1 Basic Instruction

1.1 Contact Instruction

1.1.1 LD, LDI

Supported PLC series

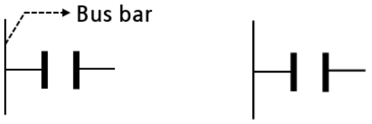
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

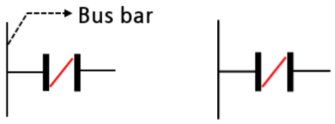
LD starts contact ‘A’ operation and LDI starts contact ‘B’ operation.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
LD(I)	*S*	O	O	O	O	O	O	O	O	O	-	-	-	-	-	-	1	-	-	-

LD



LDI



S	Bit device number

- LD indicates a logical start and creates an ON/OFF execution condition based on the ON/OFF status of the assigned bit device.
- LDI indicates a logical start and creates an ON/OFF execution condition based on the reverse of the ON/OFF status of the assigned bit device.
- LD and LDI instruction are connected to the bus bar or at the beginning of the logic block by ANB or ORB.

Execution Condition

LD and LDI are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the LD and LDI instructions.

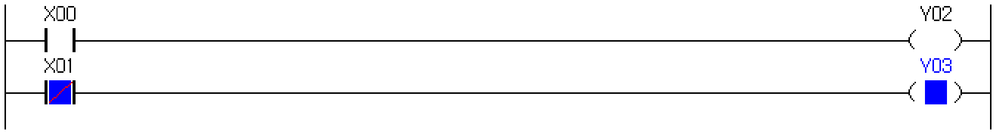
Program Example

LD, LDI

In the following example, LD and LDI execute as below.

- When the X00 turns ON, Y02 turns ON.
- When the X01 is ON, Y03 turns OFF.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
OUT	Y02
LDI	X01
OUT	Y03

1.1.2 AND, ANDI

Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

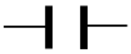
Function

AND is contact ‘A’ series connection instruction and ANDI is contact ‘B’ series connection instruction.

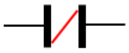
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero (Old) (CP)	Carry
AND	*S*	O	O	O	O	O	O	O	O	O	-	-	-	-	-	-	1	-	O	-

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ANDI	*S*	O	O	O	O	O	O	O	O	O	-	-	-	-	-	-	1	-	-	-

AND



ANDI



S	Bit device number

- AND instruction performs AND operation based on the ON/OFF status of the assigned bit device.
- ANDI instruction performs AND operation based on the reverse of the ON/OFF status of the assigned bit device.
- AND and ANDI perform an AND operation with the operation result to that point, and use the resulting value as a new operation result.

Execution Condition

AND and ANDI are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the AND and ANDI instructions.

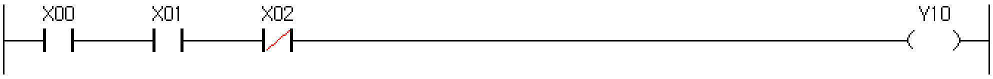
Program Example

AND, ANDI

In the following example, AND and ANDI execute as below.

- When the X00 and X01 turn ON, then Y10 turns on.
- When the X00, X01 and X02 turn ON, then Y10 turns OFF.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
AND	X01
ANDI	X02
OUT	Y10

1.1.3 OR, ORI

Supported PLC series

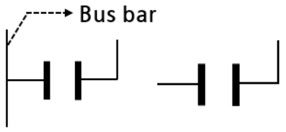
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

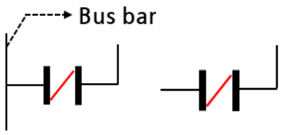
OR is the contact 'A' single parallel connection instruction and ORI is the contact 'B' single parallel connection instruction.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
OR(I)	*S*	O	O	O	O	O	O	O	O	O	-	-	-	-	-	-	1	-	-	-

OR



ORI



S	Bit device number (Size=1)

- OR instruction performs OR operation based on the ON/OFF status of the assigned bit device.
- ORI instruction performs OR operation based on the reverse of the ON/OFF status of the assigned bit device.
- OR and ORI instructions are connected to the bus bar or at the beginning of the logic block.

Execution Condition

OR and ORI are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the OR and ORI instructions.

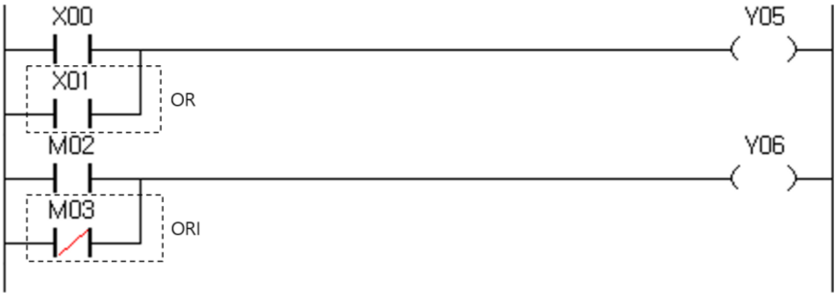
Program Example

OR, ORI

In the following example, OR and ORI execute as below.

- When the X00 or X01 turns ON, then Y05 turns ON.
- When the M02 or M03 turns ON, then Y06 turns ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
OR	X01
OUT	Y05
LD	M02
ORI	M03
OUT	Y06

1.1.4 LDP, LDF

Supported PLC series

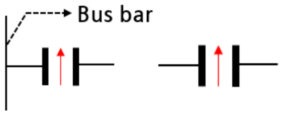
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

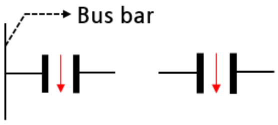
LDP starts rising edge pulse operation and LDF starts falling edge pulse operation.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
LDP(F)	S	O	O	O	O	O	O	O	O	-	-	-	-	-	-	-	1	-	-	-

LDP



LDF



S	Bit device number. (Size=1)

LDP

- LDP is ON for 1 scan only, at the rising edge of the assigned bit device.
- LDP turns ON when the assigned bit device goes from OFF to ON.

LDF

- LDF is ON for 1 scan only, at the falling edge of the assigned bit device.
- LDF turns ON when the assigned bit device goes from ON to OFF.

Execution Condition

LDP and LDF are executed every scan independently of the device status and operation result.

Operation Error

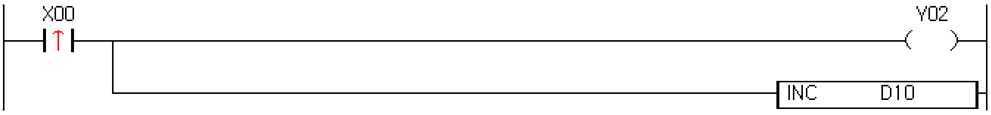
There is no operation error in the LDP and LDF instructions.

Program Example

LDP

In the following example, LDP executes as below.

- When the X00 goes from OFF to ON.
- Then the Y02 turns ON for 1 scan and D10 increases once.



Instruction List (IL)

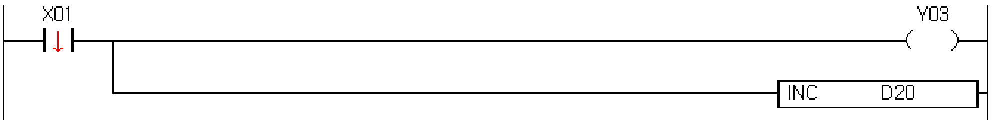
Instruction	Device
LDP	X00
OUT	Y02
INC	D10

LDF

In the following example, LDF executes as below.

- When the X01 goes from ON to OFF.
- Then the Y03 turns ON for 1 scan and D20 increases once.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LDF	X01
OUT	Y03
INC	D20

1.1.5 ANDP, ANDF

Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

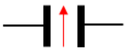
Function

ANDP starts rising edge pulse operation in series connection and ANDF starts falling edge pulse operation in series connection.

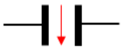
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero (Old) (CP)	Carry
ANDP	*S*	O	O	O	O	O	O	O	O	-	-	-	-	-	-	-	1	-	O	-

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ANDF	*S*	O	O	O	O	O	O	O	O	-	-	-	-	-	-	-	1	-	-	-

ANDP



ANDF



S	Bit device number (Size=1)

- ANDP and ANDF perform an AND operation with the operation result to that point, and use the resulting value as a new operation result.

ANDP

- ANDP is ON for 1 scan only, at the rising edge of the assigned bit device.
- ANDP turns ON when the assigned bit device goes from OFF to ON.

ANDF

- ANDF is ON for 1 scan only, at the falling edge of the assigned bit device.
- ANDF turns ON when the assigned bit device goes from ON to OFF.

Execution Condition

ANDP and ANDF are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the ANDP and ANDF instructions.

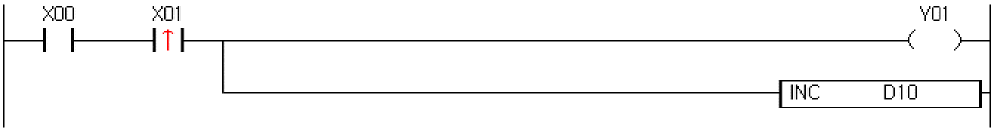
Program Example

ANDP

In the following example, ANDP executes as below.

- When the X00 turns ON, the Y01 does not turns ON.
- When the X00 turns ON, and the X01 goes from OFF to ON,
- Then the Y01 turns ON for 1 scan and D10 increases once.

Ladder Diagram (LD)



Instruction List (IL)

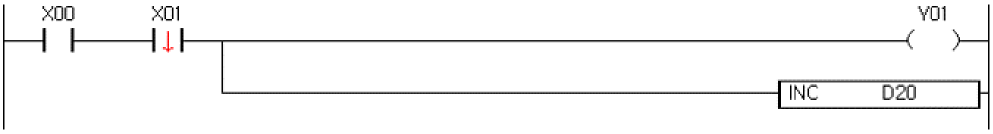
Instruction	Device
LD	X00
ANDP	X01
OUT	Y01
INC	D10

ANDF

In the following example, ANDF executes as below.

- When the X00 turns ON, the Y01 does not turns ON.
- When the X00 turns ON, and the X01 goes from ON to OFF,
- Then the Y01 turns ON for 1 scan and D20 increases once.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
ANDF	X01
OUT	Y01
INC	D20

1.1.6 ORP, ORF

Supported PLC series

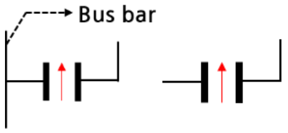
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

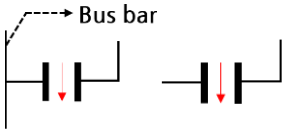
ORP starts rising edge pulse operation in parallel connection and ORF starts falling operation in series connection.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ORP(F)	*S*	O	O	O	O	O	O	O	O	-	-	-	-	-	-	-	1	-	-	-

ORP



ORF



S	Bit device number. (Size=1)

- ORP and ORF perform an OR operation with the operation result to that point, and use the resulting value as a new operation result.

ORP

- ORP is ON for 1 scan only, at the rising edge of the assigned bit device.
- ORP turns ON when the assigned bit device goes from OFF to ON.

ORF

- ORF is ON for 1 scan only, at the falling edge of the assigned bit device.
- ORF turns ON when the assigned bit device goes from ON to OFF.

Execution Condition

ORP and ORF are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the ORP and ORF instructions.

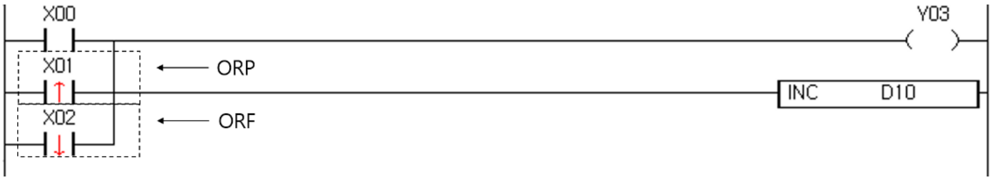
Program Example

ORP, ORF

In the following example, ORP and ORF execute as below.

- When the X01 goes from OFF to ON then the Y03 turns ON for 1 scan and D10 increases once.
- When the X02 goes from ON to OFF then the Y03 turns ON for 1 scan and D10 increases once.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
ORP	X01
ORF	X02
OUT	Y03
INC	D10

1.1.7 INV

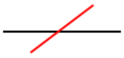
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

INV performs a reverse operation with the operation result to that point, and use the resulting value as a new operation result.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
INV	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



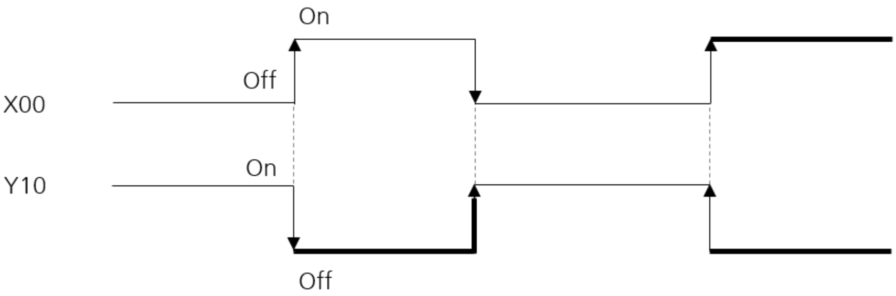
INV

- Contact ‘A’ reverses to contact ‘B’, and contact ‘B’ reverses to contact ‘A’.
- Series connection reverses to parallel connection, and parallel connection reverses to series connection.

Execution Condition

INV operates based on the results made until the INV instruction. Then the INV instruction reverses the resulting value. Accordingly, use it in the same position as that of the AND instruction.

Timing Chart



⚠ INV cannot be used at the LD and OR positions.

⚠ When using a ladder block, the operation result is inverted within the range of the ladder block. To operate a ladder using the INV with the ANB, pay attention to the range of inversion.

Operation Error

There is no operation error in the INV instruction.

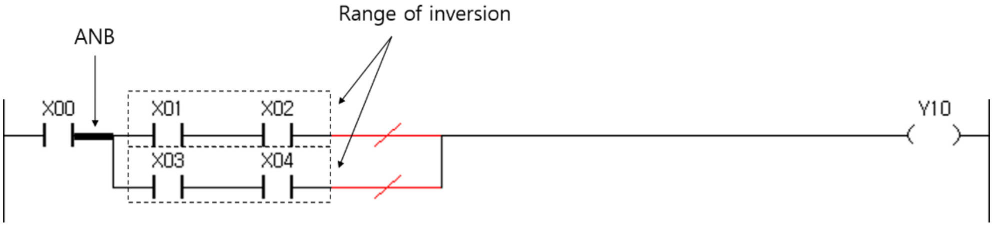
Program Example

INV

In the following example, INV reverses the result in the block and use the resulting value to perform ANB operation with the X00.

- When the X00 turns ON, the Y10 turns ON.
- To operate a ladder using the INV with the ANB, pay attention to the range of inversion. (*) Refer the cautions above.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
LD	X01
AND	X02
INV	
LD	X03
AND	X04
INV	
ORB	
ANB	
OUT	Y10

1.1.8 LDBT, LDBTI

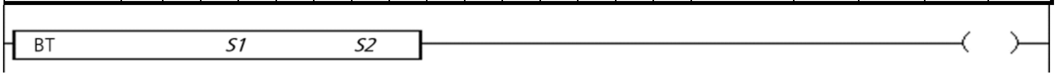
Supported PLC Series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	XP1A/R	BP	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

LDBT(I) refer to the ON/OFF status of bit data from a word device and use it as an operation result.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error (New)	Zero	Carry
LDBT(I)	*S1*	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-	3	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S1	Address of word device to refer from.
S2	Location of bit data in word device assigned to *S1*. ON/OFF status of specified bit data is referred.

