_VALUE1	CR .
		CR DINT	.
MUL(D_VALUE2	CR DINT	DINT
		D_VALUE2 CR	.
SUB	D_VALUE3	CR DINT D_VALUE3	
		CR	.
)		CR	CR
		CR	.
ST	D_VALUE4	CR DINT B_VALUE4	.
		D_VALUE4 <== D_VALUE1 * (D_VALUE2 - D_VALUE3)	

(11) DIV

	CR	CR
	CR	CR
	(: CR	CR .(
)	
	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL	
	가	가
LD	I_VALUE1	INT I_VALUE1 CR
		CR INT
DIV	I_VALUE2	CR INT I_VALUE2
		CR
ST	I_VALUE3	CR INT I_VALUE3
		I_VALUE3 <== I_VALUE1 / I_VALUE2
LD	D_VALUE1	DINT D_VALUE1 CR
		CR DINT
DIV(D_VALUE2	CR DINT
		D_VALUE2 CR
ADD	D_VALUE3	CR DINT D_VALUE3
		CR
)		CR CR
		CR
ST	D_VALUE4	CR DINT B_VALUE4
		D_VALUE4 <== D_VALUE1 / (D_VALUE2 + D_VALUE3)

(12) GT

	<pre> CR BOOL CR CR CR 1 CR 0 CR CR BOOL </pre>
	<pre> (: CR CR .() </pre>
	<pre> ARRAY 가 가 </pre>
<pre> LD I_VAL1 GT I_VAL2 ST B_VAL1 LD I_VAL2 GT I_VAL1 ST B_VAL2 LD I_VAL1 GT(I_VAL2 SUB I_VAL3) ST B_VAL3 </pre>	<pre>) I_VAL1 = 50, I_VAL2 = 100 I_VAL3 = 70 INT I_VAL1 CR CR INT I_VAL2 CR .(I_VAL1 < I_VAL2 CR 0) CR BOOL B_VAL1 B_VAL1 <== FALSE INT I_VAL2 CR CR INT I_VAL1 CR .(I_VAL1 < I_VAL2 CR 1) CR BOOL B_VAL2 B_VAL2 <== TRUE INT I_VAL1 CR CR INT I_VAL2 CR CR INT I_VAL3 CR CR CR CR .(CR > CR CR 1) CR BOOL B_VAL3 B_VAL3 <== TRUE </pre>

(13) GE

	<pre> CR BOOL CR CR , CR 1 , CR 0 CR CR BOOL </pre>
	<pre> (: CR CR .() </pre>
	<pre> ARRAY 가 가 </pre>
	<pre>) I_VAL1 = 50, I_VAL2 = 100 I_VAL3 = 70 LD I_VAL1 INT I_VAL1 CR GE I_VAL2 CR INT I_VAL2 CR .(I_VAL1 < I_VAL2 CR 0) ST B_VAL1 CR BOOL B_VAL1 B_VAL1 <== FALSE LD I_VAL2 INT I_VAL2 CR GE I_VAL1 CR INT I_VAL1 CR .(I_VAL1 < I_VAL2 CR 1) ST B_VAL2 CR BOOL B_VAL2 B_VAL2 <== TRUE LD I_VAL1 INT I_VAL1 CR GE(I_VAL2 CR INT I_VAL2 CR SUB I_VAL3 CR INT I_VAL3 CR) CR CR CR .(CR > CR CR 1) ST B_VAL3 CR BOOL B_VAL3 B_VAL3 <== TRUE </pre>

(14) EQ

	CR	BOOL	CR	.
	CR	CR	1	CR
	CR			0
			CR	BOOL
	(:	CR	CR
)			.(
	ARRAY		가	.
	가			.
) I_VAL1 = 50, I_VAL2 = 100 I_VAL3 = 50	
LD	I_VAL1	INT	I_VAL1	CR
EQ	I_VAL2	CR	INT	I_VAL2
		CR	.(I_VAL1 < I_VAL2	CR
ST	B_VAL1	CR	BOOL	B_VAL1
			B_VAL1 <== FALSE	.
LD	I_VAL1	INT	I_VAL2	CR
EQ	I_VAL3	CR	INT	I_VAL1
		CR	.(I_VAL1 = I_VAL3	CR
ST	B_VAL2	CR	BOOL	B_VAL2
			B_VAL2 <== TRUE	.
LD	I_VAL1	INT	I_VAL1	CR
EQ(I_VAL2	CR		INT
		CR		I_VAL2
SUB	I_VAL3	CR	INT	I_VAL3
			CR	.
)			CR	CR
		CR	.(CR =
ST	B_VAL3	CR	BOOL	B_VAL3
			B_VAL3 <== TRUE	.
			CR	CR
			CR	1)

(15) NE

	<pre> CR BOOL CR . CR , CR 1 CR 0 . CR CR BOOL </pre>
	<pre> (: CR CR .() </pre>
	<pre> ARRAY 가 가 </pre>
<pre> LD I_VAL1 NE I_VAL3 ST B_VAL1 LD I_VAL1 NE I_VAL2 ST B_VAL2 LD I_VAL1 NE(I_VAL2 SUB I_VAL3) ST B_VA3 </pre>	<pre>) I_VAL1 = 50, I_VAL2 = 100 I_VAL3 = 50 INT I_VAL1 CR . CR INT I_VAL3 CR .(I_VAL1 = I_VAL3 CR 0) CR BOOL B_VAL1 . B_VAL1 <== FALSE INT I_VAL1 CR . CR INT I_VAL2 CR .(I_VAL1 <> I_VAL2 CR 1) CR BOOL B_VAL2 . B_VAL2 <== TRUE INT I_VAL1 CR . CR INT I_VAL2 CR CR INT I_VAL3 CR . CR CR CR .(CR = CR CR 0) CR BOOL B_VAL3 . B_VAL2 <== FALSE </pre>

(16) LE

	<pre> CR BOOL CR CR , CR 1 , CR 0 CR CR BOOL </pre>
	<pre> (: CR CR .() </pre>
	<pre> ARRAY 가 가 </pre>
<pre> LD I_VAL2 LE I_VAL1 ST B_VAL1 LD I_VAL1 LE I_VAL2 ST B_VAL2 LD I_VAL1 LE(I_VAL2 SUB I_VAL3) ST B_VA3 </pre>	<pre>) I_VAL1 = 50, I_VAL2 = 100 I_VAL3 = 70 INT I_VAL2 CR CR INT I_VAL1 CR .(I_VAL1 < I_VAL2 CR 0) CR BOOL B_VAL1 B_VAL1 <== FALSE INT I_VAL1 CR CR INT I_VAL2 CR .(I_VAL1 < I_VAL2 CR 1) CR BOOL B_VAL2 B_VAL2 <== TRUE INT I_VAL1 CR CR INT I_VAL2 CR CR INT I_VAL3 CR CR CR CR .(CR > CR CR 0) CR BOOL B_VAL3 B_VAL2 <== FALSE </pre>

(17) LT

	<pre> CR BOOL CR . CR CR 1 . CR 0 CR CR CR CR BOOL </pre>
	<pre> (: CR CR .() </pre>
	<pre> ARRAY 가 가 </pre>
	<pre>) I_VAL1 = 50, I_VAL2 = 100 I_VAL3 = 70 LD I_VAL2 INT I_VAL2 CR . LT I_VAL1 CR INT I_VAL1 CR CR .(I_VAL1 < I_VAL2 CR 0) ST B_VAL1 CR BOOL B_VAL1 . B_VAL1 <== FALSE LD I_VAL1 INT I_VAL1 CR . LT I_VAL2 CR INT I_VAL2 CR CR .(I_VAL1 < I_VAL2 CR 1) ST B_VAL2 CR BOOL B_VAL2 . B_VAL2 <== TRUE LD I_VAL1 INT I_VAL1 CR . LT(I_VAL2 CR INT I_VAL2 CR SUB I_VAL3 CR INT I_VAL3 CR CR .) CR CR CR CR .(CR > CR CR 0) ST B_VA3 CR BOOL B_VAL3 . B_VAL2 <== FALSE </pre>

(18) JMP

	C :	BOOL	CR	TRUE(1)	
		BOOL	CR	FALSE(0)	
	N :	BOOL	CR	FALSE(0)	
		BOOL	CR	TRUE(1)	
	가		CR		
				BOOL	B_VAL1
				I_VAL2	I_VAL3
	LD	B_VAL1		BOOL	B_VAL1
	JMPC	THERE1	CR	1	THERE1
					, 0
	LD	I_VAL1	CR	<==	I_VAL1
	JMP	THERE2			THERE2
	THERE1:				THERE1
	LD	I_VAL2	CR	<==	I_VAL2
	THERE2:				THERE2
	ST	I_VAL3			I_VAL3 <== CR
				BOOL	B_VAL2
				1	SEL
	LD	B_VAL2	CR	<==	B_VAL2
	JMPN	THERE3	CR	0(FALSE)	THERE3
	LD	B_VALUE	CR	<==	B_VALUE
	SEL				SEL
	G:=	CURRENT			RESULT
	IN1:=	I_VAL1			
	IN2:=	I_VAL2			
	ST	I_VAL3			I_VAL3 <== CR
	THERE3:				THERE3

(19) CAL

	C :	BOOL	CR	TRUE(1)	
		BOOL	CR	FALSE(0)	
	N :	BOOL	CR	FALSE(0)	
		BOOL	CR	TRUE(1)	
	가	CR			
	LD B_VAL1	BOOL	B_VAL1	1(TRUE)	
	CALC TON TIMER1	TON			
	IN:= T_INPUT	BOOL	B_VAL1	CR	
	PT:= PRE_TIME	가	TIMER1		
		CR	1		
	LD B_VAL2	CTU			
	CALN CTU COUNT1	BOOL	B_VAL2	0(FALSE)	
	CU:= B_UP	가	COUNT1		CTU
	R:= B_RESET	CR	1		
	PV:= 100				
		CR			CTD
	CAL CTD COUNT2	가	COUNT2		CTD
	CD:= B_DOWN				
	LD:= B_LDV				
	PV:= 300				

(20) RET

	C : BOOL CR TRUE(1) . BOOL CR FALSE(0) . N : BOOL CR FALSE(0) . BOOL CR TRUE(1) .	
	가	CR
LD I_VAL1 MUL I_VAL2 ST I_VAL3 LD _ERR RETN LD 0 ST I_VAL3 RET	INT I_VAL2 INT I_VAL1 I_VAL3 0 CR <== CR 0 가 I_VAL3 <== 0	I_VAL3 가

(21))

	(.
	.	.
	.	.
LD	I_VAL1	I_VAL4 <== (I_VAL1 + I_VAL2) * I_VAL3
ADD	I_VAL2	
MUL	I_VAL3	
ST	I_VAL4	
LD	I_VAL1	I_VAL4 <== I_VAL1 + (I_VAL2 * I_VAL3)
ADD(I_VAL2	
MUL	I_VAL3	
)		
ST	I_VAL4	
LD	L_VAL1	L_VAL7 <== (L_VAL1 + (L_VAL2 * (L_VAL3 - L_VAL4) +
ADD(L_VAL2	L_VAL5)) / L_VAL6
MUL(L_VAL3	
SUB	L_VAL4	
)		
ADD	L_VAL5	
)		
DIV	L_VAL6	
ST	L_VAL7	

5.4.

```

CR
CR
CR
—
LD VAL
SIN
ST RESULT ( VAL RESULT REAL )
CR VAL SIN , CR SIN
, SIN CR CR RESULT
LD %IX0.0.0
SEL G:= CURRENT RESULT
INO:= VAL1
IN1:= VAL2
ST VAL3
CR , SEL
CR SEL
가 CR CR VAL3
```

JMP(JMPN, JMPC)

```

—
LD  %IX0.0.0
JMPN  THERE
LD  I_VAL1
ADD  IN1:= CURRENT RESULT
      IN2:= I_VAL2
      IN3:= I_VAL3
ST  I_VAL4
THERE:
      CR          BOOL  %IX0.0.0          1 가 0
가      0      THERE:          . %IX0.0.0      1      JMP
      CAL
      CAL  INSTANCE /*          . */
      CALN  INSTANCE /* CR      BOOL 0          . */
      CALC  INSTANCE /* CR      BOOL 1          . */
      INSTANCE
      CR          가
      CR

```

On-Delay Timer

```

—
LD  %IX0.0.0
CALC  TON  TIMER0
      IN:=  %IX0.1.2
      PT:=  T#200S
LD  TIMER0.Q
ST  %QX1.0.2
(  TIMER0  TON          가 )

```

On-Delay Timer

```

TIMER0.Q  TIMER0.ET

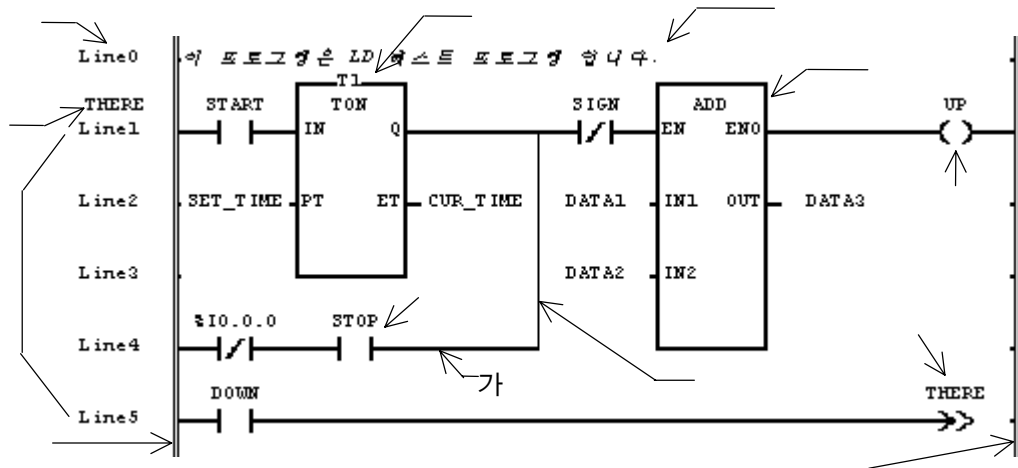
```

6. LD(Ladder Diagram)

6.1.

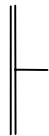

LD

PLC



6.2.

LD

No.		
1		BOOL 1 가
2		

6.3.

가 BOOL 1 ,
LD BOOL 가 (Rung)
LD LD 가

No.		
1	_____	가
2		가

6.4.

가
(Boolean AND)

BOOL
가

No.		
		(Normally Open Contact)
1	*** — —	BOOL ("****") 가 On 가 Off
		(Normally Closed Contact)
2	*** — /—	BOOL ("****") 가 Off 가 Off
		(Positive Transition-Sensing Contact)
3	*** — P —	BOOL ("****") Off On , 가 On On
		(Negative Transition-Sensing Contact)
4	*** — N —	BOOL ("****") On Off 가 On On

6.5.

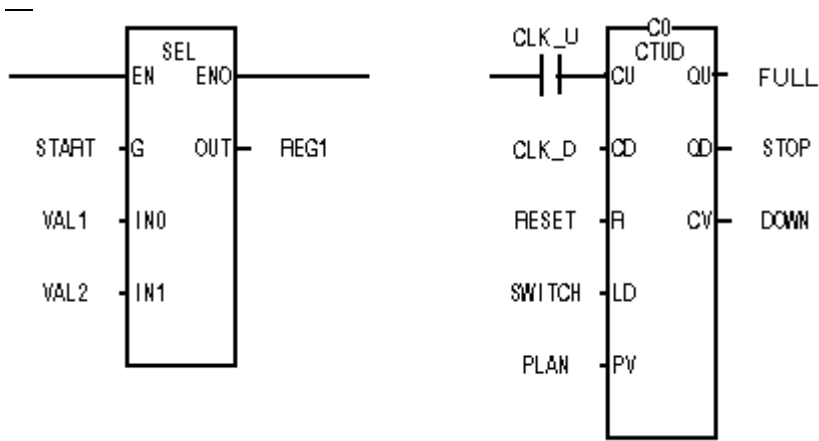
BOOL

(Momentary Coils)		
No.		
1	*** —()—	(Coil) BOOL ("****")
2	*** —(/)—	(Negated Coil) (Negated) BOOL ("****") Off On 가 On Off
(Latched Coils)		
3	*** —(S)—	Set(Latch) Coil 가 On) On Reset Off On BOOL ("****")
4	*** —(R)—	Reset(Unlatch) Coil 가 On) Off Set On Off BOOL ("****")
(Transition-Sensing Coils)		
5	*** —(P)—	(Positive Transition-Sensing Coil) 가 Off On On BOOL ("****")
6	*** —(N)—	(Negative Transition-Sensing Coil) 가 On Off On BOOL ("****")

LD 가

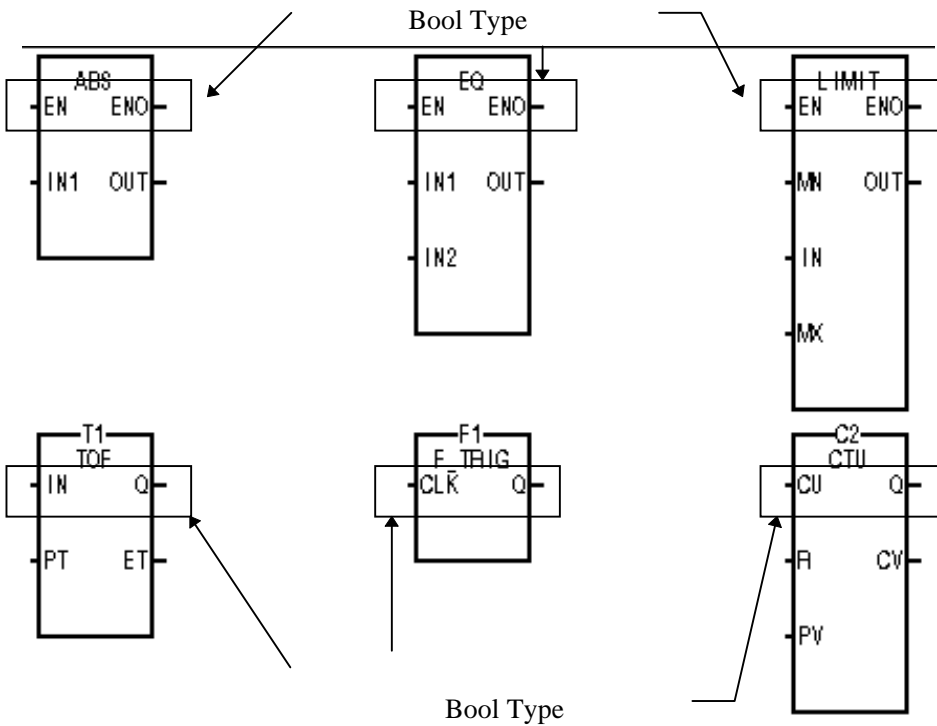
6.6.

가

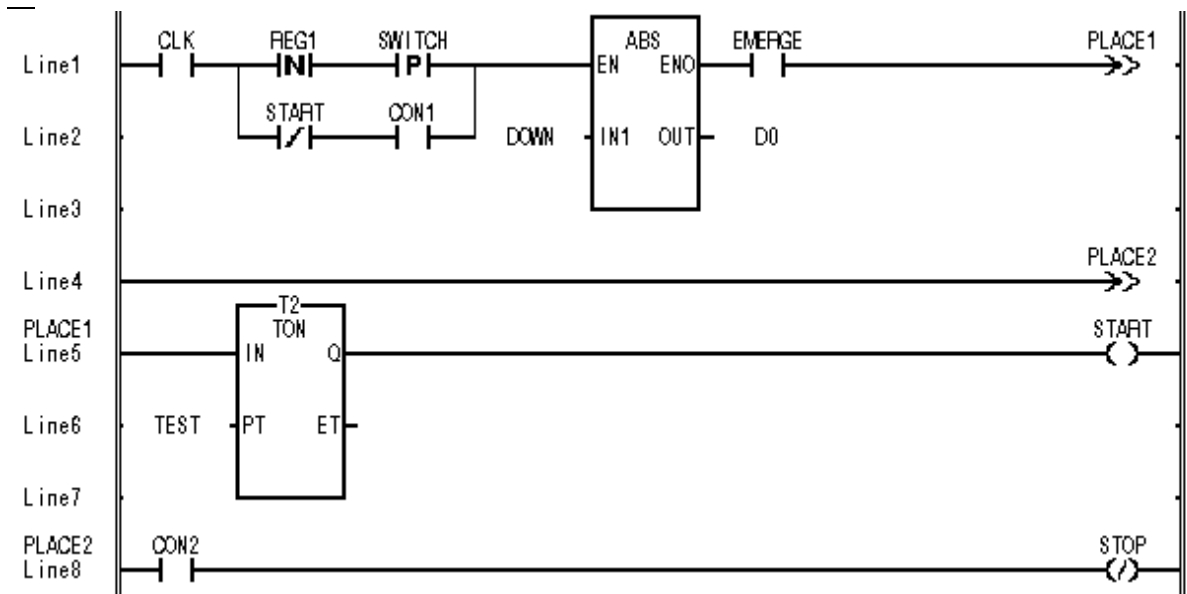
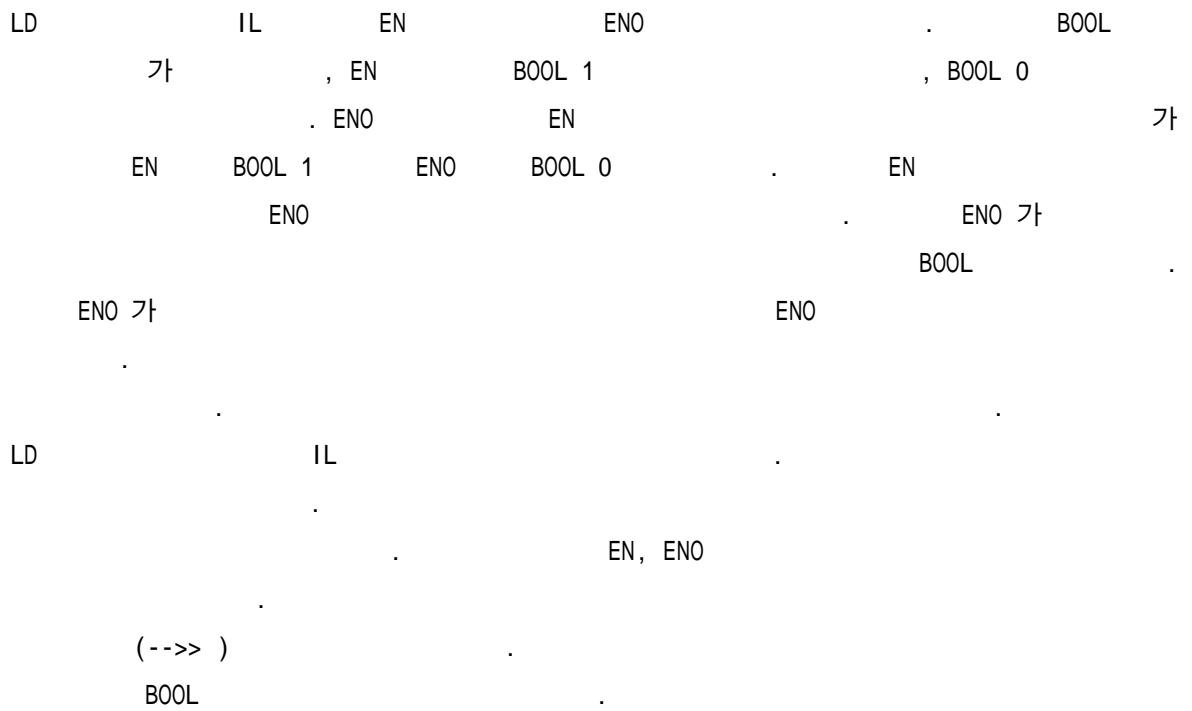


(Power Flow)

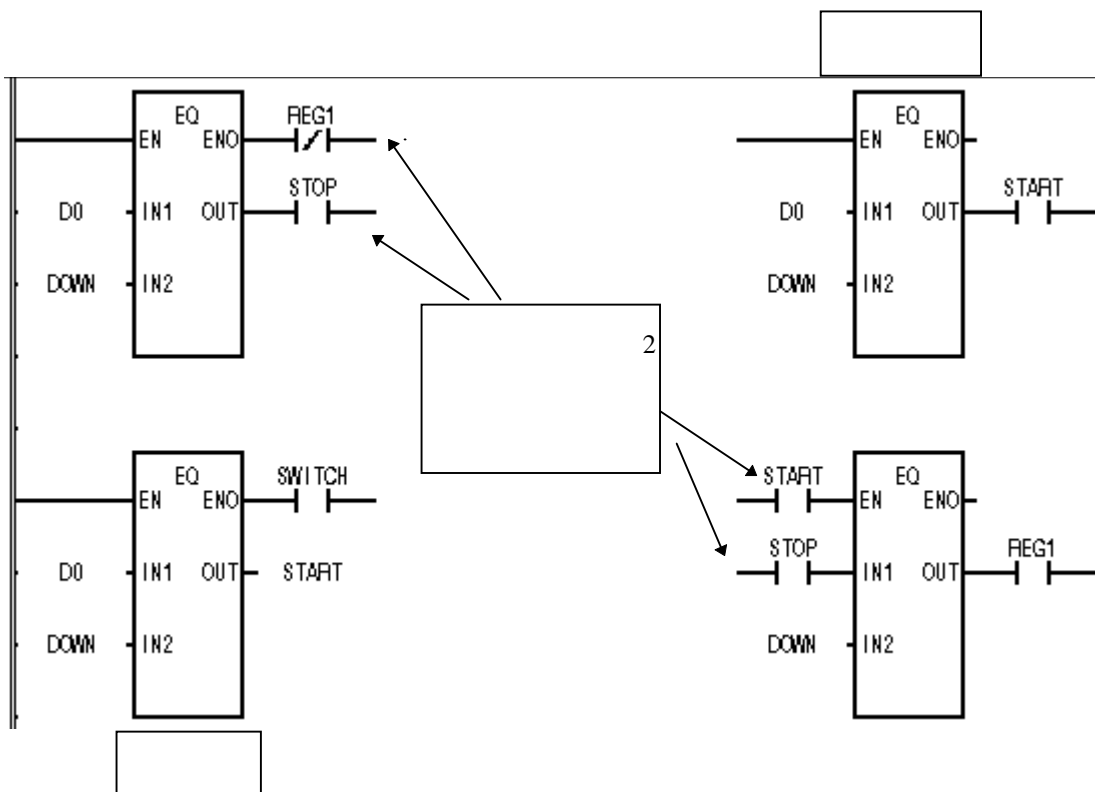
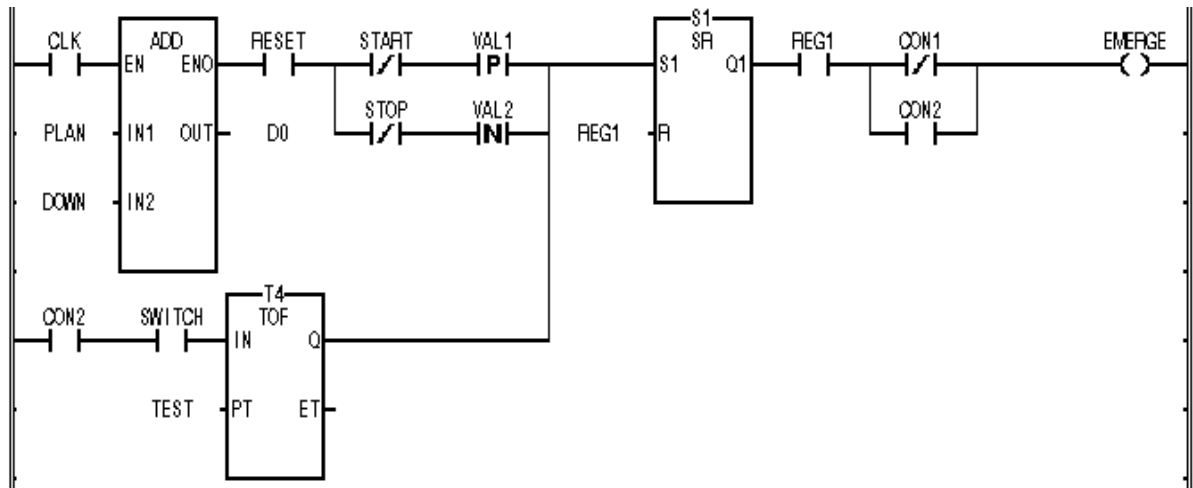
BOOL BOOL
 EN ENO 가 BOOL
 BOOL



6. LD



LD



7.

8.

7.1.

7.1.1.

				GMR 2	GM3	GM4 7
ARY_ASC_TO_***	ARY_ASC_TO_BYTE	WORD(ASCII)	BYTE			
	ARY_ASC_TO_BCD	WORD(ASCII)	BYTE(BCD)			
ARY_BYTE_TO_***	ARY_BYTE_TO_ASC	BYTE	WORD(ASCII)			
ARY_BCD_TO_***	ARY_BCD_TO_ASC	BYTE(BCD)	WORD(ASCII)			
ASC_TO_***	ASC_TO_BCD	BYTE(BCD)	USINT			
	ASC_TO_BYTE	WORD(BCD)	UINT			
BCD_TO_***	BCD_TO_SINT	BYTE(BCD)	SINT			
	BCD_TO_INT	WORD(BCD)	INT			
	BCD_TO_DINT	DWORD(BCD)	DINT			
	BCD_TO_LINT	LWORD(BCD)	LINT			
	BCD_TO_USINT	BYTE(BCD)	USINT			
	BCD_TO_UINT	WORD(BCD)	UINT			
	BCD_TO_UDINT	DWORD(BCD)	UDINT			
	BCD_TO_ULINT	LWORD(BCD)	ULINT			
TRUNC	TRUNC	REAL	DINT			
		LREAL	LINT			
REAL_TO_***	REAL_TO_SINT	REAL	SINT			
	REAL_TO_INT	REAL	INT			
	REAL_TO_DINT	REAL	DINT			
	REAL_TO_LINT	REAL	LINT			
	REAL_TO_USINT	REAL	USINT			
	REAL_TO_UINT	REAL	UINT			
	REAL_TO_UDINT	REAL	UDINT			
	REAL_TO_ULINT	REAL	ULINT			
	REAL_TO_DWORD	REAL	DWORD			
	REAL_TO_LREAL	REAL	LREAL			
LREAL_TO_***	LREAL_TO_SINT	LREAL	SINT			
	LREAL_TO_INT	LREAL	INT			
	LREAL_TO_DINT	LREAL	DINT			
	LREAL_TO_LINT	LREAL	LINT			
	LREAL_TO_USINT	LREAL	USINT			

				GMR 2	GM3	GM4 7
LREAL_TO_***	LREAL_TO_UINT	LREAL	UINT			
	LREAL_TO_UDINT	LREAL	UDINT			
	LREAL_TO_ULINT	LREAL	ULINT			
	LREAL_TO_LWORD	LREAL	LWORD			
	LREAL_TO_REAL	LREAL	REAL			
SINT_TO_***	SINT_TO_INT	SINT	INT			
	SINT_TO_DINT	SINT	DINT			
	SINT_TO_LINT	SINT	LINT			
	SINT_TO_USINT	SINT	USINT			
	SINT_TO_UINT	SINT	UINT			
	SINT_TO_UDINT	SINT	UDINT			
	SINT_TO_ULINT	SINT	ULINT			
	SINT_TO_BOOL	SINT	BOOL			
	SINT_TO_BYTE	SINT	BYTE			
	SINT_TO_WORD	SINT	WORD			
	SINT_TO_DWORD	SINT	DWORD			
	SINT_TO_LWORD	SINT	LWORD			
	SINT_TO_BCD	SINT	BYTE(BCD)			
	SINT_TO_REAL	SINT	REAL			
	SINT_TO_LREAL	SINT	LREAL			
INT_TO_***	INT_TO_SINT	INT	SINT			
	INT_TO_DINT	INT	DINT			
	INT_TO_LINT	INT	LINT			
	INT_TO_USINT	INT	USINT			
	INT_TO_UINT	INT	UINT			
	INT_TO_UDINT	INT	UDINT			
	INT_TO_ULINT	INT	ULINT			
	INT_TO_BOOL	INT	BOOL			
	INT_TO_BYTE	INT	BYTE			
	INT_TO_WORD	INT	WORD			
	INT_TO_DWORD	INT	DWORD			
	INT_TO_LWORD	INT	LWORD			
	INT_TO_BCD	INT	WORD(BCD)			
	INT_TO_REAL	INT	REAL			
	INT_TO_LREAL	INT	LREAL			

				GMR 2	GM3	GM4 7
DINT_TO_***	DINT_TO_SINT	DINT	SINT			
	DINT_TO_INT	DINT	INT			
	DINT_TO_LINT	DINT	LINT			
	DINT_TO_USINT	DINT	USINT			
	DINT_TO_UINT	DINT	UINT			
	DINT_TO_UDINT	DINT	UDINT			
	DINT_TO_ULINT	DINT	ULINT			
	DINT_TO_BOOL	DINT	BOOL			
	DINT_TO_BYTE	DINT	BYTE			
	DINT_TO_WORD	DINT	WORD			
	DINT_TO_DWORD	DINT	DWORD			
	DINT_TO_LWORD	DINT	LWORD			
	DINT_TO_BCD	DINT	DWORD(BCD)			
	DINT_TO_REAL	DINT	REAL			
DINT_TO_LREAL	DINT	LREAL				
LINT_TO_***	LINT_TO_SINT	LINT	SINT			
	LINT_TO_INT	LINT	INT			
	LINT_TO_DINT	LINT	DINT			
	LINT_TO_USINT	LINT	USINT			
	LINT_TO_UINT	LINT	UINT			
	LINT_TO_UDINT	LINT	UDINT			
	LINT_TO_ULINT	LINT	ULINT			
	LINT_TO_BOOL	LINT	BOOL			
	LINT_TO_BYTE	LINT	BYTE			
	LINT_TO_WORD	LINT	WORD			
	LINT_TO_DWORD	LINT	DWORD			
	LINT_TO_LWORD	LINT	LWORD			
	LINT_TO_BCD	LINT	LWORD(BCD)			
	LINT_TO_REAL	LINT	REAL			
LINT_TO_LREAL	LINT	LREAL				
USINT_TO_***	USINT_TO_SINT	USINT	SINT			
	USINT_TO_INT	USINT	INT			
	USINT_TO_DINT	USINT	DINT			
	USINT_TO_LINT	USINT	LINT			
	USINT_TO_UINT	USINT	UINT			
	USINT_TO_UDINT	USINT	UDINT			
	USINT_TO_ULINT	USINT	ULINT			
	USINT_TO_BOOL	USINT	BOOL			
	USINT_TO_BYTE	USINT	BYTE			
	USINT_TO_WORD	USINT	WORD			
	USINT_TO_DWORD	USINT	DWORD			
	USINT_TO_LWORD	USINT	LWORD			

				GMR 2	GM3	GM4 7
USINT_TO_***	USINT_TO_BCD	USINT	BYTE(BCD)			
	USINT_TO_REAL	USINT	REAL			
	USINT_TO_LREAL	USINT	LREAL			
UINT_TO_***	UINT_TO_SINT	UINT	SINT			
	UINT_TO_INT	UINT	INT			
	UINT_TO_DINT	UINT	DINT			
	UINT_TO_LINT	UINT	LINT			
	UINT_TO_USINT	UINT	USINT			
	UINT_TO_UDINT	UINT	UDINT			
	UINT_TO_ULINT	UINT	ULINT			
	UINT_TO_BOOL	UINT	BOOL			
	UINT_TO_BYTE	UINT	BYTE			
	UINT_TO_WORD	UINT	WORD			
	UINT_TO_DWORD	UINT	DWORD			
	UINT_TO_LWORD	UINT	LWORD			
	UINT_TO_BCD	UINT	WORD(BCD)			
	UINT_TO_REAL	UINT	REAL			
	UINT_TO_LREAL	UINT	LREAL			
	UINT_TO_DATE	UINT	DATE			
UDINT_TO_***	UDINT_TO_SINT	UDINT	SINT			
	UDINT_TO_INT	UDINT	INT			
	UDINT_TO_DINT	UDINT	DINT			
	UDINT_TO_LINT	UDINT	LINT			
	UDINT_TO_USINT	UDINT	USINT			
	UDINT_TO_UINT	UDINT	UINT			
	UDINT_TO_ULINT	UDINT	ULINT			
	UDINT_TO_BOOL	UDINT	BOOL			
	UDINT_TO_BYTE	UDINT	BYTE			
	UDINT_TO_WORD	UDINT	WORD			
	UDINT_TO_DWORD	UDINT	DWORD			
	UDINT_TO_LWORD	UDINT	LWORD			
	UDINT_TO_BCD	UDINT	DWORD(BCD)			
	UDINT_TO_REAL	UDINT	REAL			
	UDINT_TO_LREAL	UDINT	LREAL			
	UDINT_TO_TOD	UDINT	TOD			
UDINT_TO_TIME	UDINT	TIME				
ULINT_TO_***	ULINT_TO_SINT	ULINT	SINT			
	ULINT_TO_INT	ULINT	INT			
	ULINT_TO_DINT	ULINT	DINT			
	ULINT_TO_LINT	ULINT	LINT			
	ULINT_TO_USINT	ULINT	USINT			
ULINT_TO_UINT	ULINT	UINT				

				GMR 2	GM3	GM4 7
ULINT_TO_***	ULINT_TO_UDINT	ULINT	UDINT			
	ULINT_TO_BOOL	ULINT	BOOL			
	ULINT_TO_BYTE	ULINT	BYTE			
	ULINT_TO_WORD	ULINT	WORD			
	ULINT_TO_DWORD	ULINT	DWORD			
	ULINT_TO_LWORD	ULINT	LWORD			
	ULINT_TO_BCD	ULINT	LWORD(BCD)			
	ULINT_TO_REAL	ULINT	REAL			
	ULINT_TO_LREAL	ULINT	LREAL			
BOOL_TO_***	BOOL_TO_SINT	BOOL	SINT			
	BOOL_TO_INT	BOOL	INT			
	BOOL_TO_DINT	BOOL	DINT			
	BOOL_TO_LINT	BOOL	LINT			
	BOOL_TO_USINT	BOOL	USINT			
	BOOL_TO_UINT	BOOL	UINT			
	BOOL_TO_UDINT	BOOL	UDINT			
	BOOL_TO_ULINT	BOOL	ULINT			
	BOOL_TO_BYTE	BOOL	BYTE			
	BOOL_TO_WORD	BOOL	WORD			
	BOOL_TO_DWORD	BOOL	DWORD			
	BOOL_TO_LWORD	BOOL	LWORD			
	BOOL_TO_STRING	BOOL	STRING			
BYTE_TO_***	BYTE_TO_SINT	BYTE	SINT			
	BYTE_TO_INT	BYTE	INT			
	BYTE_TO_DINT	BYTE	DINT			
	BYTE_TO_LINT	BYTE	LINT			
	BYTE_TO_USINT	BYTE	USINT			
	BYTE_TO_UINT	BYTE	UINT			
	BYTE_TO_UDINT	BYTE	UDINT			
	BYTE_TO_ULINT	BYTE	ULINT			
	BYTE_TO_BOOL	BYTE	BOOL			
	BYTE_TO_WORD	BYTE	WORD			
	BYTE_TO_DWORD	BYTE	DWORD			
	BYTE_TO_LWORD	BYTE	LWORD			
	BYTE_TO_STRING	BYTE	STRING			
	BYTE_TO_ASC	BYTE	WORD(ASCII)			
WORD_TO_***	WORD_TO_SINT	WORD	SINT			
	WORD_TO_INT	WORD	INT			
	WORD_TO_DINT	WORD	DINT			
	WORD_TO_LINT	WORD	LINT			
	WORD_TO_USINT	WORD	USINT			
	WORD_TO_UINT	WORD	UINT			

				GMR 2	GM3	GM4 7
WORD_TO_***	WORD_TO_UDINT	WORD	UDINT			
	WORD_TO_ULINT	WORD	ULINT			
	WORD_TO_BOOL	WORD	BOOL			
	WORD_TO_BYTE	WORD	BYTE			
	WORD_TO_DWORD	WORD	DWORD			
	WORD_TO_LWORD	WORD	LWORD			
	WORD_TO_DATE	WORD	DATE			
	WORD_TO_STRING	WORD	STRING			
DWORD_TO_***	DWORD_TO_SINT	DWORD	SINT			
	DWORD_TO_INT	DWORD	INT			
	DWORD_TO_DINT	DWORD	DINT			
	DWORD_TO_LINT	DWORD	LINT			
	DWORD_TO_USINT	DWORD	USINT			
	DWORD_TO_UINT	DWORD	UINT			
	DWORD_TO_UDINT	DWORD	UDINT			
	DWORD_TO_ULINT	DWORD	ULINT			
	DWORD_TO_BOOL	DWORD	BOOL			
	DWORD_TO_BYTE	DWORD	BYTE			
	DWORD_TO_WORD	DWORD	WORD			
	DWORD_TO_LWORD	DWORD	LWORD			
	DWORD_TO_REAL	DWORD	REAL			
	DWORD_TO_TIME	DWORD	TIME			
	DWORD_TO_TOD	DWORD	TOD			
	DWORD_TO_STRING	DWORD	STRING			
LWORD_TO_***	LWORD_TO_SINT	LWORD	SINT			
	LWORD_TO_INT	LWORD	INT			
	LWORD_TO_DINT	LWORD	DINT			
	LWORD_TO_LINT	LWORD	LINT			
	LWORD_TO_USINT	LWORD	USINT			
	LWORD_TO_UINT	LWORD	UINT			
	LWORD_TO_UDINT	LWORD	UDINT			
	LWORD_TO_ULINT	LWORD	ULINT			
LWORD_TO_***	LWORD_TO_BOOL	LWORD	BOOL			
	LWORD_TO_BYTE	LWORD	BYTE			
	LWORD_TO_WORD	LWORD	WORD			
	LWORD_TO_DWORD	LWORD	DWORD			
	LWORD_TO_LREAL	LWORD	LREAL			
	LWORD_TO_DT	LWORD	DT			
	LWORD_TO_STRING	LWORD	STRING			
STRING_TO_***	STRING_TO_SINT	STRING	SINT			
	STRING_TO_INT	STRING	INT			
	STRING_TO_DINT	STRING	DINT			

				GMR 2	GM3	GM4 7
STRING_TO_***	STRING_TO_LINT	STRING	LINT			
	STRING_TO_USINT	STRING	USINT			
	STRING_TO_UINT	STRING	UINT			
	STRING_TO_UDINT	STRING	UDINT			
	STRING_TO_ULINT	STRING	ULINT			
	STRING_TO_BOOL	STRING	BOOL			
	STRING_TO_BYTE	STRING	BYTE			
	STRING_TO_WORD	STRING	WORD			
	STRING_TO_DWORD	STRING	DWORD			
	STRING_TO_LWORD	STRING	LWORD			
	STRING_TO_REAL	STRING	REAL			
	STRING_TO_LREAL	STRING	LREAL			
	STRING_TO_DT	STRING	DT			
	STRING_TO_DATE	STRING	DATE			
	STRING_TO_TOD	STRING	TOD			
STRING_TO_TIME	STRING	TIME				
NUM_TO_STRING	NUM_TO_STRING	ANY_NUM	STRING			
TIME_TO_***	TIME_TO_UDINT	TIME	UDINT			
	TIME_TO_DWORD	TIME	DWORD			
	TIME_TO_STRING	TIME	STRING			
DATE_TO_***	DATE_TO_UINT	DATE	UINT			
	DATE_TO_WORD	DATE	WORD			
	DATE_TO_STRING	DATE	STRING			
TOD_TO_***	TOD_TO_UDINT	TOD	UDINT			
	TOD_TO_DWORD	TOD	DWORD			
	TOD_TO_STRING	TOD	STRING			
DT_TO_***	DT_TO_LWORD	DT	LWORD			
	DT_TO_DATE	DT	DATE			
	DT_TO_TOD	DT	TOD			
	DT_TO_STRING	DT	STRING			

7.

7.1.2.

7.1.2.1.

GMR, GM1, GM2 . (ABS GM3, GM4, GM6, GM7 .)

No.		
1	ABS	(Absolute Value)
2	SQRT	(Square Root)
3	LN	(Natural Logarithm)
4	LOG	(Logarithm Base To 10)
5	EXP	(Natural Exponential)
6	SIN	(Sine)
7	COS	(Cosine)
8	TAN	(Tangent)
9	ASIN	(Arc Sine)
10	ACOS	(Arc Cosine)
11	ATAN	(Arc Tangent)
12	RAD_REAL	(°) (Radian)
13	RAD_LREAL	
14	DEG_REAL	(Radian) (°)
15	DEG_LREAL	

7.1.2.2.

EXPT GMR,GM1, GM2 .. (XCHG_*** GM3,GM4,GM6,GM7 .)

No.		
		(,n 8 가)
1	ADD	(OUT <= IN1 + IN2 + ... + INn)
2	MUL	OUT <= IN1 * IN2 * ... * INn)
		가
3	SUB	(OUT <= IN1 - IN2)
4	DIV	(OUT <= IN1 / IN2)
5	MOD	(OUT <= IN1 Modulo IN2)
6	EXPT	(OUT <= IN1 ^{IN2})
7	MOVE	(OUT <= IN)
8	XCHG_***	

7.1.3.

7.1.3.1.

No.			
1	SHL	N	(0)
2	SHR	N	(0)
3	SHIFT_C_***	N	(Carry)
4	ROL	N	
5	ROR	N	
6	ROTATE_C_***	N	(Carry)

7.1.3.2.

No.			(, n 8 가)
1	AND		(OUT <= IN1 AND IN2 AND ... AND INn)
2	OR		(OUT <= IN1 OR IN2 OR ... OR INn)
3	XOR		(OUT <= IN1 XOR IN2 XOR ... XOR INn)
4	NOT		(OUT <= NOT IN1)

7.1.4.

No.			(, n 8 가)
1	SEL	IN0	IN1
2	MAX	IN1, ...	INn
3	MIN	IN1, ...	INn
4	LIMIT		
5	MUX	IN0, ...	INn K

7.1.5.

No.					
1	SWAP_BYTE	BYTE	.	Nibble	
	SWAP_WORD	WORD	.	BYTE	
	SWAP_DWORD	DWORD	.	WORD	
	SWAP_LWORD	LWORD	.	DWORD	
2	ARY_SWAP_BYTE	Array	BYTE	.	Nibble
	ARY_SWAP_WORD	Array	WORD	.	BYTE
	ARY_SWAP_DWORD	Array	DWORD	.	WORD
	ARY_SWAP_LWORD	Array	LWORD	.	DWORD

7.

7.1.6.

No.		(, n 8 가)
1	GT	OUT <= (IN1>IN2) & (IN2>IN3) & ... & (INn-1 > INn)
2	GE	OUT <= (IN1>=IN2) & (IN2>=IN3) & ... & (INn-1 >= INn)
3	EQ	OUT <= (IN1=IN2) & (IN2=IN3) & ... & (INn-1 = INn)
4	LE	OUT <= (IN1<=IN2) & (IN2<=IN3) & ... & (INn-1 <= INn)
5	LT	OUT <= (IN1<IN2) & (IN2<IN3) & ... & (INn-1 < INn)
6	NE	OUT <= (IN1<>IN2) & (IN2<>IN3) & ... & (INn-1 <> INn)

7.1.7.

No.		
1	LEN	
2	LEFT	L
3	RIGHT	L
4	MID	P L
5	CONCAT	
6	INSERT	P
7	DELETE	P L
8	REPLACE	P L
9	FIND	

7.1.8.

No.		
1	ADD_TIME	,
2	SUB_TIME	,
	SUB_DATE	
	SUB_TOD	
	SUB_DT	
3	MUL_TIME	
4	DIV_TIME	
5	CONCAT_TIME	

7.1.9.

No.		
1	DI	
2	EI	가
3	STOP	
4	ESTOP	
5	DIREC_IN	(GM1- GM7)
6	DIREC_O	(GM1- GM7)
7	WDT_RST	Watch_Dog
8	MCS	Master Control
9	MCSCLR	Master Control Clear

7.1.10.

No.		
1	MEQ_***	Masking
2	DIS_***	Bit
3	UNI_***	Array Bit
4	BIT_BYTE	8 Bit BYTE
5	BYTE_BIT	BYTE 8 Bit
6	BYTE_WORD	2 BYTE WORD
7	WORD_BYTE	WORD 2 BYTE
8	WORD_DWORD	2 WORD DWORD
9	DWORD_WORD	DWORD 2 WORD
10	DWORD_LWORD	2 DWORD LWORD
11	LWORD_DWORD	LWORD 2 DWORD
12	GET_CHAR	(Character)
13	PUT_CHAR	
14	STRING_TO_ARY	BYTE Array
15	ARY_TO_STRING	BYTE Array

7.1.11.

No.		
1	FIFO_***	
2	LIFO_***	

7.2. MK(MASTER-K)

No.		(, n 8 가)
1	ENCO_***	ON
2	DECO_***	ON
3	BSUM_***	ON
4	SEG	BCD HEX 7
5	BMOV_***	,
6	INC_***	IN 가
7	DEC_***	IN

7.3. Array

No.		
1	ARY_MOVE	Array Type (OUT <= IN)
2	ARY_CMP_***	2 Array
3	ARY_SCH_***	Array
4	ARY_FLL_***	Array .
5	ARY_AVE_***	Array
6	ARY_SFT_C_***	Array Bit
7	ARY_ROT_C_***	Array Bit
8	SHIFT_A_***	Array
9	ROTATE_A_***	Array

7.4.

7.4.1.

No.		
1	SR	
2	RS	
3	SEMA	Semaphore

7.4.2.

No.		
1	R_TRIG	(Rising Edge Detector)
2	F_TRIG	(Falling Edge Detector)

7.

7.4.3.

No.		
1	CTU	가 (Up Counter)
2	CTD	(Down Counter)
3	CTUD	가 (Up Down Counter)
4	CTR	(Ring Counter)

7.4.4.

No.		
1	TP	(Pulse Timer)
2	TON	On (On-Delay Timer)
3	TOF	Off (Off-Delay Timer)
4	TMR	(Integrating Timer)
5	TP_RST	Off가 가
6	TRTG	(Retriggerable Timer)
7	TOF_RST	Off가 가 Off (Off-Delay Timer)
8	TON_UNIT	On (On-Delay Timer)
9	TOF_UNIT	Off (Off-Delay Timer)
10	TP_UNIT	(Pulse Timer)
11	TMR_UNIT	(Integrating Timer)

7.4.5.

No.		
1	SCON	
2	DUTY	Scan On/Off

8. /

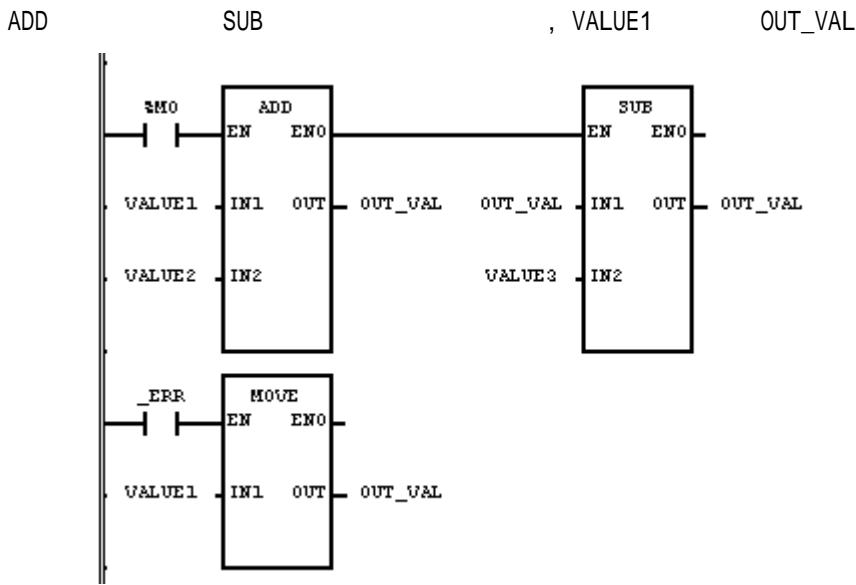
8.1



가 ENO가 0 , (_ERR, _LER) 1
 가 ENO EN .EN,ENO LD

_ERR (Error)
 - 가 _ERR
 (가 _ERR)
 - , 1
 - 가 , 0

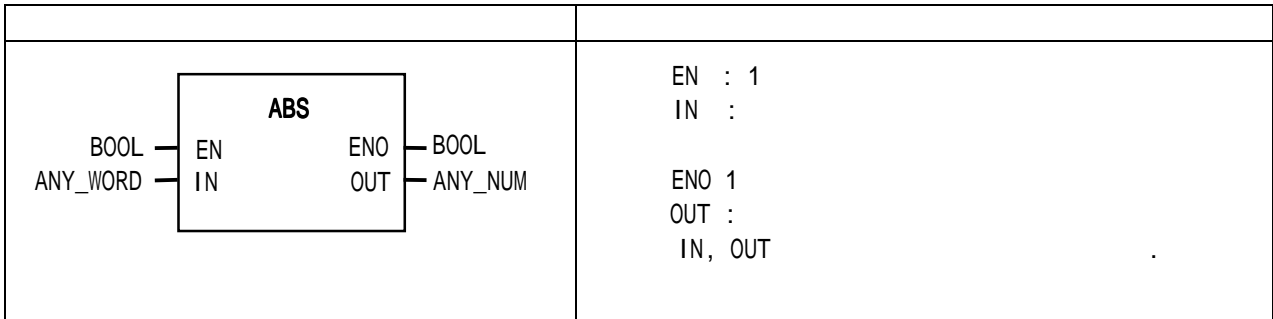
_LER (Latched Error)
 - 1 , 가
 - 0 가



- (1) (ADD) , (ADD) 가
 (IN1) : VALUE1(SINT) = 100(16#64)
 (IN2) : VALUE2(SINT) = 50(16#32)
 (OUT) : OUT_VAL(SINT) = -106(16#96)
- (2) , (ADD) ENO 0 (SUB) , OUT_VAL(SINT) , _ERR , _LER
- On
 (3)_ERR On , (MOVE)
 (IN1) : VALUE1(SINT) = 100(16#64)
 (OUT) : OUT_VAL(SINT) = 100(16#64)

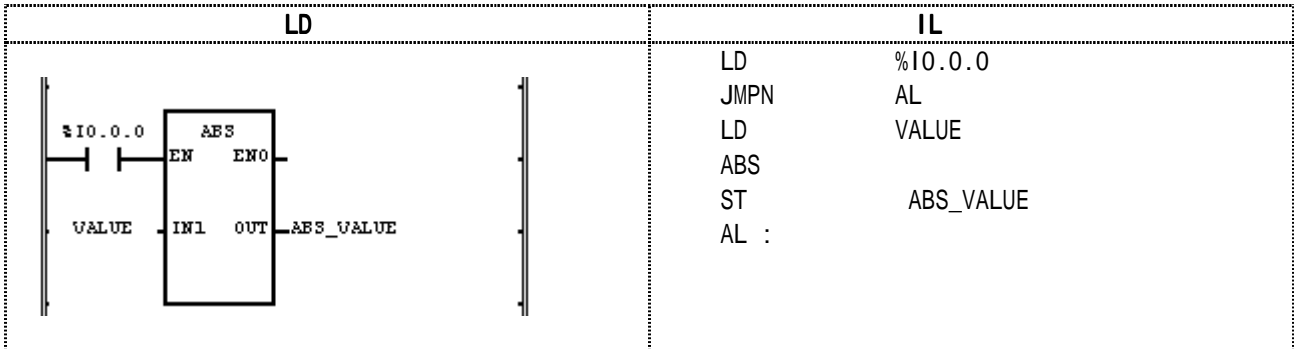
ABS

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN
X X
 X 0 X = X
 X < 0 X = -X
OUT = |IN|

IN (-) _ERR, _LER 가 (Set)
) SINT IN -128



(1) (% I0.0.0) On ABS가
(2) VALUE = -7 , ABS_VALUE = |-7| = 7
 VALUE = 200 , ABS_VALUE = |200| = 200

(IN): VALUE(INT) = -7

1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 (16#FFF9)

↓ (ABS)

(OUT): ABS_VALUE(INT) = 7

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

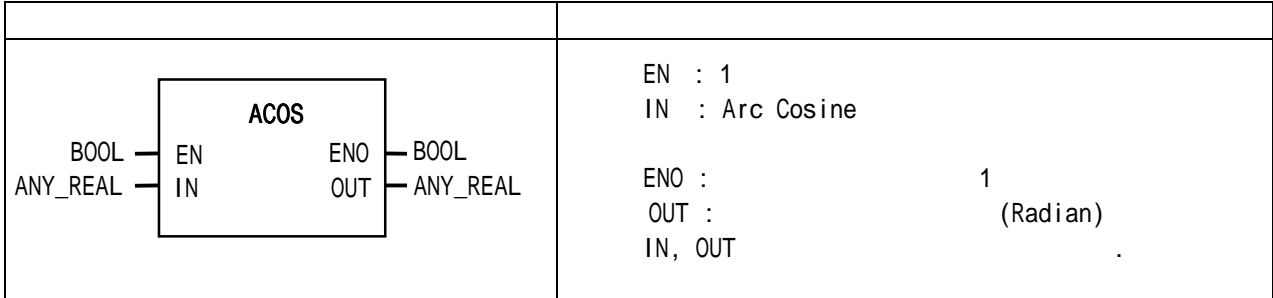
 (16#0007)

INT 2 Complement (3.2.4.)

ACOS

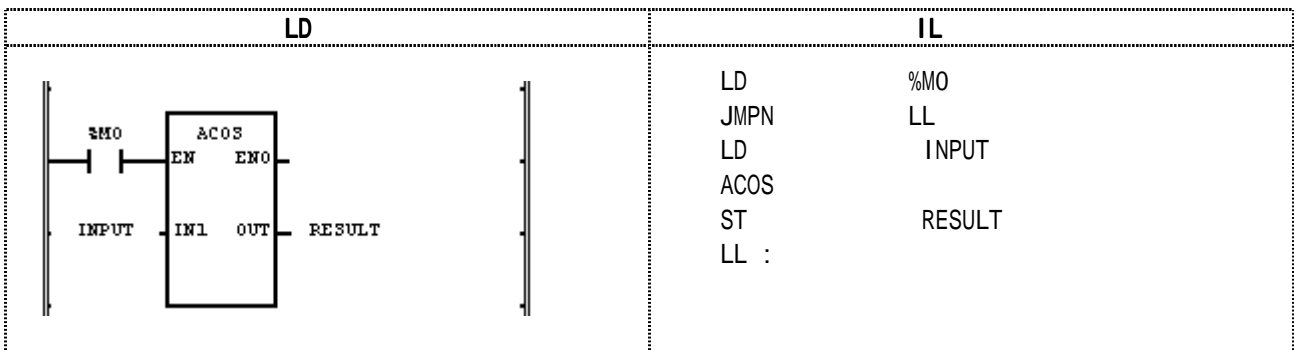
Arc Cosine

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN Arc Cosine OUT . 0 π .
 OUT = ACOS (IN)

IN1 가 -1.0 1.0 _ERR, _LER 가 (Set) .



(1) (%MO) On Arc Cosine ACOS가 .
 (2) INPUT 가 0.8660 ... (√3 / 2) RESULT 0.5235 ... (π/6 rad = 30°) .
 ACOS(√3 / 2) = π/6
 (COS π/6 = √3 / 2)

(IN1) : INPUT (REAL) = 0.866
 ↓ (ACOS)

(OUT) : RESULT (REAL) = 5.23499966E-01

REAL IEEE Standard 754-1984 (3.2.4.))

ADD



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<p>EN : 1 IN1 : IN2 : 8 가</p> <p>ENO : 1 OUT : IN1, IN2, ..., OUT</p>
--	--

IN1, IN2, ..., INn (n) OUT
 OUT = IN1 + IN2 + ... + INn

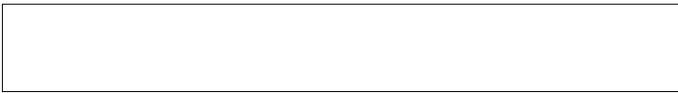
_ERR, _LER 가 (Set)(Set)

<p>LD</p>	<p>IL</p> <pre>LD %M0 JMPN CA LD VALUE1 ADD IN1:= CURRENT RESULT IN2:= VALUE2 IN3:= VALUE3 ST OUT_VAL CA :</pre>
------------------	---

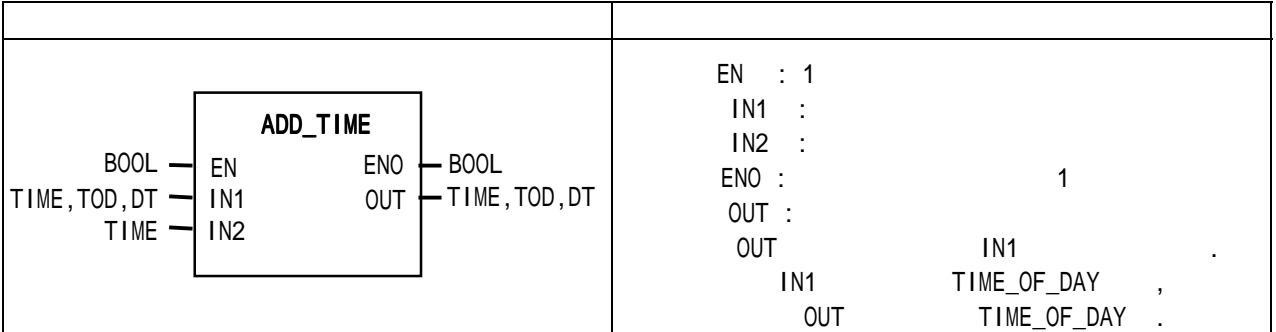
- (1) (%M0) On ADD가
- (2) VALUE1 = 300, VALUE2 = 200, VALUE3 = 100 ,
 OUT_VAL = 300 + 200 + 100 = 600

(IN1): VALUE1(INT) = 300(16#012C)	0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0
	+ (ADD)
(IN2): VALUE2(INT) = 200(16#00C8)	0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
	+ (ADD)
(IN2): VALUE3(INT) = 100(16#0064)	0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
	↓
(OUT): OUT_VAL(INT) = 600(16#0258)	0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0

ADD_TIME

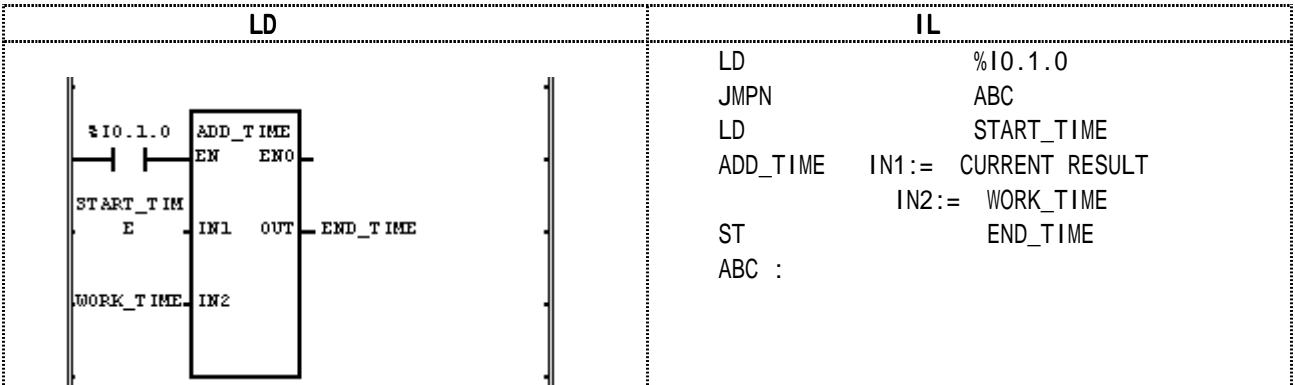


CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1 TIME
 IN1 TIME_OF_DAY
 IN1 DATE_AND_TIME

가 TIME , _ERR, _LER 가 (Set)
 T#49D17H2M47S295MS (TOD)
 (DT) 가 2083
 가 24 ,
 가



- (1) (%I0.1.0) On ADD_TIME
- (2) START_TIME TOD#08:30:00 WORK_TIME
 T#2H10M20S500MS END_TIME TOD#10:40:20.5가

(IN1) : START_TIME(TOD) = TOD#08:30:00
 + (ADD_TIME)

(IN2) : WORK_TIME(TIME) = T#2H10M20S500MS



(OUT) : END_TIME(TOD) = TOD#10:40:20.5

AND

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<p>EN : 1 IN1 : AND IN2 : AND 8 가</p> <p>ENO : EN OUT : AND</p> <p>IN1, IN2, OUT</p>
--	--

```

IN1  IN2      AND   OUT
IN1  1111  ..... 0000
      &
IN2  1010  ..... 1010
OUT  1010  ..... 0000
    
```

<p>LD</p>	<p>IL</p> <pre> LD %IO.1.1 JMPN AA LD %MB10 AND IN1:= CURRENT RESULT IN2:= ABC ST %QB0.0.0 AA : </pre>
------------------	--

(1) (% IO.1.1) 0n AND가

(2) IN1= %MBRQ IN2 = ABC AND 가 OUT = %QB0.0.0

(IN1) : %MB10 (BYTE) = 16#CC

1	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---

& (AND)

(IN2) : ABC (BYTE) = 16#F0

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---



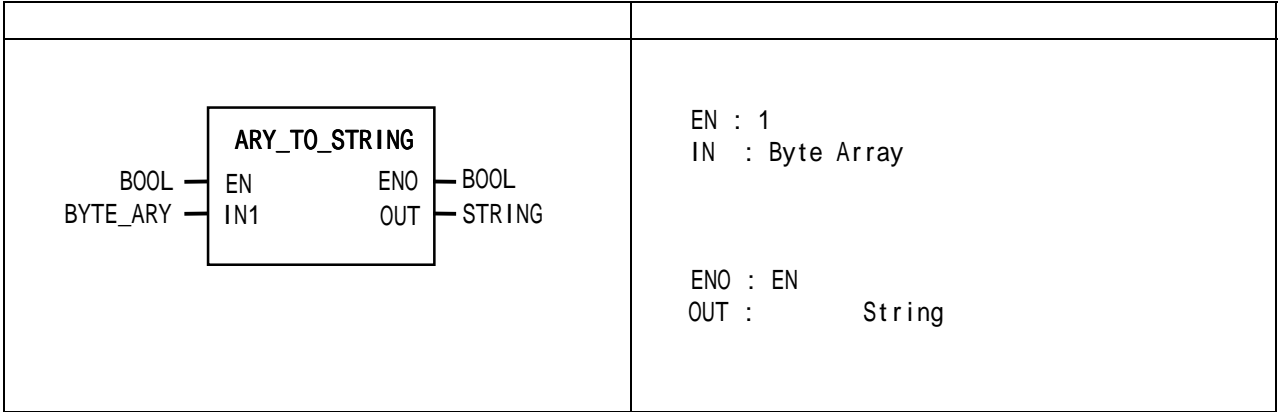
(OUT) : %QB0.0.0 (BYTE) = 16#C0

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

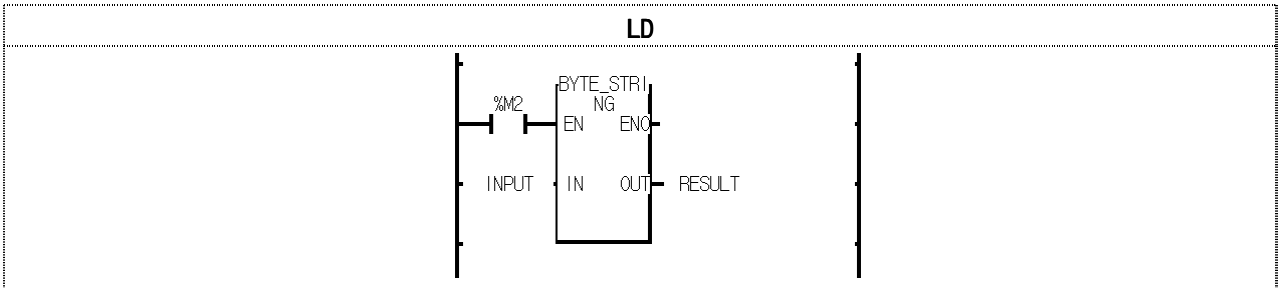
ARY_TO_STRING

Byte Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



Byte Array String .

