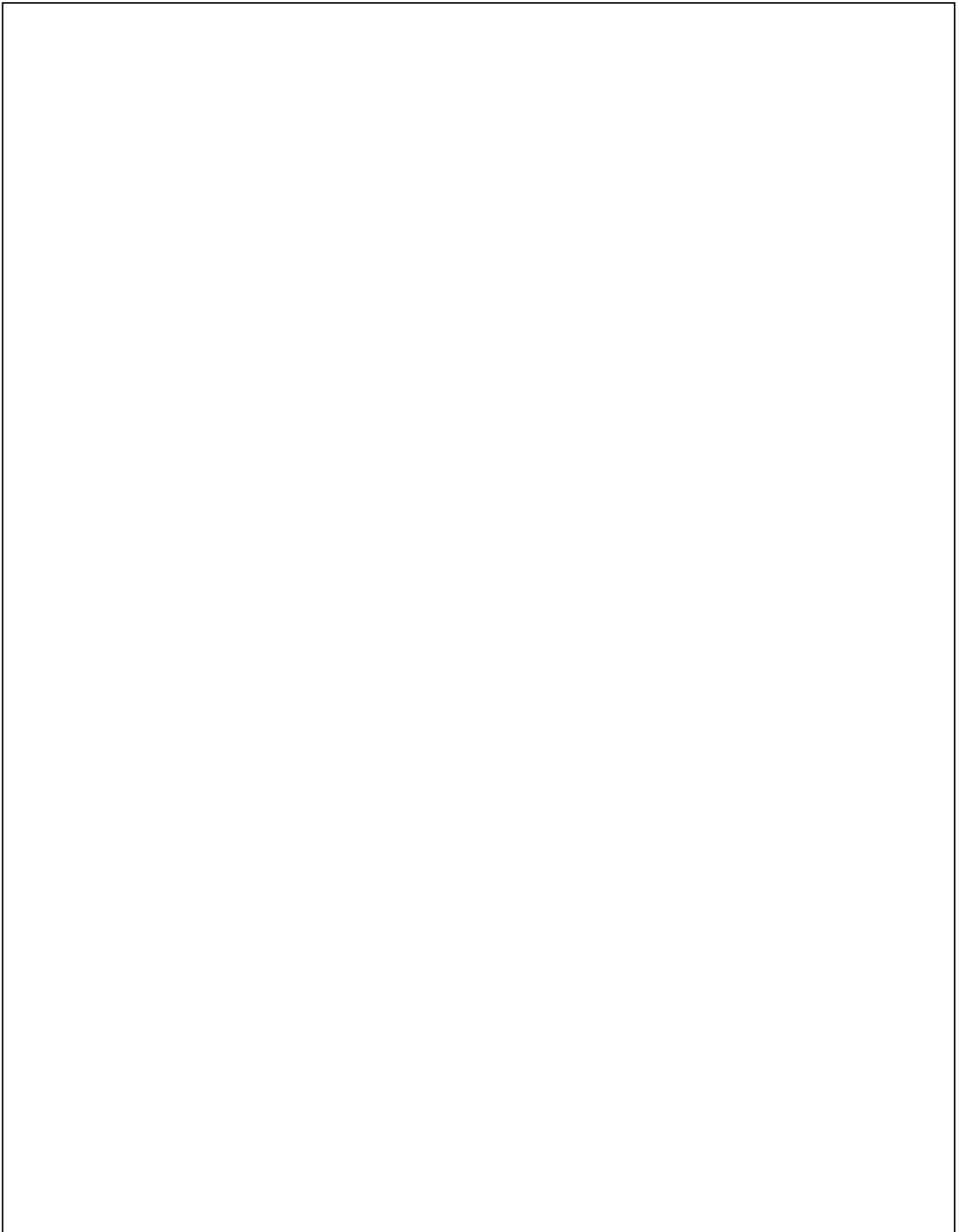


PLC

[]



4 PLC

	21
(Bit)	11
(Bit)	36
(16Bits, 32Bits)	12
(16Bits, 32Bits)	8
(16Bits, 32Bits)	12
(16Bits, 32Bits)	12
	5
MC ACCESS	3

- PLC 가 .
(MSW-MCS MMI ' ' MENU -> ' PLC ' - >
' Timer,Counter,F ' 가 .)
- Step 가 가 .(
PLC)

F2.0~F2.F	PLC (msec)
F3.0~F3.F	PLC
F4.0~F4.F	Scan PLC (msec)
F5.0~F5.F	Scan PLC

- PLC Program 1,000 STEP Compile 가 .
- PLC Program STEP ‘/’
- Scan PLC 가 .

4.1

4.1 PLC

			Page
LOAD	, ,TIMER, COUNTER	(a)	4-8
LOAD NOT	, ,TIMER, COUNTER	(b)	4-8
AND	, ,TIMER, COUNTER	(a)	4-9
AND NOT	, ,TIMER, COUNTER	(b)	4-9
OR	, ,TIMER, COUNTER	(a)	4-10
OR NOT	, ,TIMER, COUNTER	(b)	4-10
AND LOAD		Block	4-11
OR LOAD		Block	4-11
OUT	,	(a)	4-12
OUT NOT	,	(b)	4-
D	,	1 Pulse	4-12
D NOT	,	1 Pulse	4-13
TMR	TIMER	Timer()	4-37
TON	TIMER	On-Delay Timer()	4-38
TOFF	TIMER	Off-Delay Timer()	4-39
CTR	COUNTER	Counter	4-40
SET	,	Bit Self-holding(ON)	4-13
RST	,	Bit Self-holding(OFF)	4-13
MCS		Interlock Set	4-14
MCS NOT		Interlock Reset	4-14
END		PLC	4-63

) 1. PLC Program 1,000 STEP Compile 가 .
 2. PLC Program STEP ‘/*’ .

4.2 가 PLC

			Page
(Bit)			
LOADP	,	Edge (a)	4-15
LOADN	, , TIMER, COUNTER	Edge (b)	4-16
ANDP	, , TIMER, COUNTER	Edge (a)	4-16
ANDN	, , TIMER, COUNTER	Edge (b)	4-16
ORP	, , TIMER, COUNTER	Edge (a)	4-17
ORN	, , TIMER, COUNTER	Edge (b)	4-17
OUTP	,	1 Pulse	4-18
OUTN	,	1 Pulse	4-18
MPUSH		PUSH	4-14
MLOAD		READ	4-14
MPOP		POP	4-14
(Bit)			
LOAD= DLOAD=	, , TIMER, COUNTER	16Bits 32Bits EQ (=)	4-19
LOAD> DLOAD>	, , TIMER, COUNTER	16Bits 32Bits GT (>)	4-20
LOAD>= DLOAD>=	, , TIMER, COUNTER	16Bits 32Bits GE (>=)	4-21
LOAD< DLOAD<	, , TIMER, COUNTER	16Bits 32Bits LT (<)	4-22
LOAD<= DLOAD<=	, , TIMER, COUNTER	16Bits 32Bits LE (<=)	4-23
LOAD<> DLOAD<>	, , TIMER, COUNTER	16Bits 32Bits NE (<>)	4-24
AND= DAND=	, , TIMER, COUNTER	16Bits 32Bits EQ (=)	4-25
AND> DAND>	, , TIMER, COUNTER	16Bits 32Bits GT (>)	4-27
AND>= DAND>=	, , TIMER, COUNTER	16Bits 32Bits GE (>=)	4-26
AND< DAND<	, , TIMER, COUNTER	16Bits 32Bits LT (<)	4-29
AND<= DAND<=	, , TIMER, COUNTER	16Bits 32Bits LE (<=)	4-28
AND<> DAND<>	, , TIMER, COUNTER	16Bits 32Bits NE (<>)	4-30

OR= DOR=	, , TIMER, COUNTER	16Bits	32Bits	EQ (=)	4-31
OR> DOR>	, , TIMER, COUNTER	16Bits	32Bits	GT (>)	4-33
OR>= DOR>=	, , TIMER, COUNTER	16Bits	32Bits	GE (>=)	4-32
OR< DOR<	, , TIMER, COUNTER	16Bits	32Bits	LT (<)	4-34
OR<= DOR<=	, , TIMER, COUNTER	16Bits	32Bits	LE (<=)	4-35
OR<> DOR<>	, , TIMER, COUNTER	16Bits	32Bits	NE (<>)	4-36
(16Bits, 32Bits)					
MOV DMOV	, , TIMER, COUNTER	16Bits	32Bits		4-41
ADD DADD	, , TIMER, COUNTER	16Bits	32Bits	가	4-42
SUB DSUB	, , TIMER, COUNTER	16Bits	32Bits		4-43
MUL DMUL	, , TIMER, COUNTER	16Bits	32Bits		4-44
DIV DDIV	, , TIMER, COUNTER	16Bits	32Bits	()	4-45
MOD DMOD	, , TIMER, COUNTER	16Bits	32Bits	()	4-46
(16Bits, 32Bits)					
AND DAND	, , TIMER, COUNTER	16Bits	32Bits	Bit AND	4-47
OR DOR	, , TIMER, COUNTER	16Bits	32Bits	Bit OR	4-48
XOR DXOR	, , TIMER, COUNTER	16Bits	32Bits	Bit XOR	4-49
NOT DNOT	, , TIMER, COUNTER	16Bits	32Bits	Bit NOT	4-50
(16Bits, 32Bits)					
SHL DSHL	, , TIMER, COUNTER	16Bits	32Bits	Shift Left	4-51
SHR DSHR	, , TIMER, COUNTER	16Bits	32Bits	Shift Right	4-52
ROL DROL	, , TIMER, COUNTER	16Bits	32Bits	Rotate Left	4-53
ROR DROR	, , TIMER, COUNTER	16Bits	32Bits	Rotate Right	4-54
ROLC DROL	, , TIMER, COUNTER	16Bits	32Bits	Carry Rotate Left	4-55
RORC DROR	, , TIMER, COUNTER	16Bits	32Bits	Carry Rotate Right	4-56

(16Bits, 32Bits)				
EQ DEQ	, , TIMER, COUNTER	16Bits	32Bits	EQ (=) 4-57
GT DGT	, , TIMER, COUNTER	16Bits	32Bits	GT (>) 4-58
GE DGE	, , TIMER, COUNTER	16Bits	32Bits	GE (>=) 4-59
LT DLT	, , TIMER, COUNTER	16Bits	32Bits	LT (<) 4-61
LE DLE	, , TIMER, COUNTER	16Bits	32Bits	LE (<=) 4-60
NE DNE	, , TIMER, COUNTER	16Bits	32Bits	NE (<>) 4-62
LABL, JME	0~99			4-63
JPC	0~99		가 ON	4-63
JPN	0~99		가 OFF	4-63
JMP	0~99			4-63

4.3 가 MC

PLC

MC		ACCESS	
			Page
LGET	L	32Bits (L → PLC)	4-64
LPUT	L	32Bits (PLC → L)	4-65
FWR	L	L (FLASH)	4-66

) Flash Memory

10

가

Flash Memory

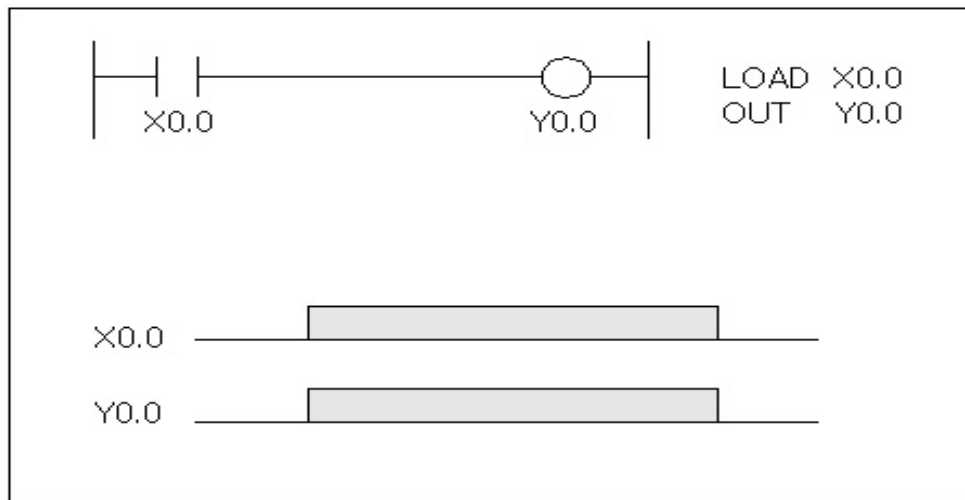
.

4.2

1) (Bit)

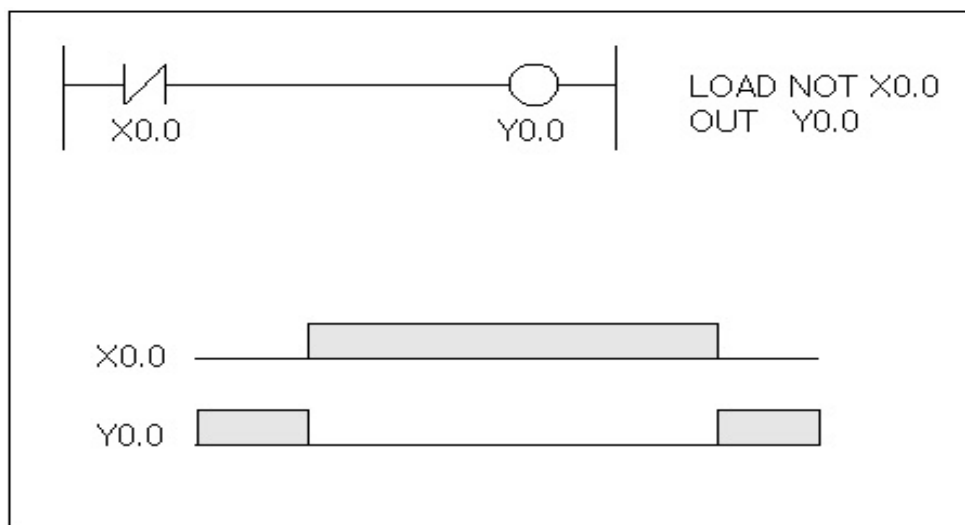
(1) LOAD

- LOAD (0 1) . (AND LOAD, OR LOAD)
- Block (AND LOAD, OR LOAD)



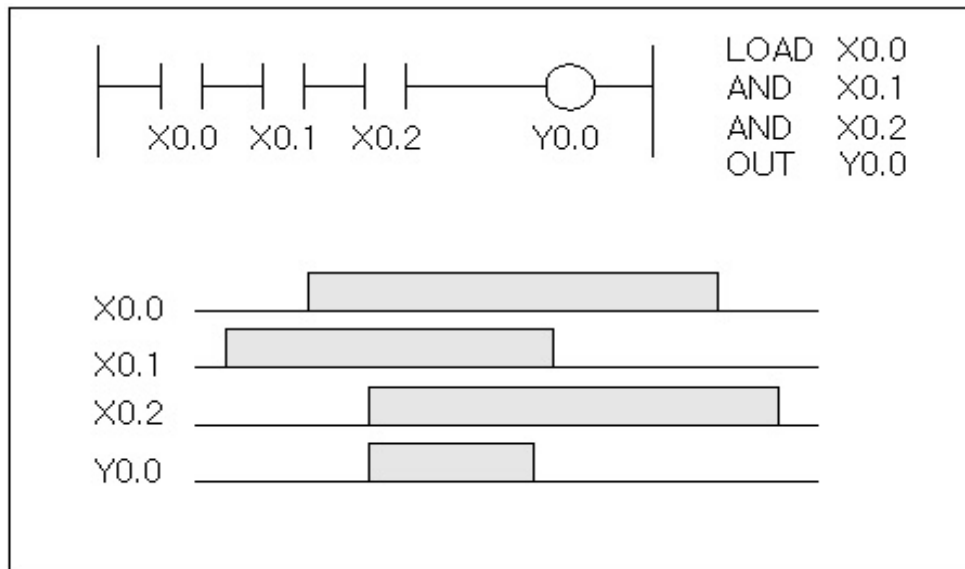
(2) LOAD NOT

- LOAD NOT . (AND LOAD, OR LOAD)
- Block (AND LOAD, OR LOAD)