LDBT, LDBTI

- LDBT and LDBTI refer to the status of bit data which is assigned to *S2*. *S2* indicates the location of bit data in word device assigned to *S1*.
- The referred status of bit data which is assigned to *S2* is used as a contact. With the referred status of bit data, LDBT starts contact 'A' operation and LDBTI starts contact 'B' operation.
- LDBT indicates a logical start and creates an ON/OFF execution condition based on the ON/OFF status of the specified bit data (*S2*) from word device (*S1*).
- LDBTI indicates a logical start and creates an ON/OFF execution condition based on the reverse of the ON/OFF status of the specified bit data (*S2*) from word device (*S1*).
- LDBT and LDBTI instructions are connected to the bus bar or used at the beginning of logic block.
- Value of *S2* is valid from 0 to 15. If value of *S2* is greater than or equal to 16, the location of bit is indicated by the remainder of n/16. (e.g. When *S2* is 17, it is equivalent to the case when *S2* is 1.)

Execution Condition

LDBT and LDBTI are executed every scan independently of the device status and operation result.

Operation Error

Error Flag (F110)

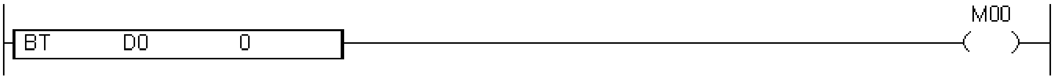
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

LDBT, LDBTI

The program that turns M00 ON when bit 0 of D0 is ON.

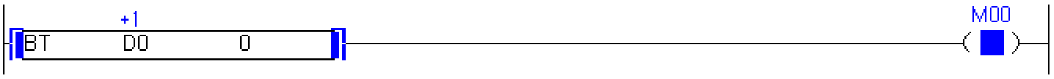
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LDBT	D0	0
OUT	M00	

- When D0 is “1” (BIN=0000 0000 0000 0001), the bit 0 is “1”. M00 is turned ON.



1.1.9 ANDBT, ANDBTI

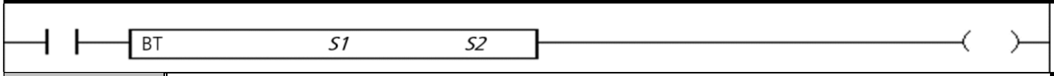
Supported PLC Series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	XP1A/R	BP	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

ANDBT(I) refer to the ON/OFF status of bit data from a word device and use it as an operation result.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error (New)	Zero	Carry
ANDBT(I)	*S1*	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-	3	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S1	Address of word device (D) to refer from.
S2	Location of bit data in word device assigned to *S1*. ON/OFF status of specified bit data is referred.

ANDBT, ANDBTI

- ANDBT and ANDBTI refer to the status of bit data which is assigned to *S2*. *S2* indicates the location of bit data in word device assigned to *S1*.
- The referred status of bit data which is assigned to *S2* is used as a contact. With the referred status of bit data, ANDBT starts contact 'A' operation and ANDBTI starts contact 'B' operation.
- ANDBT instruction performs AND operation based on the ON/OFF status of the specified bit data (*S2*) from word device (*S1*).
- ANDBTI instruction performs AND operation based on reverse of the ON/OFF status of the specified bit data (*S2*) from word device (*S1*).
- ANDBT and ANDBTI instructions cannot be connected to the bus bar or used at the beginning of logic block. ANDBT and ANDBTI instructions can be used when they are connected in series with LD(I), LDBT(I) etc.
- Value of *S2* is valid from 0 to 15. If value of *S2* is greater than or equal to 16, the location of bit is indicated by the remainder of n/16. (e.g. When *S2* is 17, it is equivalent to the case when *S2* is 1.)

Execution Condition

ANDBT and ANDBTI are executed every scan independently of the device status and operation result.

Operation Error

Error Flag (F110)

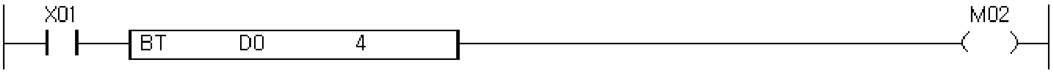
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

ANDBT, ANDBTI

The program that turns M02 ON when bit 4 of D0 is ON and X01 is ON.

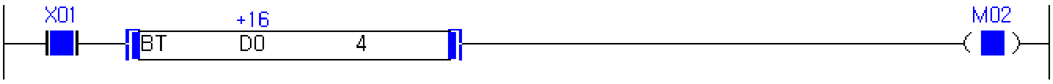
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X01	
ANDBT	D0	4
OUT	M02	

- When X01 is ON and D0 is “16”, (BIN=0000 0000 0001 0000), the bit 4 of D0 is “1”. M02 is turned ON.



1.1.10 ORBT, ORBTI

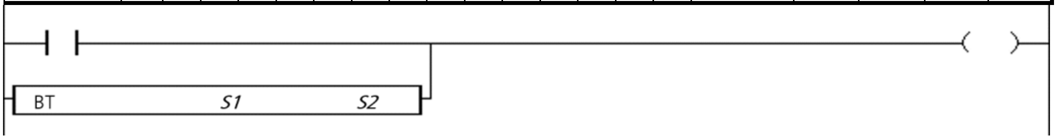
Supported PLC Series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	XP1A/R	BP	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

ORBT(I) refer to the ON/OFF status of bit data from a word device and use it as an operation result.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error (New)	Zero	Carry
ORBT(I)	*S1*	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-	3	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S1	Address of word device to refer from.
S2	Location of bit data in word device assigned to *S1*. ON/OFF status of specified bit data is referred.

ORBT, ORBTI

- ORBT and ORBTI refer to the status of bit data which is assigned to *S2*. *S2* indicates the location of bit data in word device assigned to *S1*.
- The referred status of bit data which is assigned to *S2* is used as a contact. With the referred status of bit data, ORBT starts contact 'A' operation and ORBTI starts contact 'B' operation.
- ORBT instruction performs OR operation based on the ON/OFF status of the specified bit data (*S2*) from word device (*S1*).
- ORBTI instruction performs OR operation based on reverse of the ON/OFF status of the specified bit data (*S2*) from word device (*S1*).
- ORBT and ORBTI instructions are connected to the bus bar or used at the beginning of logic block. But ORBT or ORBTI instructions can be used when they are arranged in a row with the LD(I), LDBT(I) etc.
- Value of *S2* is valid from 0 to 15. If value of *S2* is greater than or eqnal to 16, the location of bit is indicated by the remainder of n/16. (e.g. When *S2* is 17, it is equivalent to the case when *S2* is 1.)

Execution Condition

ORBT and ORBTI are executed every scan independently of the device status and operation result.

Operation Error

Error Flag (F110)

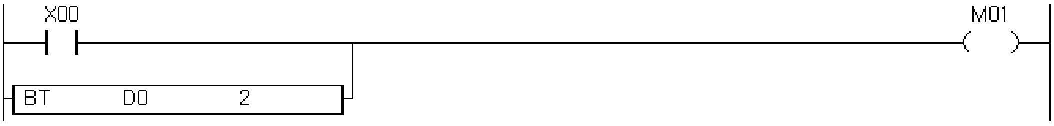
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

ORBT, ORBTI

The program that turns M01 ON when bit 2 of D0 is ON or X00 is ON.

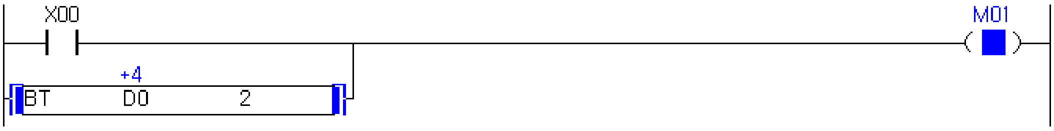
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
ORBT	D0	2
OUT	M01	

- When D0 is “4”, (BIN=0000 0000 0000 0100), the bit 2 of D0 is “1”. M01 is turned ON.



1.2 Connection Instruction

1.2.1 ANB, ORB

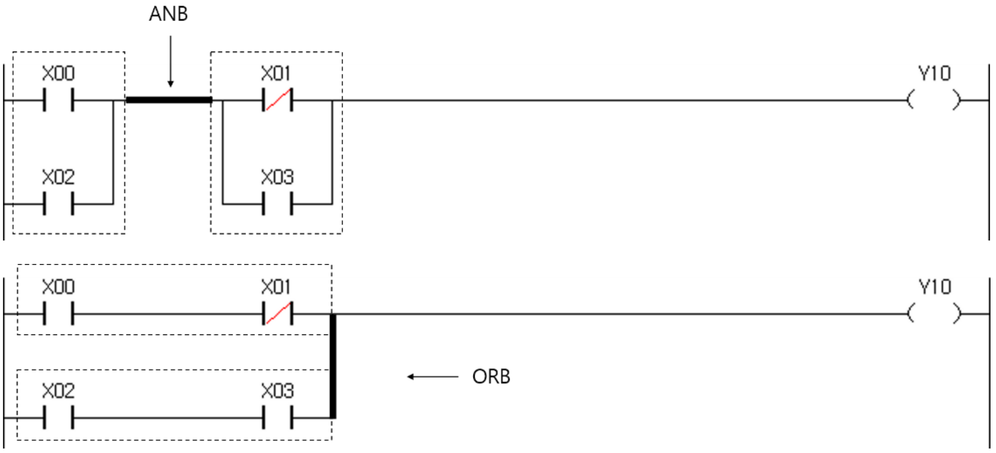
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

ANB is block series connection and ORB is block parallel connection.

Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ANB, ORB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



ANB

- ANB performs an AND operation of block A and block B, and uses the result as a new operation result.
- The symbol of ANB is not a contact symbol, but a connection symbol.
- In the instruction list(IL) mode, ANB can be written consecutively up to 15 instructions (16 blocks). If more ANBs are written consecutively, the operation may not be performed properly.

ORB

- ORB performs an OR operation of block A and block B, and uses the result as a new operation result.
- ORB performs parallel connections of circuit blocks with two or more contacts.
- Use OR or ORI for circuit blocks with single contact. ORB is not required to circuit blocks with single contact.
- The symbol of ORB is not a contact symbol, but a connection symbol.
- In the instruction list(IL) mode, ORB can be written consecutively up to 15 instructions(16 blocks). If more ORBs are written consecutively, the operation may not be performed properly.

Execution Condition

ANB and ORB are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the ANB and ORB instructions.

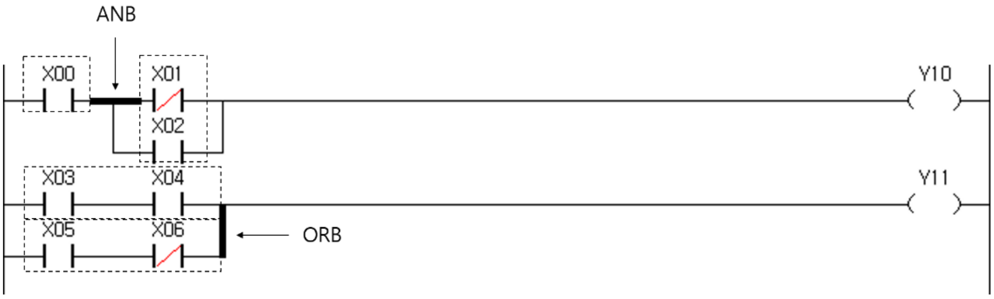
Program Example

ANB, ORB

In the following example, ANB, ORB execute as below.

- ANB performs an AND operation before the Y10, and ORB performs an OR operation before the Y11.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
LDI	X01
OR	X02
ANB	
OUT	Y10
LD	X03
AND	X04
LD	X05
ANDI	X06
ORB	
OUT	Y11

1.2.2 MPS, MRD, MPP

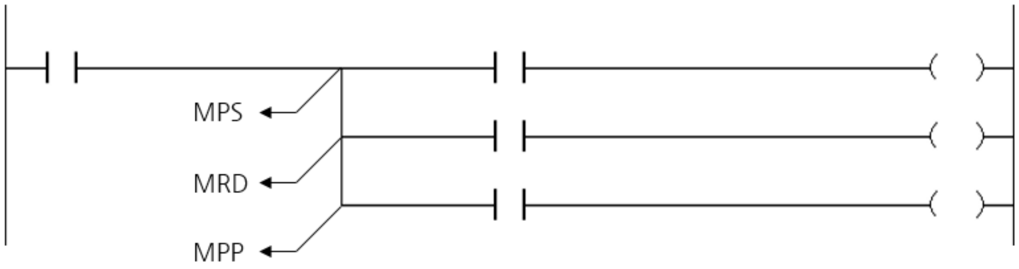
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

MPS stores the operation result, MRD reads the result stored by the MPS, and MPP clears the result stored by the MPS.

Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
MPS,MRD, MPP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



- MPS, MRD and MPP are not displayed in the ladder diagram.

MPS

- MPS stores the preceding operation result(ON/OFF) preceding of the MPS.
- Then performs the operation with the resulting value in the next step.
- MPS can be used successively up to 16 instructions.

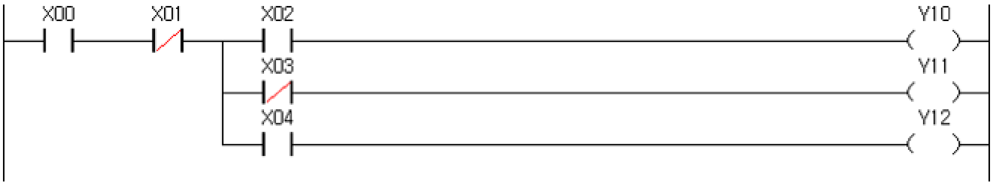
MRD

- MRD reads the operation result stored by the MPS.
- Then performs the operation with the resulting value in the next step.

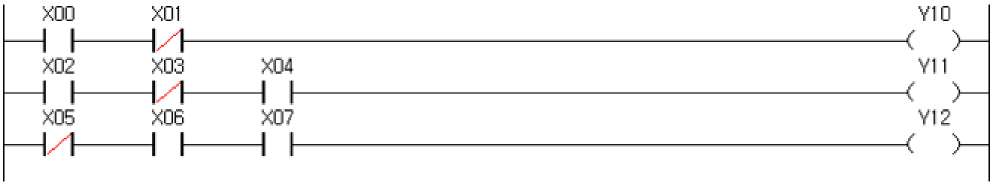
MPP

- MPP clears the operation result stored by the MPS and reset the operation.
- MPP reads the operation result stored by the MPS.
- Then performs the operation with the resulting value, in the next step.

- Following is a ladder diagram(LD) when using MPS, MRD and MPP.



- Following is a ladder diagram(LD) when not using MPS, MRD and MPP.



Execution Condition

MPS, MRD and MPP are executed every scan independently of the device status and operation result.

Operation Error

There is no operation error in the MPS, MRD and MPP instructions.

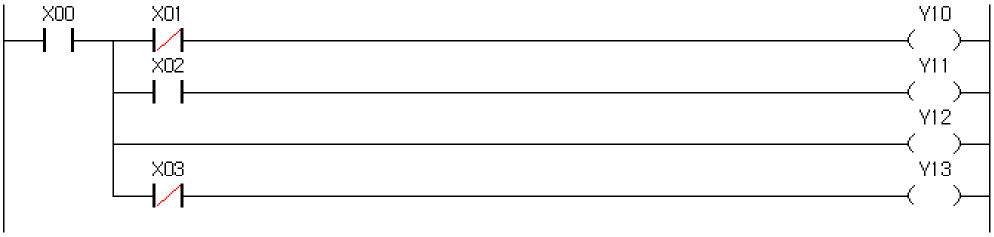
Program Example

MPS, MRD, MPP

In the following example MPS, MRD and MPP execute as below.

- MPS stores the operation result before the X01.
- MRD reads operation result stored by the MPS.
- MPP clears the operation which stored by the MPS.