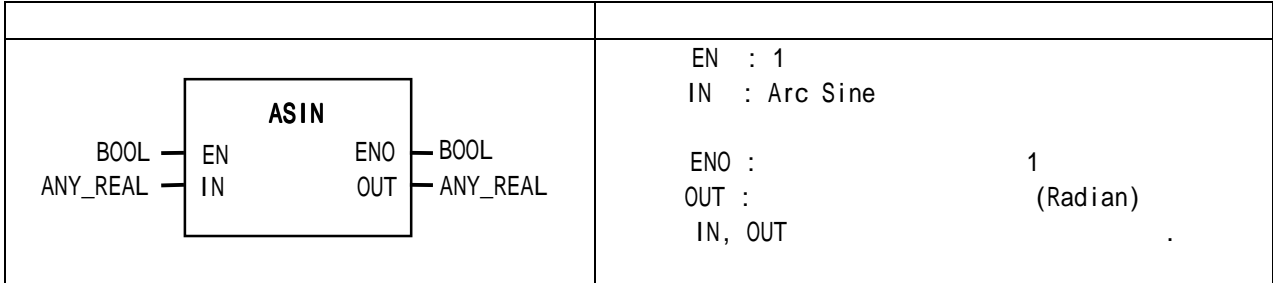


- (1) (%M2) On BYTE_STRING
- (2) Input
16#{22("), 47(G), 4D(M), 34(4), 2D(-), 43(C), 50(P), 55(U), 41(A), 22(")}
"GM4-CPUA"가 RESULT

ASIN

Arc Sine

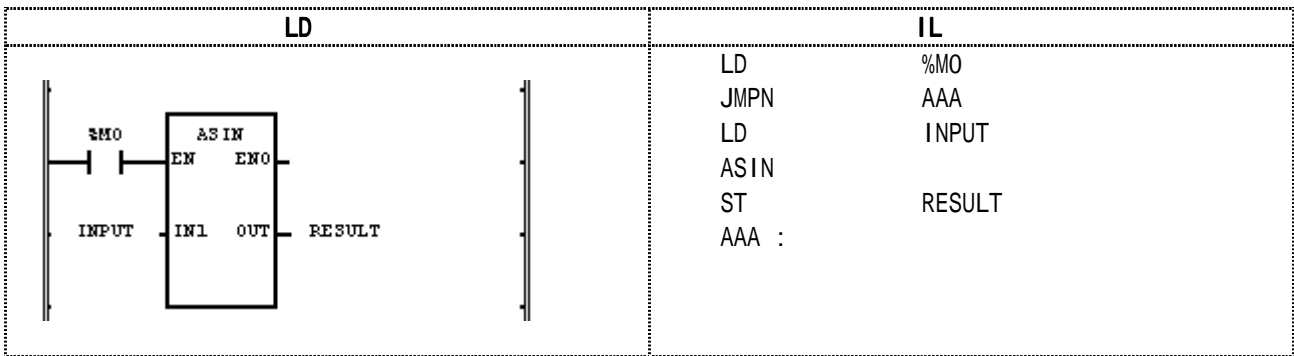
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN Arc Sine OUT $-\pi/2$ $\pi/2$

OUT = ASIN (IN)

가 -1.0 1.0 _ERR, _LER 가 (Set)



(1) (%MO) On Arc Sine ASIN가

(2) INPUT 가 0.8660 ($\sqrt{3} / 2$) RESULT 1.0471

($\pi/3$ rad = 60°)

$$ASIN (\sqrt{3} / 2) = \pi/3$$

$$(SIN(\pi/3) = \sqrt{3} / 2)$$

(IN1) : INPUT(REAL) = 0.866

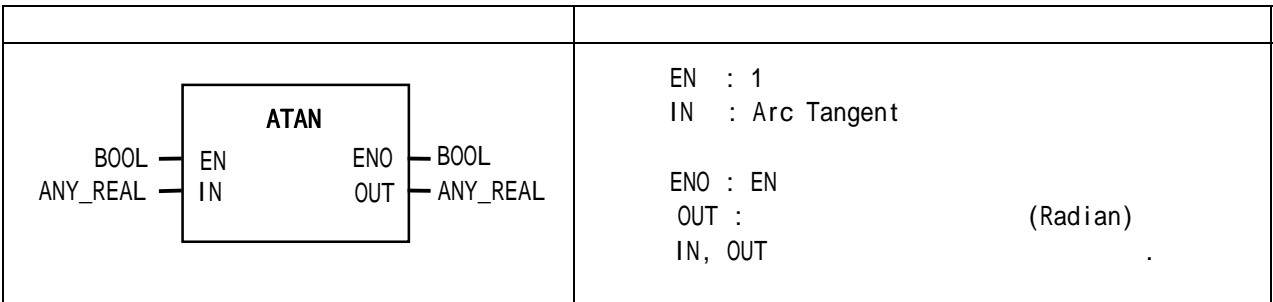
↓ (ASIN)

(OUT) : RESULT(REAL) =1.04714680E+00

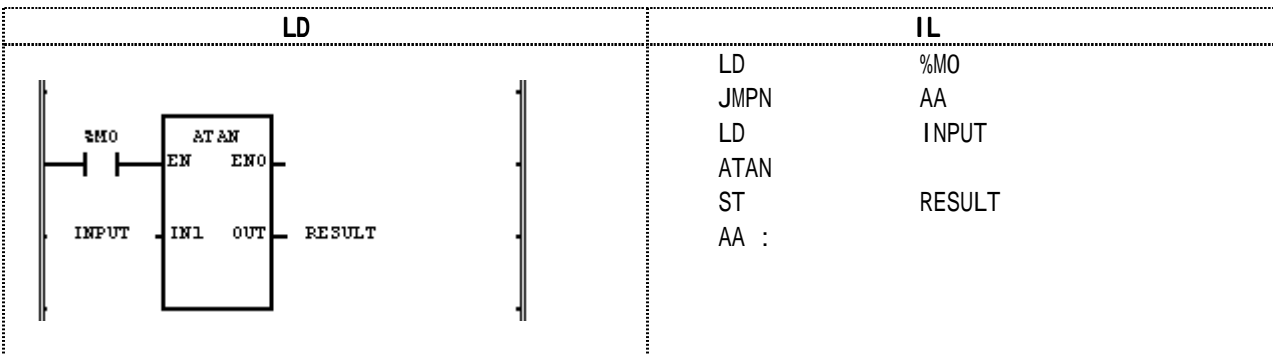
ATAN

Arc Tangent

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN Arc Tangent OUT $-\pi/2$ $\pi/2$
 OUT = ATAN (IN)



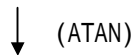
(1) (%MO) On Arc Trangent ATAN
 (2) INPUT = 1.0 ,

RESULT = $\pi/4$ = 0.7853

ATAN (1) = $\pi/4$

(TAN($\pi/4$) = 1)

(IN1) : INPUT(REAL) = 1.0

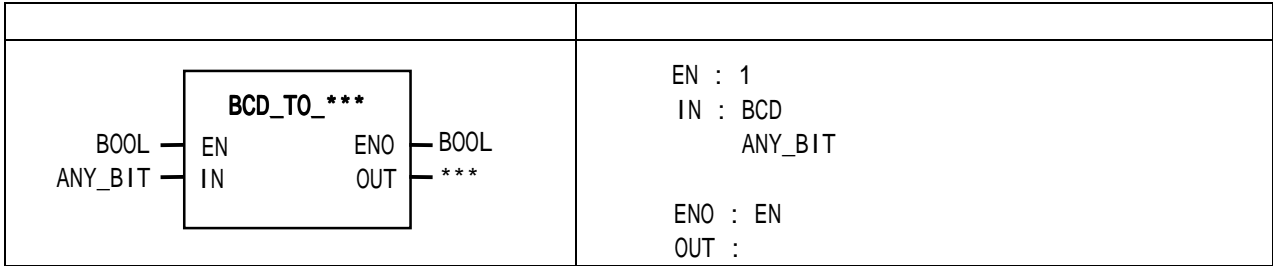


(OUT) : RESULT(REAL) = 7.85398185E

BCD_TO_***

BCD

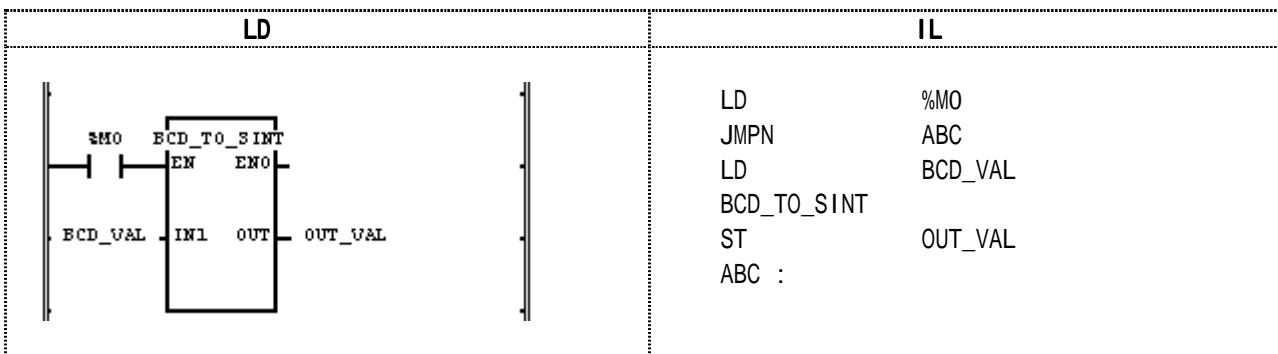
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



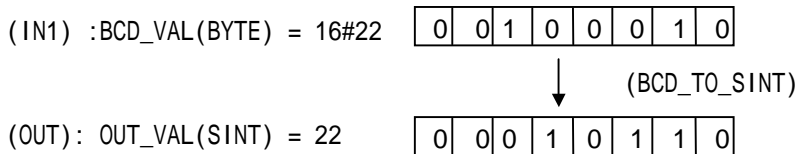
IN OUT

BCD_TO_SINT	BYTE	SINT	BCD (BCD WORD 0 16#9999 .)
BCD_TO_INT	WORD	INT	
BCD_TO_DINT	DWORD	DINT	
BCD_TO_LINT	LWORD	LINT	
BCD_TO_USINT	BYTE	USINT	
BCD_TO_UINT	WORD	UINT	
BCD_TO_UDINT	DWORD	UDINT	
BCD_TO_ULINT	LWORD	ULINT	

IN BCD 가 , 0 _ERR, _LER 가 (Set)



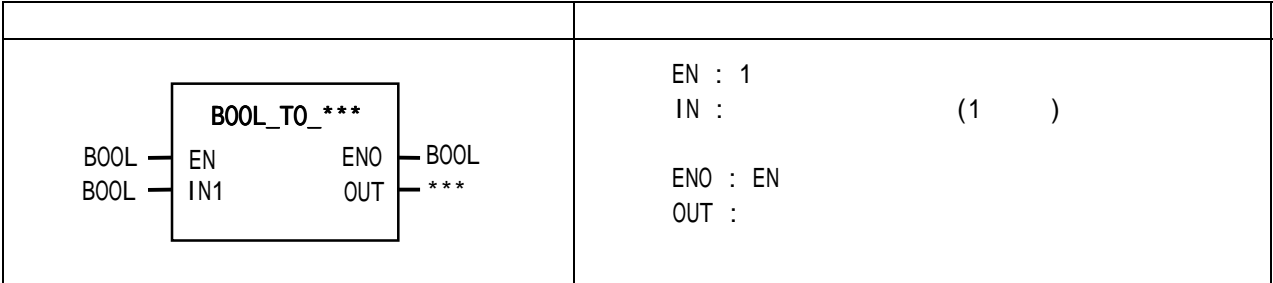
(1) (%MO) On BCD_TO_***
 (2) BCD_VAL(BYTE) = 16#22(2#0010_0010) , OUT_VAL(SINT) = 22(2#0001_0110)가



BOOL_TO_***

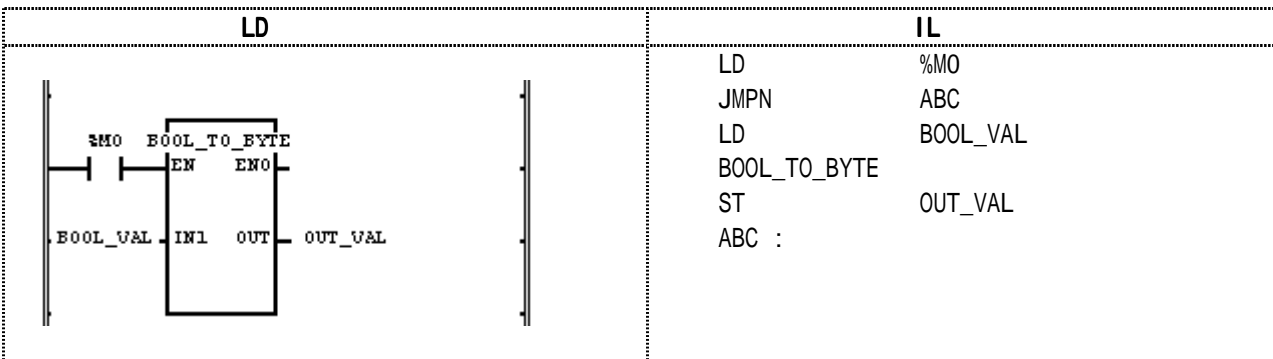
BOOL

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT

BOOL_TO_SINT	SINT	BOOL 2#0 '0' , 2#1 '1'
BOOL_TO_INT	INT	
BOOL_TO_DINT	DINT	
BOOL_TO_LINT	LINT	
BOOL_TO_USINT	USINT	
BOOL_TO_UINT	UINT	
BOOL_TO_UDINT	UDINT	
BOOL_TO_ULINT	ULINT	
BOOL_TO_BYTE	BYTE	
BOOL_TO_WORD	WORD	
BOOL_TO_DWORD	DWORD	
BOOL_TO_LWORD	LWORD	
BOOL_TO_STRING	STRING	BOOL STRING '0' '1'



(1) (%MO) On BOOL_TO_***

(2) BOOL_VAL(BOOL) = 2#1 , OUT_VAL(BYTE) = 2#0000_0001

 (IN1) : BOOL_VAL(BOOL) = 2#1

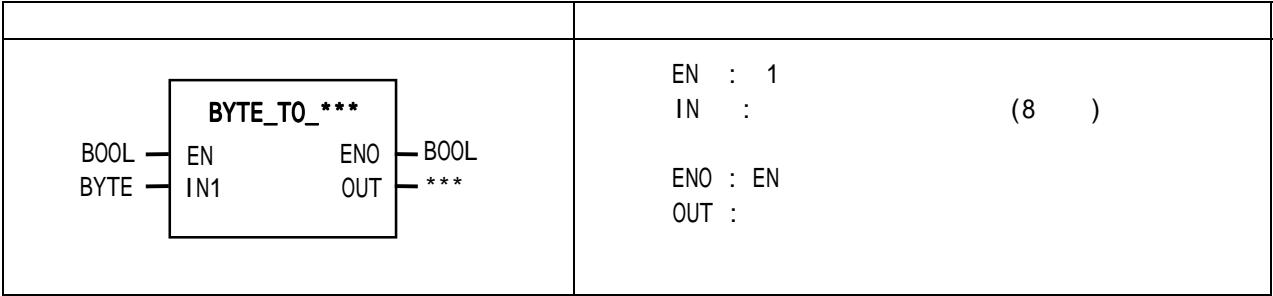
 (OUT) : OUT_VAL(BYTE) = 16#1

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

BYTE_TO_***

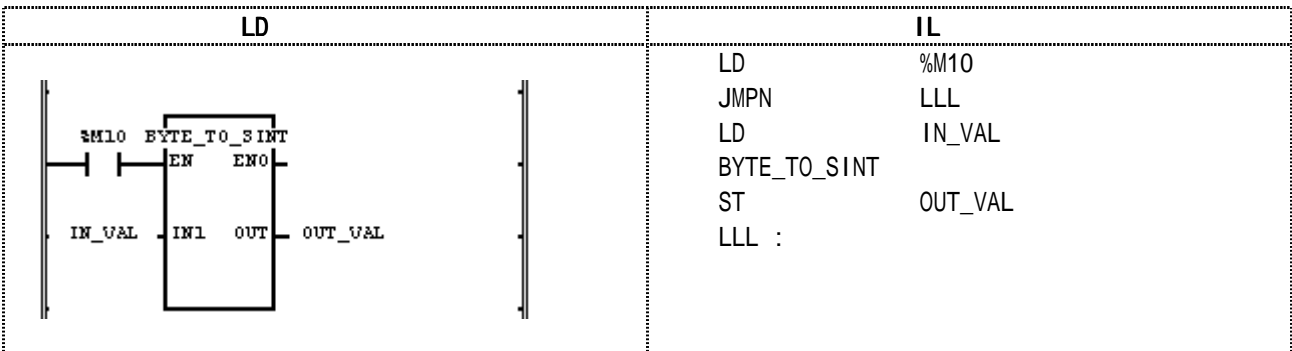
BYTE

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

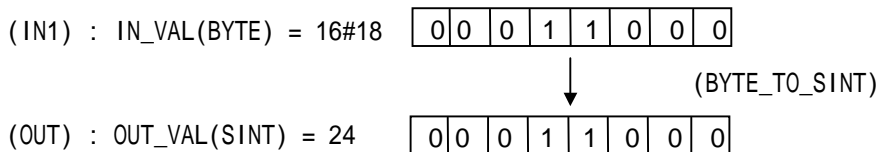


IN OUT

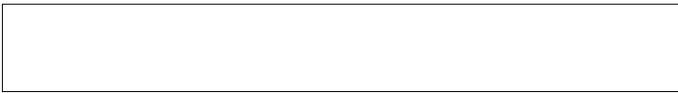
BYTE_TO_SINT	SINT		SINT
BYTE_TO_INT	INT	0	INT
BYTE_TO_DINT	DINT	0	DINT
BYTE_TO_LINT	LINT	0	LINT
BYTE_TO_USINT	USINT		SINT
BYTE_TO_UINT	UINT	0	UINT
BYTE_TO_UDINT	UDINT	0	UDINT
BYTE_TO_ULINT	ULINT	0	ULINT
BYTE_TO_BOOL	BOOL	1	BOOL
BYTE_TO_WORD	WORD	0	WORD
BYTE_TO_DWORD	DWORD	0	DWORD
BYTE_TO_LWORD	LWORD	0	LWORD
BYTE_TO_STRING	STRING		STRING



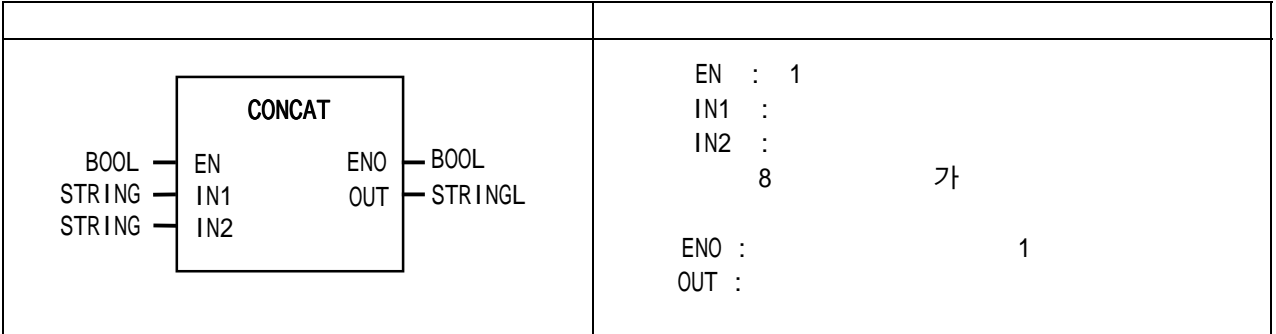
- (1) (%M10) On BYTE_TO_SINT
- (2) IN_VAL(BYTE) = 2#0001_1000 , OUT_VAL(SINT) = 24(2#0001_1000)가



CONCAT

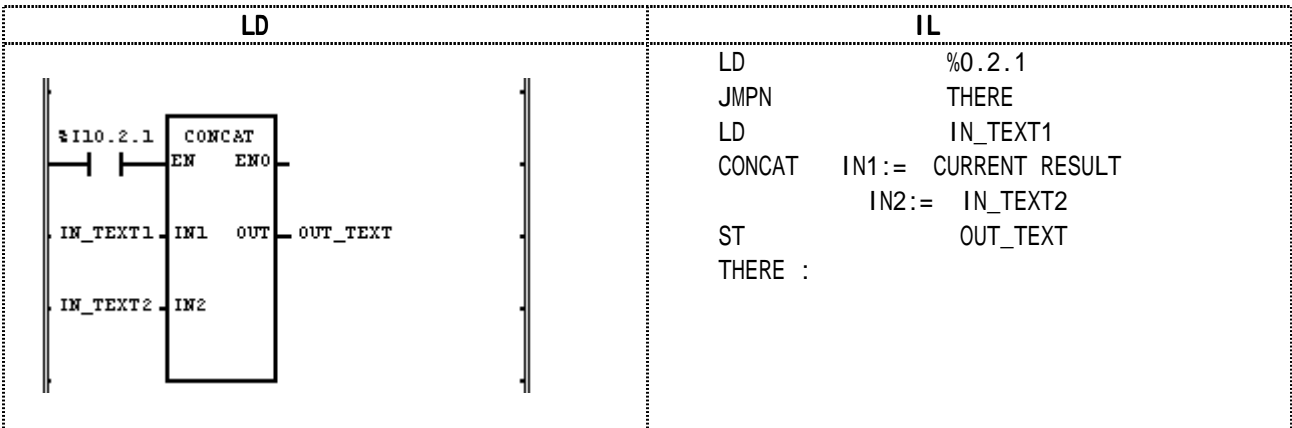


CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1, IN2, IN3, ..., INn(n) OUT

(, _ERR, _LER 가 (Set)) > 30 , OUT 30 CONCAT

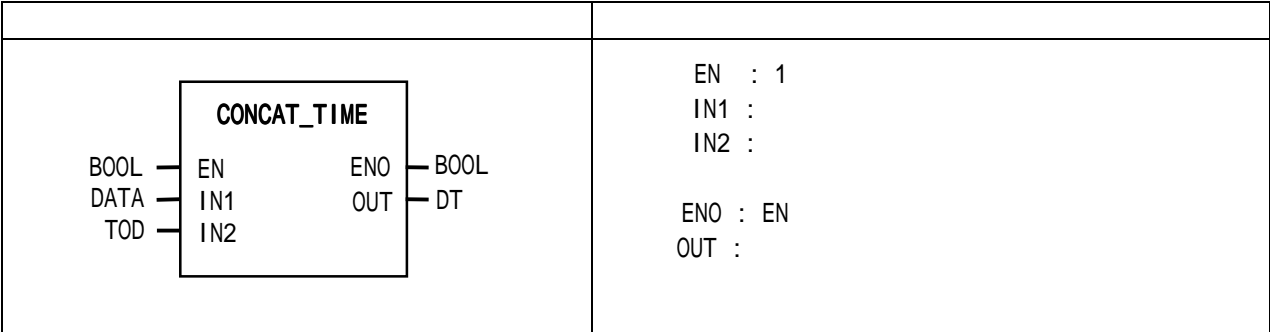


(1) (%I0.2.1) On CONCAT
 (2) IN_TEXT1='ABCD', IN_TEXT2='DEF',
 OUT_TEXT='ABCDEF' 가

(IN1) : IN_TEXT1 (STRING) = `ABCD`
 (CONCAT)
 (IN2) : IN_TEXT2 (STRING) = `DEF`
 ↓
 (OUT) : OUT_TEXT (STRING) = `ABCDEF`

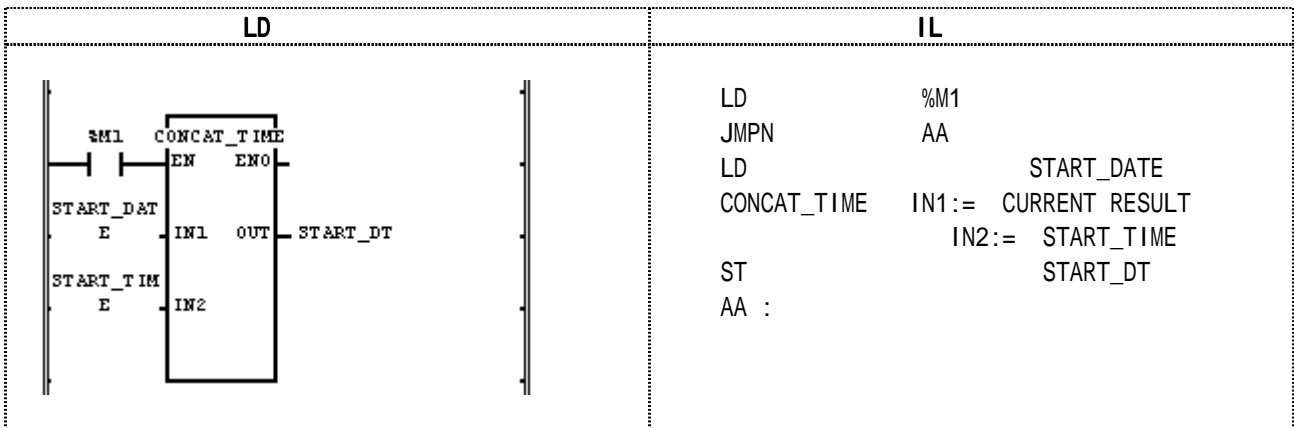
CONCAT_TIME

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN : 1
 IN1 :
 IN2 :
 ENO : EN
 OUT :

IN1() IN2() (DATE_AND_TIME) OUT .



(1) (%M1) On CONCAT_TIME .
 (2) START_DATE = D#1995-12-06 START_TIME = T
 OD#08:30:00 START_DT DT#1995-12-06-08:30:00 .

(IN1) : START_DATE1 (DATE) = D#1995-12-06
 (CONCAT_TIME)

(IN2) : START_TIME (TOD) = TOD#08:30:00



(OUT) : START_DT (DT) = DT#1995-12-06-08:30:00

1 가 EI 가 DI

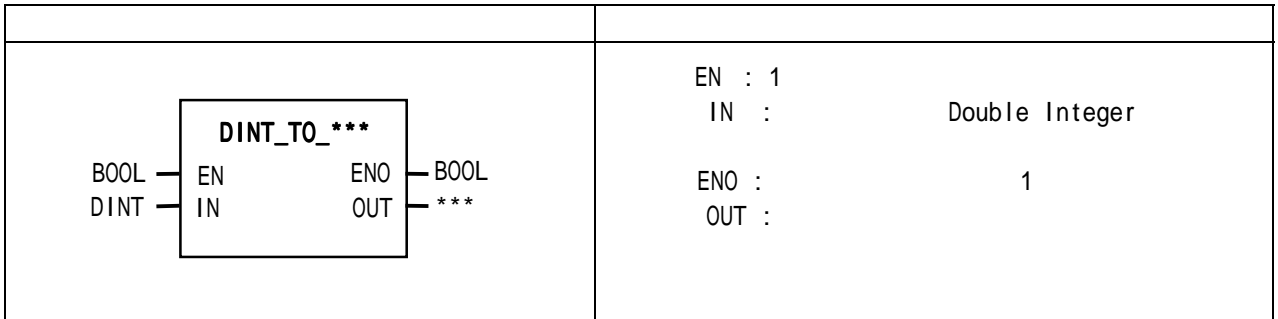
LD	IL
<p>(1) Scan (TASK)</p>	<p>(1) Scan (TASK)</p> <pre> LDN %M100 JMPN KK LD %I0.1.14 DI ST DI_OK KK : LDN %M100 JMPN LL LD %I0.1.15 EI ST EI_OK LL : </pre>
<p>(2)1 가</p>	<p>(2) 1 가</p> <pre> LDN %M1 JMPN MM LD %IWO.0.0 MOVE ST %MW100 MM : </pre>

- (1)DI ()) REQ (%I0.1.14)가 On
DI가 DI_OK 1 .
- (2) DI가 1 .
- (3)EI (가) 가 REQ (%I0.1.15)가 On
EI가 EI_OK 1 .
- (4) EI가 DI .

DINT_TO_***

DINT

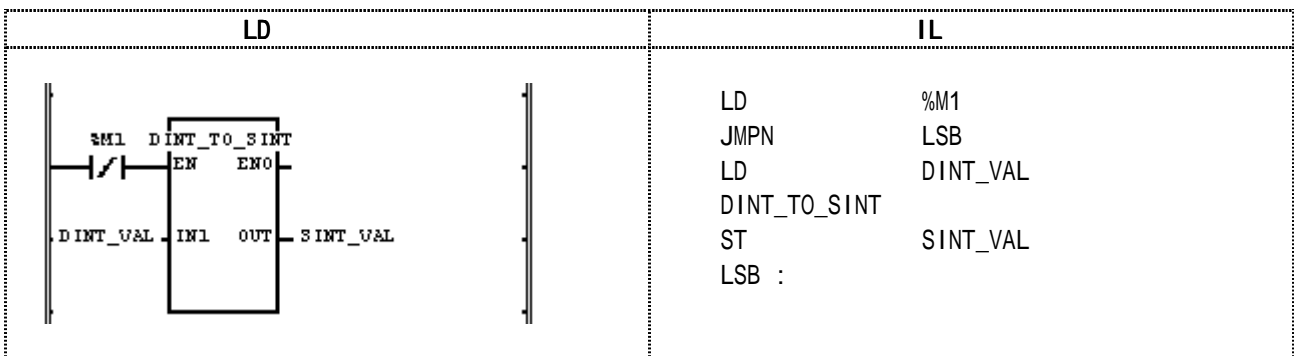
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT

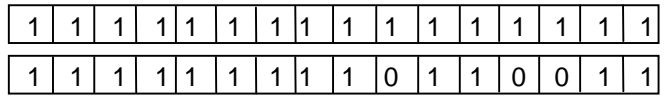
DINT_TO_SINT	SINT	-128 127	가
DINT_TO_INT	INT	-32768 32767	가
DINT_TO_LINT	LINT	LINT	
DINT_TO_USINT	USINT	0 255	가
DINT_TO_UINT	UINT	0 65535	가
DINT_TO_UDINT	UDINT	0 2147483647	가
DINT_TO_ULINT	ULINT	0 2147483647	가
DINT_TO_BOOL	BOOL	1	BOOL
DINT_TO_BYTE	BYTE	8	BYTE
DINT_TO_WORD	WORD	16	WORD
DINT_TO_DWORD	DWORD		DWORD
DINT_TO_LWORD	LWORD	0	LWORD
DINT_TO_BCD	DWORD	0 99,999,999	가
DINT_TO_REAL	REAL	DINT REAL	가
DINT_TO_LREAL	LREAL	DINT LREAL	가

_ERR, _LER 가 (Set)



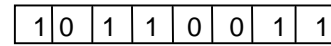
- (1) (%M1) On DINT_TO_SINT가
- (2) INI = DINT_VAL(DINT) = -77 , SINT_VAL(SINT) = -77

(IN1) : DINT_VAL(DINT) = -77



(DINT_TO_SINT)

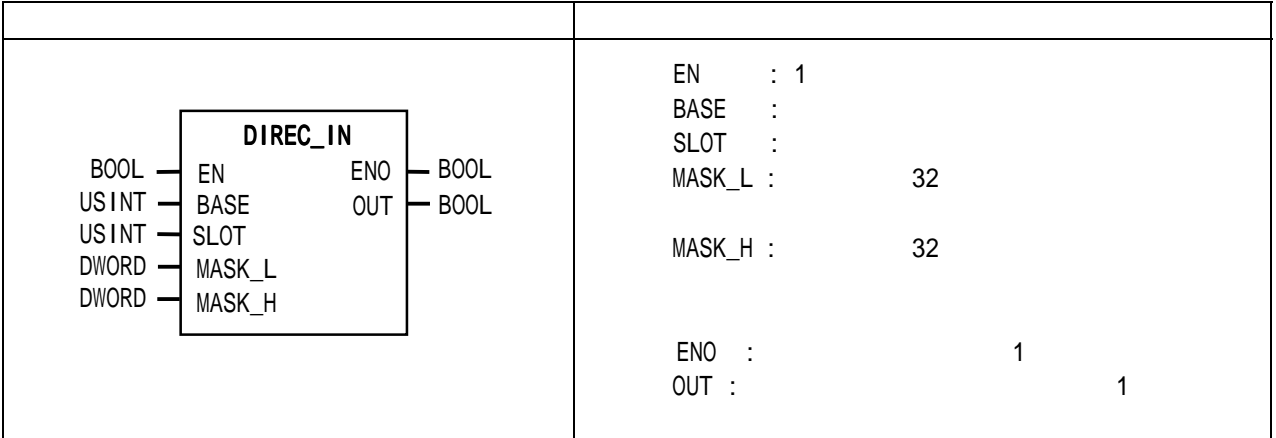
(OUT) : OUT_VAL(SINT) = -77



DIREC_IN



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



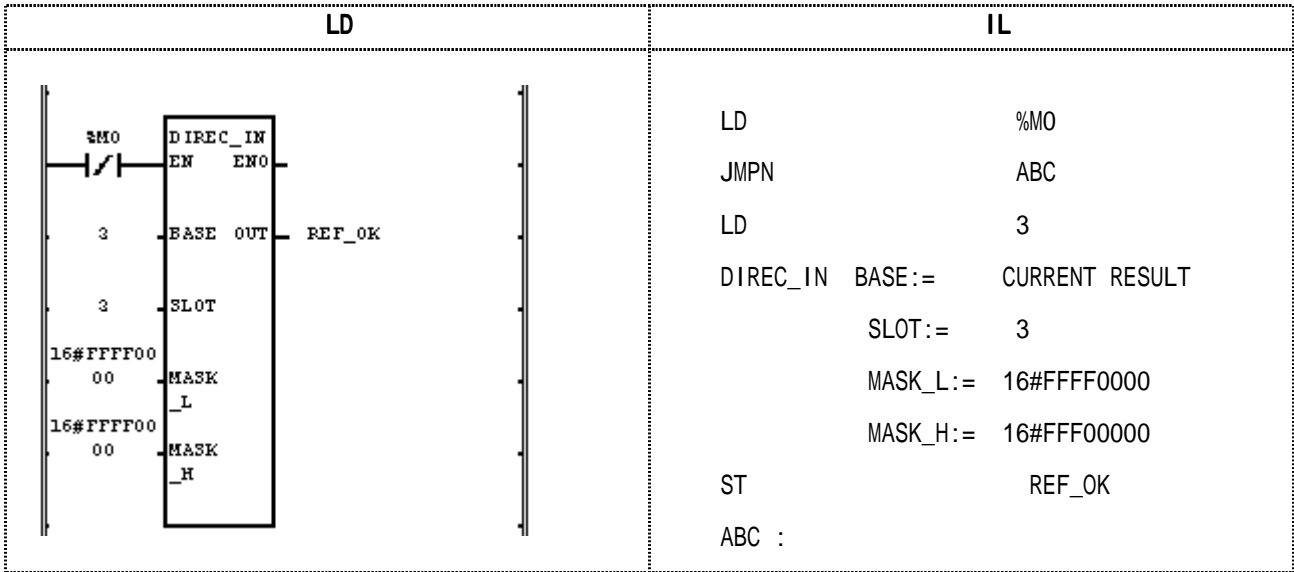
DIREC_IN() EN 1 BASE, SLOT
64 .

DIREC_IN (%I) On/Off 가 .

, 1Scan 가 .

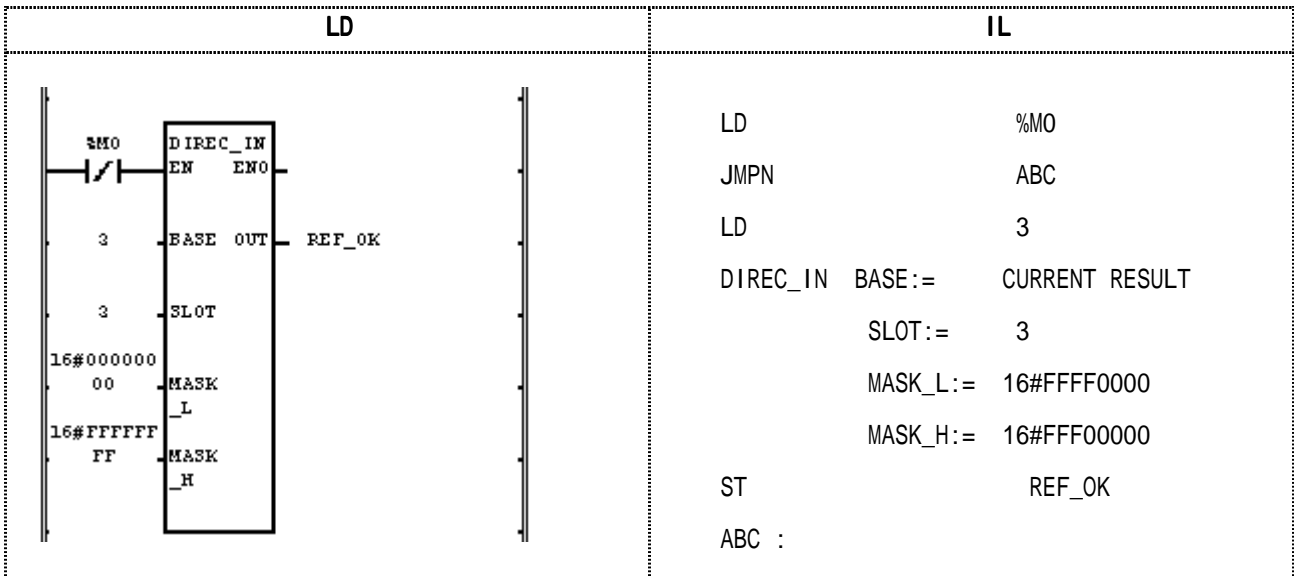
DIREC_IN .

1.3 Base,3 Slot 16 , 가 2# 1010_1010_1110_1011



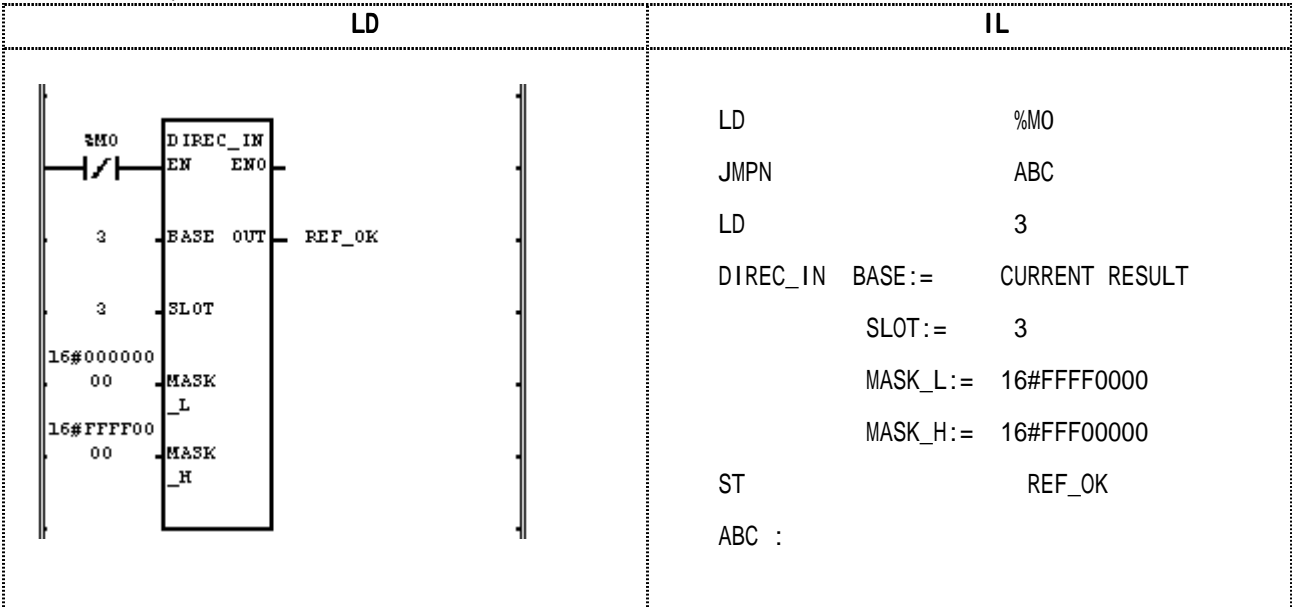
- (1) (%MO)가 On DIREC_IN()
- (2) 16 %IW3.3.0
- (3) MASK_L(32) 16Bit가 %IW3.3.0
#1010_1010_1110_1011
MASK_H(32) , 16

2.3 Base, 3 Slot 32 , 가
2#0000_0000_1111_1111_1100_1100_0011_0011 16Bit



- (1) (%MO)가 On DIREC_IN()
- (2) 32 %ID3.3.0가 ,
- (3) MASK_L(32) 16 가 %IW3.3.0
2#1100_1100_0011_0011

3. 3 Base, 3 Slot 64 가
 16#0000_FFFF_AAAA_7777(2#0000_0000_0000_1111_1111_1111_1111_1010_1010_1010_1010_0111_0111_0111_0111) 64 48 .

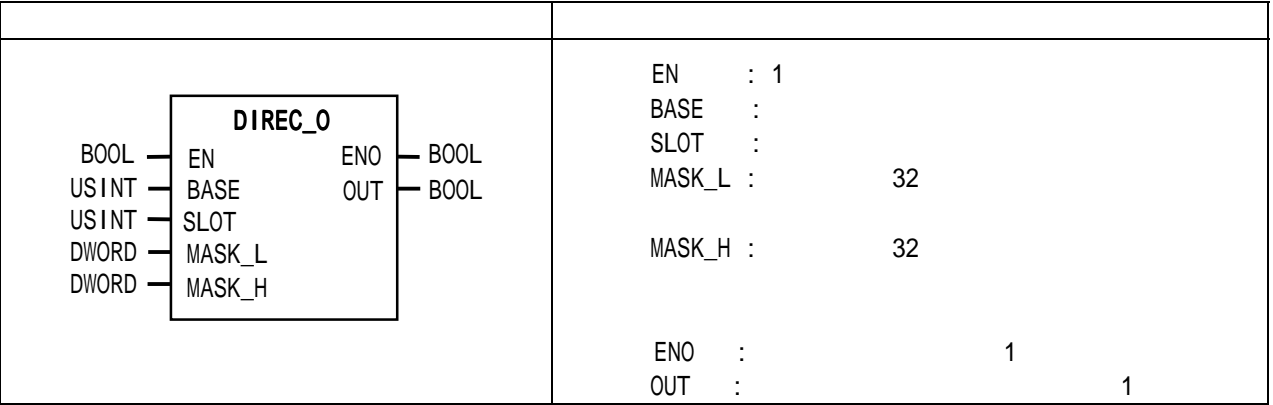


- (1) (#M0)가 On DIREC_IN()
 (2) 64 %IL3.3.0 %ID3.3.0 %ID3.3.1
 32 (MASK_L) %ID3.3.0
 32 (MASK_H) 32Bit %ID3.3.1 %IW.3.3.2
 , %IW3.3.3
- %IL3.3.0 [%ID3.3.0 [%IW.3.3.0:2#0111_0111_0111_0111
 %ID3.3.1 [%IW.3.3.1:2#1010_1010_1010_1010
 %IW3.3.2:2#1111_1111_1111_1111
 %IW3.3.3:
- (3) REF_OK() 1 .

DIREC_O



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



```

DIREC_O(          ) EN(DIREC_O          ) 1  BASE  SLOT
                64  MASK(1)

DIREC_O 1          (%Q) On/Off          가 .

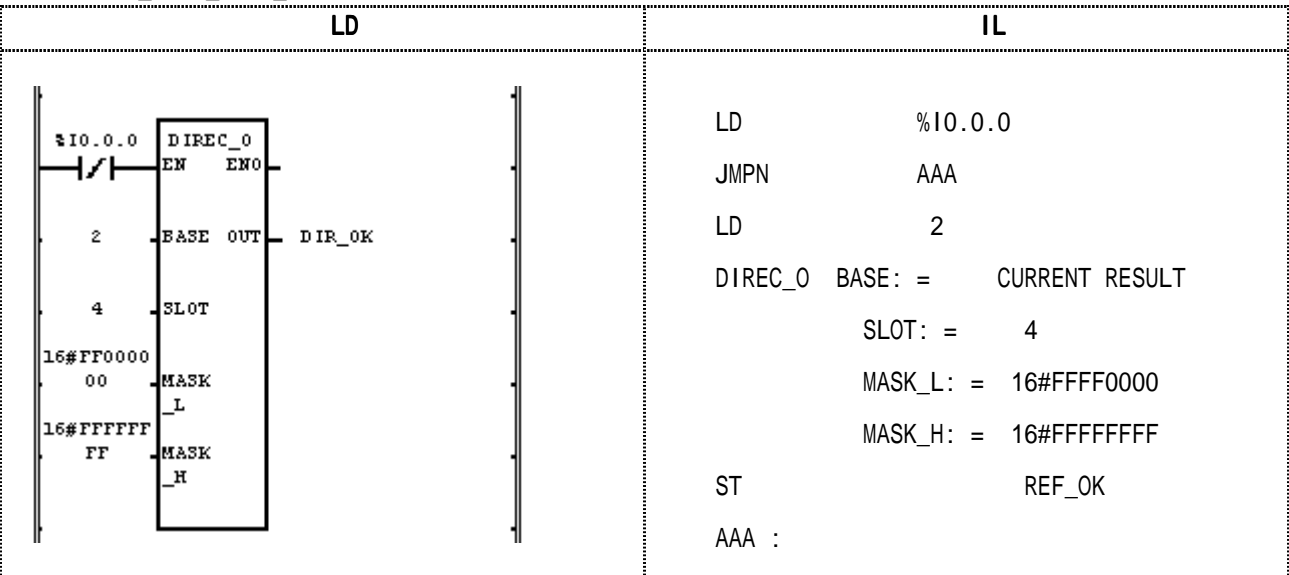
                1          가 .

DIREC_O          가          가

                가          ENO

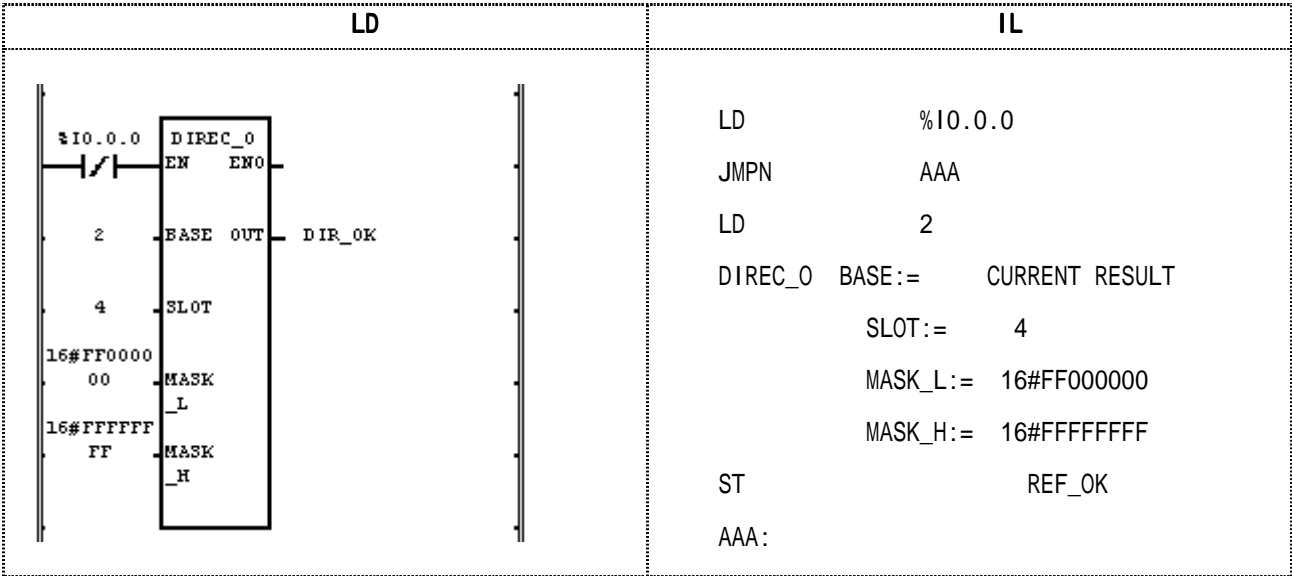
OUT 0          . (          1          )
    
```

1. 3 Slot 16 Relay
2#0111_0111_0111_0111



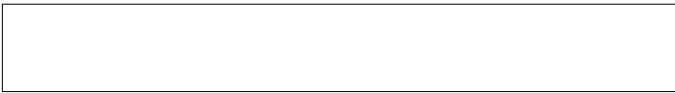
- (1) Base 2 SLOT 4
- (2) 가 16 MASK_L 16
- (3) .(16#FFFF0000)
(%I0.0.0)가 On DIREC_O(
가 2#0111_0111_0111_0111)

2.2 4 Slot 32 TR. 24
2#1111_0000_1111_0000_1111_0000

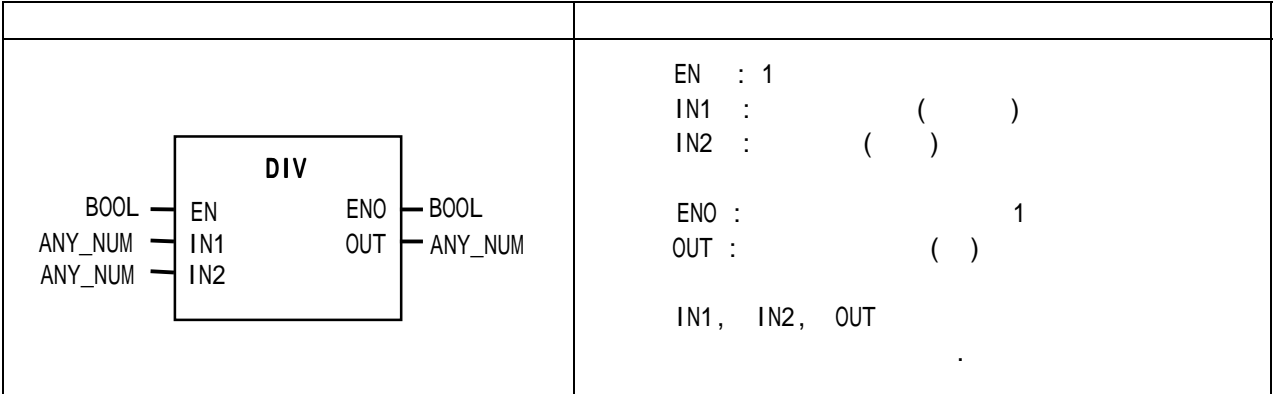


- (1) Base 2 SLOT 4
- (2) 가 24 MASK_L 24
- (3) .(16#FF000000)
(%I0.0.0)가 Off DIREC_O(
가 2# _1111_0000_1111_0000_1111_0000)

DIV



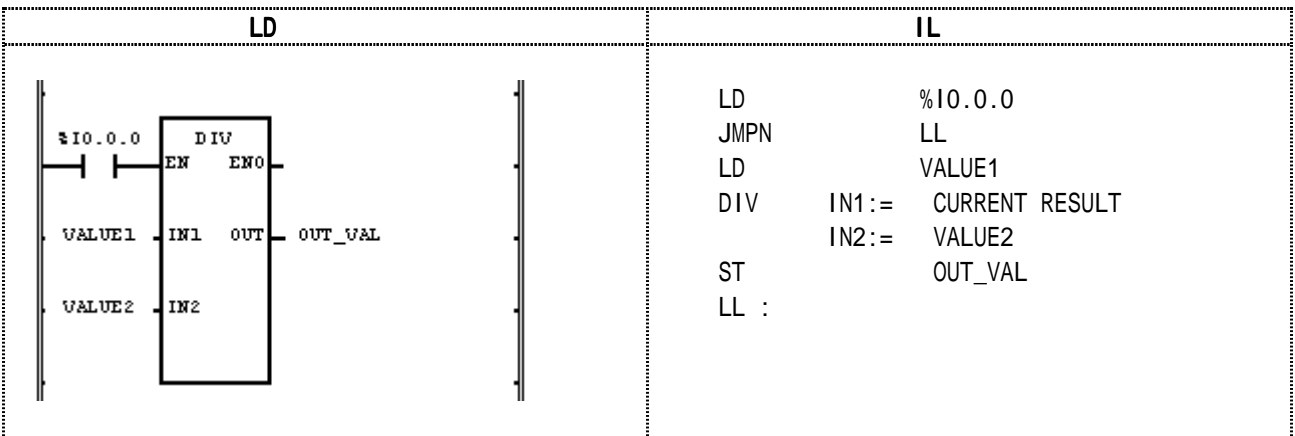
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



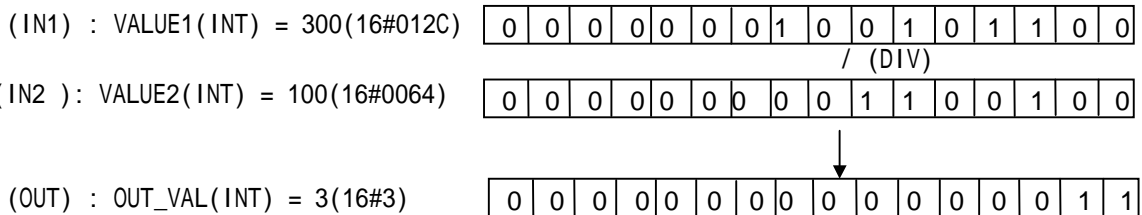
IN1 IN2 OUT
 OUT = IN1/IN2

IN1	IN2	OUT
7	2	3
7	-2	-3
-7	2	-3
-7	-2	3
7	0	x

() '0' _ERR, _LER 가 (Set)

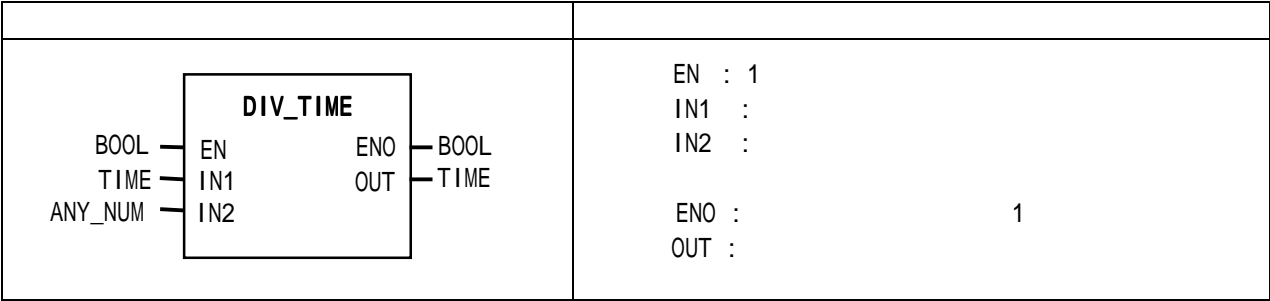


- (1) (%I0.0.0) On DIV가
- (2) VALUE1 = 300, VALUE2 = 100 , OUT_VAL = 300/100 = 3



DIV_TIME

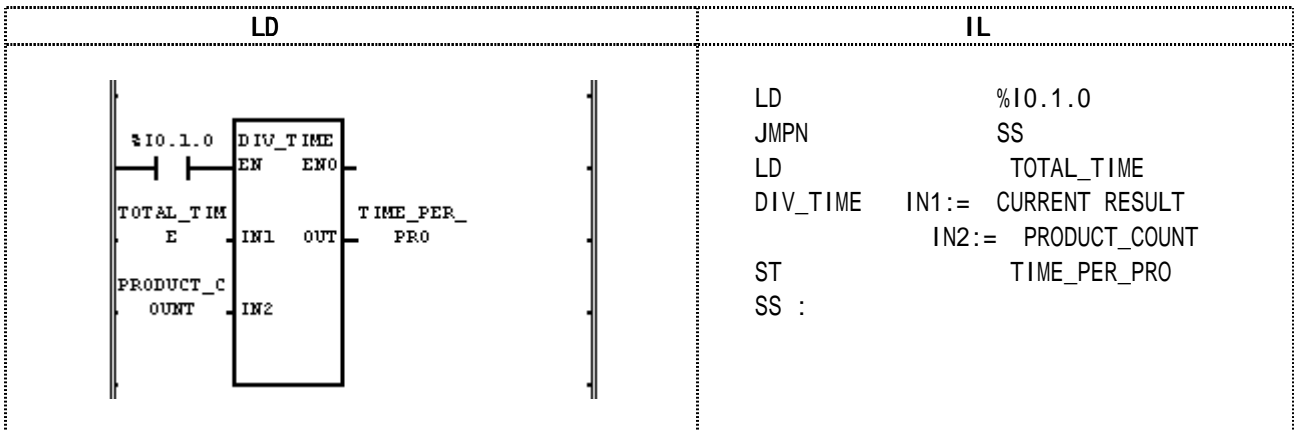
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1() IN2() OUT .

(IN2)가 0 , _ERR, _LER 가 (Set) .

Line 12 24 24 , 12



(1) (%I0.1.0) On DIV_TIME .
 (2) TOTAL_TIME(T#12H24M24S) PRODUCT_COUNT(12) ,
 TIME_PER_PRO(T#1H2M2S) , 1 2 2 .

$$(IN1) : \text{TOTAL_TIME}(\text{TIME}) = \text{T\#12H24M24S} / (\text{DIV_TIME})$$

$$(IN2) : \text{PRODUCT_COUNT}(\text{INT}) = 12$$

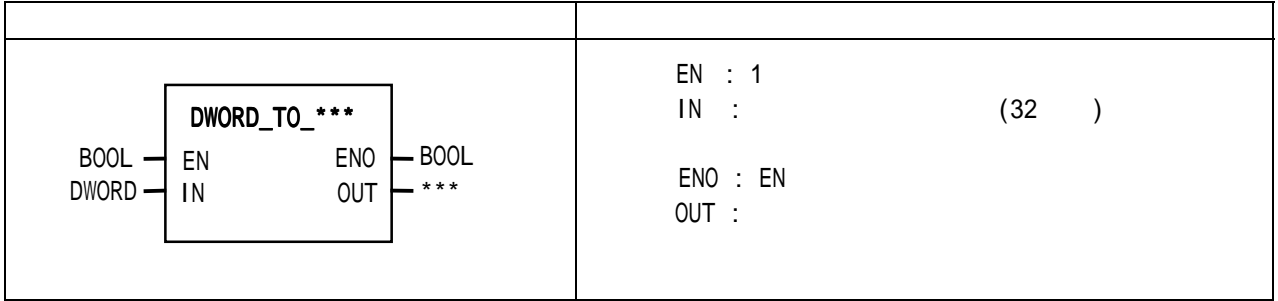
↓

$$(\text{OUT}) : \text{TIME_PER_PRO}(\text{TIME}) = \text{T\#1H2M2S}$$

DWORD_TO_***

DWORD

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



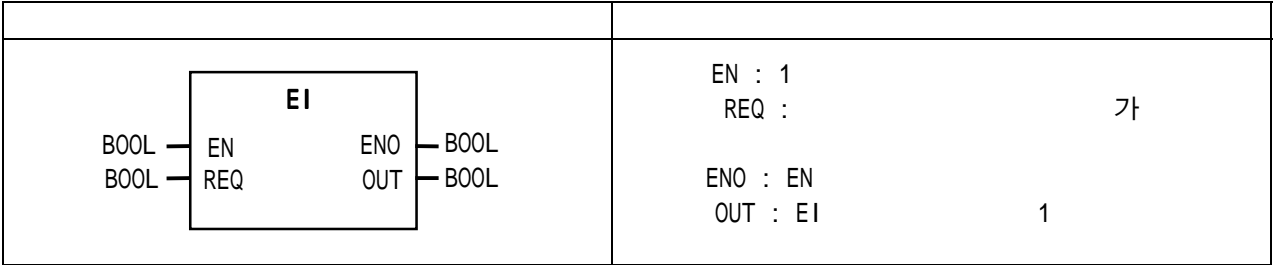
IN OUT .

DWORD_TO_SINT	SINT	8	SINT	.
DWORD_TO_INT	INT	16	INT	.
DWORD_TO_DINT	DINT		DINT	.
DWORD_TO_LINT	LINT	0	LINT	.
DWORD_TO_USINT	USINT	8	USINT	.
DWORD_TO_UINT	UINT	16	UINT	.
DWORD_TO_UDINT	UDINT		UDINT	.
DWORD_TO_ULINT	ULINT	0	ULINT	.
DWORD_TO_BOOL	BOOL	1	BOOL	.
DWORD_TO_BYTE	BYTE	8	BYTE	.
DWORD_TO_WORD	WORD	16	WORD	.
DWORD_TO_LWORD	LWORD	0	LWORD	.
DWORD_TO_REAL	REAL		REAL	.
DWORD_TO_TIME	TIME		TIME	.
DWORD_TO_TOD	TOD		TOD	.
DWORD_TO_STRING	STRING	Decimal	STRING	.

EI

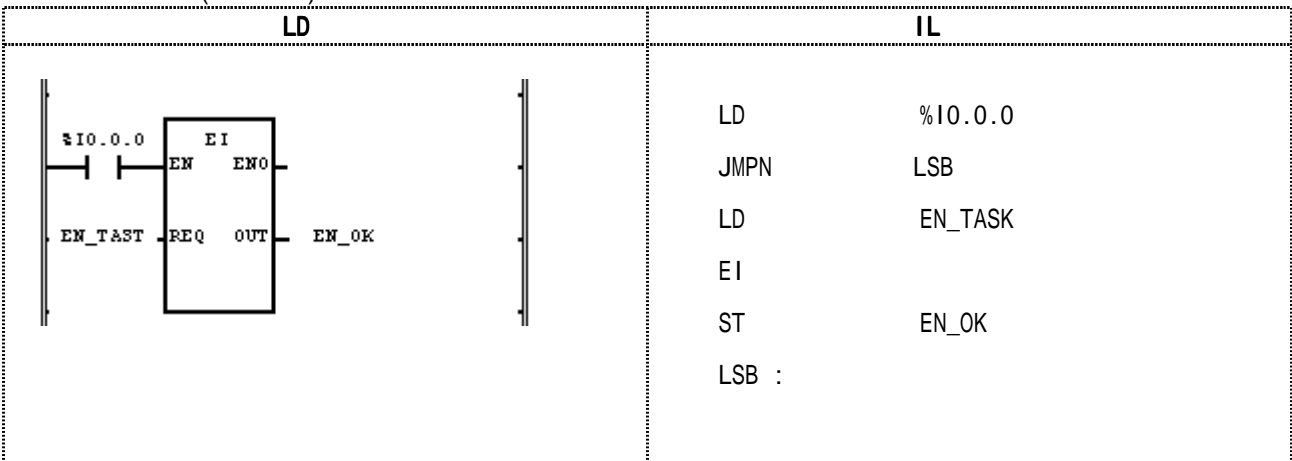
(DI) 가

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1 REQ 1 'DI'
'EI' REQ 0 'EI'

(DI)

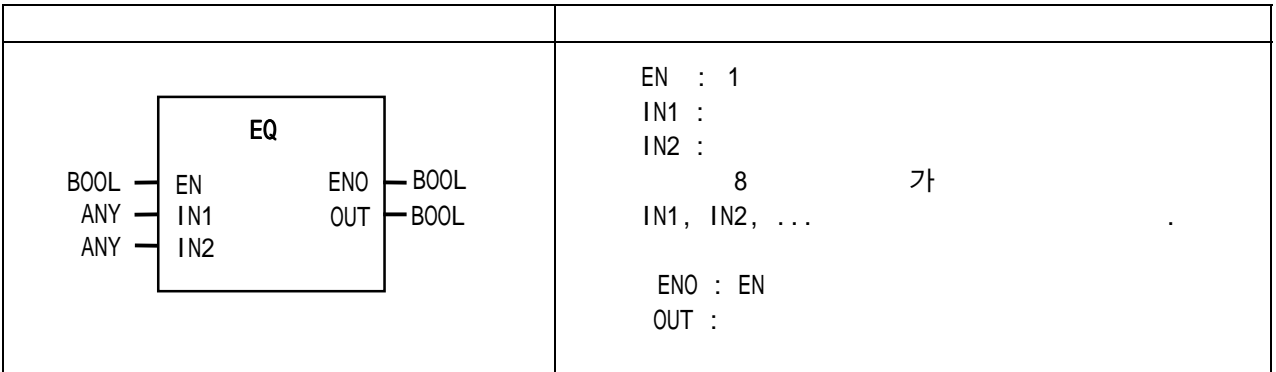


EN_TASK가 1
'EI' 가 가 EN_OK 1

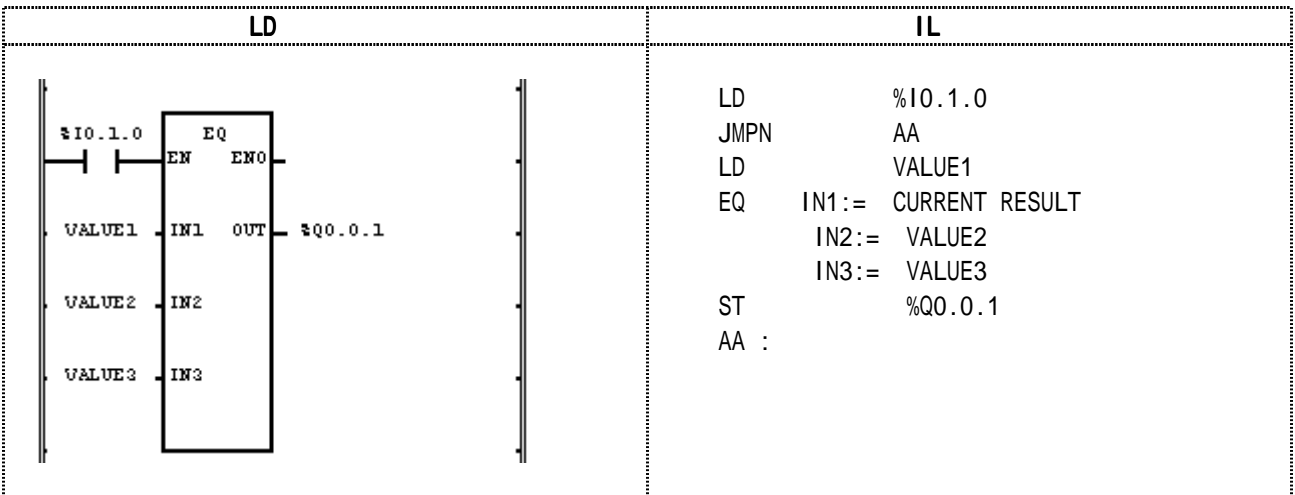
EQ

--

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1=IN2=IN3...=INn(n) OUT 1
OUT 0



(1) (%I0.1.0) On 'EQ'가
(2) VALUE1 = 300, VALUE2 = 300, VALUE3 = 300 , VALUE1 = VALUE2 = VALUE3
%Q0.0.1 = 1

(IN1) : VALUE1(INT) = 300(16#012C)

0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0
= (EQ)

(IN2) : VALUE2(INT) = 300(16#012C)

0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0
= (EQ)

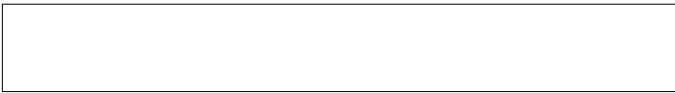
(IN3) : VALUE1(INT) = 300(16#012C)

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

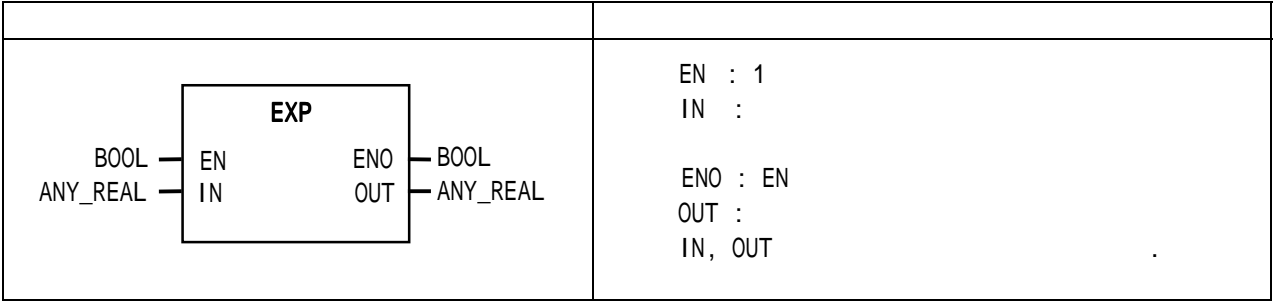
(OUT) : %Q0.0.1(BOOL) = 1(16#1)

↓
1

EXP

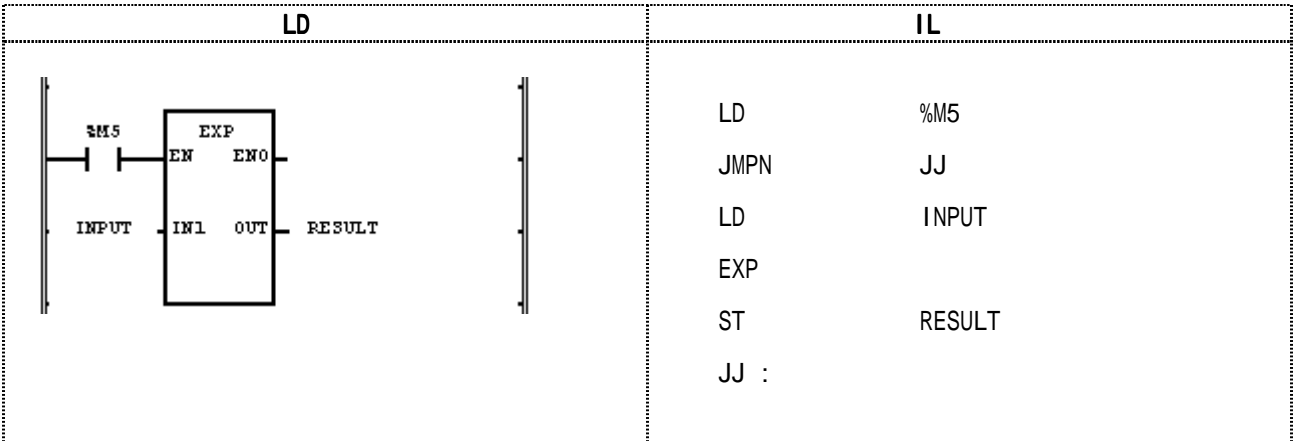


CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT

$OUT = e^{IN}$



(1) (%M5) On 'EXP'가

(2) INPUT 2.0 , RESULT 7.3890

$e^{2.0} = 7.3890.....$

(IN1) : INPUT(REAL) = 2.0

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

(16#40000000)

↓ (EXP)

(OUT) : RESULT(REAL) = 7.38905621E+00

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

(16#40EC7326)

EXPT



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<pre> EN : 1 IN1 : IN2 : ENO : 1 OUT : IN1 OUT </pre>
--	--

IN1 IN2 OUT
OUT = IN1^{IN2}

_ERR, _LER 가 (Set)

LD	IL
	<pre> LD %I0.1.0 JMPN LSB LD IN_VAL EXPT IN1:= CURRENT RESULT IN2:= VALUE ST OUT_VAL LSB : </pre>

- (1) (%I0.1.0) On 'EXPT'가
- (2) IN_VAL = 1.5, VALUE = 3 , OUT_VAL = 1.5³ = 1.5 × 1.5 × 1.5 = 3.375가

(IN1) : IN_VAL(REAL) = 1.5

(IN2) : VALUE(INT) = 3

↓ (EXPT)

(OUT) : OUT_VAL(REAL) = 3.37500000E+00

GE

· ,

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<p>EN : 1 IN1 : IN2 : 8 가 IN1, IN2, ... ENO : EN OUT :</p>
--	--

IN1 ≥ IN2 ≥ IN3 ... ≥ INn (n) OUT 1
 OUT 0

<p>LD</p>	<p>IL</p> <pre>LD %M77 JMPN YY LD VALUE1 GE IN1= CURRENT RESULT IN2= VALUE2 IN3= VALUE3 ST %Q0.0.1 YY:</pre>
------------------	---

(1) (%M77) On GE(:)
 (2) VALUE1=300, VALUE3=200 , VALUE1 ≥ VALUE2 ≥ VALUE 3 ,
 %Q0.01=1

(IN1) : VALUE1(INT) = 300(16#012C)

0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 (GE)

(IN2) : VALUE2(INT) = 200(16#00C8)

0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 (GE)

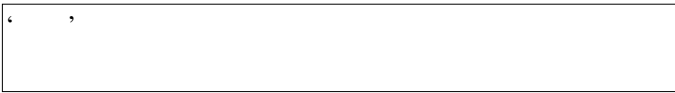
(IN3) : VALUE3(INT) = 100(16#0064)

0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

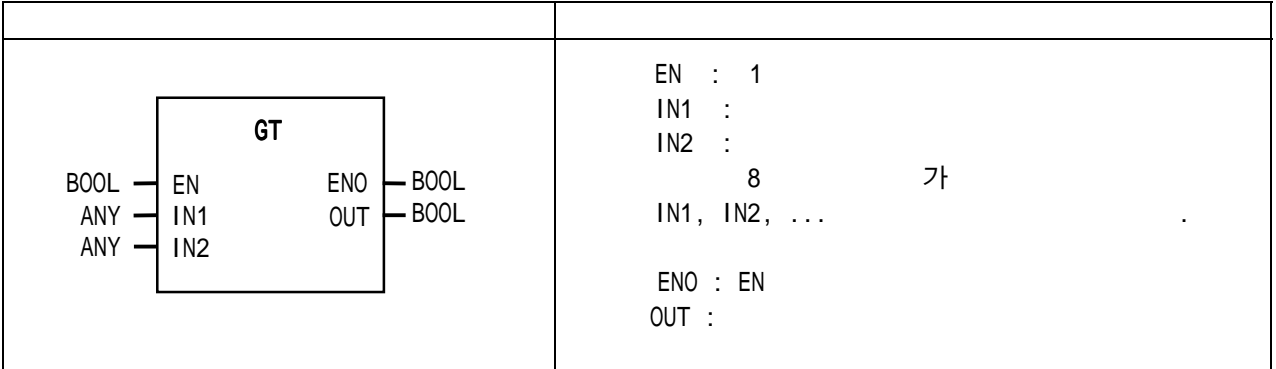
(OUT) : %Q0.0.1(BOOL) = 1(16#1)

1

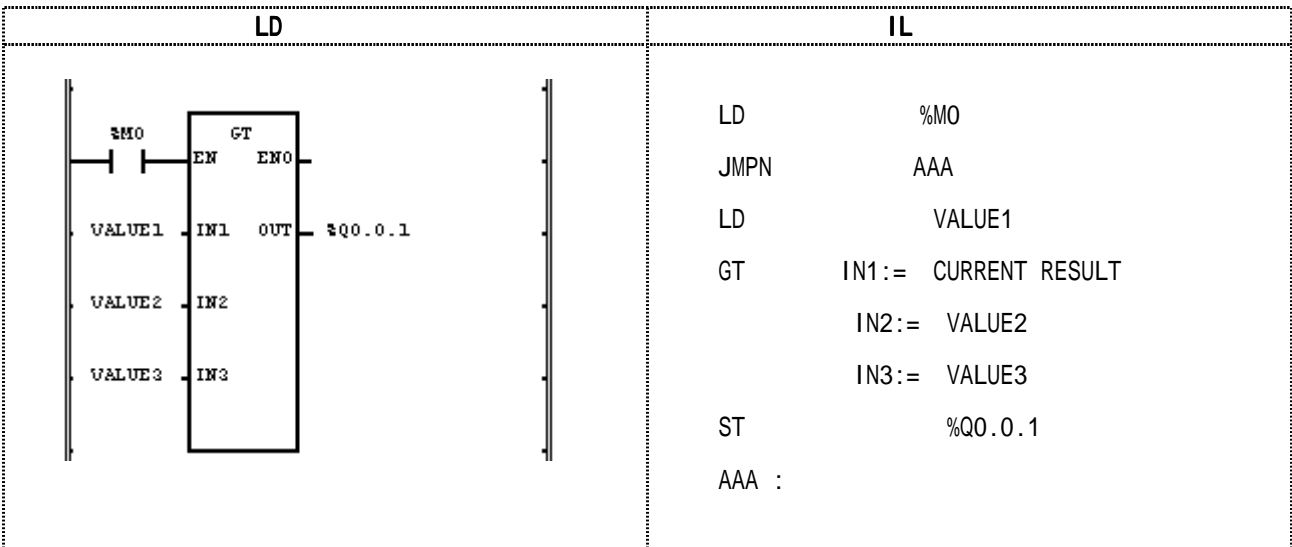
GT



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1>IN2>IN3...>INn(n) OUT 1
 OUT 0 .



(1) (%M0) On GT(:)
 (2) VALUE1 = 300, VALUE2 = 200, VALUE3 = 100 , VALUE1 > VALUE2 > VALUE3
 %Q0.0.1=1 .

(IN1) : VALUE1(INT) = 300(16#012C)

0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 > (GT)

(IN2) : VALUE2(INT) = 200(16#00C8)

0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 > (GT)

(IN3) : VALUE3(INT) = 100(16#0064)

0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

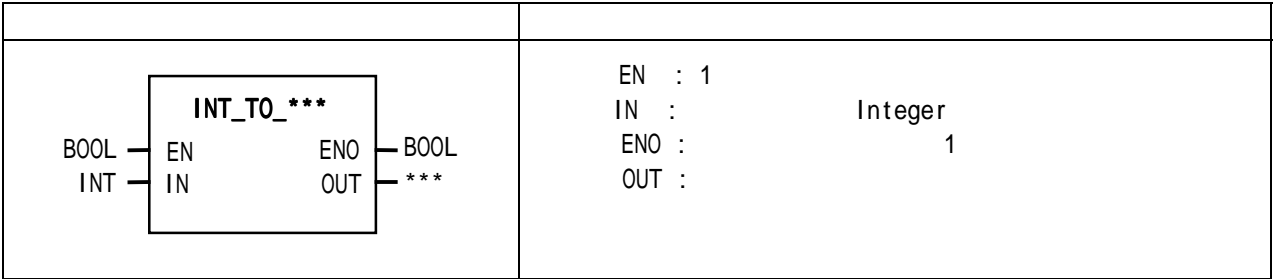
(OUT) : %Q0.0.1(BOOL) = 1(16#1)

1

INT_TO_***

INT

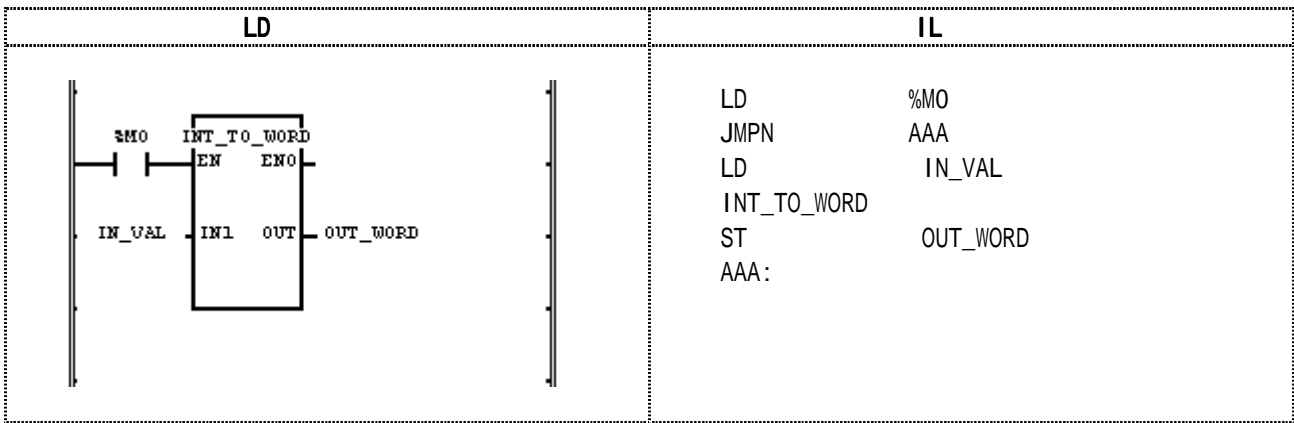
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



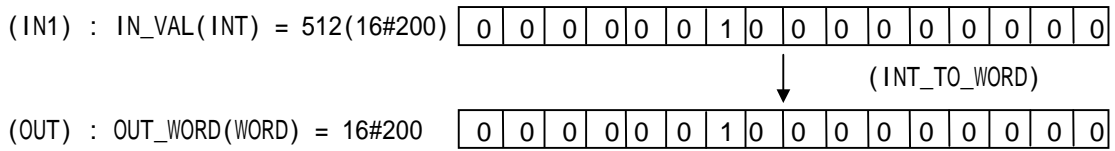
IN OUT .

INT_TO_SINT	SINT	-128 127	, 가 .
INT_TO_DINT	DINT	DINT	.
INT_TO_LINT	LINT	LINT	.
INT_TO_USINT	USINT	0 255	, 가 .
INT_TO_UINT	UINT	0 32767	, 가 .
INT_TO_UDINT	UDINT	0 32767	, 가 .
INT_TO_ULINT	ULINT	0 32767	, 가 .
INT_TO_BOOL	BOOL	1 BOOL	.
INT_TO_BYTE	BYTE	8 BYTE	.
INT_TO_WORD	WORD	WORD	.
INT_TO_DWORD	DWORD	0 DWORD	.
INT_TO_LWORD	LWORD	0 LWORD	.
INT_TO_BCD	WORD	0 9,999	, 가 .
INT_TO_REAL	REAL	INT REAL	.
INT_TO_LREAL	LREAL	INT LREAL	.

_ERR _LER 가 (Set) .



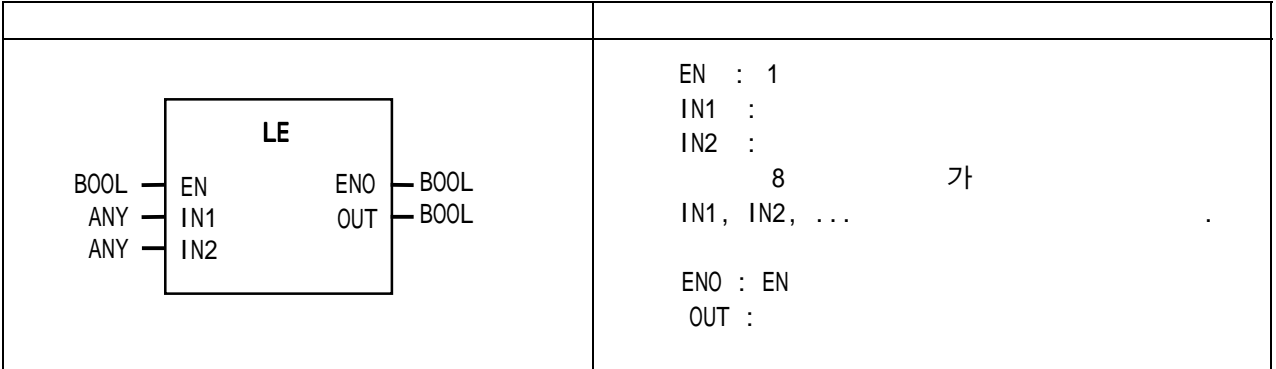
- (1) (%M0) On INT_TO_WORD .
- (2) IN_VAL(INT) = 512(16#200) , OUT_WORD(WORD)
=16#200 .



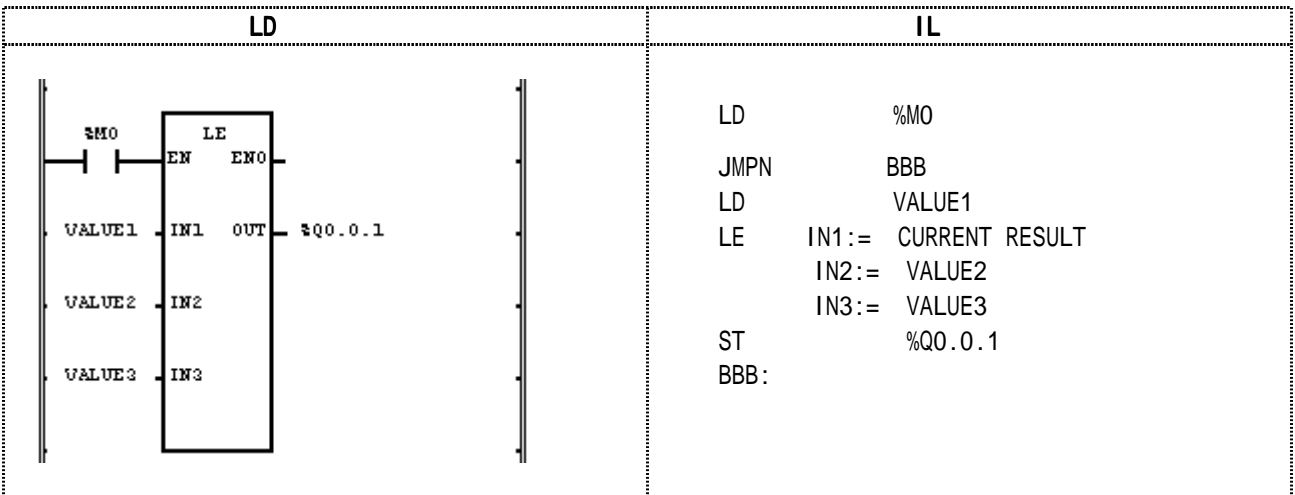
LE

--

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1 ≤ IN2 ≤ IN3... ≤ INn (n) OUT 1
 OUT 0



- (1) (%M0)가 On LE(:)
 (2) VALUE1=150, VALUE2=200, VALUE3 = 250 , VALUE1≤VALUE2≤VALUE3
 3 , %Q0.0.1=1

(IN1) : VALUE1(INT) = 150(16#0096)

0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 ≤ (LE)

(IN2) : VALUE2(INT) = 200(16#00C8)

0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 ≤ (LE)

(IN3) : VALUE1(INT) = 250(16#0064)

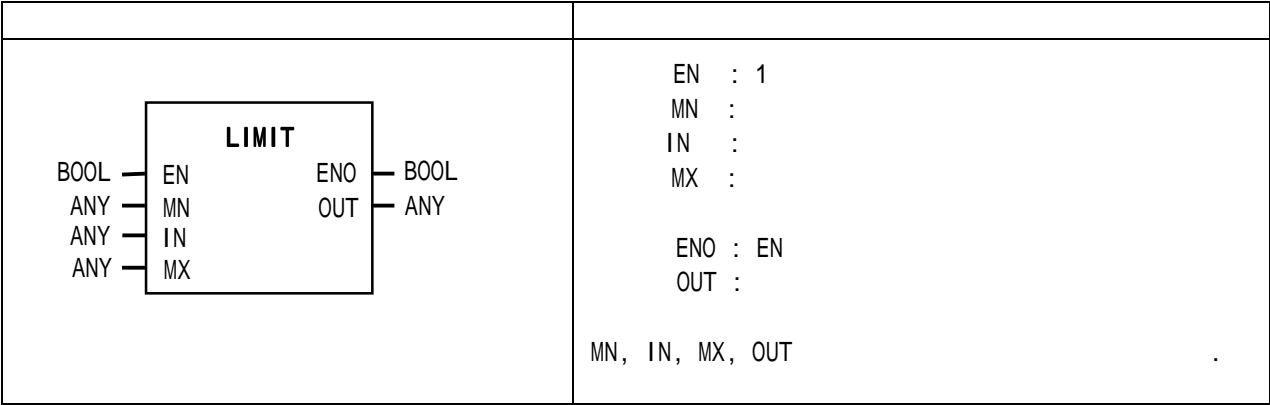
0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(OUT) : %Q0.0.1(BOOL) = 1(16#1)

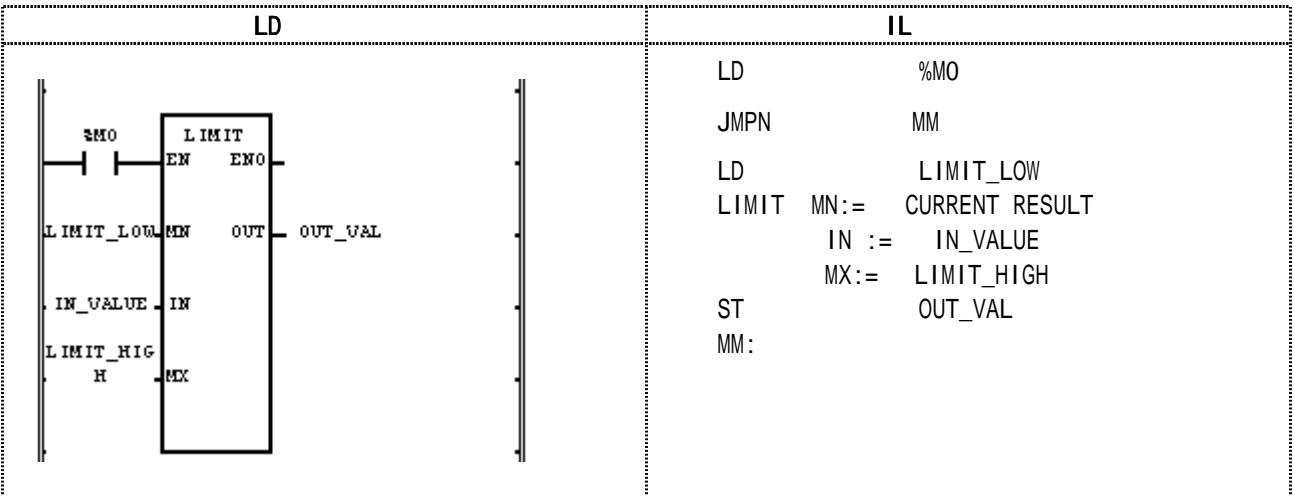
1

LIMIT

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



$IN \leq MN \leq MX$, $OUT = IN$
 $IN < MN$, $OUT = MN$
 $IN > MX$, $OUT = MX$



- (1) (%MO)가 On LIMIT()
 (2) (LIMIT_LOW) , (LIMIT_HIGH), (IN_VALUE) (OUT_VAL)

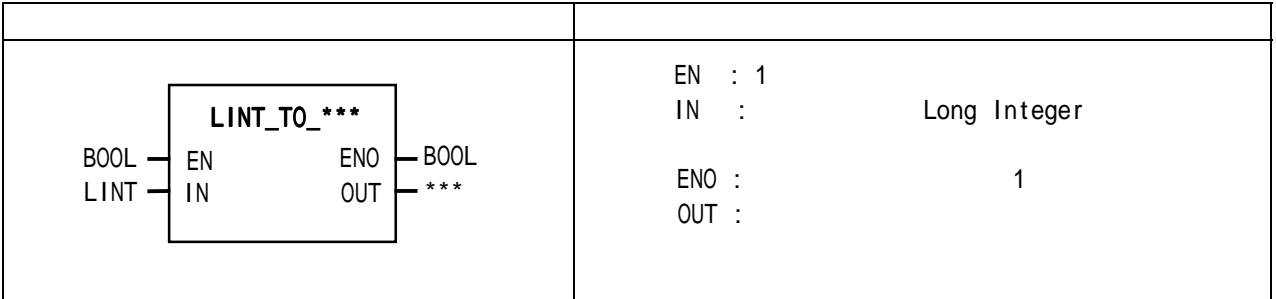
LIMIT_LOW	IN_VALUE	LIMIT_HIGH	OUT_VAL
1000	2000	3000	2000
1000	500	3000	1000
1000	4000	3000	3000

(MN): LIMIT_LOW (INT) = 1000
 (IN) : IN_VALUE (INT) = 4000
 (MX) : IN_VALUE (INT) = 3000
 ↓ (LIMIT)
 (OUT) : OUT_VAL (INT) = 3000

LINT_TO_***

LINT

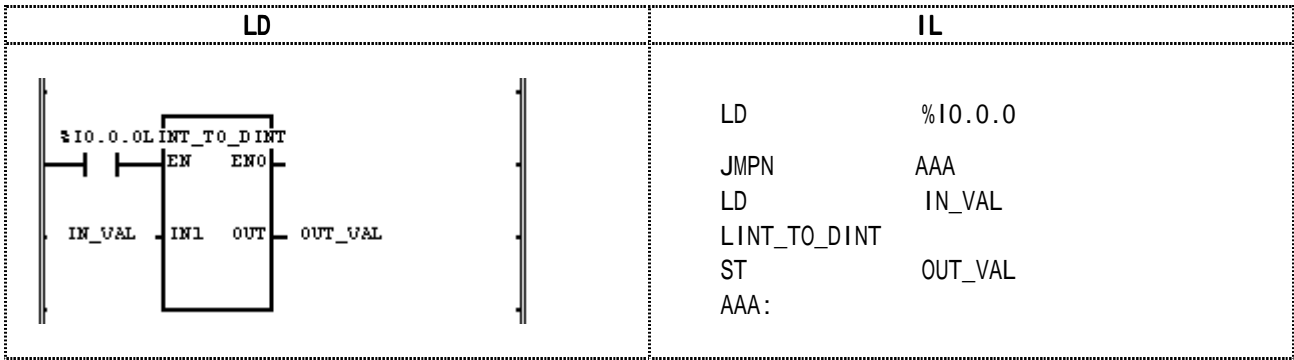
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT .

LINT_TO_SINT	SINT	-128 127	, 가 .
LINT_TO_INT	INT	-32,768 32,767	, 가 .
LINT_TO_DINT	DINT	-2^{31} $2^{31}-1$, 가 .
LINT_TO_USINT	USINT	0 255	, 가 .
LINT_TO_UINT	UINT	0 65,535	, 가 .
LINT_TO_UDINT	UDINT	0 $2^{32}-1$, 가 .
LINT_TO_ULINT	ULINT	0 $2^{63}-1$, 가 .
LINT_TO_BOOL	BOOL	1 BOOL	. .
LINT_TO_BYTE	BYTE	8 BOOL	. .
LINT_TO_WORD	WORD	16 BOOL	. .
LINT_TO_DWORD	DWORD	32 BOOL	. .
LINT_TO_LWORD	LWORD	LWORD	. .
LINT_TO_BCD	LWORD	0 9,999,999,999,999,999	가 .
LINT_TO_REAL	REAL	LINT REAL	가 .
LINT_TO_LREAL	LREAL	LINT LREAL	가 .

_ERR, _LER 가 (Set) .



- (1) (%10.0.0) On LINT_TO_DINT
- (2) IN_VAL(LINT) = 123_456_789 , OUT_VAL(DINT) = 123_456_789가

(IN1) : IN_VAL(LINT) = 123,456,789
(16#75BCD15)

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

↓ (LINT_TO_DINT)

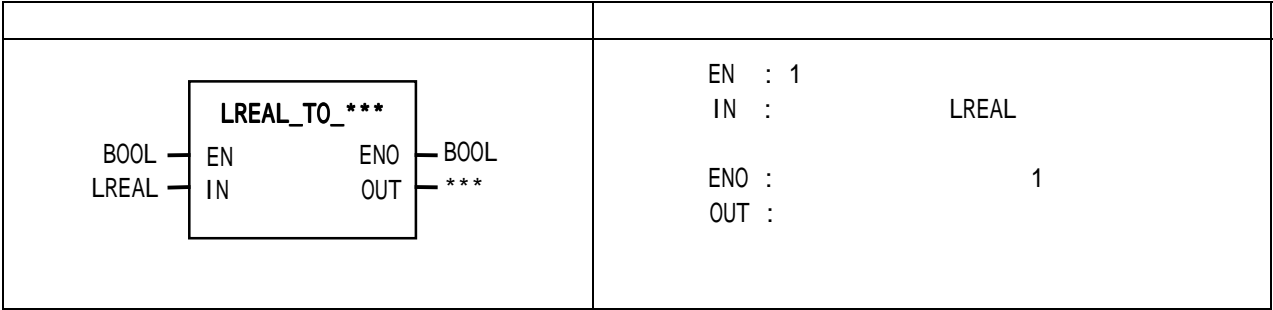
(OUT) : OUT_VAL(DINT) = 123,456,789
(16#75BCD15)

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

LREAL_TO_***

LREAL

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT

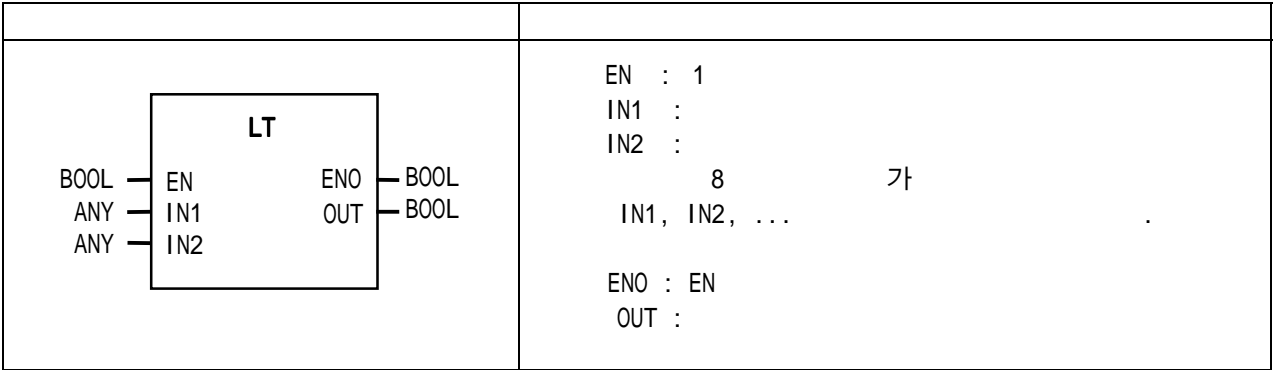
LREAL_TO_SINT	SINT	가 -128 127 · ()
LREAL_TO_INT	INT	가 -32768 32767 · ()
LREAL_TO_DINT	DINT	가 -2 ³¹ 2 ³¹ -1 · ()
LREAL_TO_LINT	LINT	가 -2 ⁶³ 2 ⁶³ -1 · ()
LREAL_TO_USINT	USINT	가 0 255 · ()
LREAL_TO_UINT	UINT	가 0 65,535 · ()
LREAL_TO_UDINT	UDINT	가 0 2 ³² -1 · ()
LREAL_TO_ULINT	ULINT	가 0 2 ⁶⁴ -1 · ()
LREAL_TO_LWORD	LWORD	LWORD
LREAL_TO_REAL	REAL	LREAL REAL 가

(Set) 0 (Overflow) _ERR, _LER 가

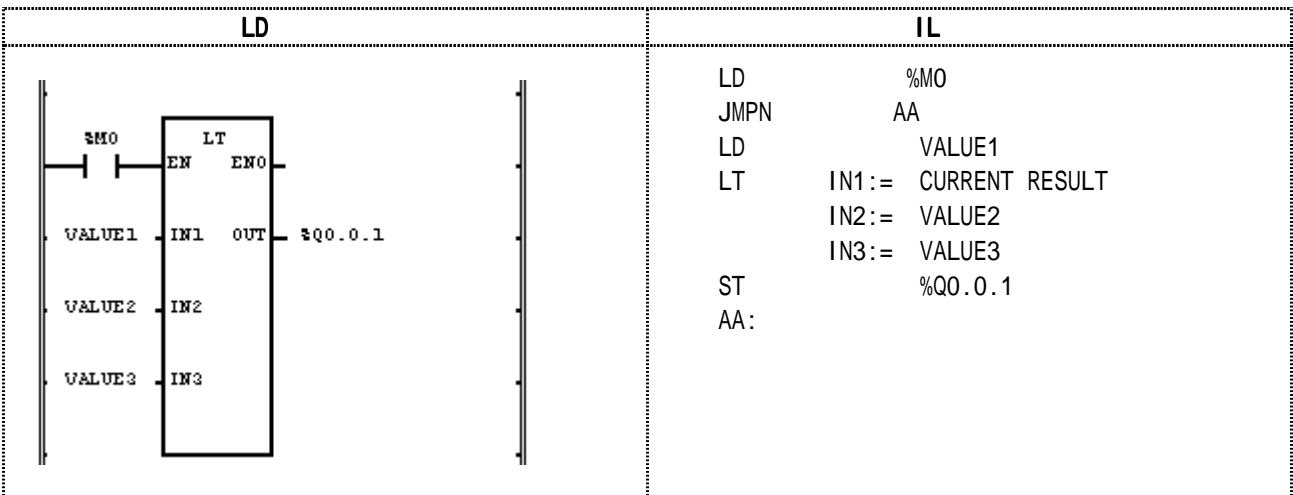
LT

‘ , ’

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1<IN2<IN3...<INn(n) OUT 1
OUT 0 .



- (1) (%M0) On LT(:)
- (2) VALUE1 = 100, VALUE2 = 200, VALUE3 = 300 , VALUE1 < VALUE2 < VALUE3
%Q0.0.1=1

(IN1) : VALUE1(INT) = 100(16#0064)

0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

< (LT)

(IN2) : VALUE2(INT) = 200(16#00C8)

0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

< (LT)

(IN3) : VALUE3(INT) = 300(16#012C)

0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

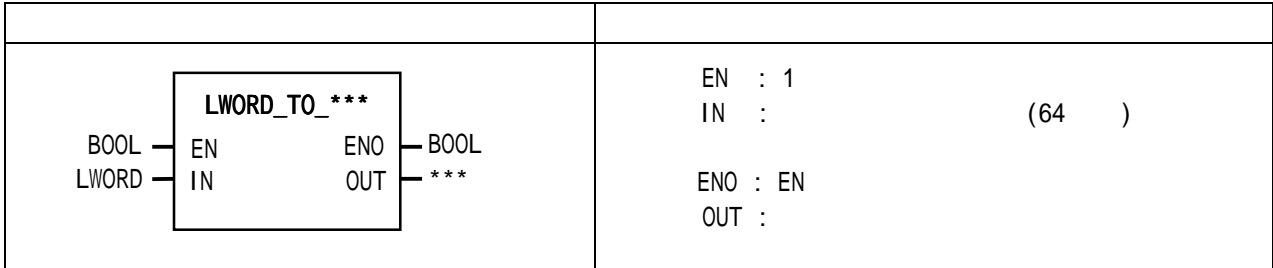
(OUT) : %Q0.0.1(BOOL) = 1(16#1)

1

LWORD_TO_***

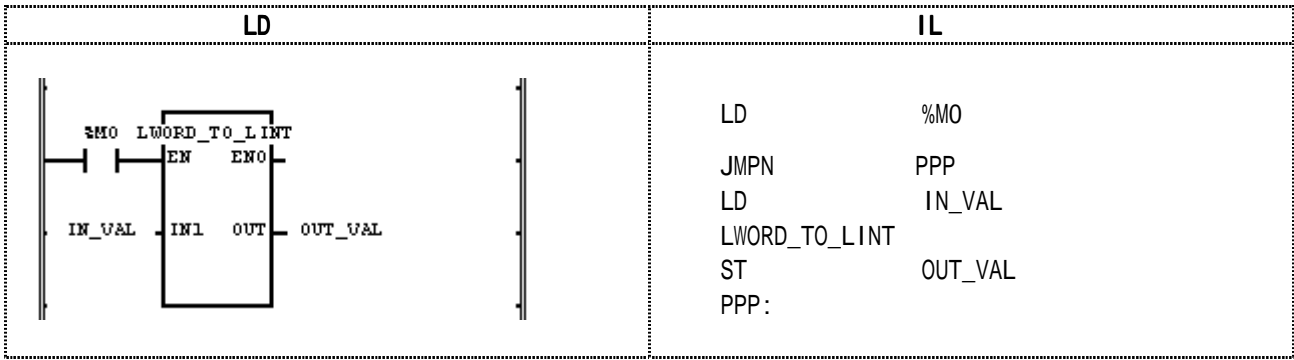
LWORD

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT .

LWORD_TO_SINT	SINT	8	SINT	.
LWORD_TO_INT	INT	16	INT	.
LWORD_TO_DINT	DINT	32	DINT	.
LWORD_TO_LINT	LINT		LINT	.
LWORD_TO_USINT	USINT	8	USINT	.
LWORD_TO_UINT	UINT	16	UINT	.
LWORD_TO_UDINT	UDINT	32	UDINT	.
LWORD_TO_ULINT	ULINT		ULINT	.
LWORD_TO_BOOL	BOOL	1	BOOL	.
LWORD_TO_BYTE	BYTE	8	BYTE	.
LWORD_TO_WORD	WORD	16	WORD	.
LWORD_TO_DWORD	DWORD	32	DWORD	.
LWORD_TO_LREAL	LREAL	LWORD	LREAL	.
LWORD_TO_DT	DT		DT	.
LWORD_TO_STRING	STRING		STRING	.



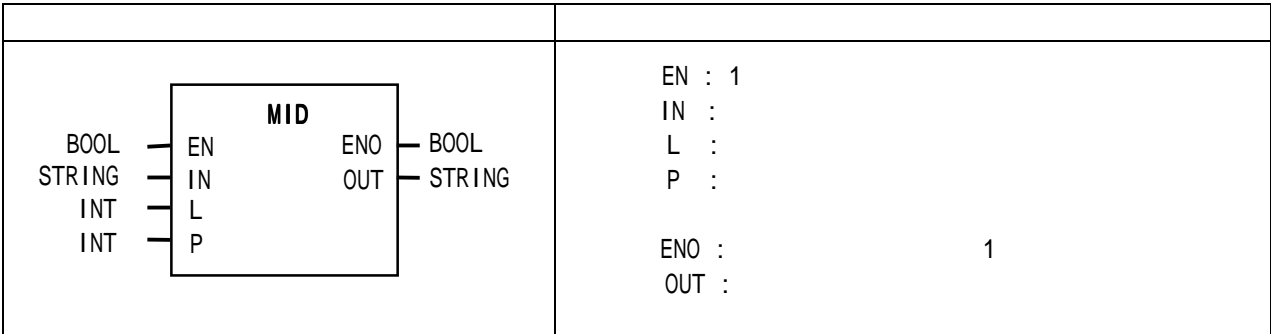
- (1) (%M0) On LWORD_TO_LINT .
- (2) IN_VAL(LWORD) = 16#FFFFFFFFFFFFFFF ,
 OUT_VAL(LINT) = -1(16#FFFFFFFFFFFFFFF) .

- (IN1) : IN_VAL(LWORD) = 16#FFFFFFFFFFFFFFF

↓
 (LWORD_TO_LINT)
- (OUT) : OUT_VAL(LINT) = -1

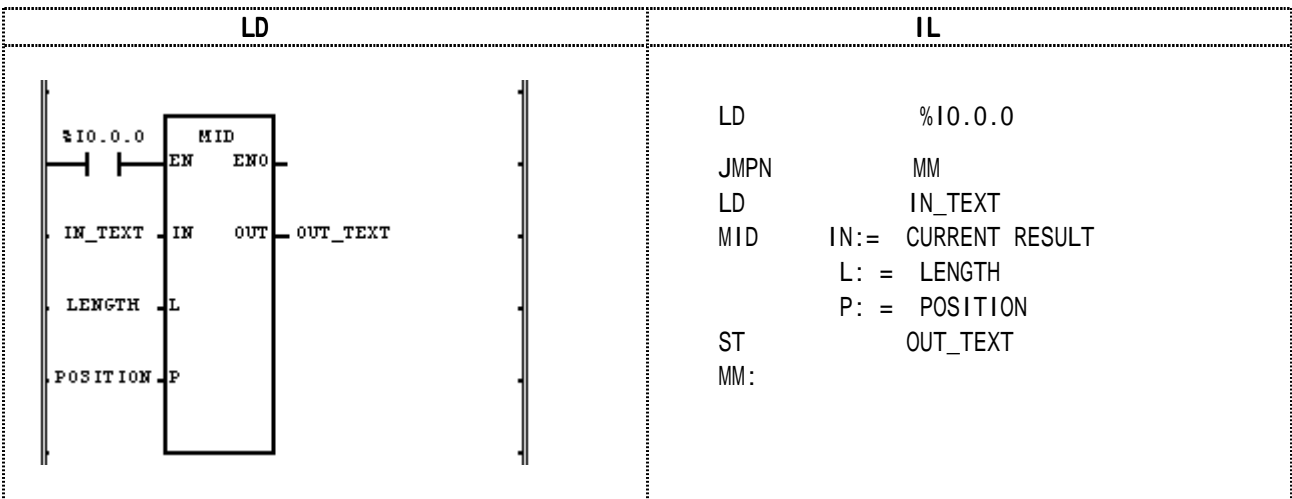
MID

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN P L OUT

(IN) < P , P <= 0 L < 0 _ERR, _LER 가 (Set)

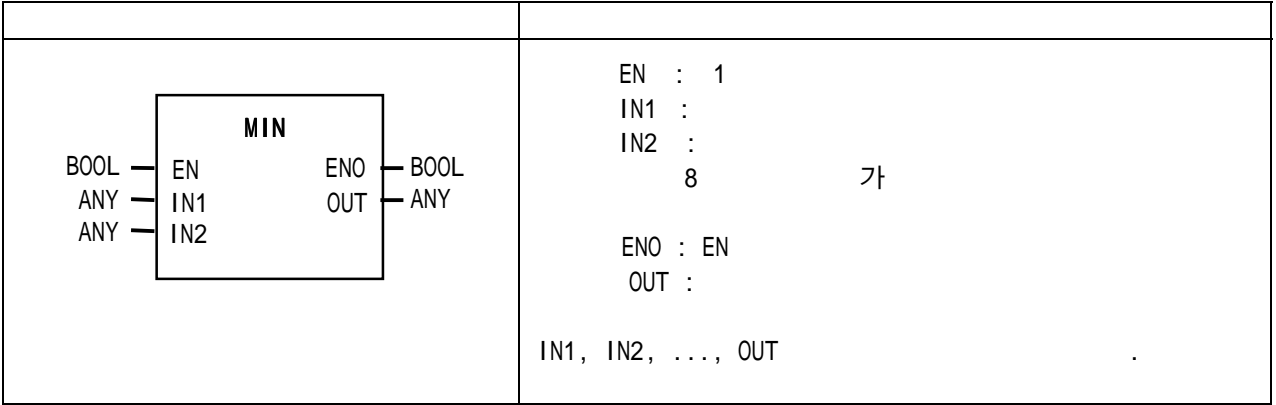


(1) (%I0.0.0) On MID()
 (2) IN_TEXT='ABCDEFG', LENGTH=3, POSITION=2, OUT_TEXT='BCD' 가

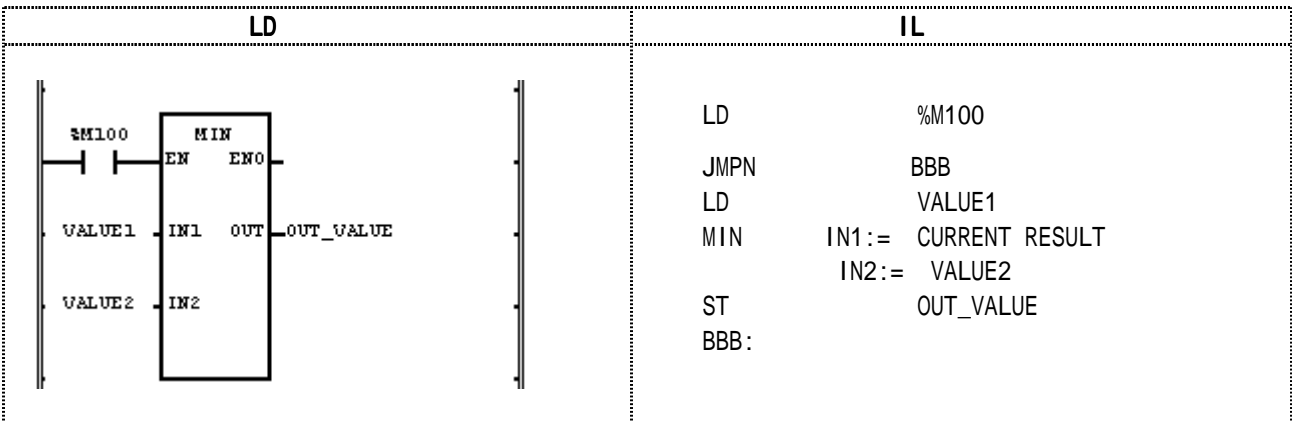
(IN) : IN_TEXT1 (STRING) = 'ABCDEFG'
 (L) : LENGTH (INT) = 3
 (P) : POSITION (INT) = 2
 ↓ (MID)
 (OUT) : OUT_TEXT = 'BCD'

MIN

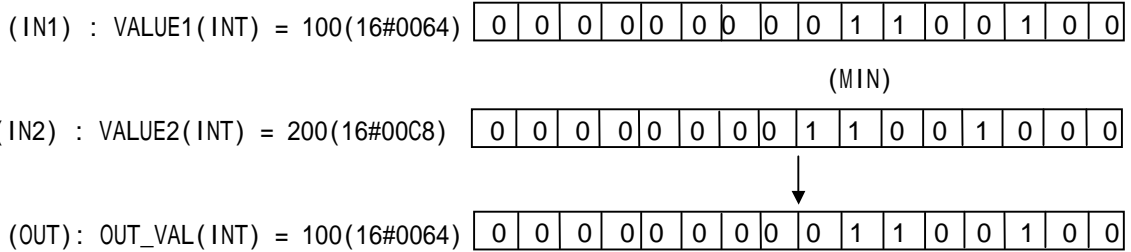
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1, IN2, ..., INn(n)) OUT .

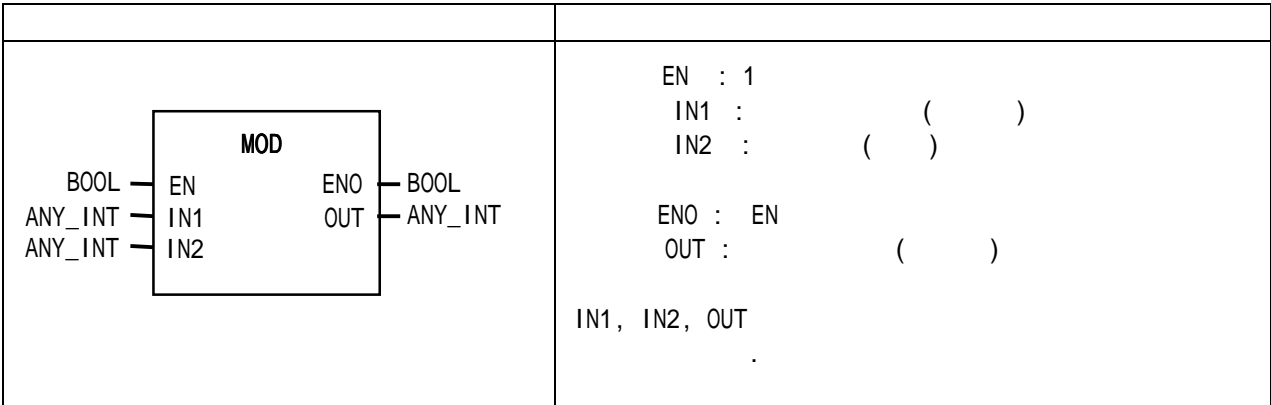


(1) (%M100) On MIN()
 (2) VALUE1 = 100, VALUE2 = 200 100
 OUT_VALUE = 100 .



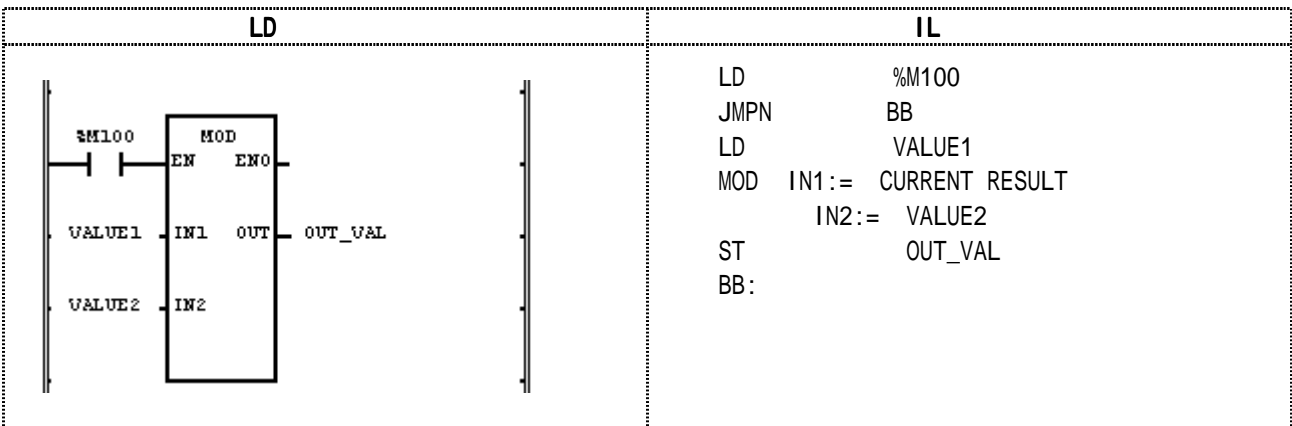
MOD

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



$$OUT = IN1 - (IN1/IN2) \times IN2 \quad (IN2 = 0 \quad OUT = 0)$$

IN1	IN2	OUT
7	2	1
7	-2	1
-7	2	-1
-7	-2	-1
7	0	0



(1) (%M100) On MON()

(2) VALUE1=37, VALUE2=10, OUT_VAL

37 10 7

(IN1) : VALUE1(INT) = 37(16#0025) 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1

(MOD)

(IN2) : VALUE2(INT) = 10(16#000A) 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

↓

(OUT) : OUT_VAL(INT) = 7(16#0007) 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

MUL



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<p>EN : 1 IN : () IN2 : () 가 8 가</p> <p>ENO : 1 OUT :</p> <p>IN1, IN2, ..., OUT</p>
--	---

IN1, IN2, ..., INn (n) OUT .
 OUT = IN1 × IN2 × ... × INn

_ERR, _LER 가 (Set) .

LD	IL
	<pre>LD %M0 JMPN ABC LD VALUE1 MUL IN1:= CURRENT RESULT IN2:= VALUE2 IN3:= VALUE3 ST OUT_VAL ABC:</pre>

(1) (%M0) On MUL() On .
 (2) MUL VALUE1 = 30, VALUE2 = 20, VALUE3 = 10 ,
 OUT_VAL = 30 × 20 × 10 = 6000 .

(IN1) : VALUE1(INT) = 30(16#001E) 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0

+ (MUL)

(IN2) : VALUE2(INT) = 20(16#0014) 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0

+ (MUL)

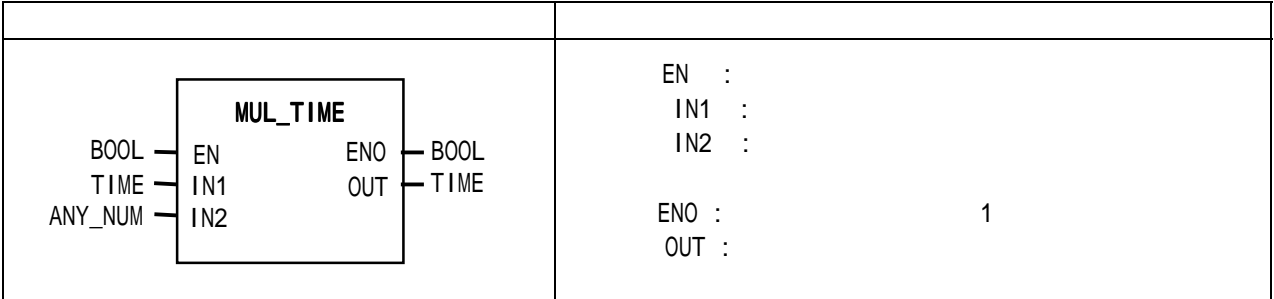
(IN3) : VALUE3(INT) = 10(16#000A) 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0



(OUT) : OUT_VAL(INT) = 6000(16#1770) 0 0 0 1 0 1 1 1 0 1 1 1 0 0 0 0

MUL_TIME

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN :
 IN1 :
 IN2 :
 ENO : 1
 OUT :

IN1() IN2()

OUT

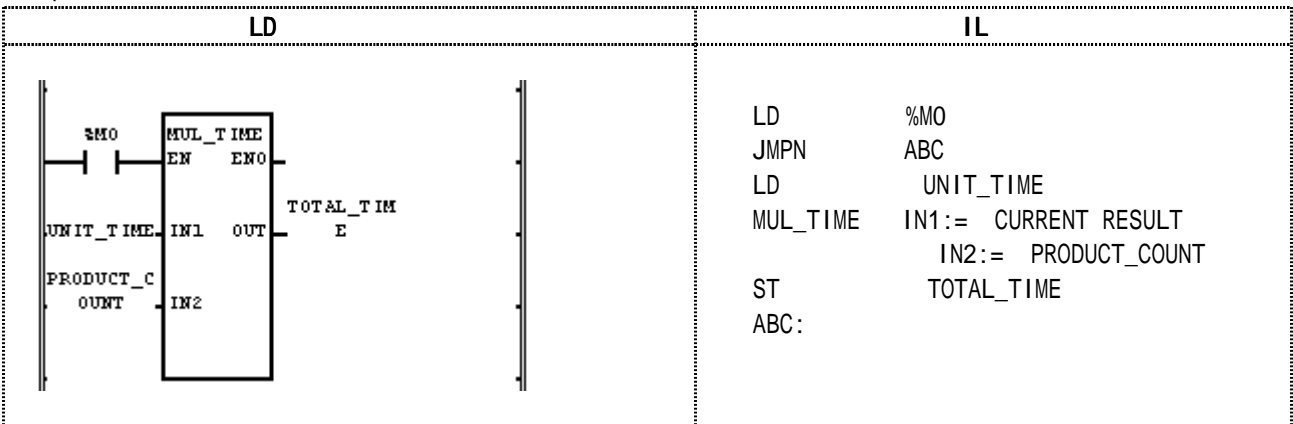
TIME

, _ERR, _LER 가 (Set)

LINE

20 2 ,

가 20



- (1) (IN1:) UNIT_TIME:T#20M2S
- (2) (IN2:) PRODUCT_COUNT:20
- (3) (OUT:) TOTAL_TIME
- (4) (% M0) On TOTAL_TIME T#6H40M40S가

(IN1) : UNIT_TIME(TIME) = T#20MS2S (MUL_TIME)

(IN2) : PRODUCT_COUNT(INT) = 16#18

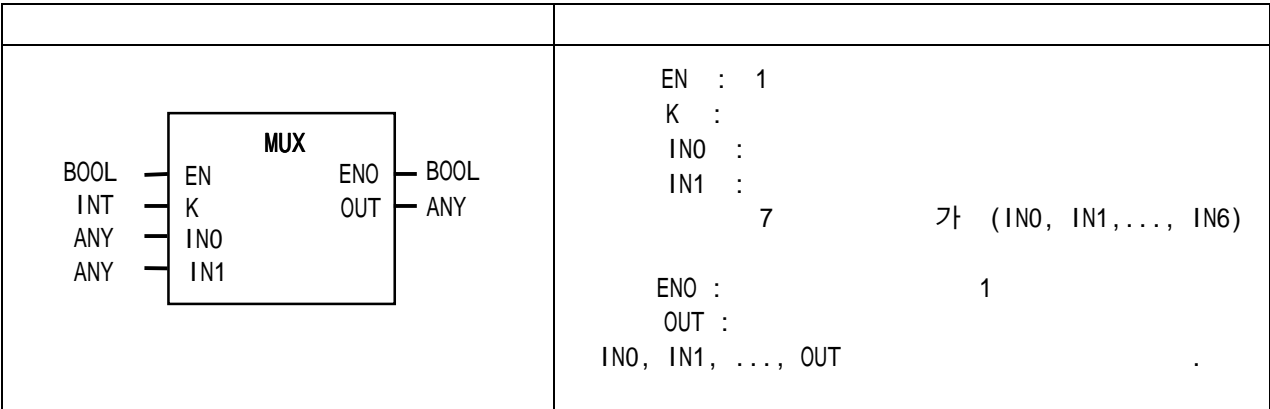
↓

(OUT) : TOTAL_TIME(TIME) = T#6H40M40S

MUX

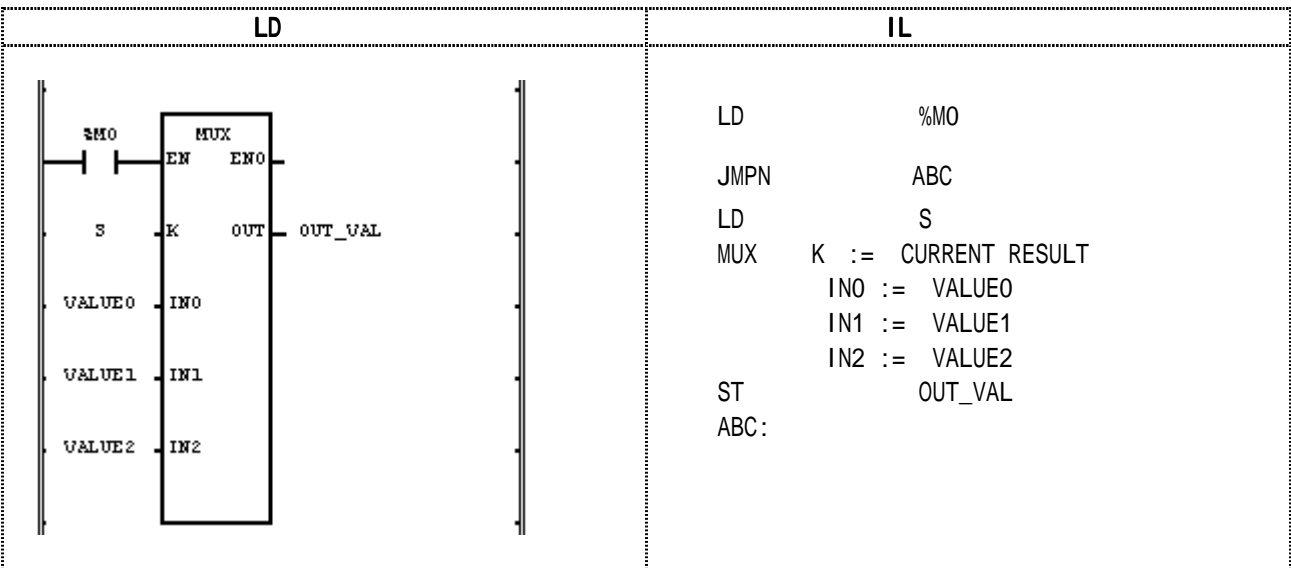


CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



K (IN0, IN1, ..., INn) .
 K = 0 IN0 , K = 1 IN1 , K = n INn OUT .

K INn OUT IN0 , _ERR, _LER
 가 (Set) .



(1) (%MO) On MUX() S .
 (2) VALUE0, 1, 2 OUT

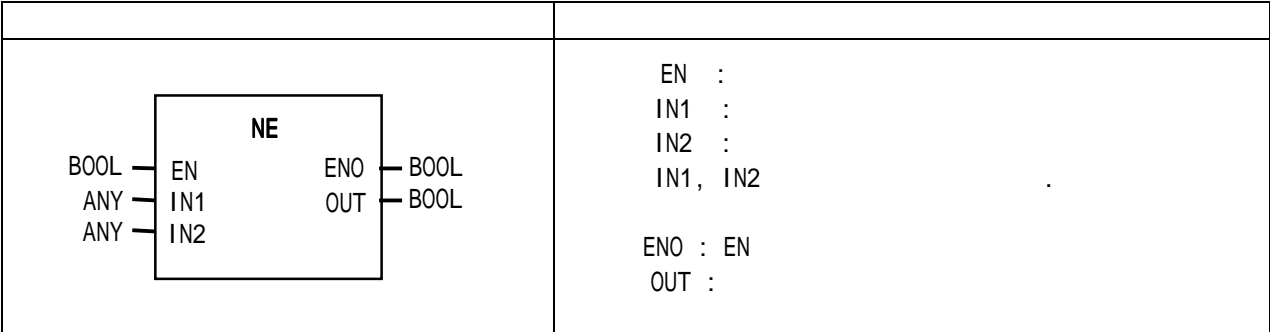
```

(K) : S(INT) = 2
(IN0) : VALUE0(WORD) = 16#11
(IN1) : VALUE1(WORD) = 16#22
(IN2) : VALUE2(WORD) = 16#33
      ↓ (MUX)
(OUT) : OUT_VAL(WORD) = 16#33
    
```

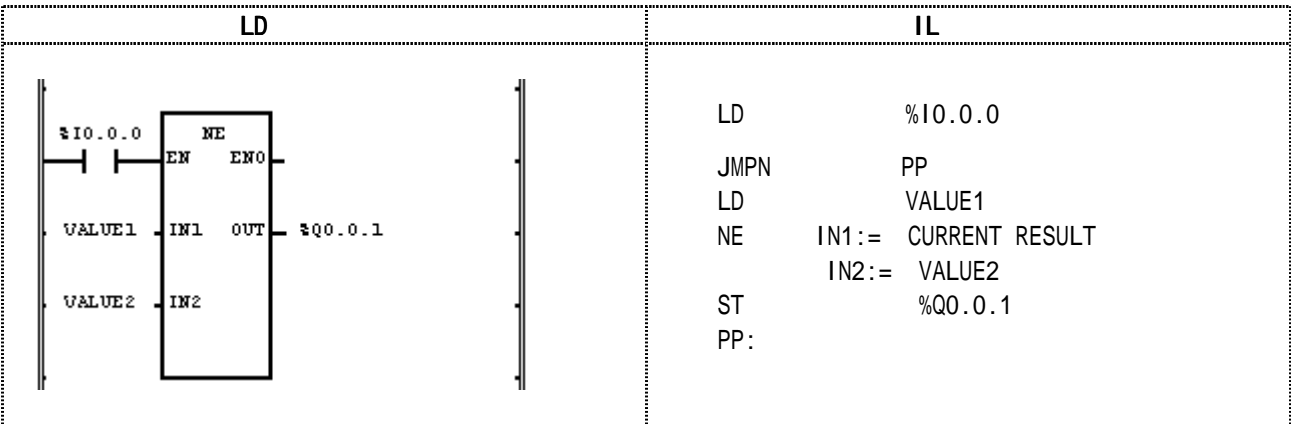

NE

· ,

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1 IN2 OUT 1
 OUT 0



(1) (%I0.0.0) On NE(:)
 (2) VALUE1 = 300, VALUE2 = 200 , VALUE1 VALUE2가
 %Q0.0.1= 1

(IN1) : VALUE1(INT) = 300(16#012C)

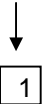
0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(NE)

(IN2) : VALUE2(INT) = 200(16#0C8)

0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

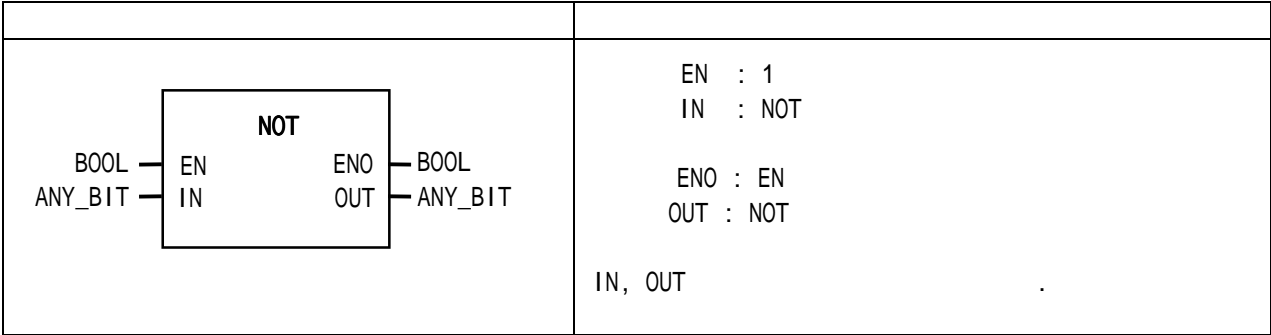
(OUT) : %Q0.0.1(BOOL) = 1(16#1)



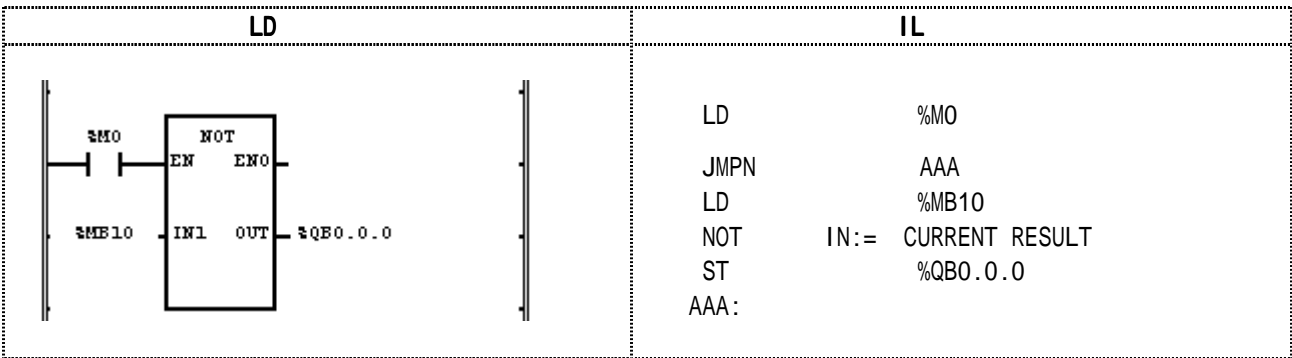
NOT



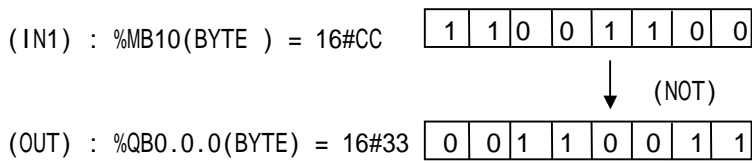
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN NOT() OUT
 IN 1100 1010
 OUT 0011 0101

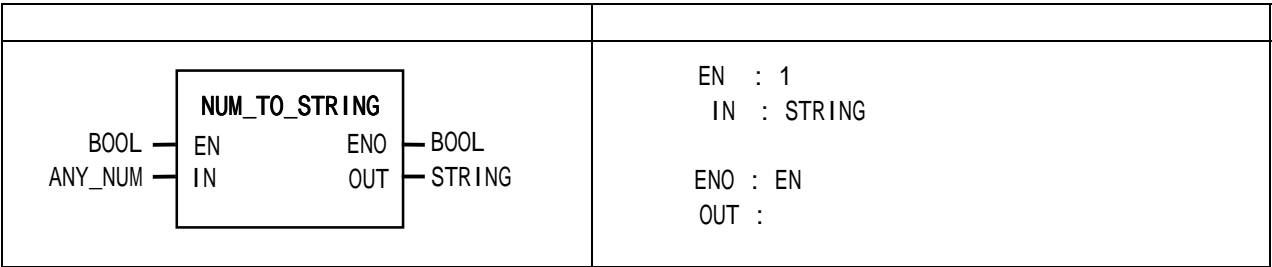


(1) (%M0) On NOT()
 (2)NOT %MB10
 %QB0.0.0.0



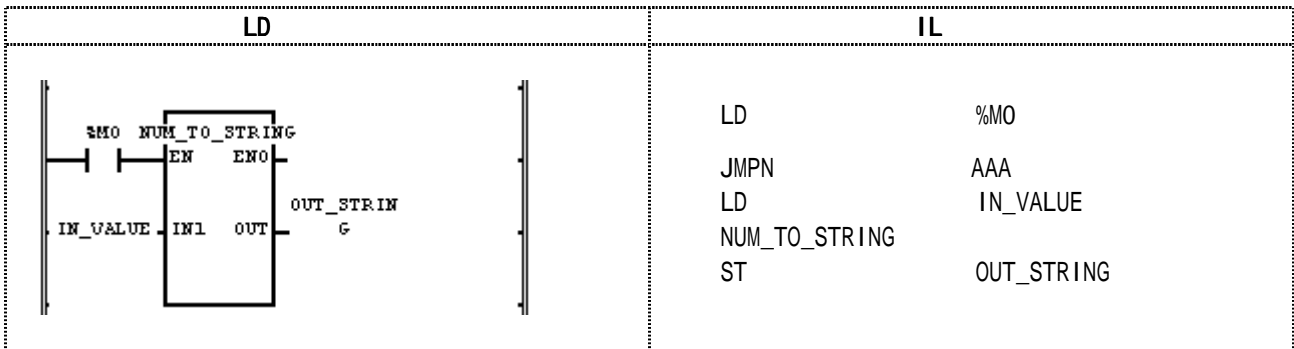
NUM_TO_STRING

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN

OUT



(1) (%MO) On NUM_TO_STRING()

(2) NUM_TO_STRING IN_VALUE(INT) = 123

OUT_STRING = '123' , IN_VALUE(REAL) = 123.0 OUT_STRING = '1.23E2'가

(IN1) : IN_VALUE(INT) = 123

↓ (NUM_TO_STRING)

(OUT) : OUT_STRING(STRING) = '123'

OR



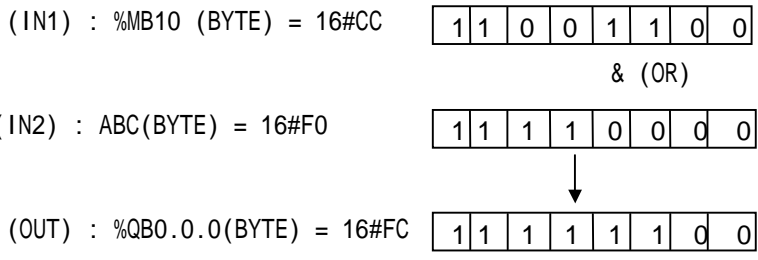
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<pre> EN : 1 IN1 : OR IN2 : OR 8 가 ENO : EN OUT : OR IN1, IN2, OUT </pre>
--	--

IN1	IN2	OR	OUT	
IN1	1111.....	0000		
OR				
IN2	1010.....	1010		
OUT	1111.....	1010		

<p>LD</p>	<p>IL</p> <pre> LD %M0 JMPN AAA LD %M10 OR IN1:= CURRENT RESULT IN2:= ABC ST %QB0.0.0 </pre>
------------------	---

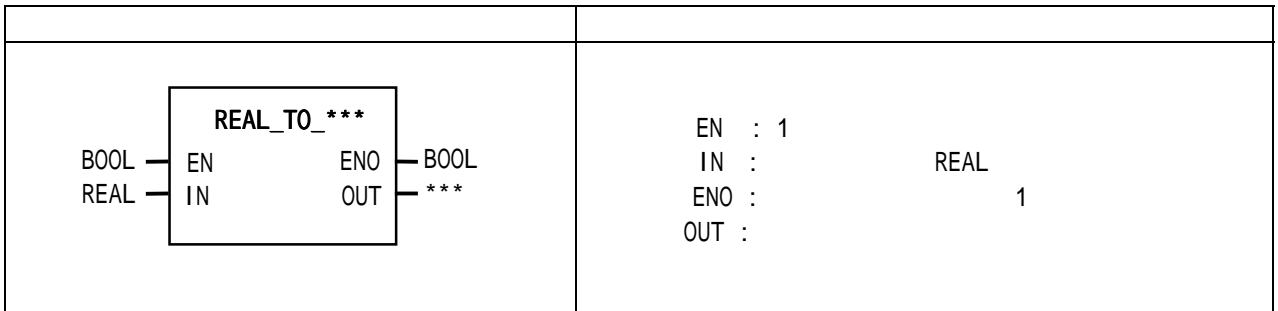
(1) (%M0) On OR
 (2) %MB10 = 11001100 ABC = 11110000 OR 가 %QB0.0.0 = 11111100



REAL_TO_***

REAL

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



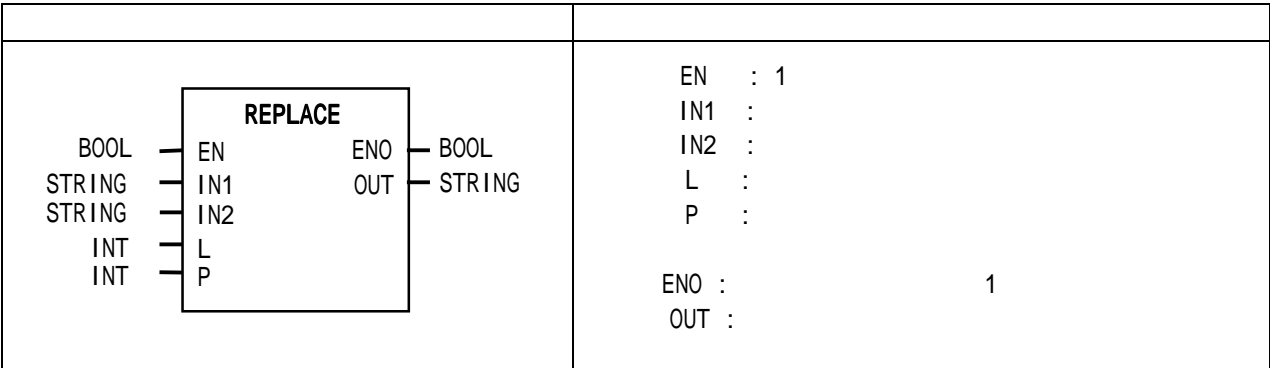
IN OUT .

REAL_TO_SINT	SINT	가 . (-128 127) ,
REAL_TO_INT	INT	가 . (-32768 32767) ,
REAL_TO_DINT	DINT	가 . (-2 ³¹ 2 ³¹ -1) ,
REAL_TO_LINT	LINT	가 . (-2 ⁶³ 2 ⁶³ -1) ,
REAL_TO_USINT	USINT	가 . (0 255) ,
REAL_TO_UINT	UINT	가 . (0 65,535) ,
REAL_TO_UDINT	UDINT	가 . (0 2 ³² -1) ,
REAL_TO_ULINT	ULINT	가 . (0 2 ⁶⁴ -1) ,
REAL_TO_DWORD	DWORD	DWORD .
REAL_TO_LREAL	LREAL	REAL LREAL .

가 (Set) . 0 . (Overflow)가 _ERR, _LER

REPLACE

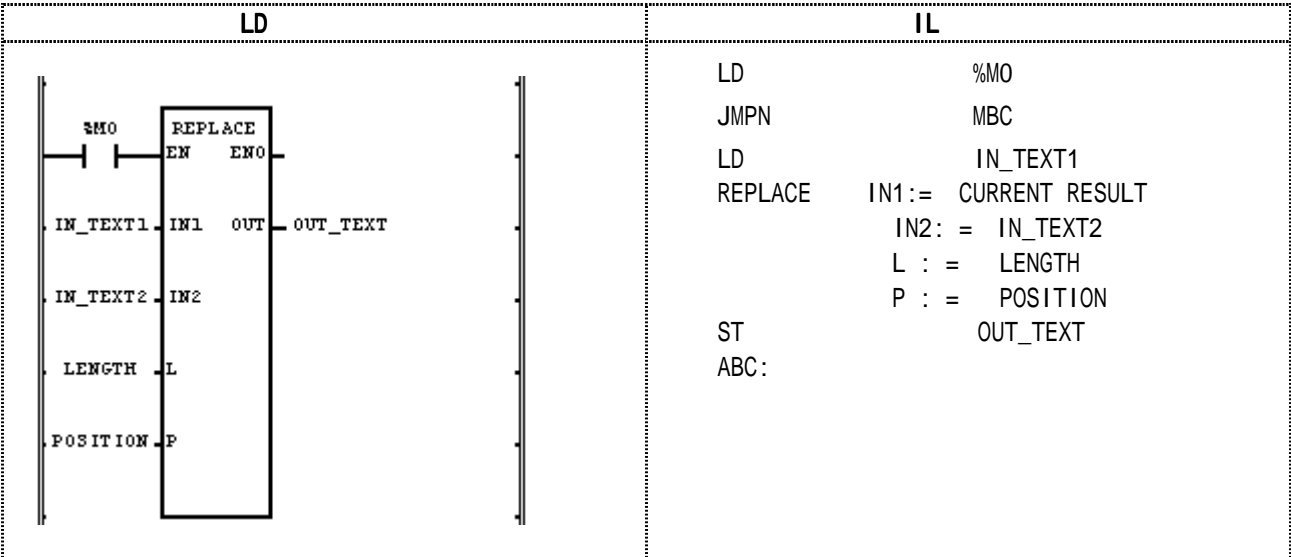
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1 P L IN2 OUT

ERR, _LER 가 (Set)

P ≤ 0 L < 0
 P > (IN1)
 > 30



```

(1)      (%M0) On REPLACE(
(2)      가 IN_TEXT1='ABCDEF' , IN_TEXT2='X' ,
          LENGTH=3, POSITION=2
IN_TEXT `BCD`가 IN_TEXT2 `X` OUT_TEXT `AXET`가

```

```
(IN1) : IN_TEXT1(STRING) = `ABCDEF`
```

```
(IN2) : IN_TEXT2(STRING) = `X`
```

```
(L) : LENGTH(INT) = 3
```

```
(P) : POSITION(INT) = 2
```

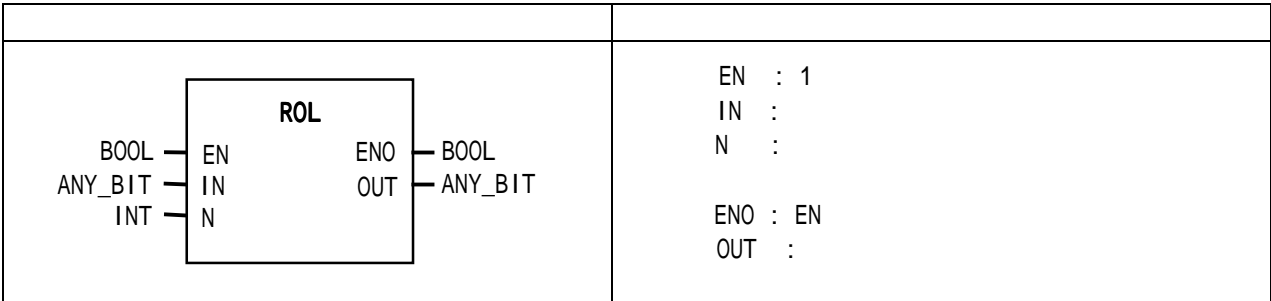
```
↓
```

```
(OUT) : OUT_TEXT(STRING) = `AXET`
```

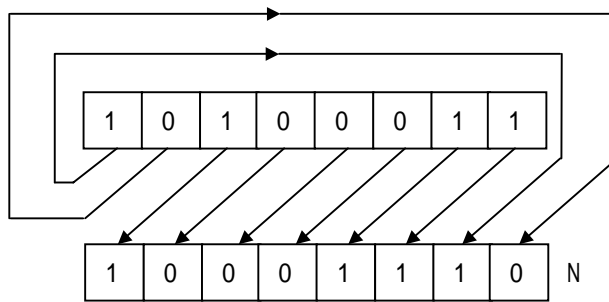

ROL

(Rotate Left)

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



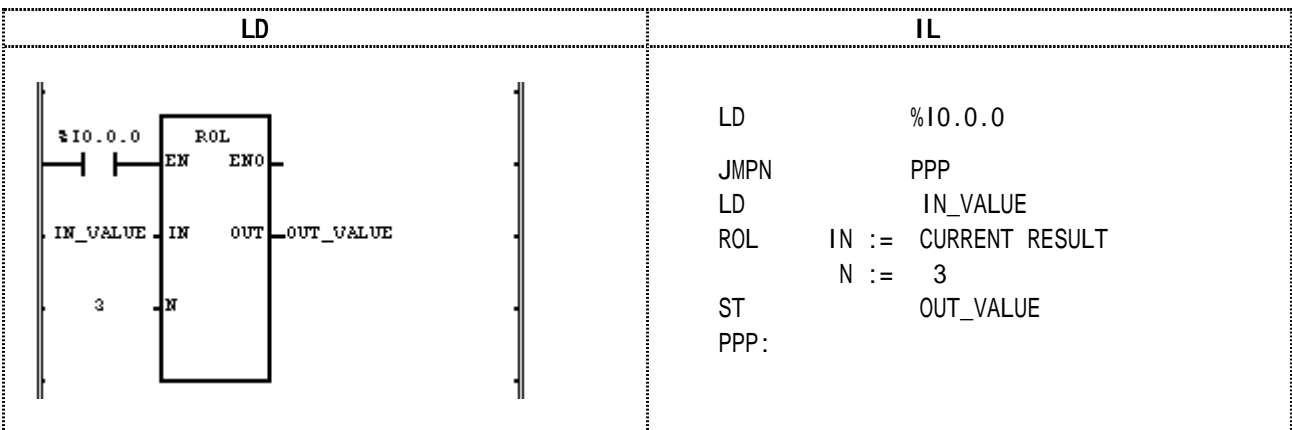
IN N



%I.0.0.0 On

(1100_1100_1100_1100:16#CCCC)

3



- (1) IN_VALUE
- (2) 3 : (N)
- (3) OUT_VALUE
- (4) %I.0.0 On ROL()

(IN1) : IN_VALUE(WORD) = 16#CCCC

(N) : 3

(OUT) : OUT_VALUE(WORD) - 16#6666

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



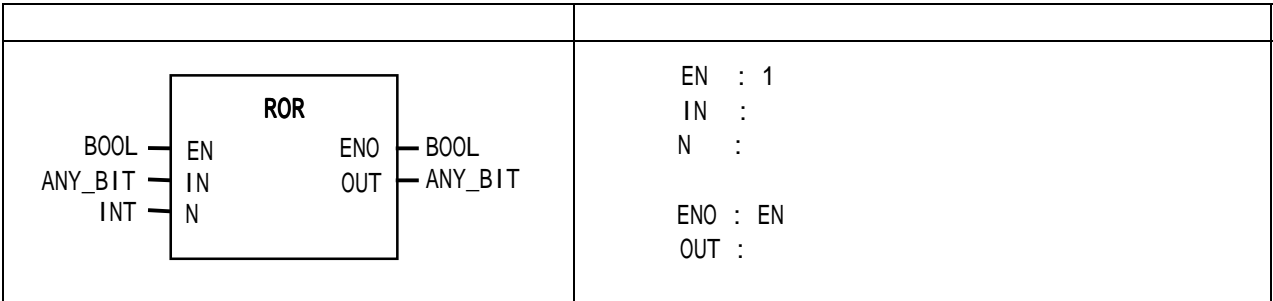
(ROL)

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

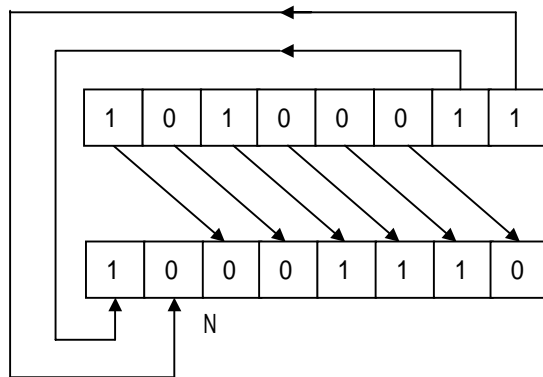
ROR

(Rotate Right)

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



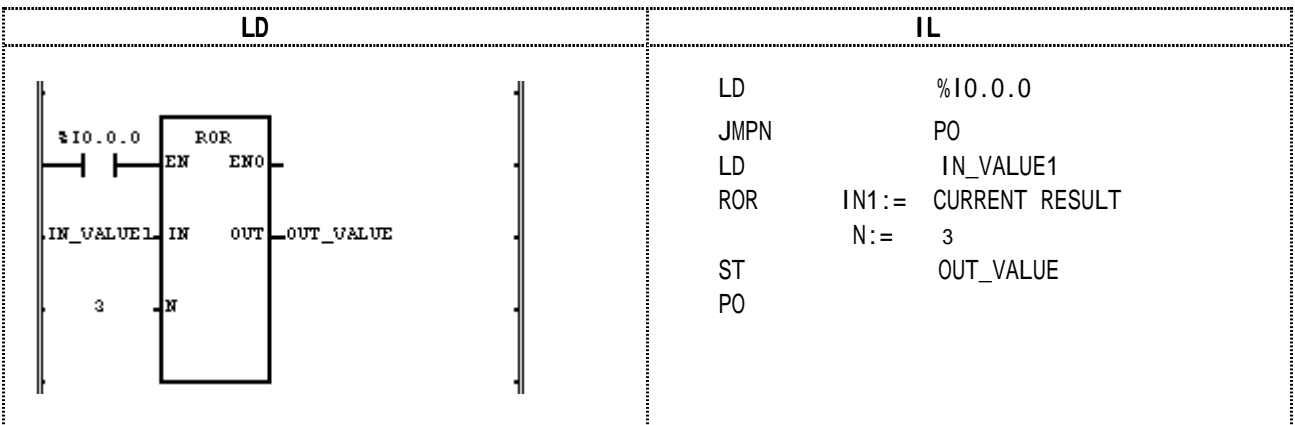
IN N



%I0.0.0 On

(1110001100110001:16#E331)

3



(1) IN_VALUE1
 (2) 3 : (N)
 (3) %I0.0.0 On ROR() 가

(IN1) : IN_VALUE1(WORD)=16#E331

(N) : 3

(OUT) : OUT_VALUE(WORD)=16#3C

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

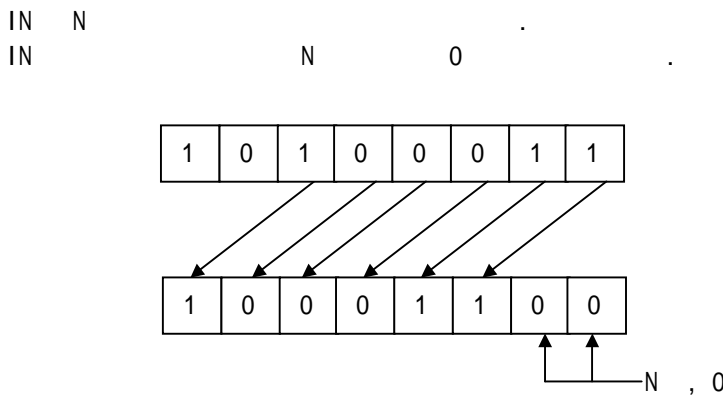
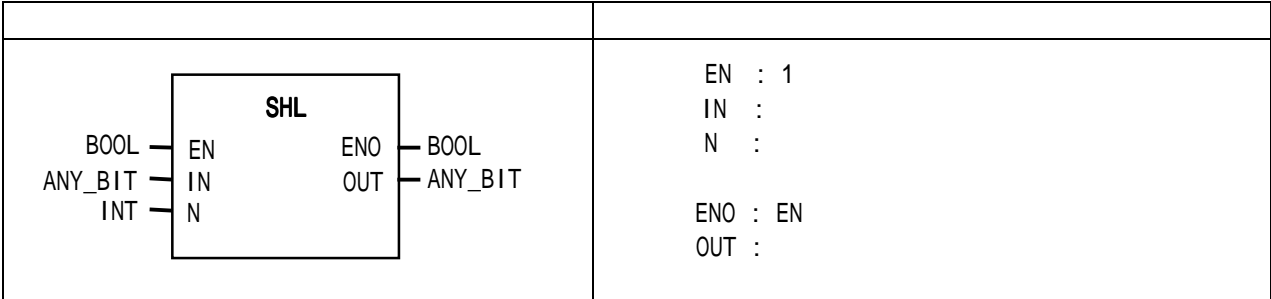


0 0 1 1 1 1 0 0 0 1 1 0 0 1 1 0

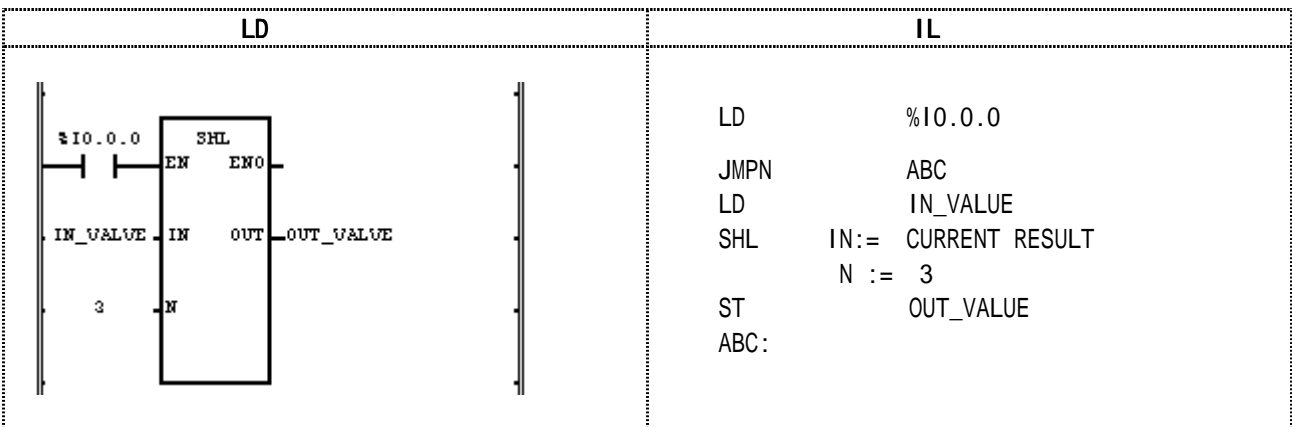
SHL

(Shift Left)

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



%I.0.0.0 On (1100_1100_1100_1100:16#CCCC) 3



(1) IN_VALUE(11001110:16#CE)
 (2) 3 (N) 가)
 (3) (%Z0.0.0) On SHL()
 가 3 , OUT_VALUE

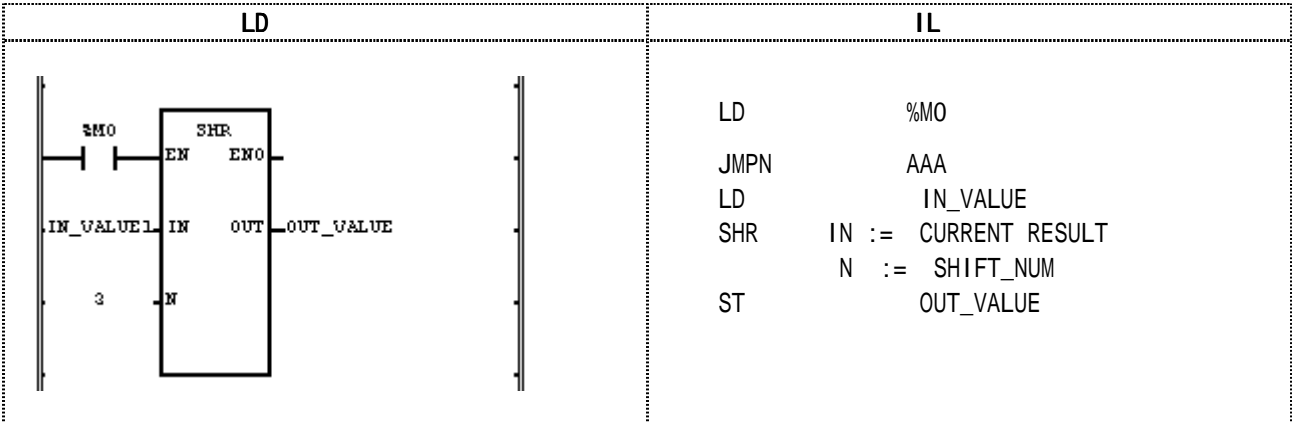
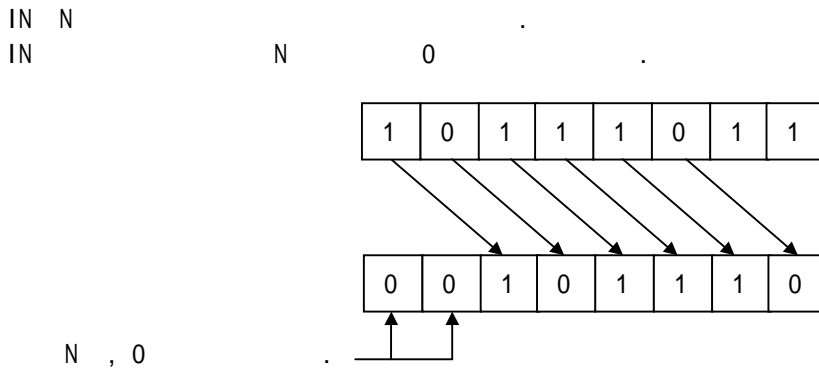
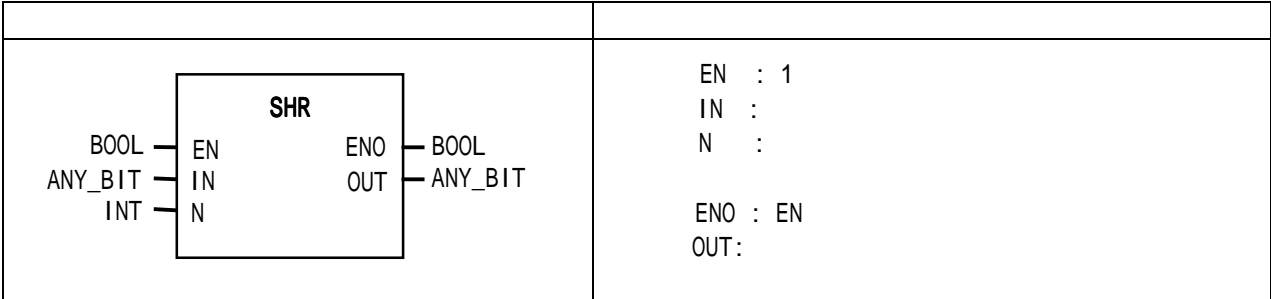
(IN1) : IN_VALUE(WORD)=16#CCCC 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
 (N) : 3
 (OUT) : OUT_VALUE(WORD)=16#6660 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0

↓ (ROL)

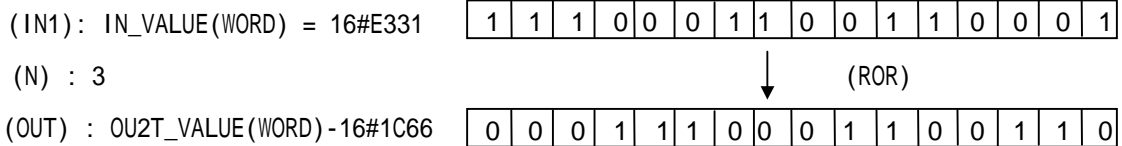
SHR

(Shift Right)

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



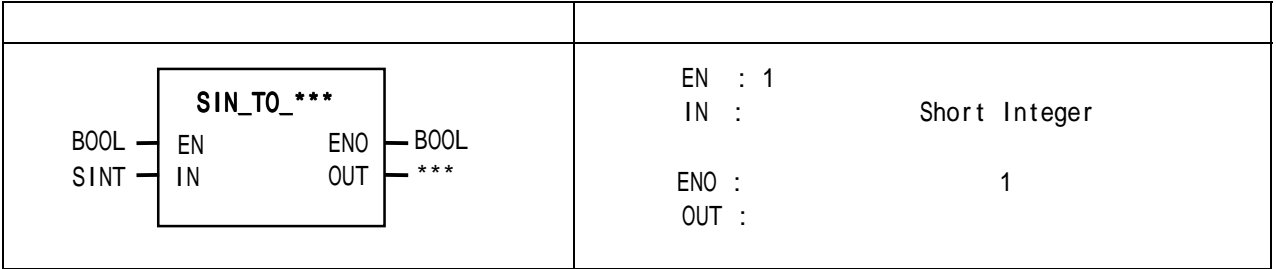
(1) (%MO) On SHL(가 3)
 (2) OUT_VALUE



SINT_TO_***

SINT

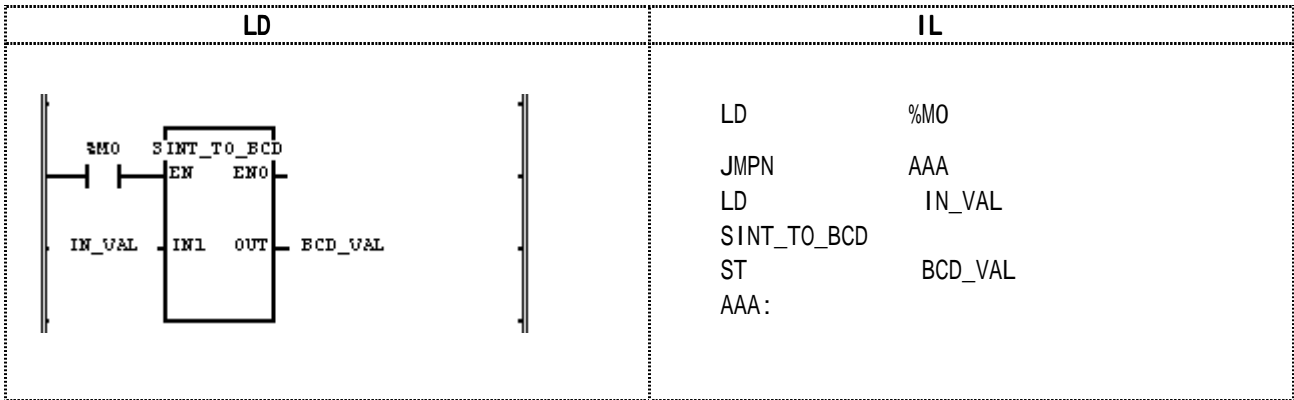
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



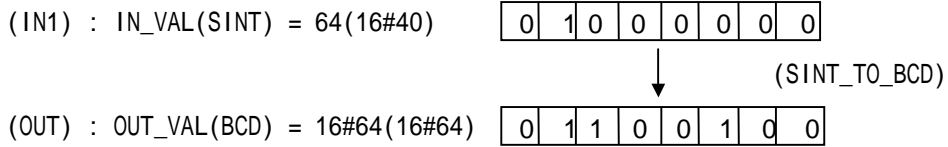
IN OUT .

SINT_TO_INT	INT	INT	.		
SINT_TO_DINT	DINT	DINT	.		
SINT_TO_LINT	LINT	LINT	.		
SINT_TO_USINT	USINT	0 127	,	가	.
SINT_TO_UINT	UINT	0 127	,	가	.
SINT_TO_UDINT	UDINT	0 127	,	가	.
SINT_TO_ULINT	ULINT	0 127	,	가	.
SINT_TO_BOOL	BOOL	1		BOOL	.
SINT_TO_BYTE	BYTE			BYTE	.
SINT_TO_WORD	WORD	0		WORD	.
SINT_TO_DWORD	DWORD	0		DWORD	.
SINT_TO_LWORD	LWORD	0		LWORD	.
SINT_TO_BCD	BYTE	0 99	,	가	.
SINT_TO_REAL	REAL	SINT REAL	.		
SINT_TO_LREAL	LREAL	SINT LREAL	.		

_ERR, _LER 가 (Set) .