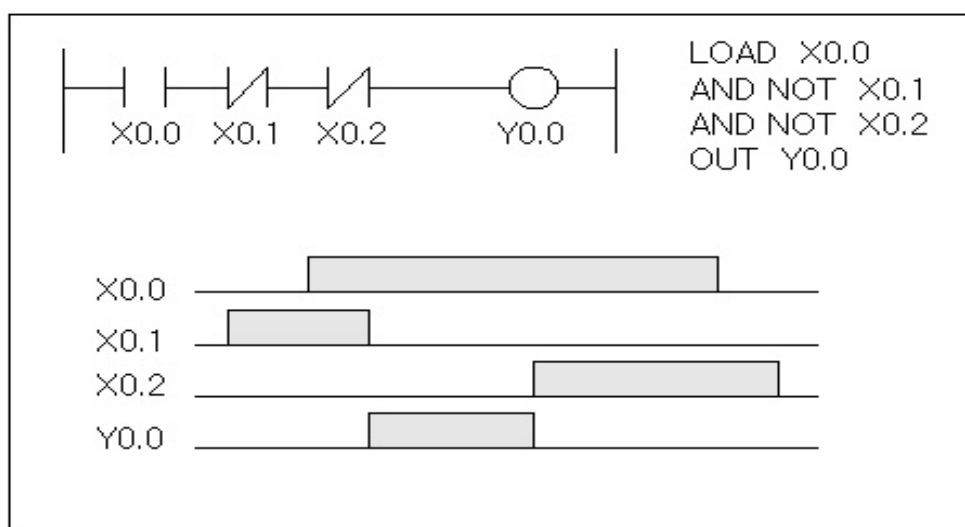
(3) AND

- AND
- AND



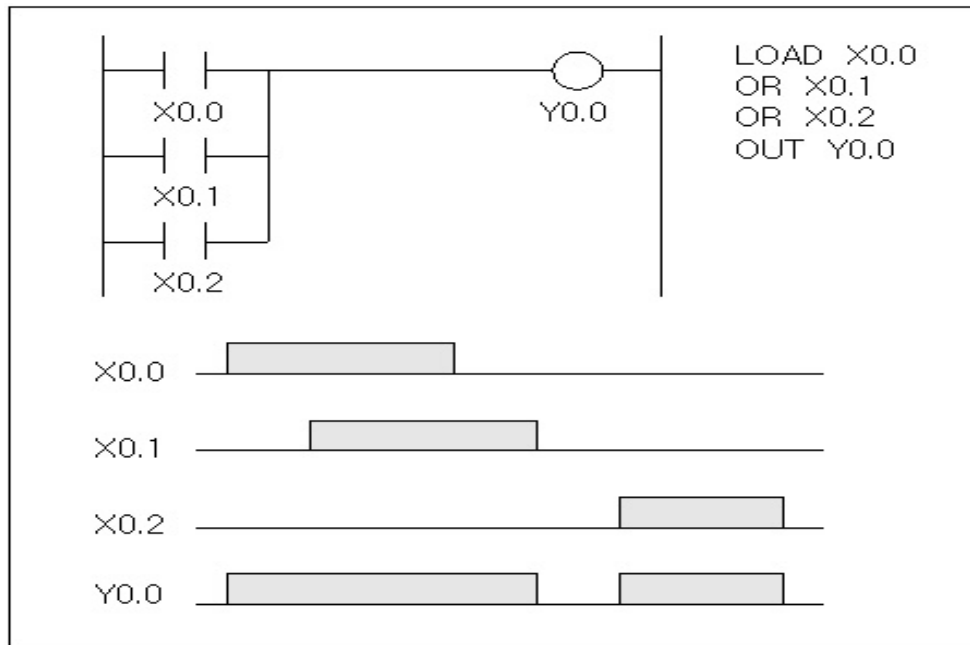
(4) AND NOT

- AND
- AND NOT



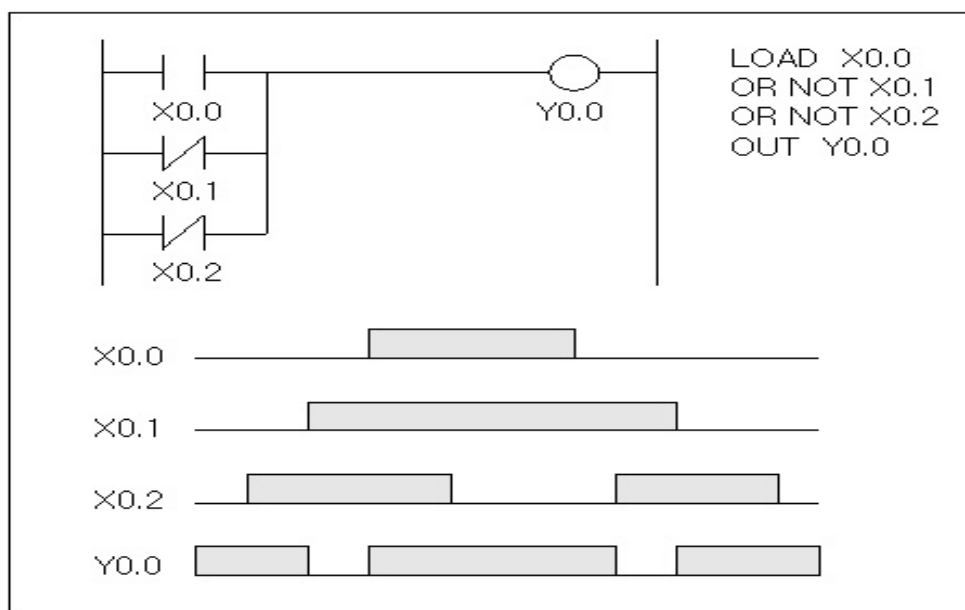
(5) OR

- OR



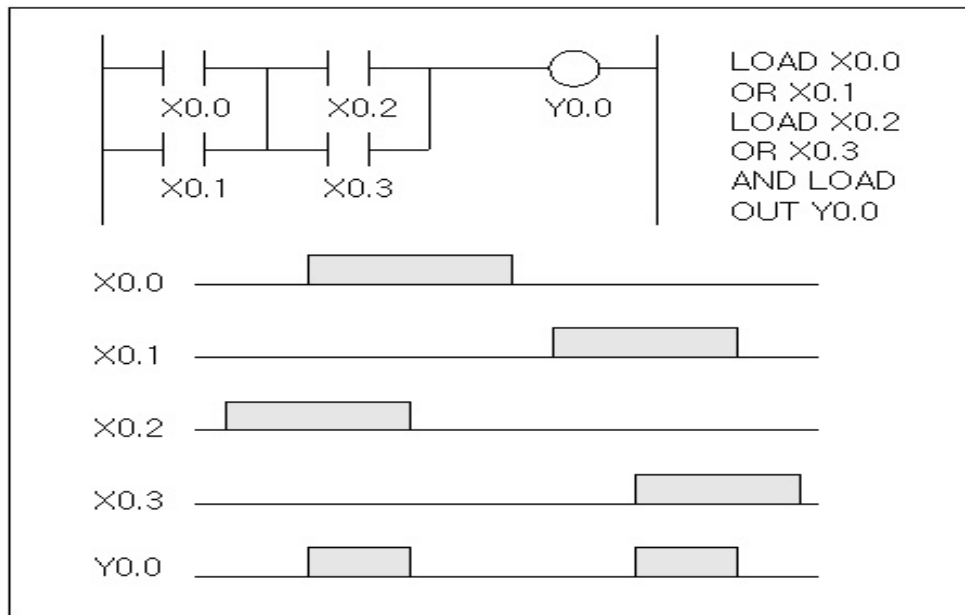
(6) OR NOT

- OR NOT



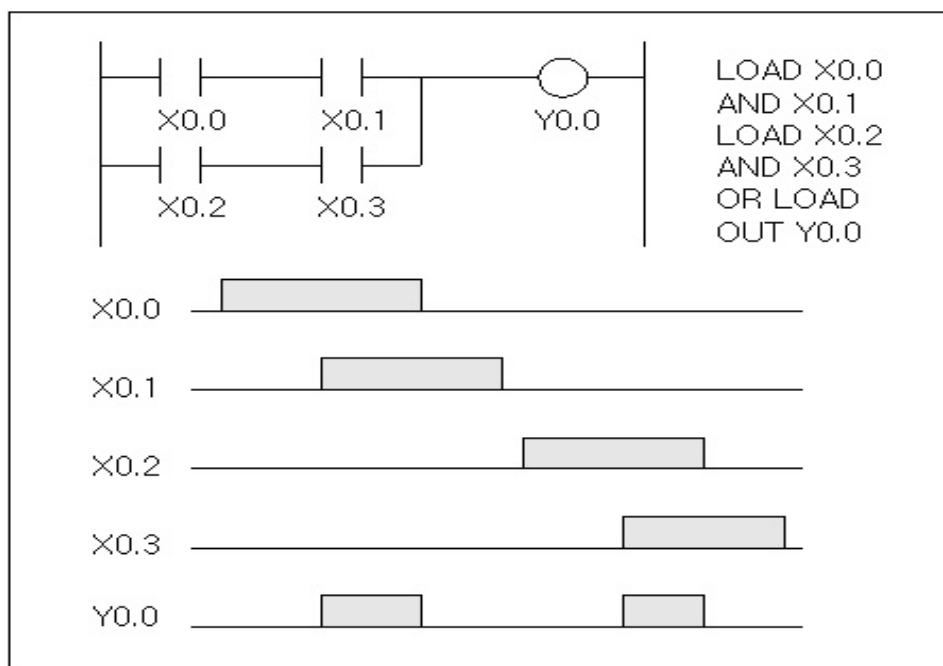
(7) AND LOAD

- Block AND
- Block LOAD LOAD NOT



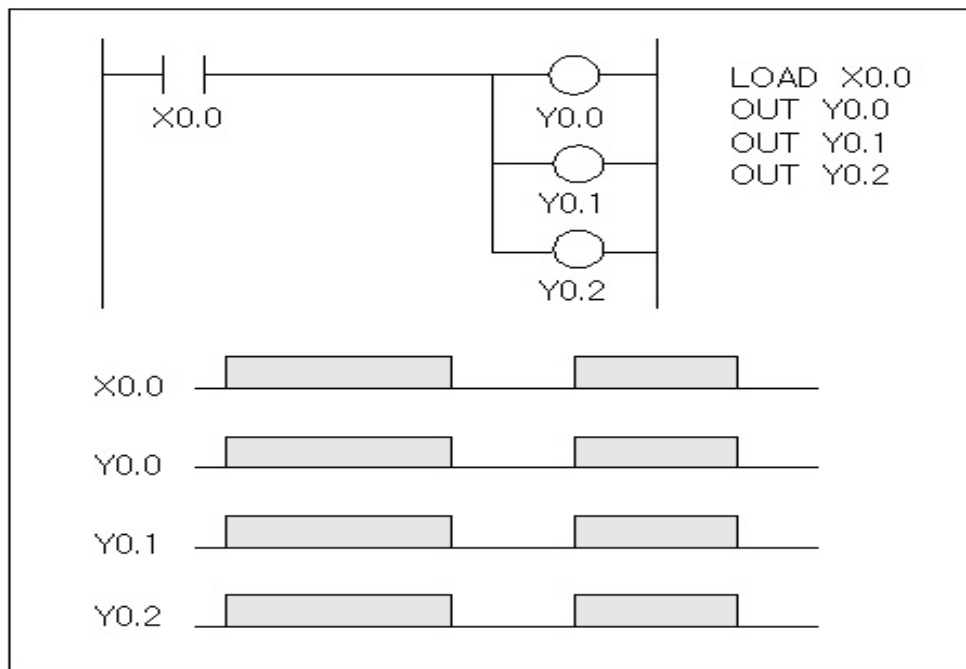
(8) OR LOAD

- Block OR
- LOAD LOAD NOT



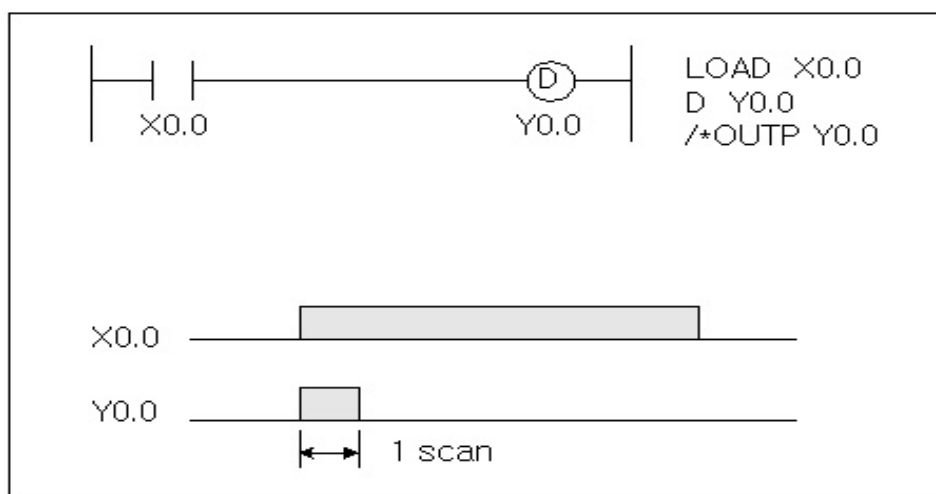
(9) OUT

- Relay, , Timer, Counter OUT 가 .



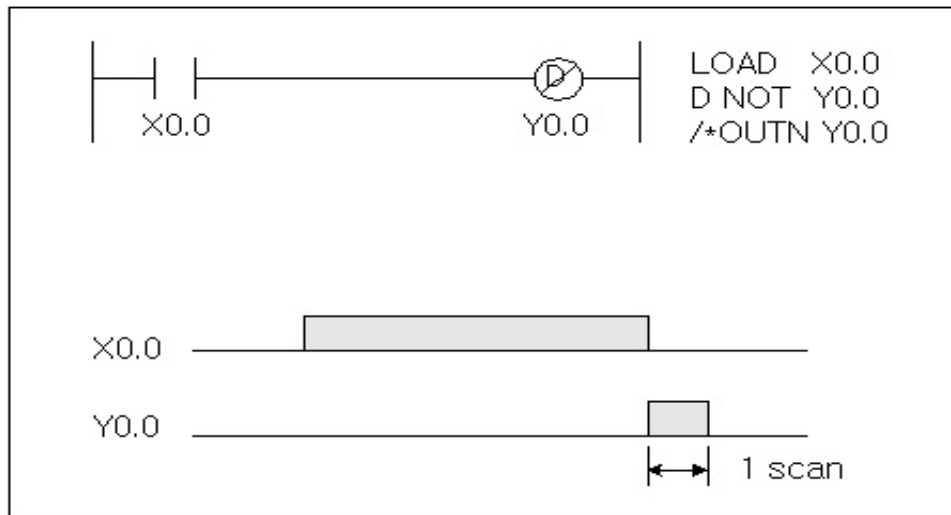
(10) D

- | | | | | | |
|-----|--------|-----|-------------|----|---|
| - | | | 1 Scan Time | ON | . |
| - D | OFF | ON | 1 Scan Time | ON | . |
| | : OUTP | - D | OUTP | . | |



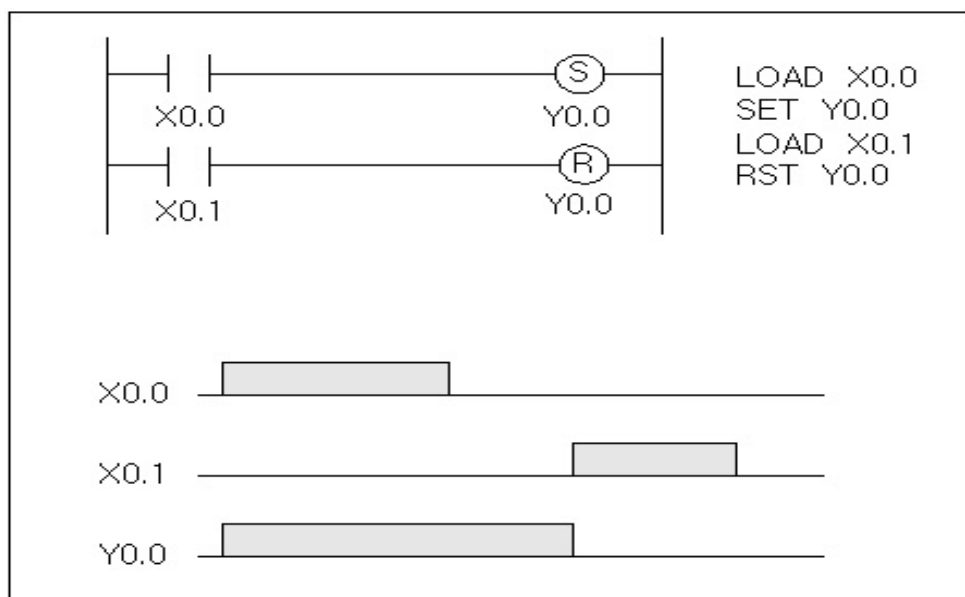
(11) D NOT

- | | | | | |
|---------|--------|-------------|-------------|----|
| - | | 1 Scan Time | ON | . |
| - D NOT | ON | OFF | 1 Scan Time | ON |
| | : OUTN | - D NOT | OUTN | . |



(12) SET, RST

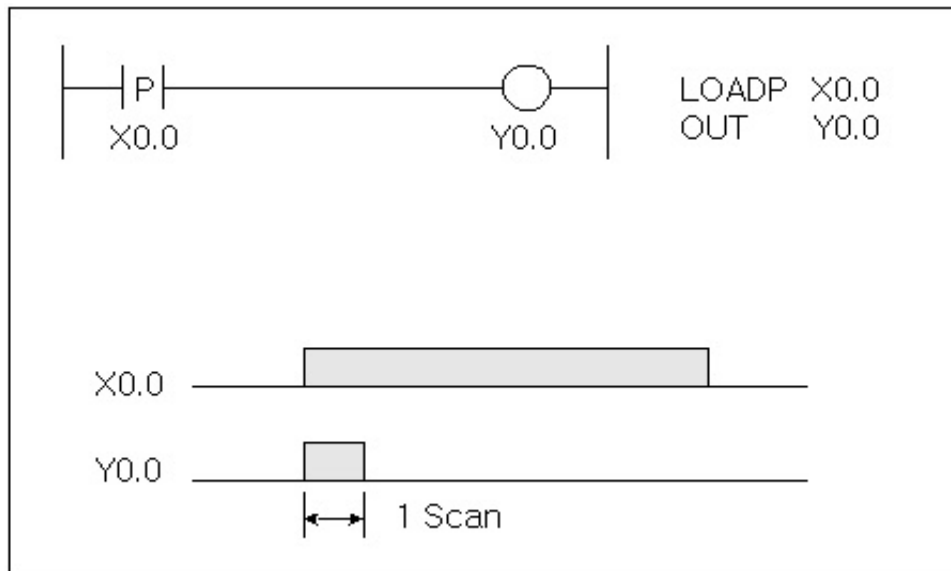
- | | | | | |
|-------|-----|-----|--------------|-----|
| - SET | ON | ON | Self-holding | OFF |
| - RST | ON | OFF | Self-holding | OFF |
| | OFF | | | |





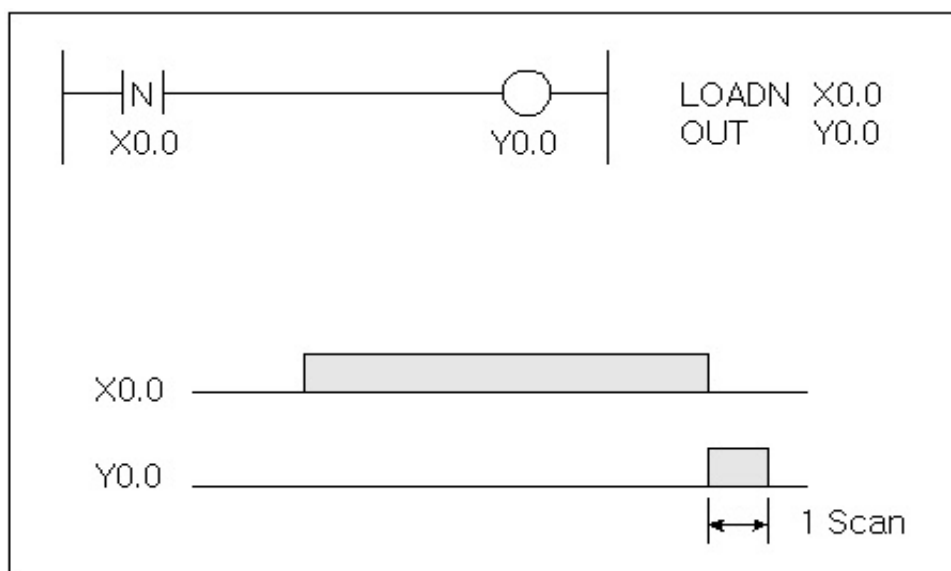
(15) LOADP

LOADP : OFF ON 1 Scan Time ON
 LOAD .



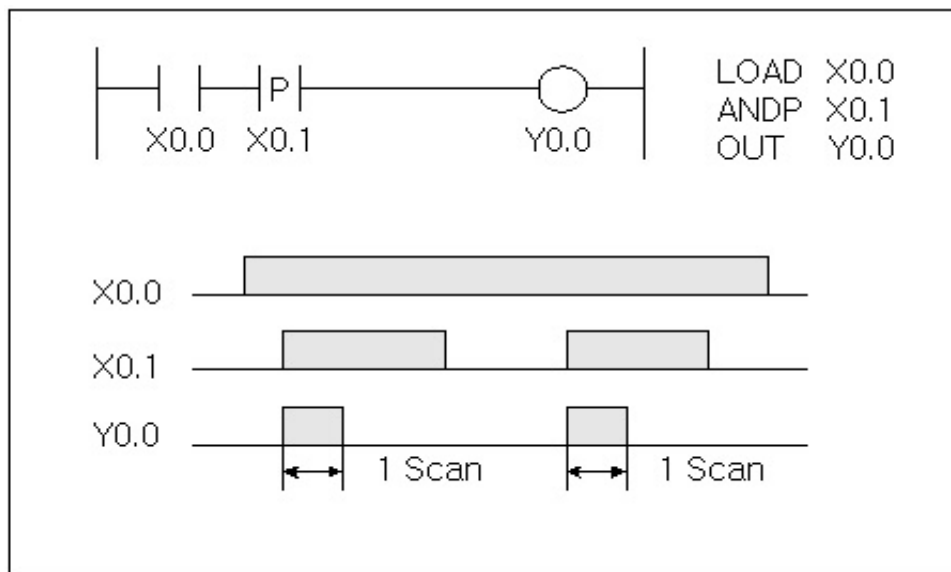
(16) LOADN

LOADN : ON OFF 1 Scan Time ON
 LOAD .



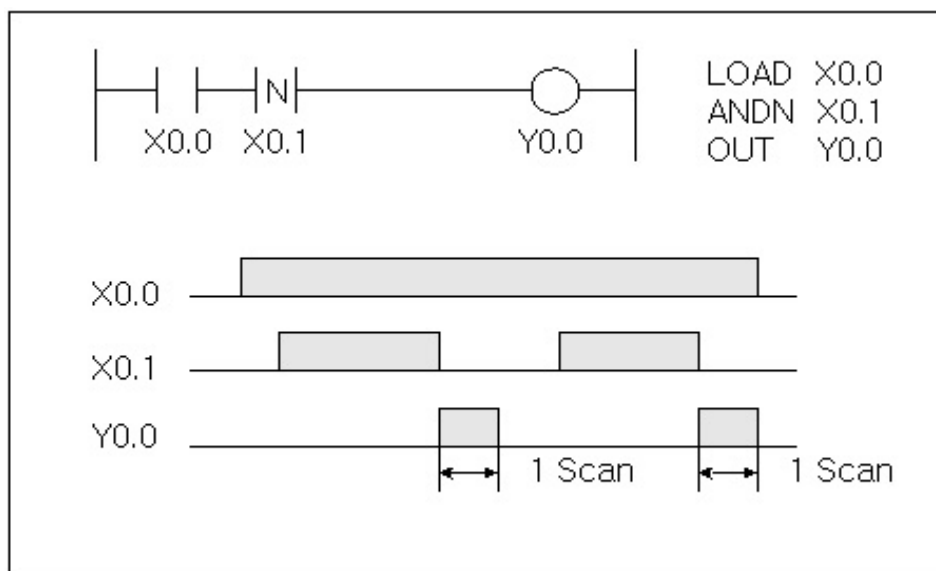
(17) ANDP

ANDP : OFF ON 1 Scan Time ON AND .



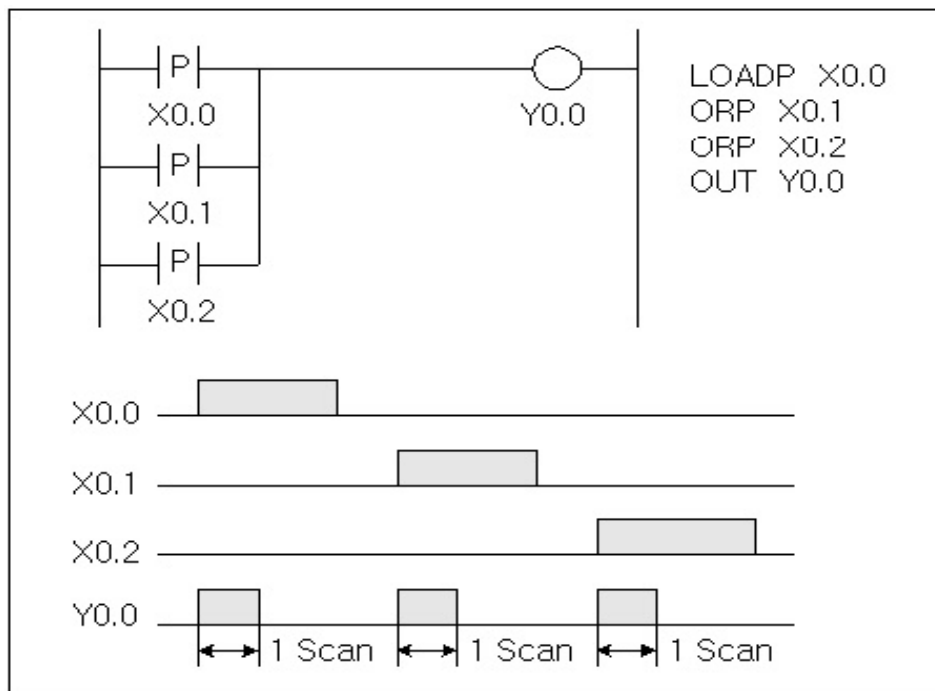
(18) ANDN

ANDN : ON OFF 1 Scan Time ON AND .

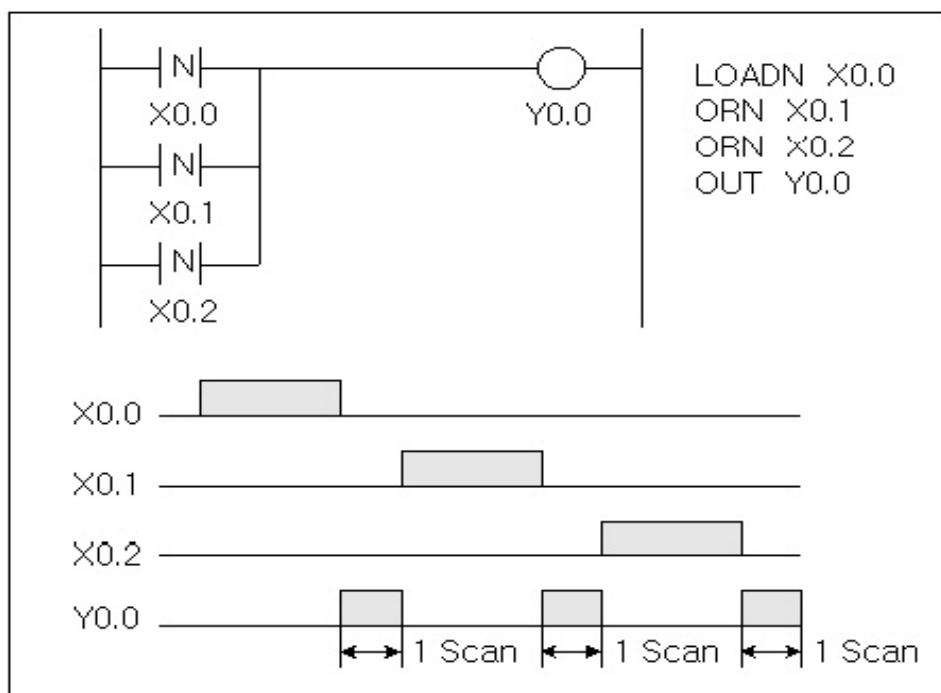


(19) ORP

- ORP OFF ON 1 Scan Time ON OR .

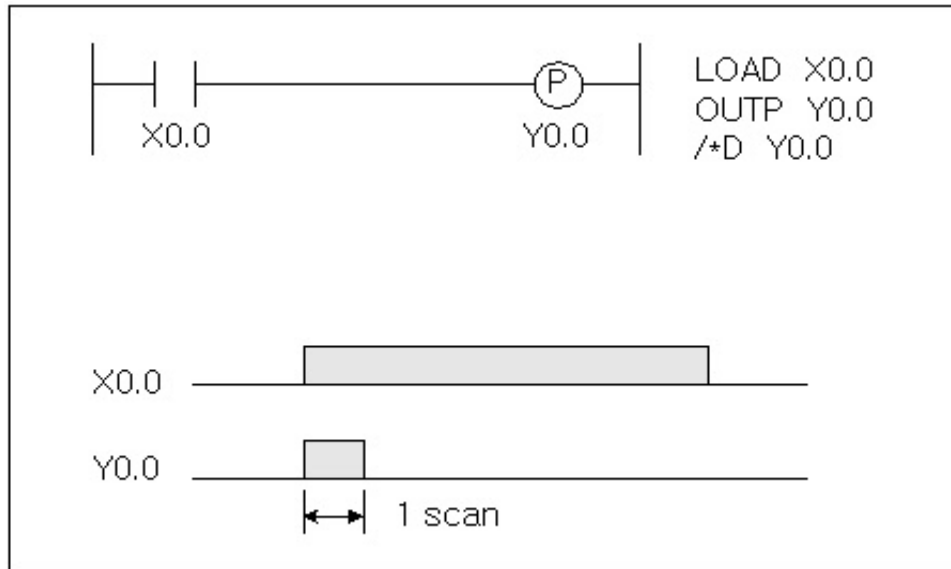
**(20) ORN**

ORN ON OFF 1 Scan Time ON OR .