Ladder Diagram (LD)



Instruction List (IL)

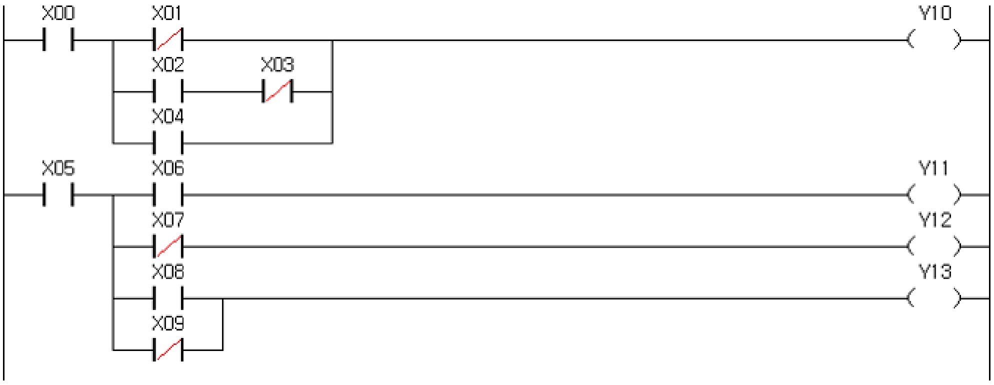
Instruction	Device
LD	X00
MPS	
ANDI	X01
OUT	Y10
MRD	
AND	X02
OUT	Y11
MRD	
OUT	Y12
MPP	
ANDI	X03
OUT	Y13

MPS, MRD, MPP and LD(I), AND(I), OR(I), ANB, ORB

In the following example, MPS, MRD, MPP and LD, LDI, AND, ANDI, OR, ORI, ANB, ORB execute as below.

- MPS stores the operation result before the X06.
- MRD reads operation result stored by the MPS.
- MPP clears the result which stored by the MPS.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
LDI	X01
LD	X02
ANDI	X03
ORB	
OR	X04
ANB	
OUT	Y10
LD	X05
MPS	
AND	X06
OUT	Y11
MRD	
ANDI	X07
OUT	Y12
MPP	
LD	X08
ORI	X09
ANB	
OUT	Y13

1.3 Out Instruction

1.3.1 OUT

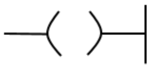
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

OUT outputs the operation result preceding the OUT instruction.

Instruction		Valid Device Type																Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Q	Constant		Error	Zero	Carry
OUT	*D*	O	-	O	O	O	-	-	-	O	-	-	-	-	-		-	1	-	-	-



D	Address of bit device. (Size=1 bit)

Execution Condition

OUT operates based on the results made until the OUT instruction. Use it in the position which outputs the resulting value.

Operation Error

There is no operation error in the OUT instruction.

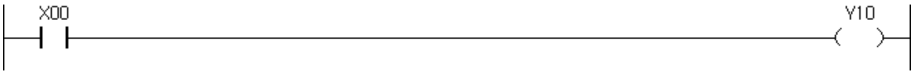
Program Example

OUT

In the following example, OUT executes as below.

- The Y10 outputs the resulting value of the X00.
- When the X00 turns ON, the Y10 turns ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X01
OUT	Y10

1.3.2 SET, RST

Supported PLC series

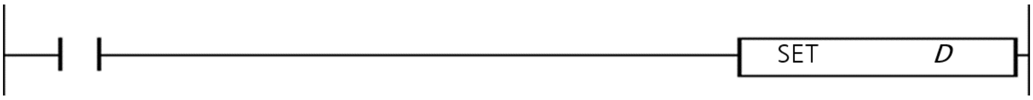
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

SET

SET turns ON the assigned bit device.

Instruction		Valid Device Type																Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Q	Constant		Error	Zero	Carry
SET	*D*	O	-	O	O	O	-	-	-	O	-	-	-	-	-	-	-	1	-	-	-



D	Address of bit device. (Size = 1 bit)

- The SET instruction turns ON the assigned bit device, when the SET execution condition turns ON.
- Once the assigned bit device turns ON, the status does not change even though the SET input turns OFF.

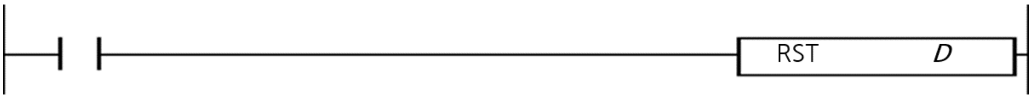
⚠ Differences between OUT and SET

- The OUT turns ON the assigned bit device when the execution condition turns ON, and turns OFF the device when the execution condition turns OFF.
- The SET turns ON the bit device, respectively, when the execution condition turns ON. Status of the bit device does not change even when the execution condition turns OFF.

RST

RST resets the status of the assigned bit device.

Instruction		Valid Device Type																Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Q	Constant		Error (Old)	Zero	Carry
RST	*D*	O	-	O	O	O	-	O	O	O	-	-	-	-	-	-	-	1	O	-	-



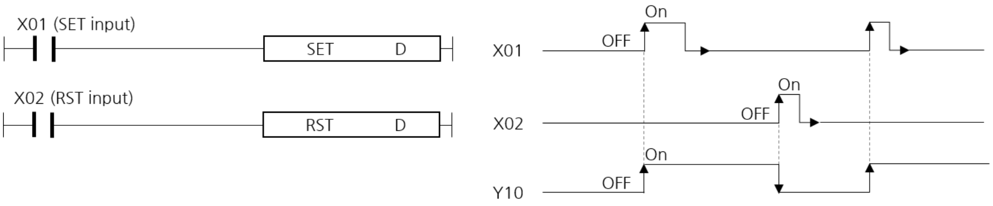
D	Address of bit device.

- The RST instruction turns OFF the assigned bit device when execution condition turns ON.
- Once the assigned bit device turns OFF, the status does not change even though the RST input turns OFF.

Device	Status
Y, M, L	The contact and coil are turned OFF.
T, C	Present value is set to 0, and the contact and coil are turned OFF.

⚠ Difference between RST and OUT

- OUT turns ON the assigned bit device when the execution condition turns ON, and turns OFF the device when the execution condition turns OFF.
- RST turns OFF the assigned bit device, respectively, when the execution condition turns ON. Status of the device does not change when the execution condition turns OFF.



Execution Condition

SET and RST are executed every scan independently of the device status and operation result.

Operation Error

SET

There is no operation error in SET instruction.

RST

Error flag of RST instruction turns ON when the instruction is processed with CPU such as BP, XPnA/1R, CP3A/B/P/U and CP4A~D/U.

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type).

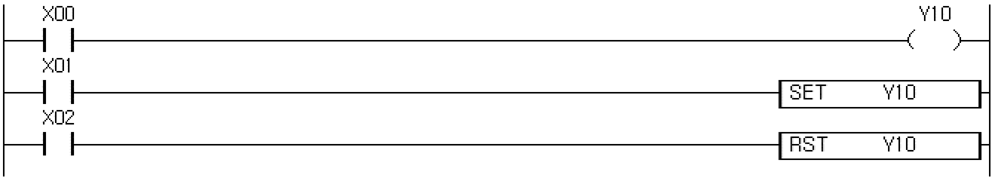
Program Example

SET, RST

In the following example, SET and RST execute as below.

- When the X01 turns ON, the Y10 turns ON.
- When X02 turns ON, the Y10 turns OFF.

Ladder Diagram (LD)



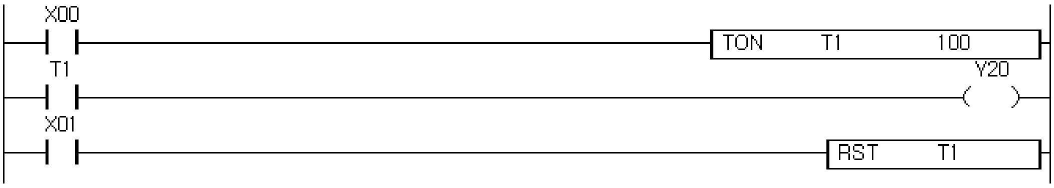
Instruction List (IL)

Instruction	Device
LD	X00
OUT	Y10
LD	X01
SET	Y10
LD	X02
RST	Y10

RST

- When X00 turns ON, the current value of T1 increases. When the current value reaches “100” (= 10 sec), T1 turns ON. When T1 turns ON, Y20 turns ON.
- If X01 turns ON, RST instruction executes. Therefore, the current value becomes “0” and T1 is turned OFF.

Ladder Diagram (LD)



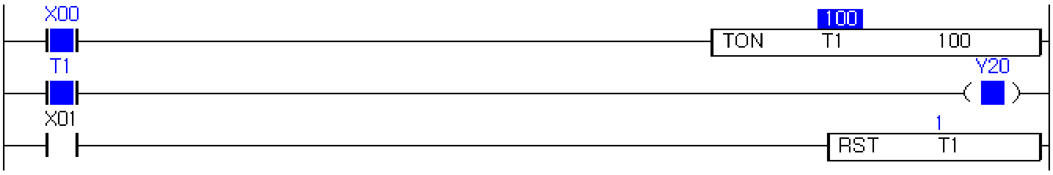
Instruction List (IL)

Instruction	Device	
LD	X00	
TON	T1	100
LD	T1	
OUT	Y20	
LD	X01	
RST	T1	

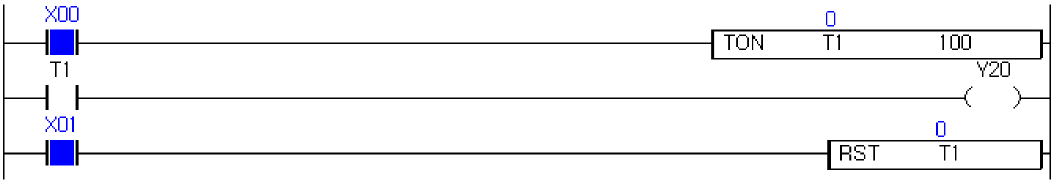
The program operates as following:

- When X00 is ON, the program starts timer device T1. The current value of T1 increases. When it reaches 100, T1 is turned ON. When T1 is ON, it turns ON Y20.
- If X00 is turned OFF before the current value of timer reaches the setting value, the

current value is reset.



- When X01 is ON, RST instruction operates and the current value of timer becomes “0”.



1.3.3 PLS, PLF

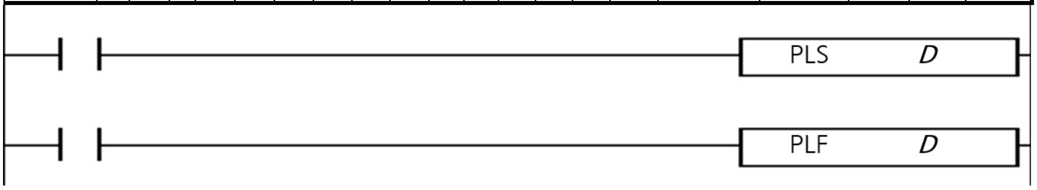
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

PLS turns ON the assigned device for 1 scan when execution condition goes from OFF to ON, and PLF turns ON the assigned device for 1 scan when execution condition goes from ON to OFF.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PLS, PLF	*D*	O	-	O	O	O	-	-	-	-	-	-	-	-	-	-	1	-	-	-



D	Address of bit device.

PLS

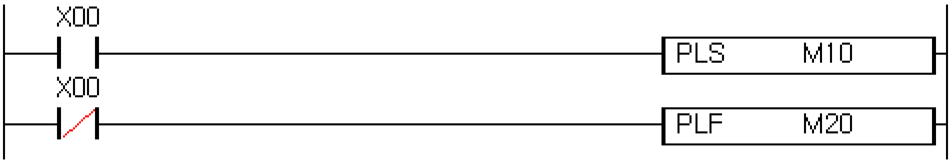
- When the execution condition goes from OFF to ON, the device assigned to D turns ON for 1 scan.
- When the execution condition goes from OFF to OFF, ON to ON, and ON to OFF, the device assigned by *D* does not turn ON.

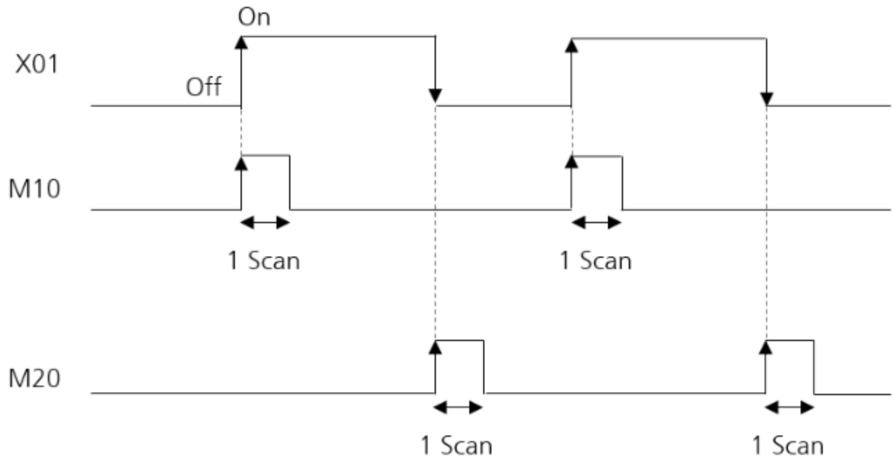
⚠ Even RUN/STOP key switch is changed (STOP→RUN) after the execution of PLS instruction, the instruction will not be executed again.

PLF

- When the execution condition goes from ON to OFF, the device assigned by *D* turns ON for 1 scan.
- When the execution condition goes from OFF to OFF, OFF to ON, and ON to ON, the device assigned by *D* does not turn ON.

⚠ Even RUN/STOP key switch is changed (RUN→STOP) after the execution of PLF instruction, the instruction will not be executed again.





Execution Condition

(*) Refer the function for execution condition of PLS and PLF.

Operation Error

There is no operation error in the PLS and PLF instructions.

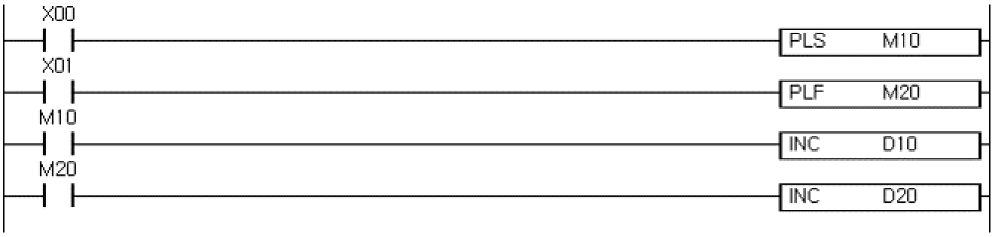
Program Example

PLS, PLF

In the following example, PLS and PLF execute as below.

- When the X00 goes from OFF to ON, then M10 turns ON for 1 scan and value of D10 increases by one.
- When the X01 goes from ON to OFF, Then M20 turns ON for 1 scan and value of D20 increases by one.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
PLS	M10
LD	X01
PLF	M20
LD	M10
INC	D10
LD	M20
INC	D20

1.4 Master Control Instruction

1.4.1 MC, MCR

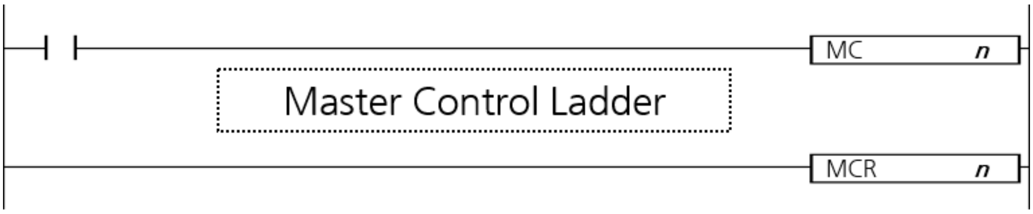
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

The MC sets master control and the MCR resets the master control.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
MC,MCR	*n*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	1	-	-	-



n	Nesting number (0 ~ 7)

MC

- MC indicates the start of Master Control area.
- By that, MC allows the sequence program to perform circuit switching efficiently.
- When the execution condition of the MC is ON, master control starts operation.
- When master control starts, program is executed from MC to MCR which has identical number.
- When execution condition of the MC is OFF, master control does not start operation and MCR instruction is not executed.
- The MC instructions can be used in nesting in the program. Different master control areas are distinguished by nesting number (*n*).
- MC~MCR can be nested up to 7 levels.
- If MC~MCR instructions are nested in other MC~MCR, the nesting number (*n*) is available from lower to higher with MC instruction and higher to lower with MCR instruction.
- (*) Refer to Program Example (1) below.
- When the nesting is composed in reverse, then compile error occurs.