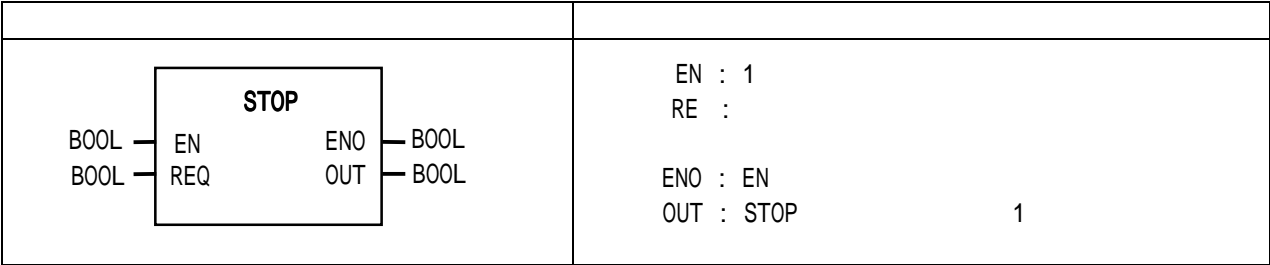
- (1) (% M0) On SINT_TO_BCD .
- (2) IN_VAL(SINT)=64(2#0100_0000) ,OUT_VAL(BCD)=16#64(2#0110_0100)가



STOP



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

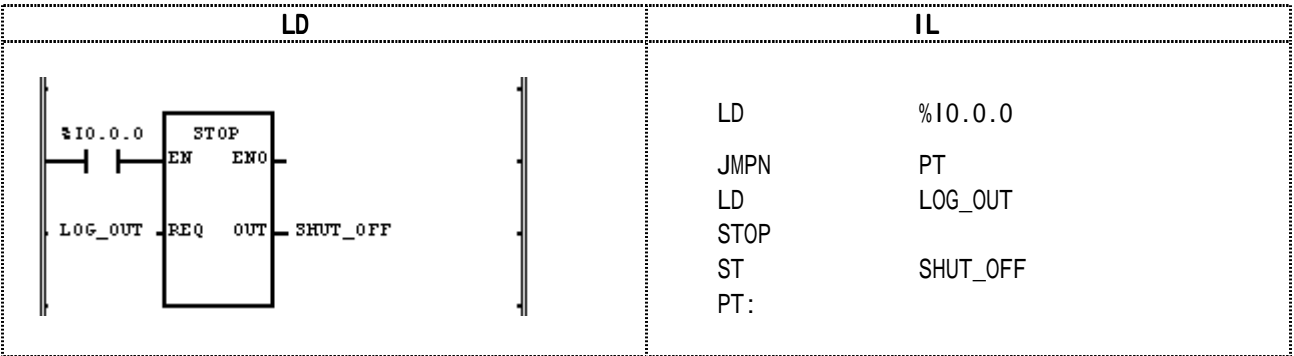


EN 1 REQ 1
'STOP'

STOP

STOP

가 RUN



(1) (%I0.0.0) On LOG_OUT가 1

STOP

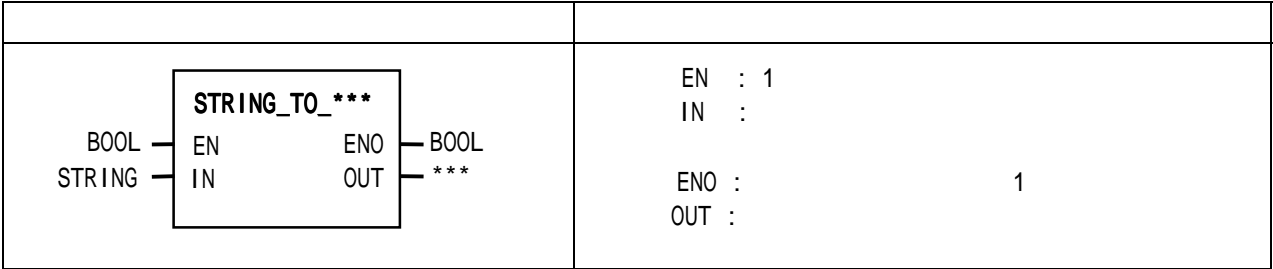
(2) 'STOP'

PLC

STRING_TO_***

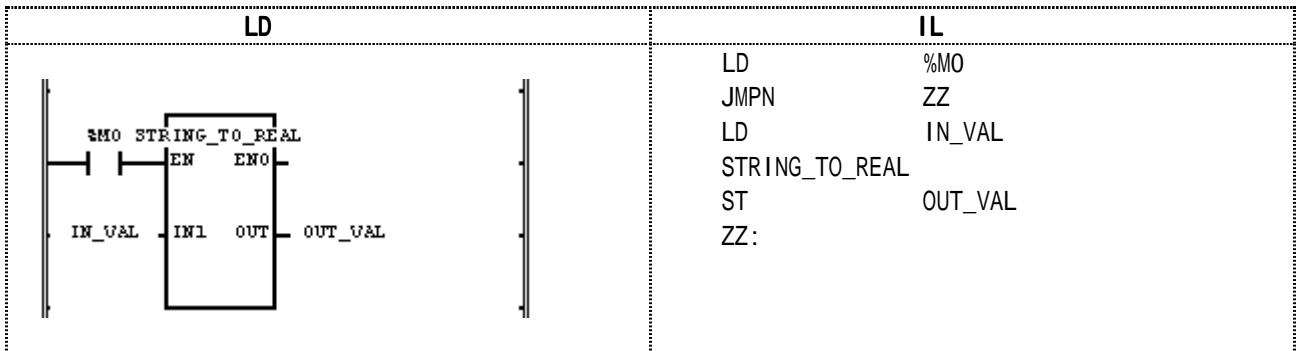
STRING

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN	OUT	
STRING_TO_SINT	SINT	STRING SINT .
STRING_TO_INT	INT	STRING INT .
STRING_TO_DINT	DINT	STRING DINT .
STRING_TO_LINT	LINT	STRING LINT .
STRING_TO_USINT	USINT	STRING USINT .
STRING_TO_UINT	UINT	STRING UINT .
STRING_TO_UDINT	UDINT	STRING UDINT .
STRING_TO_ULINT	ULINT	STRING ULINT .
STRING_TO_BOOL	BOOL	STRING BOOL .
STRING_TO_BYTE	BYTE	STRING BYTE .
STRING_TO_WORD	WORD	STRING WORD .
STRING_TO_DWORD	DWORD	STRING DWORD .
STRING_TO_LWORD	LWORD	STRING LWORD .
STRING_TO_REAL	REAL	STRING REAL .
STRING_TO_LREAL	LREAL	STRING LREAL .
STRING_TO_DT	DT	STRING DT .
STRING_TO_DATE	DATE	STRING DATE .
STRING_TO_TOD	TOD	STRING TOD .
STRING_TO_TIME	TIME	STRING TIME .

_ERR, _LER 가 (Set) .



(1) (%MO) On STRING_TO_REAL

(2) IN_VAL(STRING) = '-1.34E12' , OUT_VAL(REAL= -1.34E12

가 .

(IN1) : IN_VAL(STRING) = '-1.34E12'



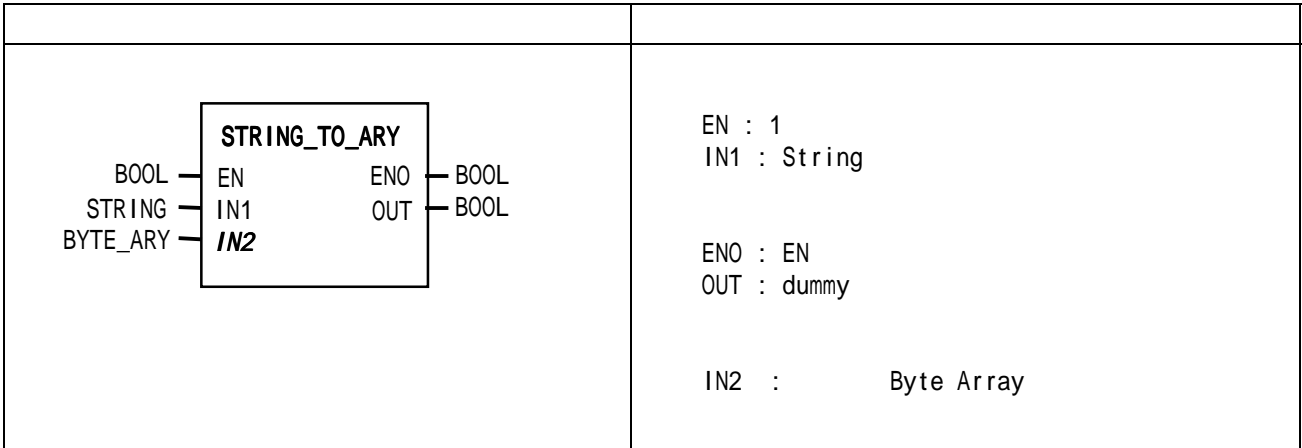
(STRING_TO_REAL)

(OUT) : OUT_VAL(REAL) = -1.34E12

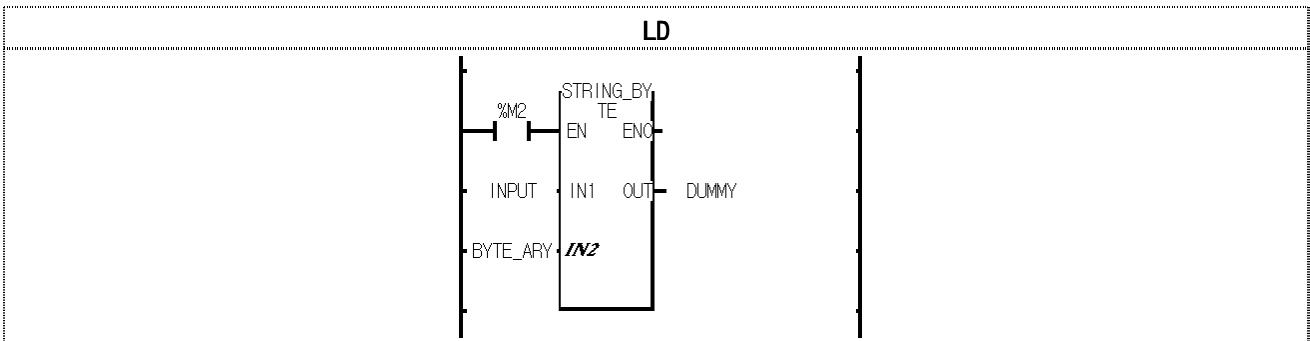
STRING_TO_ARY

Byte Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



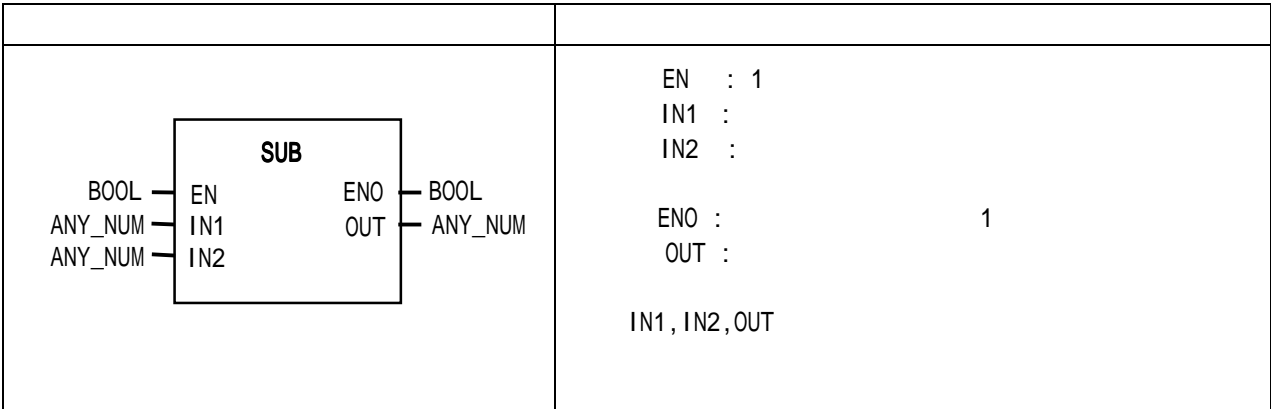
String 30 Byte Array .



- (1) (%M2) On STRING_BYTE .
 - (2) INPUT "GM4-CPUA" BYTE_ARY
- 16#{22(“), 47(G), 4D(M), 34(4), 2D(-), 43(C), 50(P), 55(U), 41(A), 22(“)}가 .

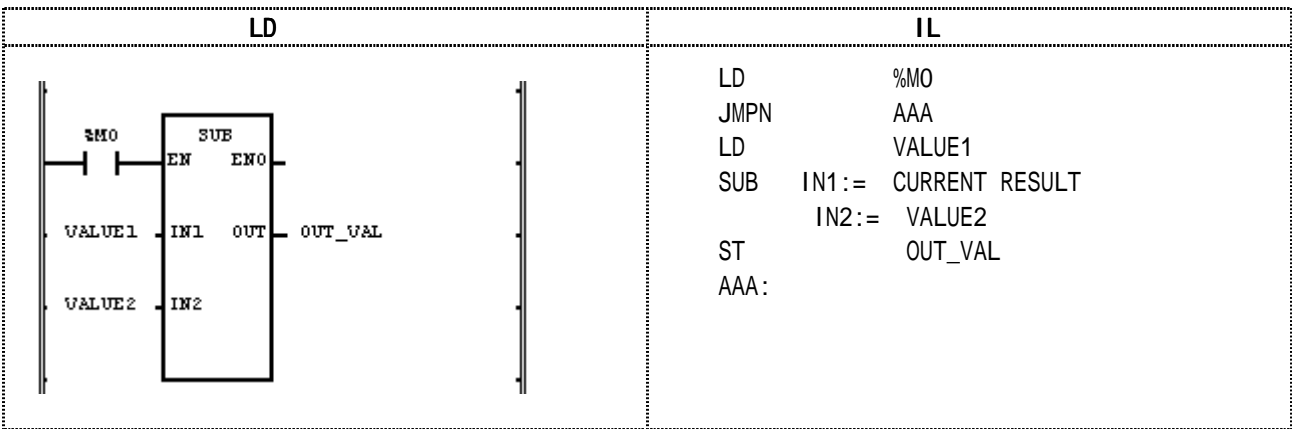
SUB

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN1 IN2 OUT
 OUT = IN1 — IN2

_ERR, _LER 가 (Set)

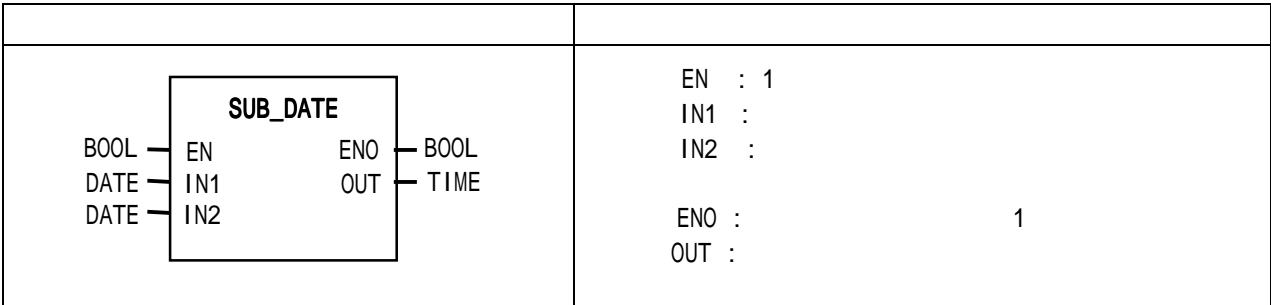


- (1) (%M0) On SUB()
- (2) VALUE1 = 300, VALUE2 = 200 , OUT_VAL
- (300-200=100)가

(IN1) : VALUE1(INT) = 300(16#012C)	0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0
- (SUB)	
(IN2) : VALUE2(INT) = 200(16#00C8)	0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
↓	
(OUT) : OUT_VAL(INT) = 100(16#0064)	0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0

SUB_DATE

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

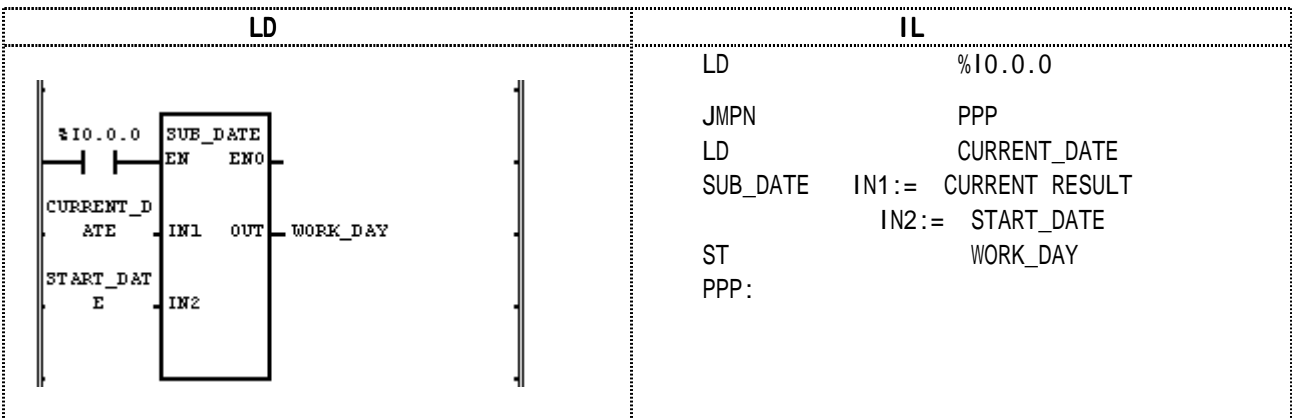


IN1() IN2() OUT .

TIME , _ERR, _LER 가 (Set) .

가 TIME T#49D17H2M47S295MS 가 가

가 .



(1) (%10.0.0) On SUB_DATE() .

(2) CURRENT_DATE가 D#1995-12-15 가

START_DATE가 D#1995-11-1 , WORK_DAY T#44D 가

(IN1) : CURRENT_DATE(DATE) = D#1995-12-15

(SUB_DATE)

(IN2) : START_DATE(DATE) = D#1995-11-1

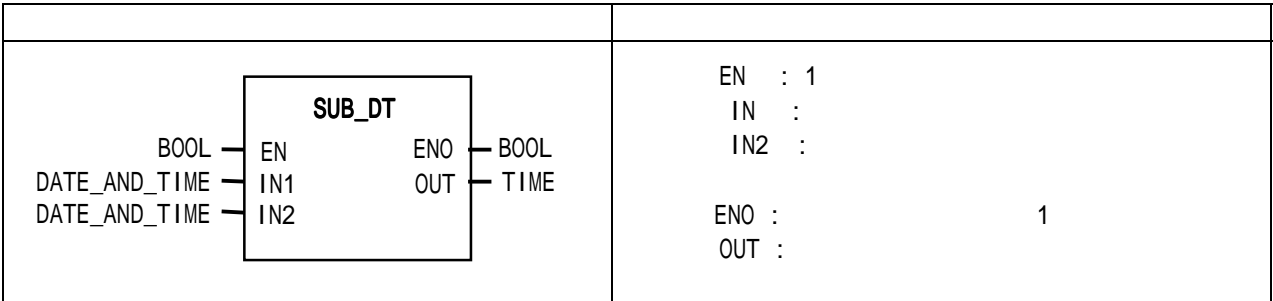
↓

(OUT) : WORK_DAY(TIME) = T#44D

SUB_DT

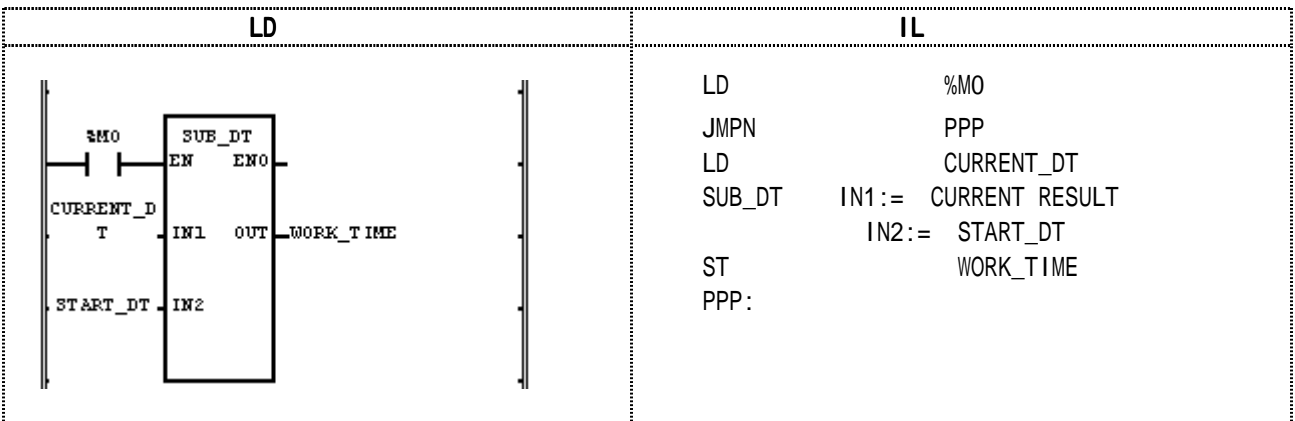


CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN : 1
 IN :
 IN2 :
 ENO : 1
 OUT :

IN1() IN2() OUT .
 TIME 가 가 , _ERR, _LER 가 (Set) .

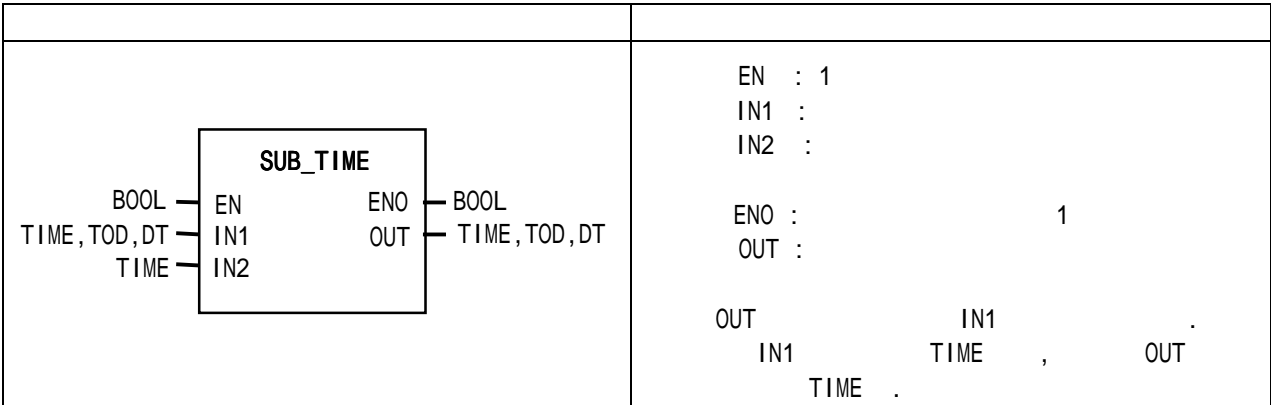


(1) (%M0) On SUB_DT()
 (2) CURRENT_DT가 DT#1995-12-15-14:30:00
 START_DT가 DT#1995-12-13-12:00:00 , WORK_TIME
 T#2D2H30M가

(IN1) : CURRENT_DT(DT) = DT#1995-12-15-14:30:00
 (SUB_DATE)
 (IN2) : START_DT(DT) = DT#1995-12-13-12:00:00
 ↓
 (OUT) : WORK_TIME(TIME) = T#2D2H30M

SUB_TIME

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



```

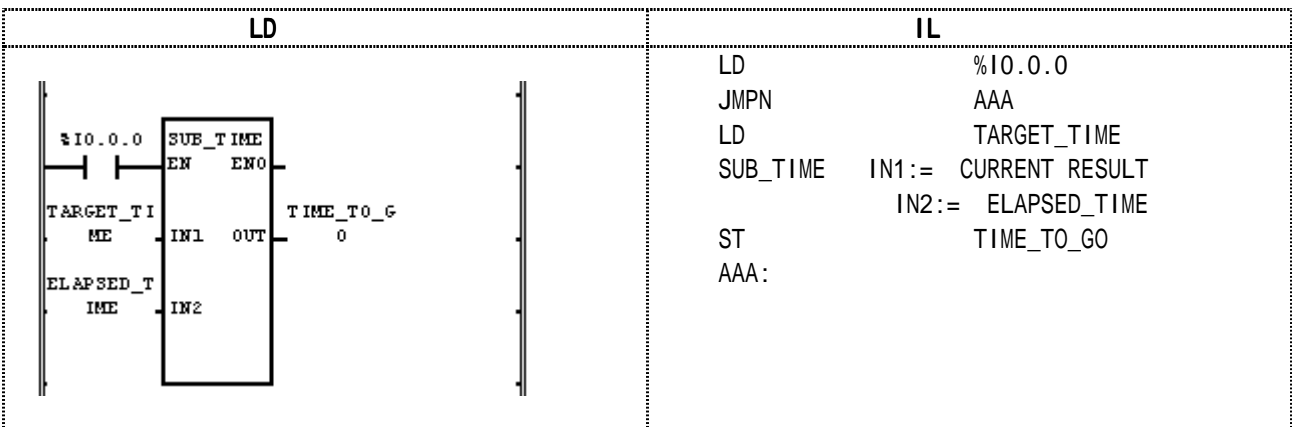
EN : 1
IN1 :
IN2 :

ENO : 1
OUT :

OUT IN1 OUT
IN1 TIME , OUT
TIME .
    
```

IN1 TIME
 IN1 TIME_OF_DAY
 IN1 DATE_AND_TIME

가 가 (TOD) , _ERR, _LER 가 (Set) 가 가 가

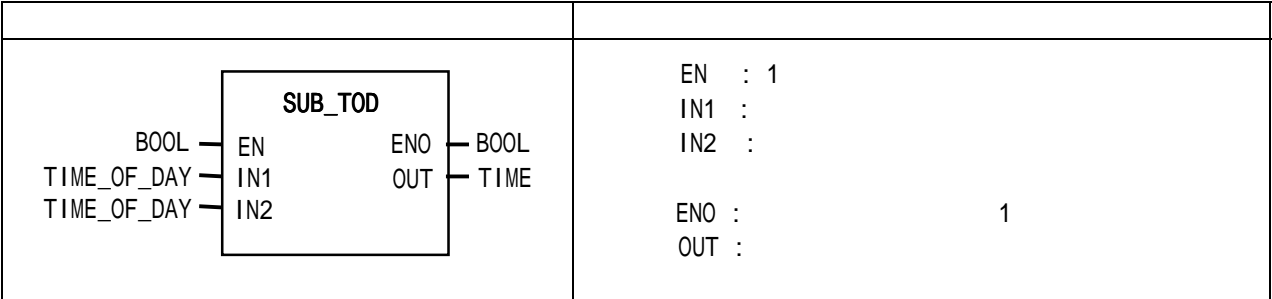


(1) (%I0.0.0) On SUB_TIME()
 (2) TARGET_TIME T#2H30M , ELAPSED_TIME T#1H10M30S300MS
 TIME_TO_GO T#1H19M29S700MS가

(IN1) : TARGET_TIME(TIME) = T#2H30M
 (SUB_TIME)
 (IN2) : ELAPSED_TIME(TIME) = T#1H10M30S300MS
 ↓
 (OUT) : TIME_TO_GO(TIME) = T#1H19M29S700MS

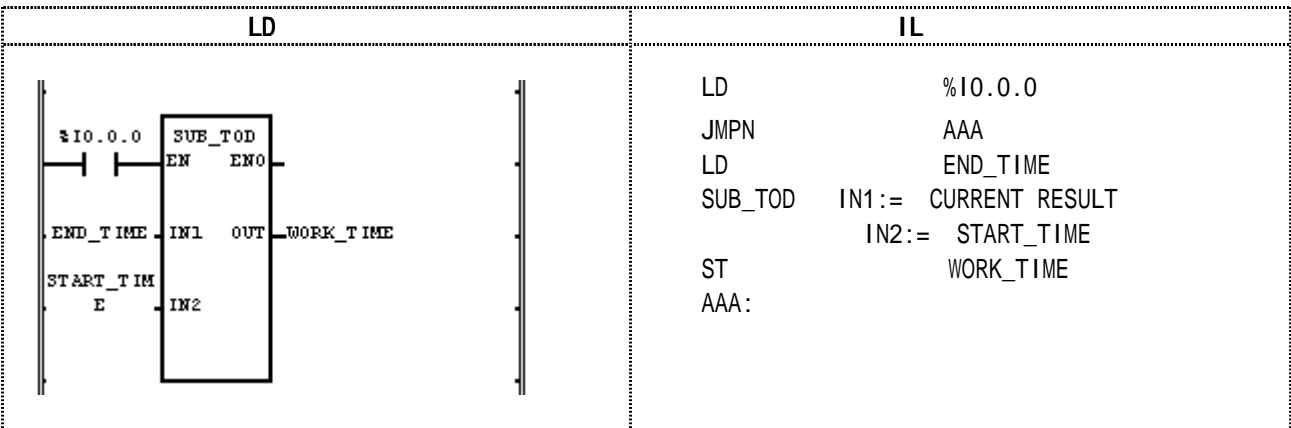
SUB_TOD

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN : 1
 IN1 :
 IN2 :
 ENO : 1
 OUT :

IN1() IN2() OUT .
 가 가 가 .



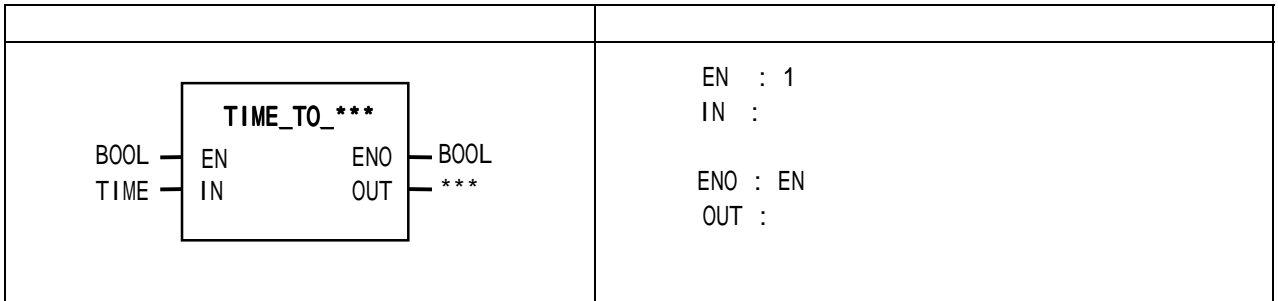
(1) (%I0.0.0) On SUB_TOD()
 (2) END_TIME TOD#14:20:30.5 START_TIME TOD#12:00:00 ,
 WORK_TIME T#2H20M30S500MS 가 .

(IN1) : END_TIME(TOD) = TOD#14:20:30.5
 (SUB_TOD)
 (IN2) : START_TIME(TOD) = TOD#12:00:00
 ↓
 (OUT) : WORK_TIME(TIME) = T#2H20M30S500MS

TIME_TO_***

TIME

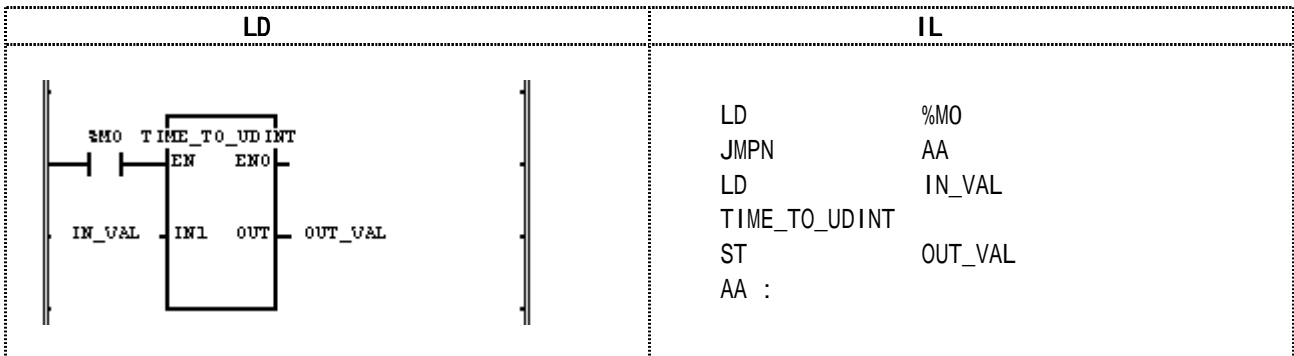
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



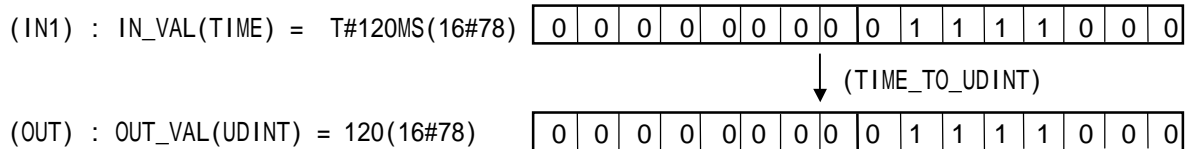
EN : 1
 IN :
 ENO : EN
 OUT :

IN OUT .

TIME_TO_UDINT	UDINT	TIME UDINT	. ()
TIME_TO_DWORD	DWORD	TIME DWORD	. ()
TIME_TO_STRING	STRING	TIME STRING	.



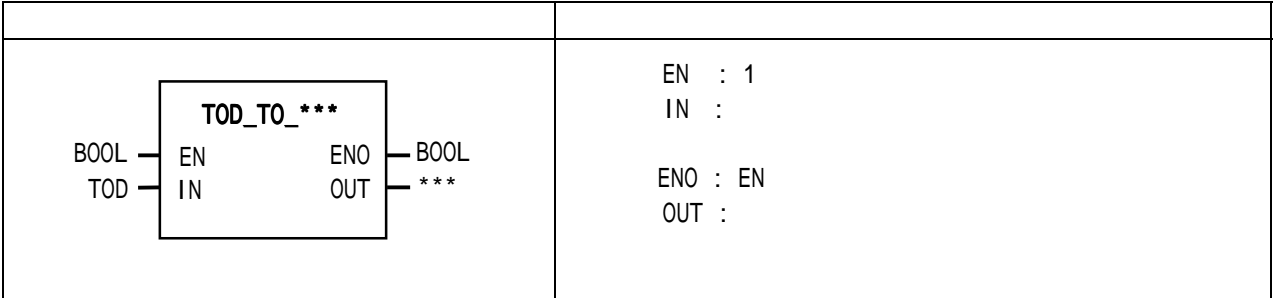
- (1) (%MO) On TIME_TO_UDINT
- (2) IN_VAL(TIME) = T#120MS , OUT_VAL(UDINT) = 120



TOD_TO_***

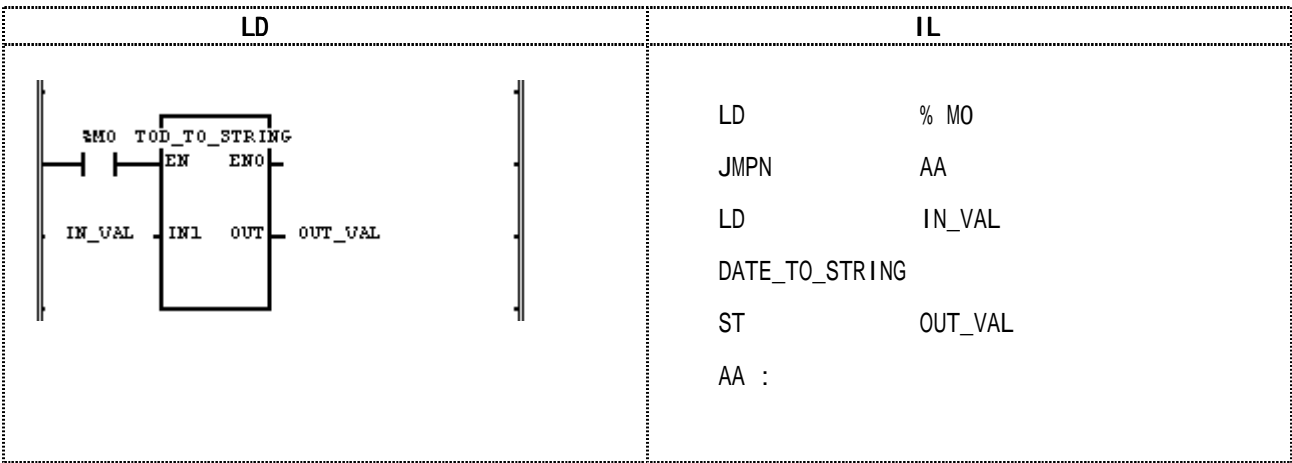
TOD

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT .

TOD_TO_UDINT	UDINT	TOD UDINT ()	.
TOD_TO_DWORD	DWORD	TOD DWORD ()	.
TOD_TO_STRING	STRING	TOD STRING	.



(1) (%MO) On TOD_TO_STRING .

(2) IN_VAL(TOD) = TOD#12:00:00 , OUT_VAL(STRING) = 'TOD#12:00:00'

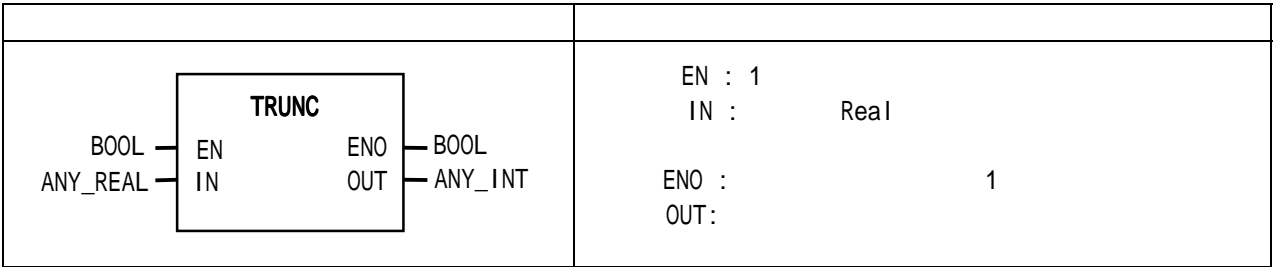
(IN1) : IN_VAL(TOD) = TOD#12:00:00

↓ (TOD_TO_STRING)

(IN2) : OUT_VAL(STRING) = 'TOD#12:00:00'

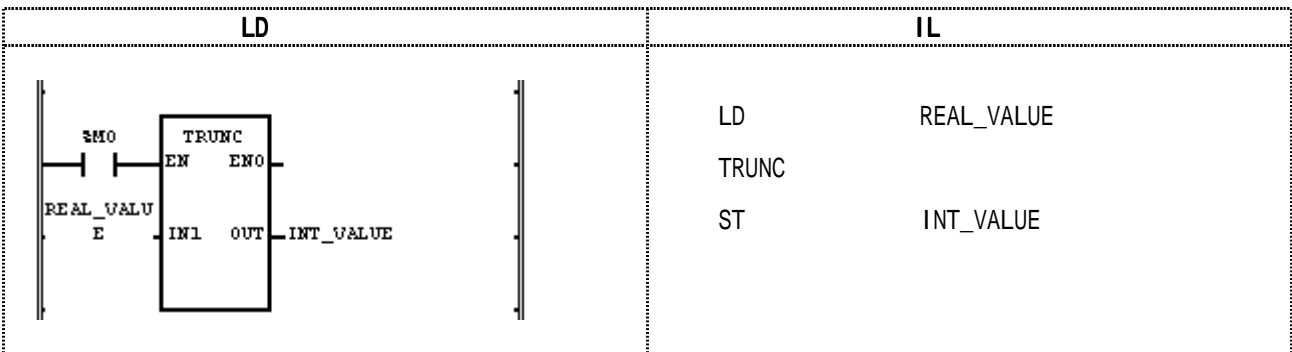
TRUNC

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



TRUNC	REAL LREAL	DINT LINT	IN OUT
-------	---------------	--------------	-----------

Integer OUT Integer OUT 0 OUT Unsigned 가 (Set) 가 (Set) _ERR, _LER



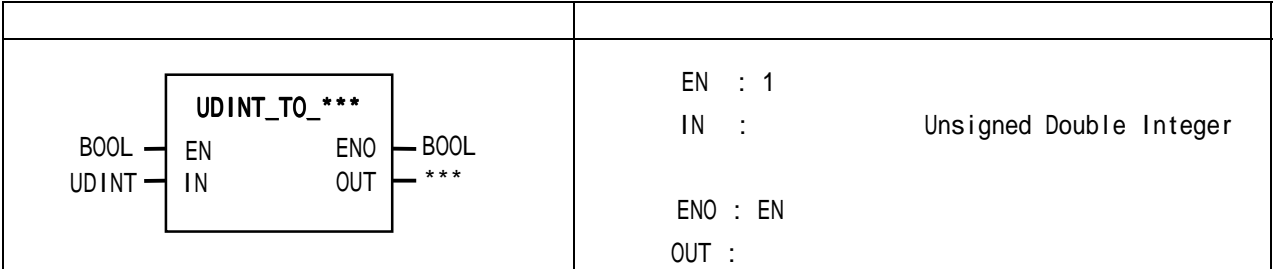
(1) (%M0) On TRUNC ()
 (2) REAL_VALUE(REAL) = 1.6 INT_VALUE(INT) = 1
 REAL_VALUE(REAL) = -1.6 INT_VALUE(INT) = -1

(IN1) : REAL_VALUE(REAL) = 1.6
 ↓ (TRUNC)
 (OUT) : INT_VALUE(INT) = 1

UDINT_TO_***

UDINT

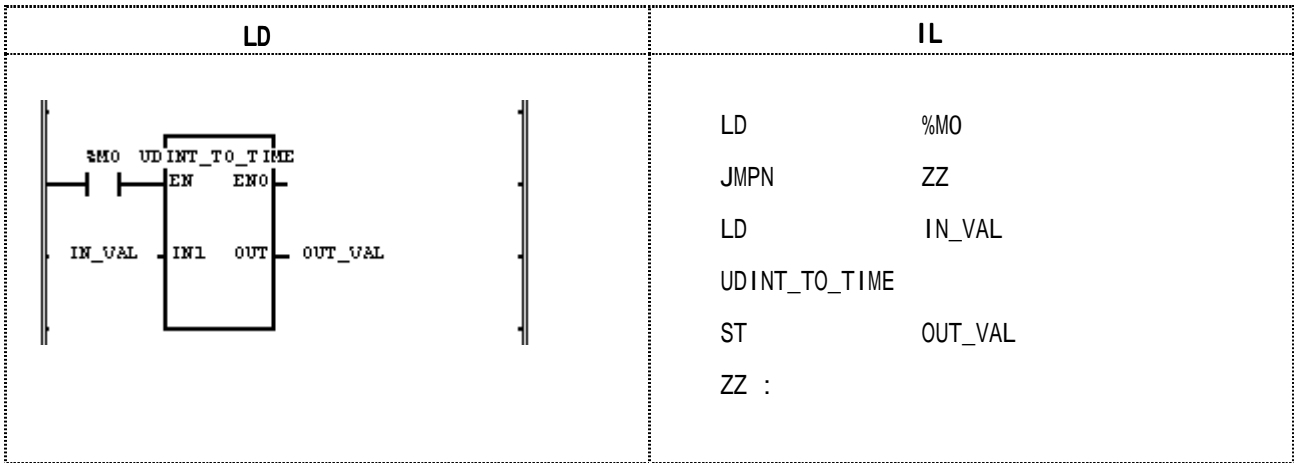
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT

UDINT_TO_SINT	SINT	0 127	가 .
UDINT_TO_INT	INT	0 32767	가 .
UDINT_TO_DINT	DINT	0 2,147,483,64	가 .
UDINT_TO_LINT	LINT	UDINT LINT	.
UDINT_TO_USINT	USINT	0 255	가 .
UDINT_TO_UINT	UINT	0 65535	가 .
UDINT_TO_ULINT	ULINT	UDINT ULINT	.
UDINT_TO_BOOL	BOOL	1	BOOL .
UDINT_TO_BYTE	BYTE	8	BYTE .
UDINT_TO_WORD	WORD	16	WORD .
UDINT_TO_DWORD	DWORD		DWORD .
UDINT_TO_LWORD	LWORD	0	LWORD .
UDINT_TO_BCD	DWORD	0 99,999,999	, 가 .
UDINT_TO_REAL	REAL	UDINT REAL	가 .
UDINT_TO_LREAL	LREAL	UDINT LREAL	가 .
UDINT_TO_TOD	TOD		TOD .
UDINT_TO_TIME	TIME		TIME .

_ERR , _LER 가 (Set) .



(1) (%MO) On UDINT_TO_TIME
 (2) IN_VAL(UDINT) = 123 , OUT_VAL(TIME) = T#123MS

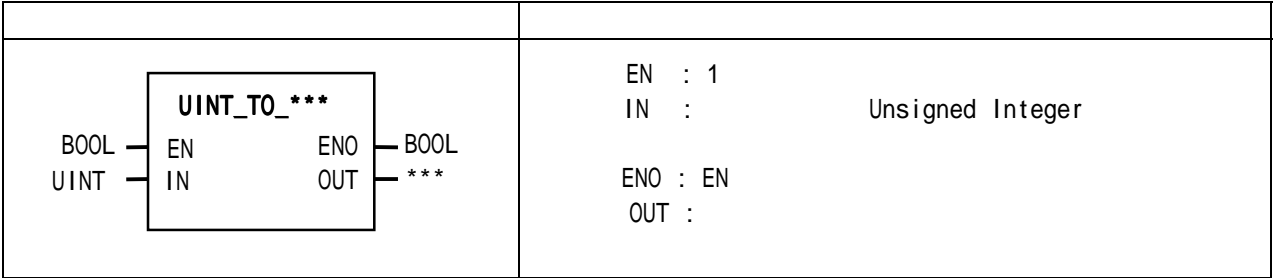
가

(IN1) : IN_VAL(UDINT) = 123
 ↓
 (OUT) : OUT_VAL(TIME) = T#123MS

UINT_TO_***

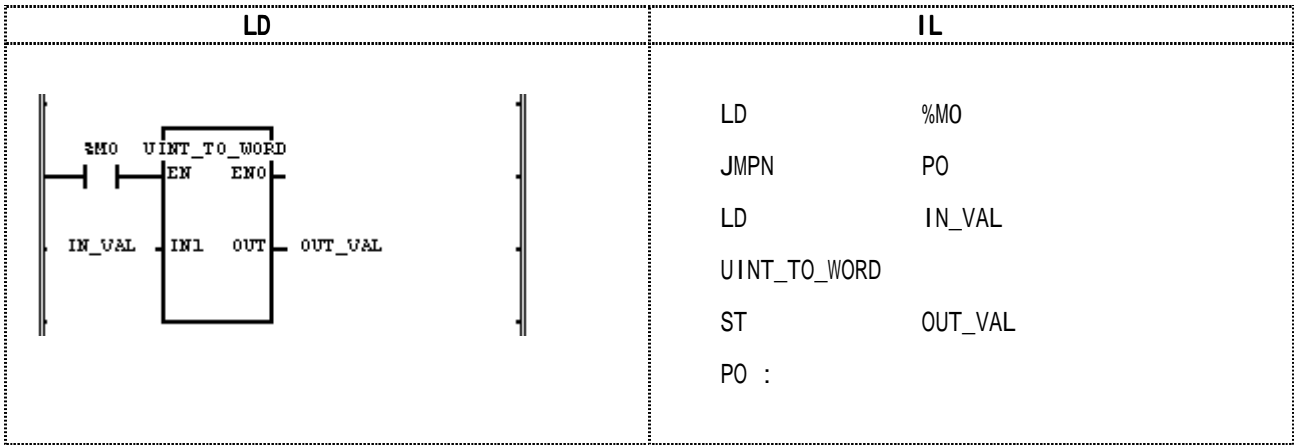
UINT

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

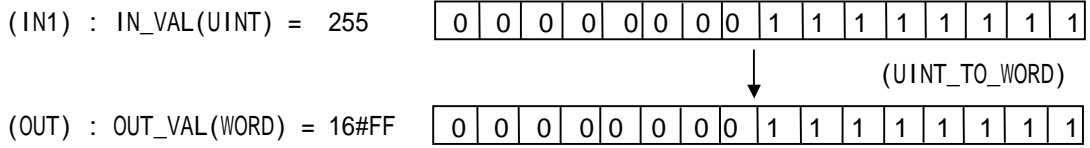


IN	OUT	
UINT_TO_SINT	SINT	0 127 가 .
UINT_TO_INT	INT	0 32,767 가 .
UINT_TO_DINT	DINT	UINT UDINT .
UINT_TO_LINT	LINT	UINT ULINT .
UINT_TO_USINT	USINT	0 255 가 .
UINT_TO_UDINT	UDINT	UINT UDINT .
UINT_TO_ULINT	ULINT	UINT ULINT .
UINT_TO_BOOL	BOOL	1 BOOL .
UINT_TO_BYTE	BYTE	8 BYTE .
UINT_TO_WORD	WORD	WORD .
UINT_TO_DWORD	DWORD	0 DWORD .
UINT_TO_LWORD	LWORD	0 LWORD .
UINT_TO_BCD	BCD	0 99,999,999 , 가 .
UINT_TO_REAL	REAL	UINT REAL .
UINT_TO_LREAL	LREAL	UINT LREAL .
UNIT_TO_DATE	DATE	DATE .

_ERR , _LER 가 (Set) .



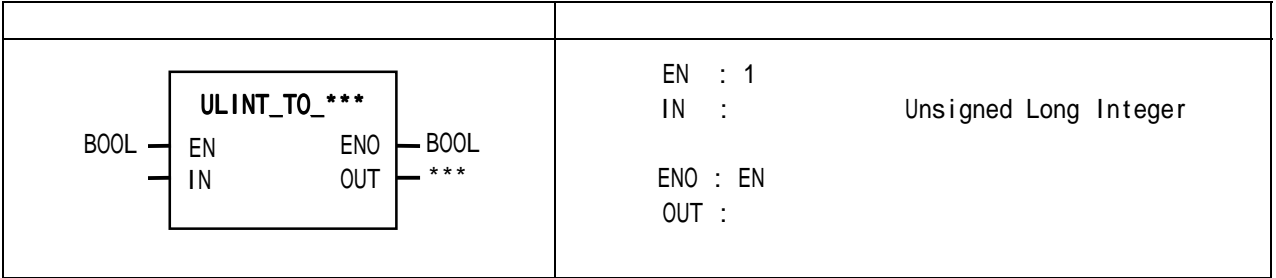
- (1) (%MO) On UINT_TO_WORD .
- (2) IN_VAL(UINT) = 255(2#0000_0000_1111_1111) ,
 OUT_VAL(WORD) = 2#0000_0000_1111_1111 .



ULINT_TO_***

ULINT

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT .

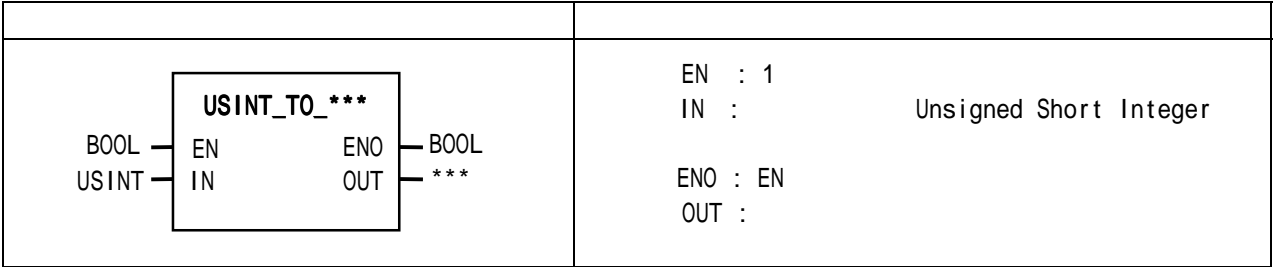
ULINT_TO_SINT	SINT	0 127	가	.
ULINT_TO_INT	INT	0 32,767	가	.
ULINT_TO_DINT	DINT	0 2 ³¹ -1	가	.
ULINT_TO_LINT	LINT	0 2 ⁶³ -1	가	.
ULINT_TO_USINT	USINT	0 255	가	.
ULINT_TO_UINT	UINT	0 65,535	가	.
ULINT_TO_UDINT	UDINT	0 2 ³² -1	가	.
ULINT_TO_BOOL	BOOL	1	BOOL	.
ULINT_TO_BYTE	BYTE	8	BYTE	.
ULINT_TO_WORD	WORD	16	WORD	.
ULINT_TO_DWORD	DWORD	32	DWORD	.
ULINT_TO_LWORD	LWORD		LWORD	.
ULINT_TO_BCD	BCD	0 9,999,999,999,999,999	가	.
ULINT_TO_REAL	REAL	ULINT REAL	가	.
ULINT_TO_LREAL	LREAL	ULINT LREAL	가	.

_ERR, _LER 가 (Set) .

USINT_TO_***

USINT

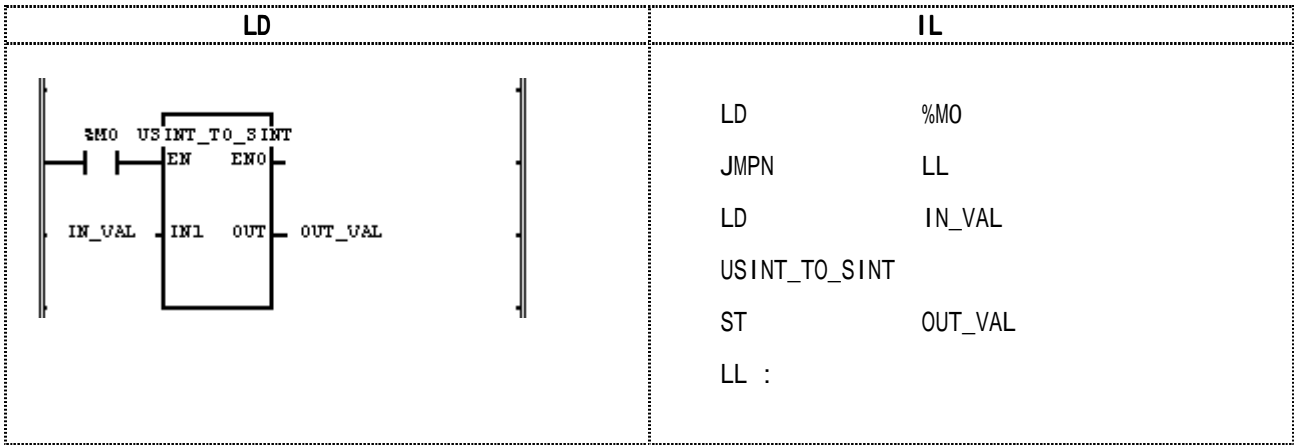
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN OUT

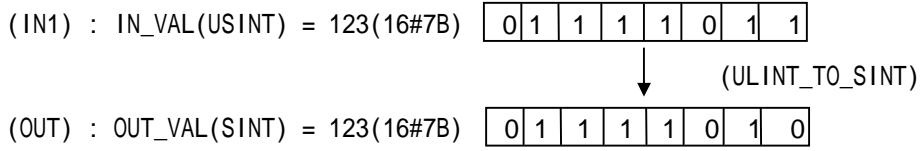
USINT_TO_SINT	SINT	0 127	가
USINT_TO_INT	INT	INT	
USINT_TO_DINT	DINT	DINT	
USINT_TO_LINT	LINT	LINT	
USINT_TO_UINT	UINT	UINT	
USINT_TO_UDINT	UDINT	UDINT	
USINT_TO_ULINT	ULINT	ULINT	
USINT_TO_BOOL	BOOL	1 BOOL	
USINT_TO_BYTE	BYTE	BYTE	
USINT_TO_WORD	WORD	0 WORD	
USINT_TO_DWORD	DWORD	0 DWORD	
USINT_TO_LWORD	LWORD	0 LWORD	
USINT_TO_BCD	BCD	0 99 , 가	
USINT_TO_REAL	REAL	USINT REAL	
USINT_TO_LREAL	LREAL	USINT LREAL	

_ERR, _LER 가 (Set)



(1) (%MO) On ULINT_TO_SINT

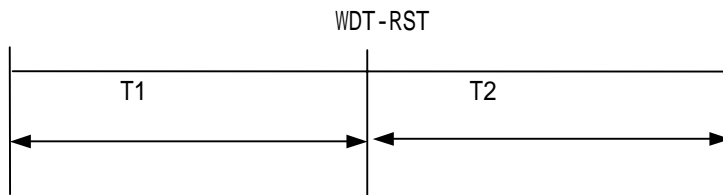
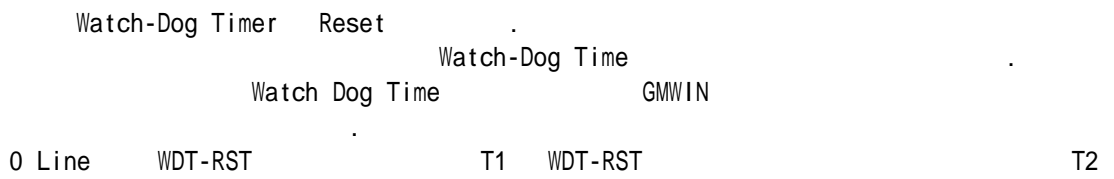
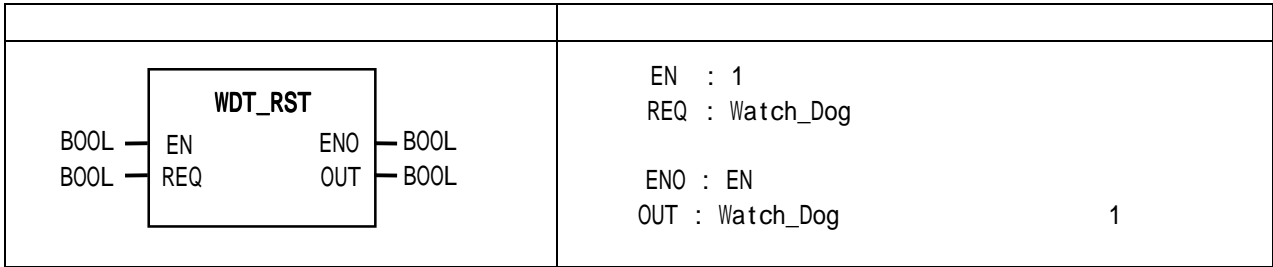
(2) IN_VAL(USINT) = 123 , OUT_VAL(SINT) = 123



WDT_RST

Watch_Dog

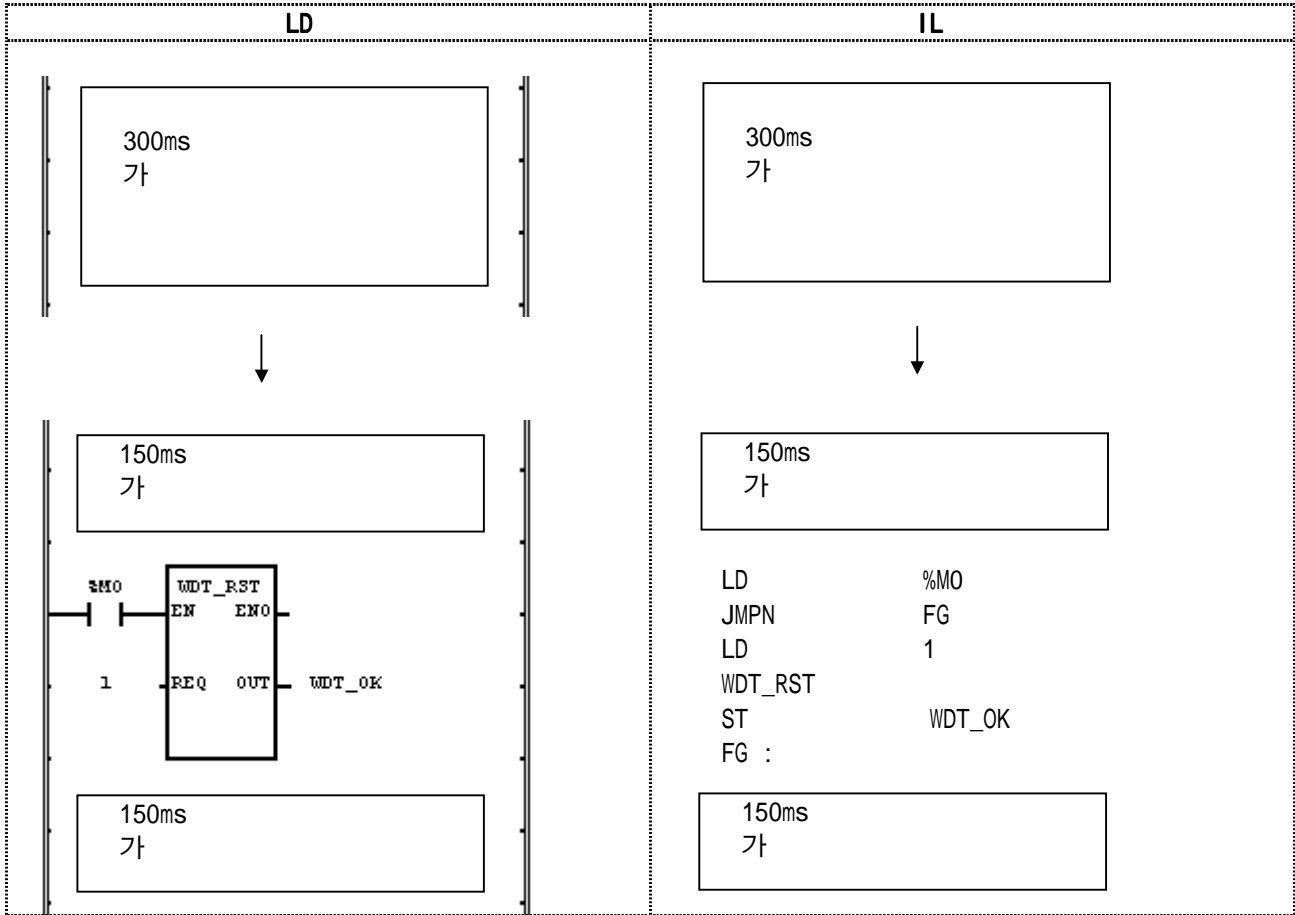
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



- WDT-RST 1 가 .

200ms

300ms가



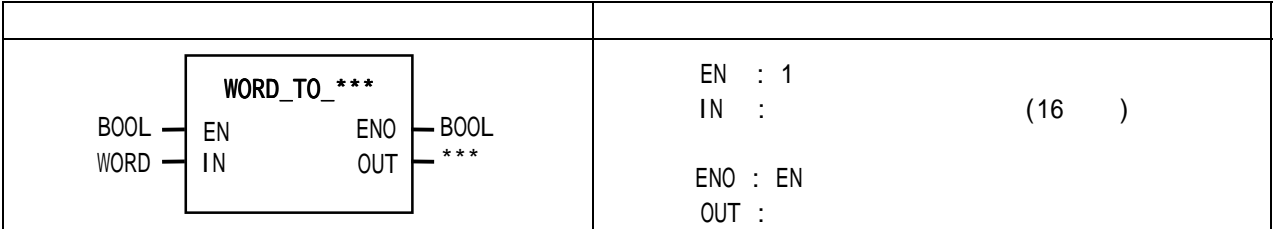
(1) (%MO)가 On WDT-RST(Watch Dog)
 (2)WDT-RST ,
 (200ms)

)
 300ms

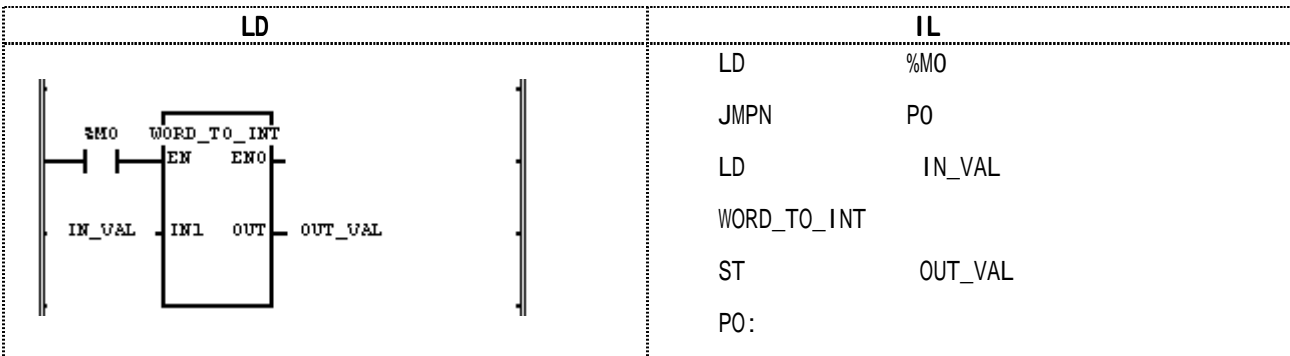
WORD_TO_***

WORD

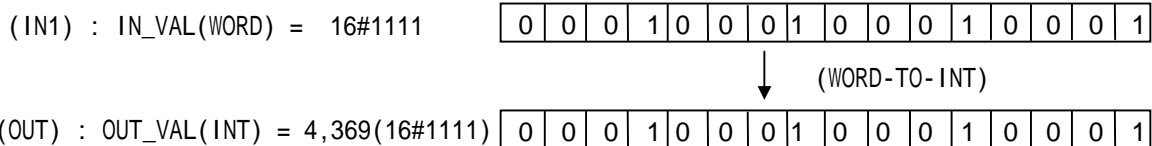
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN	OUT		
WORD_TO_SINT	SINT	8	SINT
WORD_TO_INT	INT		INT
WORD_TO_DINT	DINT	0	DINT
WORD_TO_LINT	LINT	0	LINT
WORD_TO_USINT	USINT	8	SINT
WORD_TO_UINT	UINT		INT
WORD_TO_UDINT	UDINT	0	DINT
WORD_TO_ULINT	ULINT	0	LINT
WORD_TO_BOOL	BOOL	1	BOOL
WORD_TO_BYTE	BYTE	8	SINT
WORD_TO_DWORD	DWORD	0	DWORD
WORD_TO_LWORD	LWORD	0	LWORD
WORD_TO_DATE	DATE		DATE
WORD_TO_STRING	STRING	WORD	STRING



- (1) (%MO) On WORD-TO-INT
- (2) IN_VAL(WORD) = 2#0001_0001_0001_0001 ,
OUT_VAL(INT) = 4096 + 256 + 16 + 1 = 4,369가



XOR



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

	<p>EN : 1 IN1 : XOR IN2 : XOR 8 가</p> <p>ENO : EN OUT : XOR</p> <p>IN1, IN2, OUT .</p>
--	--

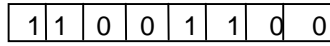
```

IN1  IN2      XOR  OUT
IN1  1111  ..... 0000
XOR
IN2  1010  ..... 1010
OUT  0101  ..... 1010
    
```

<p style="text-align: center;">LD</p>	<p style="text-align: center;">IL</p> <pre> LD %M0 JMPN ZZ LD %MB10 XOR IN1:= CURRENT RESULT IN2:= ABC ST %QB0.0.0 ZZ : </pre>
--	---

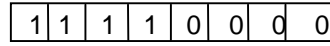
- (1) (%M0) On XOR()
- (2) %MB10=11001100, ABC=11110000 , XOR 가 %QB0.0.0 = 00111100

(IN1) : %MB10(BYTE) = 16#CC

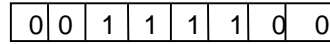


(XOR)

(IN2) : ABC(BYTE) = 16#F0



(OUT) : %QB0.0.0(BYTE) = 16#3C



8. /

8.2

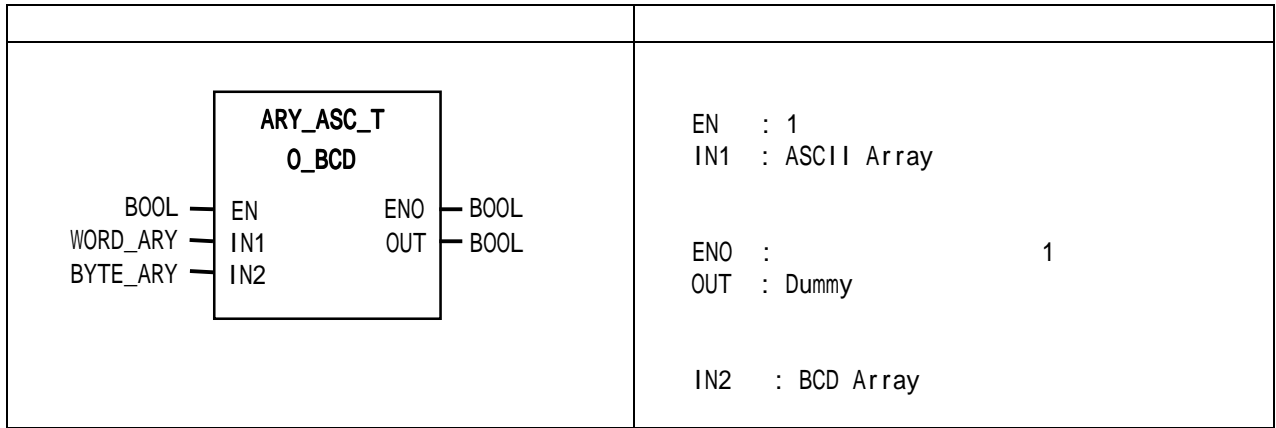
8.1

(MASTER-K)

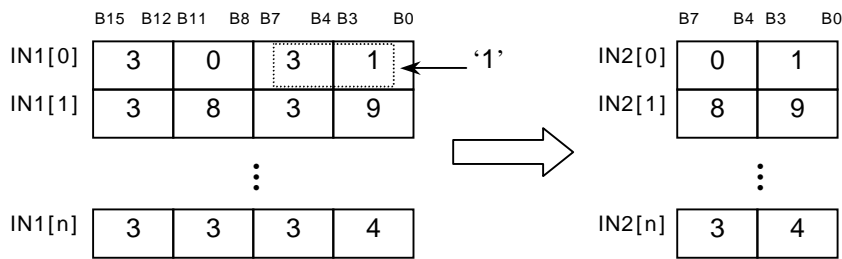
ARY_ASC_TO_BCD

ASCII Array
BCD Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

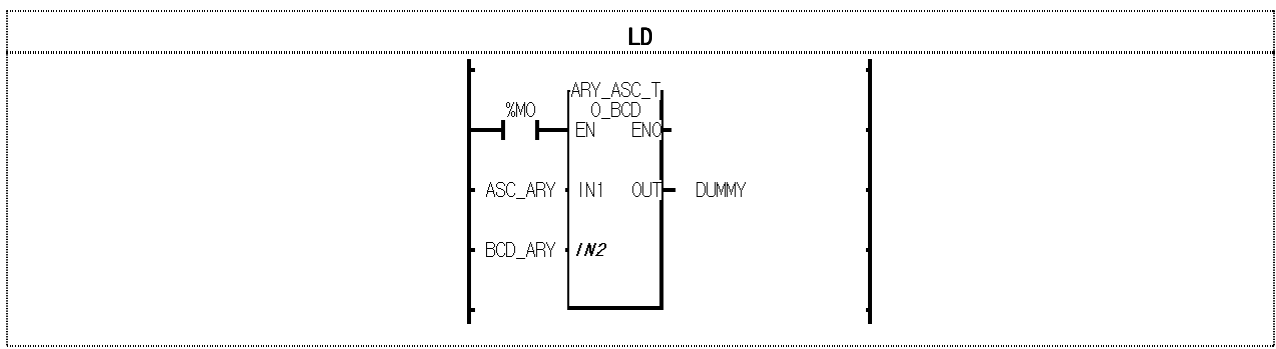


ASCII WORD Array , BCD(Binary Coded Decimal) BYTE Array



(Set) IN1 Array 16#00 가 , IN2 가 _ERR/_LER 가

IN1 , IN1 IN2 Array 가 (Set)



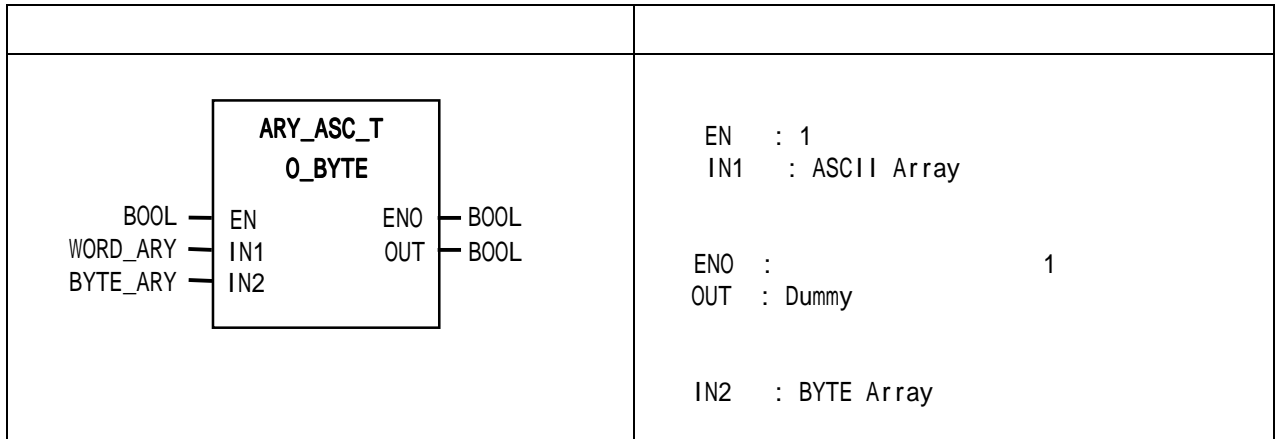
- (1) (%M0) On , ARY_ASC_TO_BCD
 - (2) ASC_ARY
- | | |
|------------|-------|
| ASC_ARY[0] | 3031H |
| ASC_ARY[1] | 3839H |
| ASC_ARY[2] | 3334H |

	BCD_ARY
BYTE_ARY[0]	01H
BYTE_ARY[1]	89H
BYTE_ARY[2]	34H

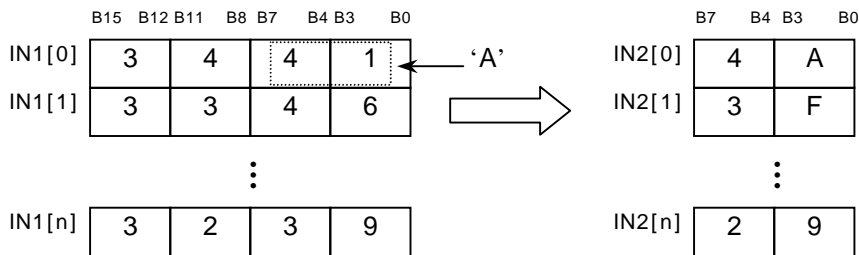
ARY_ASC_TO_BYTE

ASCII Array
 BYTE Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

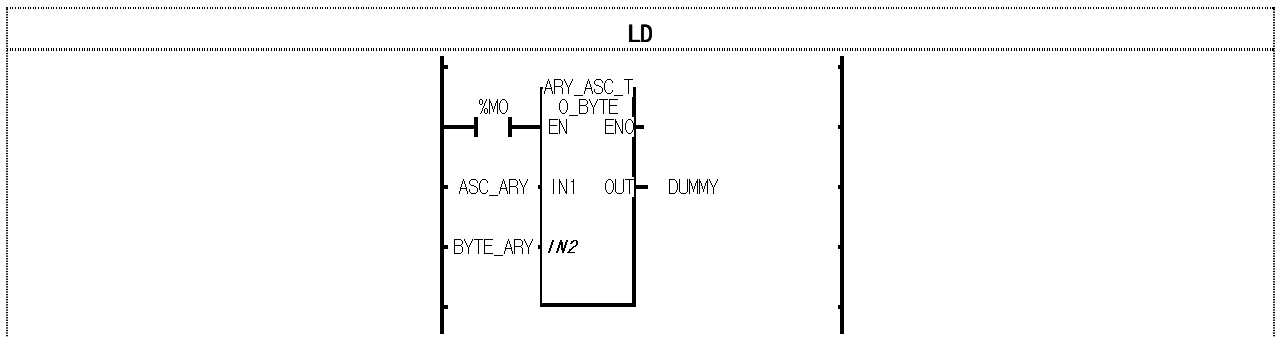


ASCII WORD Array , 16 (HEX) BYTE Array .



(Set) IN1 Array 16 '0' 'F' , IN1 IN2 Array 가 _ERR/_LER 가

0 _ERR/_LER 가 (Set) .