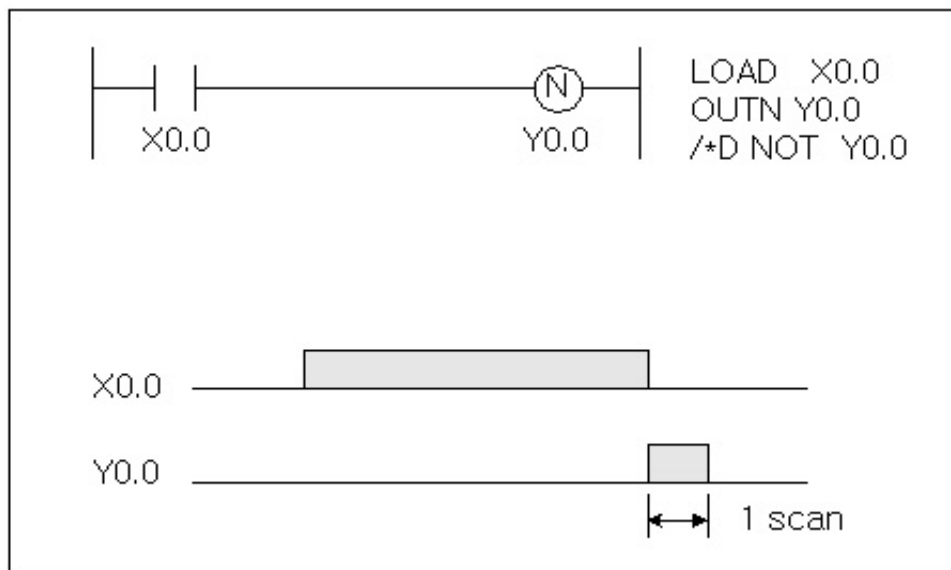
(21) OUTP

- 1 Scan Time ON
 - OUTP : OFF ON 1 ScanTime ON
 : D - OUTP D



(22) OUTN

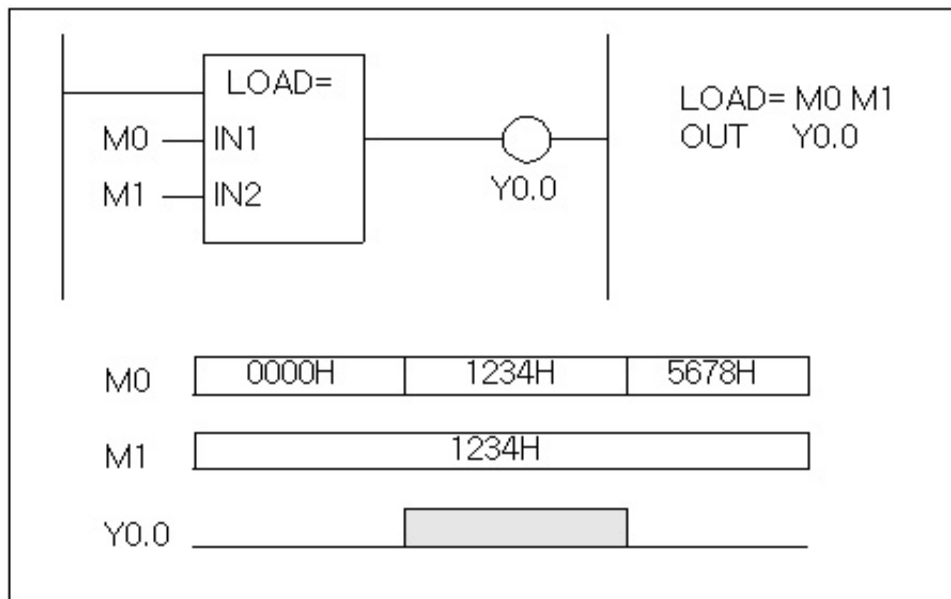
- 1 Scan Time ON
 - OUTN : D NOT ON OFF 1 Scan Time ON
 : D NOT - OUTN D NOT



2) (Bit)

(1) LOAD=, DLOAD=

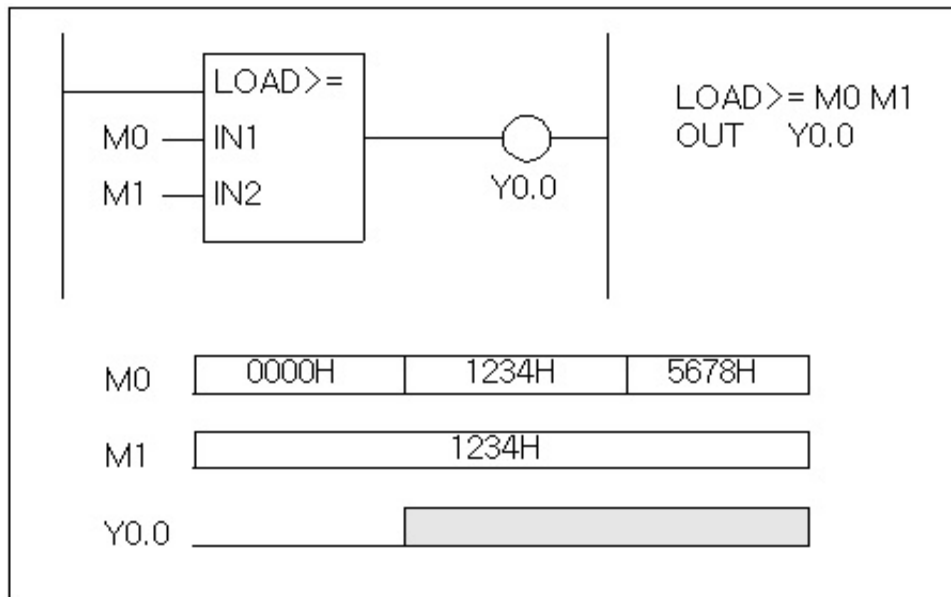
- INT(16Bits) DINT(32Bits) ' (=) LOAD
- LOAD= : INT(16Bits) INT(16Bits)
- INT(16Bits) ' (=) (Bit
- Result)
- DLOAD= : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (=)
- (Bit Result)



- LOAD= M0 1234
- LOAD= 1234 M0
- LOAD= M0 -1234
- LOAD= -1234 M0
- LOAD= M0 1234H
- LOAD= 1234H M0
- LOAD= M0 M1
- DLOAD= M0 1234567890
- DLOAD= 1234567890 M0
- DLOAD= M0 -1234567890
- DLOAD= -1234567890 M0
- DLOAD= M0 89ABCDEFH
- DLOAD= 89ABCDEFH M0
- DLOAD= M0 M2

(2) **LOAD>=, DLOAD>=**

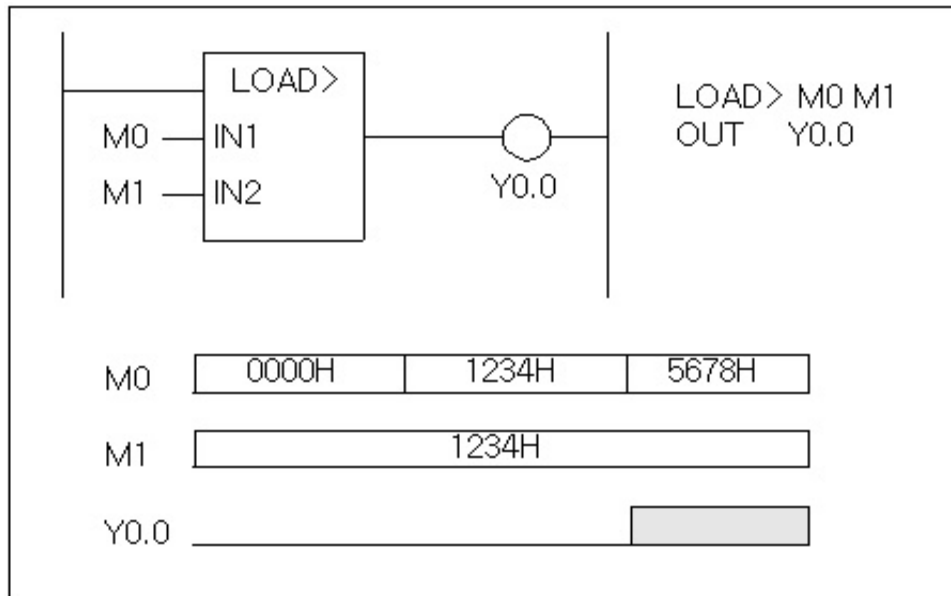
- INT(16Bits) DINT(32Bits) '(>=) LOAD
- LOAD>= : INT(16Bits) INT(16Bits)
- INT(16Bits) '(>=)
- (Bit Result)
- DLOAD>= : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(>=)
- (Bit Result)



- LOAD>= M0 1234
- LOAD>= 1234 M0
- LOAD>= M0 -1234
- LOAD>= -1234 M0
- LOAD>= M0 1234H
- LOAD>= 1234H M0
- LOAD>= M0 M1
- DLOAD>= M0 1234567890
- DLOAD>= 1234567890 M0
- DLOAD>= M0 -1234567890
- DLOAD>= -1234567890 M0
- DLOAD>= M0 89ABCDEFH
- DLOAD>= 89ABCDEFH M0
- DLOAD>= M0 M2

(3) LOAD>, DLOAD>

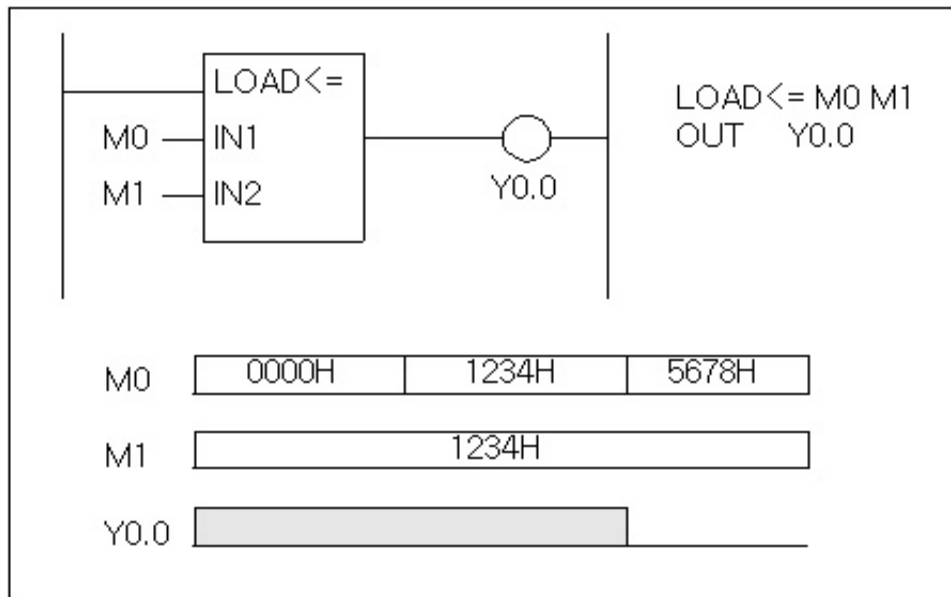
- INT(16Bits) DINT(32Bits) '(>) LOAD .
- LOAD> : INT(16Bits) INT(16Bits)
- INT(16Bits) '(>) (Bit
- Result)
- DLOAD> : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(>) .
- (Bit Result)



- LOAD> M0 1234
- LOAD> 1234 M0
- LOAD> M0 -1234
- LOAD> -1234 M0
- LOAD> M0 1234H
- LOAD> 1234H M0
- LOAD> M0 M1
- DLOAD> M0 1234567890
- DLOAD> 1234567890 M0
- DLOAD> M0 -1234567890
- DLOAD> -1234567890 M0
- DLOAD> M0 89ABCDEFH
- DLOAD> 89ABCDEFH M0
- DLOAD> M0 M2

(4) **LOAD<=, DLOAD<=**

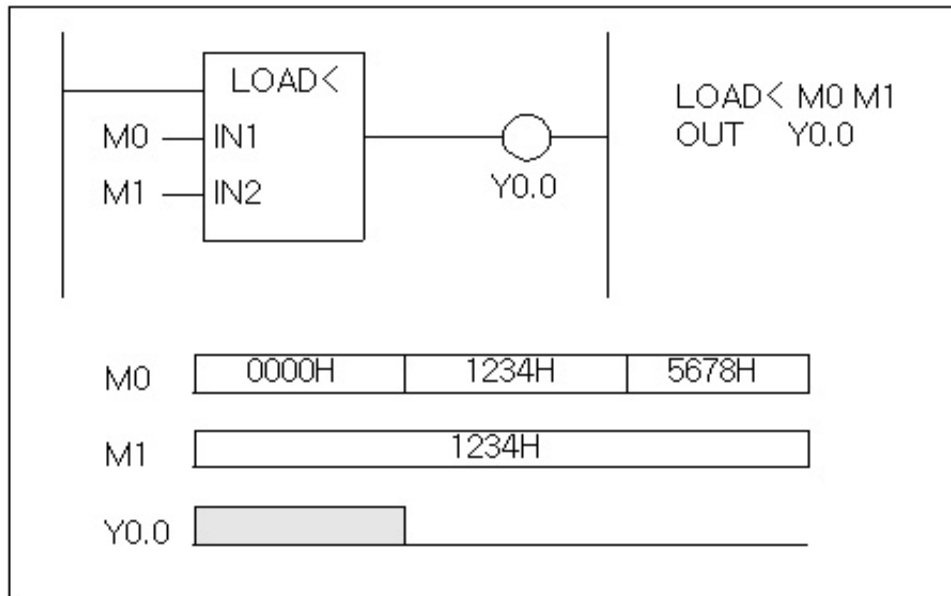
- INT(16Bits) DINT(32Bits) '(<=) LOAD
- LOAD<= : INT(16Bits) INT(16Bits)
- INT(16Bits) '(<=)
- (Bit Result)
- DLOAD<= : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(<=)
- (Bit Result)



- OAD<= M0 1234
- LOAD<= 1234 M0
- LOAD<= M0 -1234
- LOAD<= -1234 M0
- LOAD<= M0 1234H
- LOAD<= 1234H M0
- LOAD<= M0 M1
- DLOAD<= M0 1234567890
- DLOAD<= 1234567890 M0
- DLOAD<= M0 -1234567890
- DLOAD<= -1234567890 M0
- DLOAD<= M0 89ABCDEFH
- DLOAD<= 89ABCDEFH M0
- DLOAD<= M0 M2

(5) LOAD<, DLOAD<,

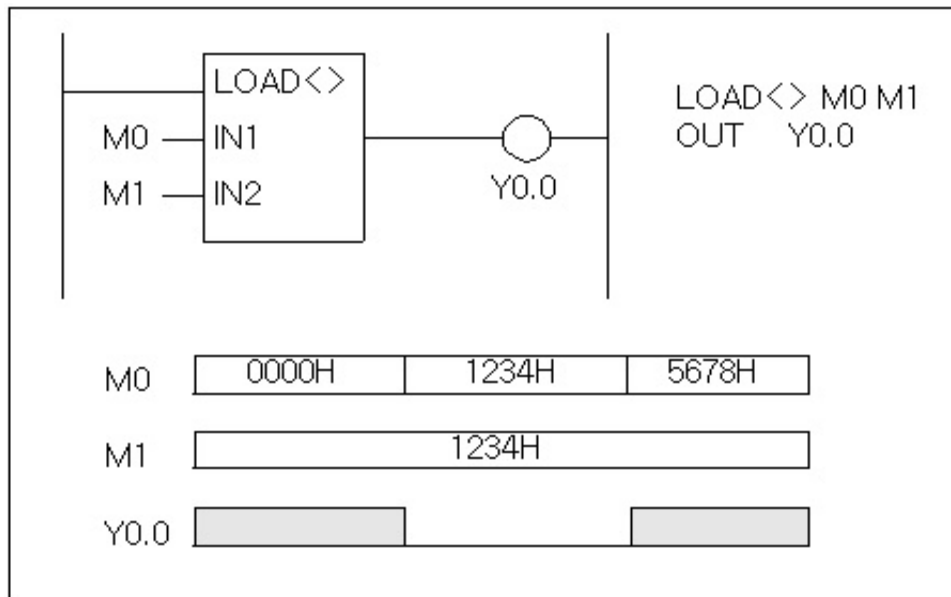
- INT(16Bits) DINT(32Bits) ' (<) LOAD .
- LOAD< : INT(16Bits) INT(16Bits)
- INT(16Bits) ' (<) (Bit
- Result)
- DLOAD< : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (<) .
- (Bit Result)



- LOAD< M0 1234
- LOAD< 1234 M0
- LOAD< M0 -1234
- LOAD< -1234 M0
- LOAD< M0 1234H
- LOAD< 1234H M0
- LOAD< M0 M1
- DLOAD< M0 1234567890
- DLOAD< 1234567890 M0
- DLOAD< M0 -1234567890
- DLOAD< -1234567890 M0
- DLOAD< M0 89ABCDEFH
- DLOAD< 89ABCDEFH M0
- DLOAD< M0 M2

(6) LOAD<>, DLOAD<>

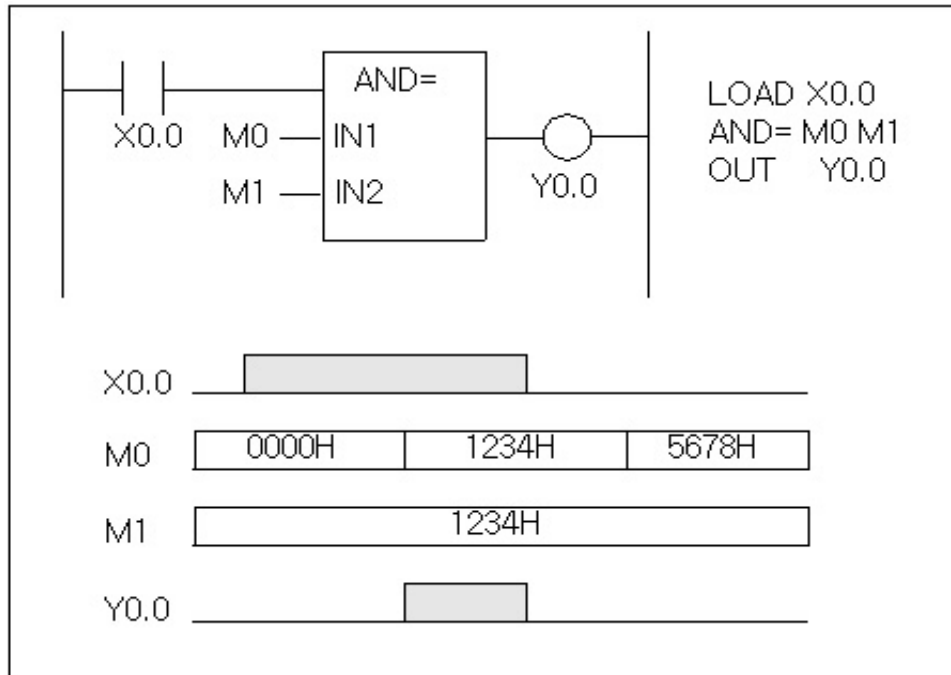
- INT(16Bits) DINT(32Bits) '(<>) LOAD
- LOAD<> : INT(16Bits) INT(16Bits)
- INT(16Bits) '(<>)
- (Bit Result)
- DLOAD<> : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(<>)
- (Bit Result)



- LOAD<> M0 1234
- LOAD<> 1234 M0
- LOAD<> M0 -1234
- LOAD<> -1234 M0
- LOAD<> M0 1234H
- LOAD<>1234H M0
- LOAD<> M0 M1
- DLOAD<> M0 1234567890
- DLOAD<> 1234567890 M0
- DLOAD<> M0 -1234567890
- DLOAD<> -1234567890 M0
- DLOAD<> M0 89ABCDEFH
- DLOAD<> 89ABCDEFH M0
- DLOAD<> M0 M2

(7) AND=, DAND=

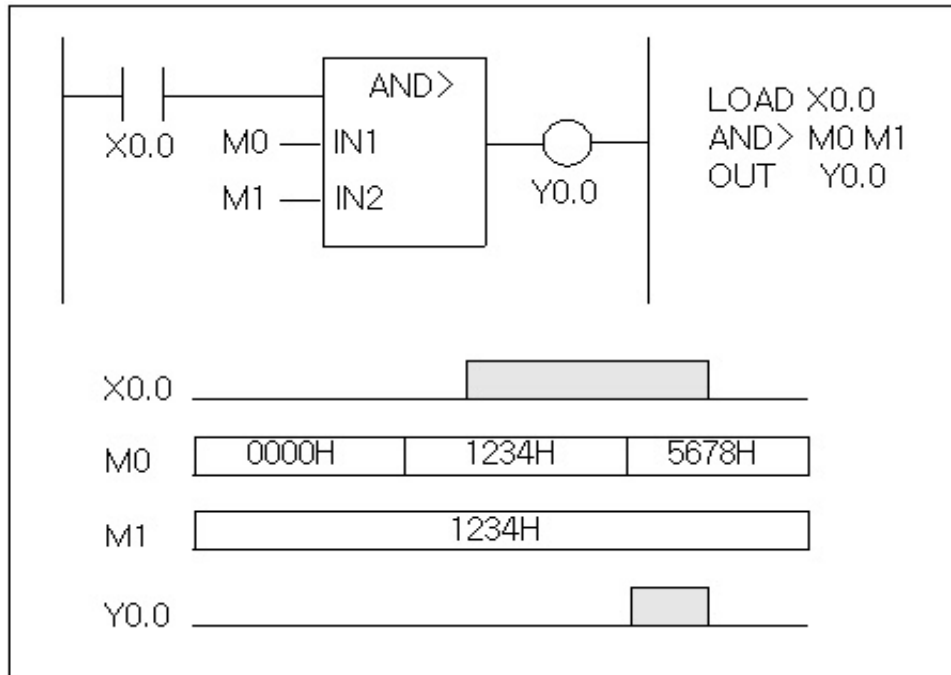
- INT(16Bits) DINT(32Bits) ' (=) AND .
- AND= : INT(16Bits) INT(16Bits)
- INT(16Bits) ' (=) (Bit
- Result) .
- DAND= : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (=) (Bit
- Result) .



- AND= M0 1234
- AND= 1234 M0
- AND= M0 -1234
- AND= -1234 M0
- AND= M0 1234H
- AND= 1234H M0
- AND= M0 M1
- DAND= M0 1234567890
- DAND= 1234567890 M0
- DAND= M0 -1234567890
- DAND= -1234567890 M0
- DAND= 89ABCDEFH M0
- DAND= M0 M2

(9) AND>, DAND>

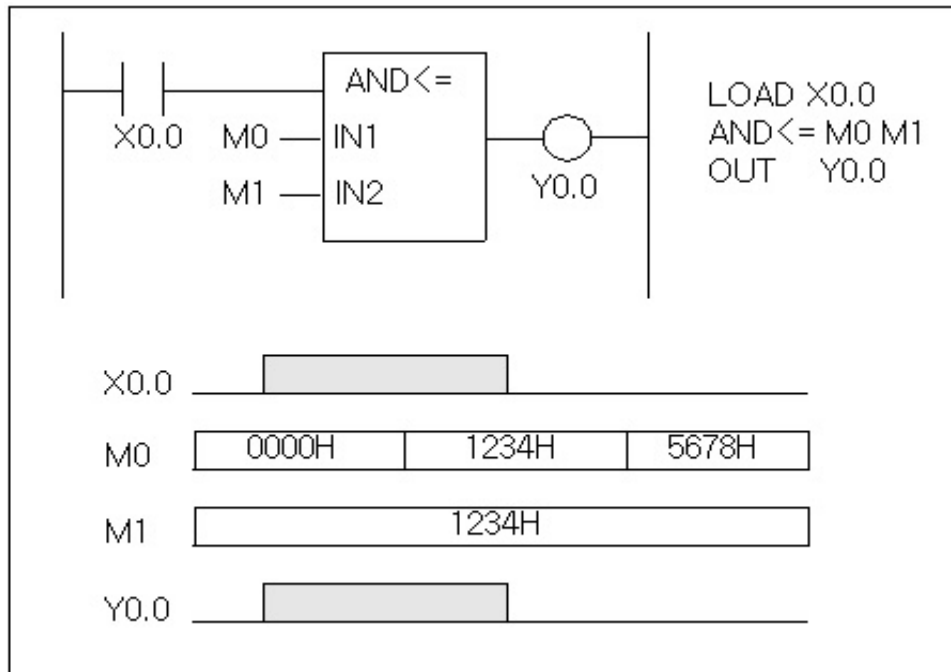
- INT(16Bits) DINT(32Bits) '(>) AND .
- AND> : INT(16Bits) INT(16Bits)
- INT(16Bits) '(>) (Bit
- Result)
- DAND> : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(>)
- (Bit Result)



- AND> M0 1234
- AND> 1234 M0
- AND> M0 -1234
- AND> -1234 M0
- AND> M0 1234H
- AND> 1234H M0
- AND> M0 M1
- DAND> M0 1234567890
- DAND> 1234567890 M0
- DAND> M0 -1234567890
- DAND> -1234567890 M0
- DAND> 89ABCDEFH M0
- DAND> M0 M2