MCR

- MCR is an instruction to end the Master Control area.
- MCR recovers from the master control and starts the process of scan program from the following step.
- Identical nesting number of MC and MCR instruction are used as a set.

- MCR with lowest nesting number can terminate all master controls when:
  - MCR instructions are nested in one place.
  - There are MC instructions with higher nesting number (*n*) in the MC~MCR area with lowest nesting number (*n*).
    (*) Refer to Program Example (2) below

Execution Condition

⚠ When a ladder with master control contains instructions that do not require contact instruction (such as FOR to NEXT instructions), the CPU module executes these instructions regardless of the ON/OFF status of the MC instruction execution command.

Operation Error

Compile error occurs under following circumstance:

- When the number of nests exceeds 8.

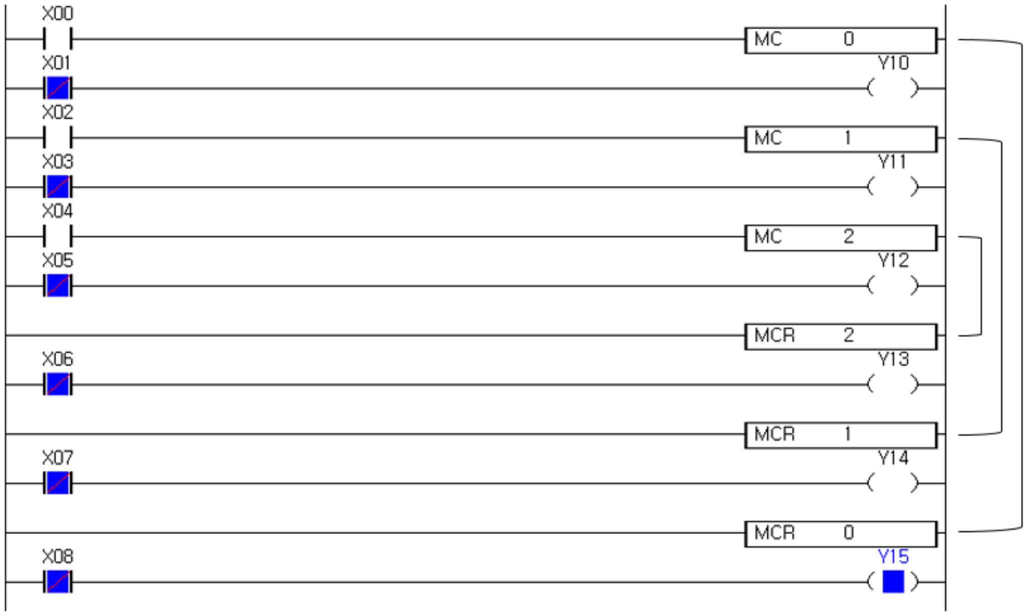
Program Example

(1) MC, MCR

In the following example, MC and MCR execute as below.

- MC 0 ~ MCR 0
  a) When the X00 turns ON, then MC 0 turns ON.
  b) When MC 0 turns ON, then Y10 and the Y14 turn ON.
- MC 1 ~ MCR 1
  a) After MC 0 is ON, when the X02 turns ON then MC 1 turns ON.
  b) When the MC 1 turns ON, then the Y11 and the Y13 turns ON.
- MC 2 ~ MCR 2
  a) After MC 0 and 1 is ON, when the X04 turns ON then MC 2 turns ON.
  b) When MC 2 turns ON, then the Y12 turns ON.
- The execution condition of the Y15 is not related to MC and MCR instructions.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
MC	0
LDI	X01
OUT	Y10
LD	X02
MC	1
LDI	X03
OUT	Y11
LD	X04
MC	2
LDI	X05
OUT	Y12
MCR	2
LDI	X06
OUT	Y13
MCR	1
LDI	X07
OUT	Y14
MCR	0
LDI	X08
OUT	Y15

(2) MC, MCR: Extraordinary but shortcut usage

In the following example, MCs and MCR execute as below.

- When MCR instructions are used in a program, MCR with the lowest nesting number can end all MC instructions.
- In the following example, MCR 0 also performs as MCR 1 and MCR 2.
- MC 0 ~ MCR 0

  a) When the X00 turns ON, then MC 0 turns ON.
  b) When MC 0 turns ON, then Y10 turns ON.

- MC 1 ~ MCR 0

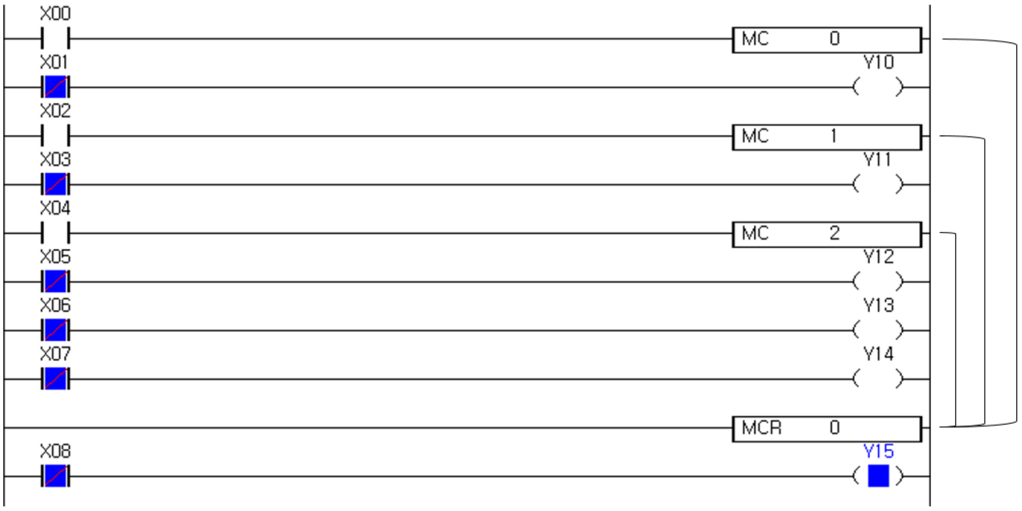
  a) After MC 0 is ON, when the X02 turns ON then MC 1 turns ON.
  b) When the MC 1 turns ON, then the Y11 turns ON.

- MC 2 ~ MCR 0

  a) After MC 0 and 1 is ON, when the X04 turns ON then MC 2 turns ON.
  b) When MC 2 turns ON, then the Y12, Y13, and Y14 turn ON.

- The execution condition of the Y15 is not related to MC and MCR instructions.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
MC	0
LDI	X01
OUT	Y10
LD	X02
MC	1
LDI	X03
OUT	Y11
LD	X04
MC	2
LDI	X05
OUT	Y12
LDI	X06
OUT	Y13
LDI	X07
OUT	Y14
MCR	0
LDI	X08
OUT	Y15

1.5 Program Branch Instruction

1.5.1 JMP, JMPP, JME

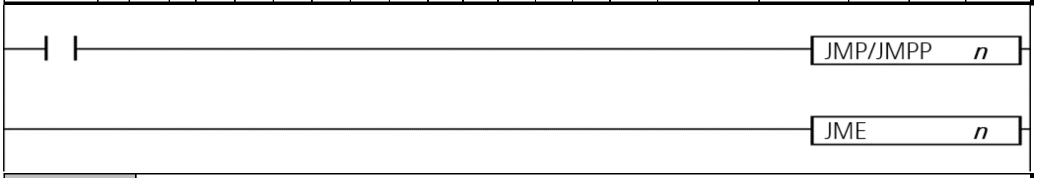
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

When the execution condition of JMP is ON, program jumps to the assigned JME within the program.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
JMP(P), JME	*n*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	2	-	-	-



n	Branch ID (0 ~ 127)

JMP

- When the execution condition of JMP is ON, program jumps to the next step of the JME with identical branch ID *n*.
- When the execution condition of JMP is OFF, JMP is not executed and program does not jump.
- When JMP is executed, all instructions between JMP and JME are not executed.
- Multiple JMP *n* can be placed before JME *n*.

JMPP

- When the execution condition of JMPP is ON, program jumps to the next step of the JME with identical branch ID *n* for 1 scan.
- When the execution condition of JMPP is OFF, JMPP is not executed and program does not jump.
- When JMPP is executed, all instructions between JMPP and JME are not executed for 1 scan.

JME

- JMP(P) and JME are used in pairs.
- Multiple JMP(P) *n* can make a pair with single JME *n*.
- Multiple JME with identical branch ID *n* cannot exist within a same program.

Execution Condition

The execution conditions of JMP(P) and JME are as below.

- Branch ID *n* of JMP(*n*) is available from 0 to 127.
- Branch ID *n* of JMP(*n*) cannot be duplicated with the number used at SBRT instruction.
- When Branch ID *n* of JMP and JME are same, JMP should be placed before JME.

Operation Error

Compile error occurs under following circumstance:

- JMP(P)~JME are used in SBRT~RET block.

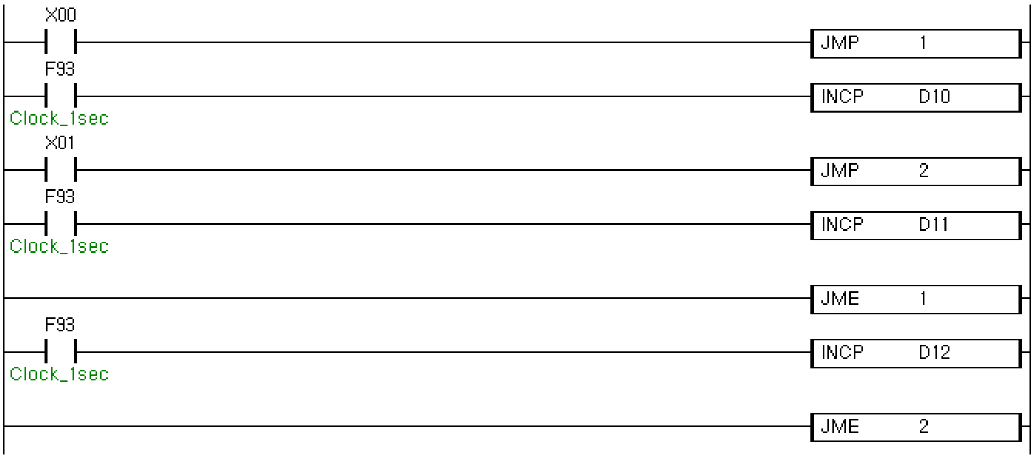
Program Example

(1) JMP, JME

In the following example, JMP(P) and JME execute as below.

- When PLC goes from STOP to RUN, the value of D10, D11 and D12 increases by 1.
- When X00 is turned ON, INCP instruction stops processing even the execution condition is met. Therefore, the value of D10 and D11 does not increase. But, the value of D12 increases by 1.
- When X00 is turned OFF and X01 is turned ON, INCP instruction stops processing even the execution is met. Therefore, the value of D11 and D12 does not increase. But, the value of D10 increases by 1.

Ladder Diagram (LD)



Instruction List (IL)

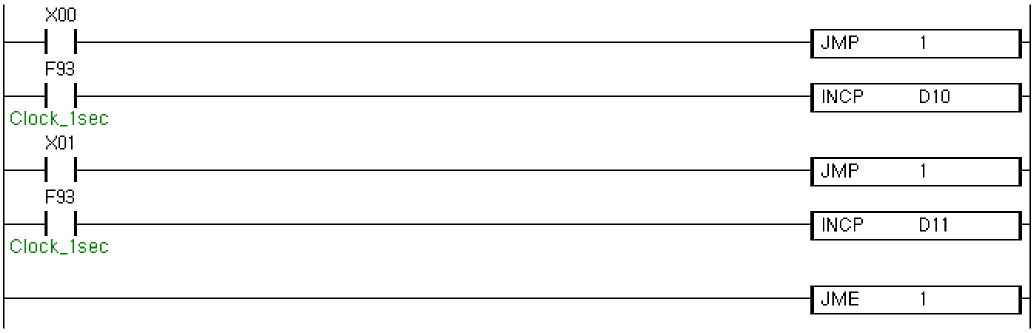
Instruction	Device
LD	X00
JMP	1
LD	F93
INC	D10
LD	X01
JMP	2
LD	F93
INC	D11
JME	1
LD	F93
INC	D12
JME	2

(2) JMP, JME

In the following example, multiple JMP instructions make a pair with single JME instruction.

- When PLC goes from STOP to RUN, INCP instruction operates and the value of D10 and D11 increases by 1.
- When X00 is turned ON, INCP instruction stops processing even the execution condition is met. Therefore, the value of D10 and D11 does not increase.
- When X00 is turned OFF and X01 is turned ON, INCP instruction which is located before the second JMP instruction operates. Therefore, the value of D10 increases by 1.
- On the contrary, INCP instruction located after the second JMP instruction does not operate. Therefore, the value of D11 does not increase.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
JMP	1
LD	F93
INC	D10
LD	X01
JMP	1
LD	F93
INC	D11
JME	1

1.5.2 CALL, CALLP, SBRT, RET

Supported PLC series

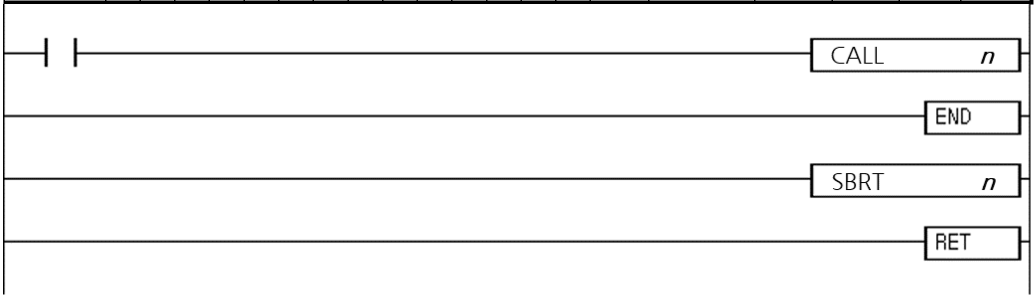
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

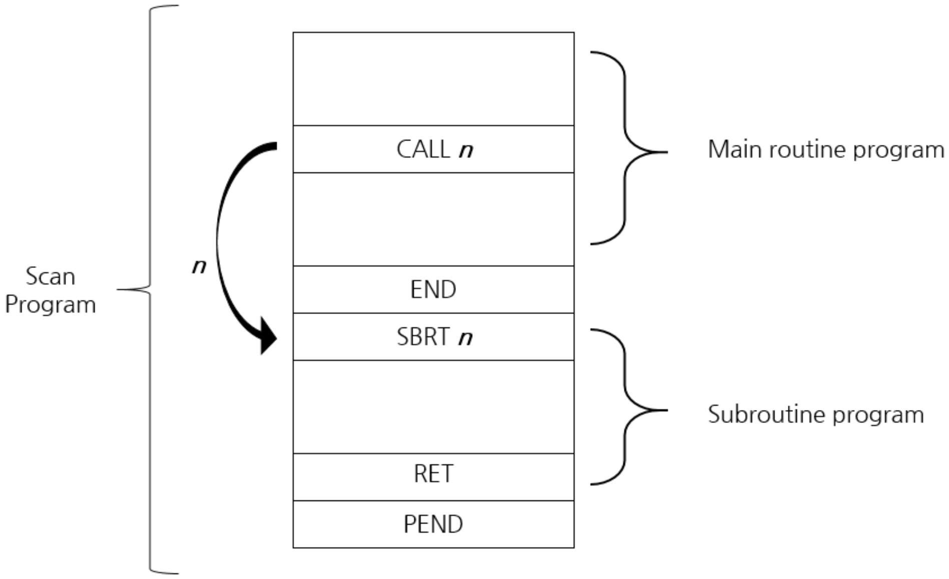
CALL and CALLP call and execute the assigned subroutine program which is located between SBRT and RET in the same program.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
CALL(P)	*n*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	2	-	-	-

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error (Old)	Zero	Carry
SBRT	*n*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	2	O	-	-
RET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	



n	Subroutine number (Branch ID, 0 ~ 127)



CALL, CALLP

- CALL(P) in a scan program call the subroutine program of SBRT *n* ~RET inside the same program.
- When execution condition is ON, CALL(P) call a subroutine program with assigned subroutine number.
- When subroutine program operates, the scan program does not operate until the process of subroutine program ends.
- When the operation of subroutine is completed, program continues execution after CALL or CALLP instruction.
- Branch ID *n* of CALL and CALLP is available from 0 to 127.
- "CALL(P) *n*" can be used multiple times in a same program.