- (1) (%MO) On , ARY_ASC_TO_BYTE .
- (2) ASC_ARRAY

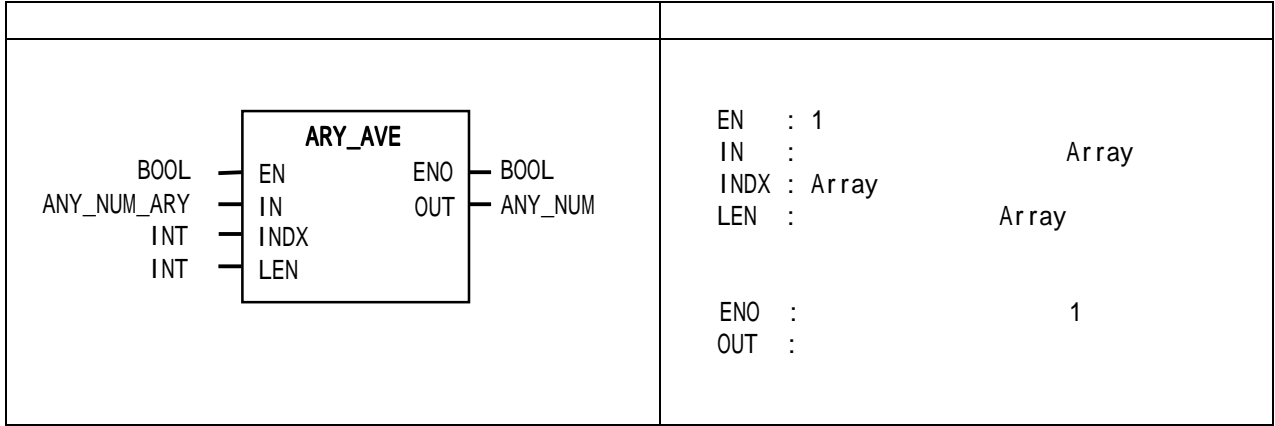
ASC_ARRAY[0]	3441H
ASC_ARRAY[1]	3346H
ASC_ARRAY[2]	3239H

BYTE_ARY	
BYTE_ARY[0]	4AH
BYTE_ARY[1]	3FH
BYTE_ARY[2]	29H

ARY_AVE_***

Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

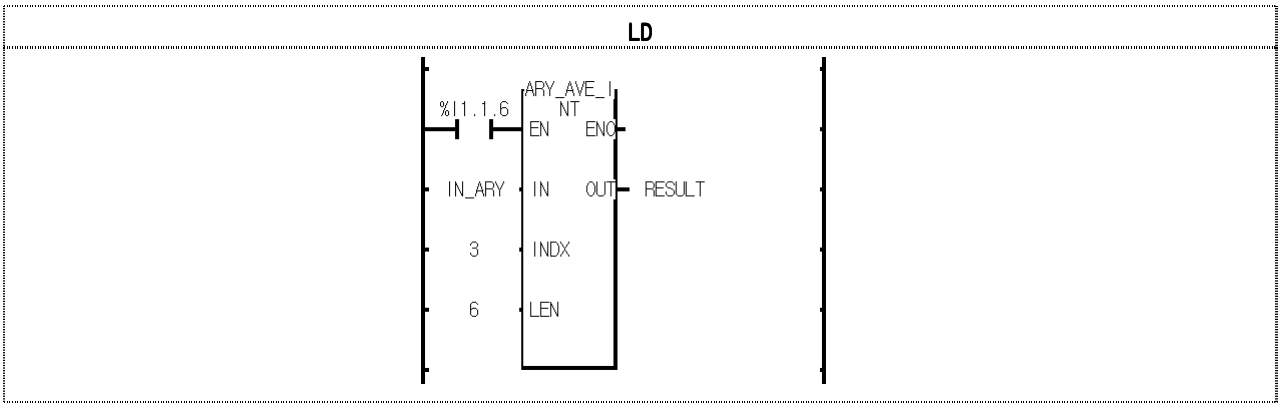


ARY_AVE_*** Array
 Array
 LEN Array (Array - |LEN|)

ARY_AVE_SINT	SINT	SINT	.()
ARY_AVE_INT	INT	SINT	.()
ARY_AVE_DINT	DINT	DINT	.()
ARY_AVE_LINT	LINT	LINT	.()
ARY_AVE_USINT	USINT	USINT	.()
ARY_AVE_UINT	UINT	UINT	.()
ARY_AVE_UDINT	UDINT	UDINT	.()
ARY_AVE_ULINT	ULINT	ULINT	.()
ARY_AVE_REAL	REAL	REAL	.
ARY_AVE_LREAL	LREAL	LREAL	.

Array OUT 0 _ERR/ _LER 가 (Set) .

가
 INDX < 0 INDX > IN
INDX + LEN > IN



0	11245
1	23454
2	8764
3	9563
4	18764
5	7765
6	29215
7	21004
8	10048
9	18081

IN_ARY
10
INT

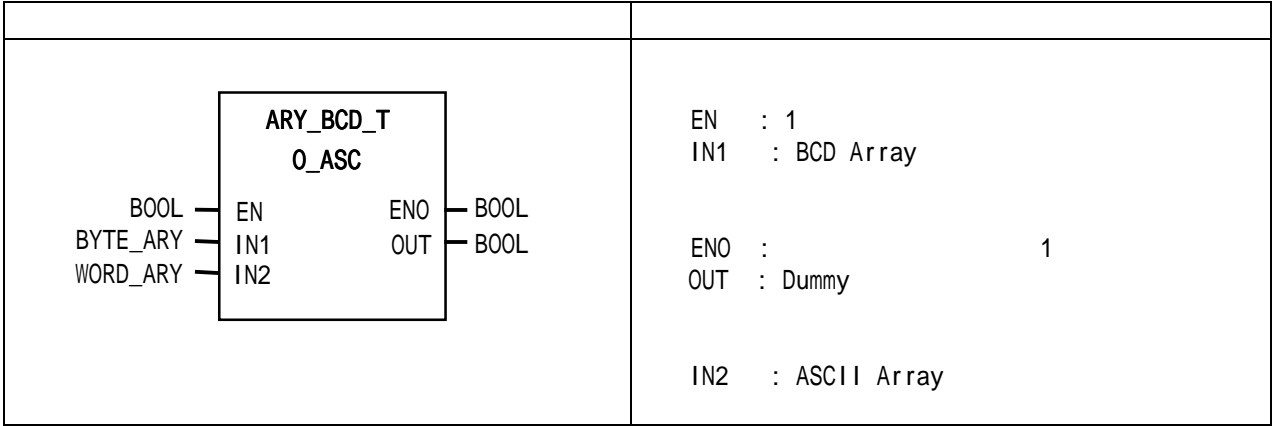
$$\frac{9563 + 18764 + 7765 + 29215 + 21004 + 10048}{6} = 16044.83 = 16045$$

- (1) (%I1.1.6) On , ARY_AVE_INT
- (2) Array Array 3 6
- (3) 16044.8 INT 16045

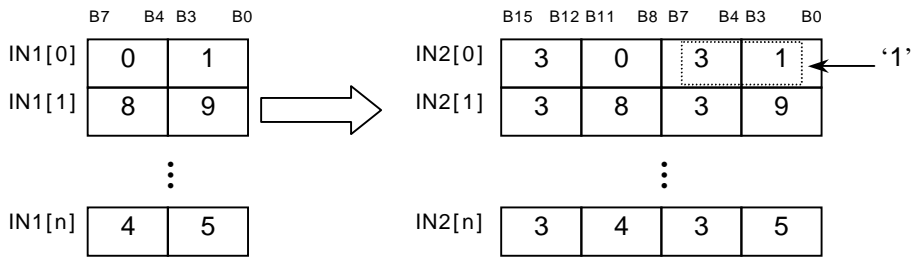
ARY_BCD_TO_ASC

BCD Array
ASCII Array

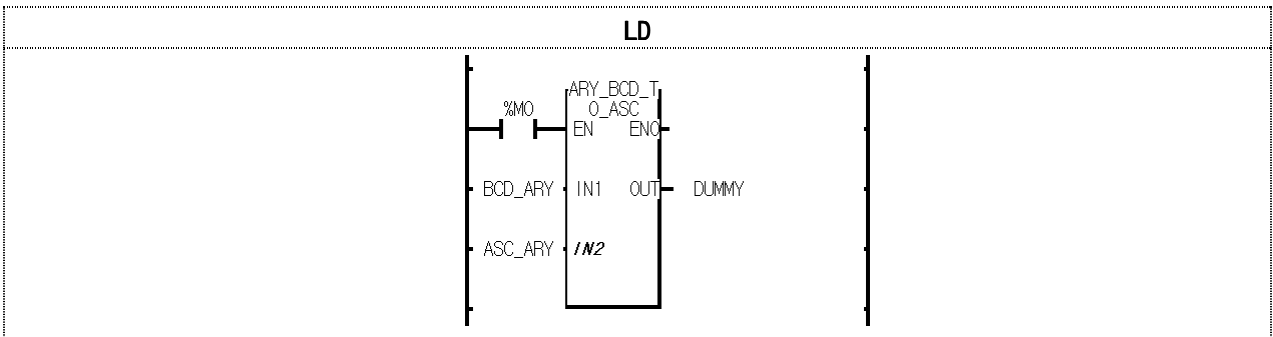
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



BCD BYTE Array , ASCII WORD Array .



(Set) IN1Array 16 0 9 , IN1 IN2 Array _ERR/_LER 가
16#3030("00") , IN2 가 _ERR/_LER 가
(Set) .



- (1) (%M0) On , ARY_BCD_TO_ASC
- (2) BCD_ARRAY

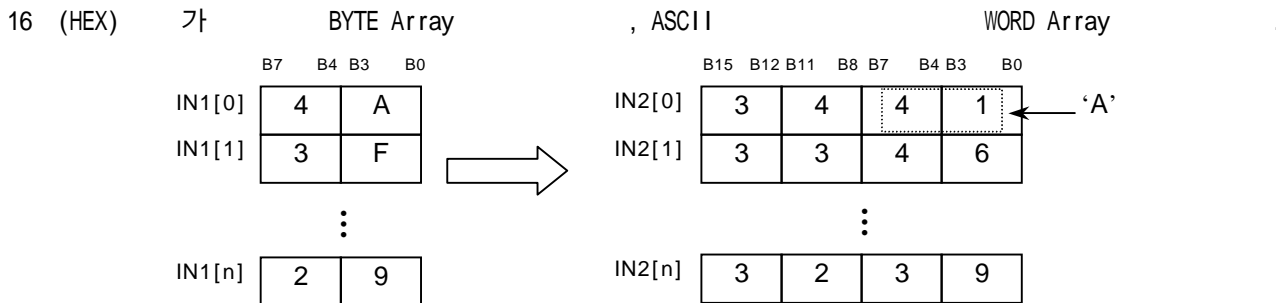
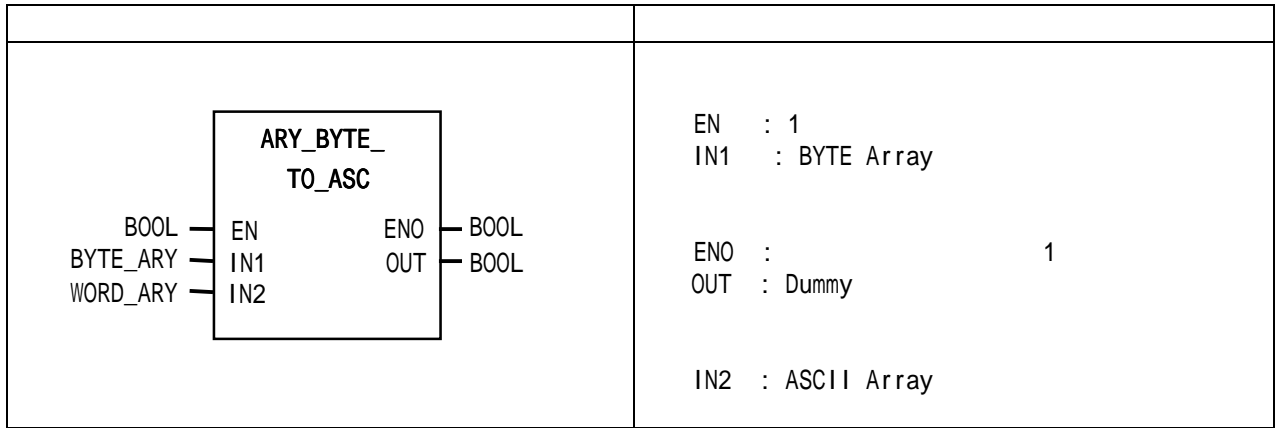
BYTE_ARRAY[0]	01H
BYTE_ARRAY[1]	89H
BYTE_ARRAY[2]	45H

ASC_ARY	
ASC_ARY[0]	3031H
ASC_ARY[1]	3839H
ASC_ARY[2]	3435H

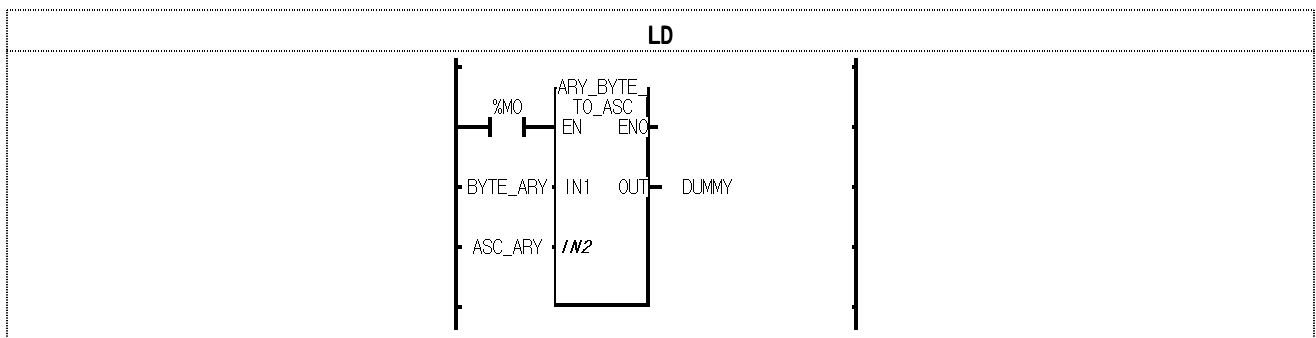
ARY_BYTE_TO_ASC

BYTE Array
ASCII Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



2 Array 가 , IN2 가 _ERR/_LER 가 (Set)



- (1) (%M0) On , ARY_BYTE_TO_ASC
- (2) BYTE_ARRAY

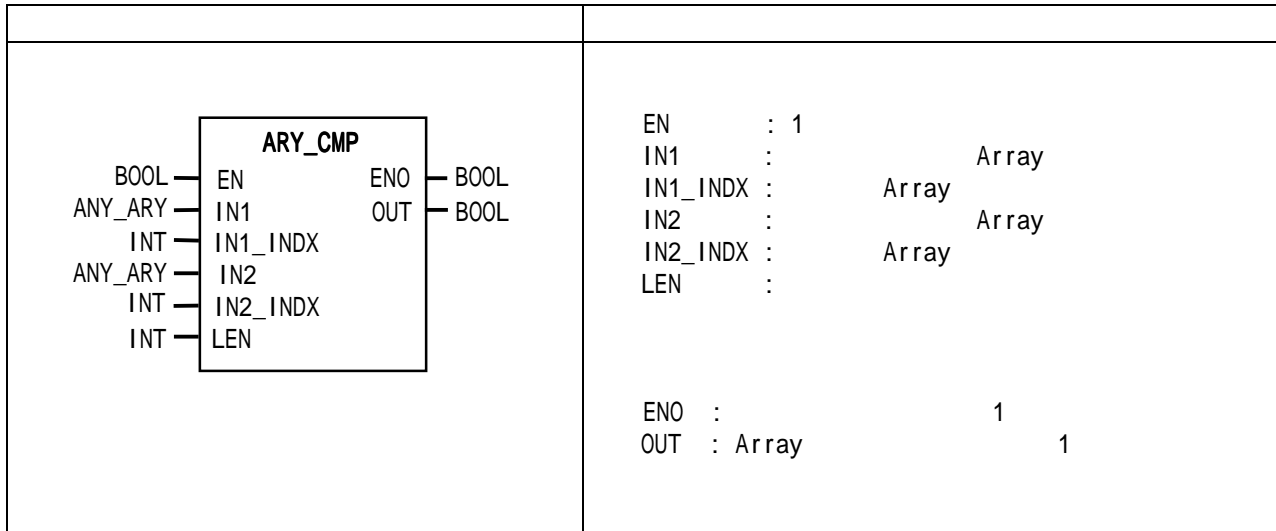
BYTE_ARRAY[0]	4AH
BYTE_ARRAY[1]	3FH
BYTE_ARRAY[2]	29H

ASC_ARY	
ASC_ARY[0]	3441H
ASC_ARY[1]	3346H
ASC_ARY[2]	3239H

ARY_CMP_***

Array

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



```

EN      : 1
IN1     :           Array
IN1_INDX :         Array
IN2     :           Array
IN2_INDX :         Array
LEN     :

ENO    :           1
OUT   : Array           1
    
```

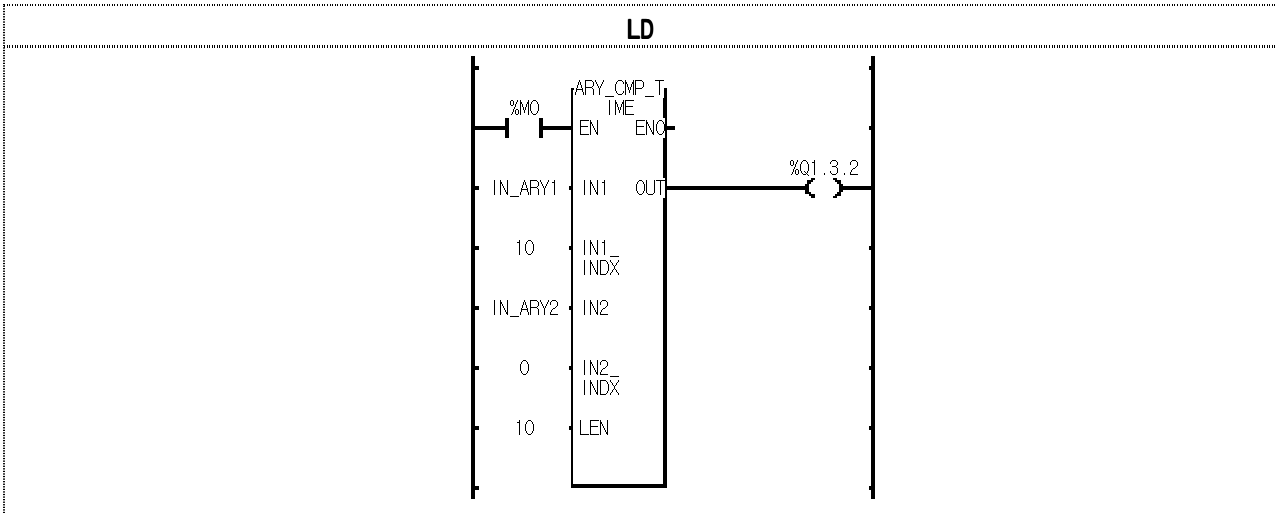
ARY_CMP_*** 2 Array 가
 LEN Array (Array - |LEN|)

	Array			
ARY_CMP_BOOL	BOOL	2	BOOL Array	.
ARY_CMP_BYTE	BYTE	2	BYTE Array	.
ARY_CMP_WORD	WORD	2	WORD Array	.
ARY_CMP_DWORD	DWORD	2	DWORD Array	.
ARY_CMP_LWORD	LWORD	2	LWORD Array	.
ARY_CMP_SINT	SINT	2	SINT Array	.
ARY_CMP_INT	INT	2	INT Array	.
ARY_CMP_DINT	DINT	2	DINT Array	.
ARY_CMP_LINT	LINT	2	LINT Array	.
ARY_CMP_USINT	USINT	2	USINT Array	.
ARY_CMP_UINT	UINT	2	UINT Array	.
ARY_CMP_UDINT	UDINT	2	UDINT Array	.
ARY_CMP_ULINT	ULINT	2	ULINT Array	.
ARY_CMP_REAL	REAL	2	REAL Array	.
ARY_CMP_LREAL	LREAL	2	LREAL Array	.
ARY_CMP_TIME	TIME	2	TIME Array	.
ARY_CMP_DATE	DATE	2	DATE Array	.
ARY_CMP_TOD	TOD	2	TOD Array	.
ARY_CMP_DT	DT	2	DT Array	.

Array ERR/_LER 가 (Set) .

가

- IN1_INDX < 0 IN1_INDX > IN1
- IN2_INDX < 0 IN2_INDX > IN2
- IN1_INDX + LEN ≥ IN1
- IN2_INDX + LEN ≥ IN2

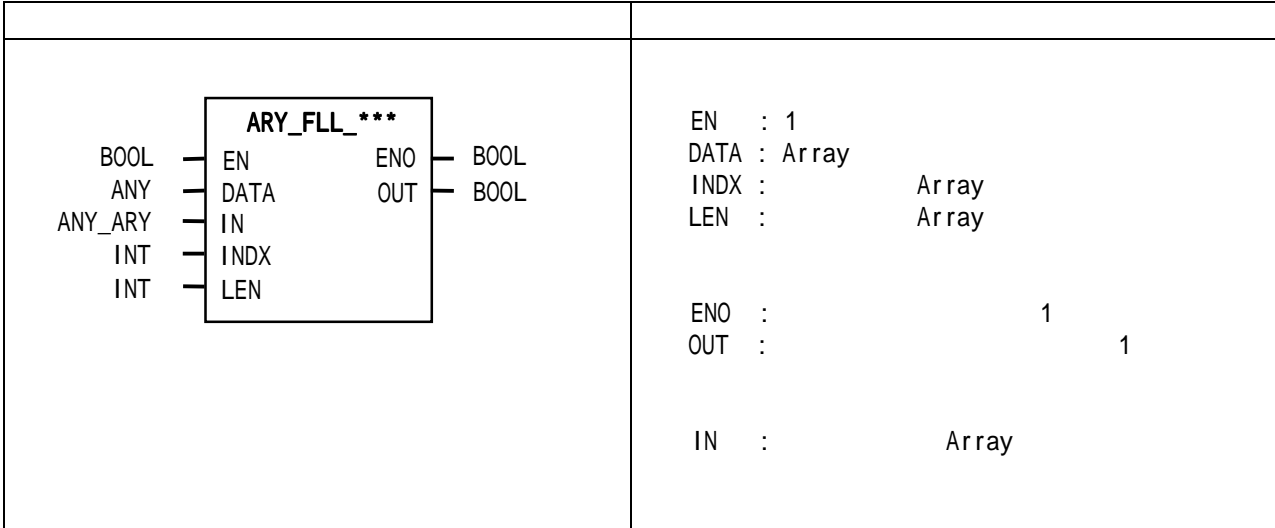


- (1) (%M0) On ARY_CMP_TIME .
- (2) IN_ARY1 100 TIME Array , IN_ARY2가 10 TIME Array
- IN_ARY1 11 20 10 IN_ARY 2 10
- 10 %Q1.3.2가 On .

ARY_FLL_***

Array

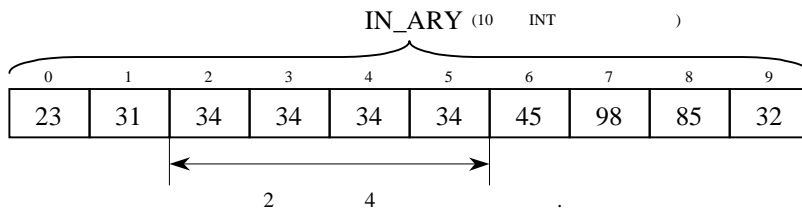
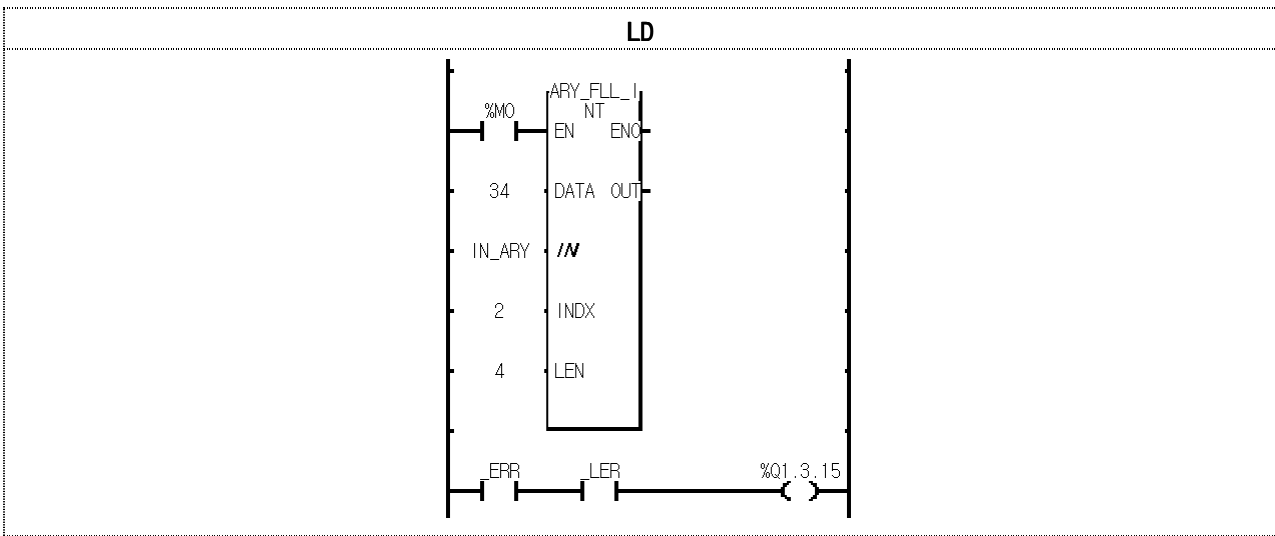
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



ARY_FLL_*** DATA Array
 LEN Array (Array - |LEN|)

	Array	
ARY_FLL_BOOL	BOOL	BOOL Array
ARY_FLL_BYTE	BYTE	BYTE Array
ARY_FLL_WORD	WORD	WORD Array
ARY_FLL_DWORD	DWORD	DWORD Array
ARY_FLL_LWORD	LWORD	LWORD Array
ARY_FLL_SINT	SINT	SINT Array
ARY_FLL_INT	INT	INT Array
ARY_FLL_DINT	DINT	DINT Array
, ARY_FLL_LINT	LINT	LINT Array
ARY_FLL_USINT	USINT	USINT Array
ARY_FLL_UINT	UINT	UINT Array
ARY_FLL_UDINT	UDINT	UDINT Array
ARY_FLL_ULINT	ULINT	ULINT Array
ARY_FLL_REAL	REAL	REAL Array
ARY_FLL_LREAL	LREAL	LREAL Array
ARY_FLL_TIME	TIME	TIME Array
ARY_FLL_DATE	DATE	DATE Array
ARY_FLL_TOD	TOD	TOD Array
ARY_FLL_DT	DT	DT Array

Array OUT OFF , IN Array 가 (Set) .
 가
 INDX < 0 INDX > IN
 INDX + LEN ≥ IN

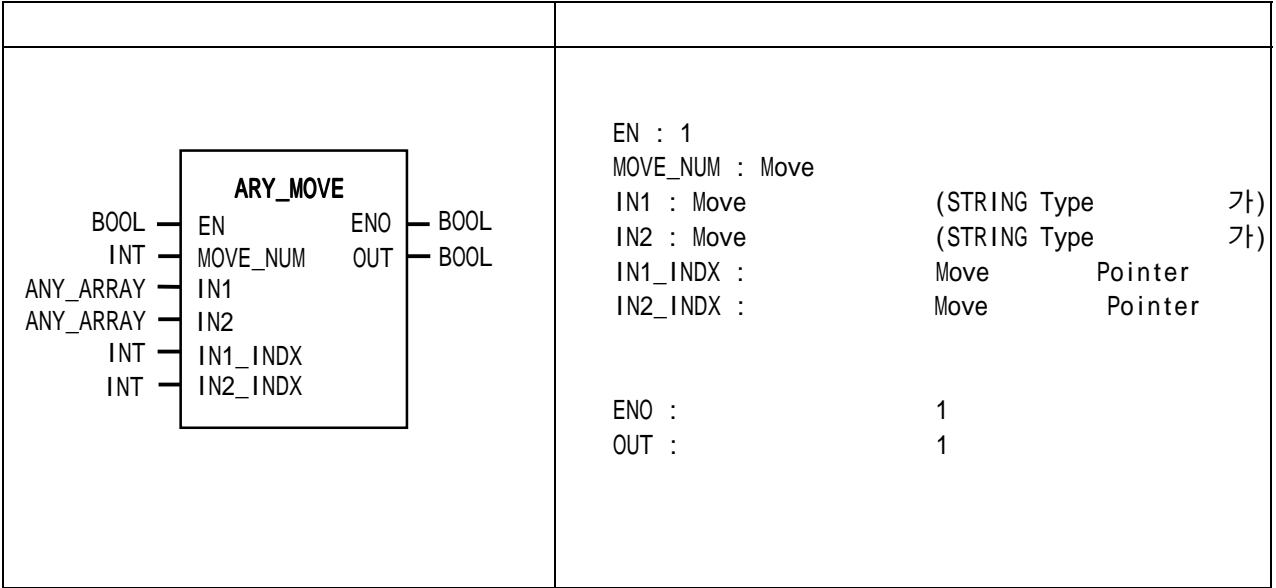


- (1) (%M0) On , ARY_FLL_INT
- (2) Array 2 4 34
- (3) LEN 9 Array 가 _ERR _LER 가
 On %Q1.13.15가 On

ARY_MOVE

Array

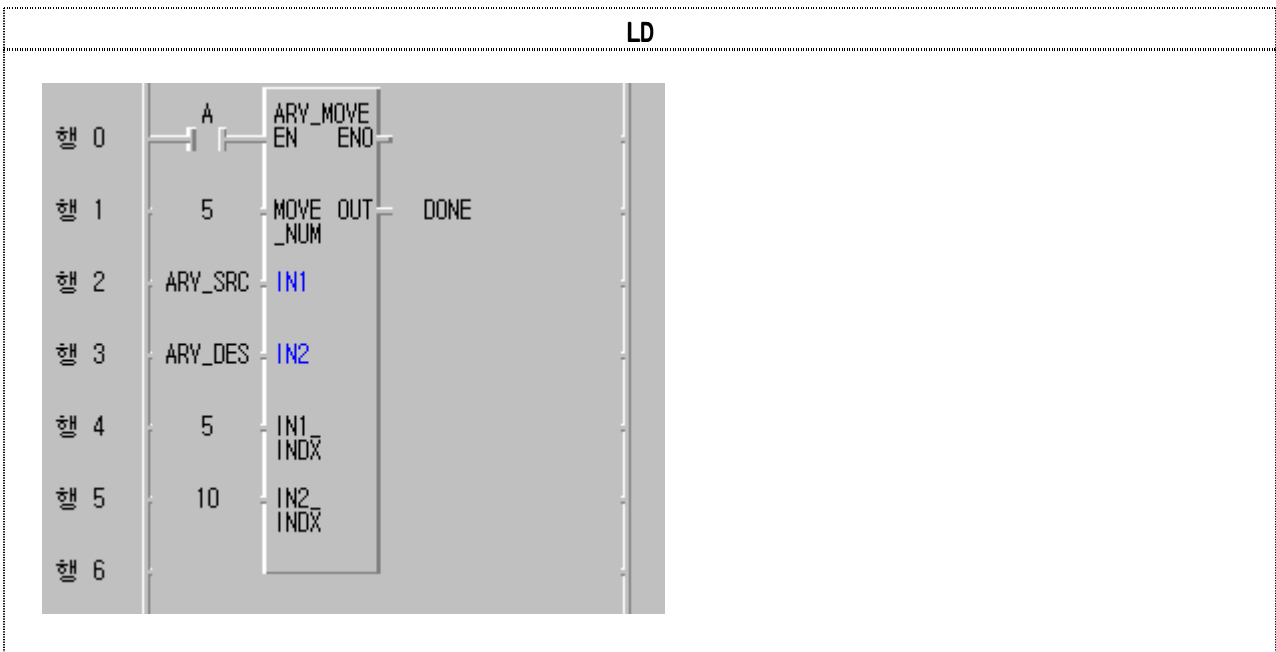
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1 , IN1 IN2
 IN1 IN1_INDX MOVE_NUM , IN2 IN2_INDX
 MOVE가 가 IN1 IN2 Size가 , IN1 IN2

Size	
1 Bit	BOOL
8 Bit	BYTE, SINT, USINT
16 Bit	WORD / INT / UINT / DATE
32 Bit	DWORD / DINT / UDINT / TIME / TOD
64 Bit	DT

IN1 IN2 Size가 가
 IN1 Array 가 (IN1_INDX + MOVE_NUM) , IN2 Array 가 (IN2_INDX +
 MOVE_NUM) 가 , OUT 0
 , ENO가 OFF _ERR,_LER 가 (Set)



ARY_SRC	INT	10
ARY_DEST	WORD	15

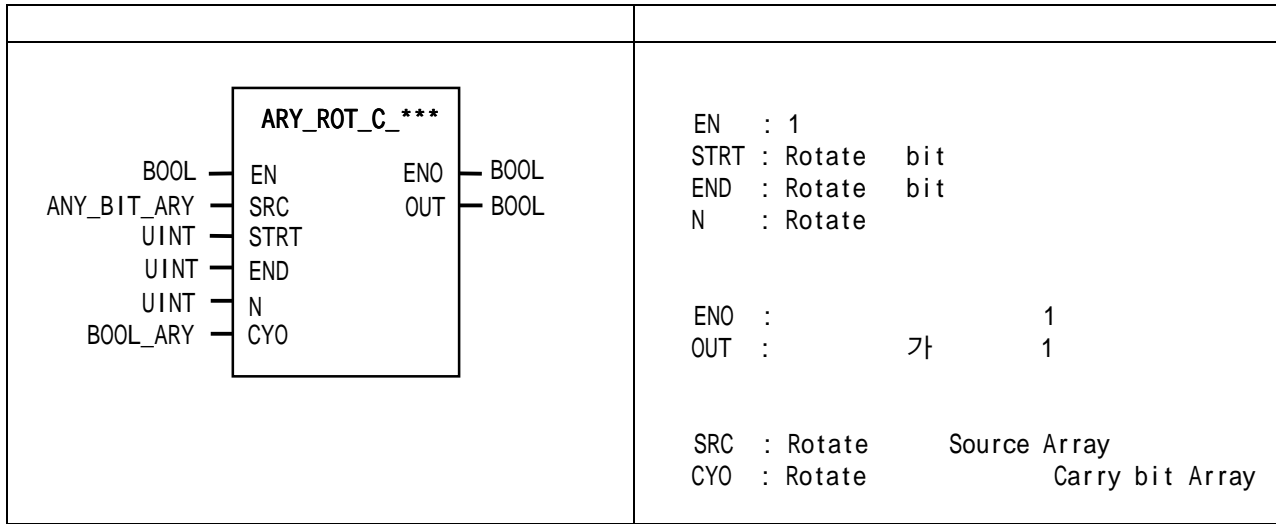
- (1) (A) On ARY_MOVE
- (2) ARY_SRC[5] 5 ARY_DEST[10]
 , ARY_DEST WORD 16

ARY_SRC[0]	0	ARY_DEST[0]	16#0	ARY_SRC[0]	0	ARY_DEST[0]	16#0
ARY_SRC[1]	11	ARY_DEST[1]	16#1	ARY_SRC[1]	11	ARY_DEST[1]	16#1
ARY_SRC[2]	22	ARY_DEST[2]	16#2	ARY_SRC[2]	22	ARY_DEST[2]	16#2
ARY_SRC[3]	33	ARY_DEST[3]	16#3	ARY_SRC[3]	33	ARY_DEST[3]	16#3
ARY_SRC[4]	44	ARY_DEST[4]	16#4	ARY_SRC[4]	44	ARY_DEST[4]	16#4
ARY_SRC[5]	55	ARY_DEST[5]	16#5	ARY_SRC[5]	55	ARY_DEST[5]	16#5
ARY_SRC[6]	66	ARY_DEST[6]	16#6	ARY_SRC[6]	66	ARY_DEST[6]	16#6
ARY_SRC[7]	77	ARY_DEST[7]	16#7	ARY_SRC[7]	77	ARY_DEST[7]	16#7
ARY_SRC[8]	88	ARY_DEST[8]	16#8	ARY_SRC[8]	88	ARY_DEST[8]	16#8
ARY_SRC[9]	99	ARY_DEST[9]	16#9	ARY_SRC[9]	99	ARY_DEST[9]	16#9
		ARY_DEST[10]	16#A			ARY_DEST[10]	16#37
		ARY_DEST[11]	16#B			ARY_DEST[11]	16#42
		ARY_DEST[12]	16#C			ARY_DEST[12]	16#4D
		ARY_DEST[13]	16#D			ARY_DEST[13]	16#58
		ARY_DEST[14]	16#E			ARY_DEST[14]	16#63

ARY_ROT_C_***

Array Bit Rotate with Carry

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



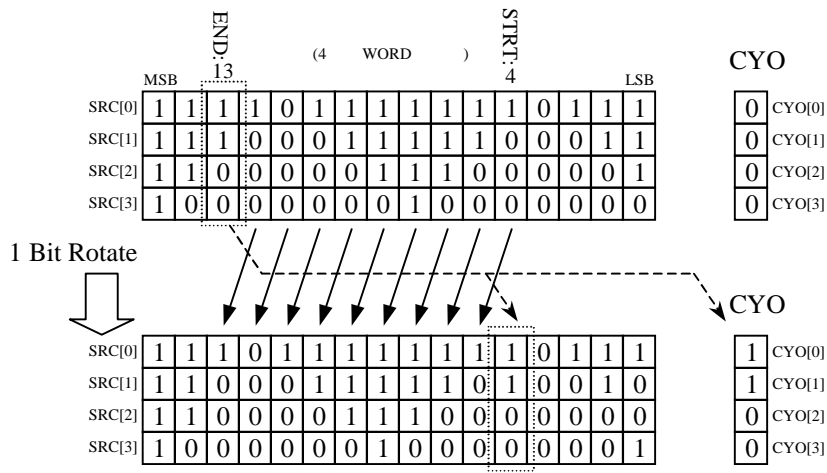
ARY_ROT_C_*** Array bit Rotate .

- : STRT END .

- : STRT END (N) Rotate

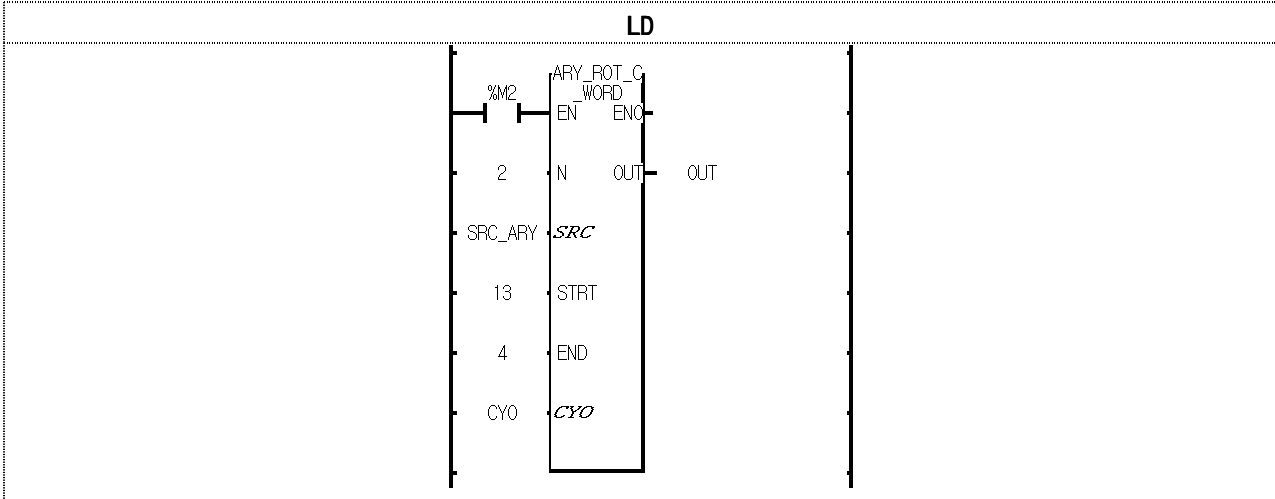
- : SRC ANY_BIT_ARY , END

STRT 가 CYO Array .



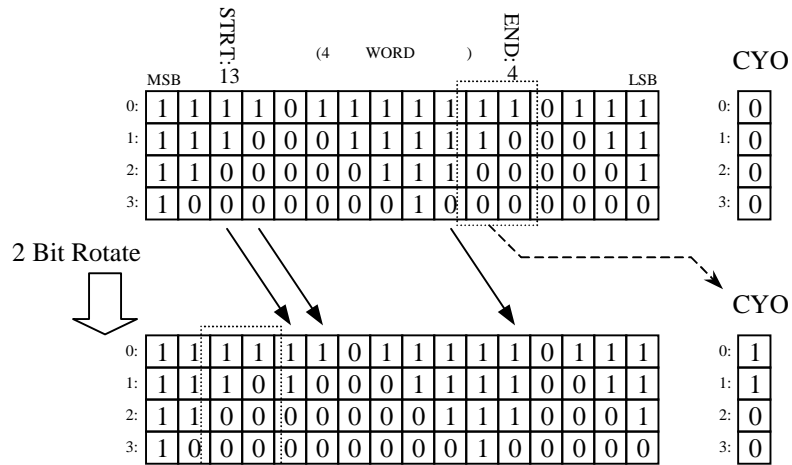
	Array	
ARY_ROT_C_BYTE	BYTE	Array Rotate .
ARY_ROT_C_WORD	WORD	
ARY_ROT_C_DWORD	DWORD	
ARY_ROT_C_LWORD	LWORD	

SRC CYO Array 가 _ERR/_LER 가 (Set)
 STRT END가 SRC 가
 SRC CYO



- (1) (%M2) On , ARY_ROT_C_WORD
- (2) STRT 13 bit END 4 STRT END 2
- (3) SRC_ARY CYO BOOL Array

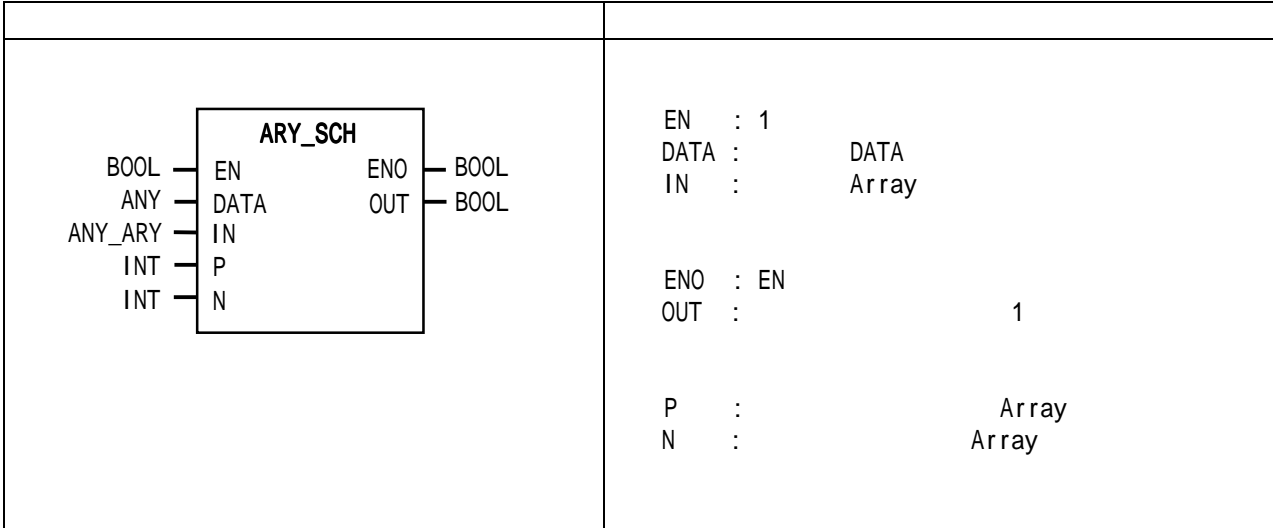
()
 SRC_ARY : 16#F7F7
 16#E3E3
 16#C1C1
 16#8080
 (N) : 2
 ()
 SRC_ARY : 16#FDF7
 16#E8F3
 16#C071
 16#8020
 CYO : 2#1100



ARY_SCH_***

Array (search)

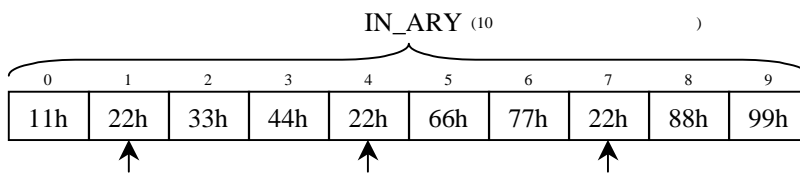
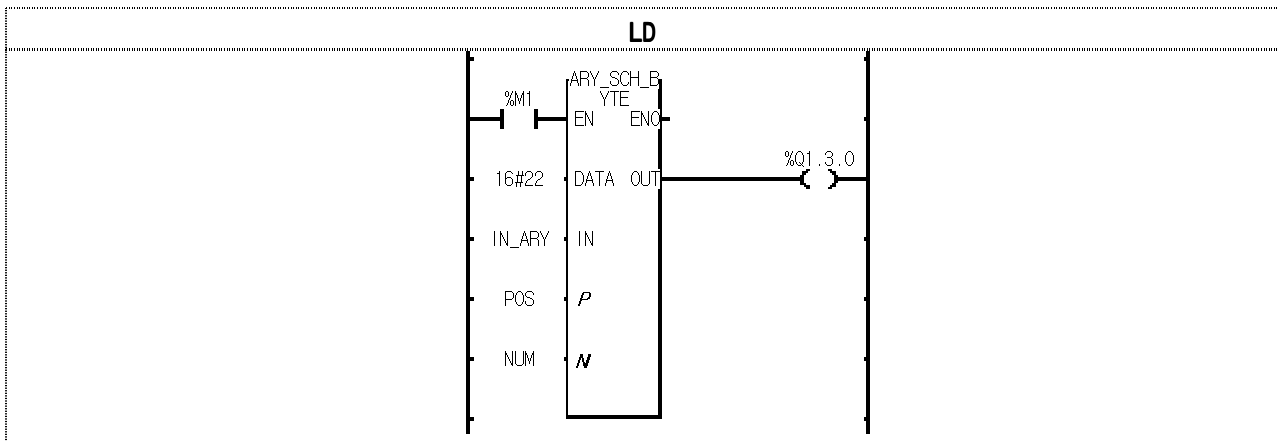
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



ARY_SCH_*** Array
. Array

Array
OUT 1 .

	Array	
ARY_SCH_BOOL	BOOL	BOOL Array .
ARY_SCH_BYTE	BYTE	BYTE Array .
ARY_SCH_WORD	WORD	WORD Array .
ARY_SCH_DWORD	DWORD	DWORD Array .
ARY_SCH_LWORD	LWORD	LWORD Array .
ARY_SCH_SINT	SINT	SINT Array .
ARY_SCH_INT	INT	INT Array .
ARY_SCH_DINT	DINT	DINT Array .
ARY_SCH_LINT	LINT	LINT Array .
ARY_SCH_USINT	USINT	USINT Array .
ARY_SCH_UINT	UINT	UINT Array .
ARY_SCH_UDINT	UDINT	UDINT Array .
ARY_SCH_ULINT	ULINT	ULINT Array .
ARY_SCH_REAL	REAL	REAL Array .
ARY_SCH_LREAL	LREAL	LREAL Array .
ARY_SCH_TIME	TIME	TIME Array .
ARY_SCH_DATE	DATE	DATE Array .
ARY_SCH_TOD	TOD	TOD Array .
ARY_SCH_DT	DT	DT Array .

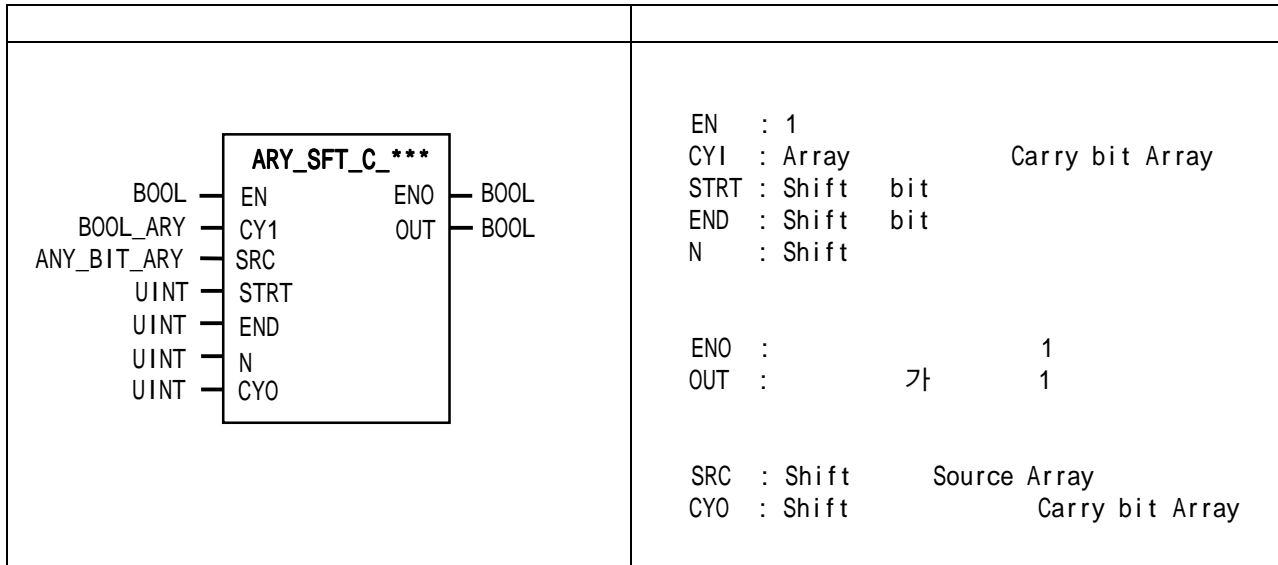


- (1) (%M1) On ARY_SCH_BYTE .
 - (2) IN_ARY가 10 Array Array 22h
 - (3) POS Array 1 , NUM 3
- 1 %Q1.3.0 On .

ARY_SFT_C_***

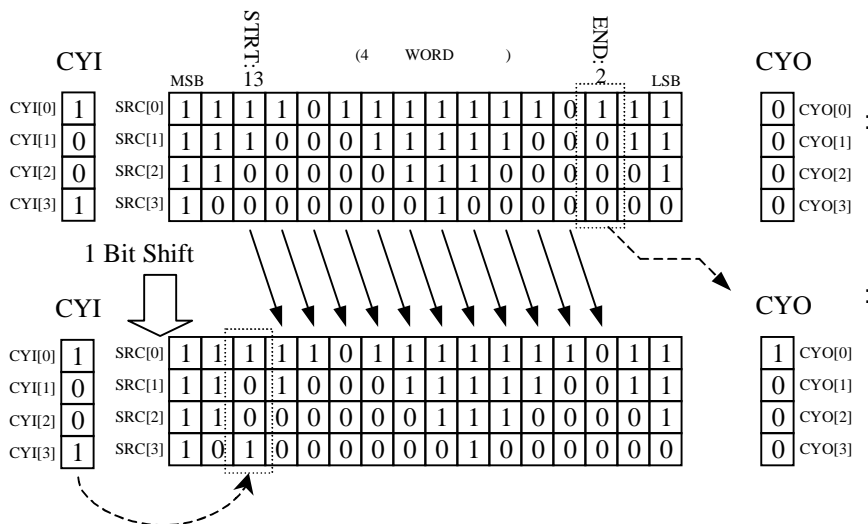
Array Bit Shift Left with Carry

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



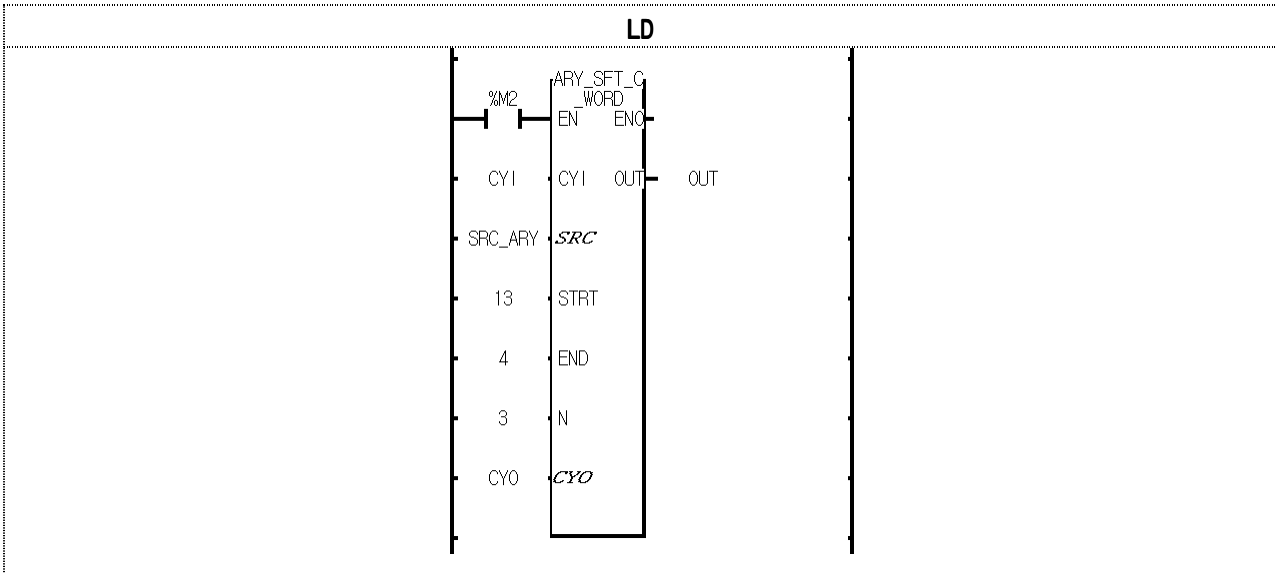
ARY_SFT_C_*** Array bit

- : STRT END
- : STRT END (N)
- : Shift (CYI)
- : SRC ANY_BIT_ARRAY, END Shift



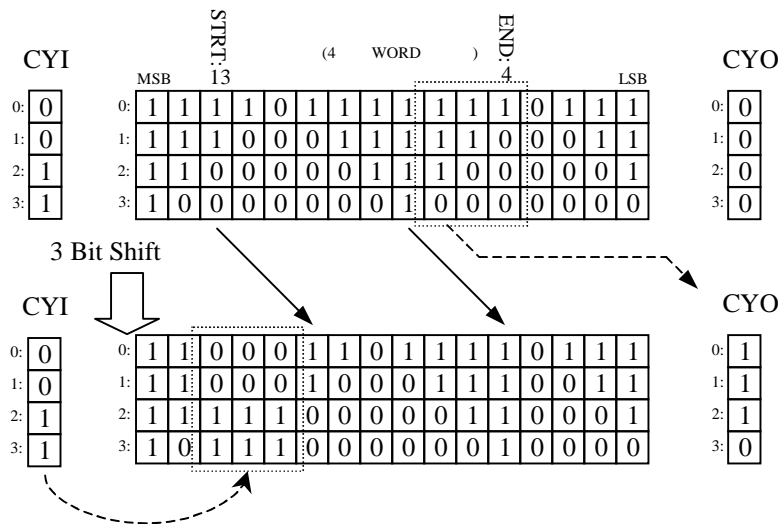
	Array	
ARY_SFT_C_BYTE	BYTE	Array Shift
ARY_SFT_C_WORD	WORD	
ARY_SFT_C_DWORD	DWORD	
ARY_SFT_C_LWORD	LWORD	

CYI, SRC, CYO Array 가 _ERR/_LER 가 (Set)
 STRT END가 SRC 가
 SRC CYO



- (1) (%M2) On , ARY_SFT_C_WORD
- (2) STRT 13 bit END 4 STRT END 3 shift
- (3) Shift CYI (2#0011)
- (4) Shift SRC_ARY CYO

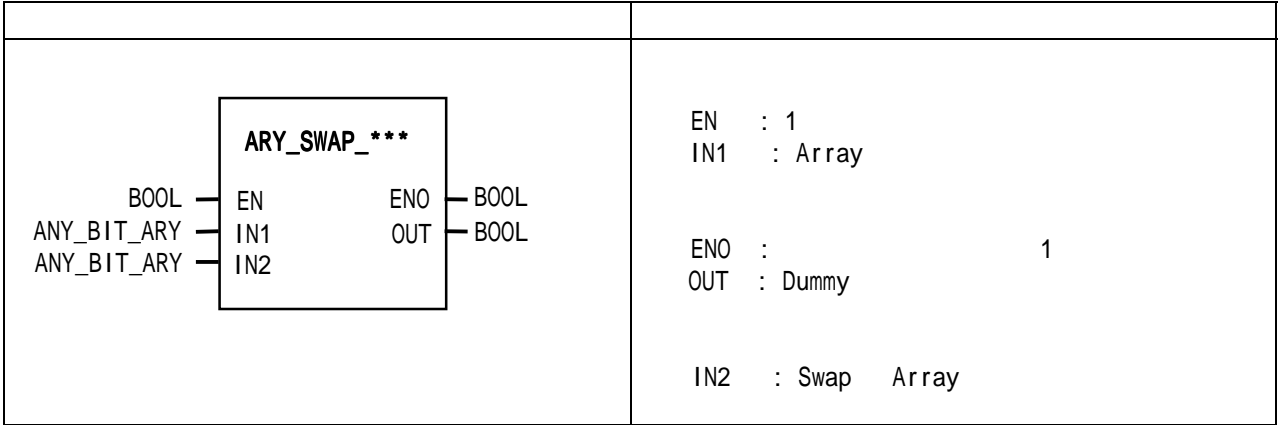
()
 CYI : 2#0011
 SRC_ARY : 16#F7F7
 16#E3E3
 16#C1C1
 16#8080
 (N) : 3
 ()
 SRC_ARY : 16#C6F7
 16#C473
 16#F831
 16#B810
 CYO : 2#1110



ARY_SWAP_***

Array

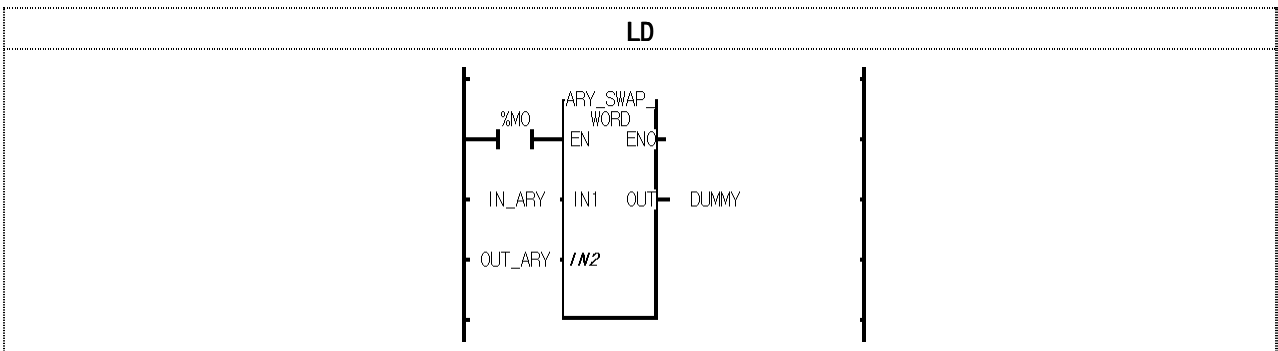
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



Array 2

ARY_SWAP_BYTE	BYTE	BYTE	(Nibble)
ARY_SWAP_WORD	WORD	WORD	BYTE
ARY_SWAP_DWORD	DWORD	DWORD	WORD
ARY_SWAP_LWORD	LWORD	LWORD	DWORD

2 Array 가 , IN2 Array 가 _ERR/_LER 가 (Set) .



- (1) (%M0) On , ARY_SWAP_WORD
- (2) IN_ARRAY

IN_ARRAY[0]	12ABH
IN_ARRAY[1]	23BCH
IN_ARRAY[2]	34CDH

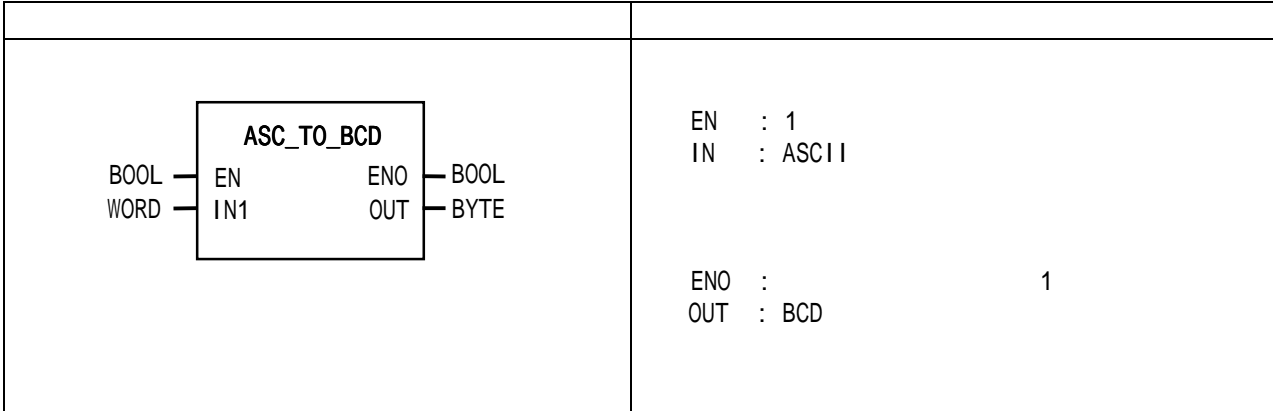
OUT_ARY

OUT_ARY[0]	AB12H
OUT_ARY[1]	BC23H
OUT_ARY[2]	CD34H

ASC_TO_BCD

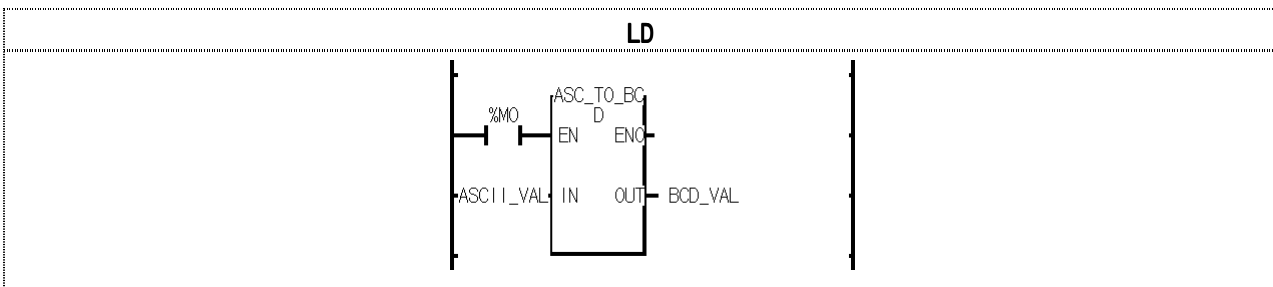
ASCII BCD

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



2 ASCII 2 BCD(Binary Coded Decimal)

IN 16 '0' '9' , 16#00 _ERR/_LER 가 (Set)

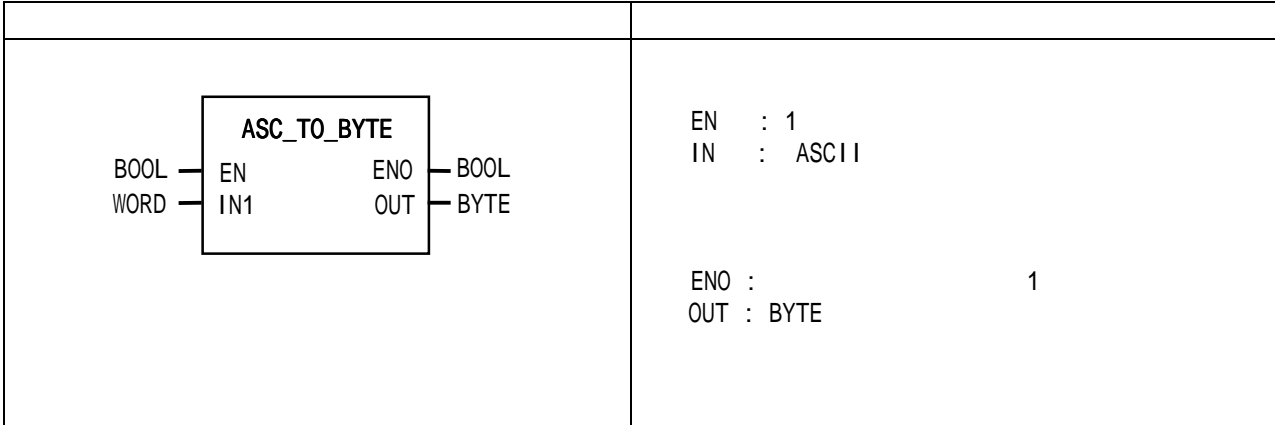


- (1) (%M0) On , ASC_TO_BCD
- (2) ASCII_VAL(WORD)=16#3732="72" ,
BCD_VAL(BYTE)=16#72 가

ASC_TO_BYTE

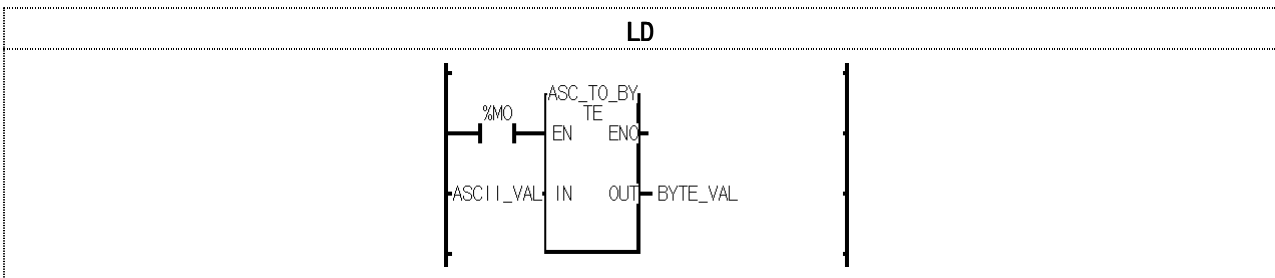
ASCII BYTE

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



2 ASCII 2 16 (HEX)

IN 16 '0' 'F' , 0 _ERR/_LER 가 (Set)

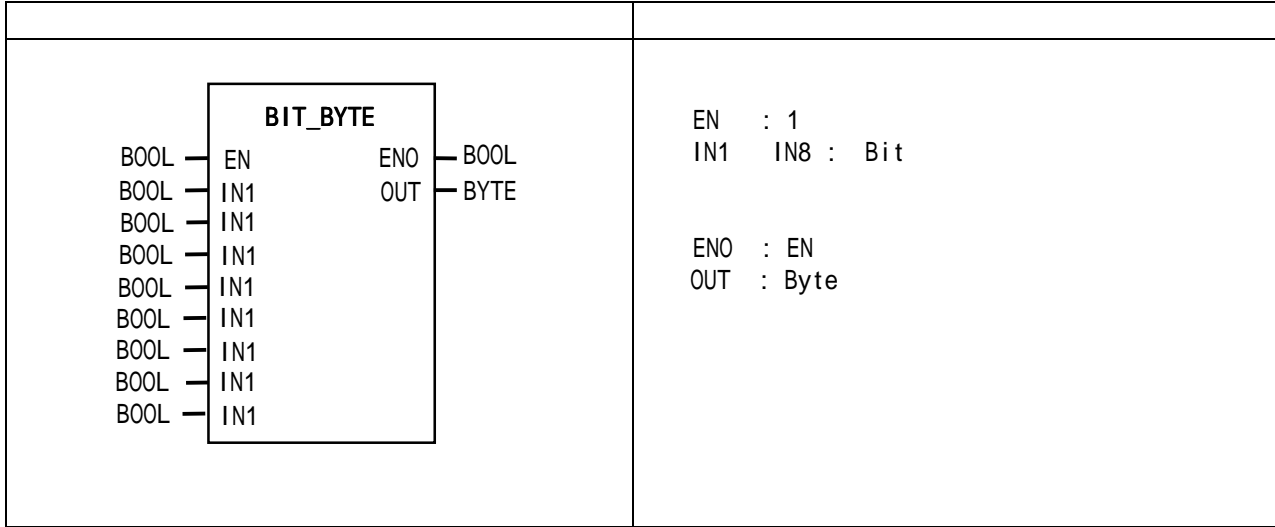


- (1) (%M0) On , ASC_TO_BYTE
- (2) ASCII_VAL(WORD)=16#4339 ,
BYTE_VAL(BYTE) = 16#C9가

BIT_BYTE

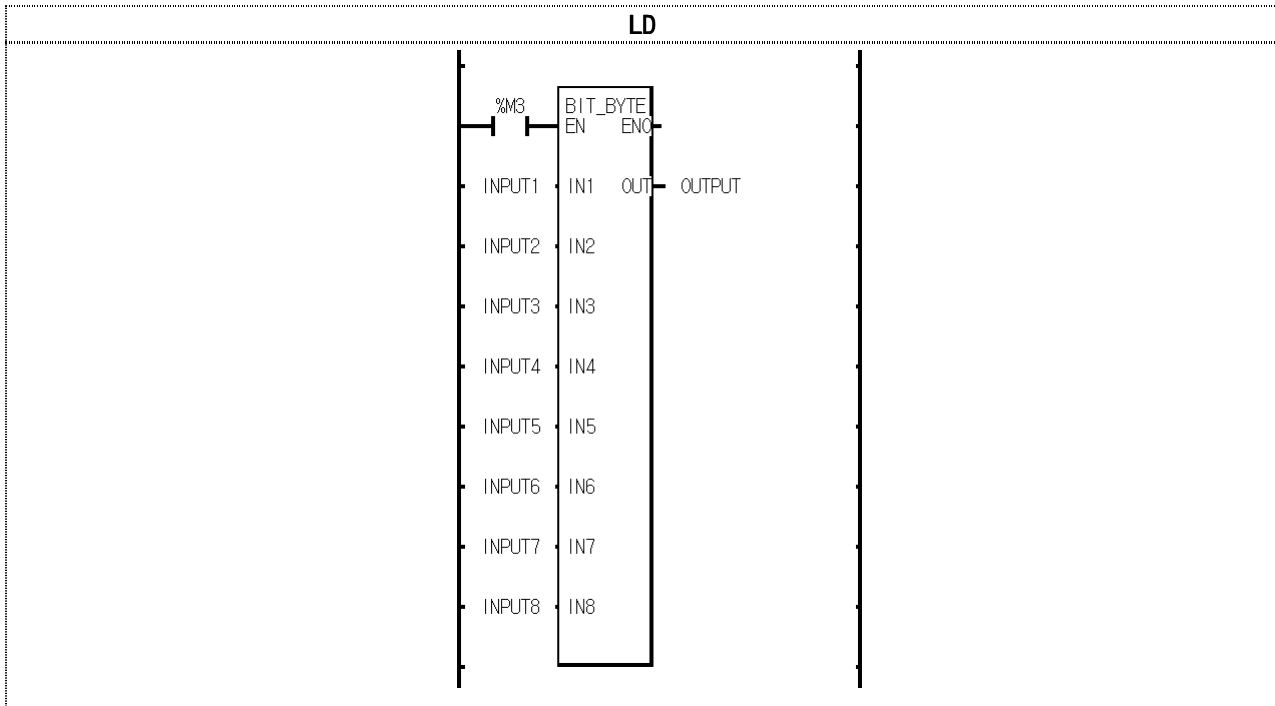
8 BIT BYTE

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



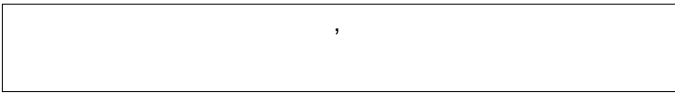
8

IN8: MSB(), IN1: LSB()

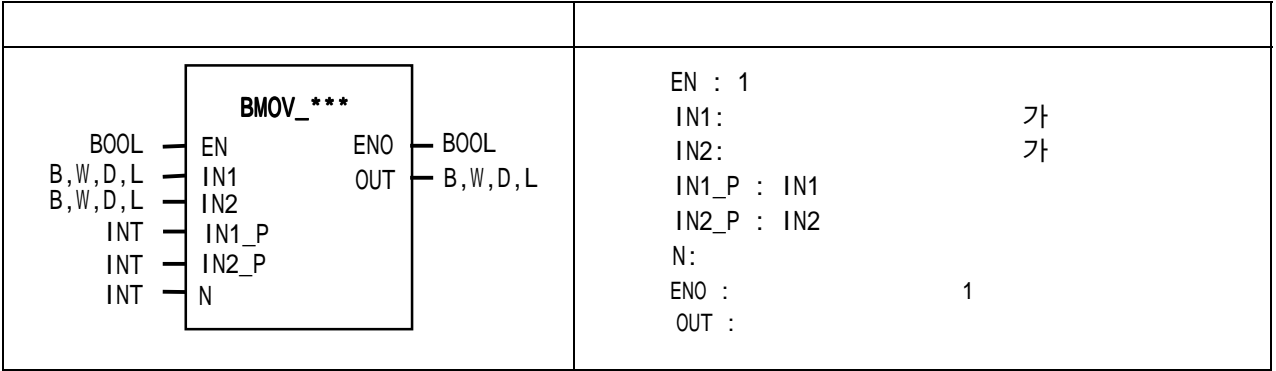


(1) (%M3) On , BIT_BYTE
 (2) 8 INPUT1 8 {0,1,1,0,1,1,0,0} ,
 OUTPUT(BYTE)= 2#00110110

BMOV_***



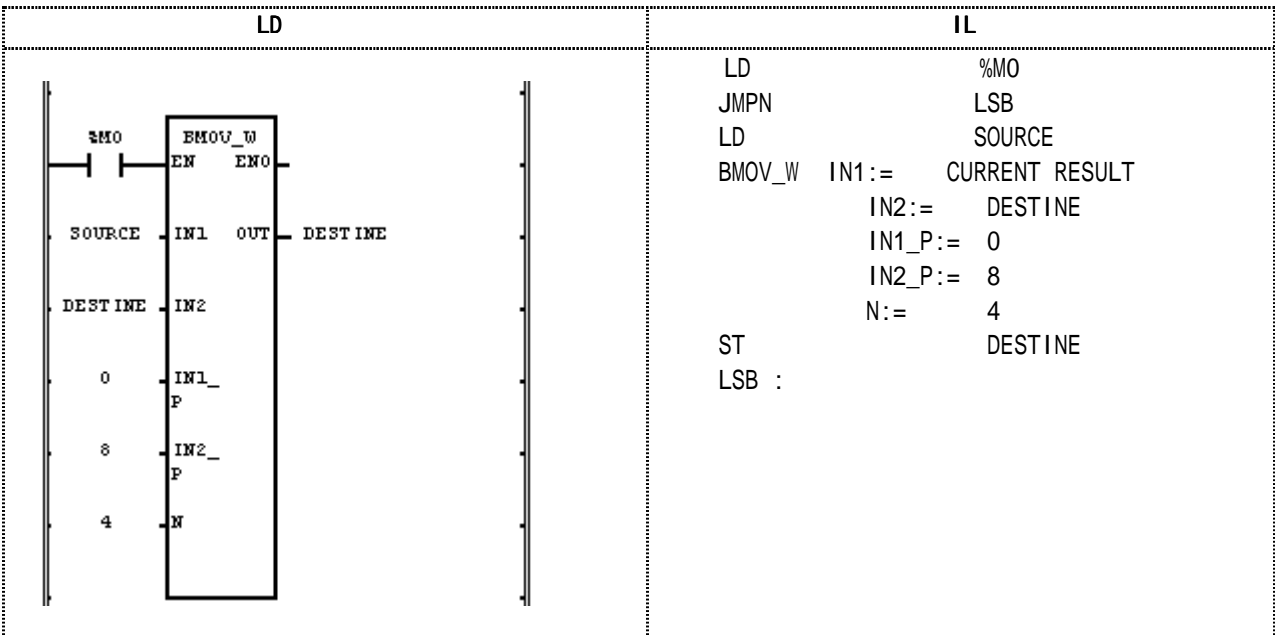
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1 IN1 IN1_P N
 , IN2 IN2_P OUT

IN1 = 1111 0000 1111 0000, IN2 = 0000 1010 1010 1111 IN1_P = 4, IN2_P = 8, N = 4 ,
 OUT = 0000 1111 1010 1111 B(BYTE), W(WORD), D(DWORD),
 L(LWORD) 가 가 , L(LWORD) GM1,2
 'ENCO_B', 'ENCO_W', 'ENCO_D', 'ENCO_L'

IN1_P, IN2_P , N IN1_P, IN2_P N
 , _ERR,_LER 가 (Set)



- (1) (%M0) On BMOV_W .
- (2) SOURCE = 2#0101 1111 0000 1010, DESTINE = 2#0000 0000 0000 0000 , IN1_P
 = 0, IN2_P = 8, N = 4 2#0000 1010 0000 0000 , DESTINE
 DESTINE = 2#0000 1010 0000 0000 .