(10) AND<=, DAND<=

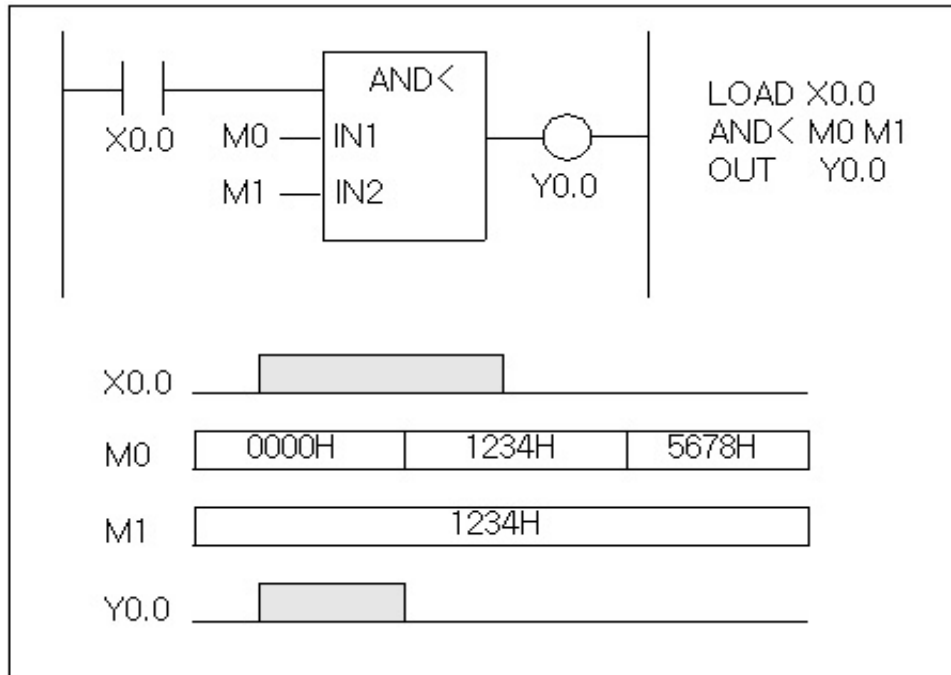
- INT(16Bits) DINT(32Bits) '(<=) AND
- AND<= : INT(16Bits) INT(16Bits)
INT(16Bits) '(<=)
(Bit Result)
- DAND<= : DINT(32Bits) DINT(32Bits)
DINT(32Bits) '(<=)
(Bit Result)



- AND<= M0 1234
- AND<= 1234 M0
- AND<= M0 -1234
- AND<= -1234 M0
- AND<= M0 1234H
- AND<= 1234H M0
- AND<= M0 M1
- DAND<= M0 1234567890
- DAND<= 1234567890 M0
- DAND<= M0 -1234567890
- DAND<= -1234567890 M0
- DAND<= 89ABCDEFH M0
- DAND<= M0 M2

(11) AND<, DAND<,

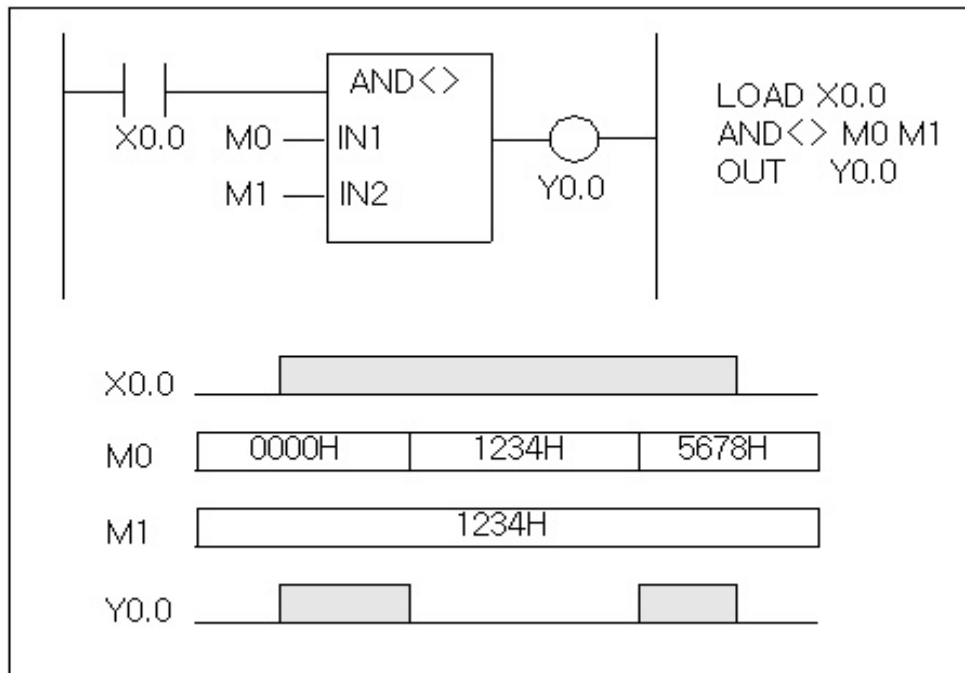
- INT(16Bits) DINT(32Bits) ' (<) AND .
- AND< : INT(16Bits) INT(16Bits)
- INT(16Bits) ' (<) (Bit
- Result)
- DAND< : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (<)
- (Bit Result)



- AND< M0 1234
- AND< 1234 M0
- AND< M0 -1234
- AND< -1234 M0
- AND< M0 1234H
- AND< 1234H M0
- AND< M0 M1
- DAND< M0 1234567890
- DAND< 1234567890 M0
- DAND< M0 -1234567890
- DAND< -1234567890 M0
- DAND< 89ABCDEFH M0
- DAND< M0 M2

(12) AND<>, DAND<>

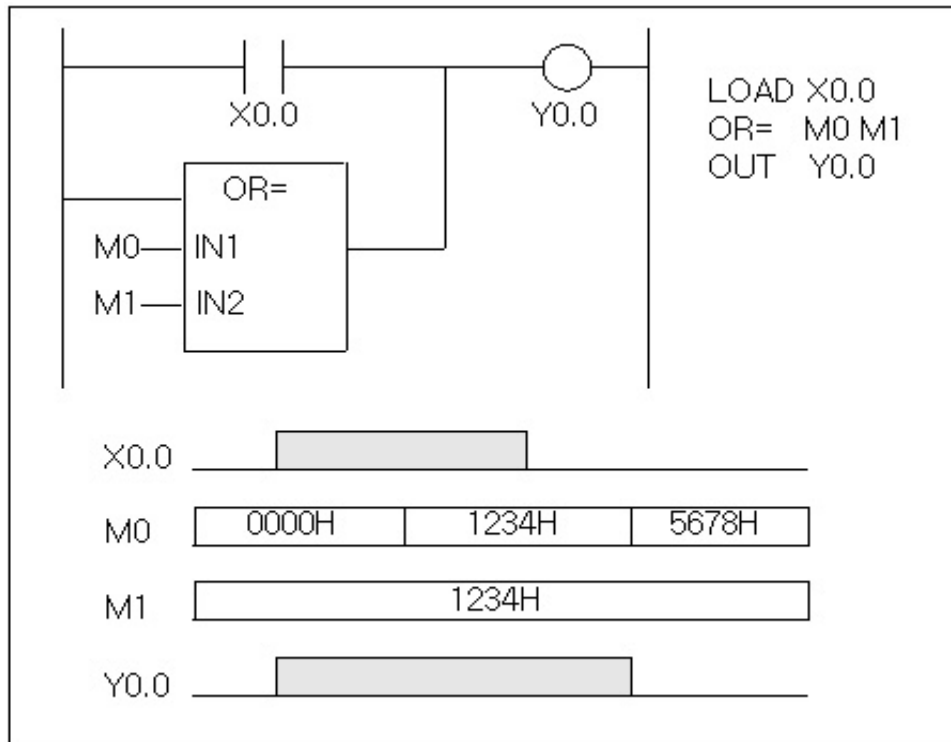
- INT(16Bits) DINT(32Bits) '(<>) AND
- AND<> : INT(16Bits) INT(16Bits)
- INT(16Bits) '(<>)
- (Bit Result)
- DAND<> : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(<>)
- (Bit Result)



- AND<> M0 1234
- AND<> 1234 M0
- AND<> M0 -1234
- AND<> -1234 M0
- AND<> M0 1234H
- AND<> 1234H M0
- AND<> M0 M1
- DAND<> M0 1234567890
- DAND<> 1234567890 M0
- DAND<> M0 -1234567890
- DAND<> -1234567890 M0
- DAND<> 89ABCDEFH M0
- DAND<> M0 M2

(13) OR=, DOR=

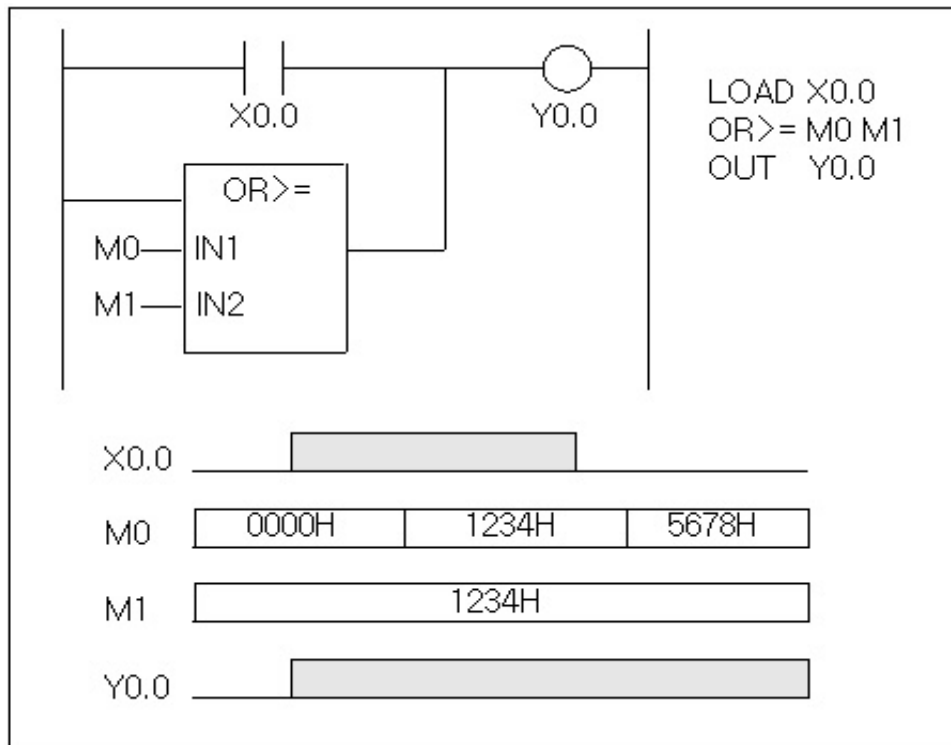
- INT(16Bits) DINT(32Bits) ' (=) OR .
- OR= : INT(16Bits) INT(16Bits)
- INT(16Bits) ' (=) (Bit
- Result)
- DOR= : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (=) (Bit
- Result)



- OR= M0 1234
- OR= 1234 M0
- OR= M0 -1234
- OR= -1234 M0
- OR= M0 1234H
- OR= 1234H M0
- OR= M0 M1
- DOR= M0 1234567890
- DOR= 1234567890 M0
- DOR= M0 -1234567890
- DOR= -1234567890 M0
- DOR= M0 89ABCDEFH
- DOR= 89ABCDEFH M0
- DOR= M0 M2

(14) OR>=, DOR>=

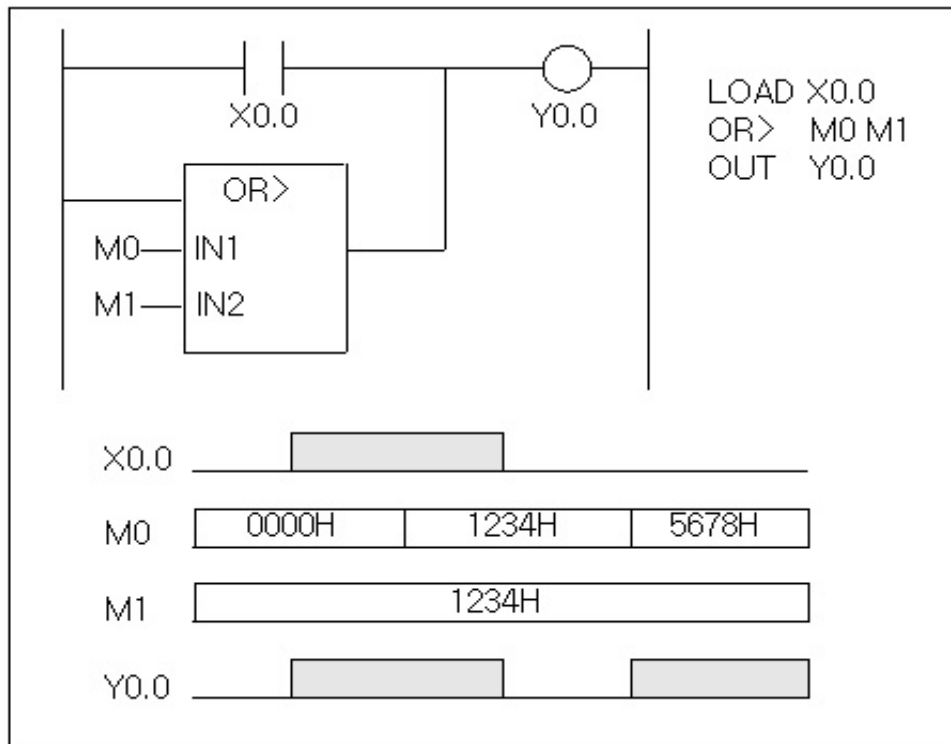
- INT(16Bits) DINT(32Bits) '(>=) OR
- OR>= : INT(16Bits) INT(16Bits)
 INT(16Bits) '(>=)
 (Bit Result)
- DOR>= : DINT(32Bits) DINT(32Bits)
 DINT(32Bits) '(>=)
 (Bit Result)



- OR>= M0 1234
- OR>= 1234 M0
- OR>= M0 -1234
- OR>= -1234 M0
- OR>= M0 1234H
- OR>= 1234H M0
- OR>= M0 M1
- DOR>= M0 1234567890
- DOR>= 1234567890 M0
- DOR>= M0 -1234567890
- DOR>= -1234567890 M0
- DOR>= M0 89ABCDEFH
- DOR>= 89ABCDEFH M0
- DOR>= M0 M2

(15) OR>, DOR>

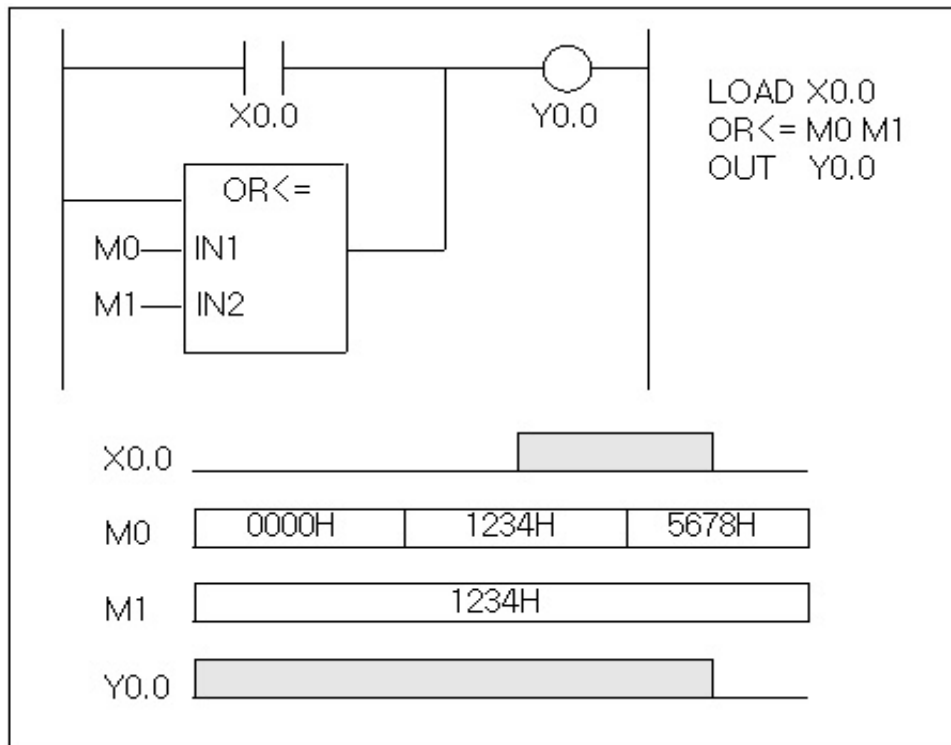
- INT(16Bits) DINT(32Bits) '(>) OR .
- OR> : INT(16Bits) INT(16Bits)
- INT(16Bits) '(>) (Bit
- Result) .
- DOR> : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(>) (Bit
- Result) .



- OR> M0 1234
- OR> 1234 M0
- OR> M0 -1234
- OR> -1234 M0
- OR> M0 1234H
- OR> 1234H M0
- OR> M0 M1
- DOR> M0 1234567890
- DOR> 1234567890 M0
- DOR> M0 -1234567890
- DOR> -1234567890 M0
- DOR> M0 89ABCDEFH
- DOR> 89ABCDEFH M0
- DOR> M0 M2

(16) OR<=, DOR<=

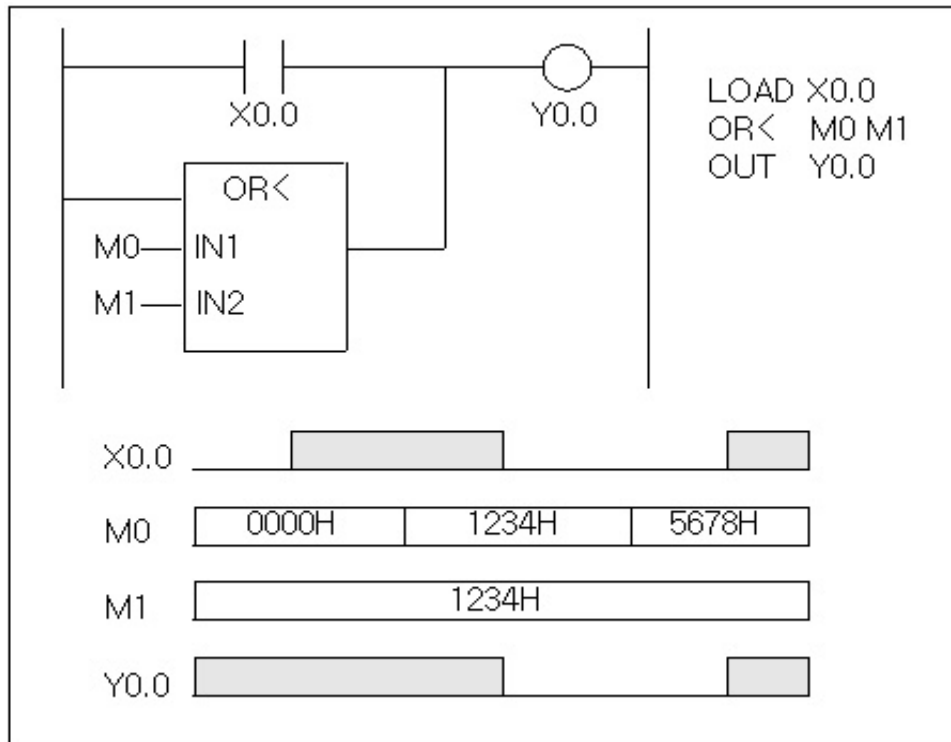
- INT(16Bits) DINT(32Bits) '(<=) OR
- OR<= : INT(16Bits) INT(16Bits)
- INT(16Bits) '(<=)
- (Bit Result)
- DOR<= : DINT(32bit) DINT(32bit)
- DINT(32bit) '(<=)
- (Bit Result)



- OR<= M0 1234
- OR<= 1234 M0
- OR<= M0 -1234
- OR<= -1234 M0
- OR<= M0 1234H
- OR<= 1234H M0
- OR<= M0 M1
- DOR<= M0 1234567890
- DOR<= 1234567890 M0
- DOR<= M0 -1234567890
- DOR<= -1234567890 M0
- DOR<= M0 89ABCDEFH
- DOR<= 89ABCDEFH M0
- DOR<= M0 M2

(17) OR<, DOR<

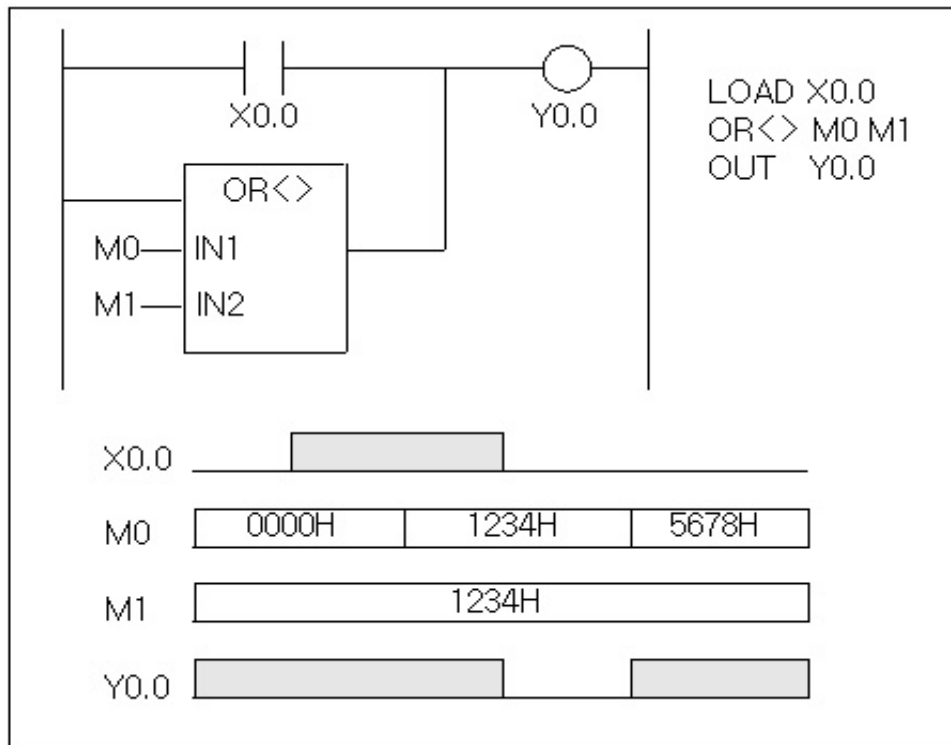
- INT(16Bits) DINT(32Bits) ' (<) OR .
- OR< : INT(16Bits) INT(16Bits)
- INT(16Bits) ' (<) (Bit
- Result)
- DOR< : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (<) (Bit
- Result)



- OR< M0 1234
- OR< 1234 M0
- OR< M0 -1234
- OR< -1234 M0
- OR< M0 1234H
- OR< 1234H M0
- OR< M0 M1
- DOR< M0 1234567890
- DOR< 1234567890 M0
- DOR< M0 -1234567890
- DOR< -1234567890 M0
- DOR< M0 89ABCDEFH
- DOR< 89ABCDEFH M0
- DOR< M0 M2

(18) OR<>, DOR<>

- INT(16Bits) DINT(32Bits) '(<>) OR
- OR<> : INT(16Bits) INT(16Bits)
 INT(16Bits) '(<>)
 (Bit Result)
- DOR<> : DINT(32Bits) DINT(32Bits)
 DINT(32Bits) '(<>)
 (Bit Result)



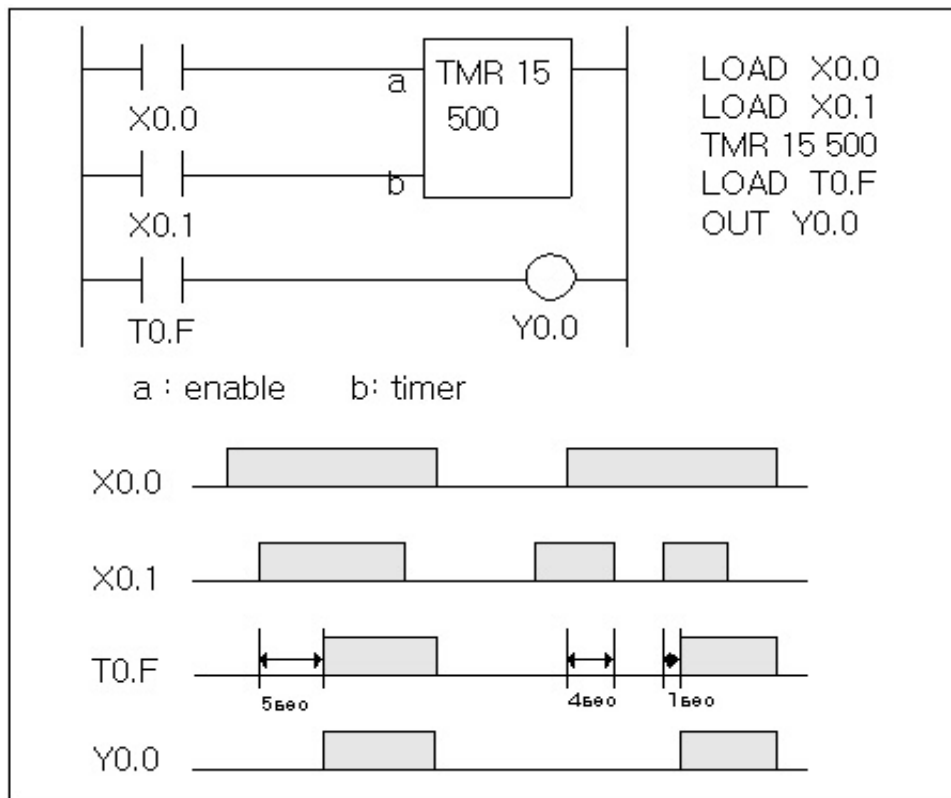
- OR<> M0 1234
- OR<> 1234 M0
- OR<> M0 -1234
- OR<> -1234 M0
- OR<> M0 1234H
- OR<> 1234H M0
- OR<> M0 M1
- DOR<> M0 1234567890
- DOR<> 1234567890 M0
- DOR<> M0 -1234567890
- DOR<> -1234567890 M0
- DOR<> M0 89ABCDEFH
- DOR<> 89ABCDEFH M0
- DOR<> M0 M2

3) ,

(1) TMR

- TMR 가 Timer 가
- 0~31 (T0.0~T1.F) . (: 10 T0.A)
- a 가 ON b 가 ON 가 , b 가 OFF 가
- a 가 OFF Timer '0' Reset .
- Timer Pulse 0.01 (10ms) .