SBRT, RET

- The subroutine program between SBRT and RET must be placed after the END instruction of scan program which is located in the same scan program.
- Up to 128 subroutines can be defined in one program.
- Calling another SBRT in SBRT (nesting CALL) is available up to 16 times.
- Error occurs when following conditions are met:

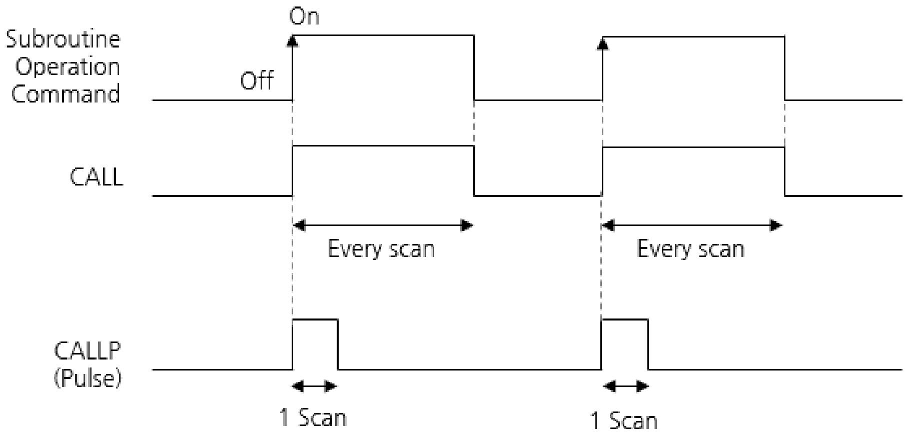
  a) Subroutine number (Branch ID) *n* exceeding the range of 0~127.

  b) CALL *n* exists, but SBRT *n* does not exist.

  c) SBRT and RET is not used as a set, and only one of them is used.

  d) There are multiple "SBRT *n*" are used in a same program.

Execution Condition



- If subroutine program is not called by CALL(P), then output coil in SBRT~RET is not executed even when the execution condition is ON.
- Subroutine number (SBRT *n* ~RET) cannot be duplicated with branch ID *n* of JMP.

Operation Error

Compile error occurs under following circumstances:

- END instruction is located between SBRT instruction and RET instruction.
- RET instruction is located before CALL(P) or SBRT instruction. (When there is only one subroutine)
- Calling another SBRT in SBRT (nesting CALL) exceeded 16 times.
- JMP(P)~JME are located between SBRT~RET.

SBRT

Error flag of SBRT instruction turns ON when the instruction is processed with CPU such as BP, XPnA/1R, CP3A/B/P/U and CP4A~D/U.

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D (Range of device D depends on CPU type).

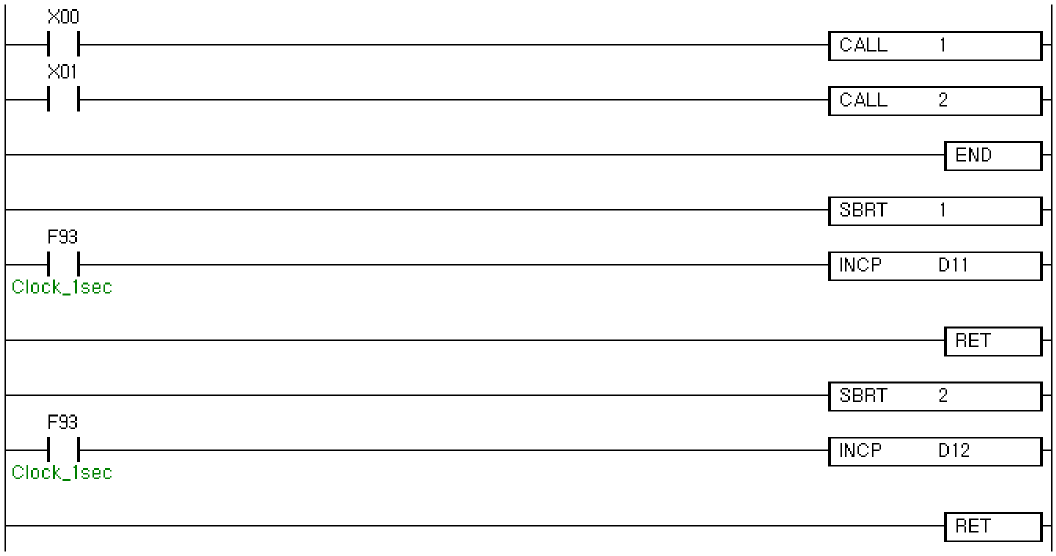
Program Example

CALL, SBRT, RET

In the following example, CALL(P) and SBRT~RET execute as below.

- When PLC goes from STOP to RUN, the value of D11 and D12 does not increase even the operation condition is ON.
- When X00 is turned ON, CALL instruction calls SBRT 1~RET. Then the value of D11 increases by 1. The value of D12 does not increase.
- When X01 is turned ON, CALL instruction calls SBRT 2~RET. Then the value of D12 increases by 1. Since X00 is not turned OFF, the value of D11 keeps increasing.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
CALL	1
LD	X01
CALL	2
END	
SBRT	1
LD	F93
INCP	D11
RET	
SBRT	2
LD	F93
INCP	D12
RET	

1.5.3 ECALL, ECALLP

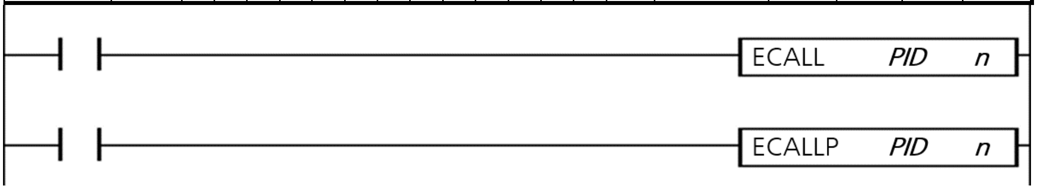
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

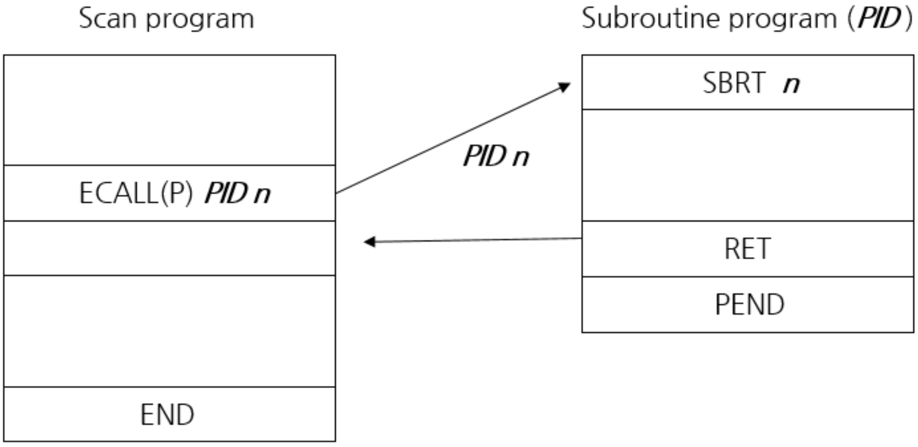
ECALL(P) instructions call and execute the subroutine program in the external program.

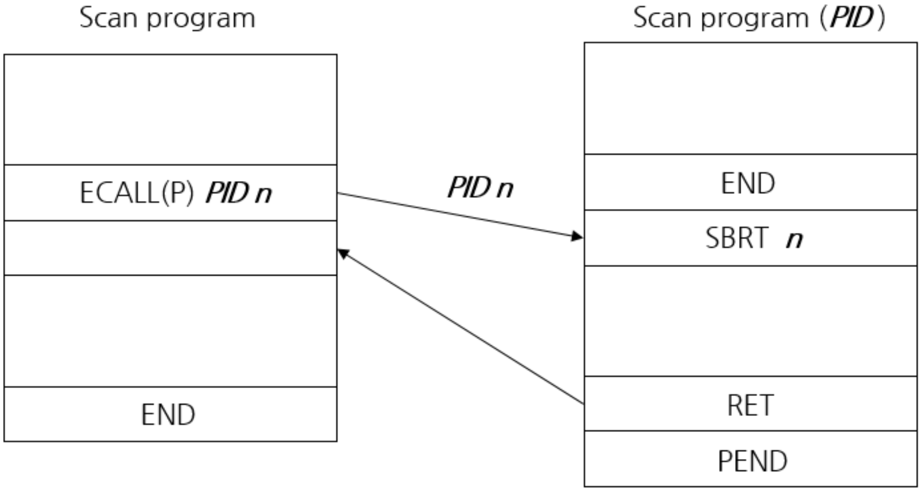
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ECALL(P)	*PID*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	3	-	-	-
	n	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O				



PID	Program ID of external program
n	Subroutine number (0 ~ 127)

ECALL, ECALLP



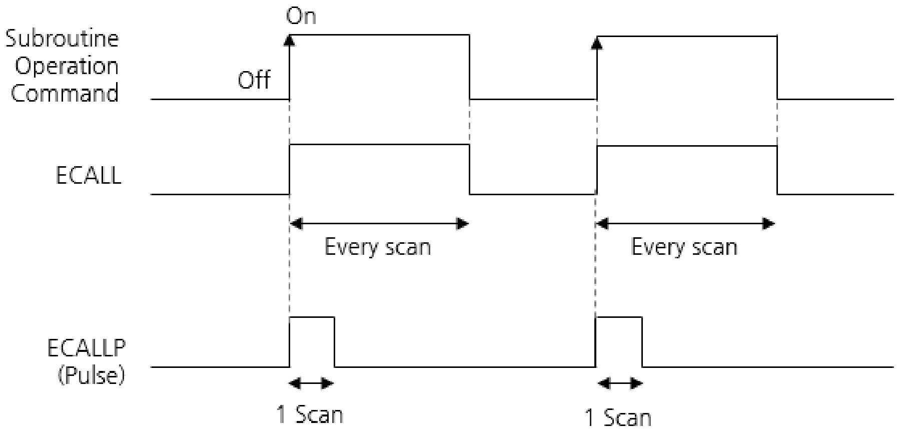


- ECALL and ECALLP call the subroutine program of SBRT~RET which is located in the external program.
- External program can be scan program and subroutine program.
- When execution condition is ON, ECALL and ECALLP call the subroutine program with the assigned number (*PID*, *n*).
- When the subroutine is completed, program continues execution after ECALL or ECALLP.
- Nesting of ECALL and ECALLP is available up to 16 times.
- Subroutine number *n* of ECALL and ECALLP is available from 0 to 127.
- "ECALL(P) *PID* *n*" can be used multiple times in a same program.
- When "SBRT *n*" is duplicated in the external program file, a compile error occurs.

⚠ Difference between CALL(P) and ECALL(P)

- CALL(P) instruction calls the subroutine program which is located in the same program.
- ECALL(P) instruction calls the subroutine program which is located external program.

Execution Condition



- If subroutine program is not called by ECALL(P), then outputs in SBRT~RET are not executed even if the execution conditions are ON.
- Subroutine number of SBRT *n* ~RET cannot be duplicated with the branch ID *n* of JMP within a same program.

Operation Error

Compile error occurs under the following circumstances:

- END instruction is located between SBRT instruction and RET instruction.
- Nesting of ECALL(P) exceeded 16 times.
- JMP(P)~JME are located between SBRT~RET block.

Program Example

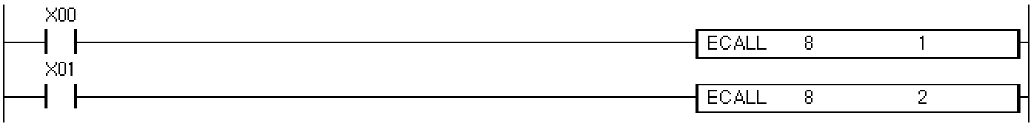
ECALL, ECALLP

In the following example, ECALL(P) and SBRT~RET execute as below.

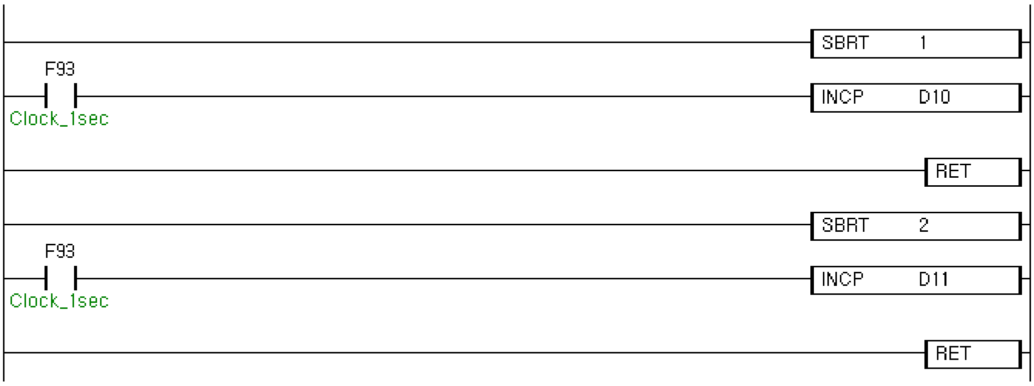
- When X00 turns ON, ECALL instruction calls SBRT 1~RET of subroutine program (PID: 8).
- Then the value of D10 increases by 1.
- When X01 turns ON, ECALL instruction calls SBRT 2~RET of subroutine program (PID: 8).
- Then the value of D11 increases by 1.

Ladder Diagram (LD)

- Scan program



- Subroutine program (PID: 8)



Instruction List (IL)

- Scan program

Instruction	Device	
LD	X00	
ECALL	8	1
LD	X01	
ECALL	8	2

- Subroutine program

Instruction	Device
SBRT	1
LD	F93
INCP	D10
RET	
SBRT	2
LD	F93
INCP	D11
RET	

1.6 Termination Instruction

1.6.1 END

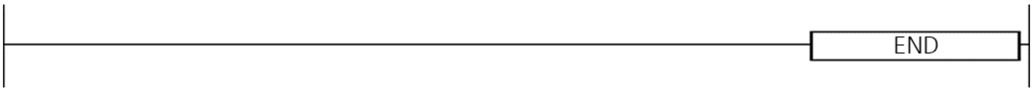
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

END instruction ends scan program.

Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
END	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	2	-	-	-



END

- END instruction terminates a scan program where END instruction is located.
- Instructions located after END instruction are not executed except SBRT~RET.
- END instruction should be used at least once in a scan program.
- END instruction can be located anywhere except the location between instructions which should make a pair.
  e.g.) END cannot be located between FOR~NEXT, SBRT~RET, JMP(P)~JME.
- When END instruction is executed, PLC CPU stops the scan program. If there are other scan programs which are not executed yet, PLC CPU starts the next scan program. The execution order of scan program is decided by the PID number. (Lower PID number → Higher PID number)

Execution Condition

No operation condition is needed to execute this instruction.