(IN1) : SOURCE(WORD) = 16#5F0A

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

(IN2) : DESTINE(WORD) = 16#0000

(IN1_P) = 0

(IN2_P) = 8

(N) = 4

↓ (BMOV_W)

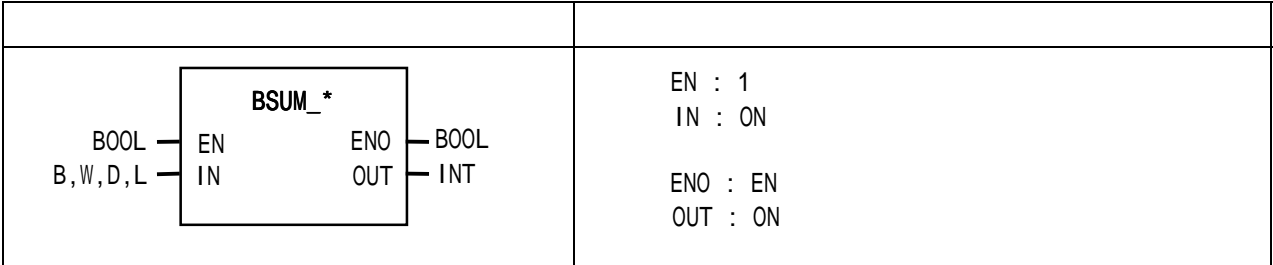
(OUT) : DESTINE(WORD) = 16#0A00

0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

BSUM_***

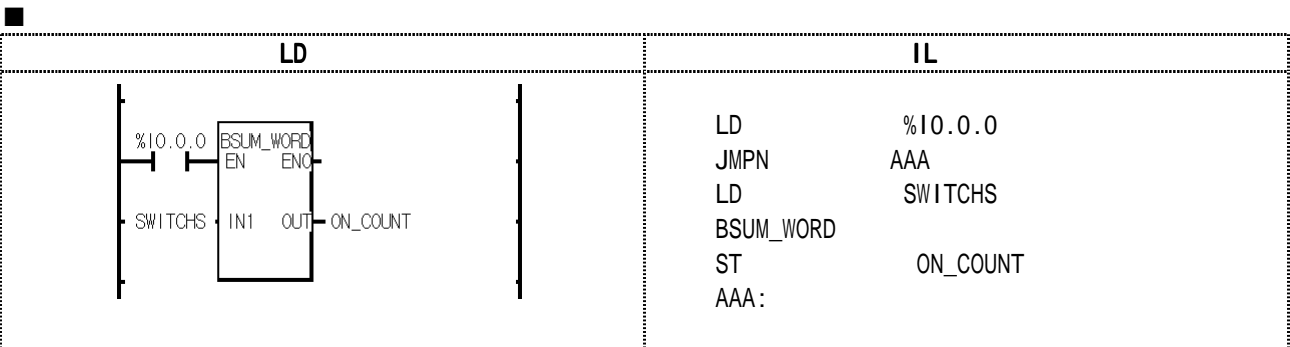
ON

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1, IN BYTE, WORD, DWORD, LWORD, 1 가 가, LWORD GM1,2

FUNCTION	IN	
BSUM_BYTE	BYTE	4가
BSUM_WORD	WORD	
BSUM_DWORD	DWORD	
BSUM_LWORD	LWORD	

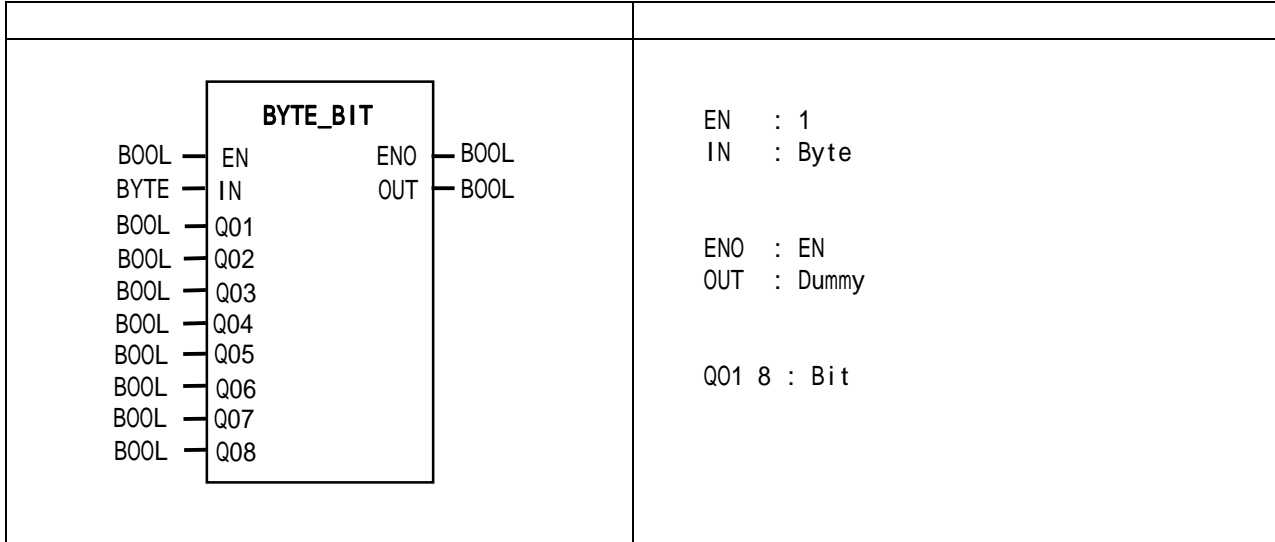


(1) (%M0) On BSUM_WORD
 (2) SWITCHS(WORD) = 2#0000 0100 0010 1000, ON
 '3' ON_COUNT(INT) '3'

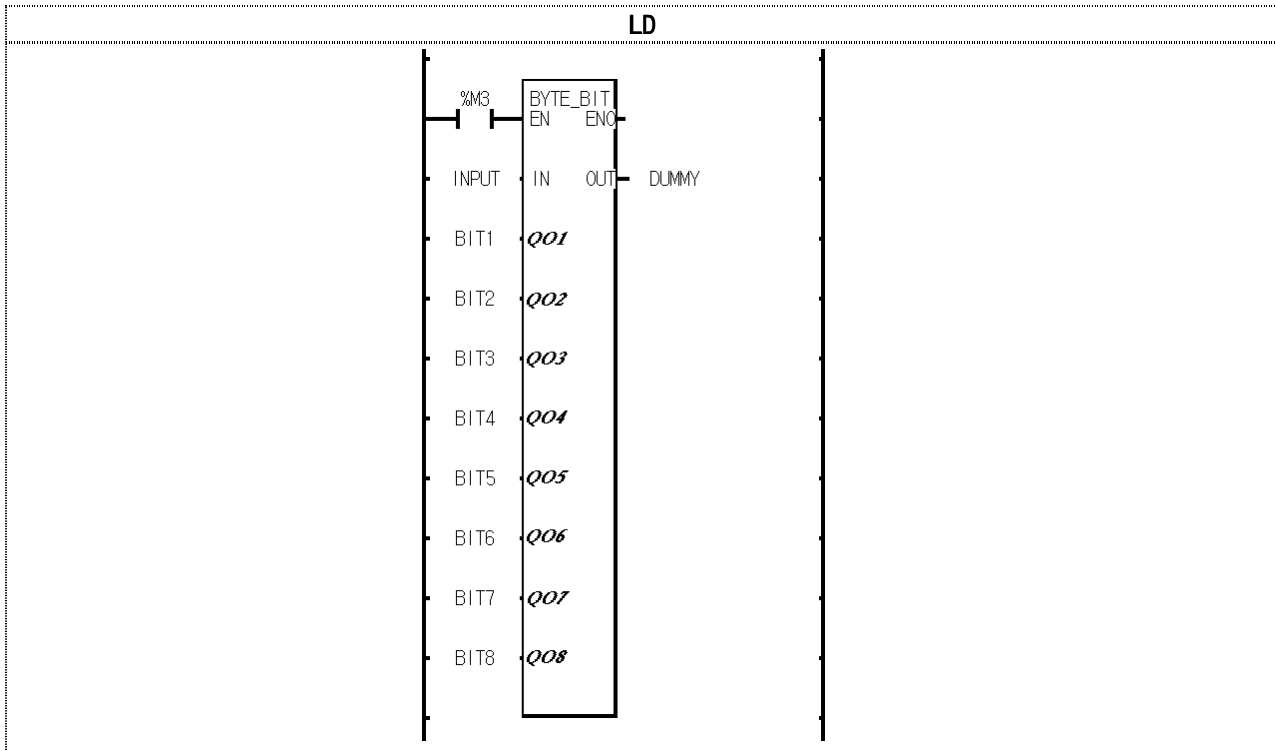
BYTE_BIT

BYTE	8	BIT
------	---	-----

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



1 Q08: MSB(), Q01: LSB() (Q01 Q02)

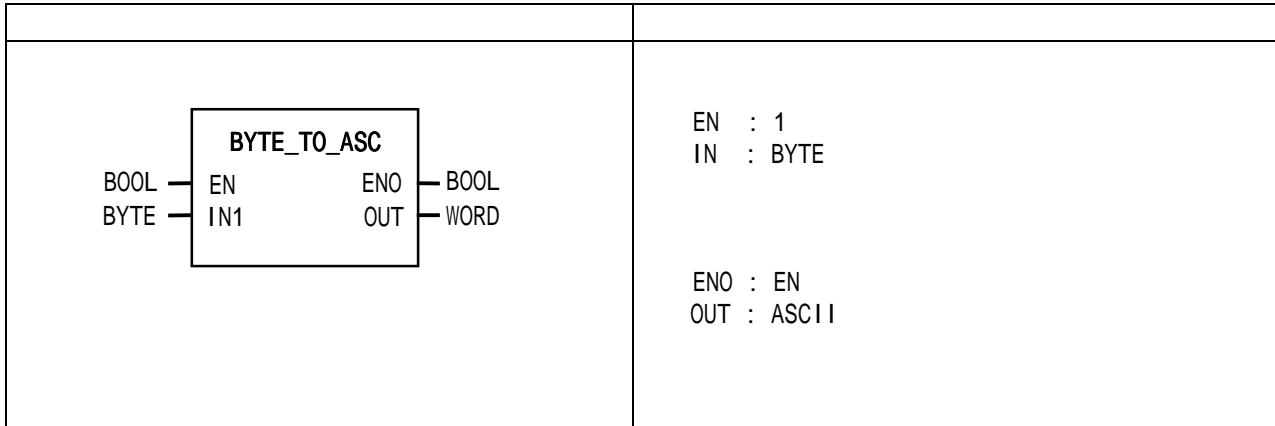


- (1) (%M0) On , BYTE_BIT
- (2) INPUT=16#AC=2#10101100 , Q01 8 Q01
2#{0, 0, 1, 1, 0, 1, 0, 1}

BYTE_TO_ASC

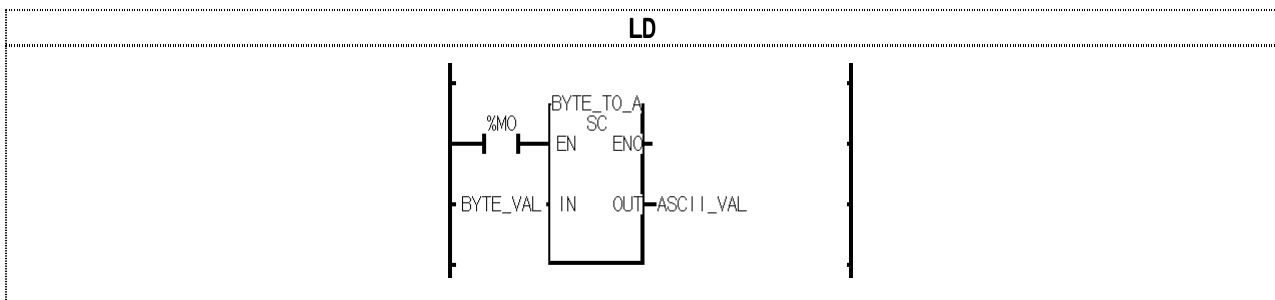
BYTE ASCII

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



2 16 (HEX)
) 16#12 -> 3132
16#A F

2

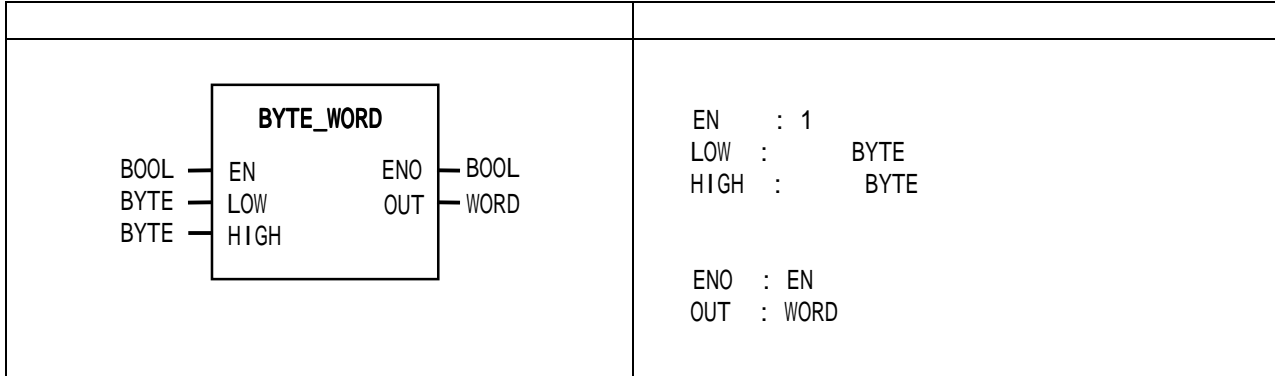


- (1) (%M0) On BYTE_TO_ASC
- (2) BYTE_VAL(BYTE) =16#3A ,
ASCII_VAL(WORD) = 16#3341 = '3','A'가

BYTE_WORD

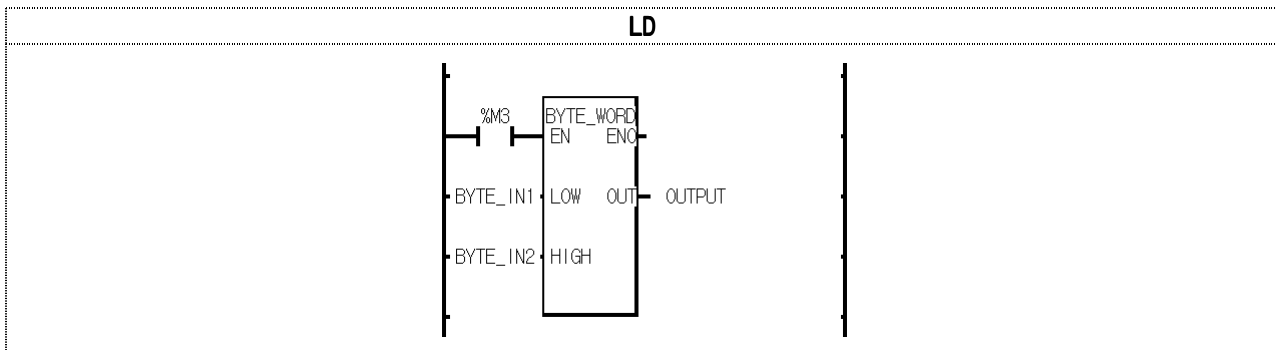
2	BYTE	WORD
---	------	------

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



2

LOW: , HIGH:

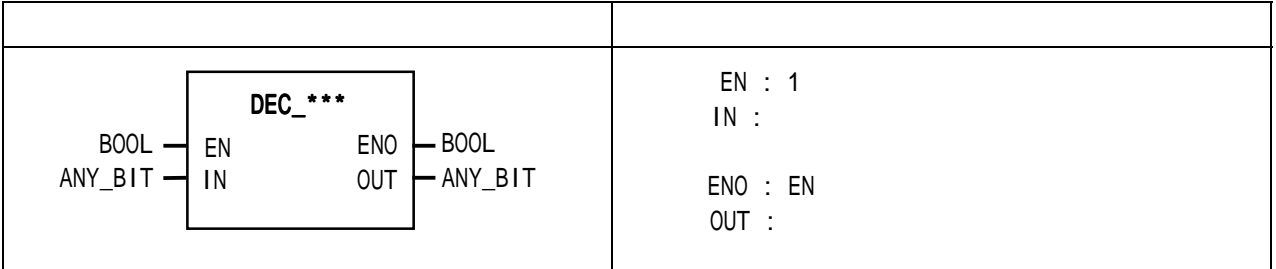


- (1) (%M3) On , BYTE_WORD
- (2) BYTE_IN1=16#56 BYTE_IN2=16#AD ,
OUTPUT=16#AD56

DEC_***

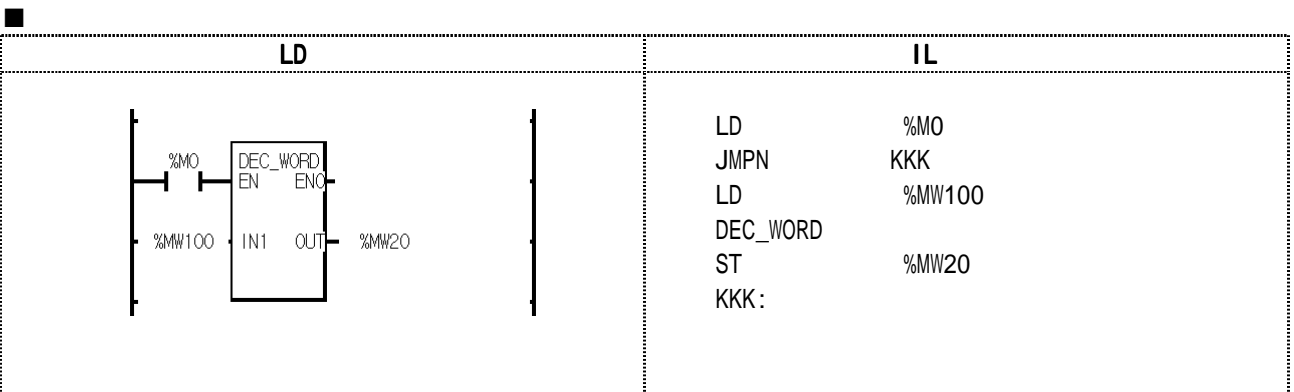
IN

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1 , IN 1 , OUT .
가 , 16#0000 , 16#FFFF가 .
BYTE, WORD, DWORD, LWORD 가 가 , LWORD GM1,2 .

FUNCTION	IN/OUT	
DEC_BYTE	BYTE	4가
DEC_WORD	WORD	
DEC_DWORD	DWORD	
DEC_LWORD	LWORD	

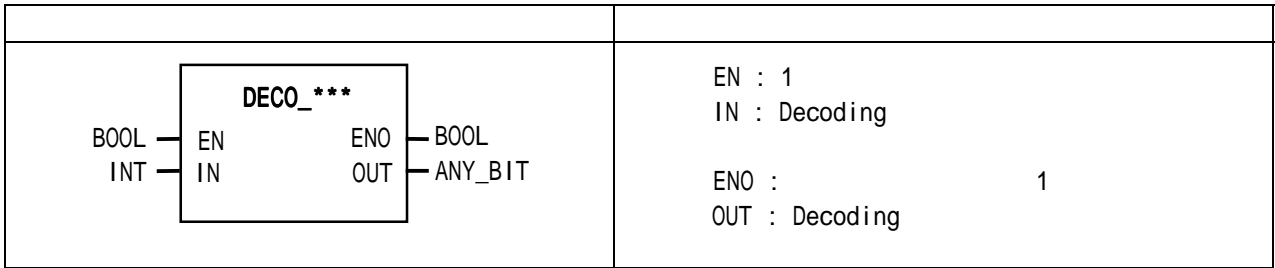


- (1) (%M0) On DEC_WORD
- (2) %MW100 = 16#0007(2#0000 0000 0000 0111) ,
%MW20 = 16#0006(2#0000 0000 0000 0110)

DECO_***



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



■

EN 1, IN 1

BYTE, WORD, DWORD, LWORD

가 가, LWORD GM1,2

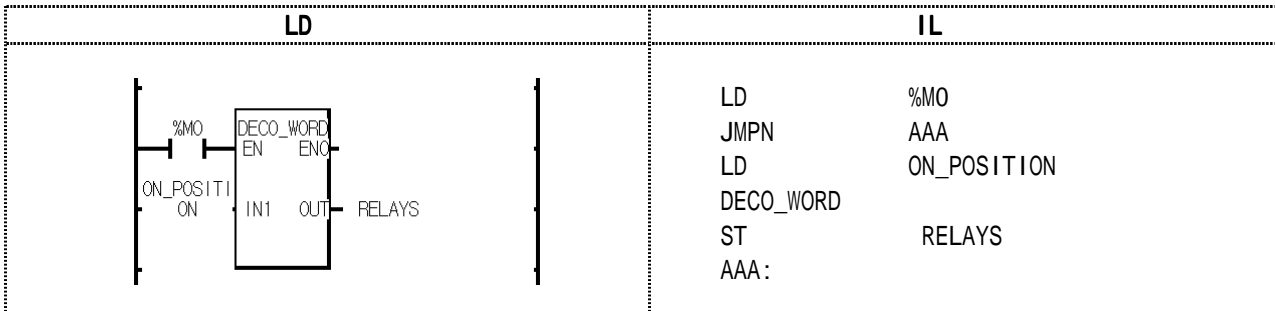
FUNCTION	OUT	
DECO_BYTE	BYTE	4가
DECO_WORD	WORD	
DECO_DWORD	DWORD	
DECO_LWORD	LWORD	

■

가, OUT 0, _ERR, _LER 가

(DECO_WORD 16

■

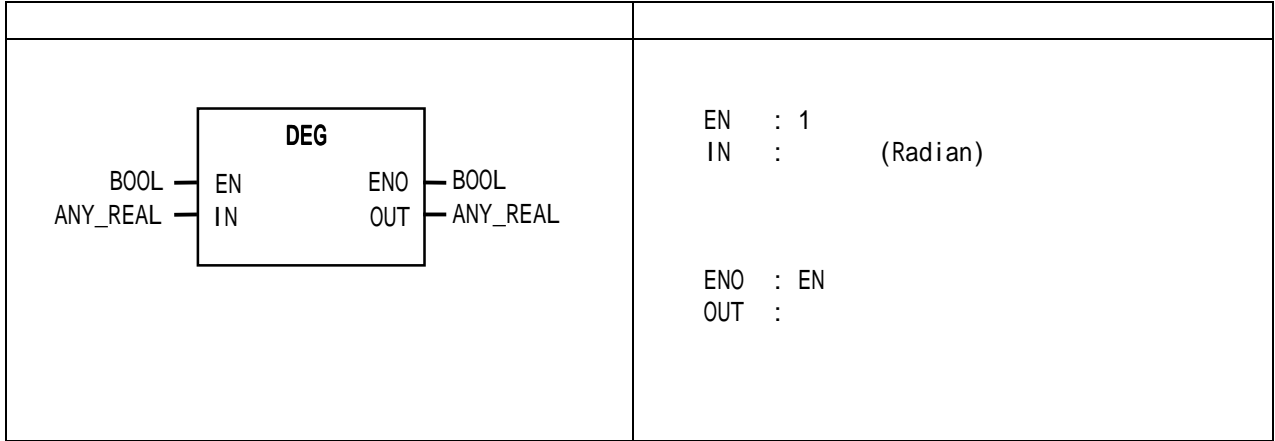


- (1) (%MO) On DECO_WORD
- (2) ON_POSITION(IN) = 5, 5 ON, RELAYS(WORD) = 2#0000 0000 0010 0000

DEG_***

Radian

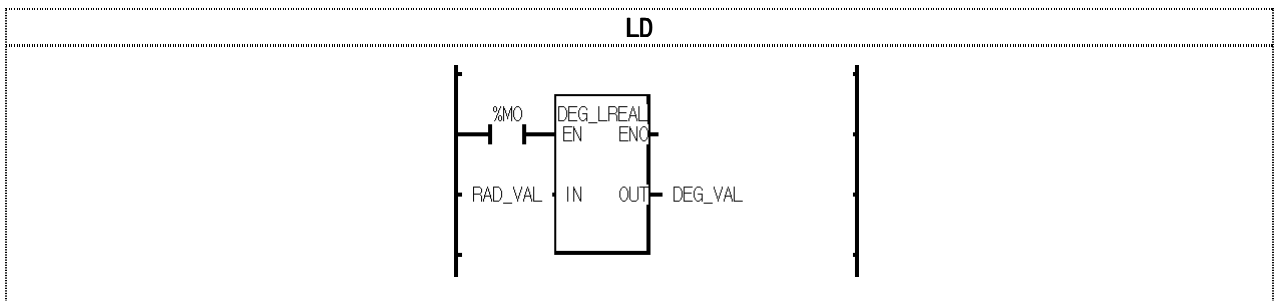
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



(Radian)

(Degree)

DEG_REAL	REAL	REAL	(Radian)
DEG_LREAL	LREAL	LREAL	

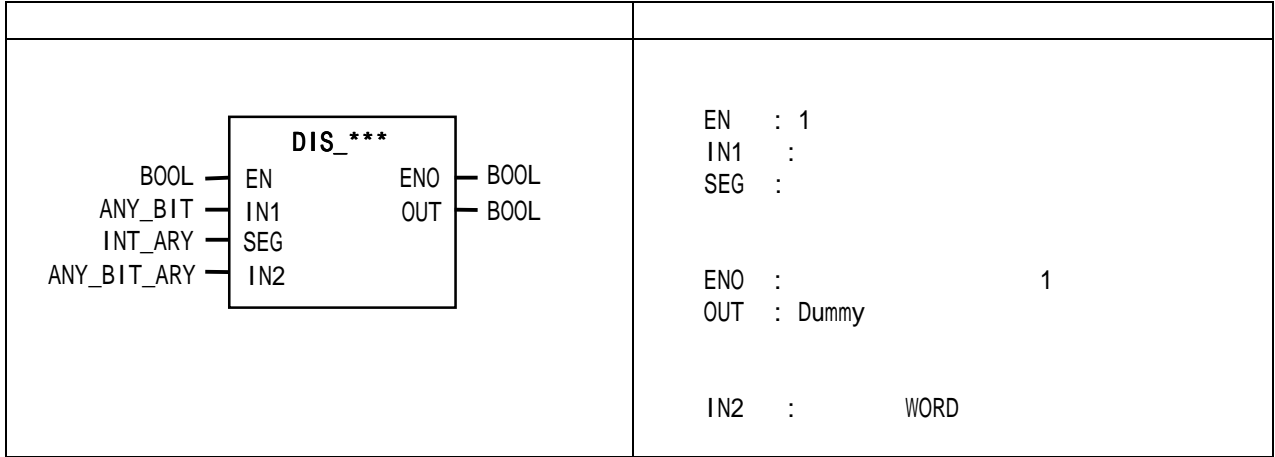


- (1) (%M0) On , DEG_LREAL
- (2) RAD_VAL=1.0 ,
DEG_VAL=5.7295779513078550e+001 가

DIS_***

(Distribution)

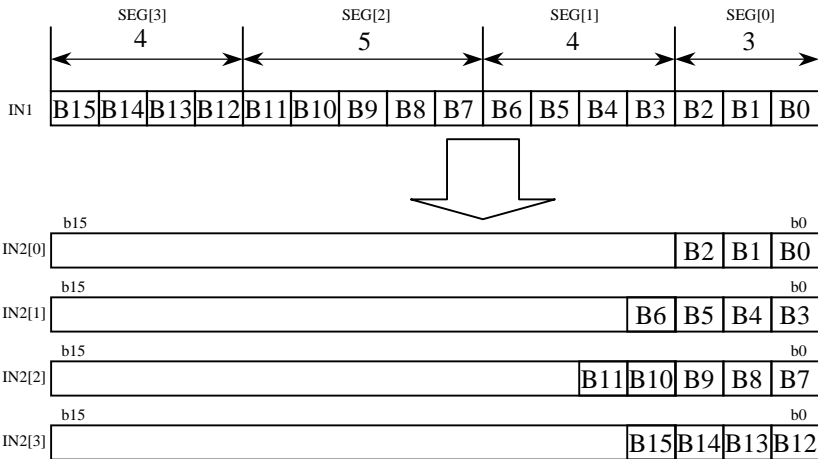
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



SEG

IN2

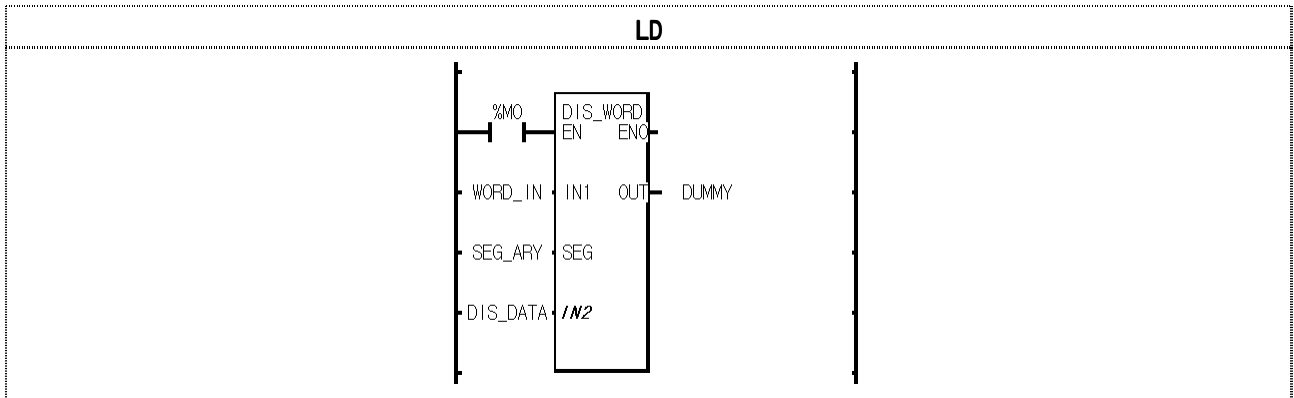
DIS_BYTE	BYTE			
DIS_WORD	WORD			
DIS_DWORD	DWORD	IN1	IN1	SEG
DIS_LWORD	LWORD			IN2



SEG

_ERR/_LER

가 (Set)

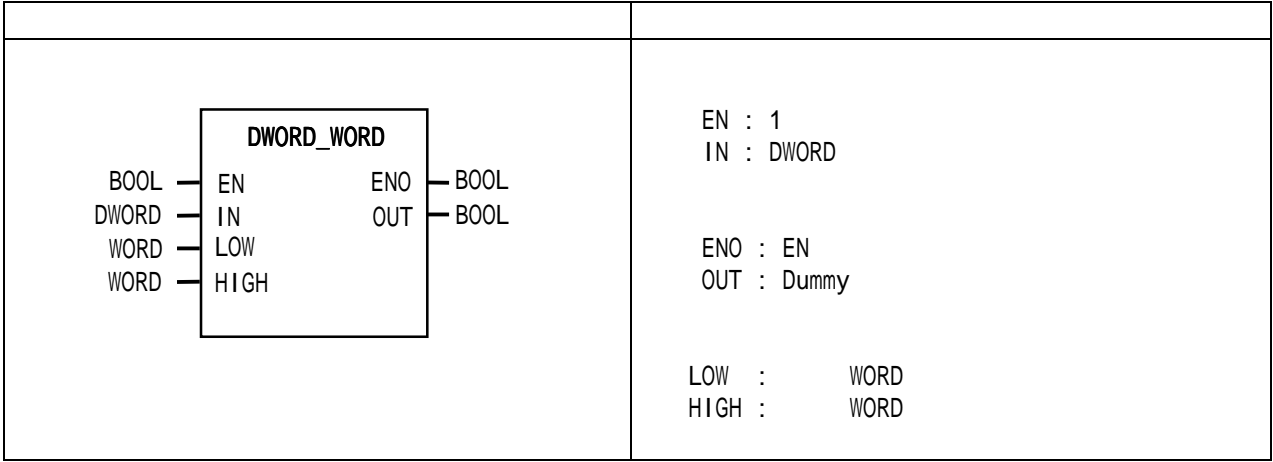


```
(1)      (%M0)  On      , DIS_WORD
(2)      WORD_IN      16#3456      ,SEG_ARY={3,4,5,4}      ,
DIS_DATA
DIS_DATA[0]=16#0003
DIS_DATA[1]=16#000A
DIS_DATA[2]=16#0008
DIS_DATA[3]=16#0003
가      .
```

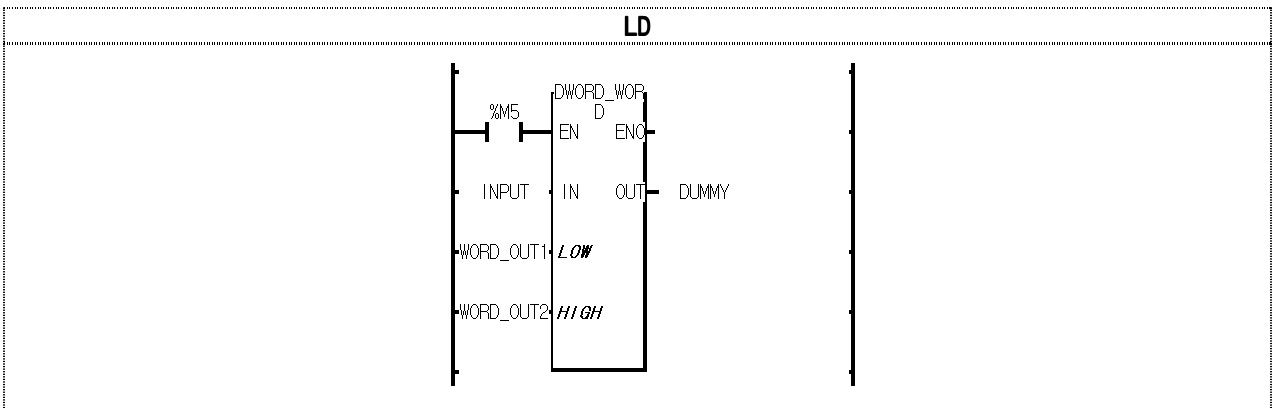

DWORD_WORD

DWORD	2	WORD
-------	---	------

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



DWORD 2 WORD .
LOW: , HIGH:

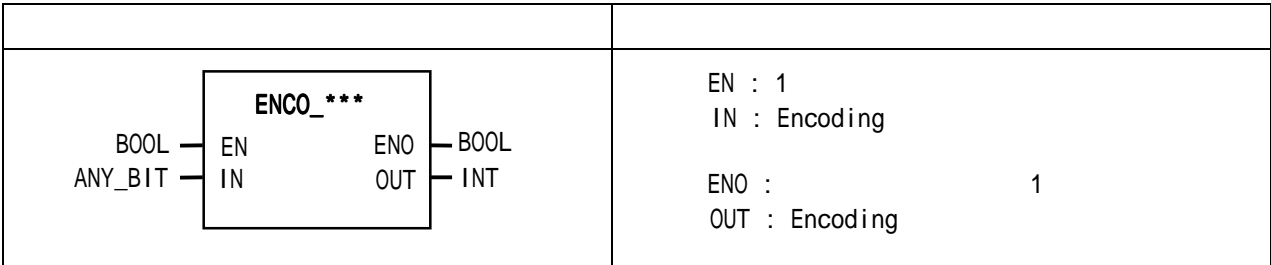


- (1) (%M5) On , DWORD_WORD .
- (2) INPUT=16#11223344AABBCCDD ,
WORD_OUT1=16#AABBCCDD
WORD_OUT2=16#11223344 .

ENCO_***

ON

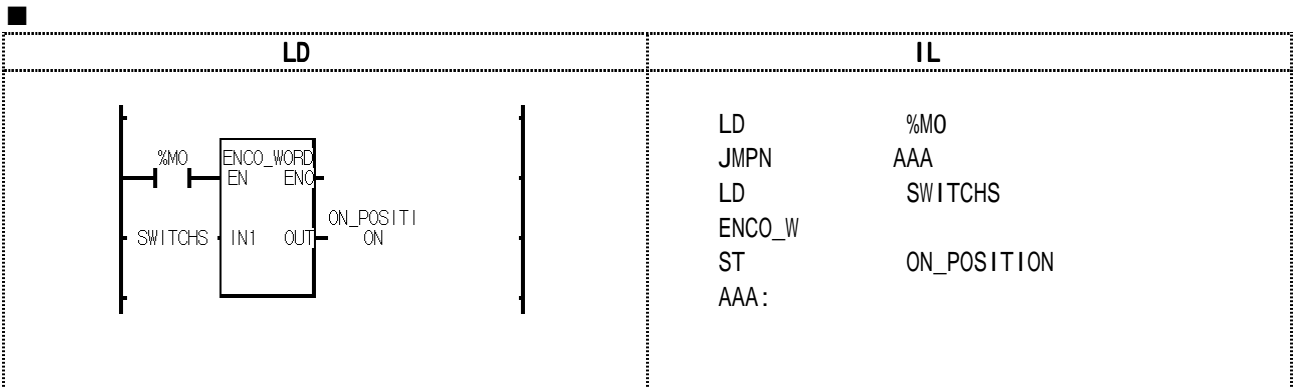
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1, IN, 1, OUT
 B(BYTE), W(WORD), D(DWORD), L(LWORD) 가 가, L(LWORD) GM1,2

FUNCTION	IN	
ENCO_BYTE	BYTE	ENCO
ENCO_WORD	WORD	
ENCO_DWORD	DWORD	
ENCO_LWORD	LWORD	

1, OUT -1, _ERR,_LER 가 (Set)

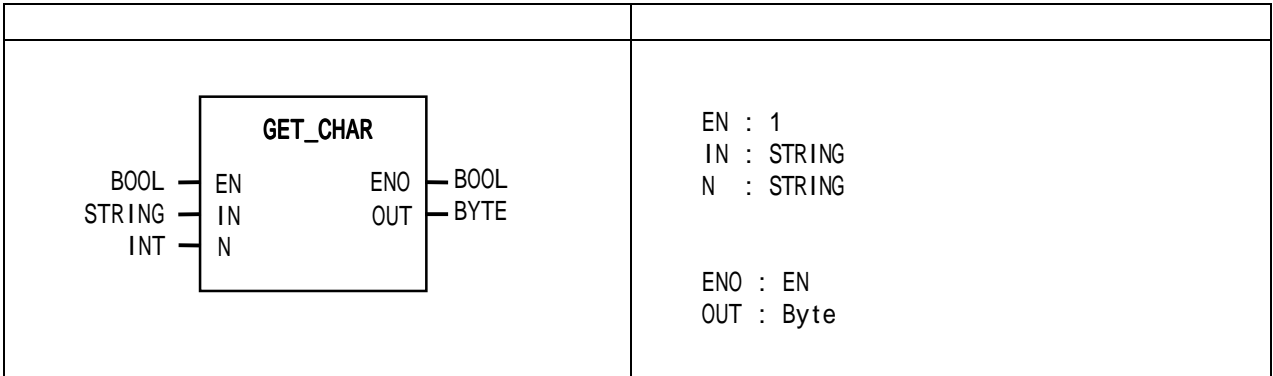


(1) (%MO) On ENCO_WORD
 (2) SWITCHS(WORD) = 2#0000 1000 0000 0010, ON 2, '11' '1'
 '11' ON_POSITION(INT)

GET_CHAR

(CHAR)

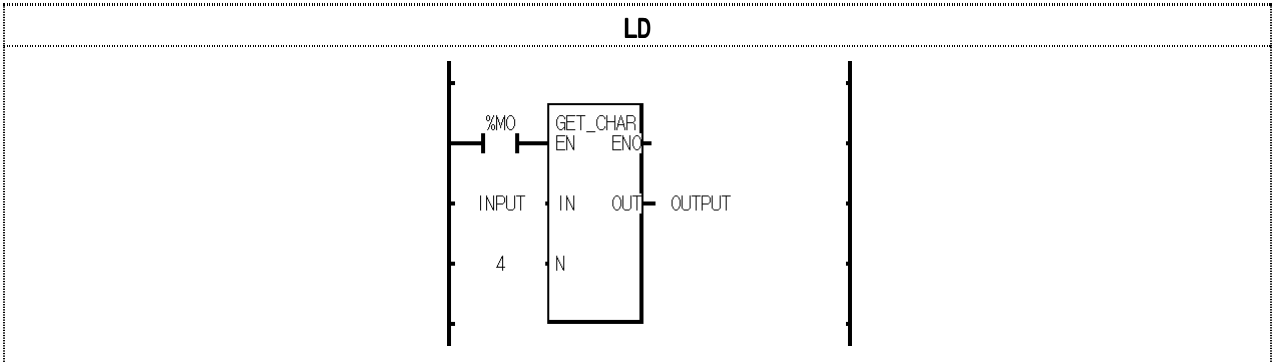
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



STRING

N 가 _ERR/_LER 가 (set) .

가 16#00 .

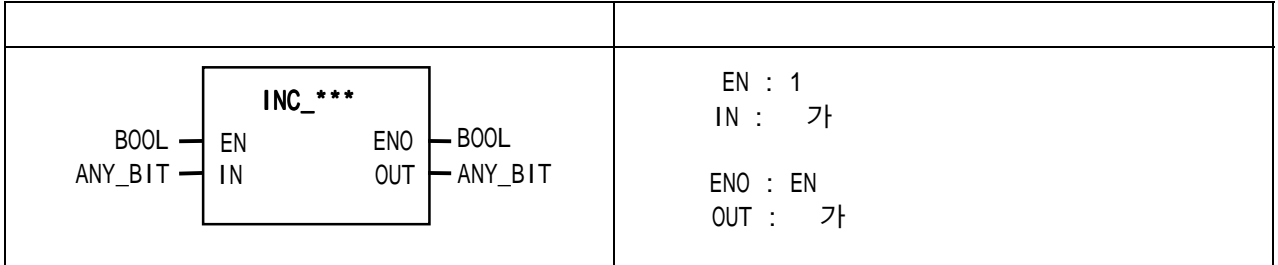


- (1) (%MO) On , GET_CHAT
- (2) INPUT(STRING)="LG GLOFA PLC" String 4
- OUTPUT 16#47("G")가

INC_***

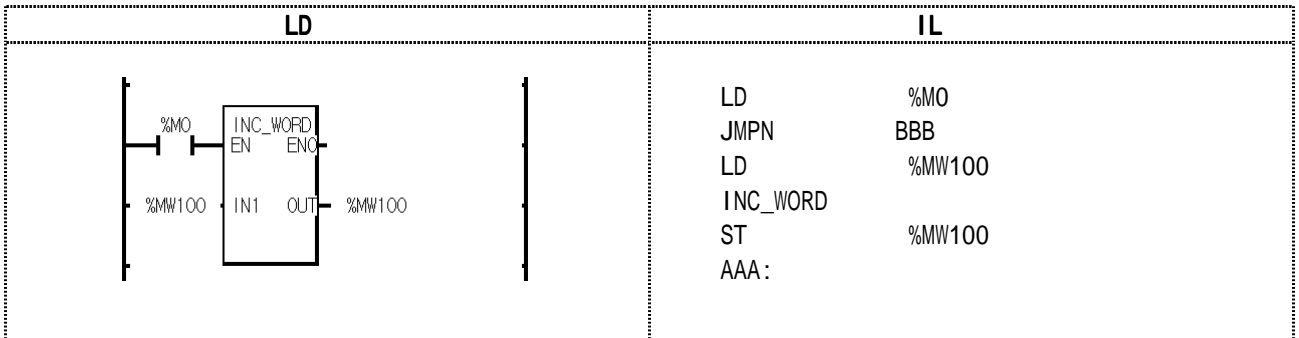
IN	가
----	---

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

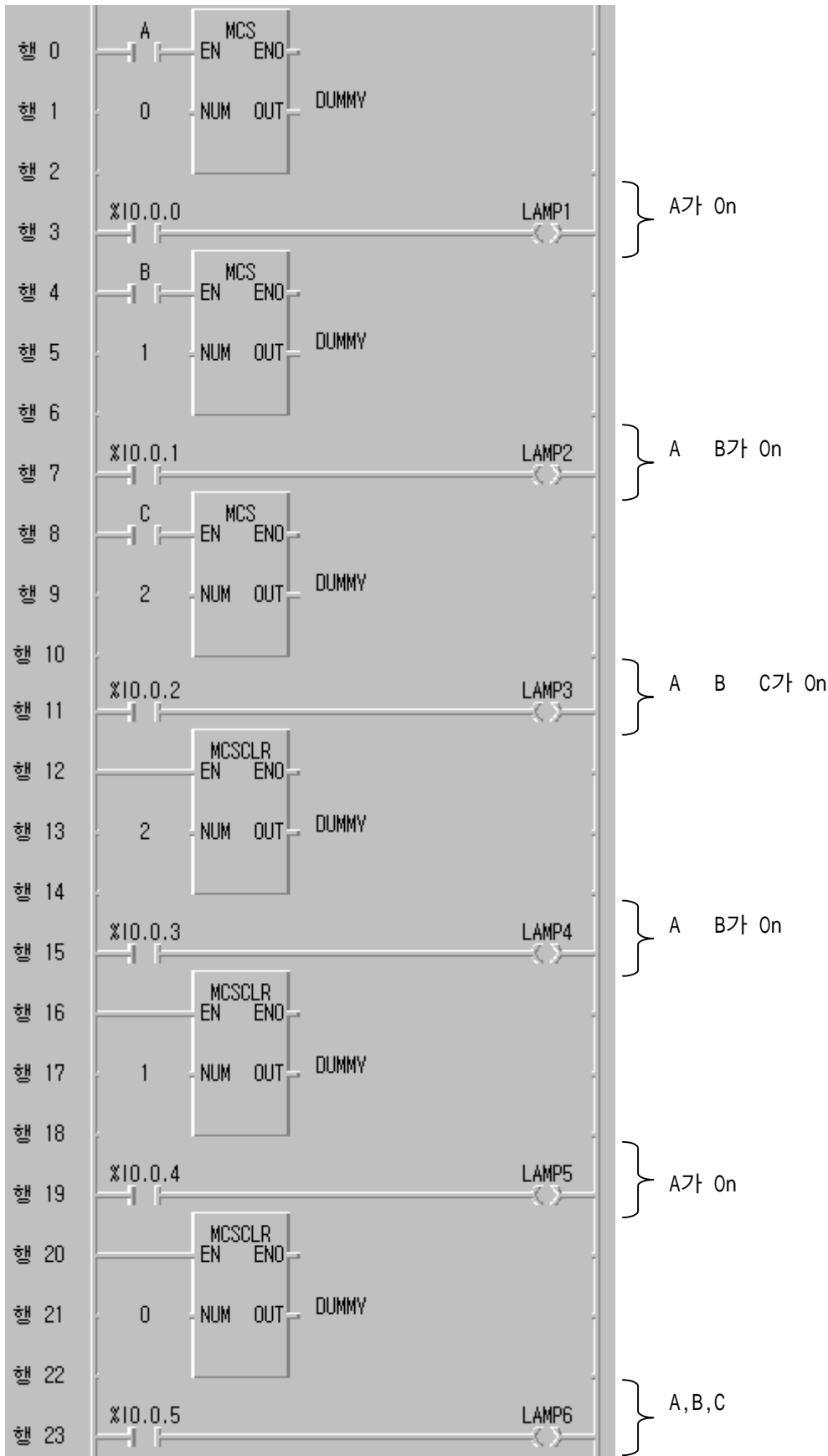


EN 1 , IN 가 , OUT 16#FFFF , 16#0000가 .
 BYTE, WORD, DWORD, LWORD 가 , LWORD GM1,2 .

FUNCTION	IN/OUT	
INC_BYTE	BYTE	4가
INC_WORD	WORD	
INC_DWORD	DWORD	
INC_LWORD	LWORD	



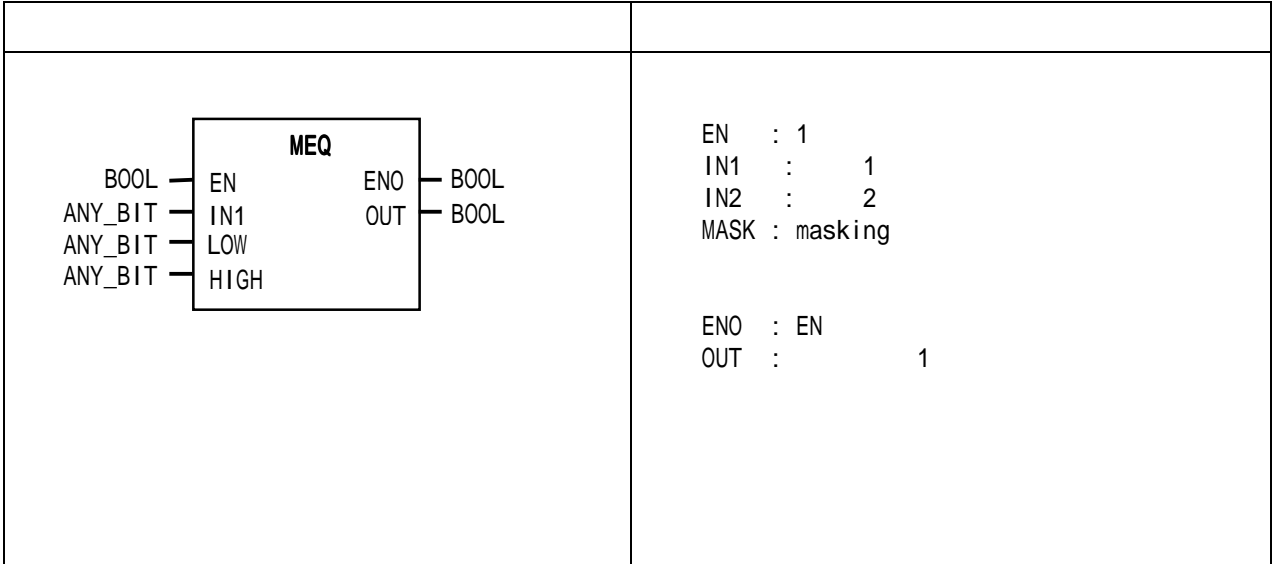
- (1) (%MO) On INC_WORD
- (2) %MW100 = 16#0007(2#0000 0000 0000 0111) ,
 %MW100 = 16#0008(2#0000 0000 0000 1000)



MEQ_***

Masked Equal

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



Masking . 8

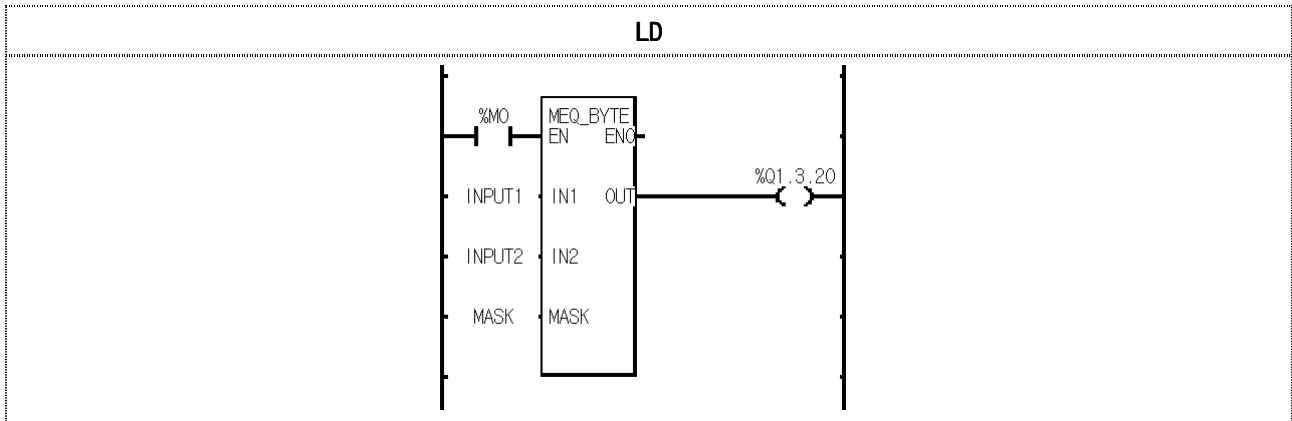
2#11111100 Masking 2 .

ON 가 , 8

IN1 IN2 16#FF MASK

(i.e. 2#00101100) ON .

MEQ_BYTE	BYTE	Masking
MEQ_WORD	WORD	
MEQ_DWORD	DWORD	
MEQ_LWORD	LWORD	

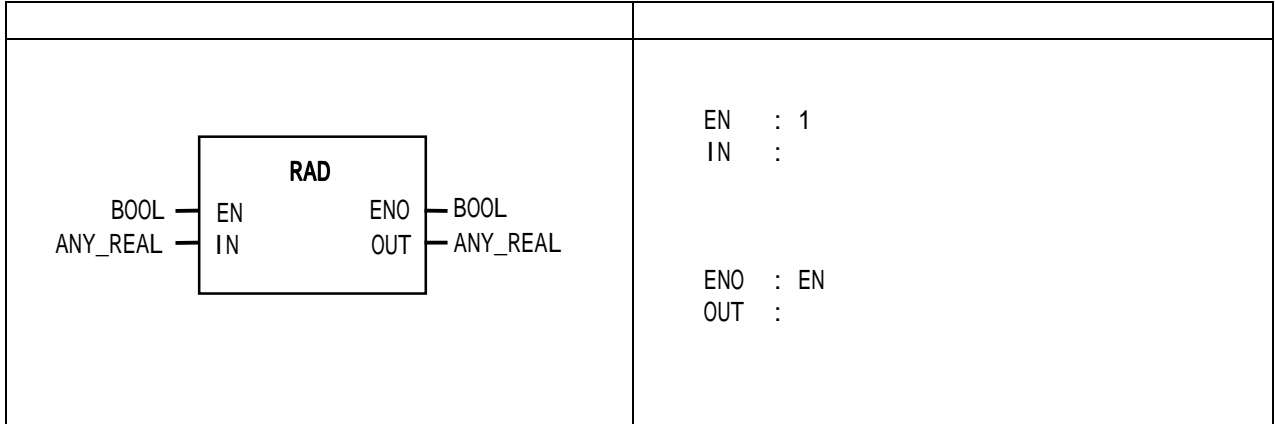


- (1) (%M0) On , MEQ_BYTE .
- (2) INPUT1(BYTE)= 2#01011100
 INPUT2(BYTE)= 2#01110101
 MASK(BYTE) = 2#11010110 Making
 INPUT1(BYTE) = 2#01010100
 INPUT2(BYTE)= 2#01010100
 가 %Q1.3.20 On

RAD_***

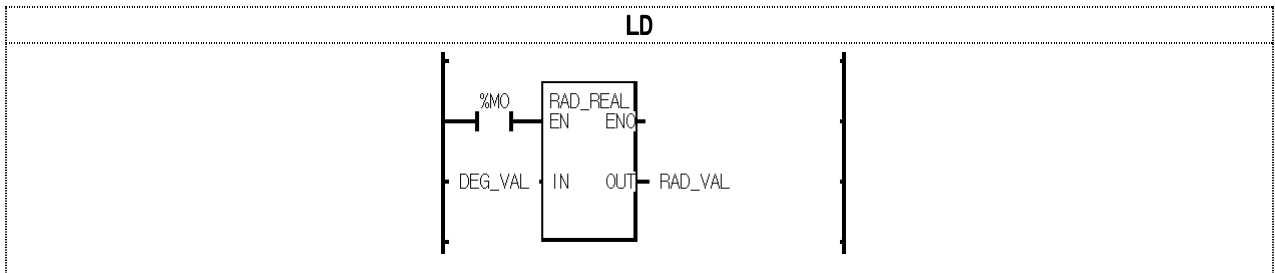
(DEG) Radian

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



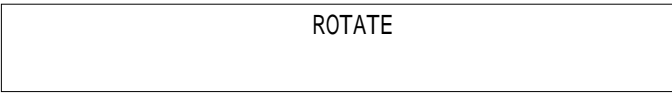
가 360° (°) (Radian) .(, 370° 360°
10° .)

RAD_REAL	REAL	REAL	(°)
RAD_LREAL	LREAL	LREAL	.

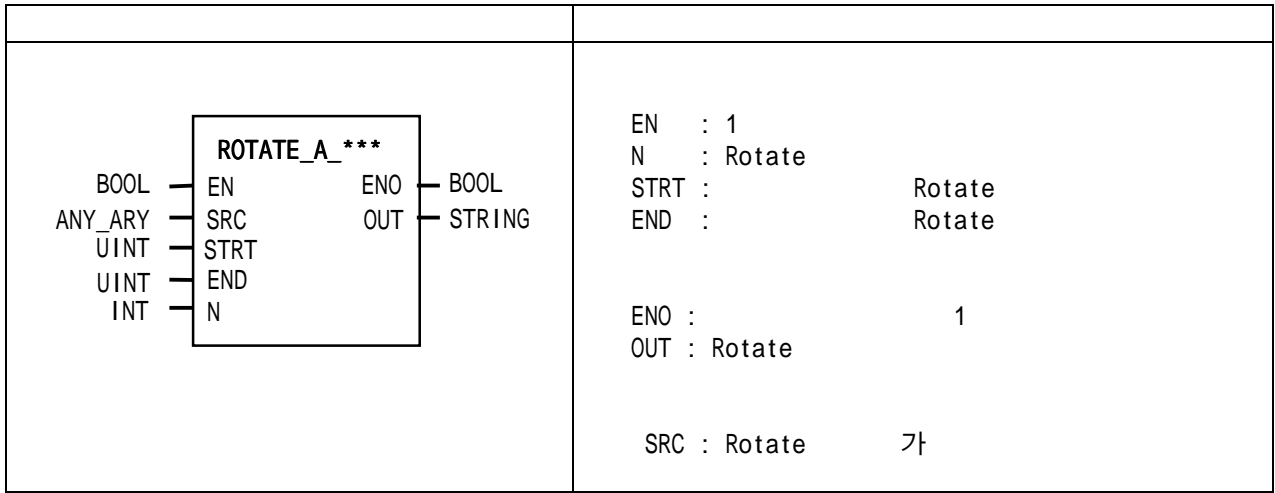


- (1) (%M0) On , RAD_REAL .
- (2) DEG_VAL=127(°) , RAD_VAL=2.21656823

ROTATE_A_***

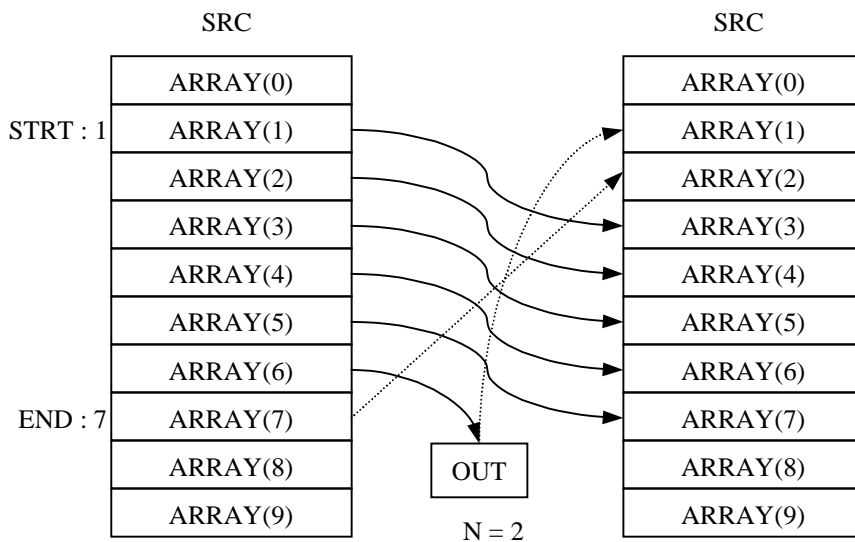


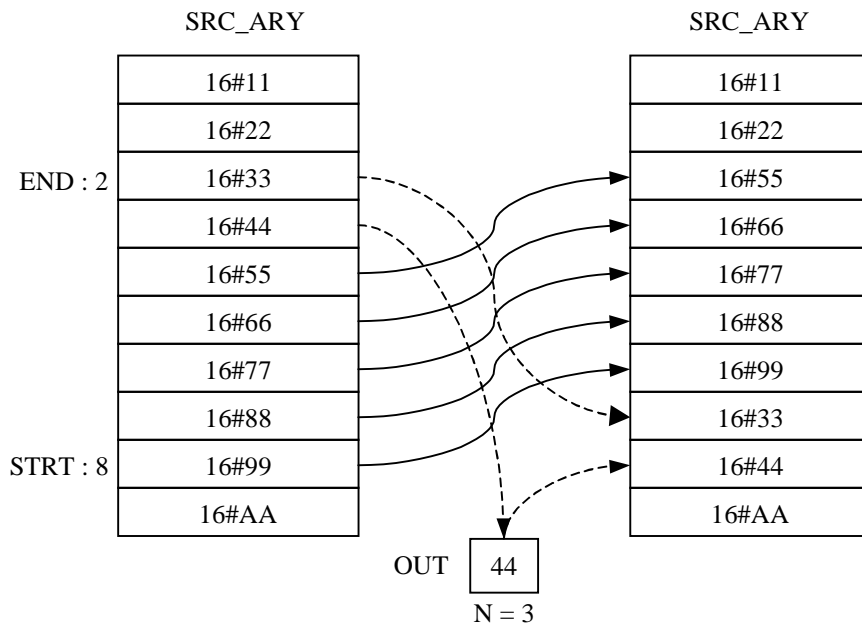
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



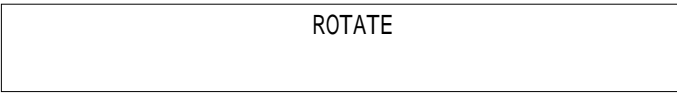
ROTATE_A_***

- : STRT END
 - : STRT END (N) Rotate
 - : Rotate (IN)
 - : SRC ANY_ARY , END Rotate
- STRT 가 OUT

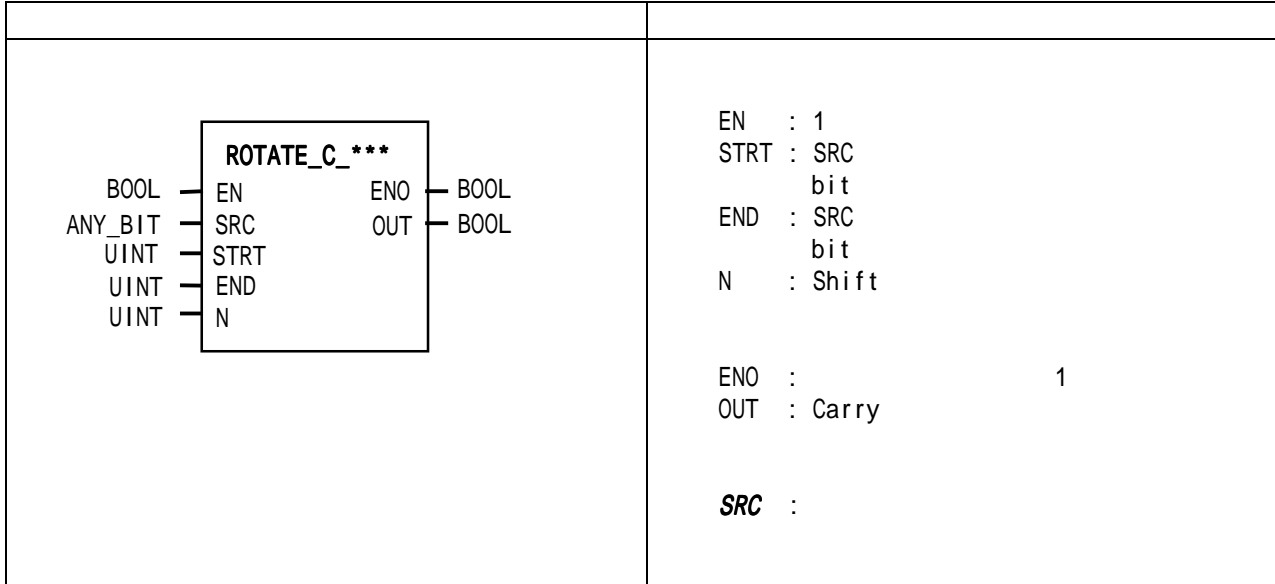




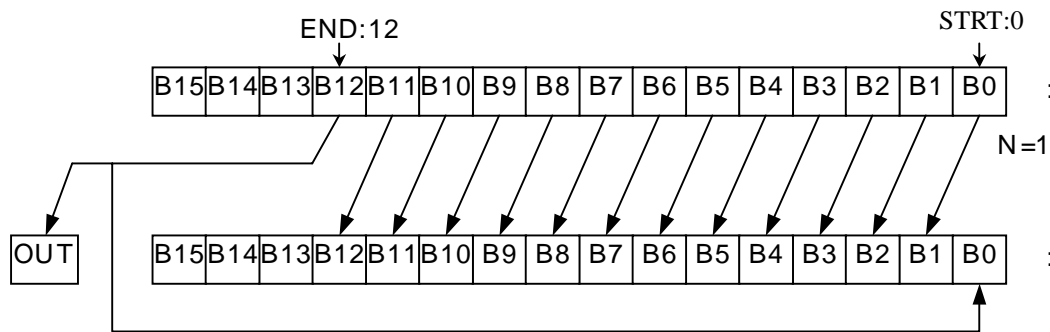
ROTATE_C_***



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

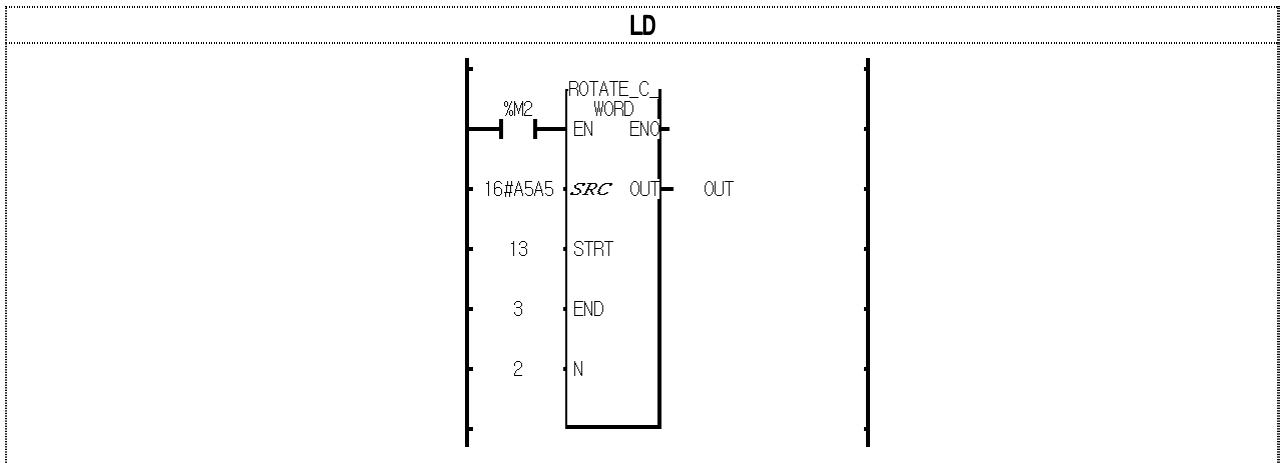


SRC
 :
 - : STRT END
 - : START END (N)
 - : SRC ANY_BIT END
 STRT 가 OUT

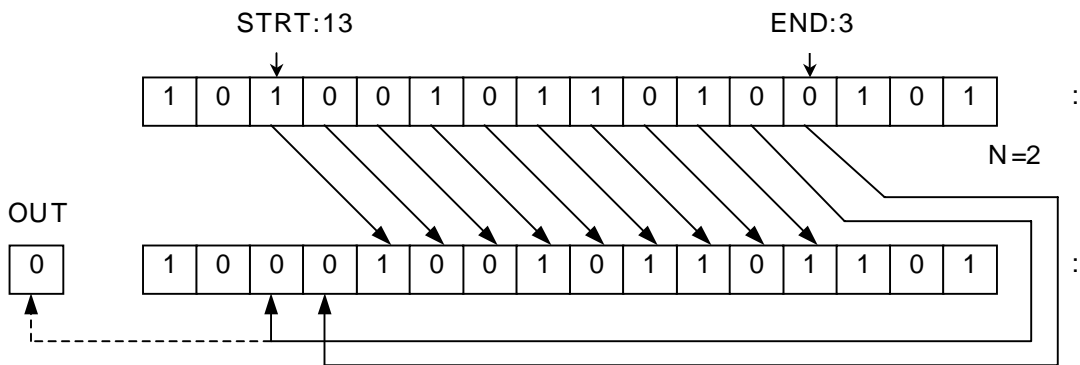


	SRC	
ROTATE_C_BYTE	BYTE	Rotate
ROTATE_C_WORD	WORD	
ROTATE_C_DWORD	DWORD	
ROTATE_C_LWORD	LWORD	

START END SRC _ERR/_LER 가 (Set)
 SRC 가



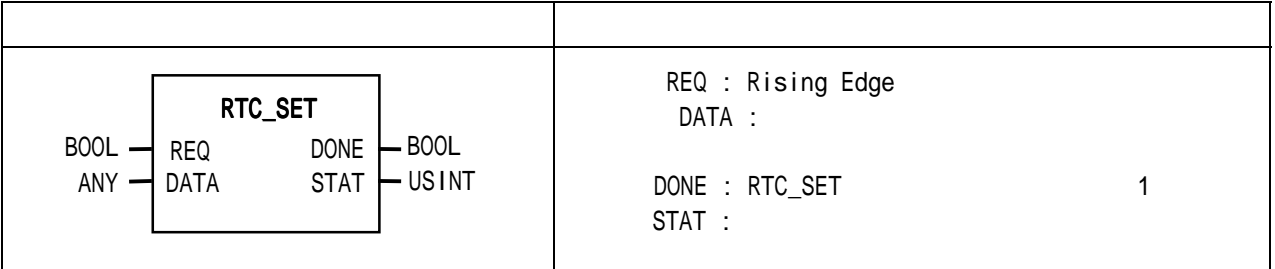
(1) (%M2) On , ROTATE_C_WORD
 (2) 16#A5A5 STRT(13) END(3) STRT END 2 STRT 가
 (3) Rotate SRC(16#896D) END STRT 가
 0 OUT



RTC_SET



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



REQ Rising Edge Setting RTC (DATA) PLC Clock Device

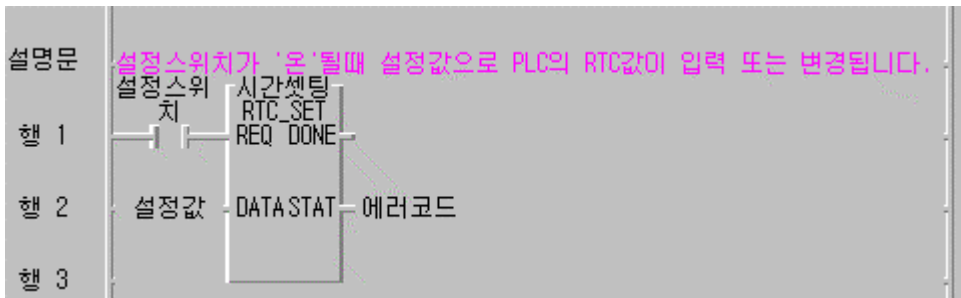
DATA[0]		16#01	DATA[4]		16#30
DATA[1]		16#03	DATA[5]		16#45
DATA[2]		16#15	DATA[6]		16#03
DATA[3]		16#18	DATA[7]		16#20

* 2001 3 15 () 18 30 45
 * : 0, : 1, : 2, : 3, : 4, : 5, : 6
 DATA BYTE BCD Setting

CPU가 RTC , RTC 가 (DONE) '0' STAT

00	No error
01	RTC * GM6 GM6-CPUB, GM6-CPUC RTC * GM7 G7E-RTCA
02	RTC) 14() 32() 25() * RTC

RTC 가 1999. 1. 17. 11:53:24,

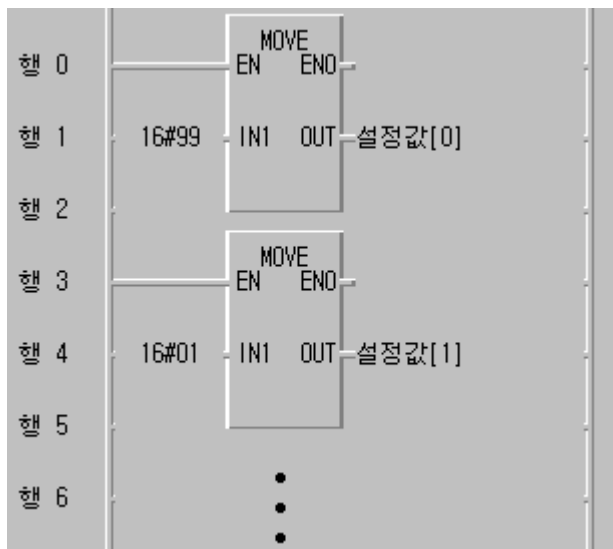


(1) 가 ' ' PLC RTC

(2) ()



(3) MOVE ()



(4) RTC

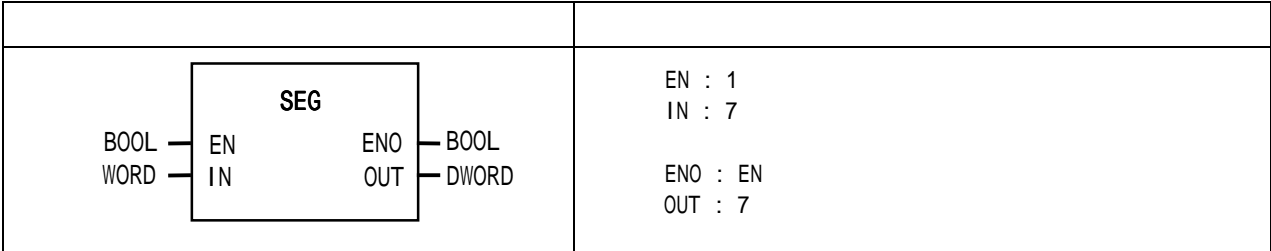
: 1998. 12. 22. 19:37:46,

	Type		Data
_RTC_TOD	TOD		TOD#19:37:46
_RTC_WEEK	UINT		*(0: , 1: , 2: , 3: , 4: , 5: , 6:) 1
_INT_DATE	DATE		(1984 1 1 2083 12 31) D#1998-12-22
_RTC_ERR	BOOL	RTC	RTC '1' 0
_RTC_TIME[n] * n : 0 to 7	BCD		RTC BCD _RTC_TIME[0]: , _RTC_TIME[1]: , _RTC_TIME[2]: , _RTC_TIME[3]: , _RTC_TIME[4]: , _RTC_TIME[5]: , _RTC_TIME[6]: , _RTC_TIME[7]: (0: , 1: , 2: , 3: , 4: , 5: , 6:) _RTC_TIME[0]: 16#98 _RTC_TIME[1]: 16#12 _RTC_TIME[2]: 16#22 _RTC_TIME[3]: 16#19 _RTC_TIME[4]: 16#37 _RTC_TIME[5]: 16#46 _RTC_TIME[6]: 16#1 _RTC_TIME[7]: 16#19

SEG

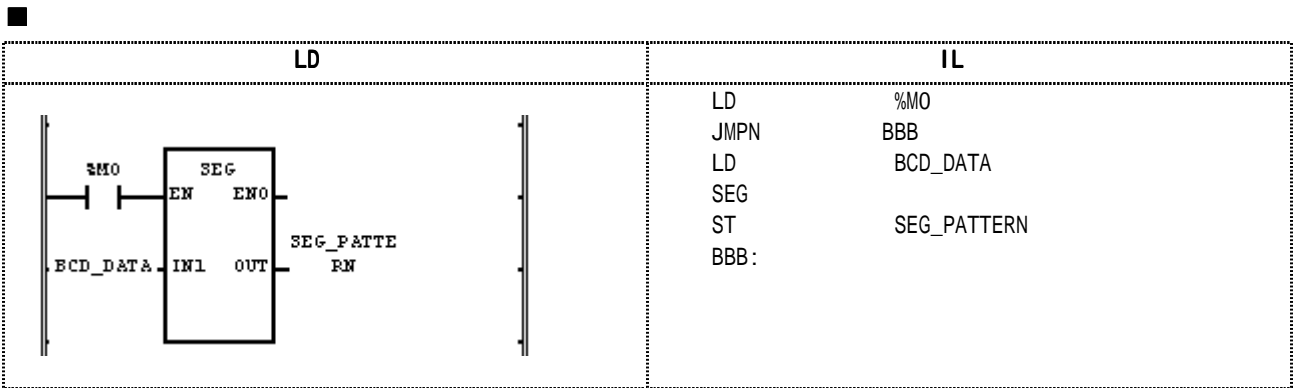
BCD	HEX	7
-----	-----	---

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



EN 1 , IN BCD HEX(16) 7
 . BCD 0000 9999 4 7 가 , HEX 0000
 FFFF 4 7 가 .