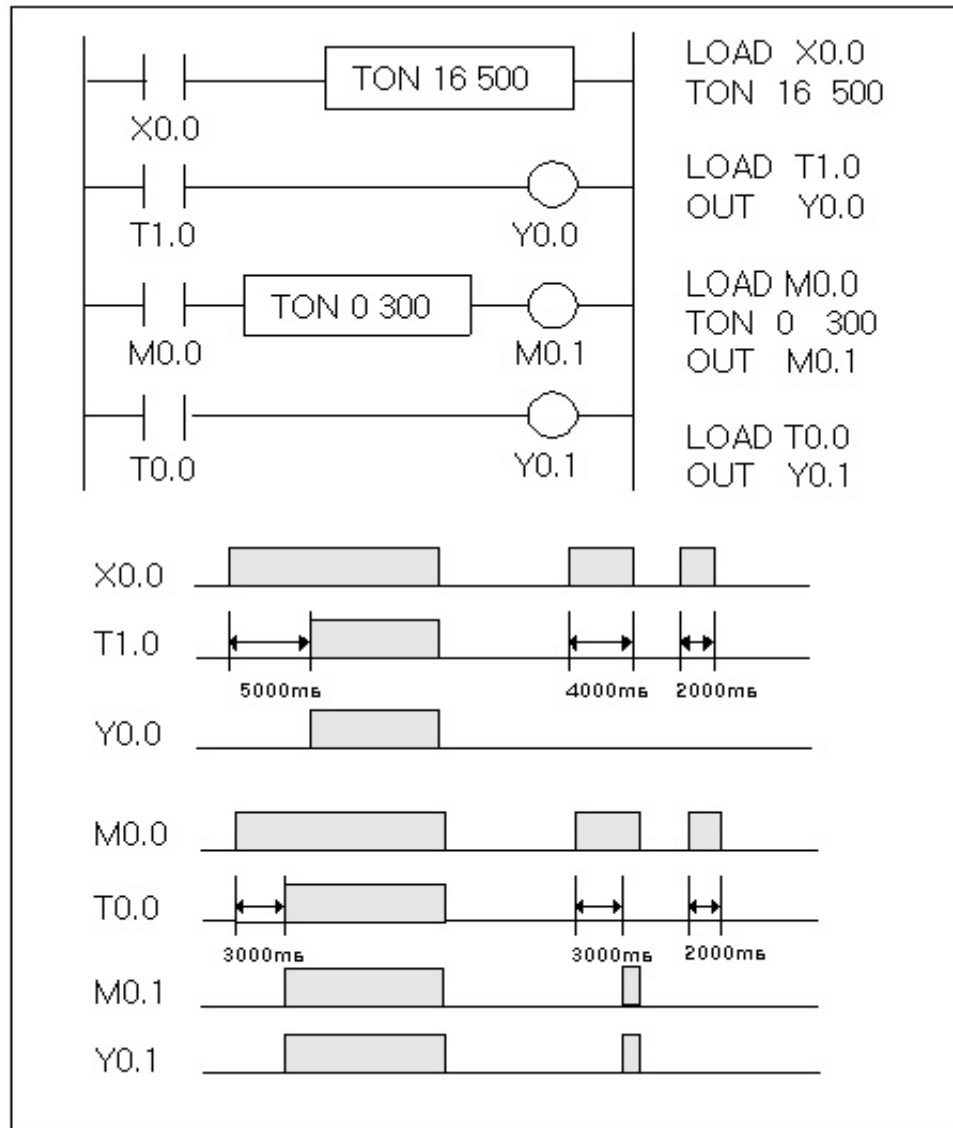
: TMR 가 .



(2) TON

- TON On-Delay Timer .
- 0~31(T0.0~T1.F) . (: 26 T1.A)
- ON 가 가 ON ,
- OFF 가 '0' OFF .
- Timer Pulse 0.01 (10ms) .

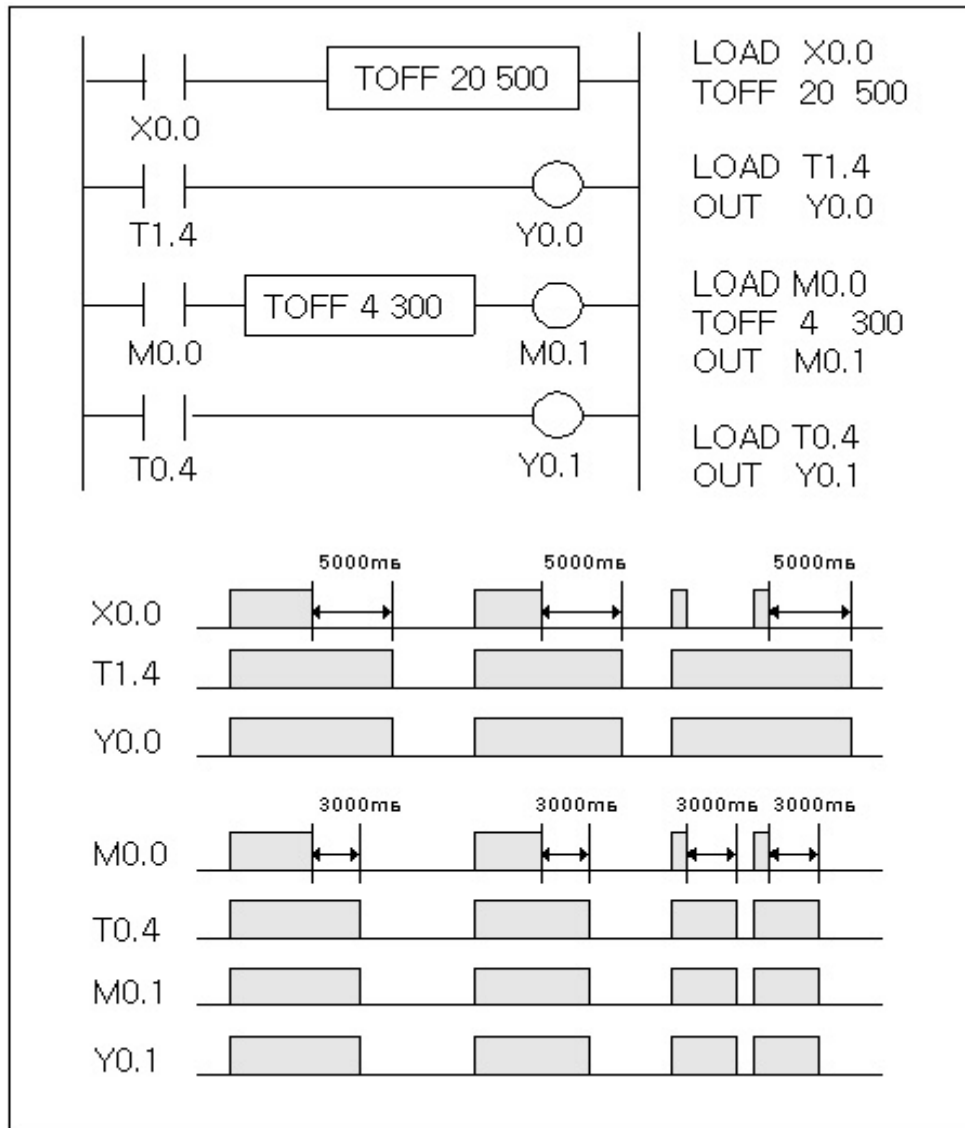
: TON 가 .



(3) TOFF

- TOFF Off-Delay Timer .
- 0~31(T0.0 ~ T1.F) . (: 10 T0.A)
- ON 가 ON , OFF
- Timer Pulse 0.01 (10ms) .

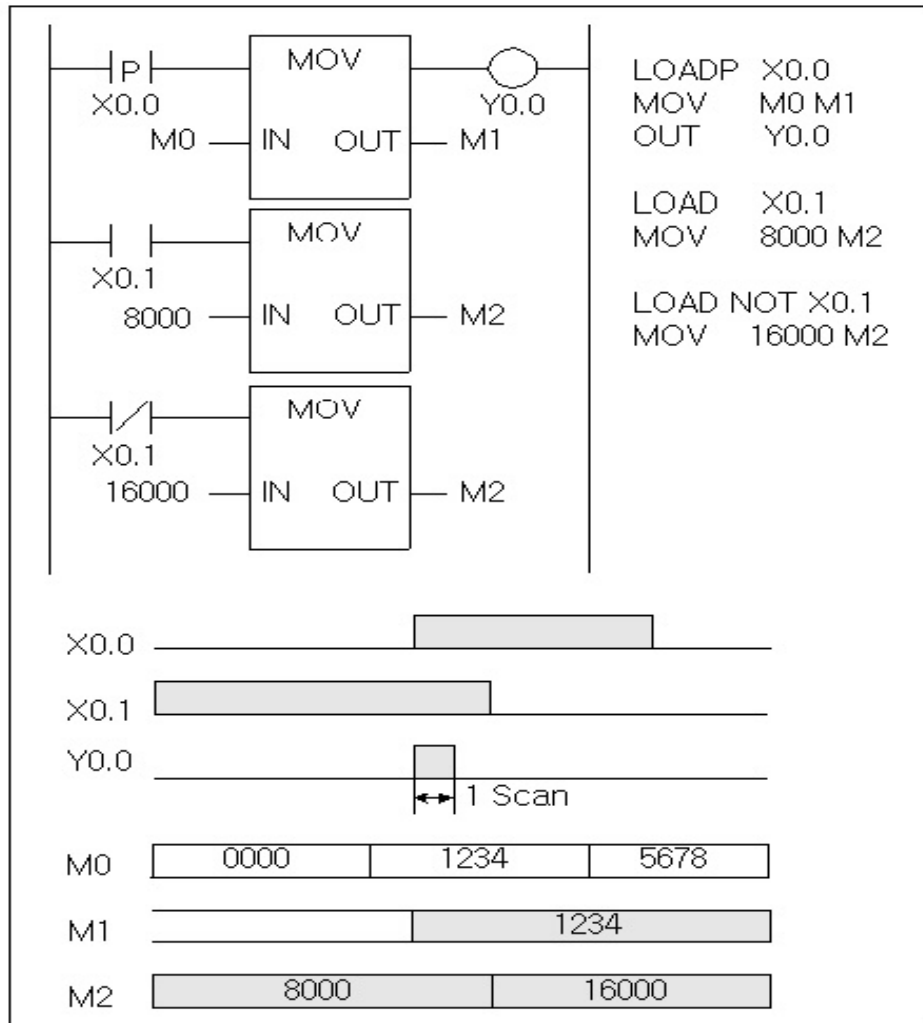
: TOFF 가 .



4) (16Bits, 32Bits)

(1) MOV, DMOV

- WORD(16Bits)/INT(16Bits) DWORD(32Bits)/DINT(32Bits)
- MOV : WORD (16Bits)/INT(16Bits) WORD(16Bits)
 /INT(16Bits)
- DMOV : DWORD(32Bits)/DINT(32Bits) DWORD(32Bits)/DINT
 (32Bits)

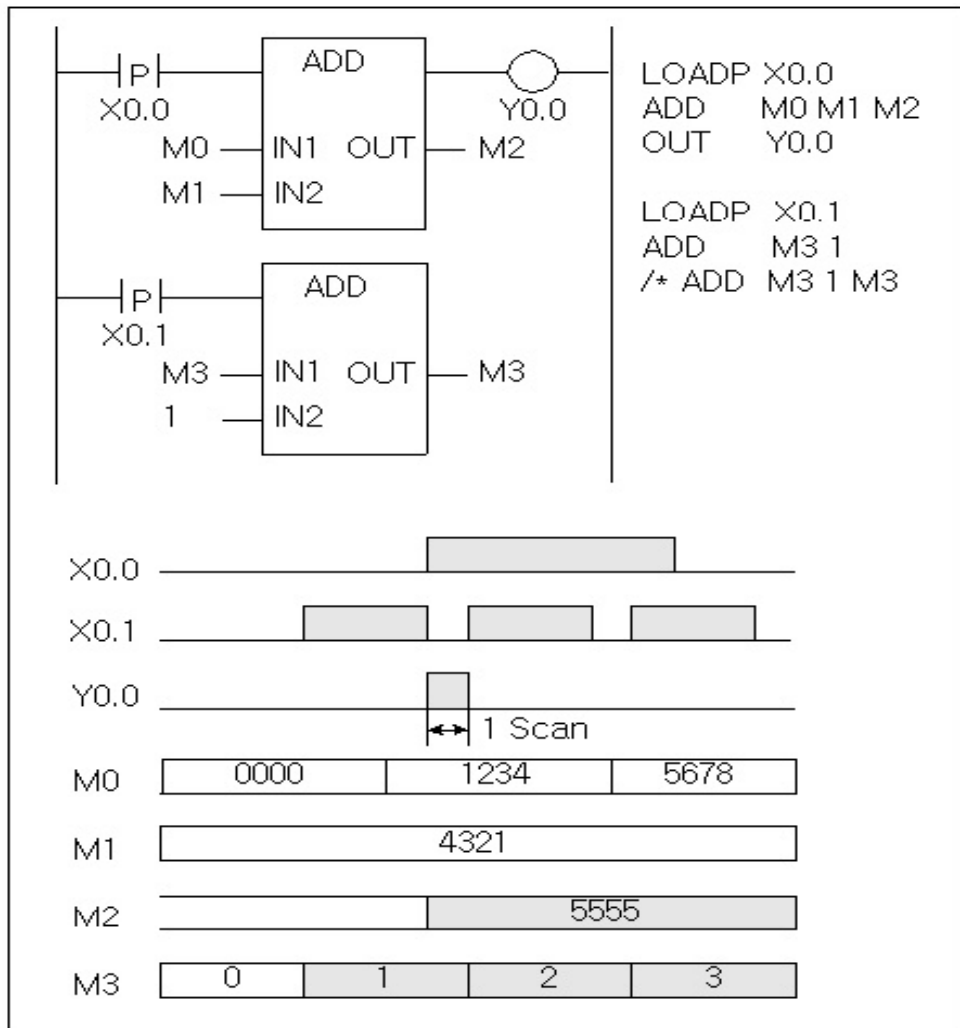


- 1) MOV M0 M1 : M0 M1
- 2) MOV 8000 M2 : 8000() M2
- 3) MOV 16000 M2 : 16000() M2

- MOV M0 D0
- MOV 1234 M0
- MOV -1234 M0
- MOV 1234H D0
- DMOV 1234567890 M0
- DMOV -1234567890 M0
- DMOV 89ABCDEFH M0

(2) ADD, DADD

- INT(16Bits) DINT(32Bits) 가 () .
- ADD : INT(16Bits) INT(16Bits)
- INT(16Bits) 가
- DADD : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) 가

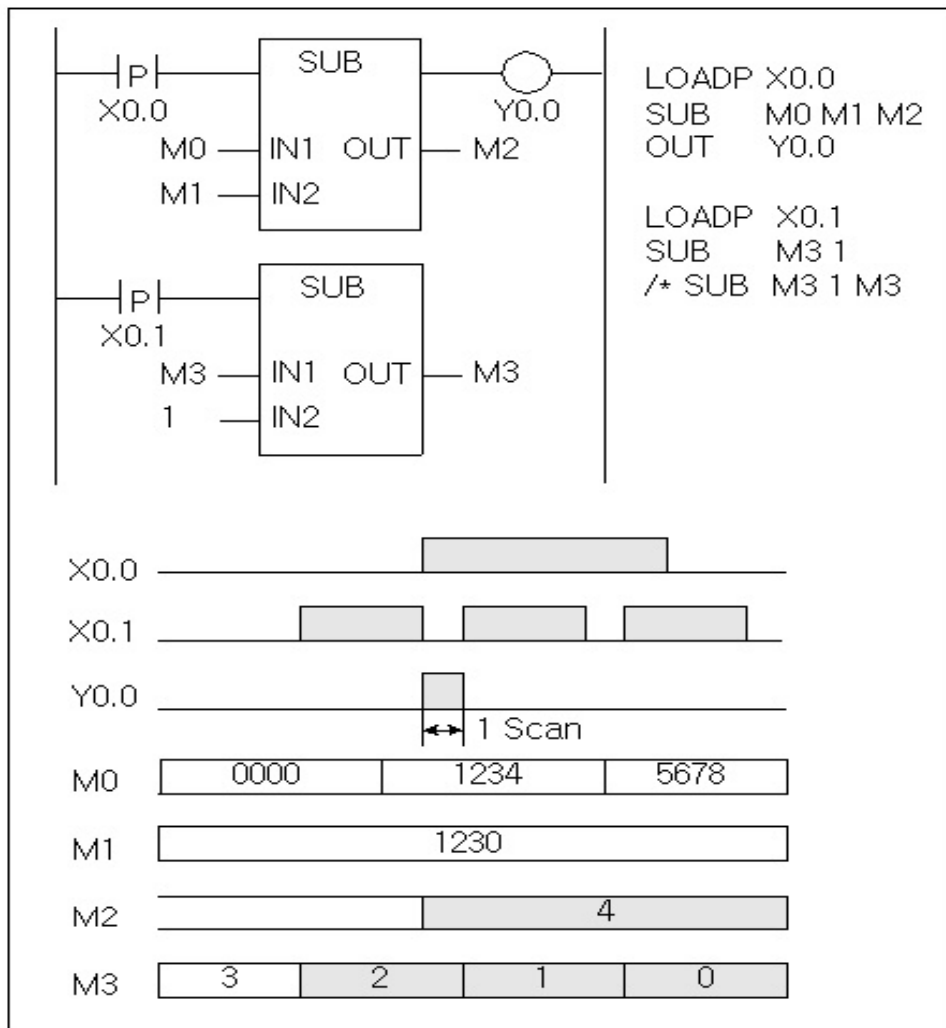


- 1) ADD M0 M1 M2 : M0 M1 M2
- 2) ADD M3 1 : M3 '1' M3
- ADD M3 1 ↔ ADD M3 1 M3

- ADD M0 M1 M2
- ADD M0 1 ↔ ADD M0 1 M0
- ADD M0 -1 ↔ ADD M0 -1 M0
- ADD D0 1234H ↔ ADD D0 1234H D0
- ADD M0 1234H M1
- DADD M0 1234567890 M2
- DADD M0 -1234567890 M2
- DADD M2 89ABCDEFH M4

(3) SUB, DSUB

- INT(16Bits) DINT(32Bits) () .
- SUB : INT(16Bits) INT(16Bits)
- INT(16Bits) .
- DSUB : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) .

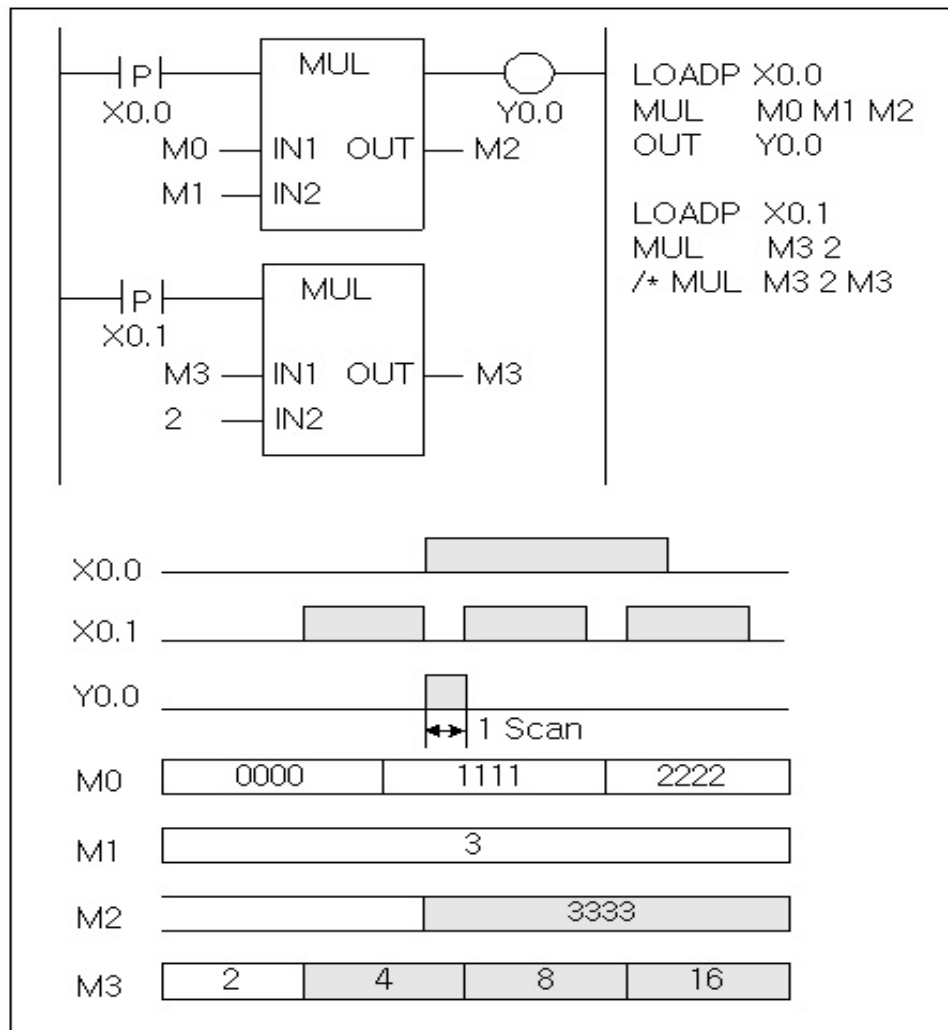


- 1) SUB M0 M1 M2 : M0 M1 M2
 - 2) SUB M3 1 : M3 '1' M3
- SUB M3 1 ↔ SUB M3 1 M3

- SUB M0 M1 M2 ↔ SUB M0 1 M0
- SUB M0 1 ↔ SUB M0 -1 M0
- SUB D0 1234H ↔ SUB D0 1234H D0
- SUB M0 1234H M1
- DSUB M0 1234567890 M2
- DSUB M0 -1234567890 M2
- DSUB M2 89ABCDEFH M4

(4) MUL, DMUL

- INT(16Bits) DINT(32Bits) () .
- MUL : INT(16Bits) INT(16Bits)
- INT(16Bits)
- DMUL : DINT(32Bits) DINT(32Bits)
- DINT(32Bits)