Operation Error

Compile error occurs when there is no END instruction in the following programs:

- Scan Program
- Initialization Program (HOT/COLD)

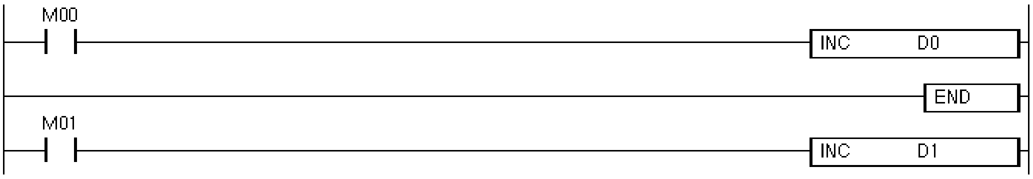
Program Example

END

In the following example, END operates as below.

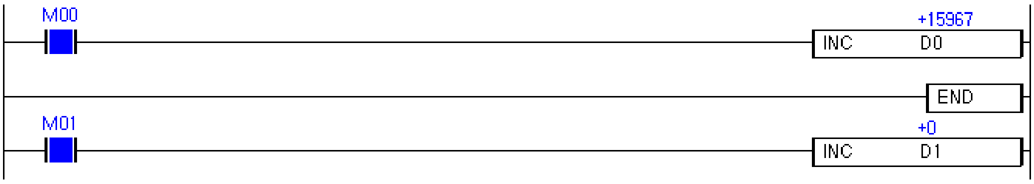
- When PLC is turned ON, the process starts from Step 0.
- When M00 is turned ON, INC instruction operates and the value of D0 increases.
- When M01 is turned ON, INC instruction does not operate and the value of D1 does not increase since it is located after END.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	M00
INC	D0
END	
LD	M01
INC	D1



1.6.2 CEND, CENDP

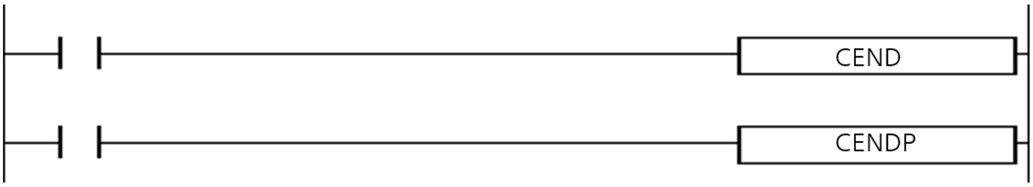
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

CEND(P) instruction ends scan program conditionally.

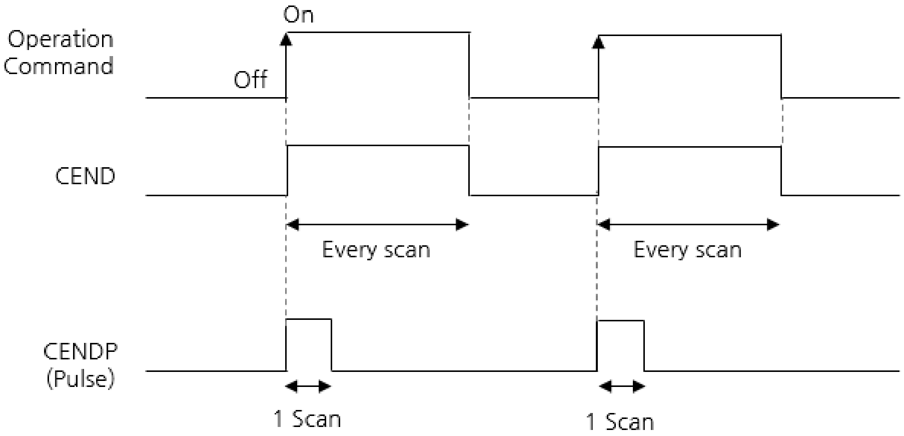
Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
CEND(P)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



CEND, CENDP

- CEND(P) instructions terminate a scan program when the execution condition is met.
- CEND(P) instructions can be located anywhere even the location between instructions which should make a pair.
- When CEND(P) are executed, instructions located after CEND(P) instructions are not executed.
- Difference between CEND(P) and END is an existence of execution condition.
- Even though CEND(P) instructions exist, END instruction must be located at the end of scan program.

Execution Condition



Operation Error

Compile error occurs under following circumstances:

- When CEND(P) instruction is the only termination instruction of the program.

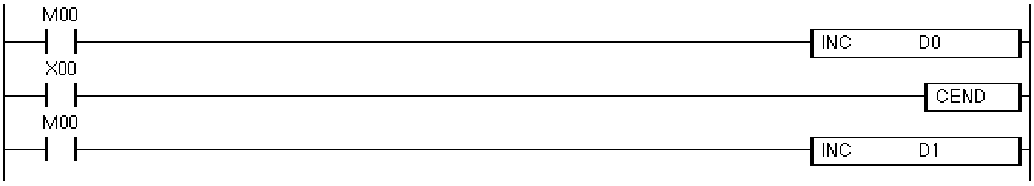
Program Example

CEND, CENDP

In the following example, CEND operates as below.

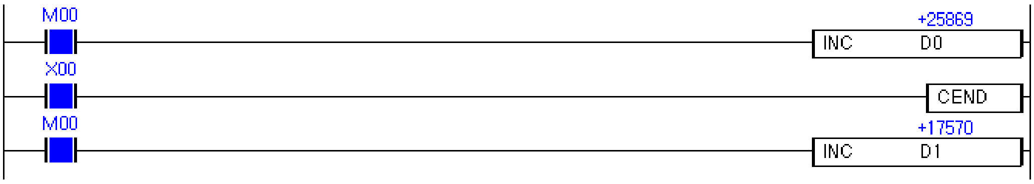
- When PLC is turned ON, the process starts from Step 0.
- When M00 is turned ON, the values of D0 and D1 increases.
- When X00 is turned ON, CEND instruction is executed. The instruction after the CEND are not executed. The value increment of D1 stops.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	M00
INC	D0
LD	X00
CEND	
LD	M00
INC	D1



1.6.3 PEND

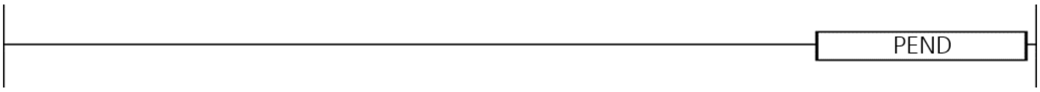
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

PEND indicates the end of the program.

Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PEND	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



PEND

- PEND instruction indicates the end of the program.
- PEND must be located at the end of program.
- PEND cannot be located among the other instructions. If you need to end a program in the middle of process, use END instruction.
- PEND is a default instruction when a new program (Scan program, subroutine program, periodic interrupt program, HOT/COLD initialization program) is created.
- You cannot manually input or delete this instruction.

Execution Condition

There is no execution condition of PEND instruction.

Operation Error

There is no operation error of PEND instruction.

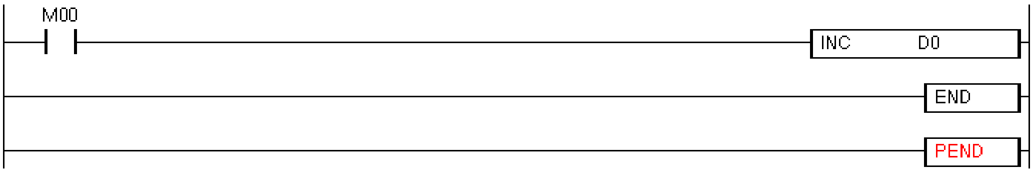
Program Example

PEND

In the following example, PEND operates as below.

- When PLC is turned ON, the process starts from step 0 of the first scan program which has the lowest PID number.
- When M00 is turned ON, INC instruction operates and the value of D0 increases by 1.
- The process reaches PEND. After the process reaches PEND, 1 scan has ended.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	M00
INC	D0
END	
PEND	

1.6.4 STOP

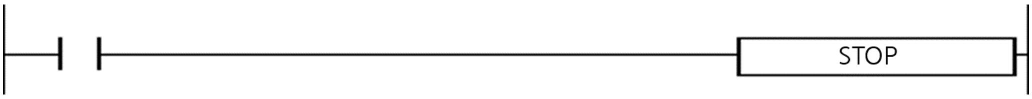
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

STOP instruction stops the process of PLC.

Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
STOP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



STOP

- When execution condition of STOP instruction is ON, the instruction resets output Y and stops PLC process.
- If this instruction is executed, it operates same as changing RUN/STOP key switch from RUN to STOP.
- To restart PLC process, change RUN/STOP key switch RUN→STOP→RUN.
- Do not locate STOP instruction in interrupt program, subroutine program or between FOR~NEXT instruction.

⚠ **Operation of STOP instruction according to the PLC**

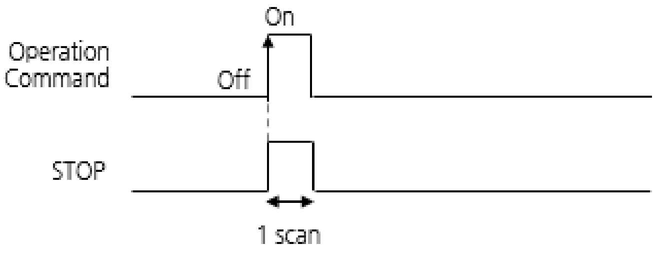
When STOP instruction is operated with CPU such as BP, XPnA/1R, CP3A/B/P/U and P4A~D/U:

- All scan programs are executed. Then, CPU mode changes.

When STOP instruction is operated with PLC-S, XPnB/E/F and CPnE/F:

- The scan program that includes STOP instruction is executed until the END instruction. Then, CPU mode changes and no more scan program is executed.

Execution Condition



Operation Error

Compile error occurs under following circumstance:

- When STOP instruction is located in the interrupt program, subroutine program and FOR~NEXT instruction.

Program Example

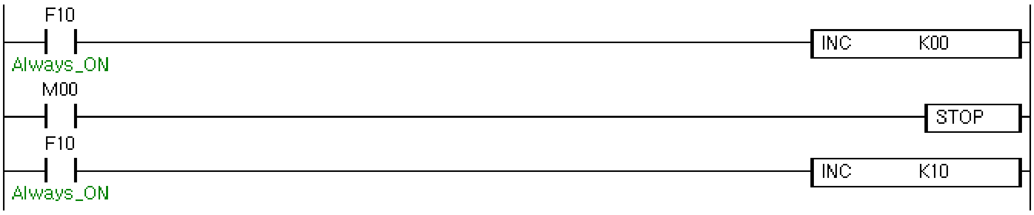
STOP

In the following example, STOP operates as below. (CPU: PLC-S)

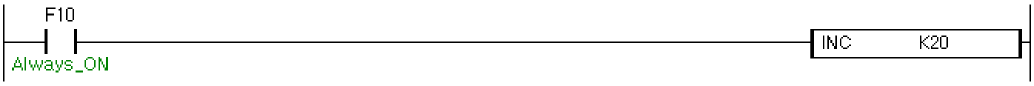
- INC instructions are executed and the values of K00, K10 and K20 increase.
- When M00 is turned ON, STOP instruction is executed.
- Scan ends at the end of the scan program where STOP instruction is executed. (PID: 0)
- When STOP executes, the INC instructions in the same scan program are executed. But, the other INC instruction in the other scan program (PID: 1) is not executed.
- Thus, after the execution of STOP instruction, the value of K20 is 1 less than those of K00 and K10.
- If CPU is BP, XPnA/1R, CP3A/B/P/U and CP4A~D/U, all scan programs are scanned. For this reason, the values of K00, K10 and K20 would be same.

Ladder Diagram (LD)

- Scan program (PID: 0)



- Scan Program (PID: 1)



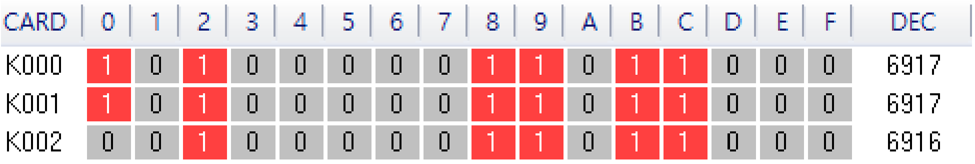
Instruction List (IL)

- Scan program (PID: 0)

Instruction	Device
LD	F10
INC	K00
LD	M00
STOP	
LD	F10
INC	K10

- Scan Program (PID: 1)

Instruction	Device
LD	F10
INC	K20



CARD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	DEC
K000	1	0	1	0	0	0	0	0	1	1	0	1	1	0	0	0	6917
K001	1	0	1	0	0	0	0	0	1	1	0	1	1	0	0	0	6917
K002	0	0	1	0	0	0	0	0	1	1	0	1	1	0	0	0	6916

1.6.5 INITEND

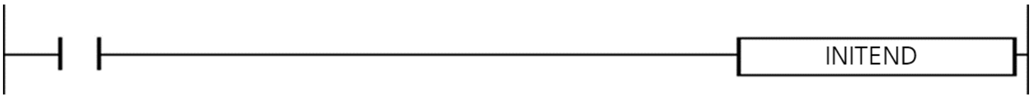
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

INITEND ends the initialization program (HOT/COLD) and runs other normal scan programs.

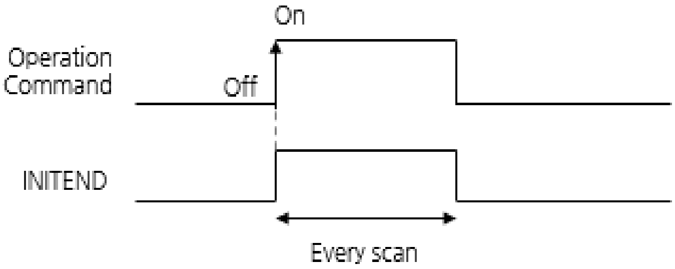
Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
INITEND	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



INITEND

- When INITEND instruction is executed, it terminates the initialization program (HOT/COLD) and runs other normal scan programs.
- Once INITEND instruction is executed, process does not operate initialization program again except when CPU mode changes (STOP→RUN) or power resets.
- To end initialization program and run other normal scan programs, INITEND instruction should be executed. If this instruction is not executed, only initialization program runs infinitely.
- When INITEND instruction is used in other program will cause a compile error.

Execution Condition



⚠ Be aware when using the contact. At 1st scan, the CPU cannot detect the rising pulse if the contact is already ON.

⚠ When you use CPU such as BP, XPnA/1R, CP3A/B/P/U and CP4A~D/U, F12 flag (ON at first scan only) does not go ON in case of Hot Restart. It is not recommended to use F12 as a contact at HOT initialization program.

Operation Error

Compile error occurs under following circumstances:

- When INITEND instruction is used in programs except the initialization program.
- When INITEND instruction is the only termination instruction in the program. (END instruction is needed)

Program Example

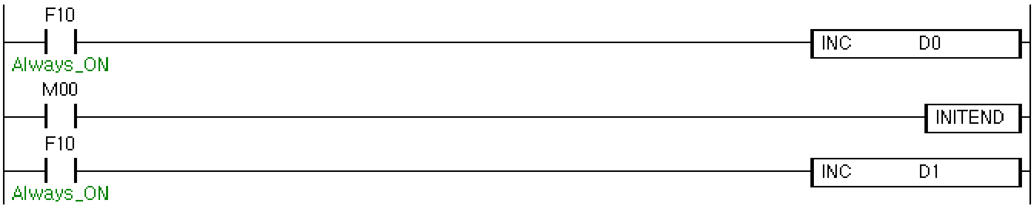
INITEND

In the following example, INITEND operates as below.