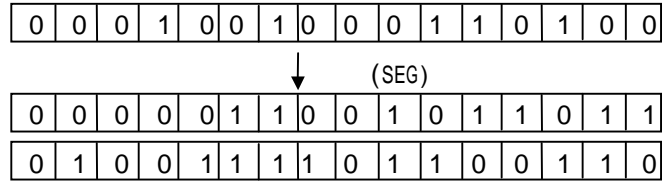
- 1) 4 BCD -> 4 7 : 'SEG'
- 2) 4 HEX -> 4 7 : 'SEG'
- 3) -> 4 BCD 7 : 'INT_TO_BCD' 'SEG'
- 4) -> 4 HEX 7 : 'INT_TO_WORD' 'SEG'
- 5) 7 가 4
- 가) BCD, HEX 4 'SEG'
-) -> 8 BCD 7 :
 10,000 'INT_TO_BCD' 'SEG'
- 4 4 7



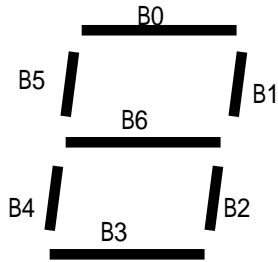
- (1) (%M0) On SEG
 (2) BCD_DATA(WORD) = 16#1234 , 7 '1234'가
 '2#00000110_01011011_01001111_01100110' SEG_PATTERN(DWORD)

(IN1) : BCD_DATA(WORD) = 16#1234

(OUT) : SEG_PATTERN(DWORD) =
 16#065B4F66



7



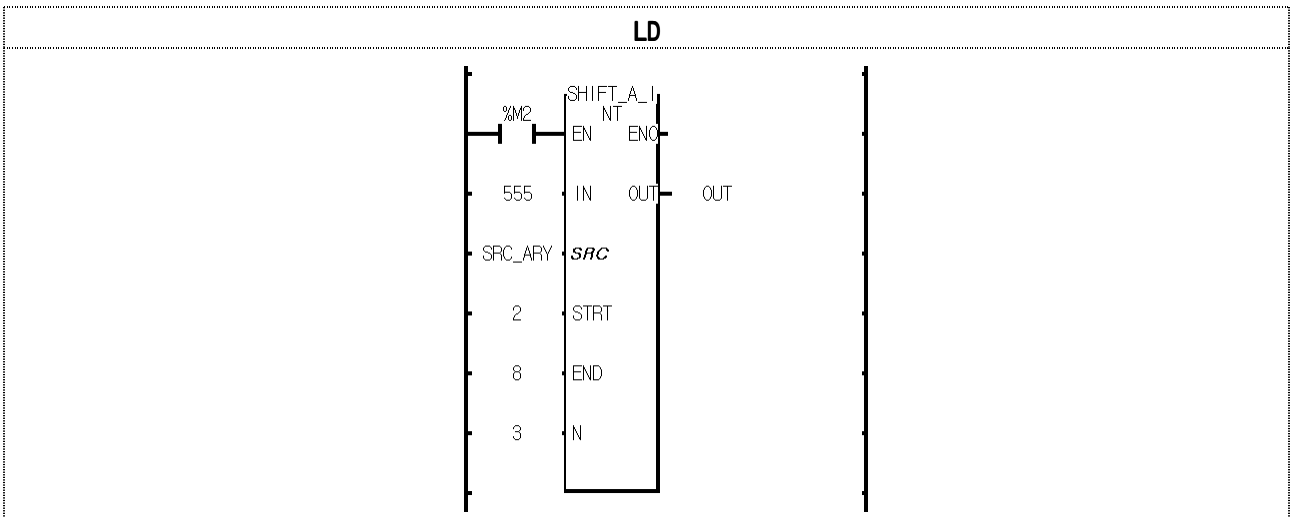
7

(BCD)	(16)		B7	B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	0	1	1	1	1	1	1	0
1	1	1	0	0	0	0	0	1	1	0	1
2	2	2	0	1	0	1	1	0	1	1	2
3	3	3	0	1	0	0	1	1	1	1	3
4	4	4	0	1	1	0	0	1	1	0	4
5	5	5	0	1	1	0	1	1	0	1	5
6	6	6	0	1	1	1	1	1	0	1	6
7	7	7	0	0	1	0	0	1	1	1	7
8	8	8	0	1	1	1	1	1	1	1	8
9	9	9	0	1	1	0	1	1	1	1	9
	A	10	0	1	1	1	0	1	1	1	A
	B	11	0	1	1	1	1	1	0	0	B
	C	12	0	0	1	1	1	0	0	1	C
	D	13	0	1	0	1	1	1	1	0	D
	E	14	0	1	1	1	1	0	0	1	E
	F	15	0	1	1	1	0	0	0	1	F

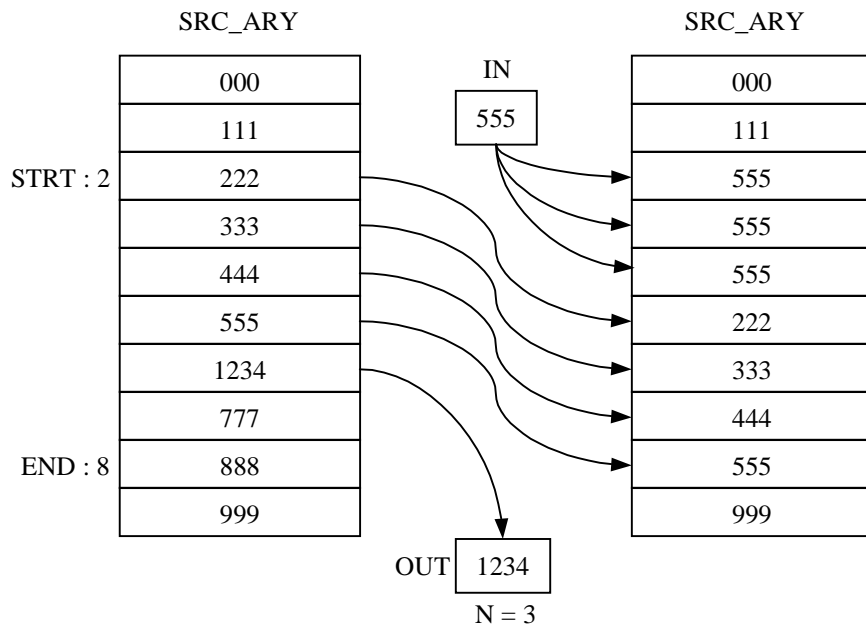
SHIFT_A_BOOL	BOOL
SHIFT_A_BYTE	BYTE
SHIFT_A_WORD	WORD
SHIFT_A_DWORD	DWORD
SHIFT_A_LWORD	LWORD
SHIFT_A_SINT	SINT
SHIFT_A_INT	INT
SHIFT_A_DINT	DINT
SHIFT_A_LINT	LINT
SHIFT_A_USINT	USINT
SHIFT_A_UINT	UINT
SHIFT_A_UDINT	UDINT
SHIFT_A_ULINT	ULINT
SHIFT_A_REAL	REAL
SHIFT_A_LREAL	LREAL
SHIFT_A_TIME	TIME
SHIFT_A_DATE	DATE
SHIFT_A_TOD	TOD
SHIFT_A_DT	DT

STRT END SRC
 SRC

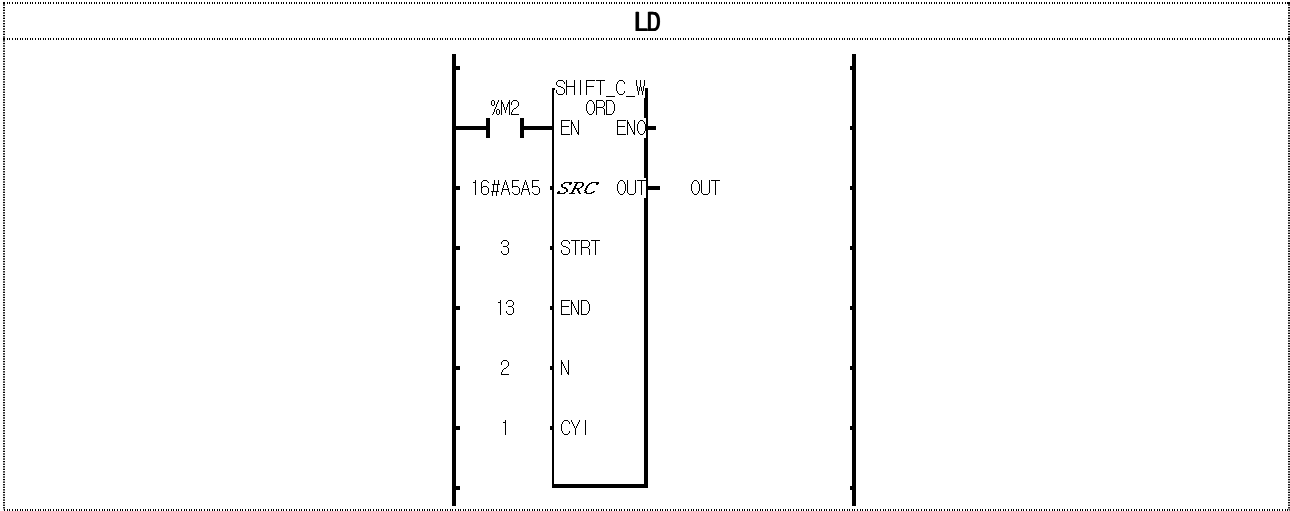
_ERR/_LER 가 (Set)
(i.e. INT=0, TIME=T#0S)



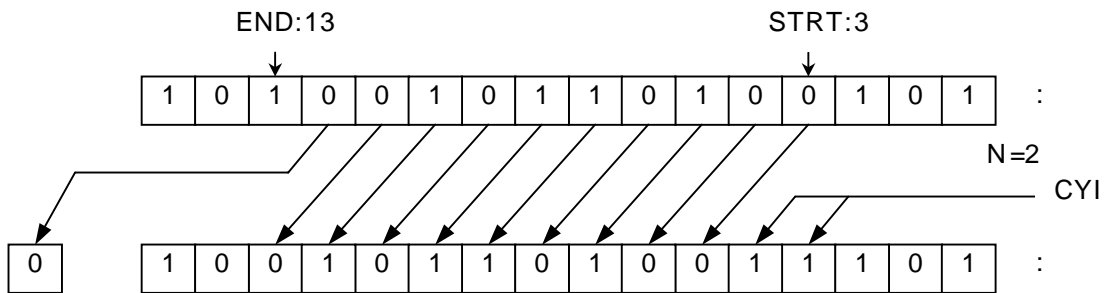
- (1) (%M2) On , SHIFT_A_INT
- (2) SRC_ARY 2 8 SHIFT
- (3) 3 SHIFT
- (4) SHIFT , 2 3 가 555
- (5) (Carry) 1234가



STRT END SRC ERR/_LER 가 (Set) .
 SRC 가



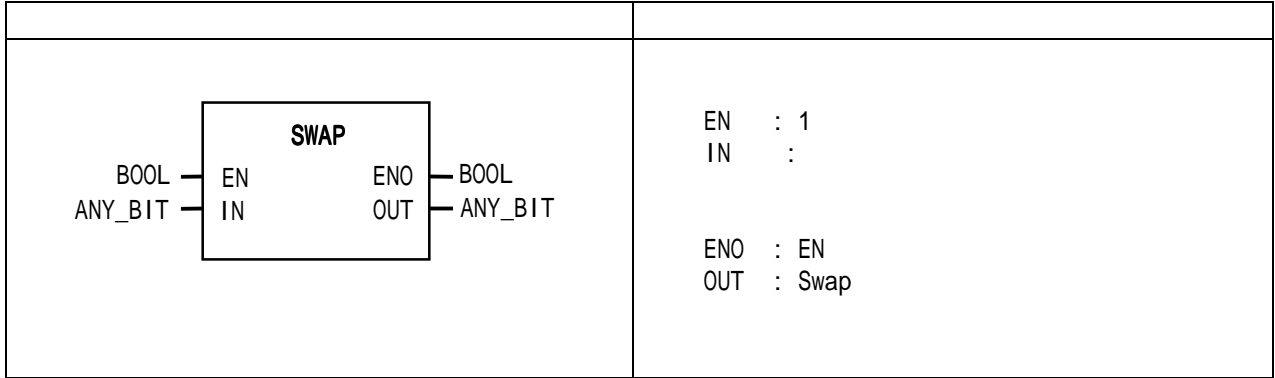
(1) (%M2) On , SHIFT_C_WORD .
 (2) 16#A5A5 STRT END 2 shift shift CYI 1 .
 (3) Shift SRC(16#969D) 2 shift 0 OUT .



SWAP_***

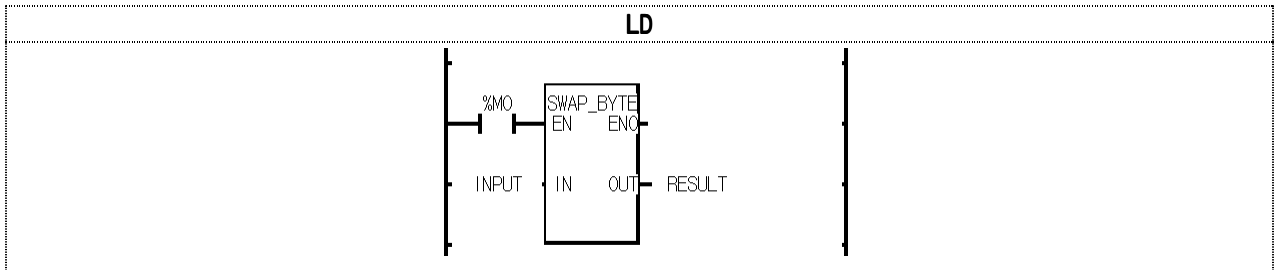


CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

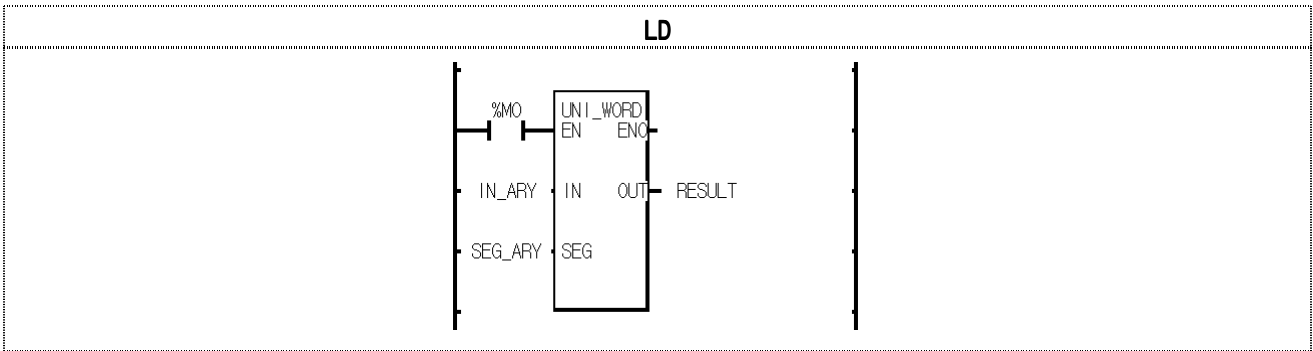


2

SWAP_BYTE	BYTE	BYTE	(Nibble)
SWAP_WORD	WORD	WORD	BYTE
SWAP_DWORD	DWORD	DWORD	WORD
SWAP_LWORD	LWORD	LWORD	DWORD



- (1) (%M0) On , SWAP_BYTE
- (2) INPUT (BYTE)=16#5F , RESULT (BYTE)=16#F5 가



- (1) (%MO) On , UNI_WORD .
- (2) IN_ARY SEG_ARY

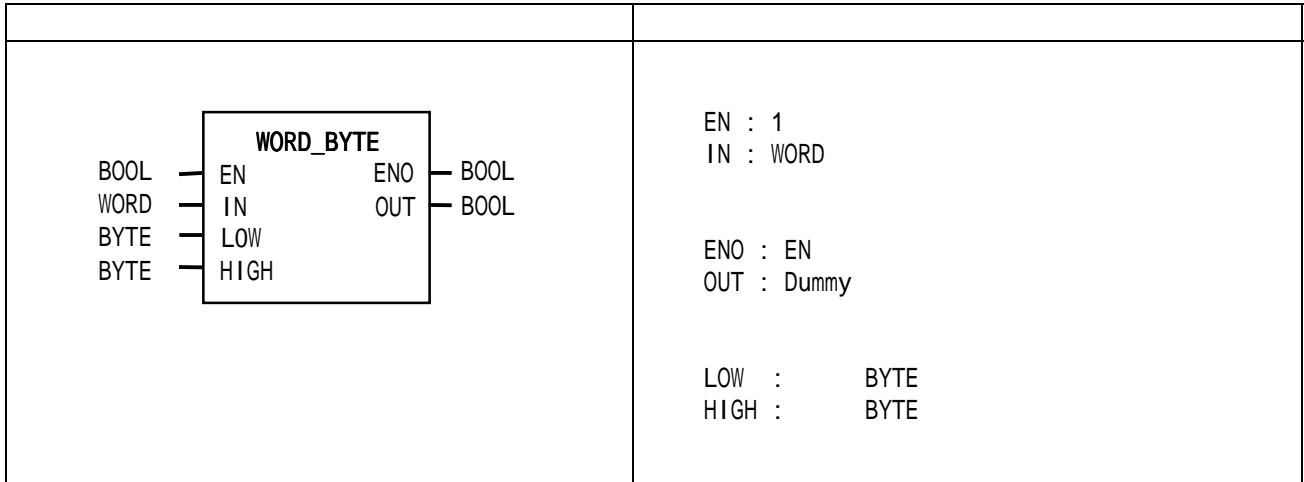
IN_ARY[0]	A 3 B 5	SEG_ARY[0]	3
IN_ARY[1]	B 4 C 6	SEG_ARY[1]	4
IN_ARY[2]	C 5 D 7	SEG_ARY[2]	7
IN_ARY[3]	D 6 E 8	SEG_ARY[3]	2

RESULT 2#00 1010111 0110 101 = 16#2BB5 .

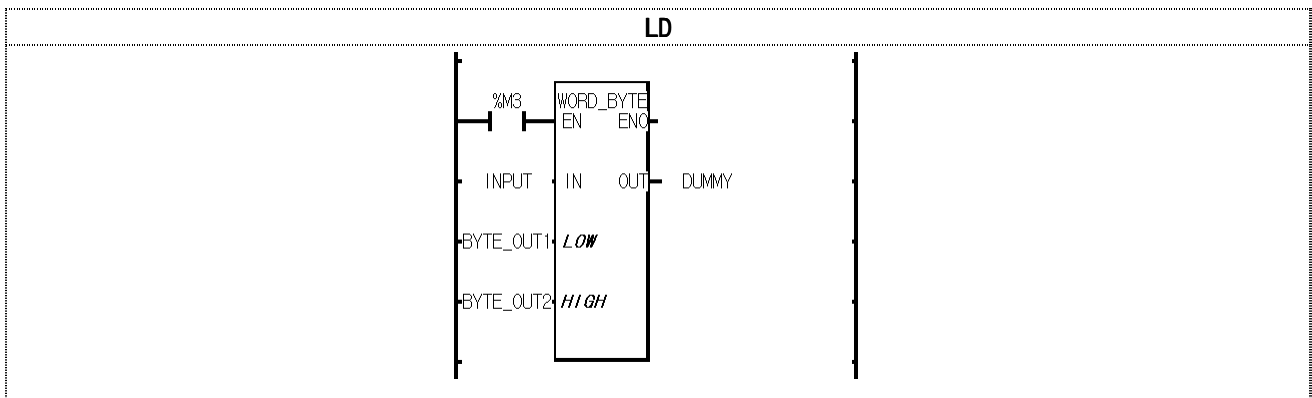
WORD_BYTE

WORD	2	BYTE
------	---	------

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



LOW: 2, HIGH: .

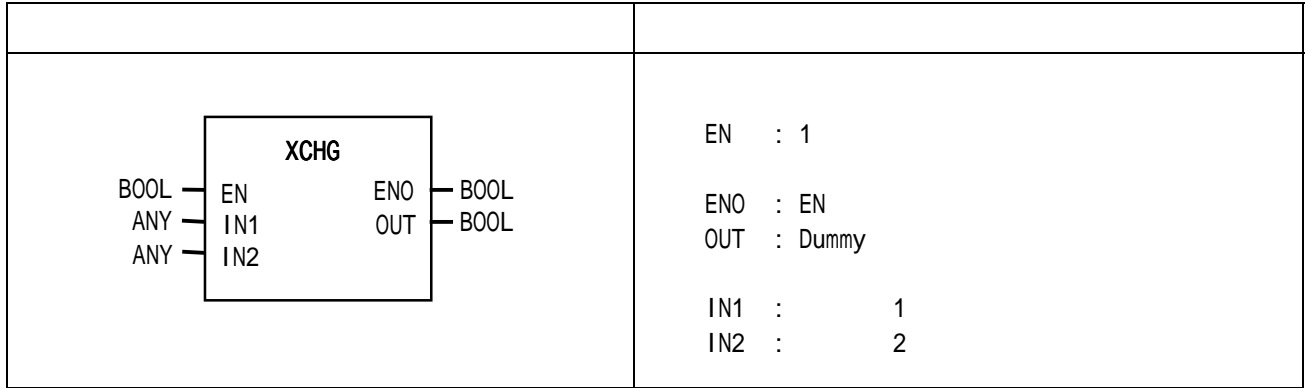


- (1) (%M3) On, WORD_BYTE
- (2) INPUT 16#ABCD, BYTE_OUT1=16#CD, BYTE_OUT2=16#AB

XCHG_***

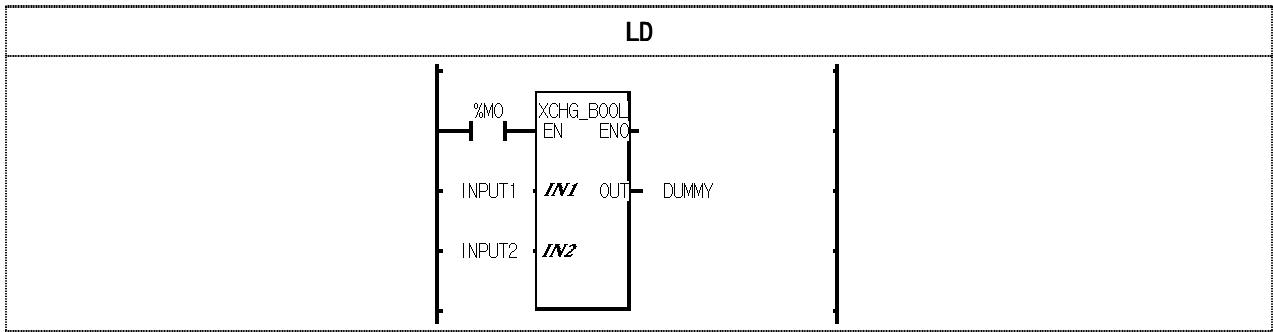
2

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



1 2 .

XCHG_BOOL	BOOL	BOOL	.
XCHG_BYTE	BYTE	BYTE	.
XCHG_WORD	WORD	WORD	.
XCHG_DWORD	DWORD	DWORD	.
XCHG_LWORD	LWORD	LWORD	.
XCHG_SINT	SINT	SINT	.
XCHG_INT	INT	INT	.
XCHG_DINT	DINT	DINT	.
XCHG_LINT	LINT	LINT	.
XCHG_USINT	USINT	USINT	.
XCHG_UINT	UINT	UINT	.
XCHG_UDINT	UDINT	UDINT	.
XCHG_ULINT	ULINT	ULINT	.
XCHG_REAL	REAL	REAL	.
XCHG_LREAL	LREAL	LREAL	.
XCHG_TIME	TIME	TIME	.
XCHG_DATE	DATE	DATE	.
XCHG_TOD	TOD	TOD	.
XCHG_DT	DT	DT	.
XCHG_STRING	STRING	STRING	.



- (1) (%M0) On , XCHG_BOOL .
- (2) INPUT1=0 INPUT2=1 2
INPUT1=1 , INPUT2=0 .

8.3

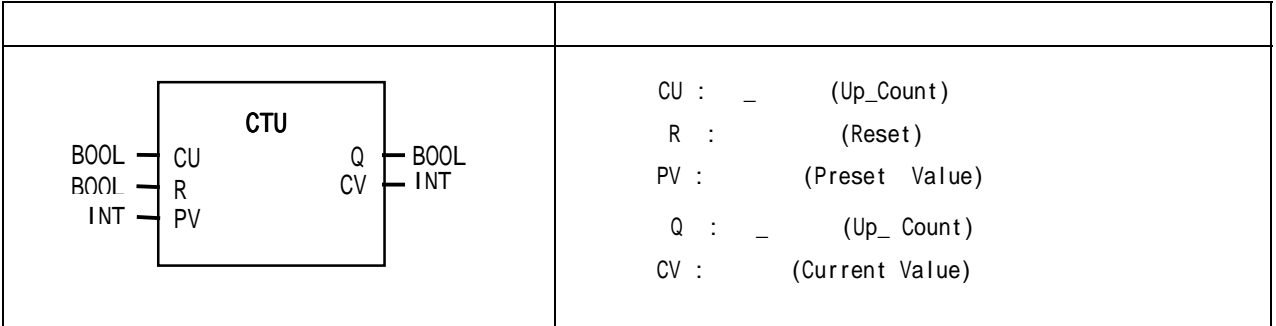
1. .
2. 3.5.2 .

(1)CTD . (COUNT_D)
 (2)CD %I0.1.14 .
 (3)PV CV _10N (On) .
 (4)PV INT (-32768 32767) 5 .
 (5)CV (COUNT_CV) .
 (6)Q (COUNT_Q) .
 (7) , PLC .
 (8) (Stop →Run) .
 (9) Run PV 5가 CV (Count_CV) .
 (10) 가 %I.0.1.14 CV (COUNT_CV) 1 .
 (11) 5 가 CV 0가 , Q (COUNT_Q) 1 .
 (12)Q (COUNT_Q)가 1 %Q0.3.0 Set .

CTU

가 ()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



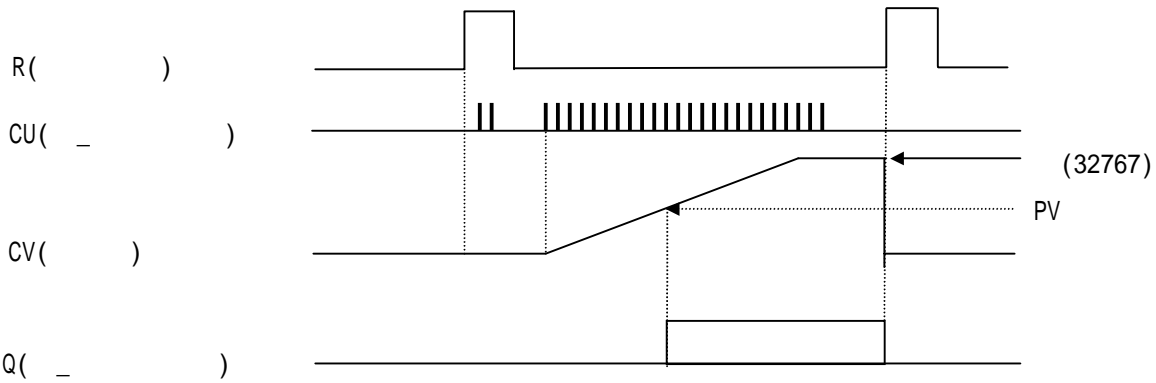
가 CTU CU가 0 1 CV가 1 가

, CV가 1INT 32767 가 , 32767 가

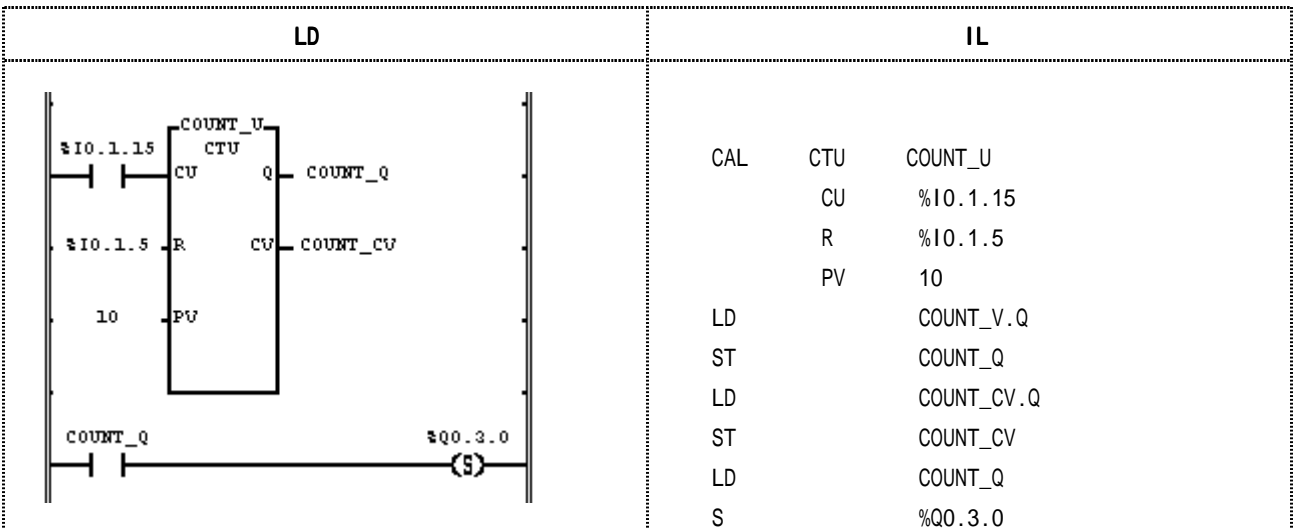
R 1 CV 0 (Clear)

Q CV가 PV 1

PV CTU 가



%I0.1.15 10 가 , %Q0.3.1 Set

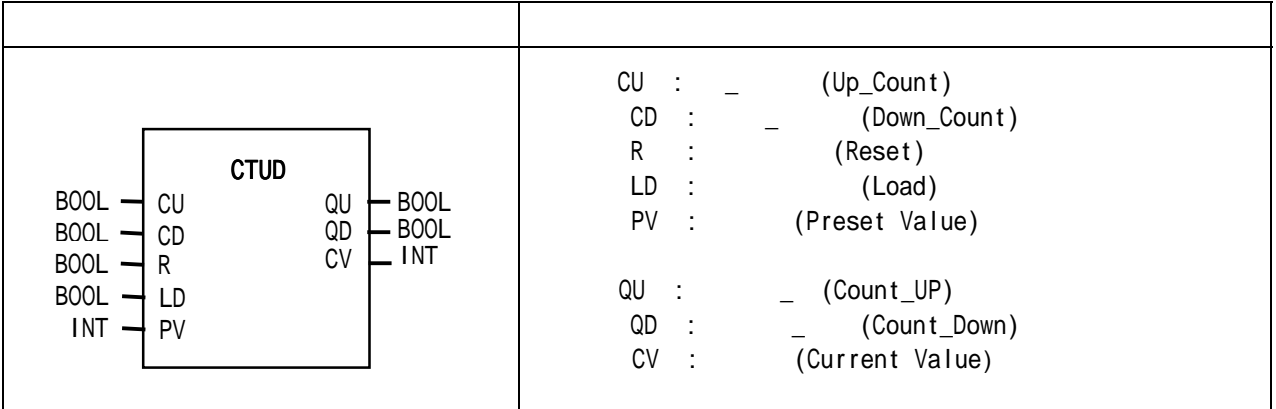


- (1)CTU . (COUNT_D)
- (2)CU %10.1.15 .
- (3)PV 10 .
- (4)CV R (COUNT_CV) . (%10.1.5)
- (5)CV (COUNT_CV) .
- (6)Q (COUNT_Q) .
- (7) , PLC .
- (8) (Stop →Run) .
- (9) 가 %10.1.15 CV (COUNT_CV) 1 가 .
- (10) 10 가 CV 10 , Q(COUNT_Q)가 1
- (11)Q(COUNT_Q)가 1 %Q0.3.0 Set .

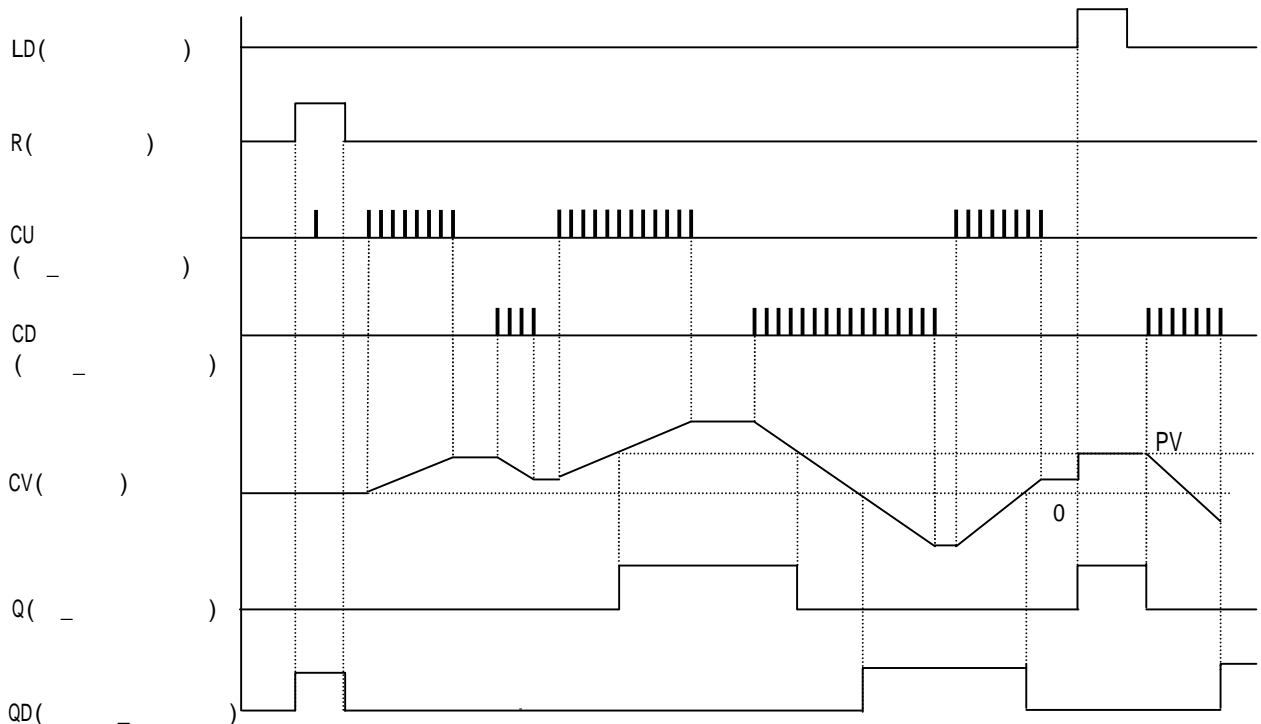
CTUD

가 ()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



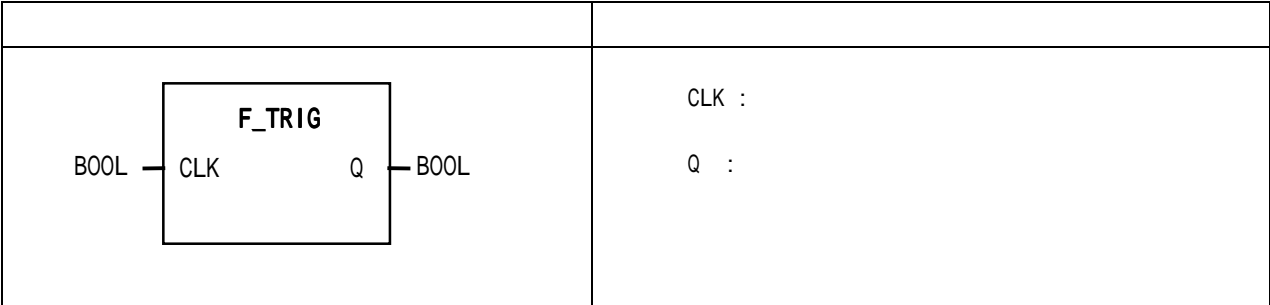
가 CTUD CU가 0 1 CV가 1 가
 , CD가 0 1 CV가 1
 , CV가 1NT -32768 32767 가 ,
 가,
 LD가 1 CV PV . (CV=PV)
 R 1 CV 0 (Clear) . (CV=0)
 QV CV가 PV 1 , QD CV가 0 1 .
 R > LD > CU > CD , 가



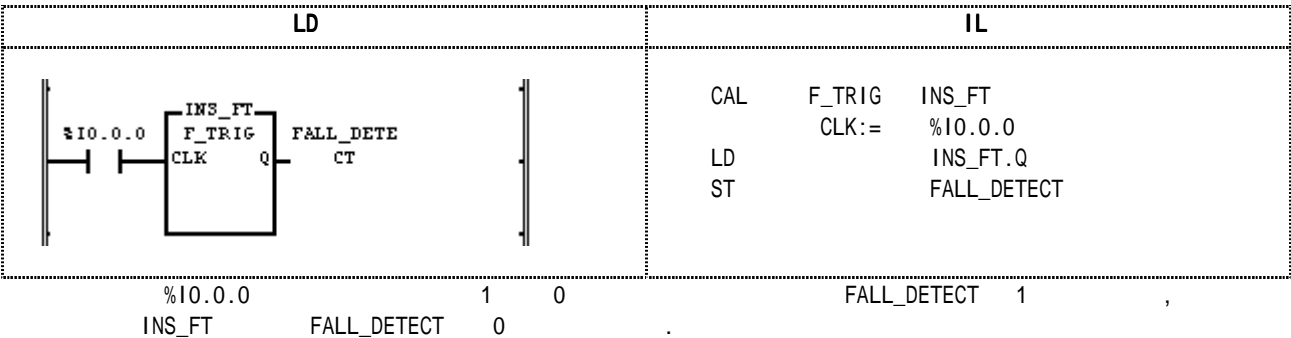
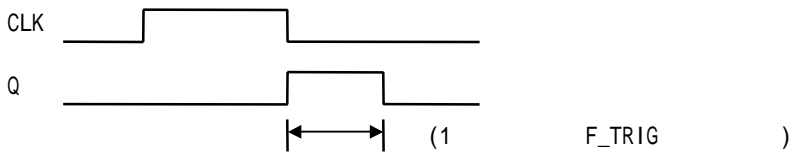
F_TRIG

()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



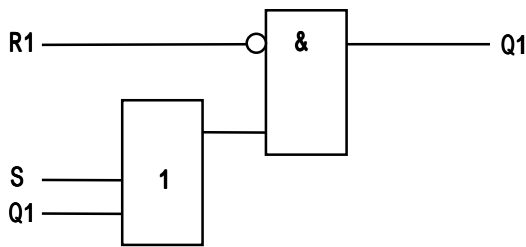
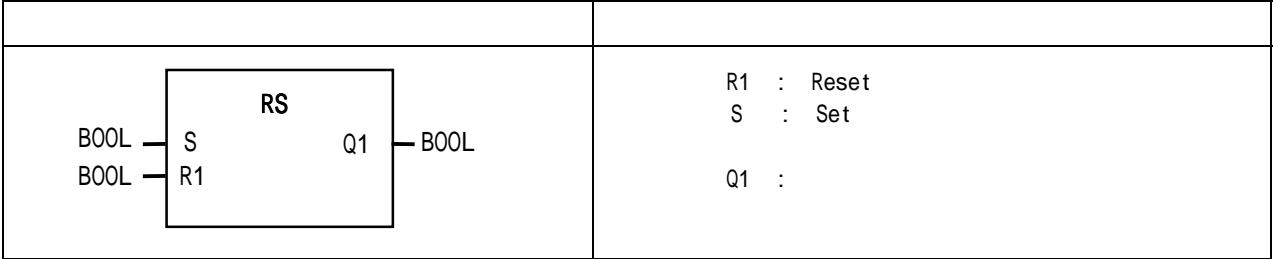
F_TRIG CLK , 가 1 0 Q 1 0



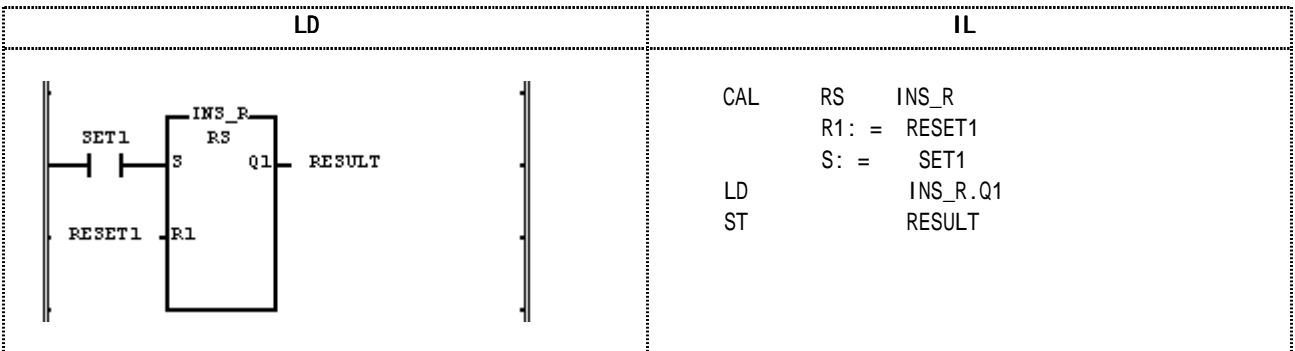
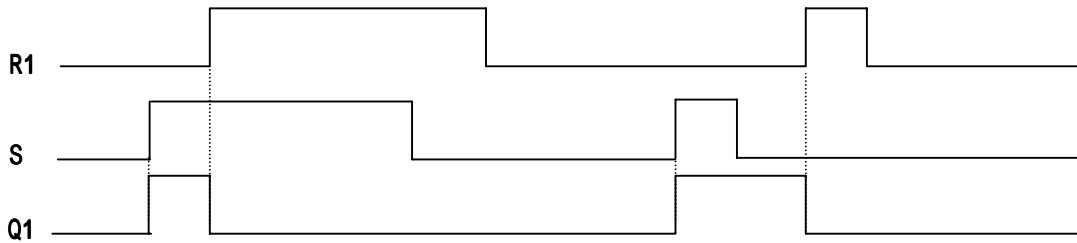
RS

Reset Bistable()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



R1 1 , S Q1 0 . Q1 , R1 0 S가 1 1
. Q1 0 .

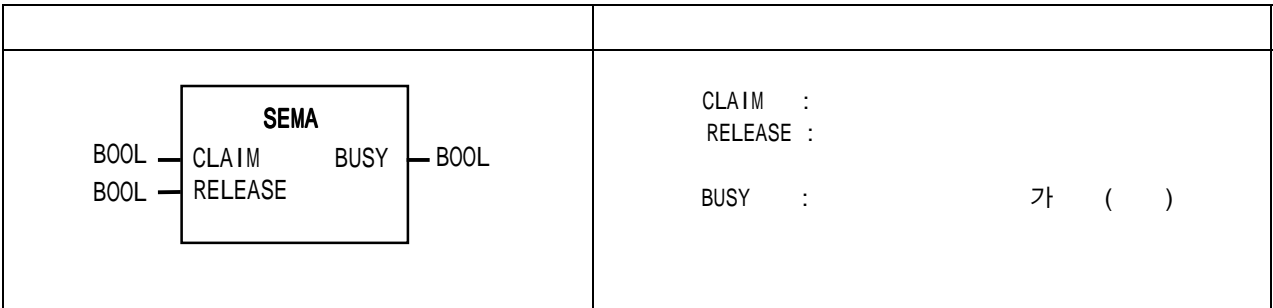


RESET1 Reset , SET1 Set RESULT
R1 RESET1, S SET1 , Q1 RESULT
(1) SET1 RESET1 On RESULT 1
(2) RESET1 On RESULT 0
(3) SET1 RESET1 On RESULT 0가

SEMA

()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



```

SEMA                    (CLAIM = 1    0, RELEASE = 0    )
BUSY 가 1                    CLAIM = 1, RELEASE = 0
SEMA                    BUSY가 0                    . BUSY가 0
                         CLAIM = 0, RELEASE = 1                    SEMA
    
```

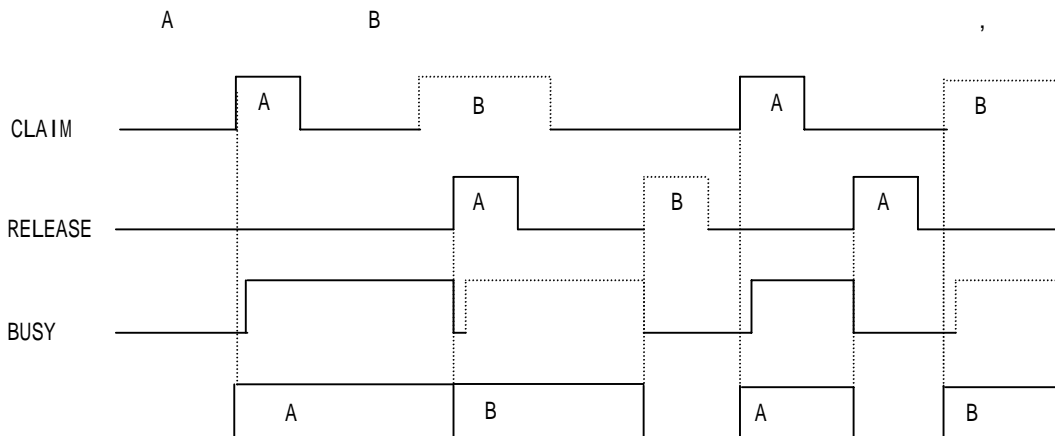
```

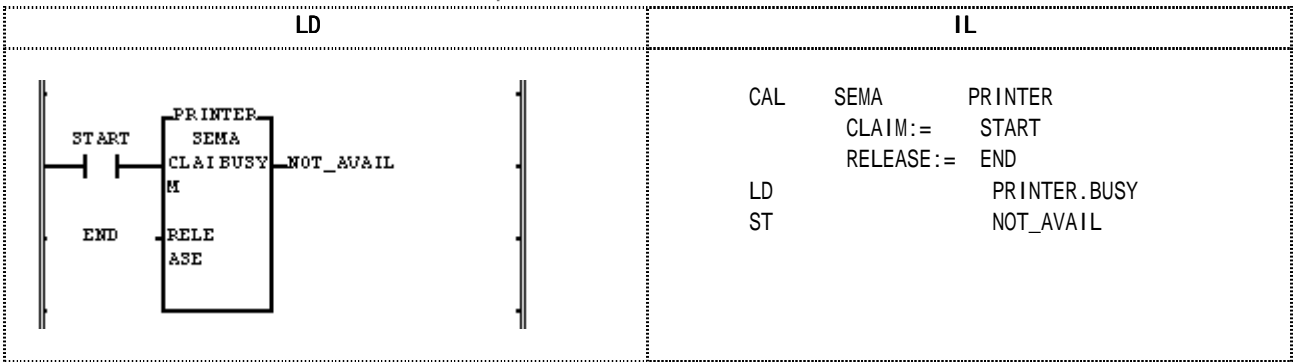
( CLAIM = 0, RELEASE = 1                    SEMA
가                    .)
    
```

- SEMA
-
- GM1 CPU
- SEMA

```

VAR X : BOOL := 0 ; END_VAR
  BUSY := X ;
  IF CLAIM THEN X := 1 ;
  ELSIF RELEASE THEN BUSY := 0; X := 0 ;
  END_IF
    
```





PLC

‘PRINTER’

‘PRINTER’

SEMA

START 1, END 0

‘PRINTER’ SEMA

BUSY 가 1

NOT_Avail 1

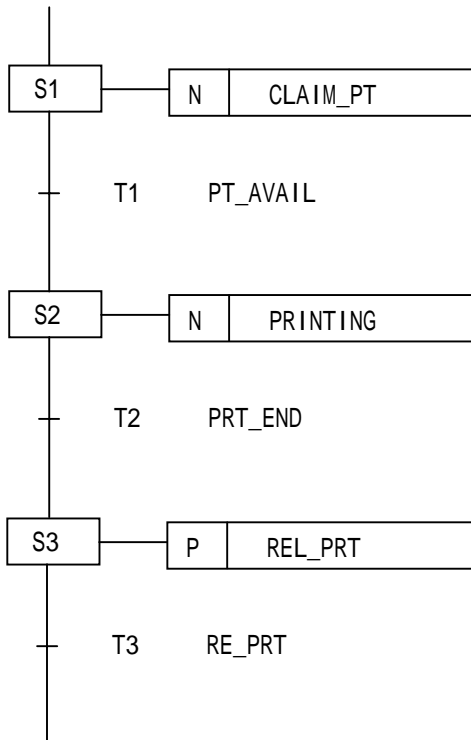
BUSY 0 가

START 0, END 1

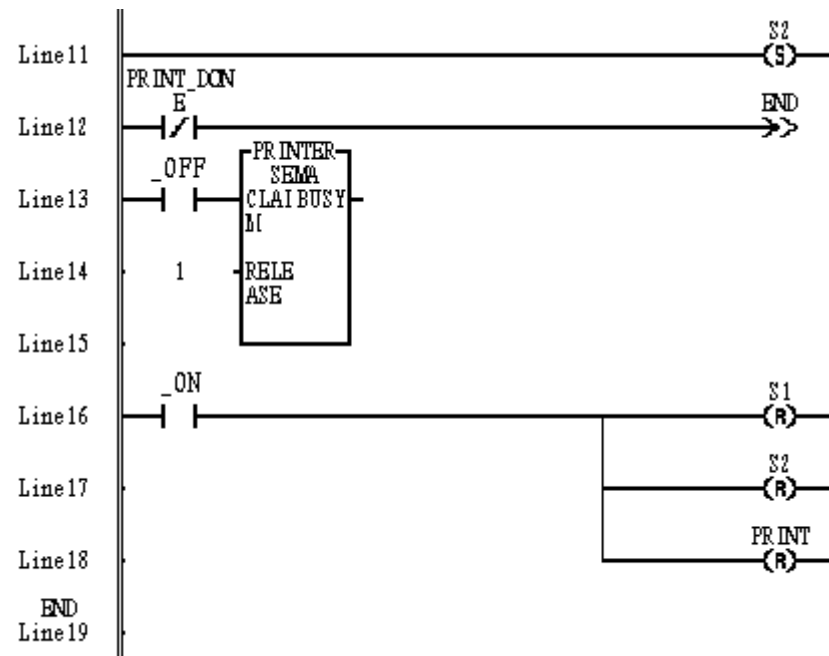
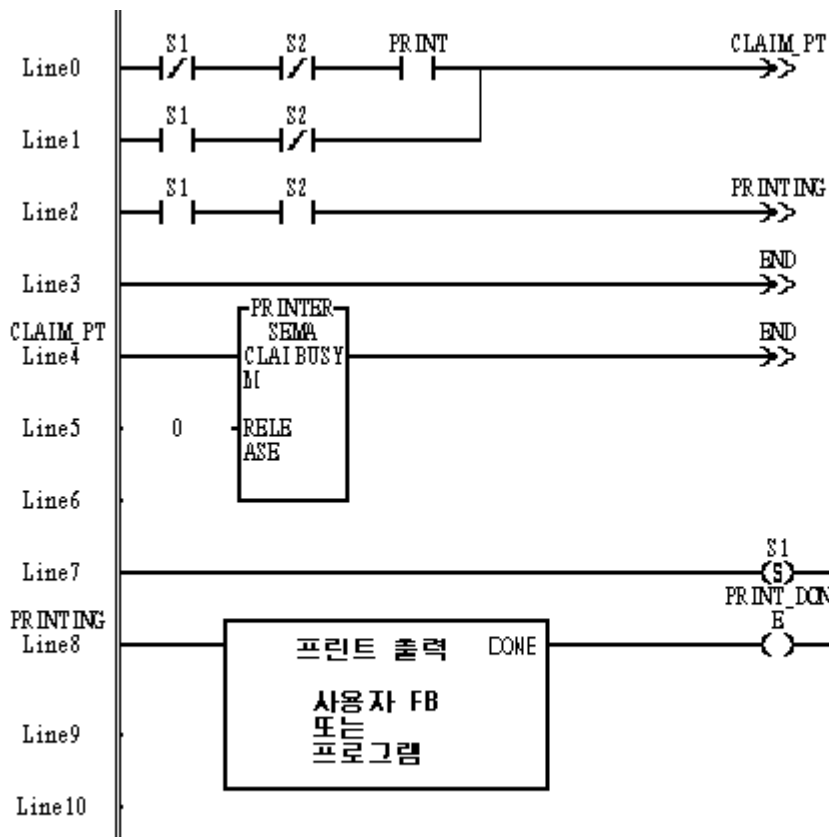
‘PRINTER’ SEMA

가

VAR_EXTERNAL
PRINTER:SEMA
END_VAR



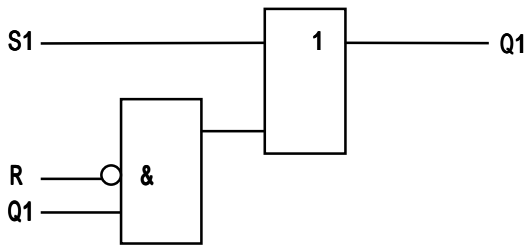
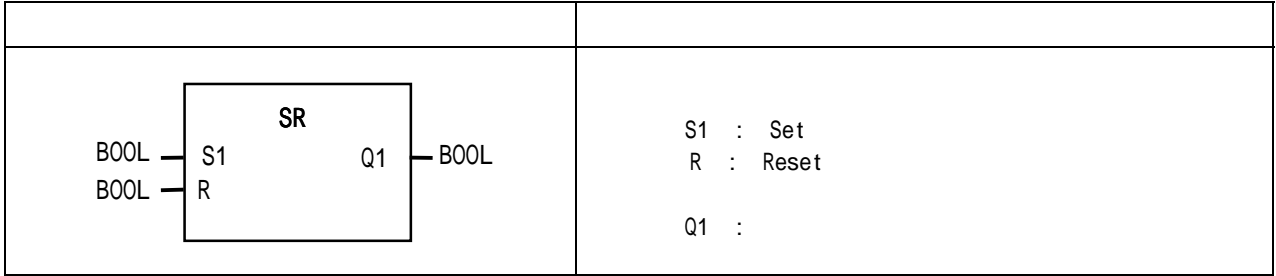
S1	CLAIM_PT;
	<pre> CAL SEMA PRINTER CLAIM:= 1 RELEASE:= 0 </pre>
T1	PT_Avail;
	<pre> LDN PRINTER.BUSY ST TRANS </pre>
S2	PRINTING;
	PRINT_DONE:=1
T2	PRT_END;
	<pre> LD PRINTER_DONE ST TRANS </pre>
S3	REL_PRT;
	<pre> CAL SEMA PRINTER CLAIM:= 0 RELEASE:= 1 </pre>
T3	RE_PRT;
	<pre> LD PRT_REQ ST TRANS </pre>



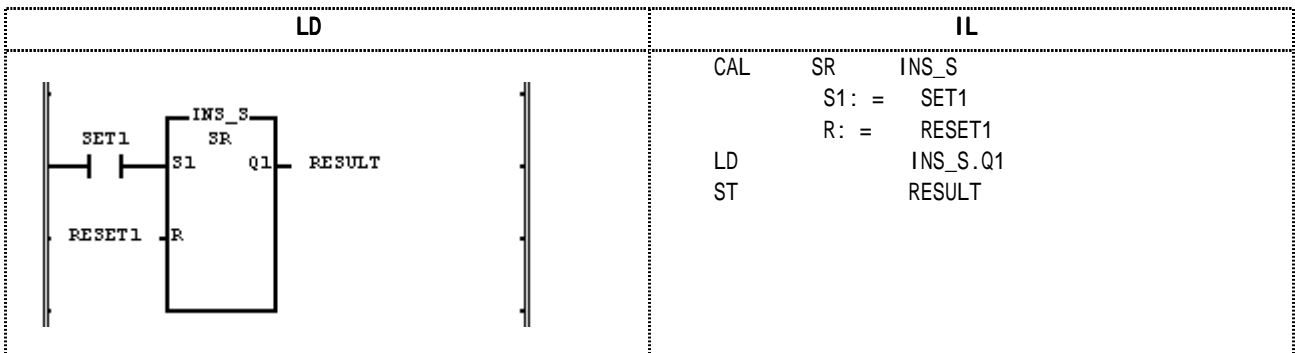
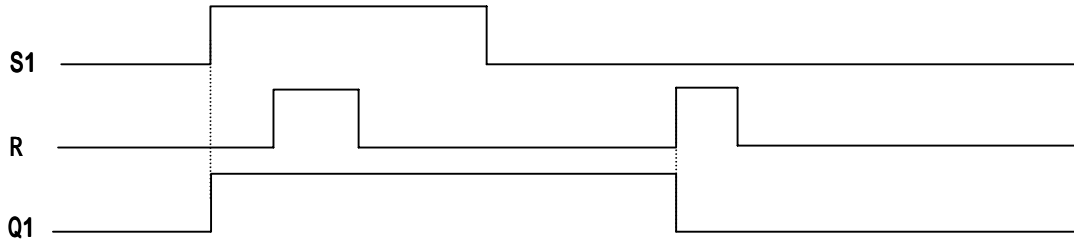
SR

Set Bistable ()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



S1 1 , R Q1 1 . Q1 , S1 0 R 1
0 . Q1 0 .



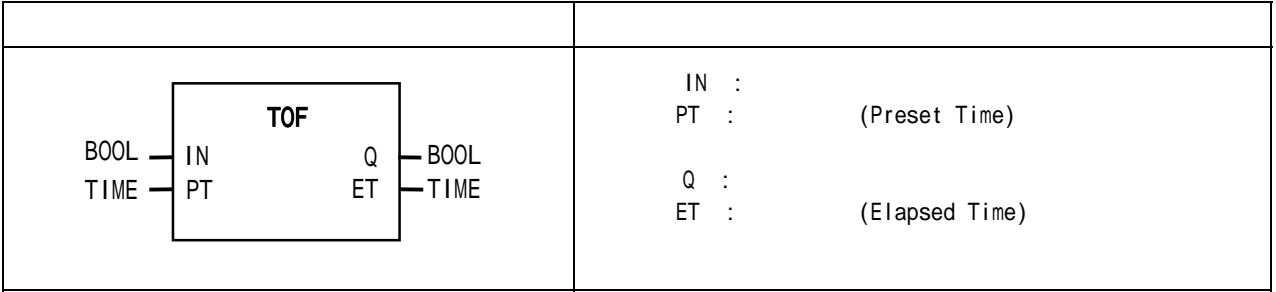
(1) SET1 On , RESULT 1

(2) On , RESULT 0 , SET1 Off , RESELT1

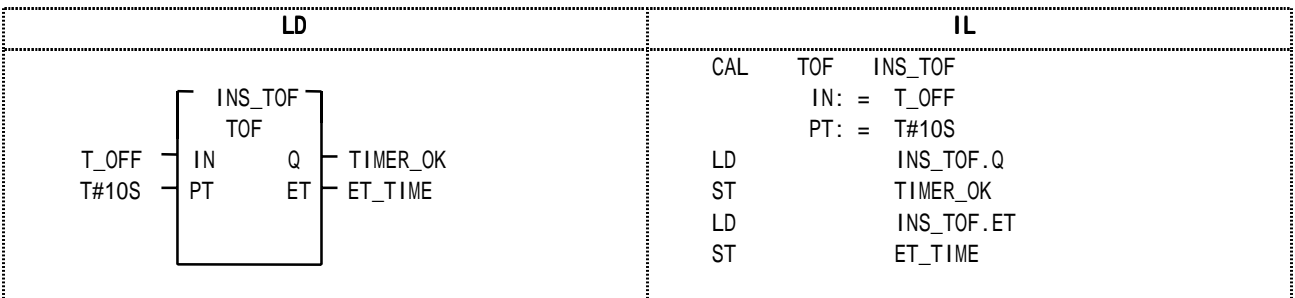
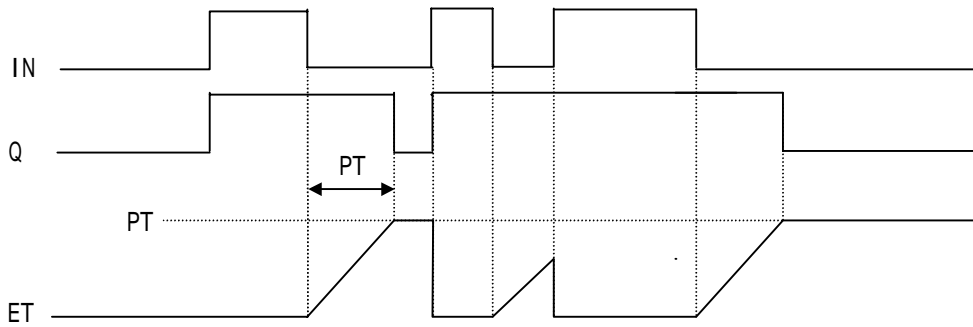
TOF

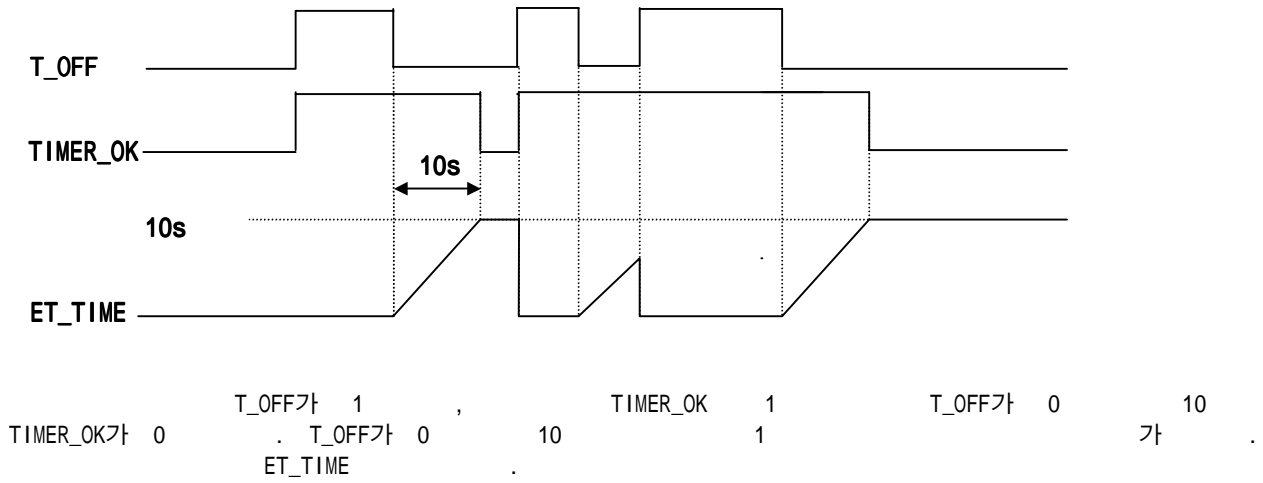
OFF ()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN 1 , Q가 1 , IN 0 PT Q가
 0 .
 IN 0 ET가 ET . IN 1 , 0 .

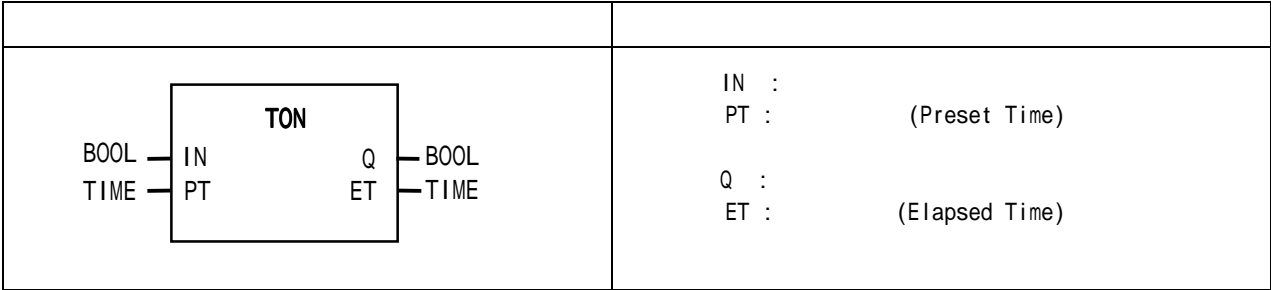




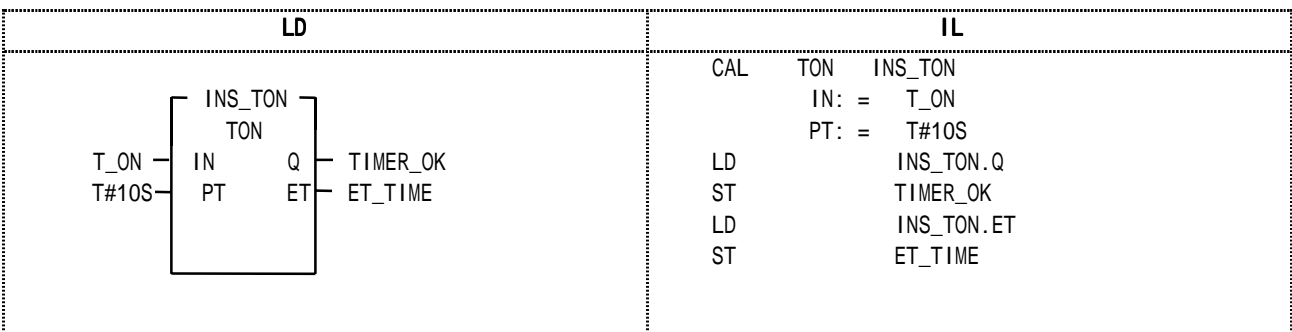
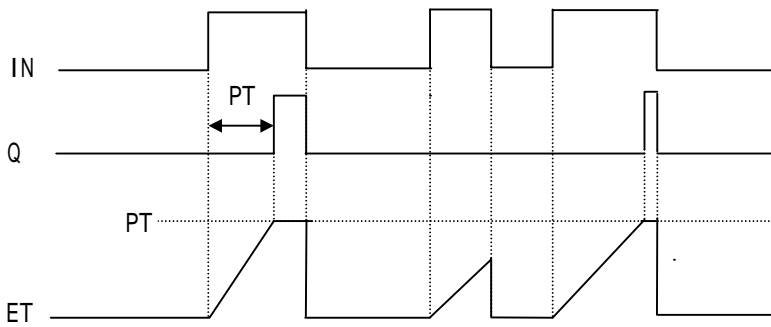
TON

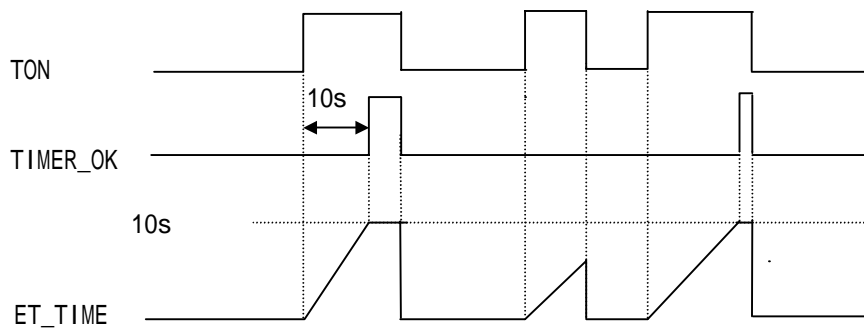
ON ()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN 1 ET가 ET . IN 0 , 0 .
Q가 1 IN 0 , Q 0 .



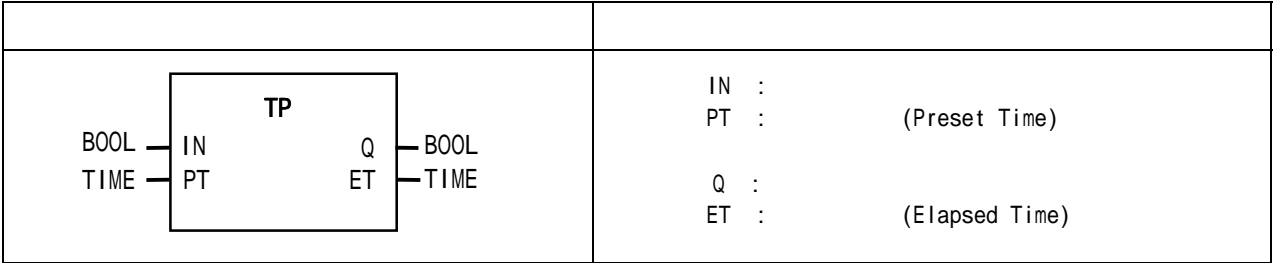


- (1) T_ON 1 10 가 TIMER_OK가 1 .
- (2) T_ON 1 ET_TIME
- (3) ET_TIME 10 T_ON 0 , ET_TIME
0
- (4) TIMER_OK가 1 T_ON 0 , TIMER_OK 0 ET_TIME 0 .

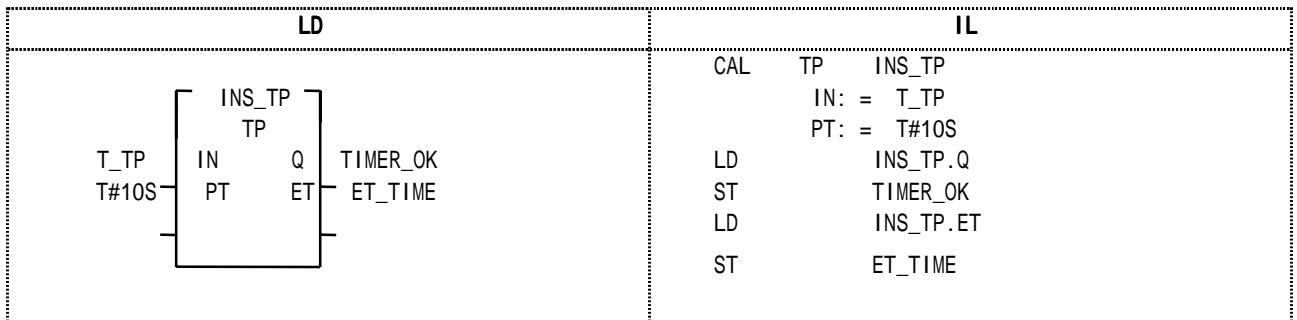
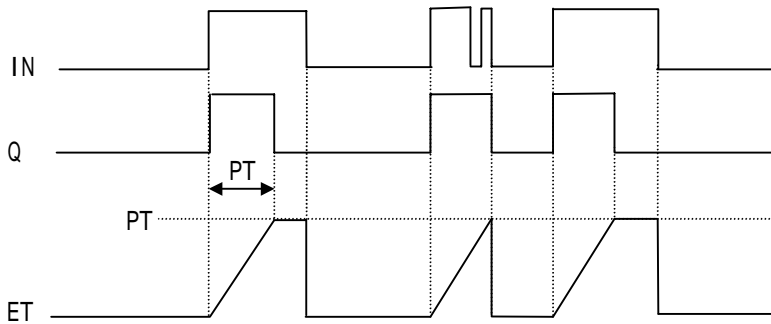
TP

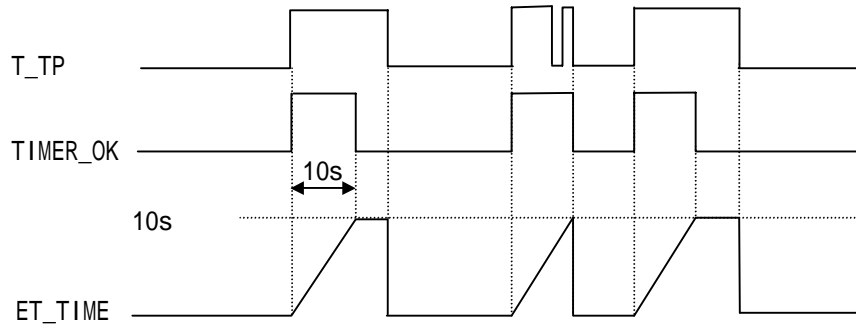
()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



IN 1 PT Q가 1 , ET가 PT
0 .
ET IN 1 가 PT 가 IN 0 0
. ET가 가 IN 0 1 .





- (1) T_TP가 0 1 10 TIMER_OK 1 . 가 가 10
 T_TP
 (2) ET_TIME 가 T#10S . T_TP가 0 0 .

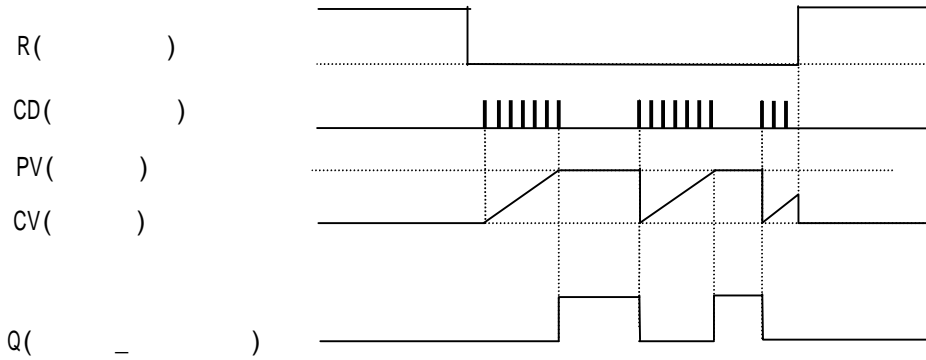
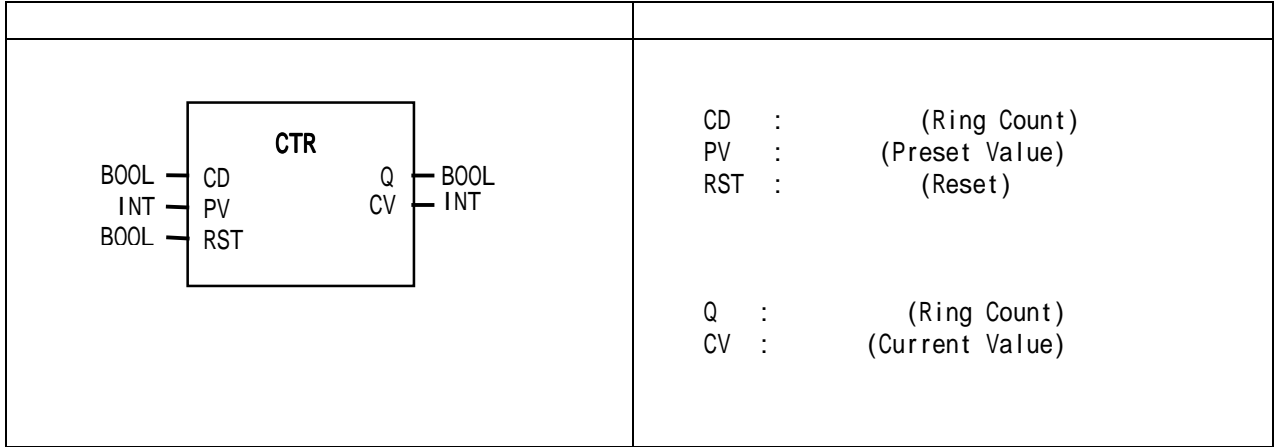
8.4

1. (MASTER-K) .
2. 3.5.2

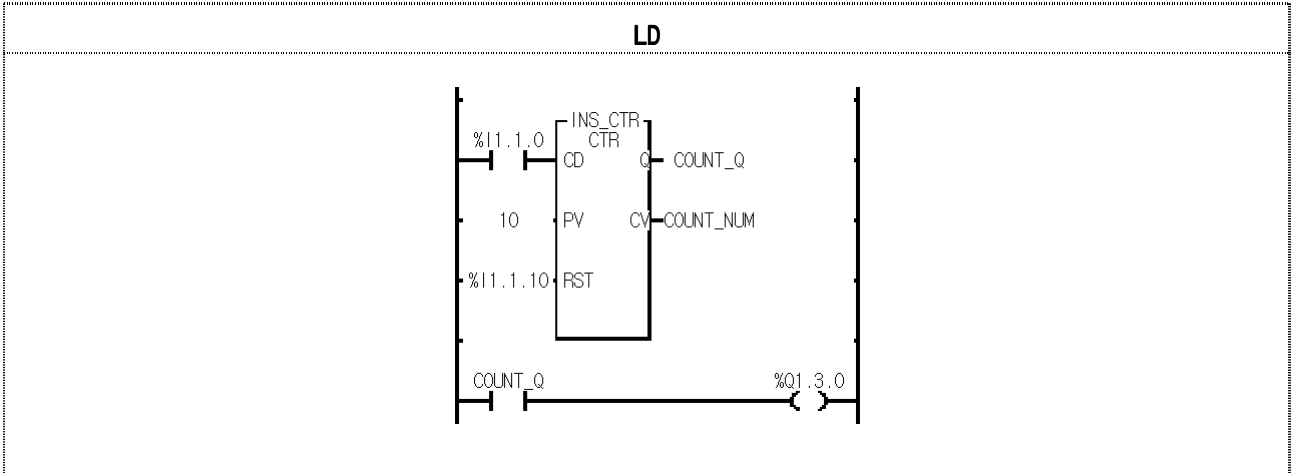
CTR

Ring

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



%I1.1.0 10 가 , %Q1.3.1 ON

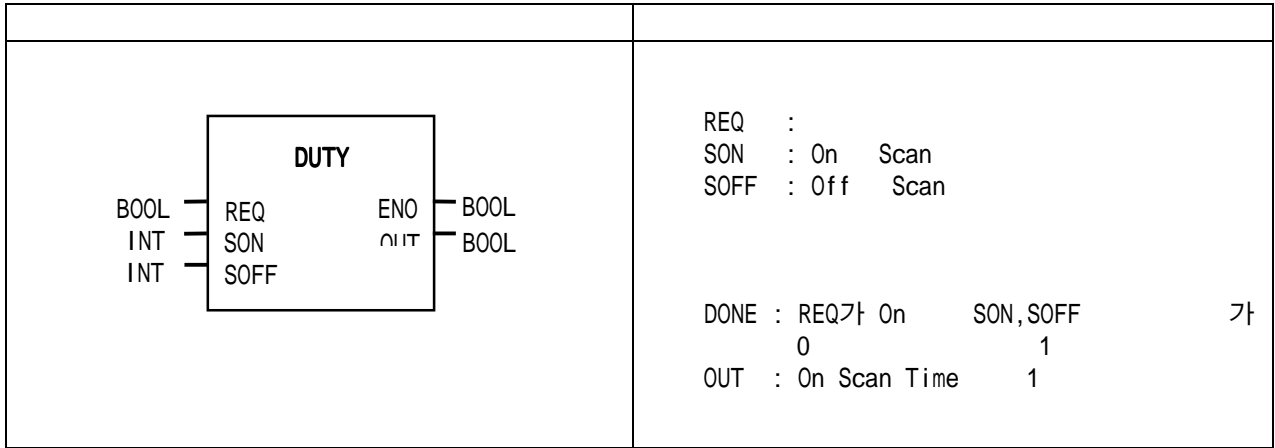


- (1) CTR (INS_CTR)
- (2) CD %I1.1.0
- (3) PV 10
- (4) CV RST .(%I1.1.10)
- (5) CV (COUNT_NUM)
- (6) Q (COUNT_Q)
- (7) , PLC
- (8) (Stop → Run)
- (9) 가 %I1.1.0 CV(COUNT_NUM) 1 가
- (10) 10 가 CV 10 , Q(COUNT_Q)가
- 1
- (11) Q(COUNT_Q)가 1 %Q1.3.0 ON
- (12) 가 %I1.1.0 Q(COUNT_Q) 0
- %Q1.3.0 OFF

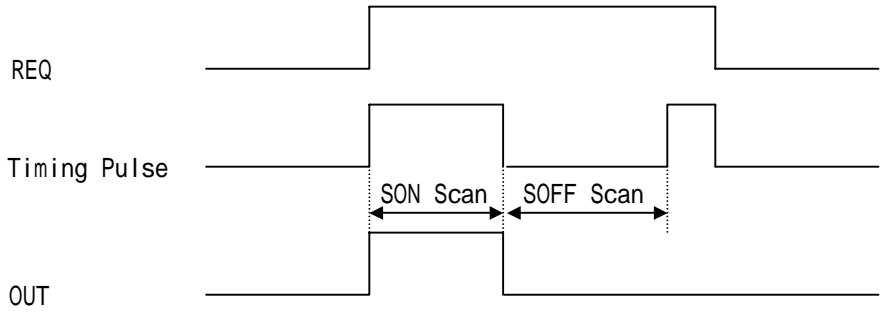
DUTY

On/Off

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



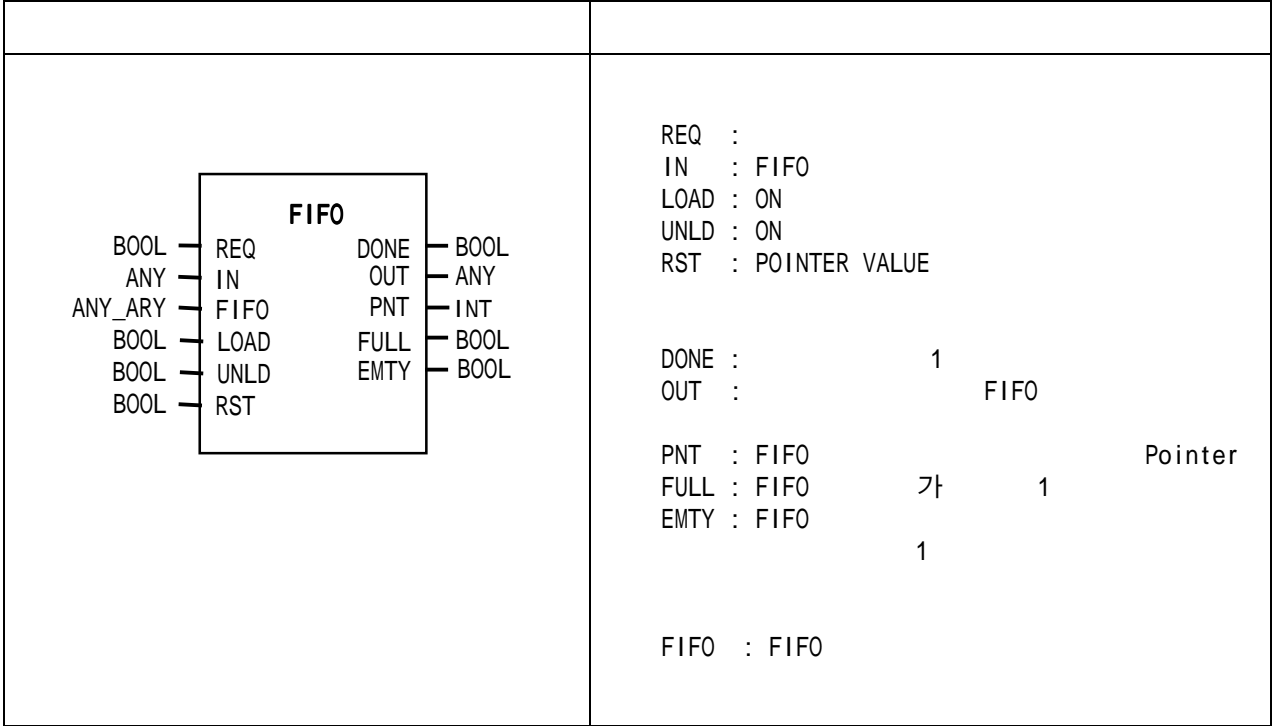
DUTY FB REQ가 ON SON Scan On, SOFF Scan Off
 SON = 0 OFF가
 SON > 0, SOFF = 0 ON
 REQ가 OFF OFF
 SON < 0 SOFF < 0 , DONE OFF OUT = 0



FIFO_***

FIFO	Load/Unload
()	

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



FIFO IN FIFO Load FIFO Unload . . .