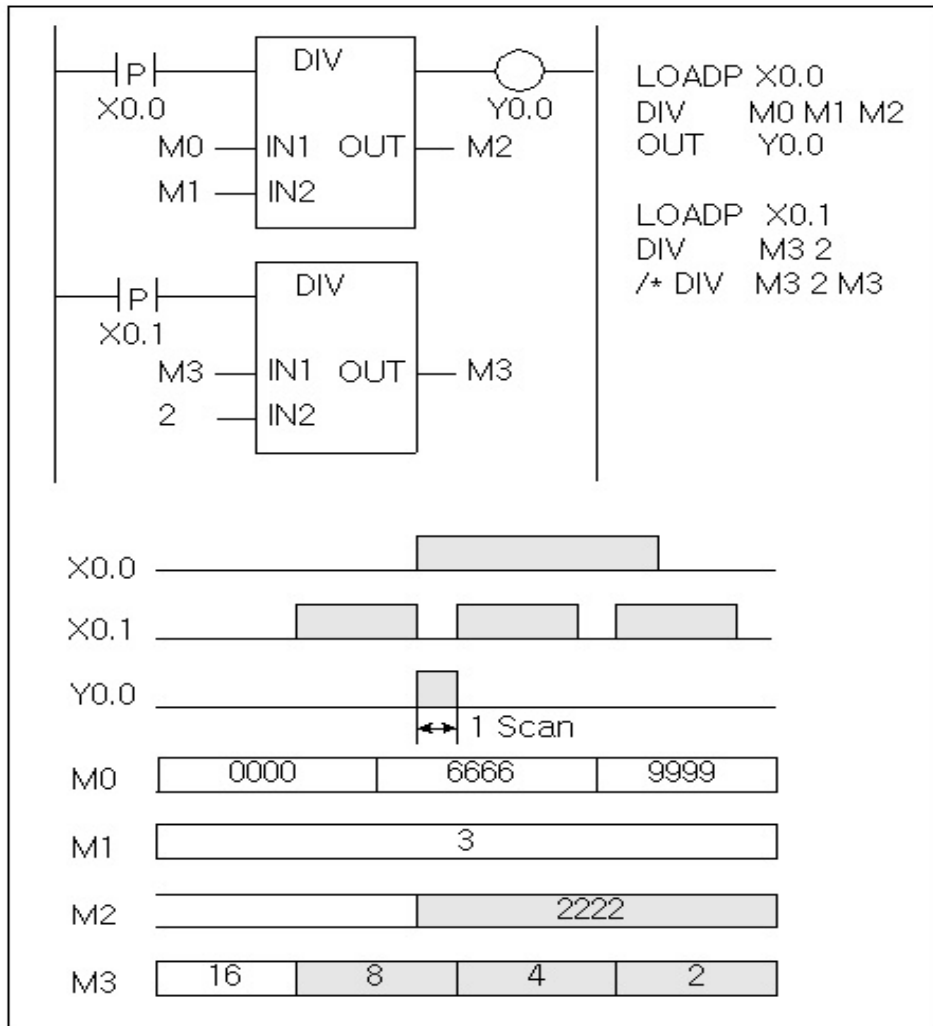


- 1) MUL M0 M1 M2 : M0 M1 M2
- 2) MUL M3 2 : M3 '2' M3
- MUL M3 2 ↔ MUL M3 2 M3

- MUL M0 M1 M2
- MUL M0 1 ↔ MUL M0 1 M0
- MUL M0 -1 ↔ MUL M0 -1 M0
- MUL D0 1234H ↔ MUL D0 1234H D0
- MUL M0 1234H M1
- DMUL M0 1234567890 M2
- DMUL M0 -1234567890 M2
- DMUL M2 89ABCDEFH M4

(5) DIV, DDIV

- INT(16Bits) DINT(32Bits) (,)
- DIV : INT(16Bits) INT(16Bits)
- INT(16Bits)
- DDIV : DINT(32Bits) DINT(32Bits)
- DINT(32Bits)



- 1) DIV M0 M1 M2 : M0 M1 M2
- 2) DIV M3 2 : M3 '2' M3
- DIV M3 2 ↔ DIV M3 2 M3

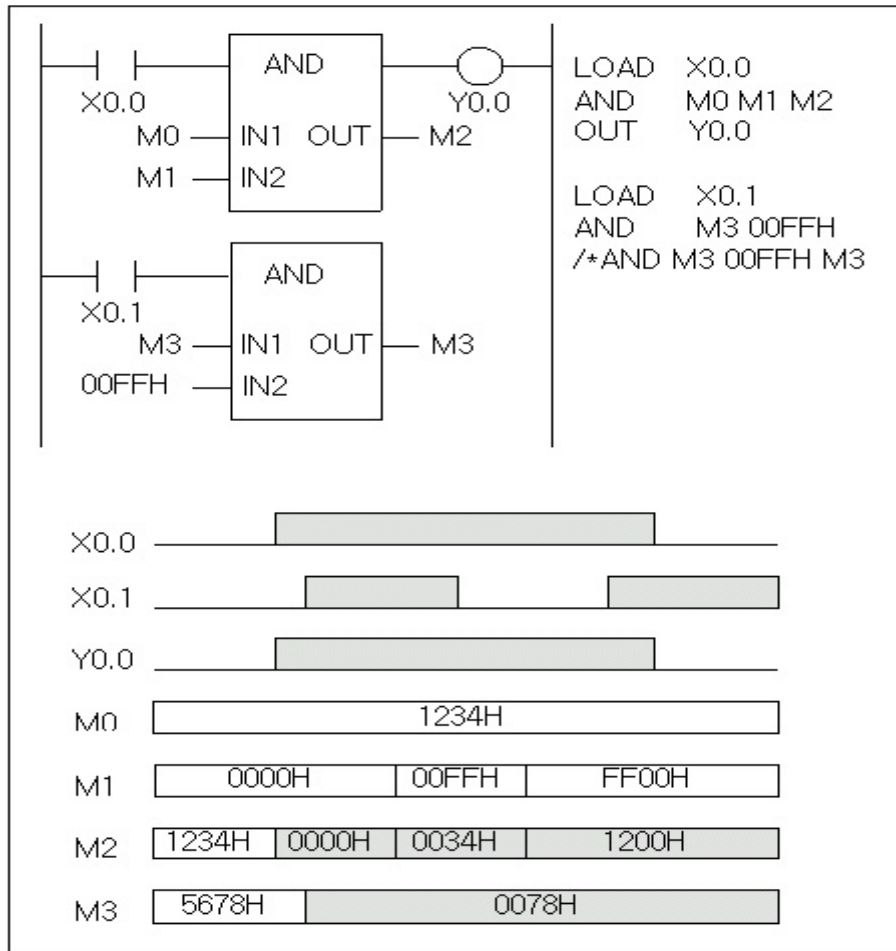
- DIV M0 M1 M2
- DIV M0 1 ↔ DIV M0 1 M0
- DIV M0 -1 ↔ DIV M0 -1 M0
- DIV D0 1234H ↔ DIV D0 1234H D0
- DIV M0 1234H M1
- DDIV M0 1234567890 M2
- DDIV M0 -1234567890 M2
- DDIV M2 89ABCDEFH M4



5) (16Bits, 32Bits)

(1) AND, DAND

- WORD(16Bits) DWORD(32Bits) AND .
- AND : WORD(16Bits) WORD(16Bits)
- DAND : WORD(16Bits) AND .
- DAND : DWORD(32Bits) DWORD(32Bits)
- DAND : DWORD(32Bits) AND .

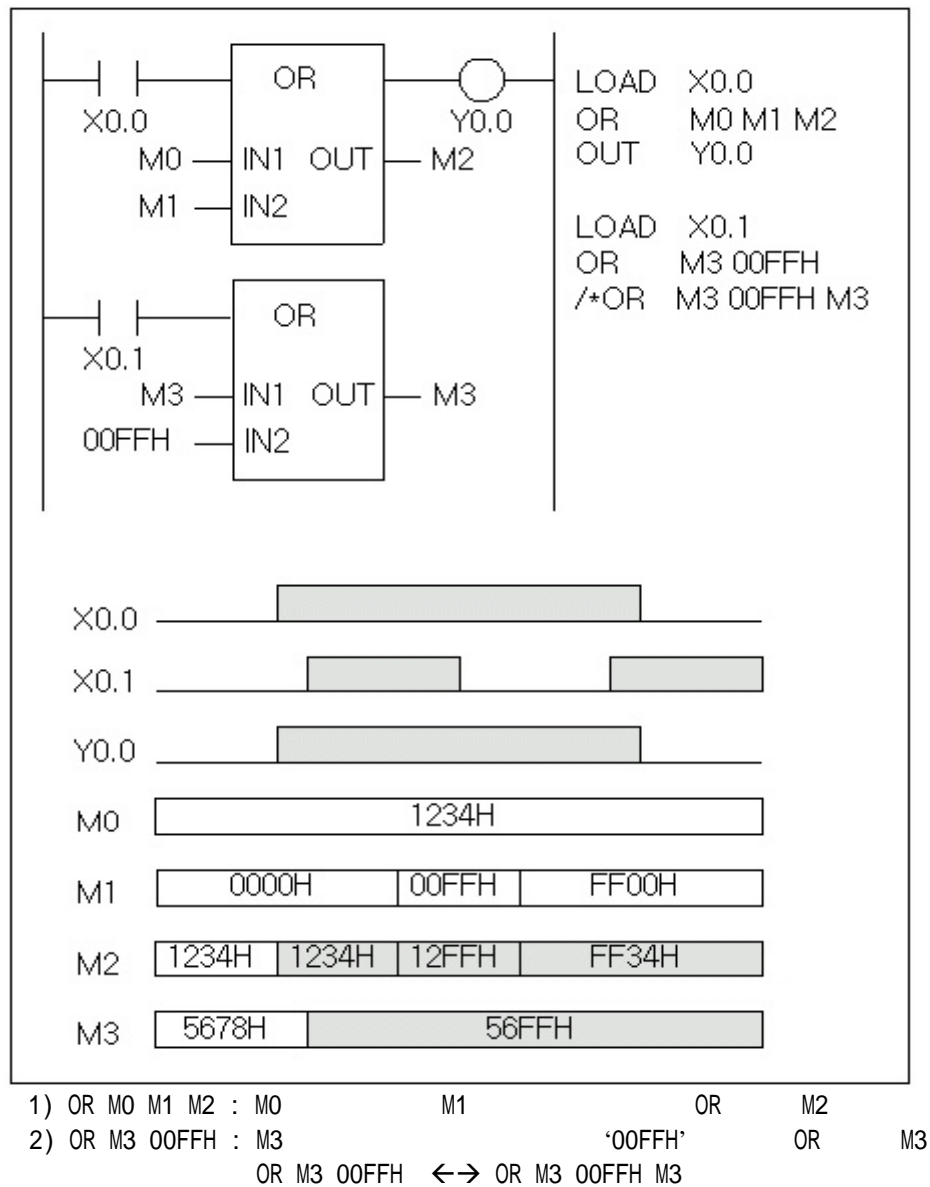


- 1) AND M0 M1 M2 : M0 M1 AND M2 .
 - 2) AND M3 00FFH : M3 '00FFH' AND M3
- AND M3 00FFH ↔ AND M3 00FFH M3

- AND M0 M1 M2
- AND M0 1 ↔ AND M0 1 M0
- AND D0 1234H ↔ AND D0 1234H D0
- AND M0 1234H M1
- DAND M0 1234567890 M2
- DAND M2 89ABCDEFH M4

(2) OR, DOR

- WORD(16Bits) DWORD(32Bits) OR
- OR : WORD(16Bits) WORD(16Bits)
- DOR : WORD(16Bits) OR WORD(16Bits)
- DOR : DWORD(32Bits) OR DWORD(32Bits)
- DOR : DWORD(32Bits) OR DWORD(32Bits)

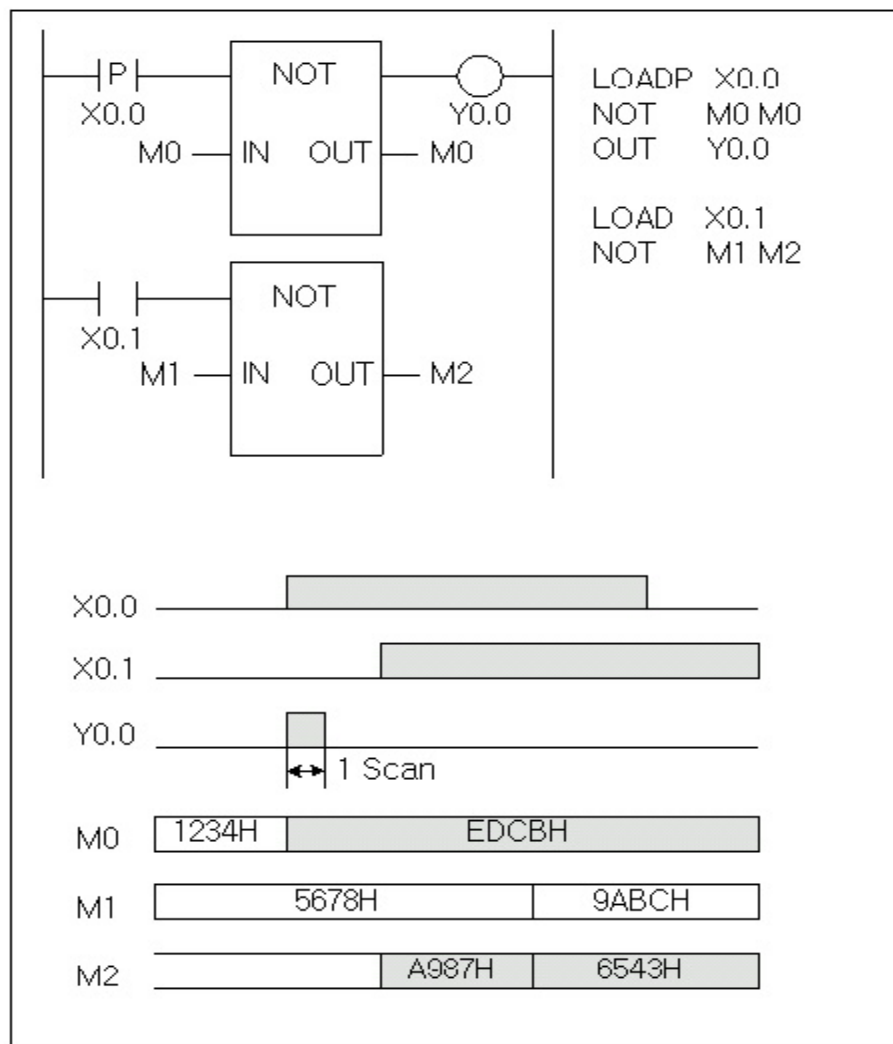


- OR M0 M1 M2
- OR M0 1 ↔ OR M0 1 M0
- OR D0 1234H ↔ OR D0 1234H D0
- OR M0 1234H M1
- DOR M0 1234567890 M2
- DOR M2 89ABCDEFH M4



(4) NOT, DNOT

- WORD(16Bits) DWORD(32Bits) NOT .
- NOT : WORD(16Bits) WORD(16Bits) ,
- WORD(16Bits) NOT .
- DNOT : DWORD(32Bits) DWORD(32Bits) ,
- DWORD(32Bits) NOT .



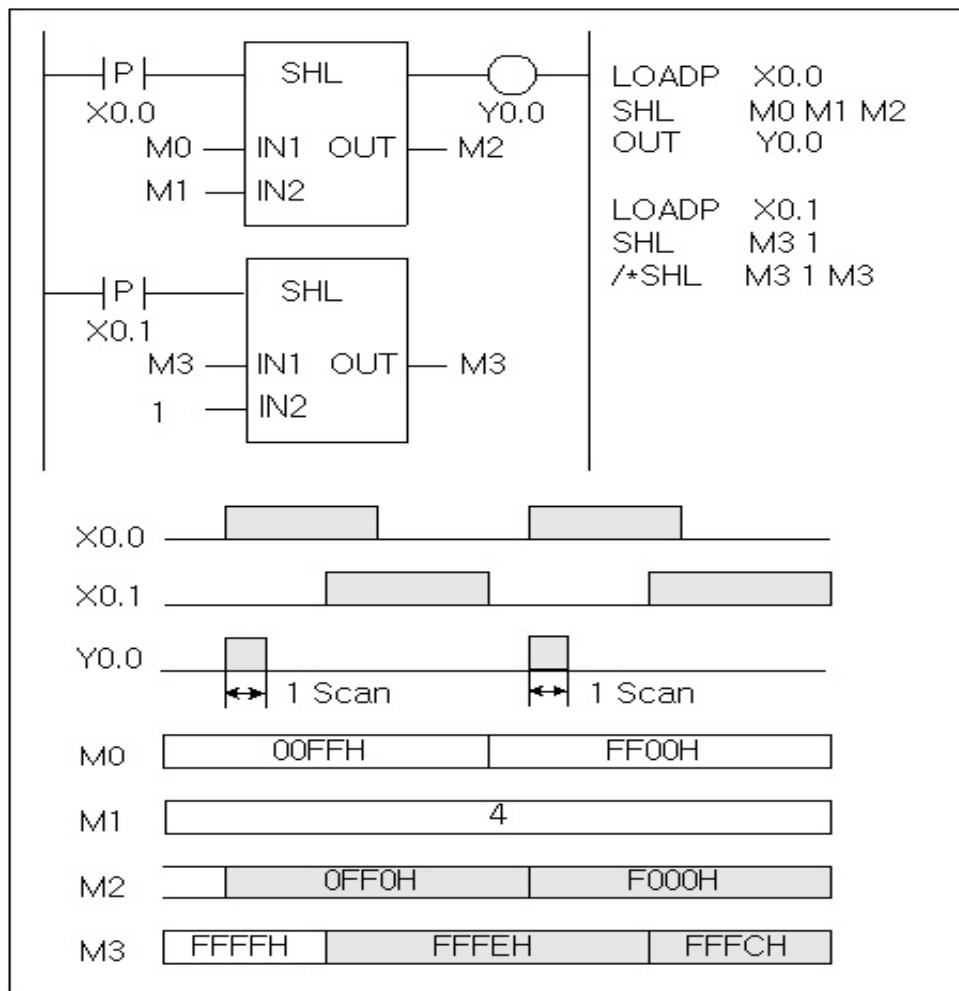
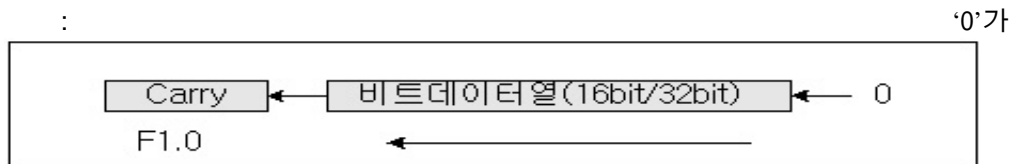
- 1) NOT M0 : M0 NOT M0 .
 NOT M0 ↔ NOT M0 M0
- 2) NOT M1 M2 : M1 NOT M2 .

- NOT M0 M1
- NOT M3 ↔ NOT M3 M3
- NOT D0 ↔ NOT D0 D0
- DNOT M2 ↔ DNOT M2 M2
- DNOT M2 D4

6) (16Bits, 32Bits)

(1) SHL, DSHL

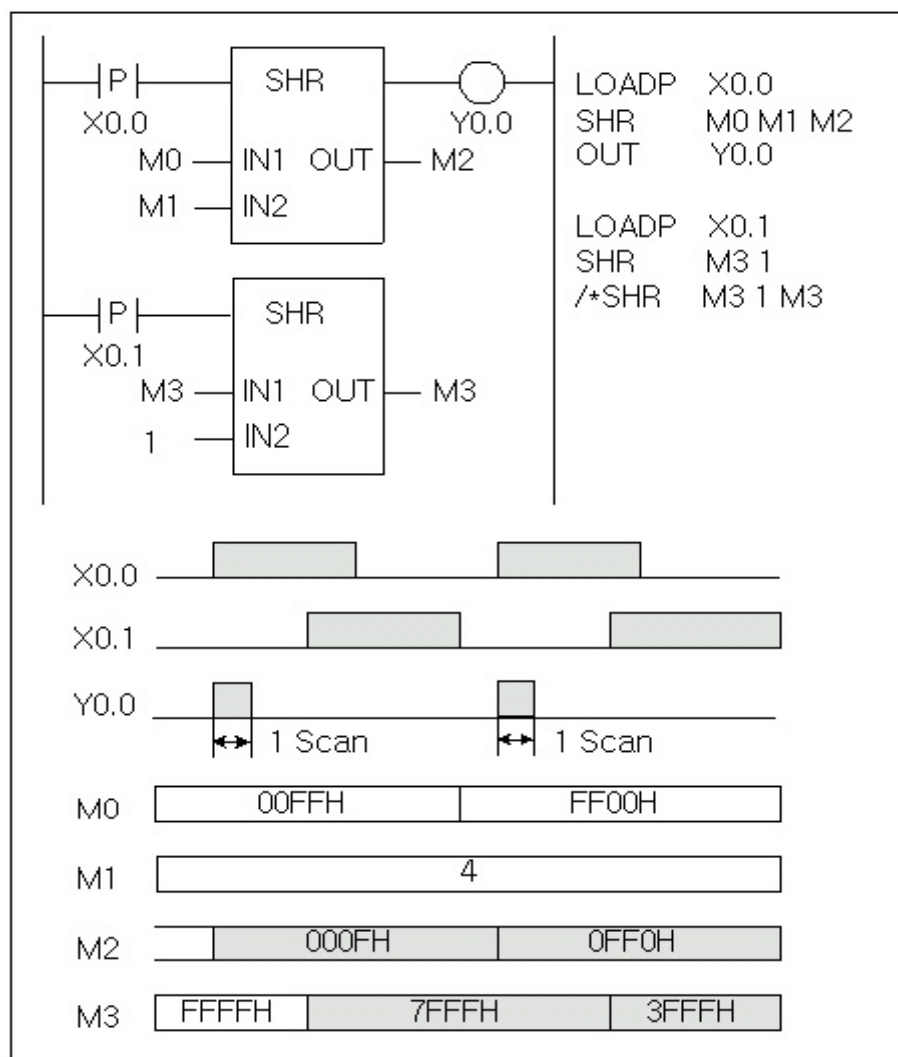
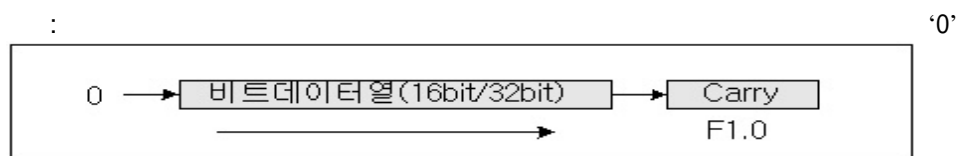
- WORD(16Bits) DWORD(32Bits)
- SHL : WORD(16Bits)
- DSHL : DWORD(32Bits)



- SHL M0 M1 M2
- SHL M0 1 ↔ SHL M0 1 M0
- SHL D0 8 ↔ SHL D0 8 D0
- DSHL M0 4 M2
- DSHL D0 6 D2

(2) SHR, DSHR

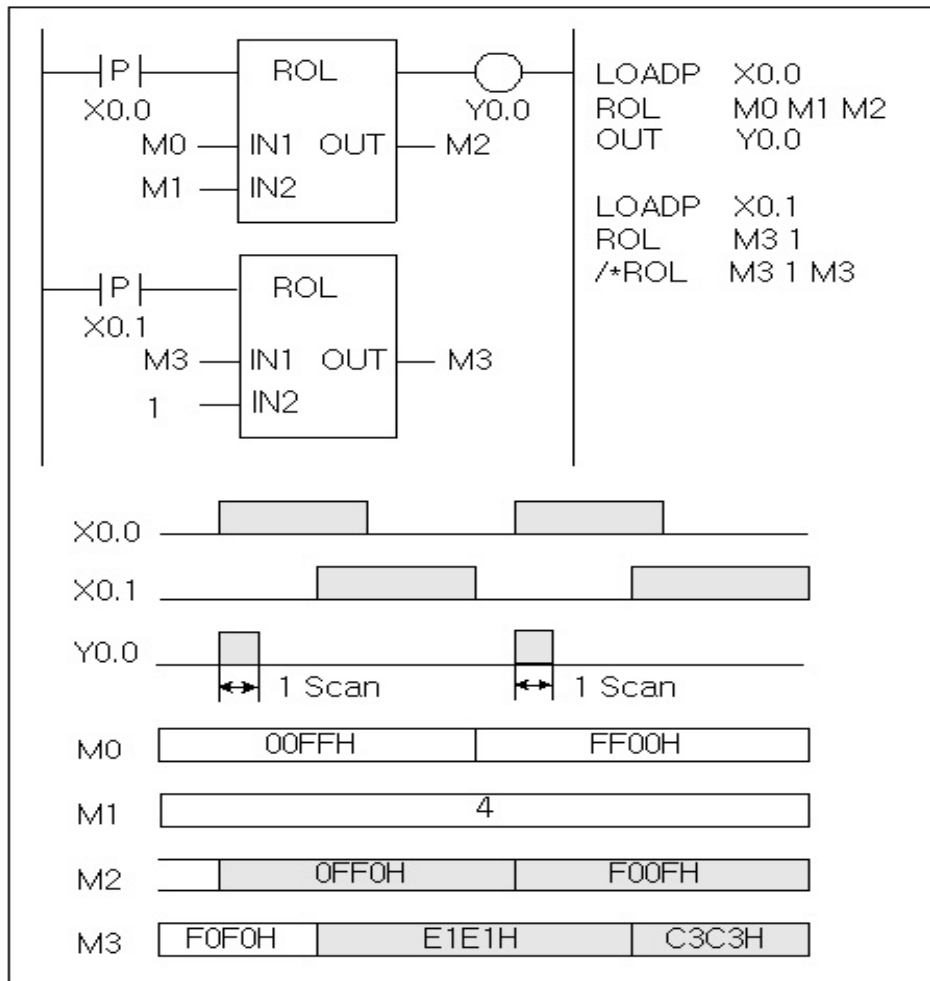
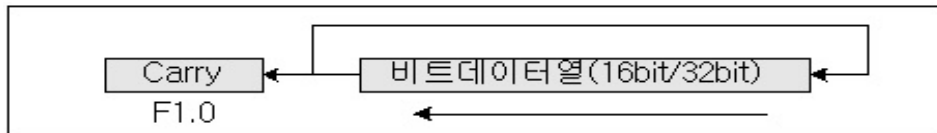
- WORD(16Bits) DWORD(32Bits)
- SHR : WORD(16Bits)
- DSHR : DWORD(32Bits)