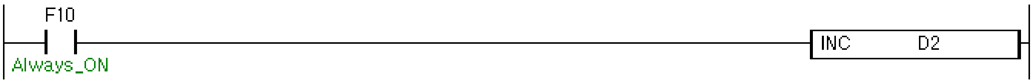
- When PLC is turned ON, the process starts from step 0 of initialization program.
- The values of D0 and D1 increase by INC instruction.
- Since M00 is OFF, the INC instruction located in the scan program does not operate. Only the values of D0 and D1 increase.
- When M00 is ON, INITEND instruction is executed and the initialization program is terminated.
- Scan ends at the end of the scan program where INITEND instruction is executed. Then, next scan starts from step 0 of scan program.
- The value of D2 increases by 1. Since the initialization program is terminated, the values of D0 and D1 do not increase.

Ladder Diagram (LD)

- Initialization Program (COLD)



- Scan Program



Instruction List (IL)

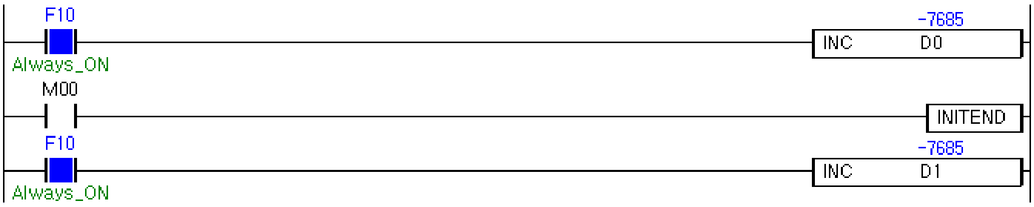
- Initialization Program (COLD)

Instruction	Device
LD	F10
INC	D0
LD	M00
INITEND	
LD	F10
INC	D1

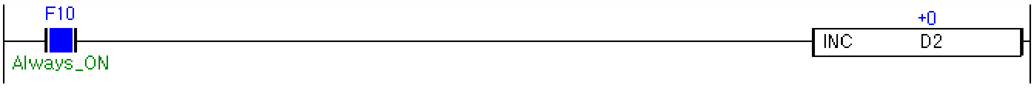
- Scan Program

Instruction	Device
LD	F10
INC	D2

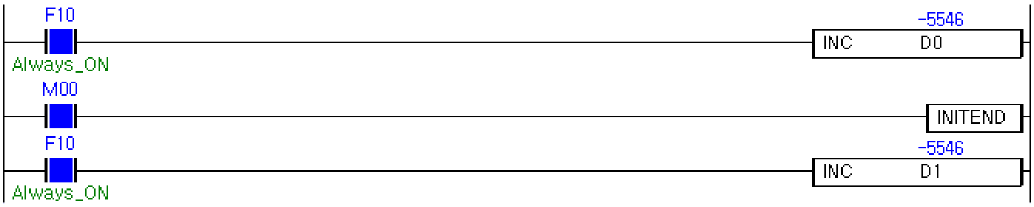
Initialization Program



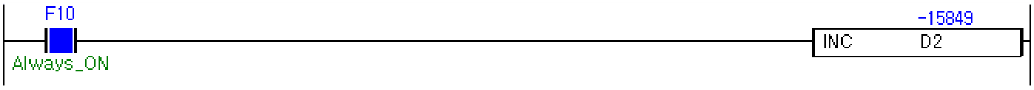
Scan Program



Initialization Program



Scan Program



1.7 Loop Instruction

1.7.1 FOR, NEXT

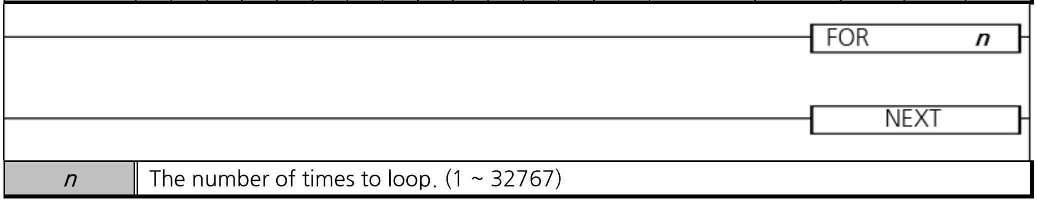
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

FOR, NEXT instructions executes a process between FOR~NEXT instructions for *n* times. After the last loop, next step of NEXT instruction is executed.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FOR	*n*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	2	O	-	-
NEXT		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



n	The number of times to loop. (1 ~ 32767)

FOR, NEXT

- FOR~NEXT instructions execute a process between FOR~NEXT instructions for *n* times. When the loop ends, next step of NEXT instruction is executed.
- From 1 to 32767 can be assigned to *n*.
- When -32767 ~ 0 are assigned to *n*, compile error occurs.
- Nesting FOR~NEXT in other FOR~NEXT is available up to 16 levels.
- FOR~NEXT instructions should be used in a pair.

⚠ When FOR~NEXT instructions are used excessively, scan time increases. System might be overloaded which might cause CPU shut down or initialization.

Execution Condition

No operation command is needed for FOR~NEXT instructions.
FOR~NEXT instructions operate at every scan.

Operation Error

Error Flag (F110)

The address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Compile error occurs under following circumstances:

- When assigned value of *n* exceeds the range of 1~32767.
- When FOR~NEXT instructions are not used in a pair.
- When nesting of FOR~NEXT instructions exceeds 16 levels.

Program Example

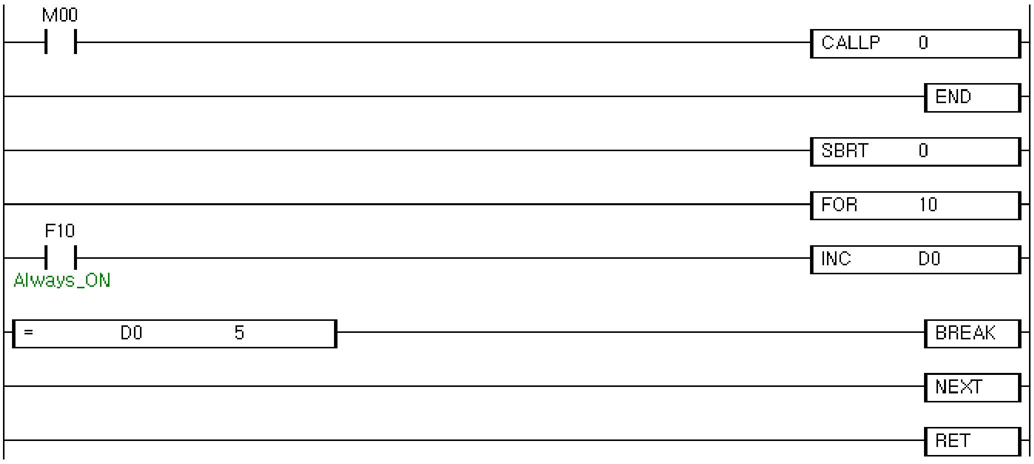
FOR, NEXT

In the following example, program operates as following.

- When M00 is ON, CALLP instruction calls subroutine program “0”.
- Then program repeats following process for 10 times:

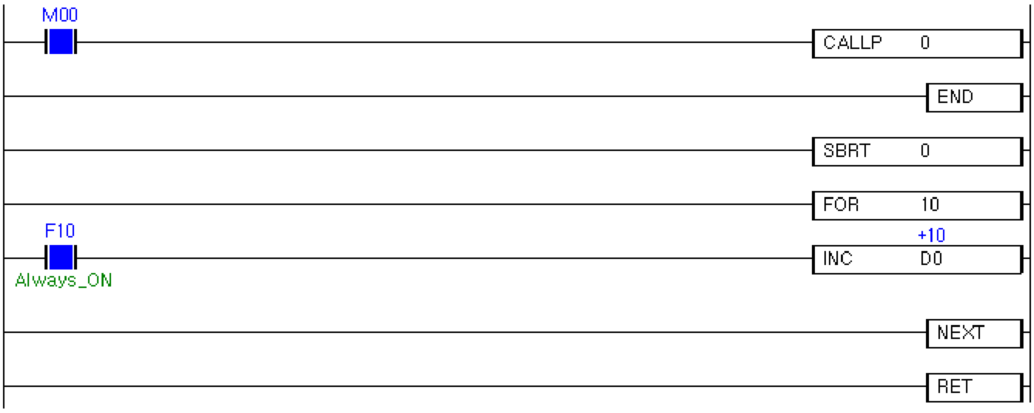
  a) INC instruction operates and the value of D0 becomes 10.
- After 10 times of loop, 1 scan is finished.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	M00
CALLP	0
END	
SBRT	0
FOR	10
LD	F10
INC	D0
NEXT	
RET	



1.7.2 BREAK, BREAKP

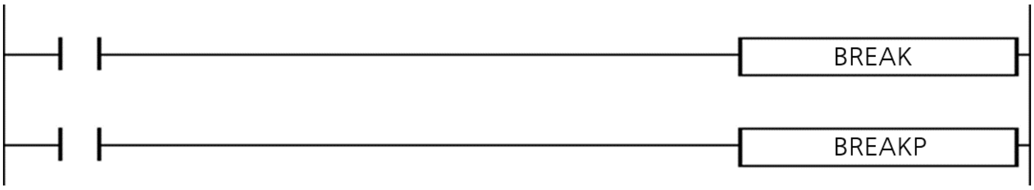
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

BREAK(P) enables an escape from FOR~NEXT area.

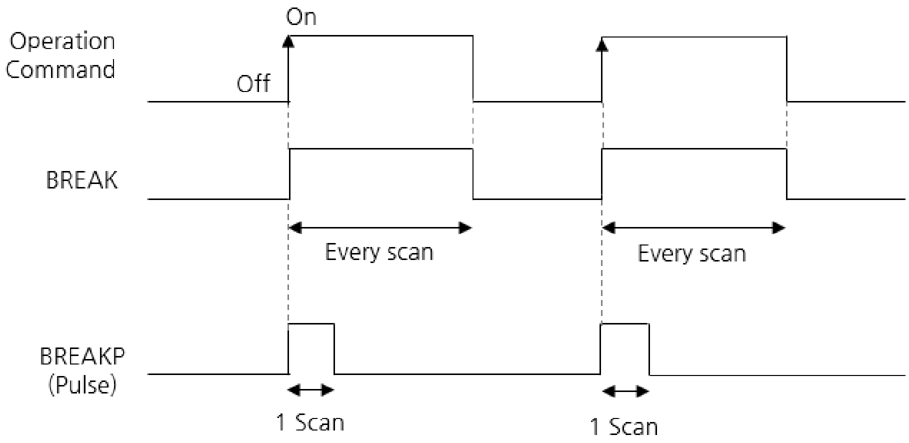
Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BREAK(P)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



BREAK, BREAKP

- BREAK(P) instructions can only be used between FOR~NEXT instruction.
- When BREAK(P) is executed, the loop between FOR~NEXT ends and the next step is executed.
- When BREAK(P) is executed, steps located between FOR and BREAK(P) operate but steps between BREAK(P) and NEXT do not operate.
- Multiple BREAK(P) instructions can be located between FOR~NEXT instruction.
- BREAK(P) instructions can be located anywhere between nesting of FOR~NEXT instruction.

Execution Condition



Operation Error

Compile error occurs under following circumstances:

- When BREAK(P) instruction is not located between FOR~NEXT instruction.

Program Example

BREAK, BREAKP

In the following example, program operates as following.

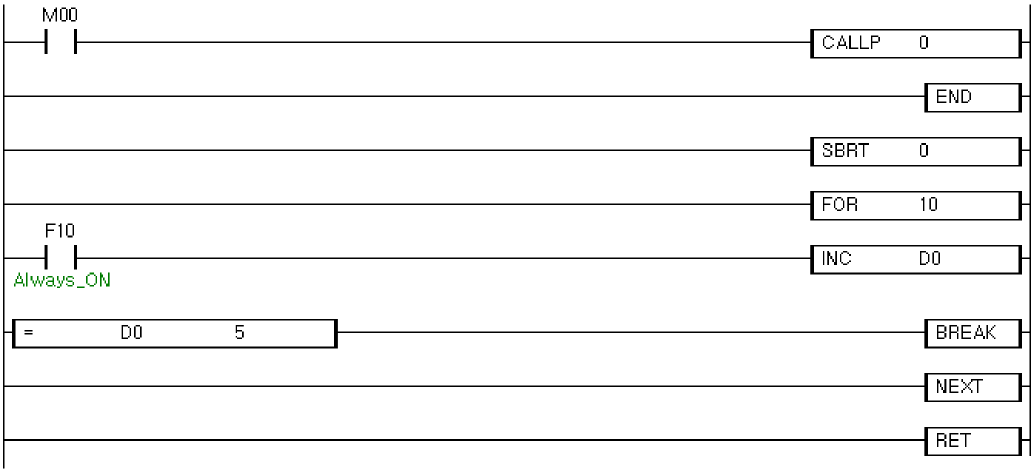
- When M00 is ON, CALLP instruction calls subroutine program “0”.
- Then program repeats following process for 10 times:

  a) INC instruction operates and the value of D0 increases by 1 at each loop.

  b) When the value of D0 reaches “5”, the execution condition of BREAK instruction turns ON.

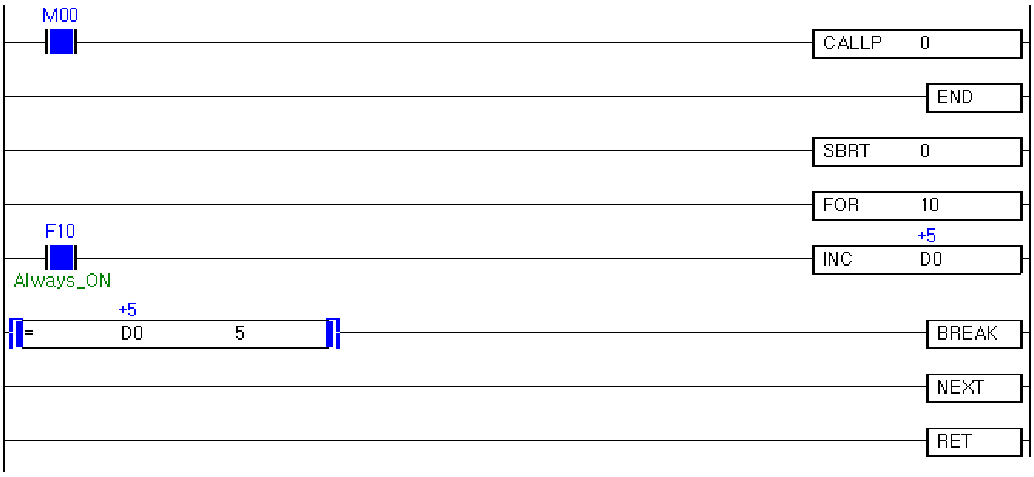
  c) BREAK instruction is executed and FOR~NEXT instruction do not operate.
- After 5 times of loop, 1 scan is finished.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	M00	
CALLP	0	
END		
SBRT	0	
FOR	10	
LD	F10	
INC	D0	
LD =	D0	5
BREAK		
NEXT		
RET		



1.8 Program Execution Instruction

1.8.1 GEI, GDI, EI, DI

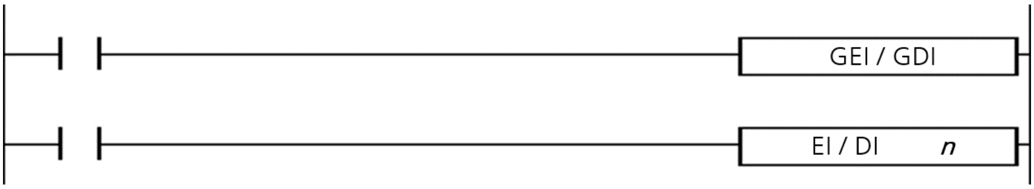
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

GEI enables all periodic interrupt program and GDI disables all periodic interrupt program. EI enables a periodic interrupt program and DI disables a periodic interrupt program.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
GEI, GDI		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EI, DI	*n*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	2	-	-	-



n	Interrupt PID number (0~14)

GEI

- Enables execution of general periodic interrupt programs. Once GEI is executed, the enabled status is maintained.
- Interrupts set by a parameter can be executed after the execution of GEI.
- When CPU has switched from STOP to RUN mode, all interrupt programs are disabled (DI). Therefore, to use general interrupt programs, execute GEI first.