FIFO Unload 가 , Shift , PNT

, PNT가 0 (clear) . , PNT

RST가 PNT 0 , EMTY 가 ON , FIFO 0

(clear) .

Off FIFO 가 .

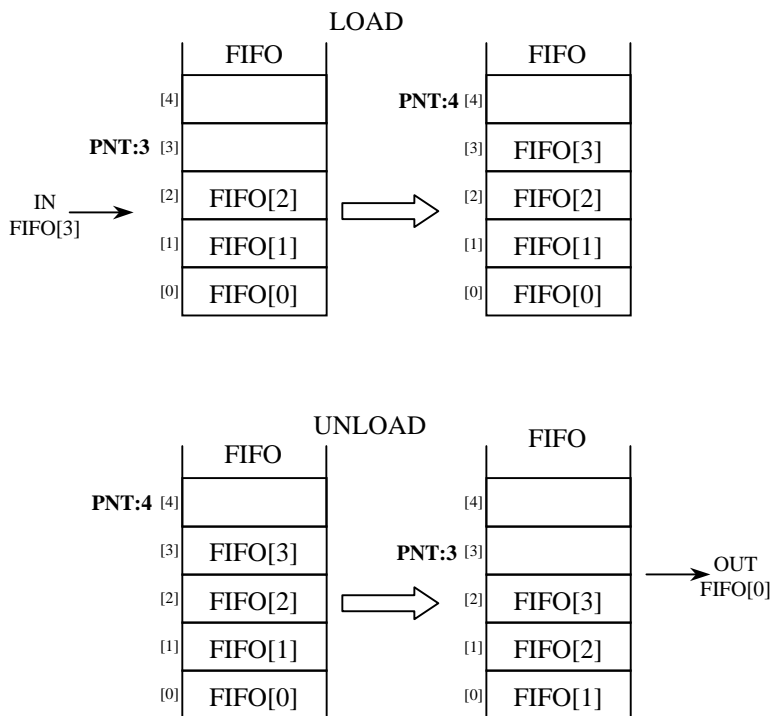
RETAIN FIFO FIFO

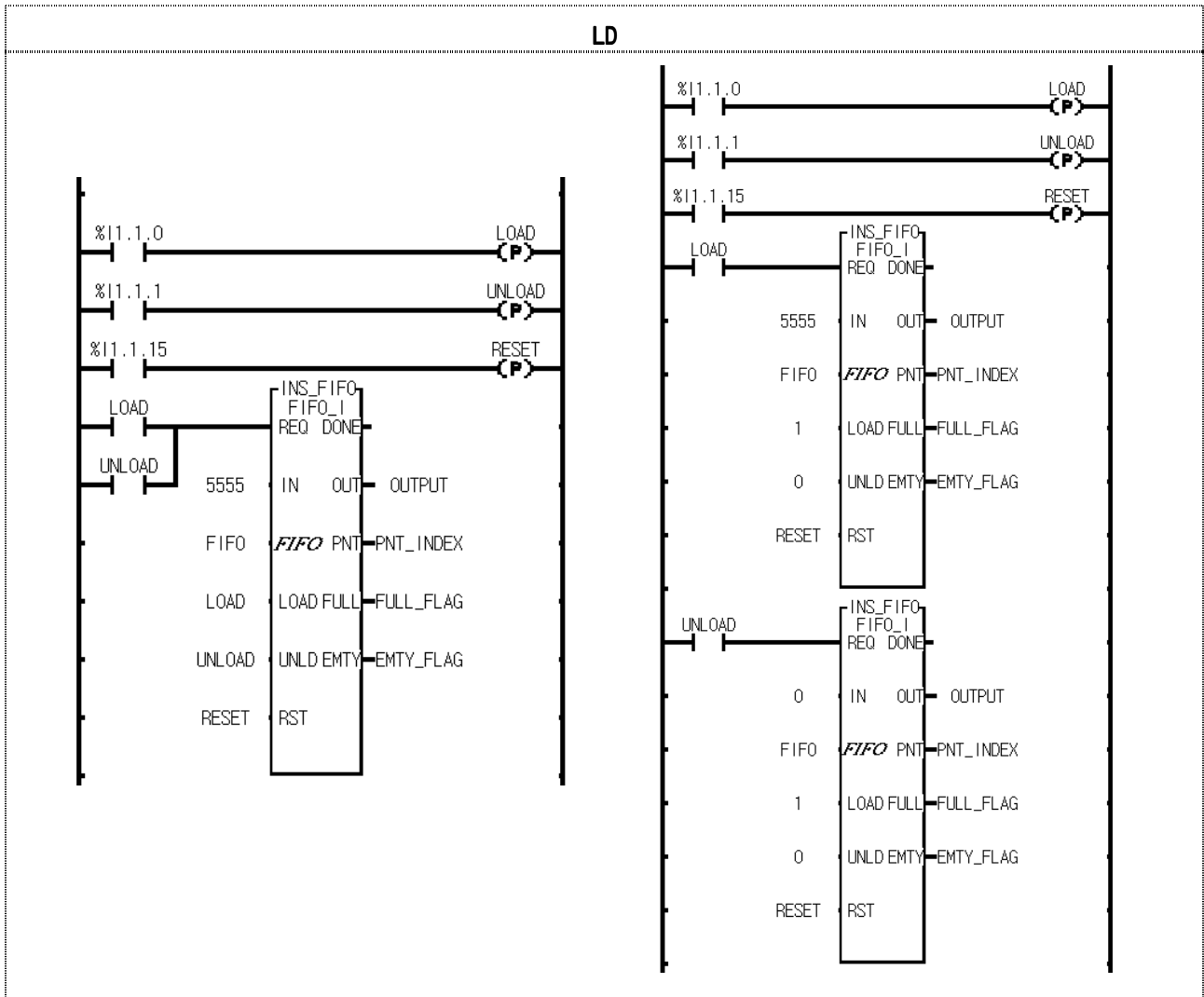
REQ 가 가 .

PNT Load IN . Load

OUT 0 .

	FIFO				
FIFO_Q	BOOL	BOOL	DATA	FIFO	.
FIFO_B	BYTE	BYTE	DATA	FIFO	.
FIFO_W	WORD	WORD	DATA	FIFO	.
FIFO_DW	DWORD	DWORD	DATA	FIFO	.
FIFO_LW	LWORD	LWORD	DATA	FIFO	.
FIFO_SI	SINT	SINT	DATA	FIFO	.
FIFO_I	INT	INT	DATA	FIFO	.
FIFO_DI	DINT	DINT	DATA	FIFO	.
FIFO_LI	LINT	LINT	DATA	FIFO	.
FIFO_USI	USINT	USINT	DATA	FIFO	.
FIFO_UI	UINT	UINT	DATA	FIFO	.
FIFO_UDI	UDINT	UDINT	DATA	FIFO	.
FIFO_ULI	ULINT	ULINT	DATA	FIFO	.
FIFO_R	REAL	REAL	DATA	FIFO	.
FIFO_LR	LREAL	LREAL	DATA	FIFO	.
FIFO_TM	TIME	TIME	DATA	FIFO	.
FIFO_DAT	DATE	DATE	DATA	FIFO	.
FIFO_TOD	TOD	TOD	DATA	FIFO	.
FIFO_DT	DT	DT	DATA	FIFO	.





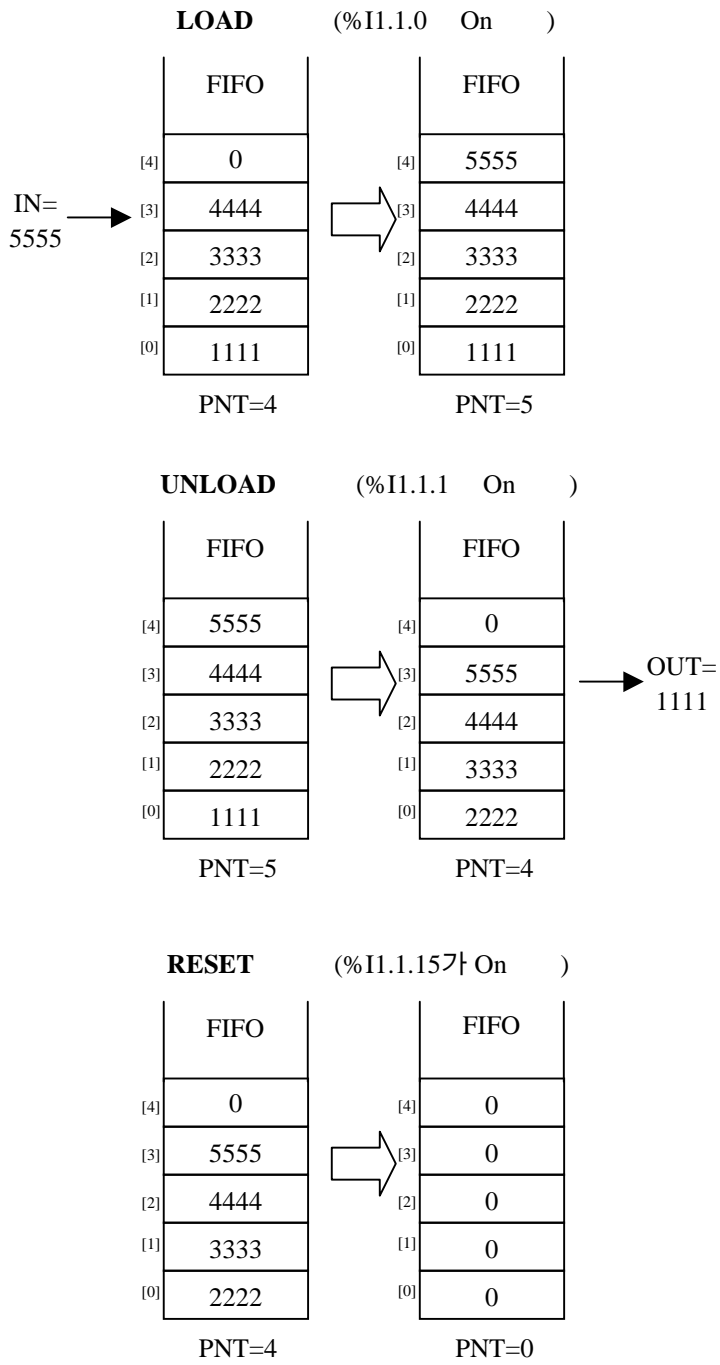
FIFO_***

2가

가

2가

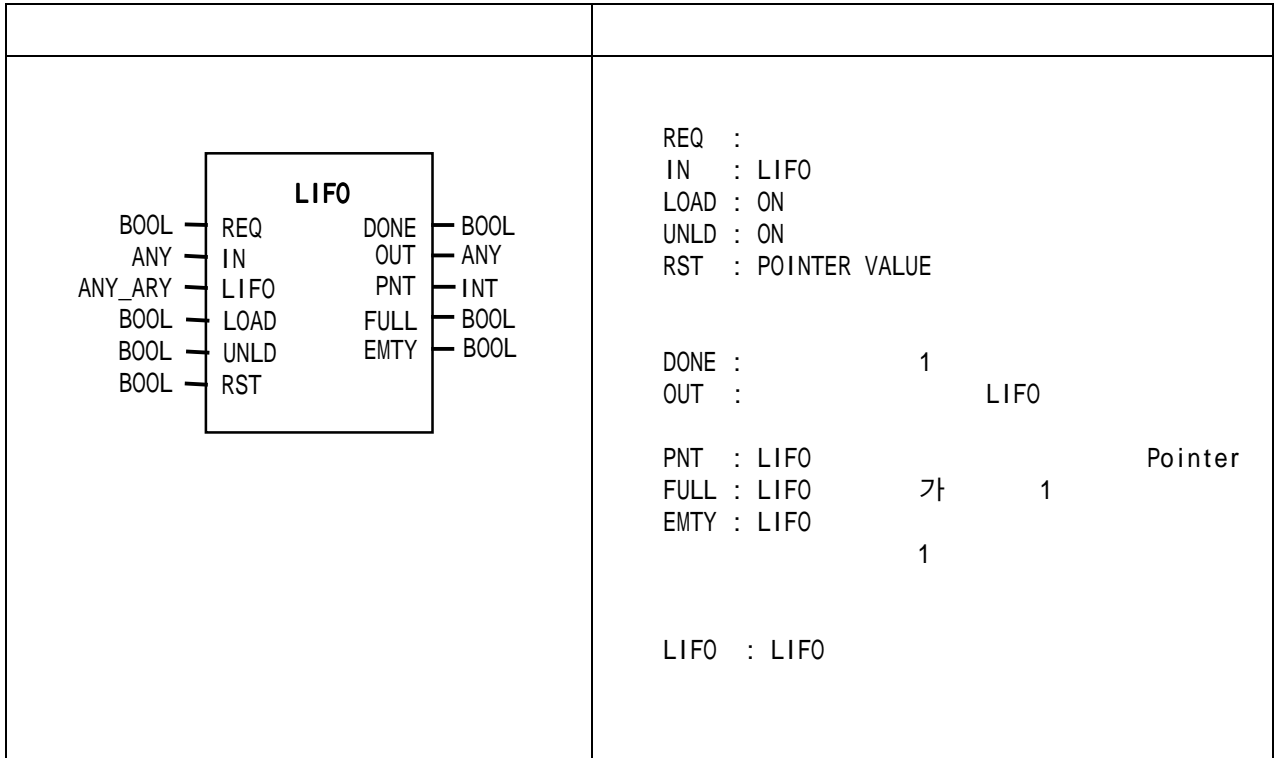
- (1) (%I1.1.0, %I1.1.1, %I1.1.15) FIFO_INT가
- (2) %I1.1.0 On Load . 5555가 FIFO PNT_INDEX가 1
- (3) %I1.1.1 On Unload . FIFO 1111
- (4) %I1.1.15가 On Reset . FIFO 0 (Clear)
- , PNT_INDEX가 0 , EMPTY_FLAG가 On



LIFO_***

LIFO	Load/Unload
()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



LIFO IN LIFO Load LIFO Unload .

LIFO Unload Unload 0

RST가 PNT 0 , EMTY 가 On , LIFO 0

(clear) LIFO 가 . LIFO LIFO

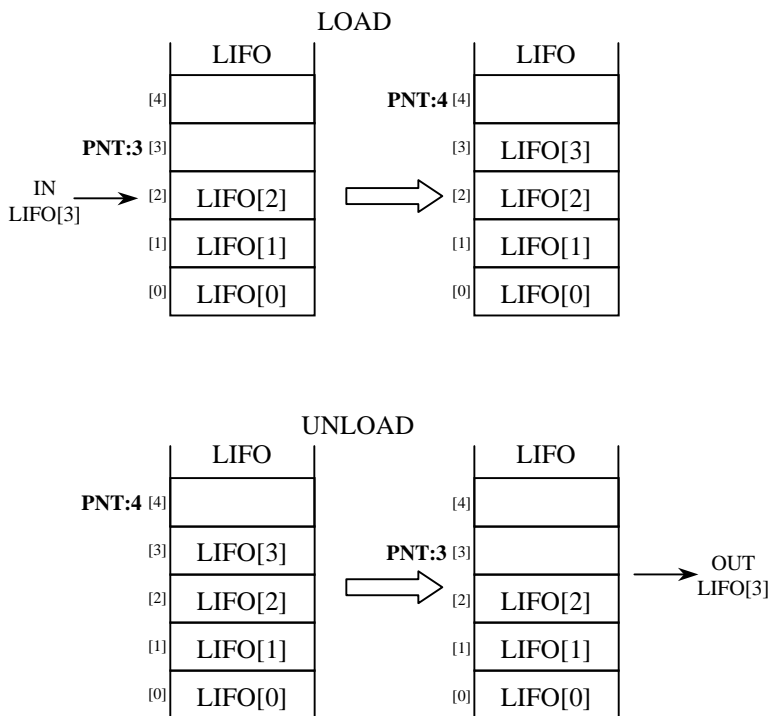
Instance Off RETAIN

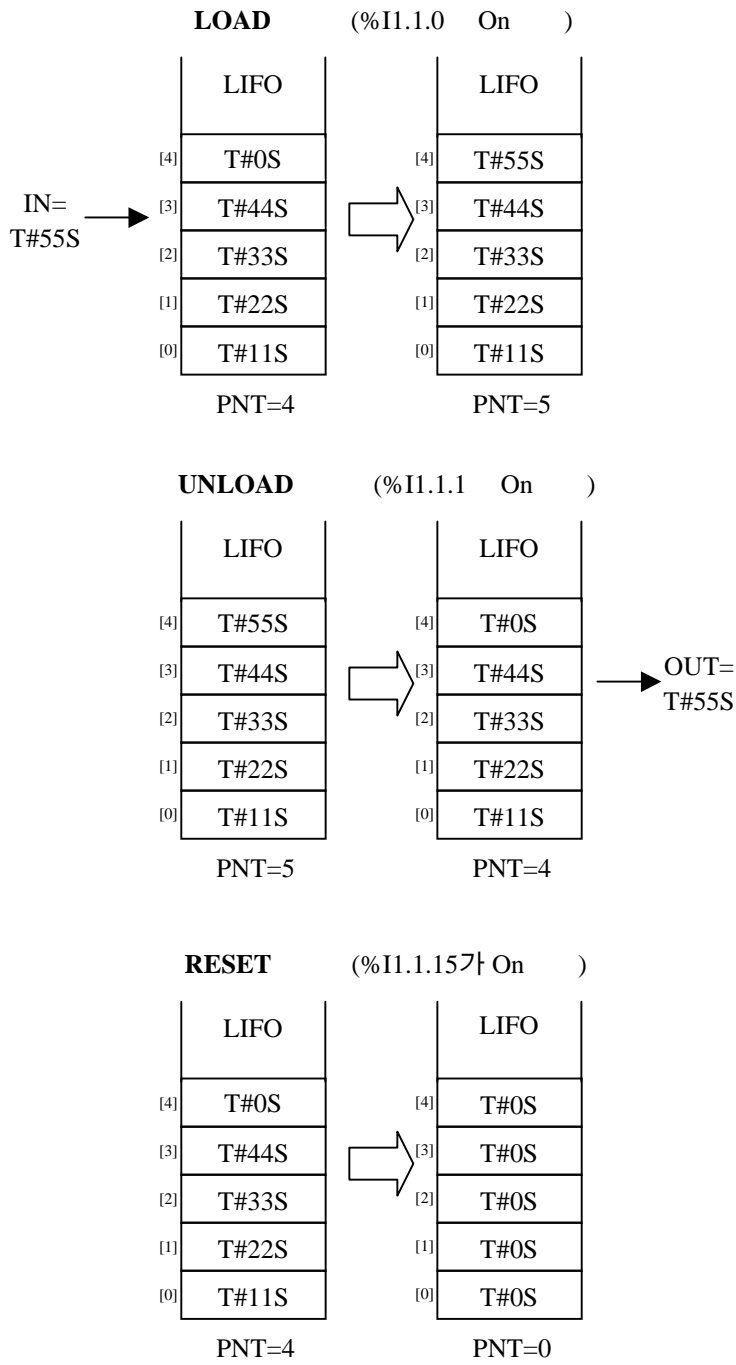
REQ 가 가 .

PNT Load IN . Load

Load Unload OUT 0 . IN OUT .

	FIFO				
LIFO_Q	BOOL	BOOL	DATA	LIFO	.
LIFO_B	BYTE	BYTE	DATA	LIFO	.
LIFO_W	WORD	WORD	DATA	LIFO	.
LIFO_DW	DWORD	DWORD	DATA	LIFO	.
LIFO_LW	LWORD	LWORD	DATA	LIFO	.
LIFO_SI	SINT	SINT	DATA	LIFO	.
LIFO_I	INT	INT	DATA	LIFO	.
LIFO_DI	DINT	DINT	DATA	LIFO	.
LIFO_LI	LINT	LINT	DATA	LIFO	.
LIFO_USI	USINT	USINT	DATA	LIFO	.
LIFO_UI	UINT	UINT	DATA	LIFO	.
LIFO_UDI	UDINT	UDINT	DATA	LIFO	.
LIFO_ULI	ULINT	ULINT	DATA	LIFO	.
LIFO_R	REAL	REAL	DATA	LIFO	.
LIFO_LR	LREAL	LREAL	DATA	LIFO	.
LIFO_TM	TIME	TIME	DATA	LIFO	.
LIFO_DAT	DATE	DATE	DATA	LIFO	.
LIFO_TOD	TOD	TOD	DATA	LIFO	.
LIFO_DT	DT	DT	DATA	LIFO	.

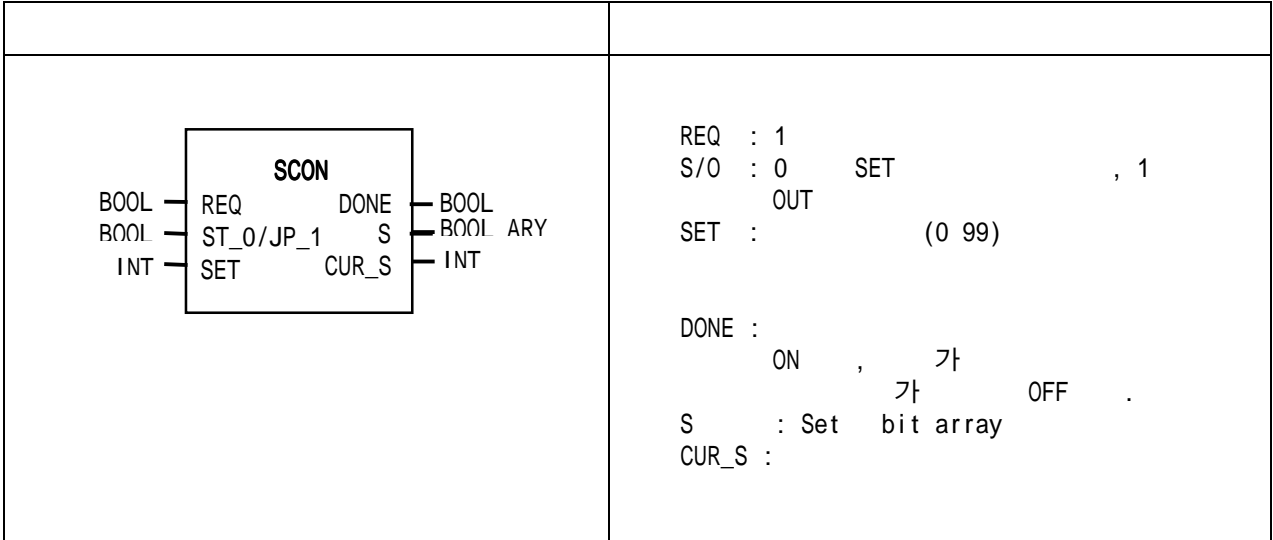




SCON

()

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



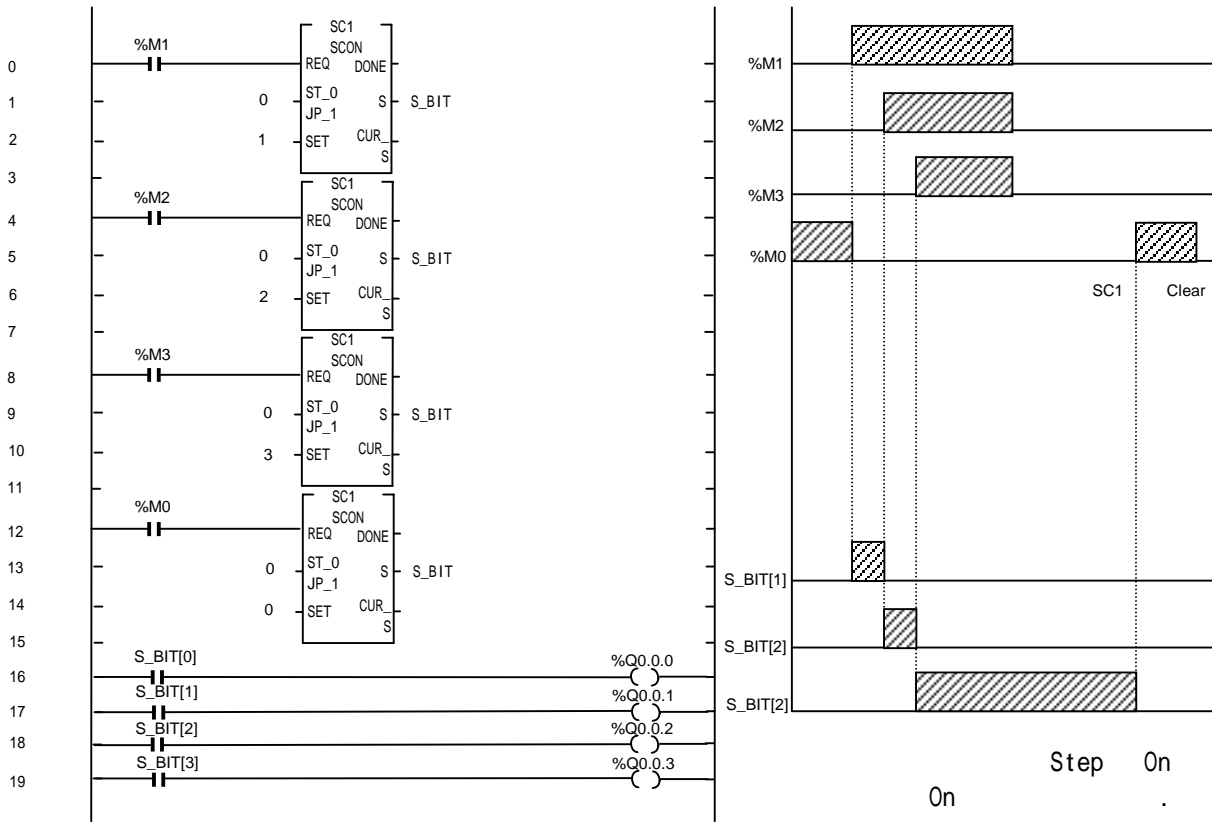
- (: S00, G01, 1
 : S00.S[1], G01.S[1], 1.S[1])

SET (ST_0/JP_1 = 0)
 - 가 On 가 On 가 On .
 - 가 On Off On .
 - On On .
 - Sxx.S[0]가 On SET Clear .

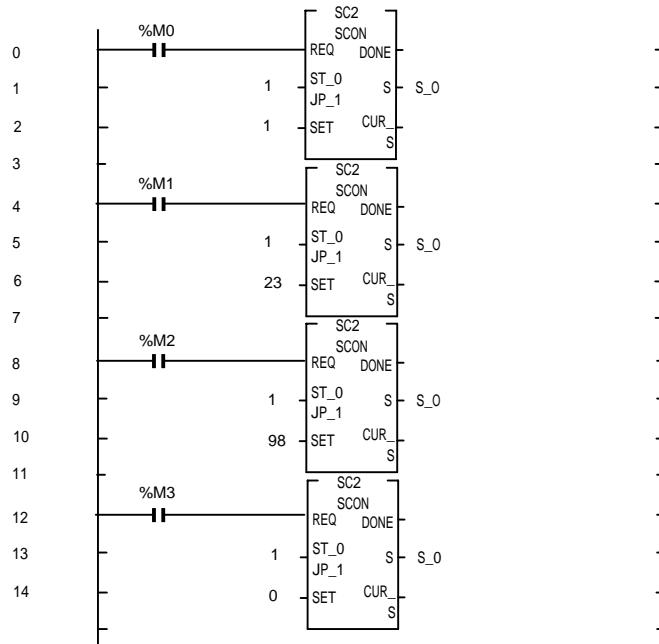
JUMP (ST_0/JP_1 = 1)
 - 가 On On .
 - On .
 - 가 On Off On .
 - Sxx.S[0] On .

(SET) (0 99) 가
 DONE OFF ,

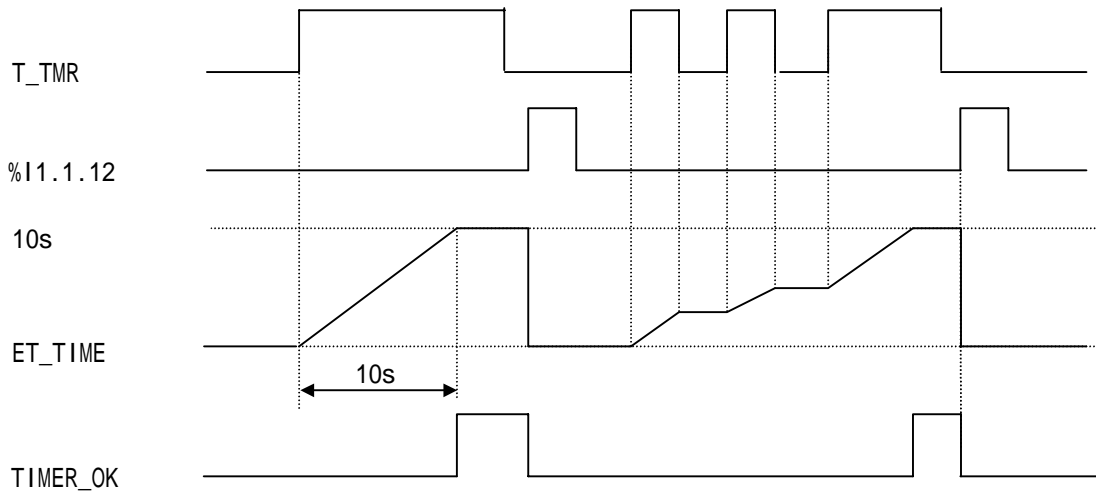
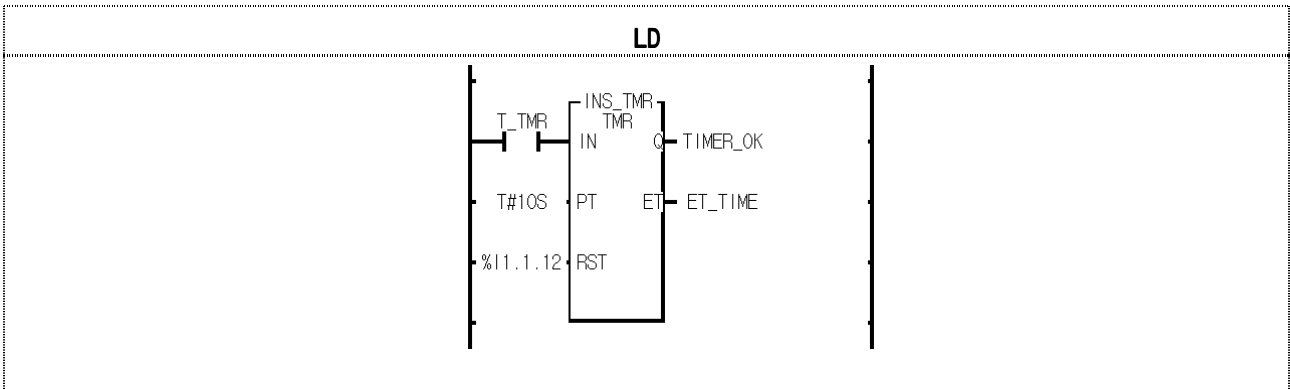
SET (ST_0/JP_1 = 0)
SC1



JUMP (ST_0/JP_1 = 1)
SC2



NO	%M1	%M2	%M3	%M4	S_0 [1]	S_0 [23]	S_0 [98]	S_0 [0]
1	On	Off	Off	Off				
2	On	On	Off	Off				
3	On	On	On	Off				
4	On	On	On	On				

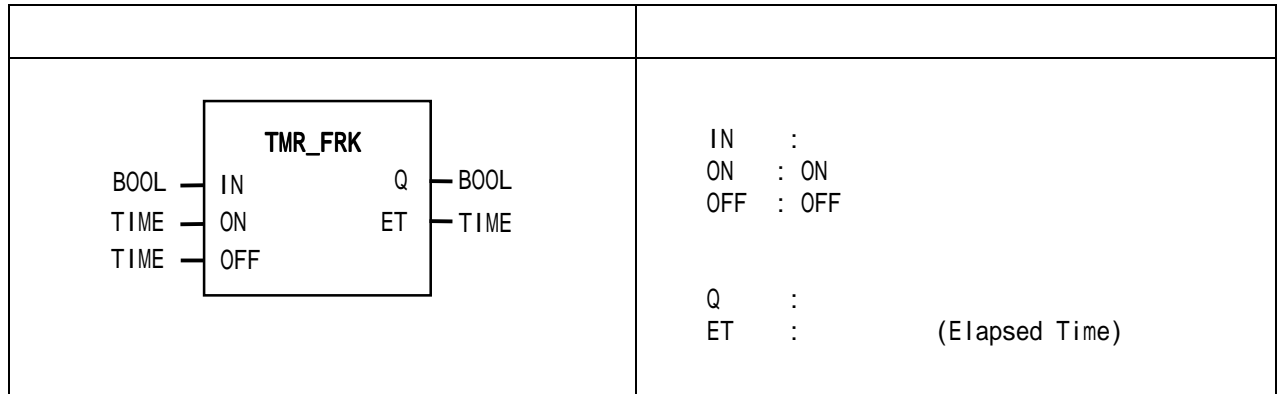


- (1) T_TMR 1 10 가 TIMER_OK가 1
- (2) T_TMR 1 ET_TIME
- (3) ET_TIME 10 T_TMR 0
- (4) T_TMR 1
- (5) %I1.1.12 가 1 ET_TIME TIMER_OK 0
- (Clear)

TMR_FRK



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



TMR_FLK IN 1 Q 1 , ON Q 1 .

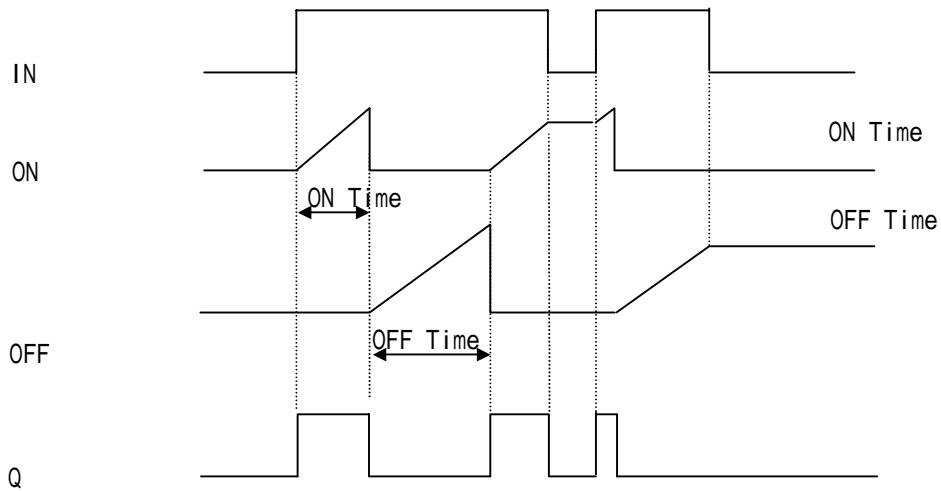
ON OFF Q 0 .

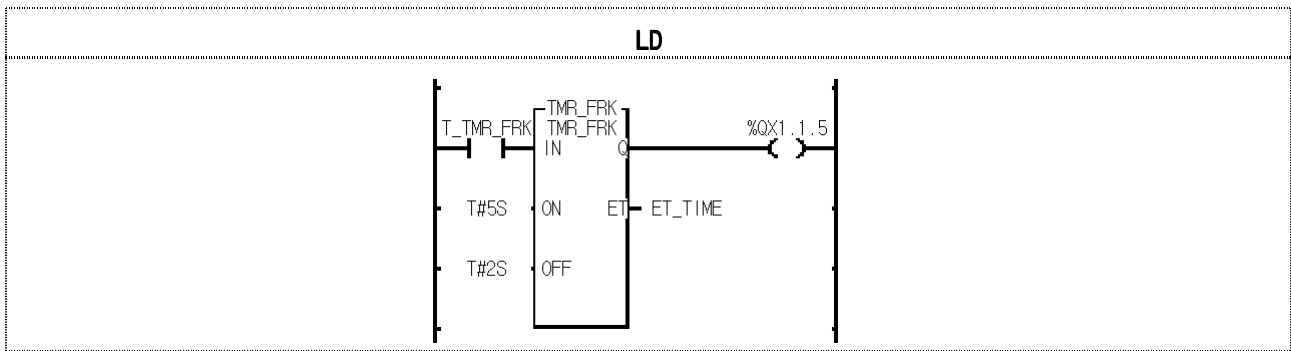
IN 0 ON OFF , IN 0 가 IN

1 가

IN 0 Q 0 .

ON 0 Q 0 .



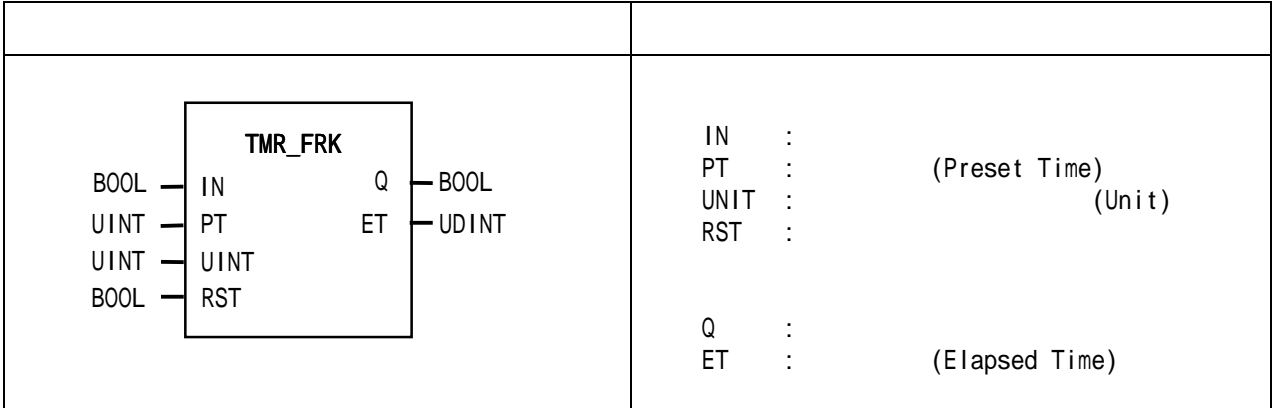


- (1) T_TMR_FRK가 0 1 , TMR_FRK
- (2) T_TMR_FRK가 1 ON 5 %QX1.1.5 1
- (3) T_TMR_FRK가 1 ON OFF 2
- (3) %QX1.1.5 0
- (3) T_TMR_FRK가 1 Q가 1 Q가 0
- (4) ET_TIME
- (4) T_TMR_FRK가 0 %QX1.1.5 0
- T_TMR_FRK가 1

TMR_UNIT



CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

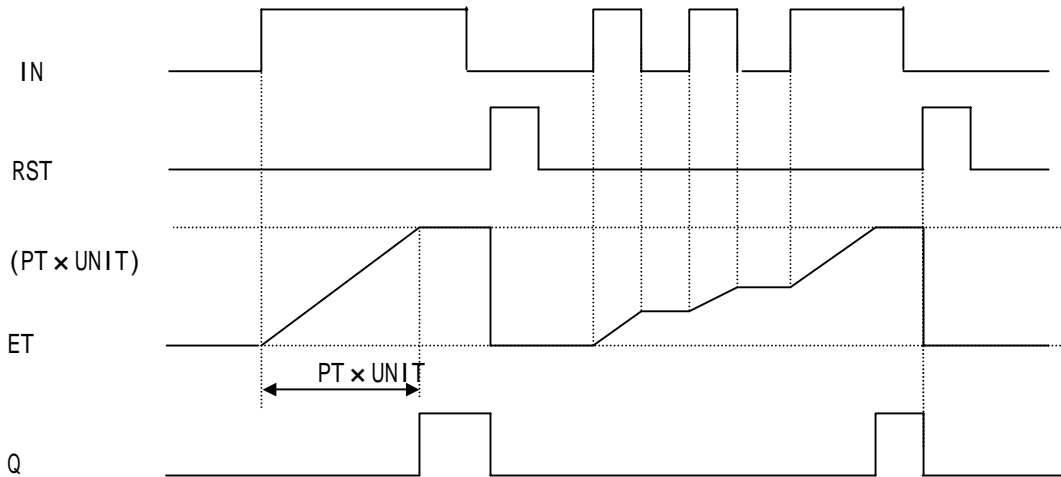


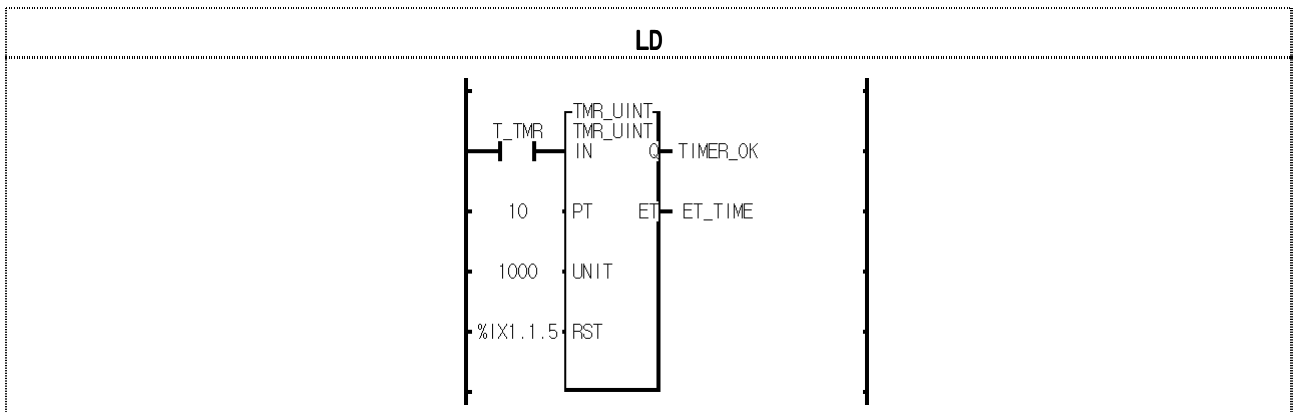
TMR_UINT IN 1 ET .

1 ET가 가 IN 0 가 IN

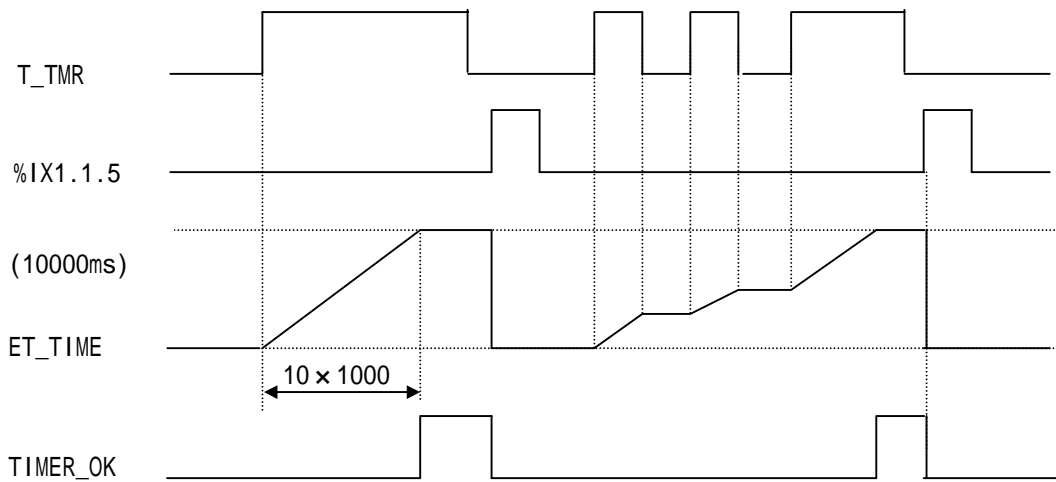
Reset Q 0 .. 0 .

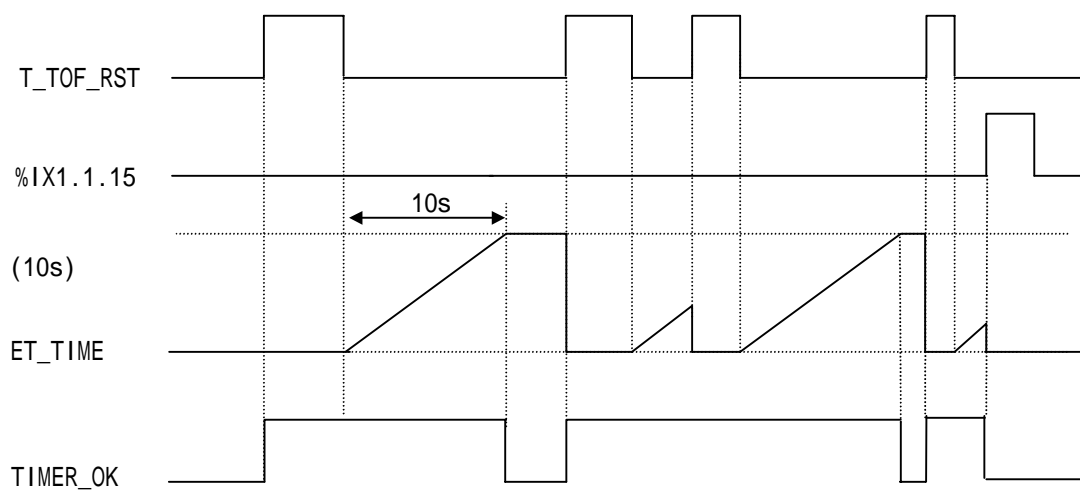
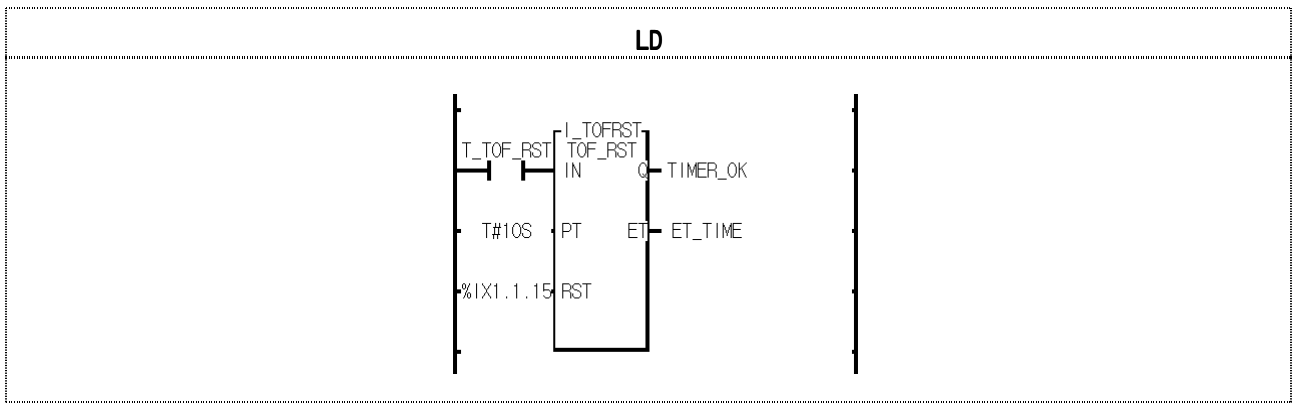
PT*UNIT[mSec]



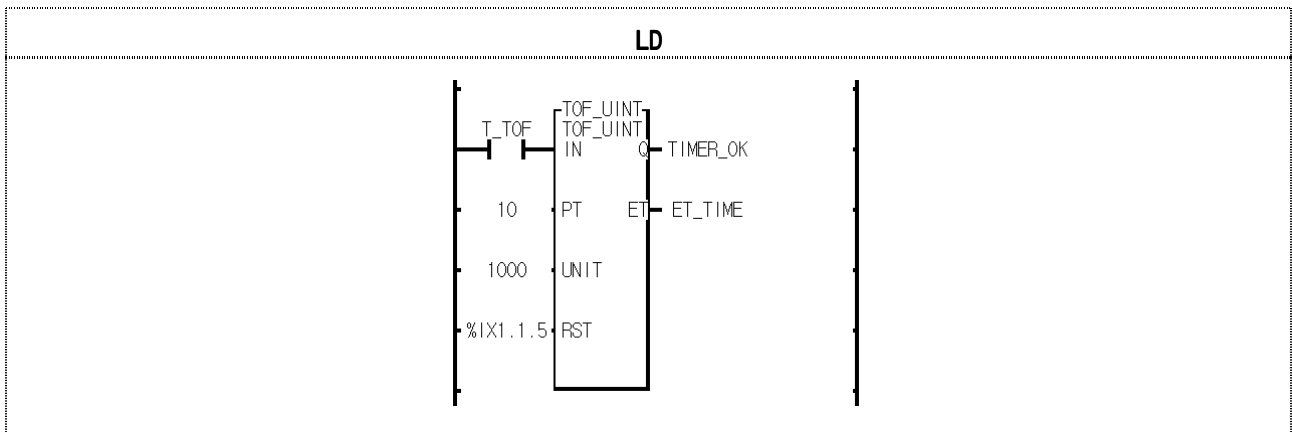


- (1) $PT * UNIT [ms] = 10 * 1000 [ms] = 10 [s]$ 가
- (2) T_TMR 1 10 가 TIMER_OK가 1
- (3) T_TMR 1 ET_TIME
- (4) ET_TIME 10 T_TMR 0
- (5) T_TMR 1
- (6) %IX1.1.5가 1 ET_TIME TIMER_OK 0
- (Clear)

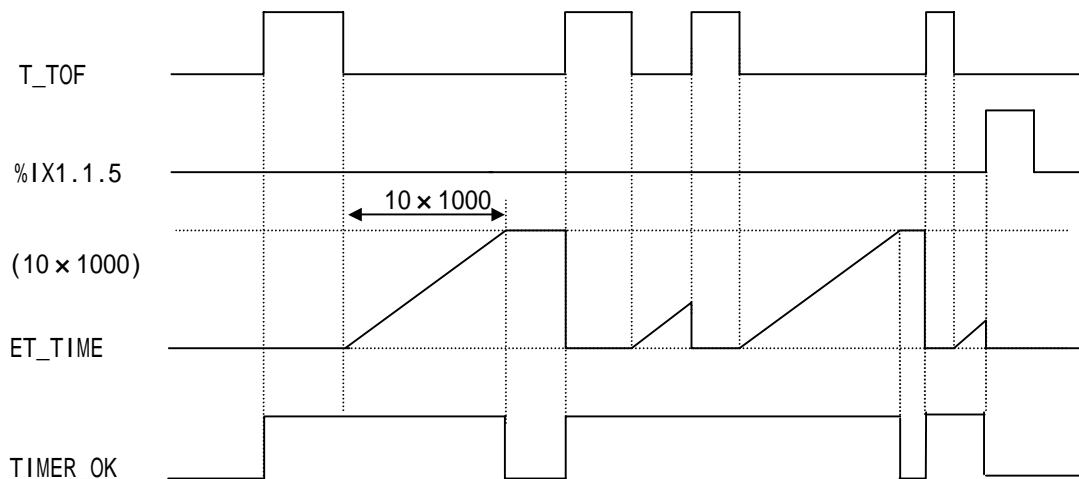




- (1) T_TOF_RST가 1 , TIMER_OK 1 T_TOF_RST가 0
10s TIMER_OK가 0 .
- (2) T_TOF_RST가 0 10 1 가 .
- (3) ET_TIME .
- (4) %IX1.1.15가 1 TIMER_OK ET_TIME 0 (Clear) .



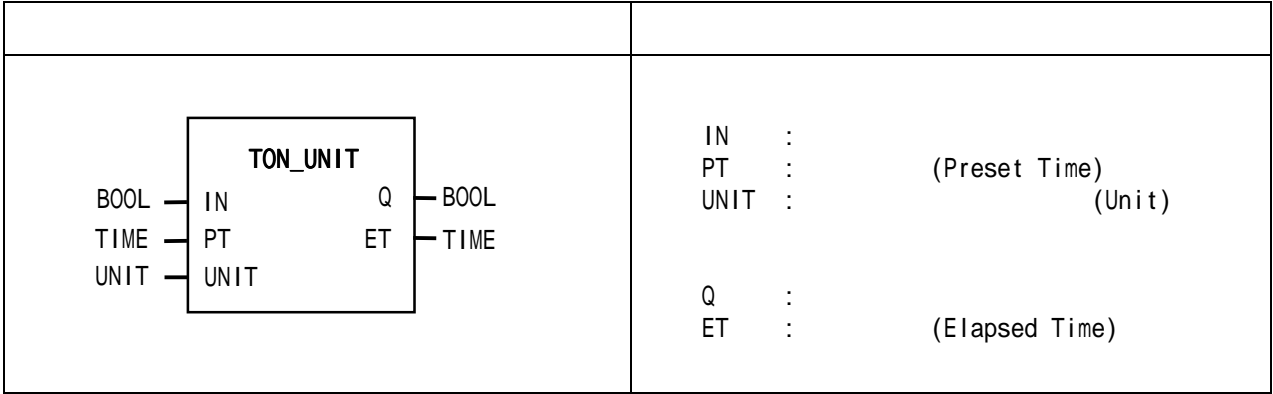
- (1) $PT \cdot UNIT[ms] = 10 \cdot 1000[ms] = 10[s]$ 가
- (2) T_TOF가 1, TIMER_OK 1, T_TOF가 0, 10
TIMER_OK가 0
- (3) T_TOF가 0, 10, 1, 가
- (4) ET_TIME
- (5) %IX1.1.5가 1, TIMER_OK, ET_TIME, 0 (Clear)



TON_UNIT

ON

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

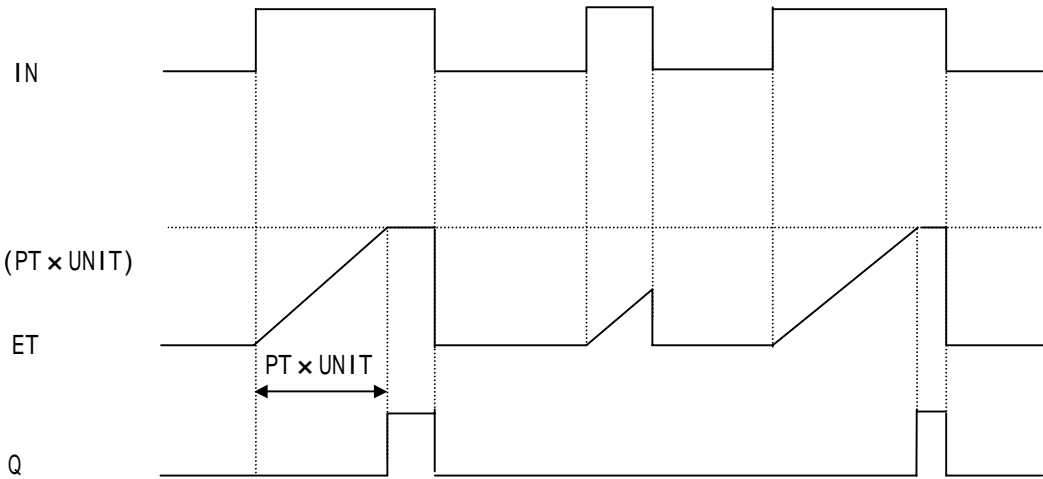


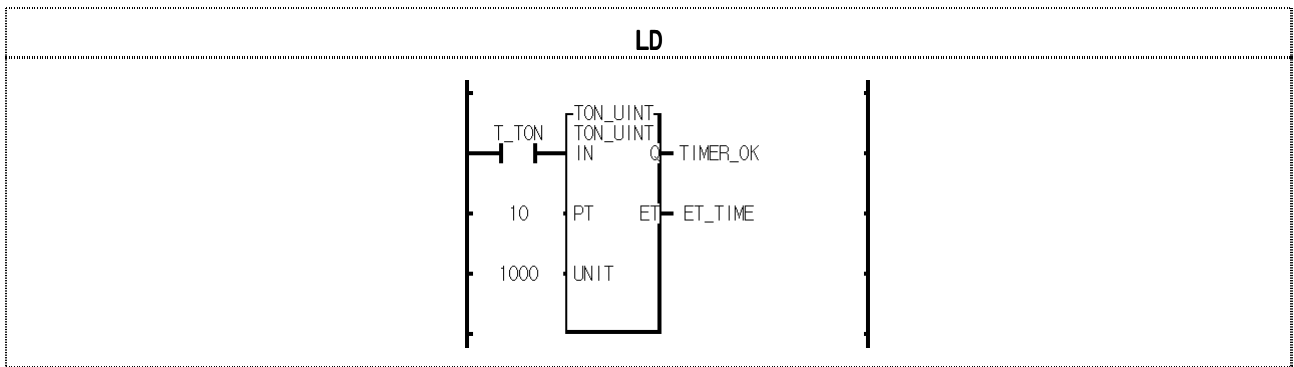
TON_UNIT IN 1 ET

 ET가 IN 0 , ET 0

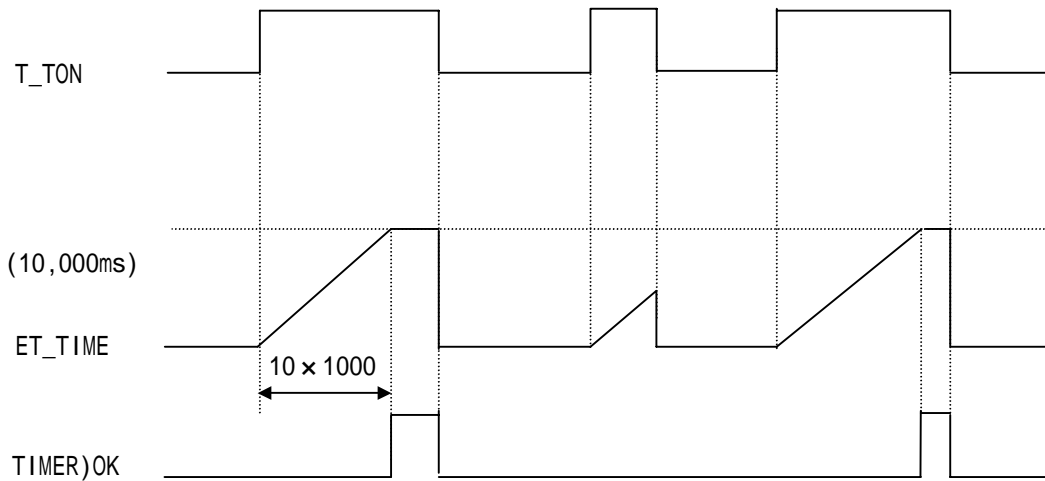
Q가 1 IN 0 , Q 0

 PT × UNIT[ms]





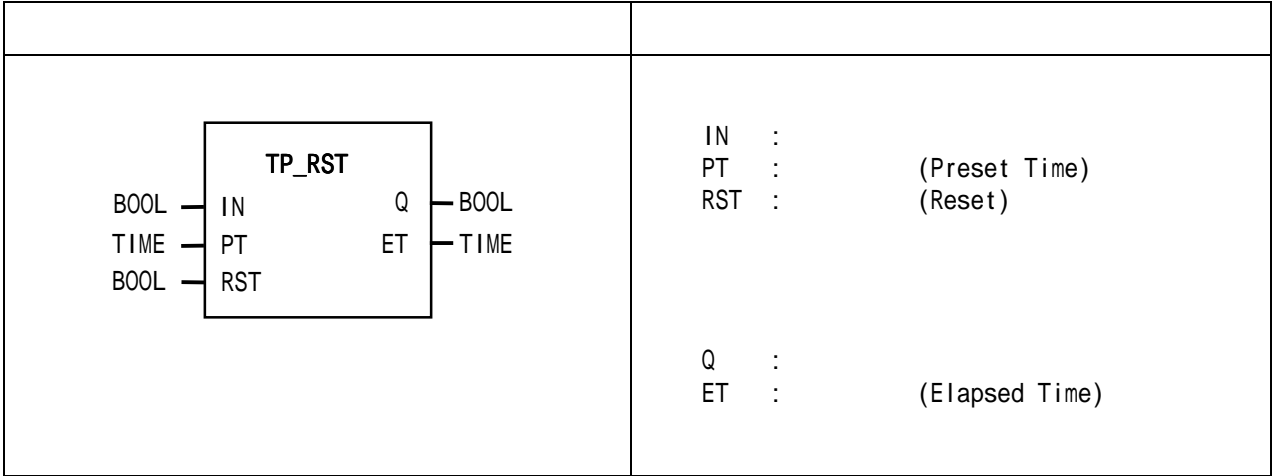
- (1) $PT \cdot UNIT[s] = 10 \cdot 1000[s] = 10[s]$ 가
- (2) T_TON On , 10 가 TIMER_OK가 1
- (3) T_TON On ET_TIME
- (4) ET_TIME 10 T_TON 0 , ET_TIME 0
- (5) TIMER_OK가 1 T_TON 0 , TIMER_OK 0 ET_TIME 0



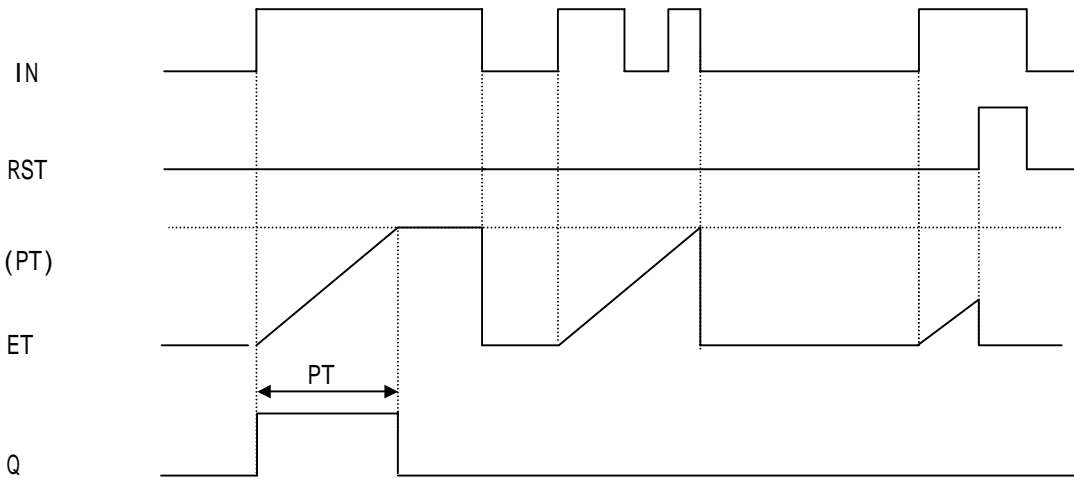
TP_RST

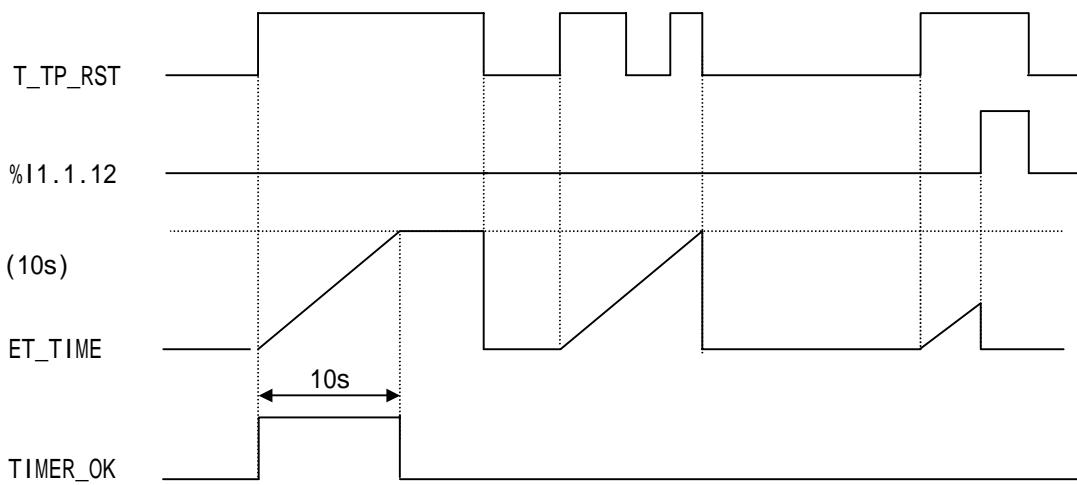
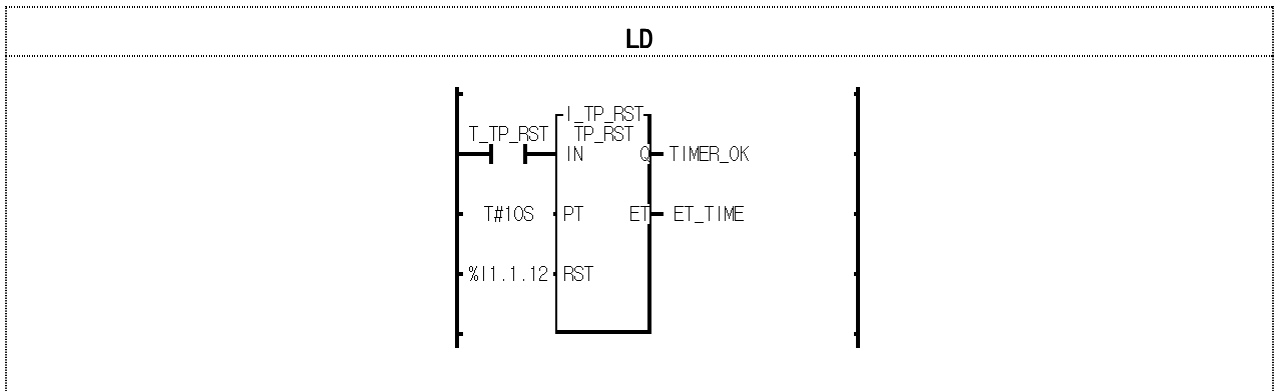
OFF가 가

CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							



TP_RST IN 1 Q 1 ,
Q 0
ET IN 1 가 PT 가 IN 0 0
(clear)
Q가 1 () IN 1,0
Reset Q 0 0 .

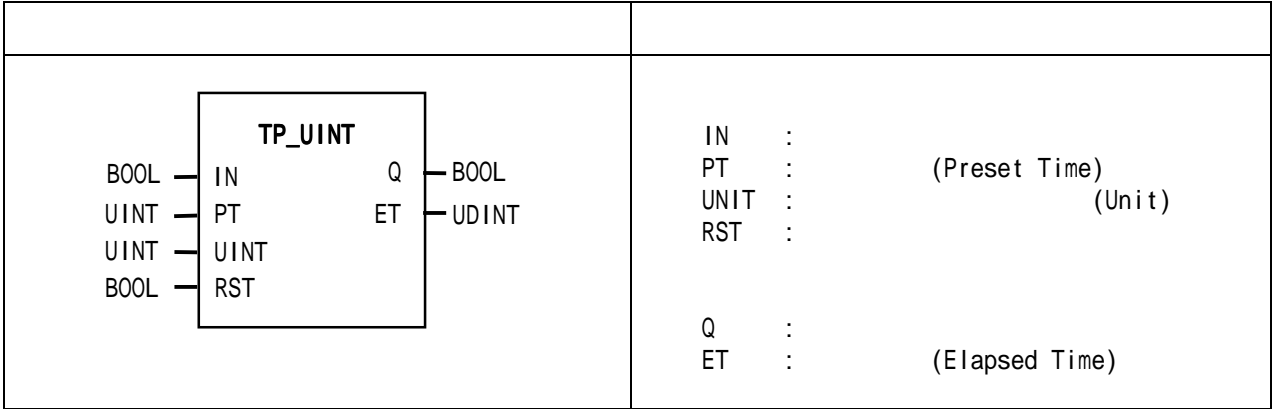




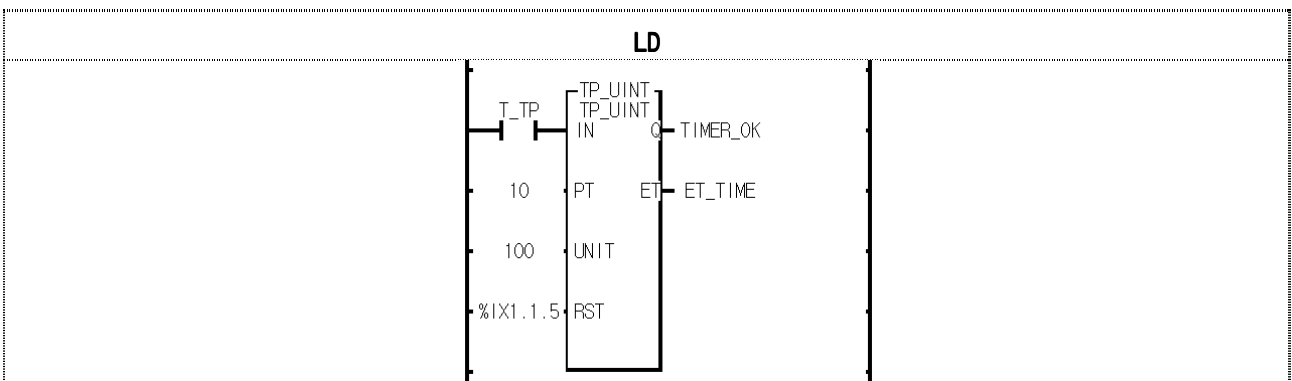
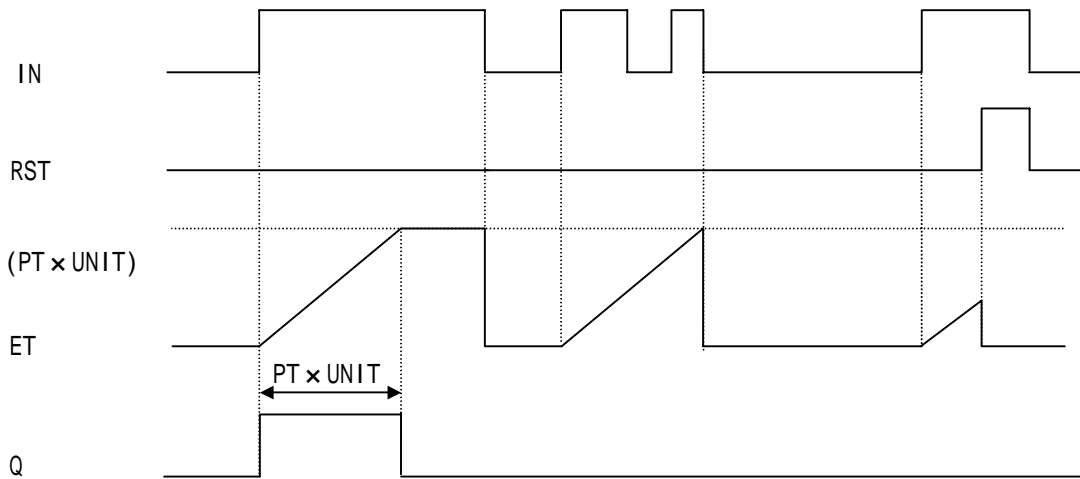
- (1) T_TP_RST가 1 가 10 가 TIMER_OK가 1 10 가 TIMER_OK
0 가 가 10 T_TP_RST
- (2) ET_TIME 가 10S T_TP_RST가 0 0
- (3) %11.1.12가 1 TIMER_OK ET_TIME 0 (Clear)

TP_UNIT

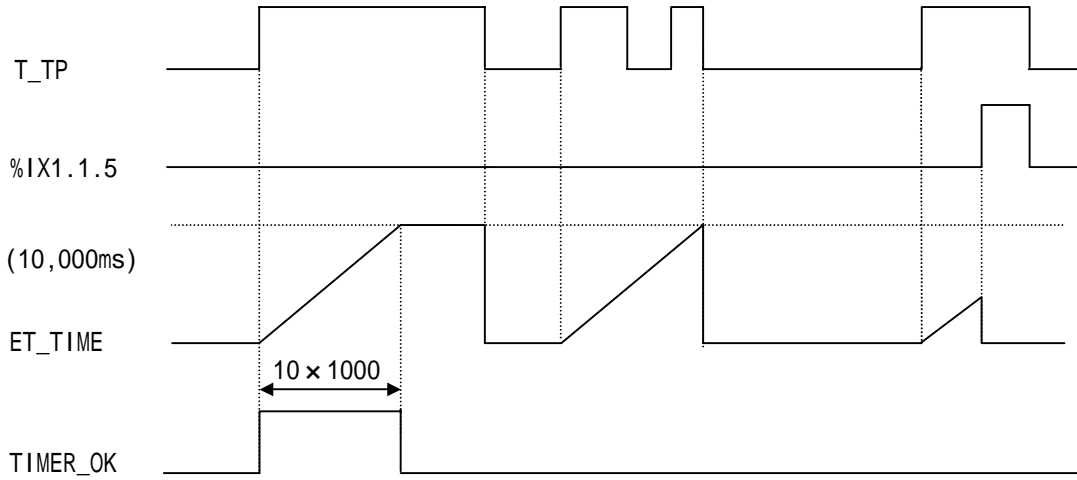
CPU	GMR	GM1	GM2	GM3	GM4	GM6	GM7
가							

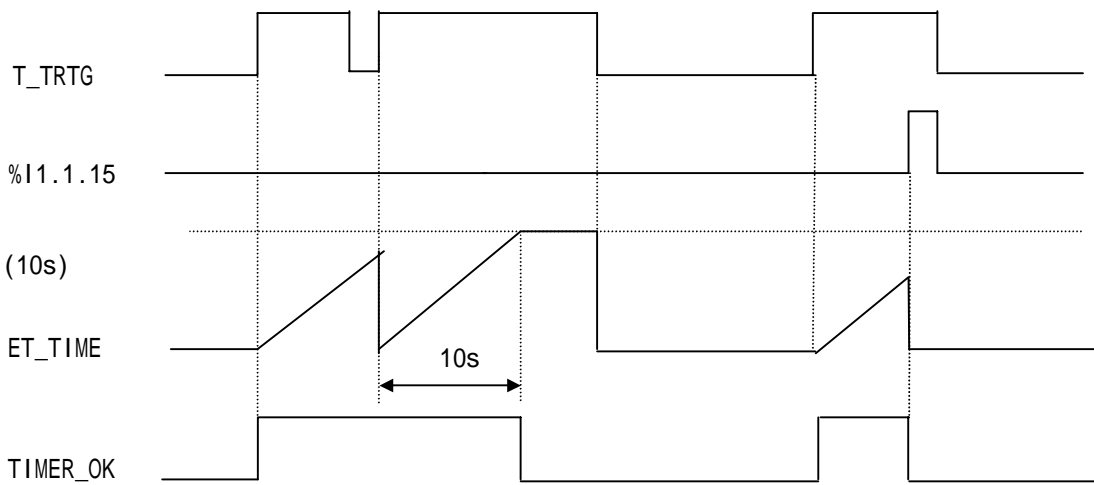
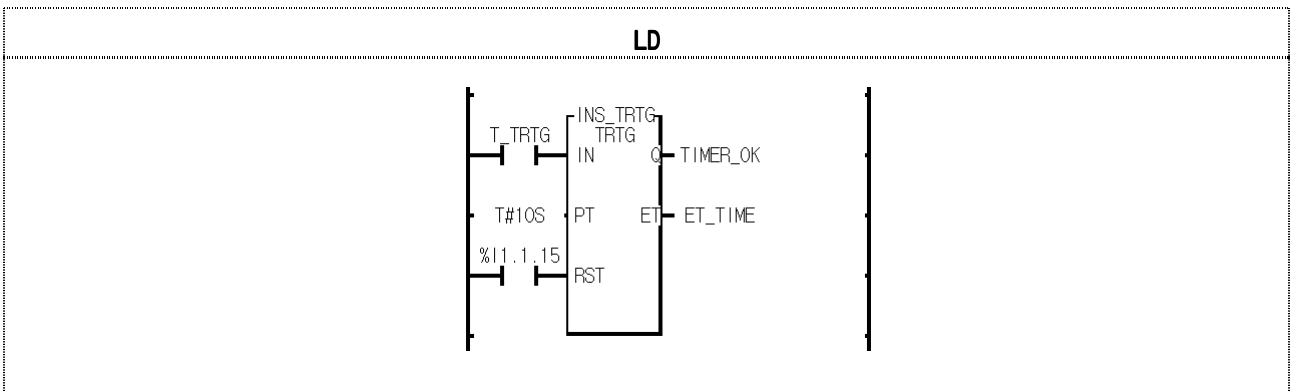


TP_UNIT IN 1 Q 1 ,
 Q 0 .
 ET IN 1 가 PT 가 IN 0 0
 (Clear)
 Q가 1 () IN 1,0
 Reset Q 0 0 .
 PT*UNIT[ms]



- (1) $PT*UNIT[ms] = 10*1000[ms] = 10[s]$ 가 .
- (2) T_TP가 1 가 가 TIMER_OK가 1 10 가 TIMER_OK 0
- (3) ET_TIME 가 10,000 . T_TP가 0 0 .
- (4) %IX1.1.5가 1 TIIMER_OK ET_TIME 0 (Clear) .





- (1) T_TRTG가 0 → 1 → 10 → TIMER_OK 1 . 가 가 T_TRTG
가 0 1 ET_TIME 0 .
- (2) T_TRTG가 1 → 0 → 10 → TIMER_OK 1 .
- (3) ET_TIME 가 T#10S . T_TRTG가 0 → 0 .
- (4) %I1.1.15가 1 → TIMER_OK ET_TIME 0 (Clear) .

- (1) $PT \cdot UNIT[ms] = 10 \cdot 1000[ms] = 10[s]$ 가 .
- (2) T_TRTG가 0 1 10 TIMER_OK 1 . 가 가 T_TRTG
가 0 1 ET_TIME 0 .
- (3) T_TRTG가 1 0 10 TIMER_OK 1 .
- (4) ET_TIME 가 10,000 . T_TRTG가 0 0 .
- (5) %IX1.1.5가 1 TIMER_OK ET_TIME 0 (Clear) .