- SHR M0 M1 M2
- SHR M0 1 ↔ SHR M0 1 M0
- SHR D0 8 ↔ SHR D0 8 D0
- DSHR M0 4 M2
- DSHR D0 6 D2

(3) ROL, DROL

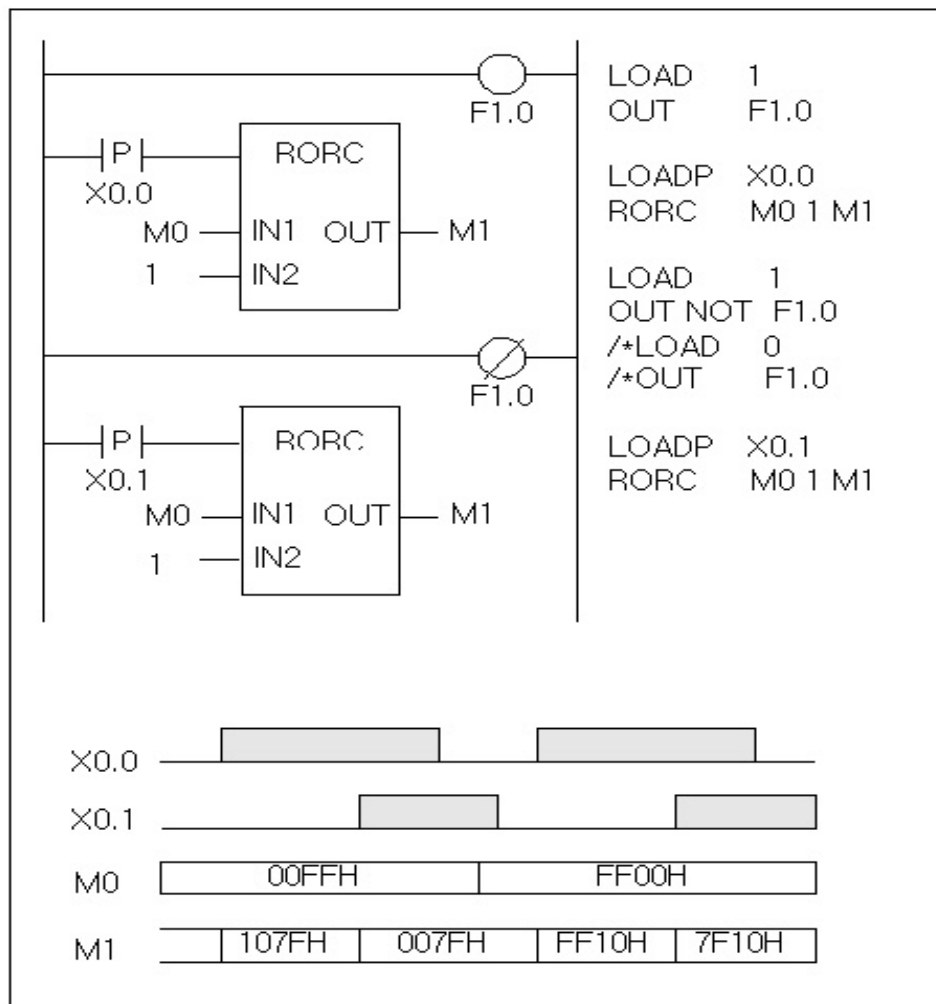
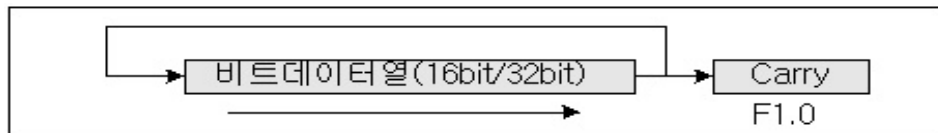
- WORD(16Bits) DWORD(32Bits)
- ROL : WORD(16Bits)
- DROL : DWORD(32Bits)



- ROL M0 M1 M2
- ROL M0 1 ↔ ROL M0 1 M0
- ROL D0 8 ↔ ROL D0 8 D0
- DROL M0 4 M2
- DROL D0 6 D2

(4) ROR, DROR

- WORD(16Bits) DWORD(32Bits)
- ROR : WORD(16Bits)
- DROR : DWORD(32Bits)



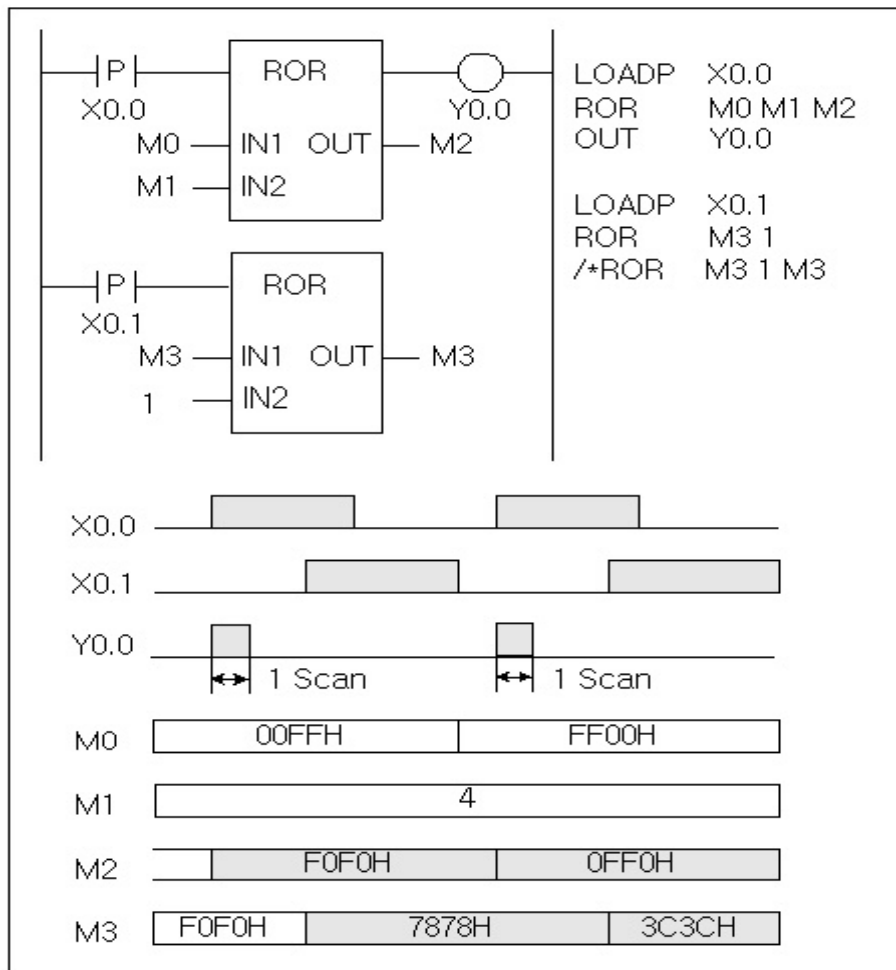
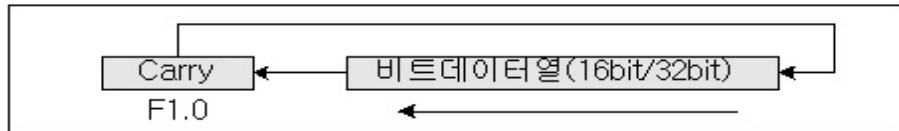
- ROR M0 M1 M2
- ROR M0 1 \leftrightarrow ROR M0 1 M0
- ROR D0 8 \leftrightarrow ROR D0 8 D0
- DROR M0 4 M2
- DROR D0 6 D2

(5) ROLC, DROLC

- WORD(16Bits) DWORD(32Bits) Carry
- ROLC : WORD(16Bits)
 Carry
- DROLC : DWORD(32Bits)
 Carry

:

Carry



- ROLC M0 M1 M2
- ROLC M0 1 ↔ ROLC M0 1 M0
- ROLC D0 8 ↔ ROLC D0 8 D0
- DROLC M0 4 M2
- DROLC D0 6 D2

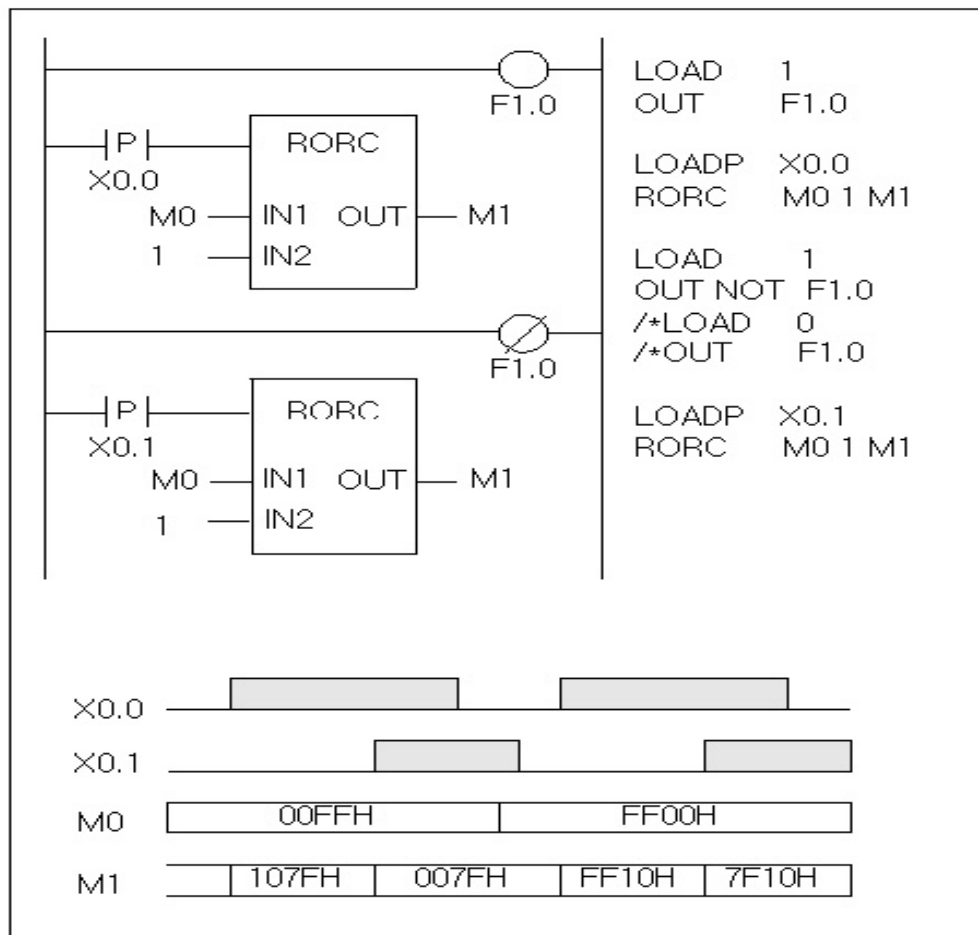
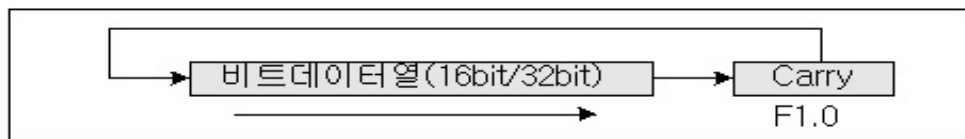
(6) RORC, DRORC

- WORD(16Bits) DWORD(32Bits) Carry

- RORC : WORD(16Bits)
Carry

- DRORC : DWORD(32Bits)
Carry

: Carry



- RORC M0 M1 M2
- RORC M0 1 ↔ RORC M0 1 M0
- RORC D0 8 ↔ RORC D0 8 D0
- DRORC M0 4 M2
- DRORC D0 6 D2

7) (16Bits, 32Bits)

(1) GT, DGT

- INT(16Bits) DINT(32Bits) '(>)

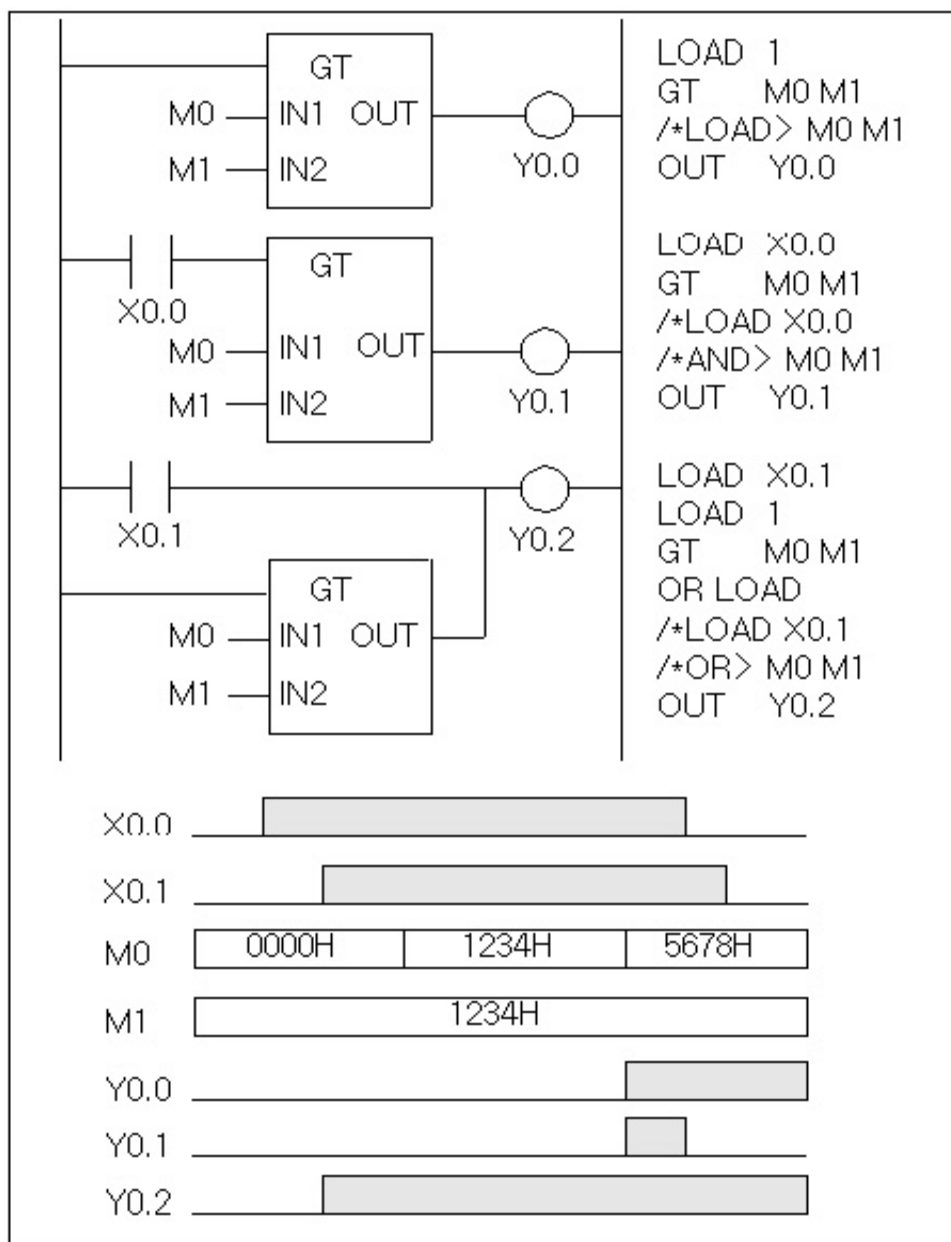
- GT : INT(16Bits) INT(16Bits)

 INT(16Bits) '(>) (Bit Result)

- DGT : DINT(32Bits) DINT(32Bits) DINT

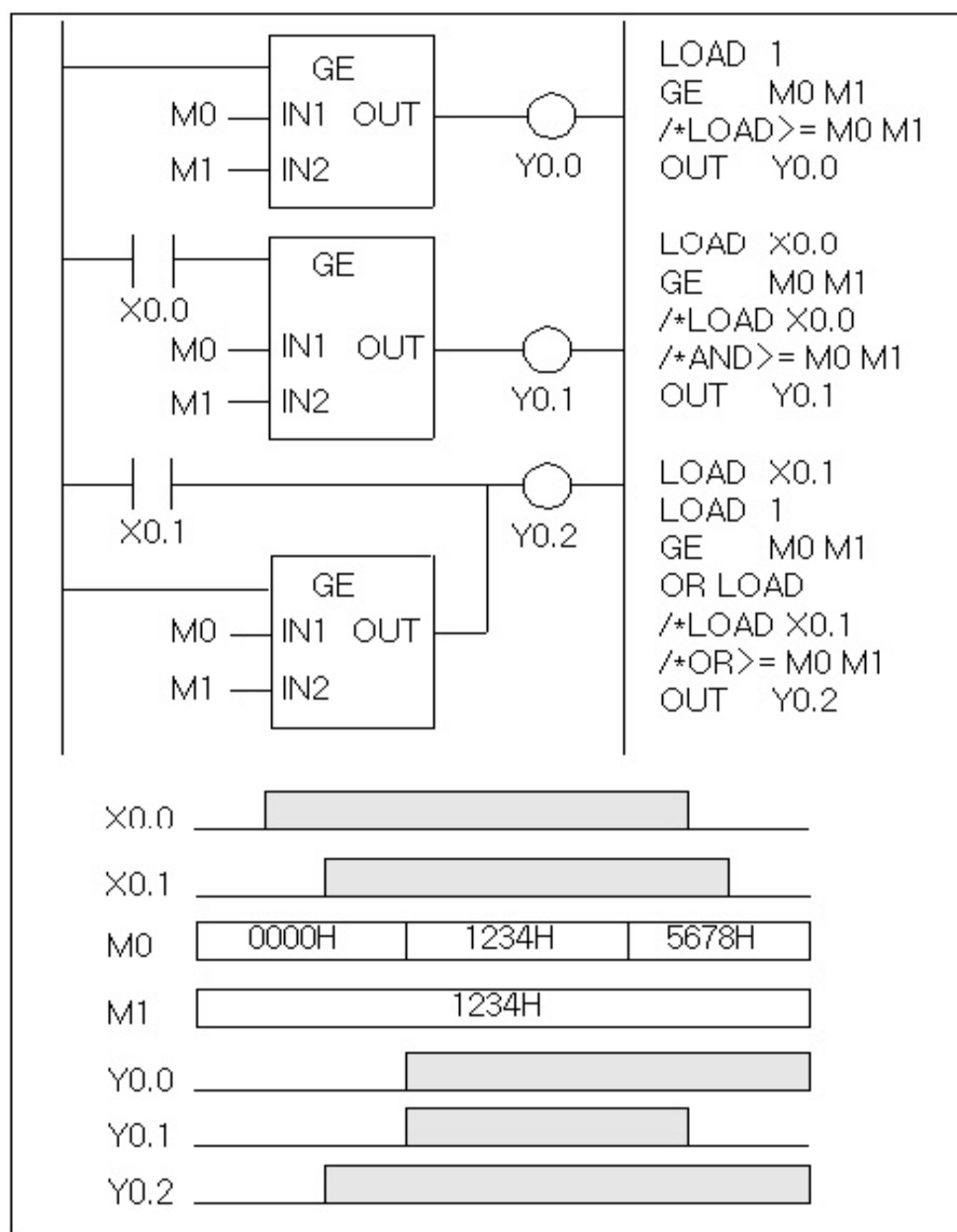
 (32Bits) '(>) (Bit Result)

: "2" (Bit) "



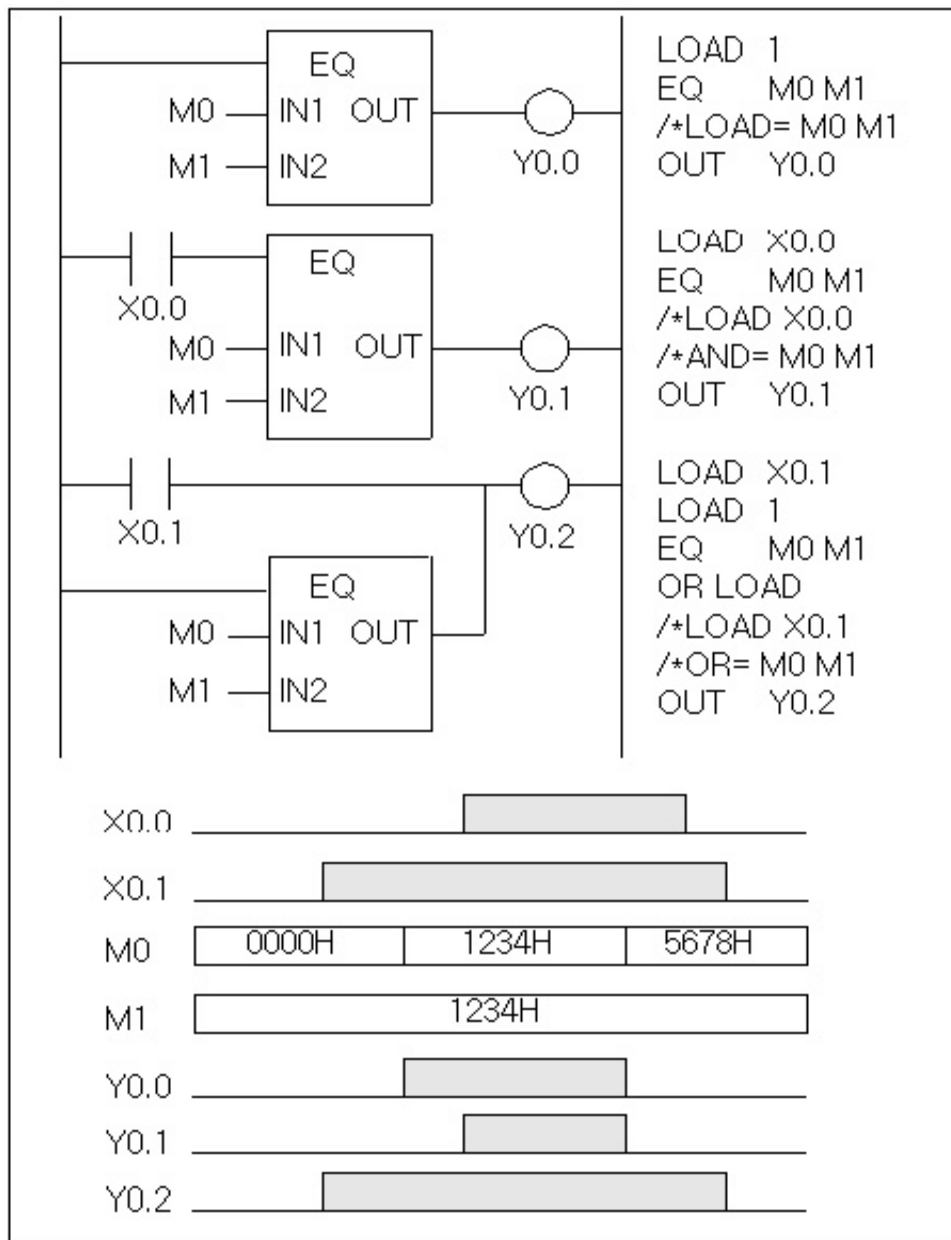
(2) GE, DGE

- INT(16Bits) DINT(32Bits) '(>=) .
- GE : INT(16Bits) INT(16Bits) (Bit
- INT(16Bits) '(>=)
- Result)
- DGE : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(>=)
- (Bit Result)
- : "2) (Bit) "



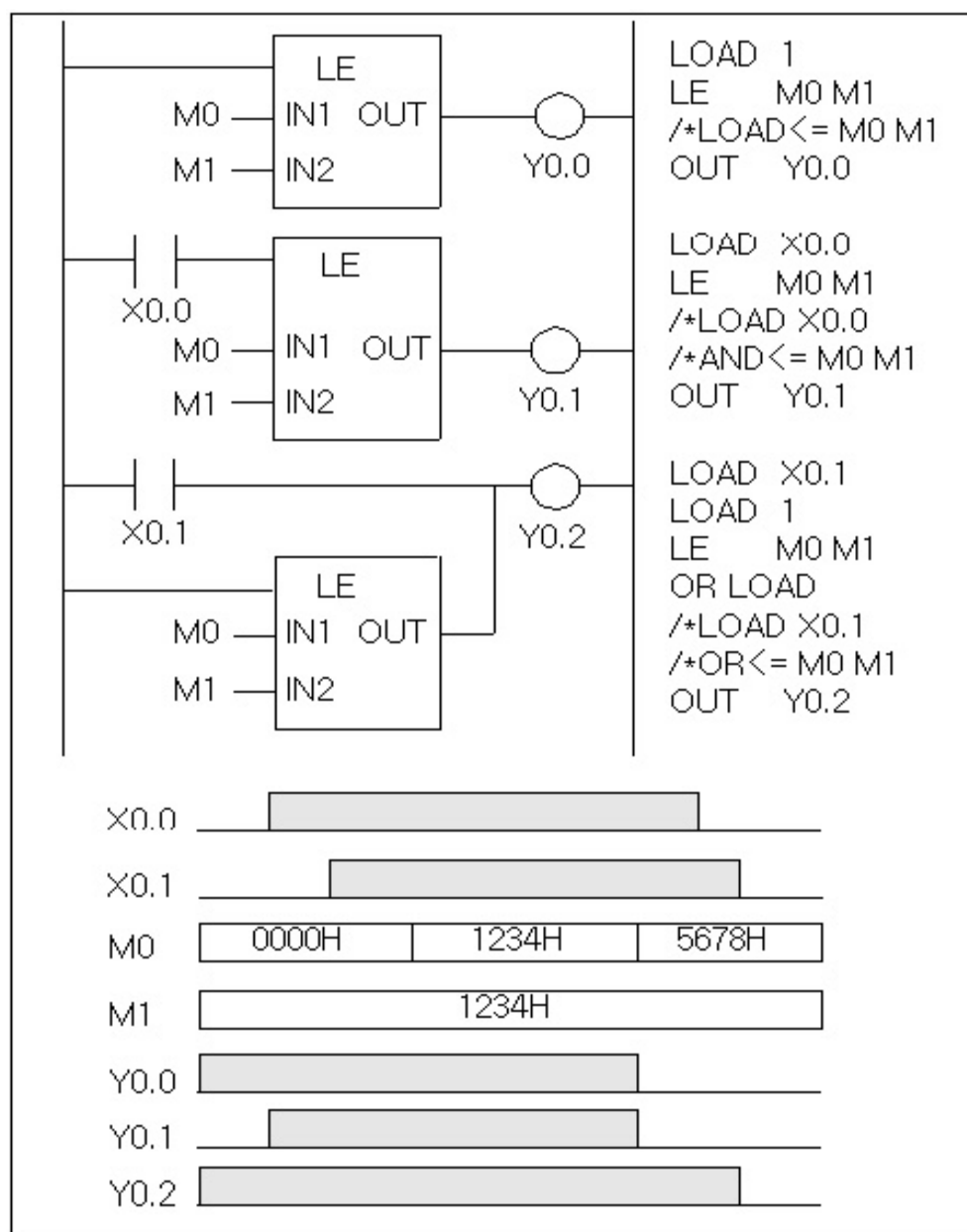
(3) EQ, DEQ

- INT(16Bits) DINT(32Bits) ' (=) .
- EQ : INT(16Bits) ' (=) (Bit Result)
- DEQ DINT(32Bits) DINT(32Bits)
- DINT(32Bits) ' (=) (Bit Result)
- : "2) (Bit) "



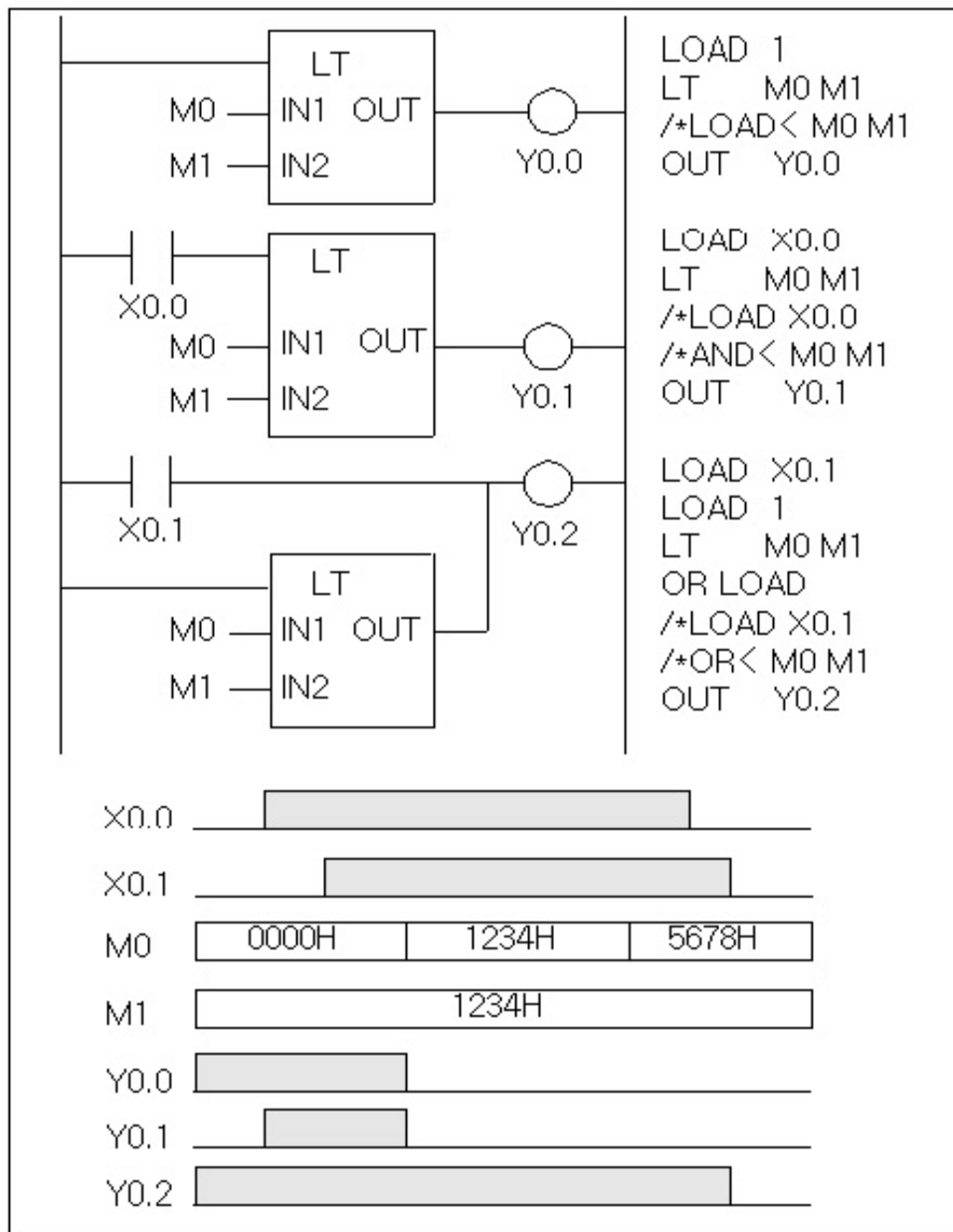
(4) LE, DLE

- INT(16Bits) DINT(32Bits) '(<=)
- LE : INT(16Bits) INT(16Bits)
- INT(16Bits) '(<=) (Bit Result)
- DLE : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(<=)
- (Bit Result)
- : "2) (Bit) "



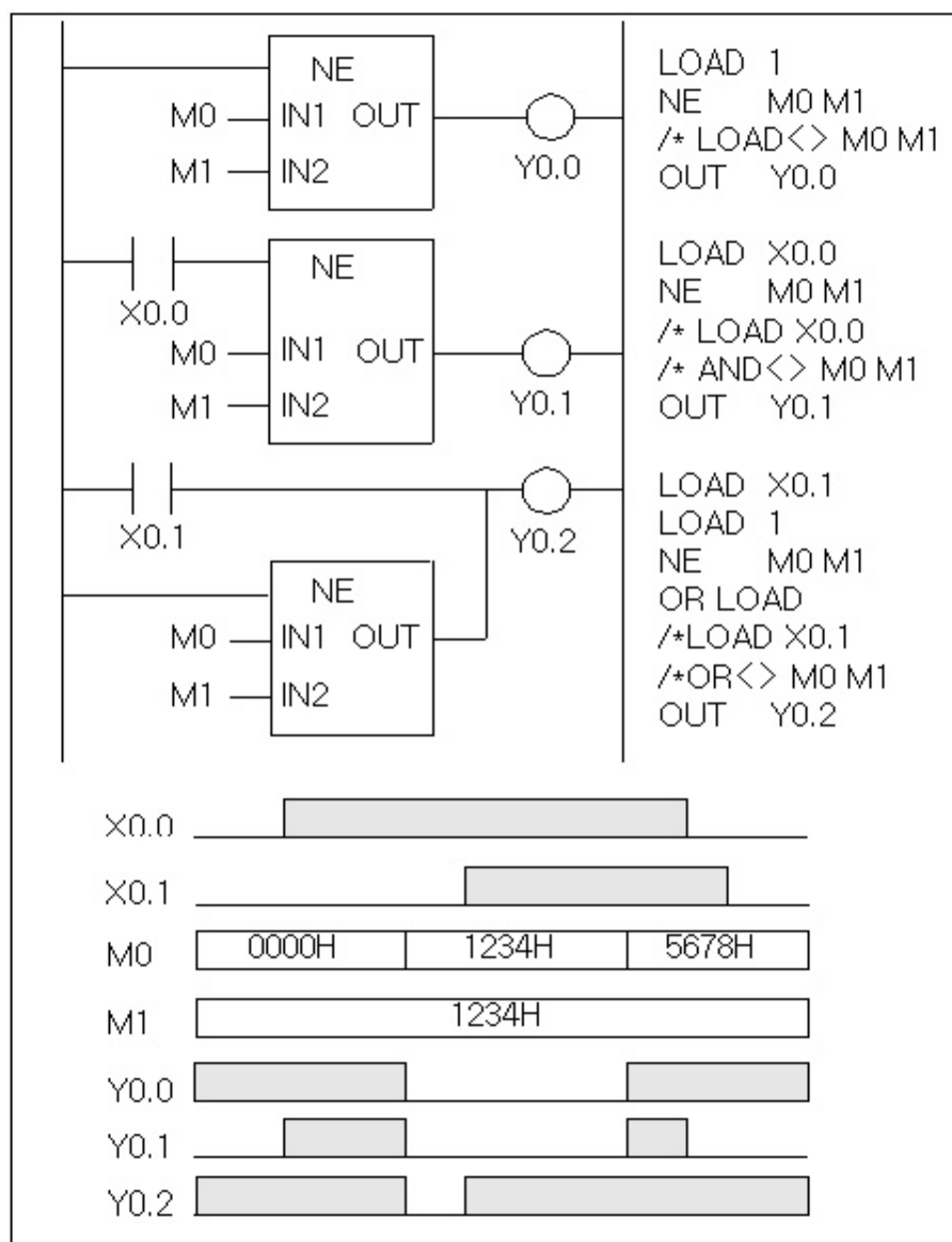
(5) LT, DLT

- INT(16Bits) DINT(32Bits) '(<)
- LT : INT(16Bits) INT(16Bits)
- INT(16Bits) '(<)
- (Bit Result)
- DLT : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(<)
- (Bit Result)
- : "2) (Bit)



(6) NE, DNE

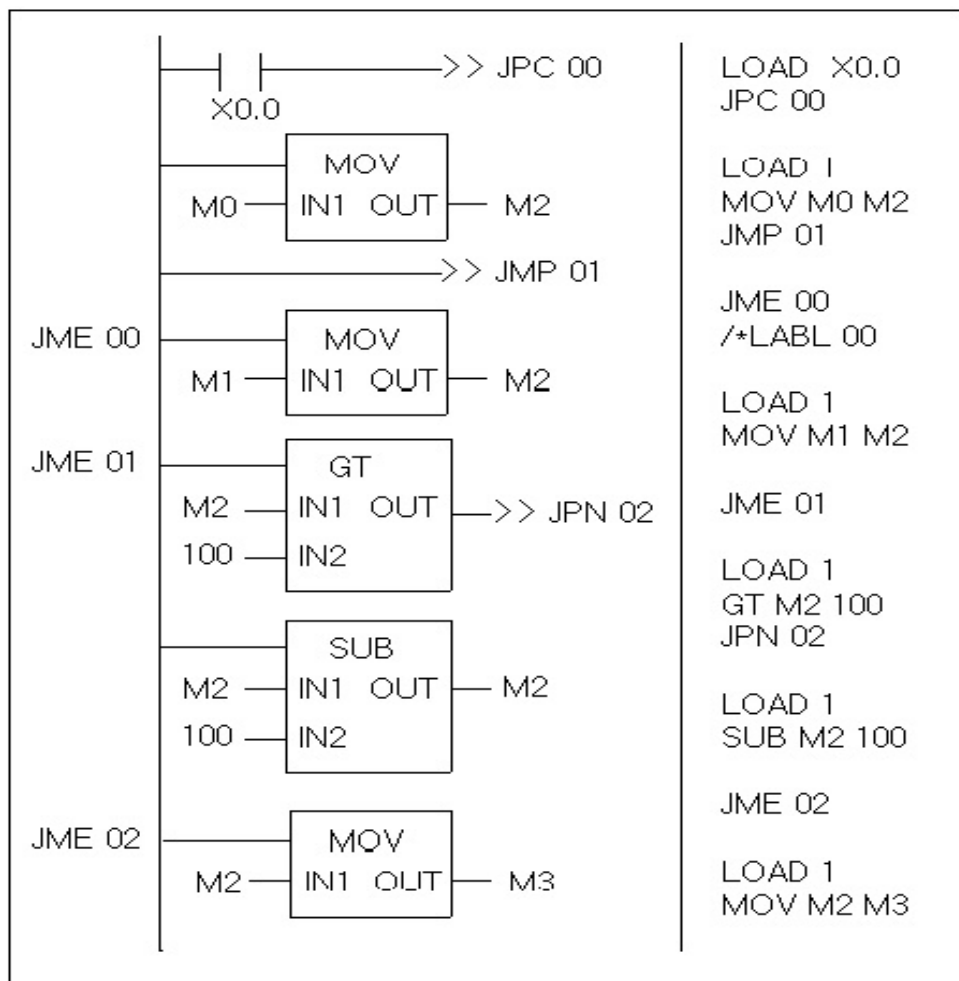
- INT(16Bits) DINT(32Bits) '(<>)
- NE ; INT(16Bits) INT(16Bits)
- INT(16Bits) '(<>) (Bit)
- Result)
- DNE : DINT(32Bits) DINT(32Bits)
- DINT(32Bits) '(<>) (Bit)
- Result)
- :“2) (Bit) ”



8)

(1) JMP, JPC, JPN \leftrightarrow JME (or LABL)

- - JMP
 - JPC
 - JPN
 - JME or LABL
- ‘ON’
‘OFF’
- “0~99”



(2) END

- - END
 - END
- 가

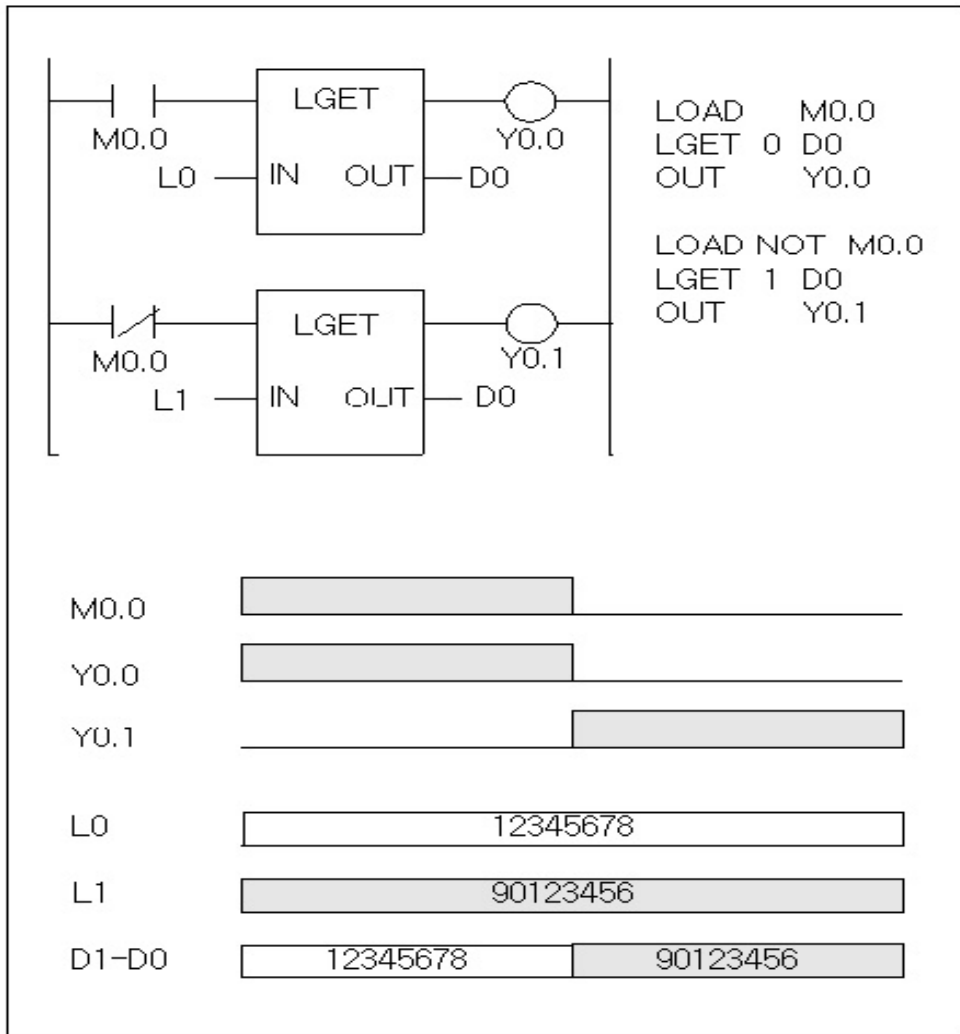
9) MC

(1) LGET

```

- LGET : L          L          FLOATING(32Bits)          DINT(32Bits)
      PLC
- L          FLOATING(32Bits)          LGET          L          DINT(32Bits)
      PLC
(      :          )

```



```

- LGET 0 D0      : L0          D1-D0 (32Bits)
- LGET 1 D0      : L1          D1-D0 (32Bits)
*      : LGET  < L      > <      >

```

- LGET 0 D0 : L0 → D1-D0
- LGET 100 D200 : L100 → D201-D200
- LGET 200 M0 : L200 → M1-M0
- LGET 1000 M10 : L1000 → M11-M10
- LGET 0 Y2 : L0 → Y3-Y2

(2) LPUT

- LPUT : PLC

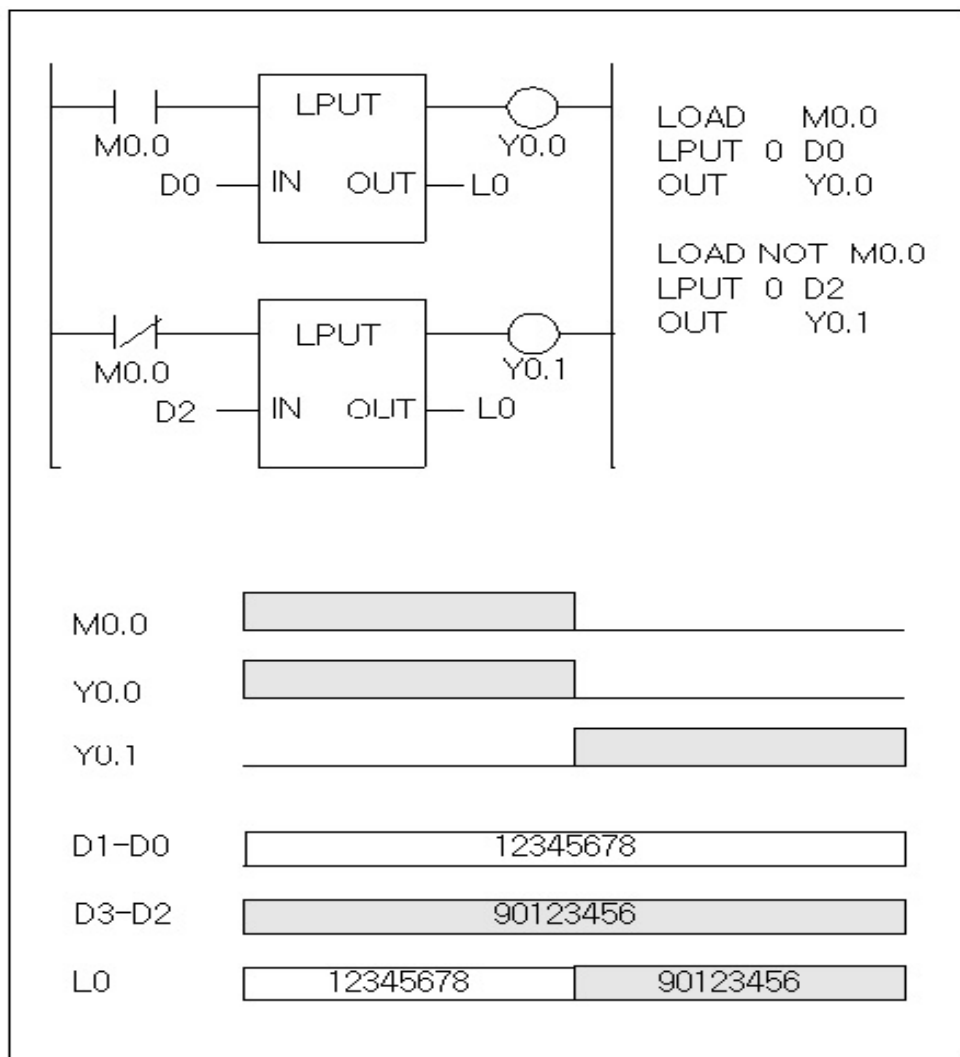
DINT(32Bits) FLOATING(32Bits)

L L

- L FLOATING(32Bits) LPUT PLC DINT

(32Bits) FLOATING(32Bits) L L

(:)



1) LPUT 0 D0 : D1-D0 (32Bits) L0 .

2) LPUT 0 D2 : D3-D2 (32Bits) L0 .

* : LPUT < L > < >

- LPUT 0 D0 : D1-D0 → L0

- LPUT 100 D200 : D201-D200 → L100

- LPUT 200 M0 : M1-M0 → L200

- LPUT 1000 M10 : M11-M10 → L1000

- LPUT 0 Y2 : Y3-Y2 → L0

(3) FWR

- FWR : L (FLASH) Edge . (: (NVRAM)
- Flash 10 가 Flash .