⚠ Interrupt programs of CPU such as XPnB/E, CP3E, CP4E/F and PLC-S are enabled with the GEI instruction as a default. For this reason, these CPU do not need to execute GEI instruction when RUN/STOP key switch is changed from STOP to RUN.

GDI

- Disables execution of general interrupt programs.
- Once GDI instruction is executed, all interrupt programs are not executed.
- Once GDI instruction is executed, the disabled status is maintained.
- To execute interrupt programs again, use GEI instruction.

EI

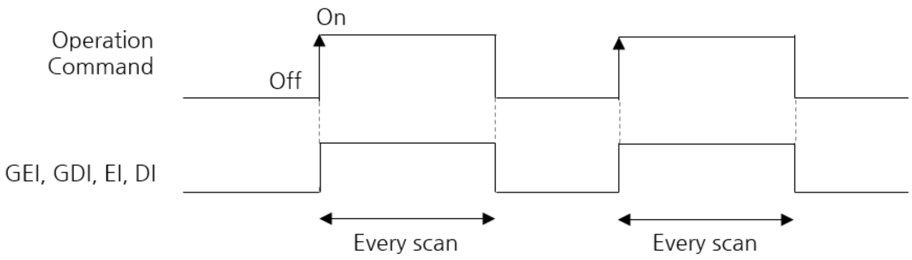
- Enables the interrupt program ID *n*.
- Enables execution of interrupt programs that were disabled by DI instruction.
- Once EI is executed, the enabled status is maintained.

DI

- Disables the interrupt program ID *n* until the EI instruction is executed.
- Once EI is executed after the execution of DI instruction, the disabled status changes into enabled status.
- After the execution of DI instruction, disabled interrupt program *n* is ignored even on the event of interrupt.
- When CPU has switched from STOP to RUN mode, all interrupt programs are disabled (DI). Therefore, to use general interrupt programs, execute GEI first.

⚠ Interrupt programs of CPU such as XPnB/E, CP3E, CP4E/F and PLC-S are enabled with the GEI instruction as a default. For this reason, these CPU do not need to execute GEI instruction when CPU mode is changed from STOP to RUN.

Execution Condition



Operation Error

There is no operation error in GEI, GDI, EI, DI instructions.

Program Example

EI, DI, GEI, GDI

In the following example, GEI, GDI, EI and DI operate as below.

- Example which enables or disables the interrupt program.

  a) Turn ON X00 and enable all periodic interrupt program by executing GEI. After, X01 and X02 turn ON and PID “1” and “2” of periodic interrupt program are enabled.

  b) When X06 and X07 are turned ON, the value of D0 and D1 increase.

  c) When X03 is turned ON, PID “1” of periodic interrupt program is disabled.

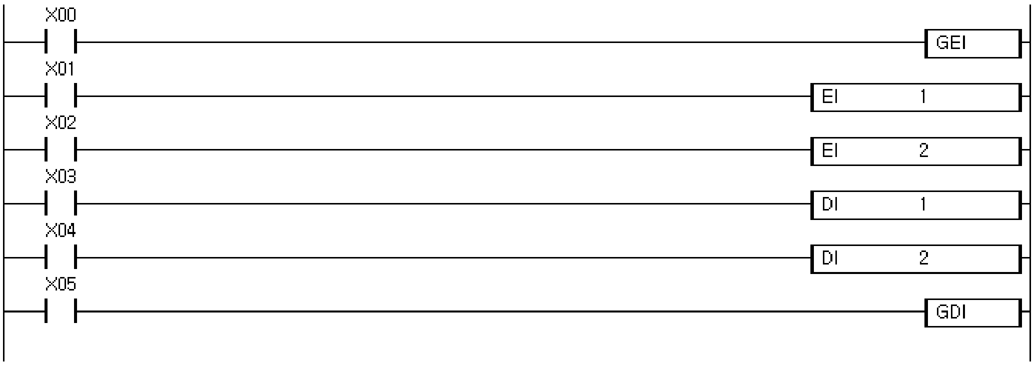
  d) When X04 is turned ON, PID “2” of periodic interrupt program is disabled.

  e) When X05 is turned ON, it disables all interrupt programs by executing GDI.

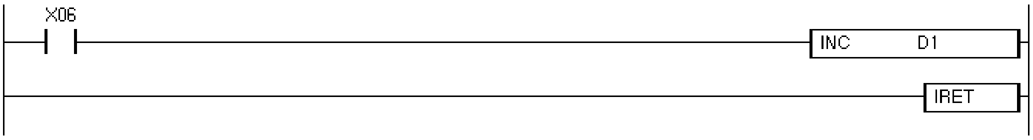
  f) Even X01 and X02 stay turned ON, interrupt programs do not operate since GDI has been executed.

Ladder Diagram (LD)

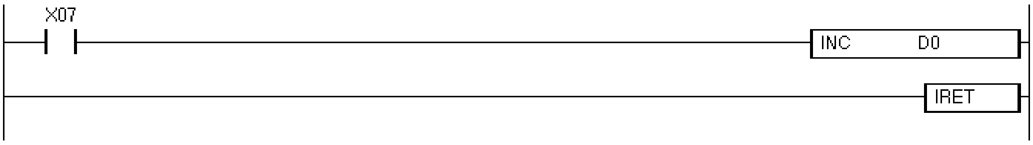
- Scan program



- Periodic interrupt program (PID: 1)



- Periodic interrupt program (PID: 2)



Instruction List (IL)

- Scan program

Instruction	Device
LD	X00
GEI	
LD	X01
EI	1
LD	X02
EI	2
LD	X03
DI	1
LD	X04
DI	2
LD	X05
GDI	

- Periodic interrupt program (PID: 1)

Instruction	Device
LD	X06
INC	D1
IRET	

- Periodic interrupt program (PID: 2)

Instruction	Device
LD	X07
INC	D0
IRET	

1.8.2 IRET

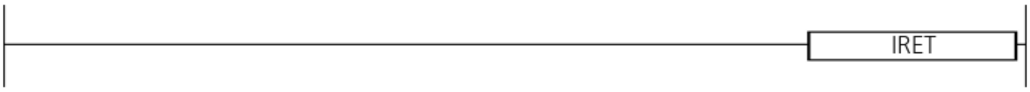
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

IRET instruction indicates the end of periodic interrupt program.

Instruction	Valid device type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
IRET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



IRET

- IRET ends the periodic interrupt program.
- After the execution of IRET, the process returns to the scan program.
- IRET instruction can be used only in the periodic interrupt program. When it is used in the other program, compile error occurs.

Execution Condition

No operation condition is needed to execute this instruction.

Operation Error

Compile error occurs under following circumstances:

- When IRET instruction is used in a program except periodic interrupt program.

Program Example

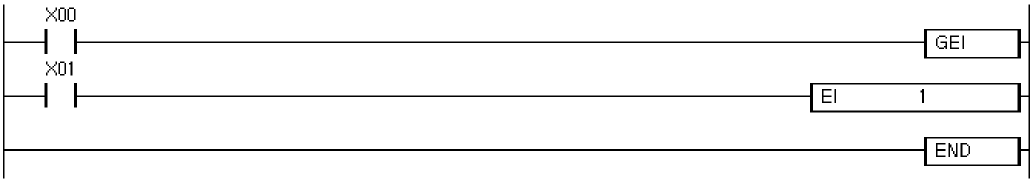
IRET

In the following example, IRET instruction operates as following.

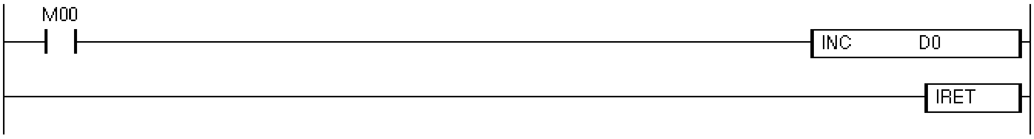
- When X00 turns ON, general interrupt program is enabled.
- When M00 of interrupt program turns ON, the value of D0 is increased.
- By the operation of IRET, the process is recovered to the scan program.

Ladder Diagram (LD)

- Scan program



- Periodic interrupt Program (PID: 1)



Instruction List (IL)

- Scan program

Instruction	Device
LD	X00
GEI	
LD	X010
EI	1
END	

- Periodic interrupt Program

Instruction	Device
LD	M00
INC	D0
IRET	

1.8.3 EPGM, DPGM

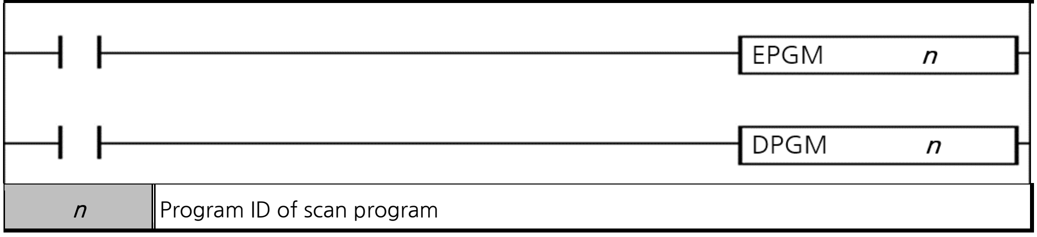
Supported PLC Series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

EPGM instruction enables the scan program designated by program ID *n*.
DPGM instruction disables the scan program designated by program ID *n*.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EPGM DPGM	*n*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	2	-	-	-



n	Program ID of scan program

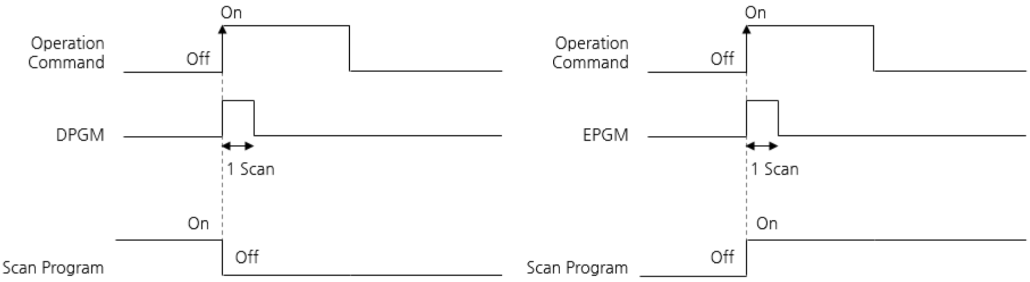
EPGM

- EPGM instruction enables the scan program designated by the program ID *n*.
- This instruction is used when a scan program is disabled by DPGM instruction.
- To enable a scan program, EPGM instruction has to be executed on the scan program which has a different program ID.

DPGM

- DPGM instruction disables the scan program designated by the program ID *n*.
- This instruction is used to disable a scan program. To enable the disabled scan program, execute EPGM instruction.
- When the instruction is executed, the specified scan program is disabled after the END instruction is executed.
- DPGM instruction can only disable normal scan programs, periodic interrupt programs and SFC programs.

Execution Condition



Operation Error

There is no error in EPGM and DPGM instructions.

Program Example

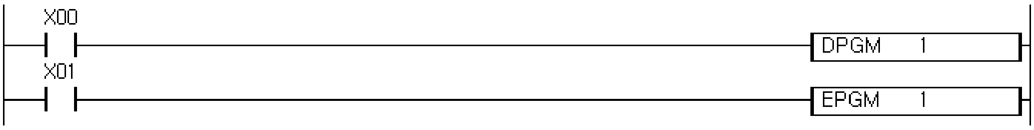
EPGM, DPGM

Following program example operates as following:

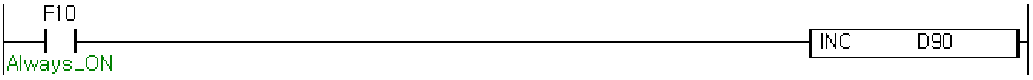
- When X00 turns ON, DPGM instruction is executed.
- Then, the scan program (Program ID = 1) is disabled. It means that INC instruction of the disabled scan program does not operate.
- When X01 turns ON, EPGM instruction is executed.
- The disabled scan program (Program ID = 1) is enabled and the INC instruction operates.

Ladder Diagram (LD)

- Scan program (Program ID = 0)



- Scan program (Program ID = 1)



Instruction List (IL)

- Scan program (Program ID = 0)

Instruction	Device
LD	X00
DPGM	1
LD	X01
EPGM	1

- Scan program (Program ID = 1)

Instruction	Device
LD	F10
INC	D90

1.9 Other Instructions

1.9.1 WDT, WDTP

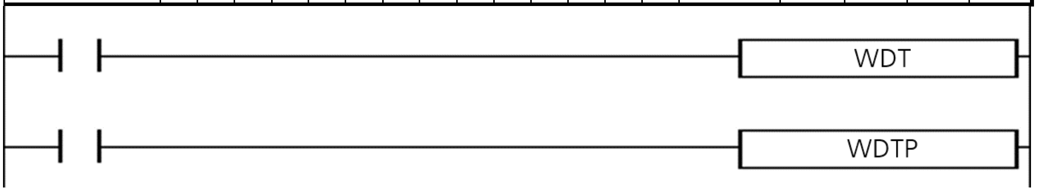
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

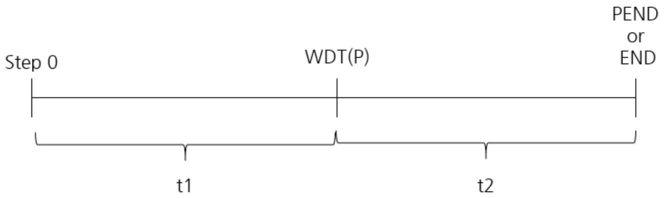
WDT(P) instructions resets the watchdog timer during the execution of a sequence program.

Instruction	Valid Device Type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
WDT(P)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-



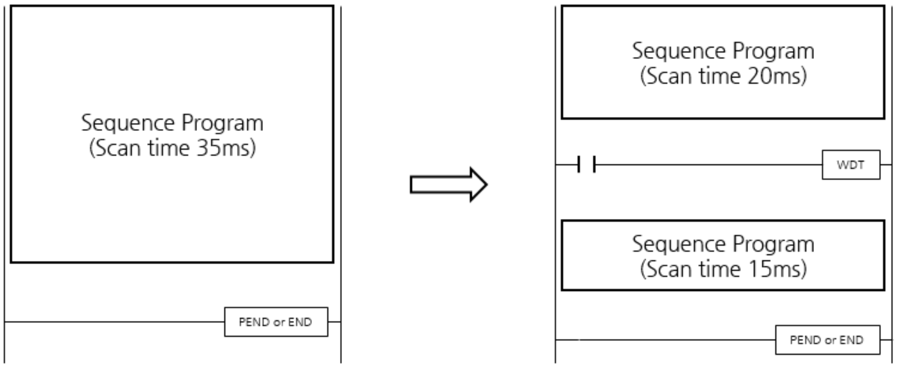
WDT, WDTP

- WDT(P) instructions resets watchdog timer during the execution of a sequence program.
- When the scan time exceeds the setting value of watchdog timer, a “User WDT error” occurs and PLC stops. To avoid the error, you must increase the setting value of watchdog timer at [Tool] - [PLC Parameter...]
- If you cannot increase the setting value of watchdog timer, you have to execute WDT(P) instruction during the execution of sequence program.
- Scan time t1, from step 0 to WDT(P) instruction, and scan time t2, from WDT(P) instruction to the PEND (END), must not exceed the setting value of watchdog timer.

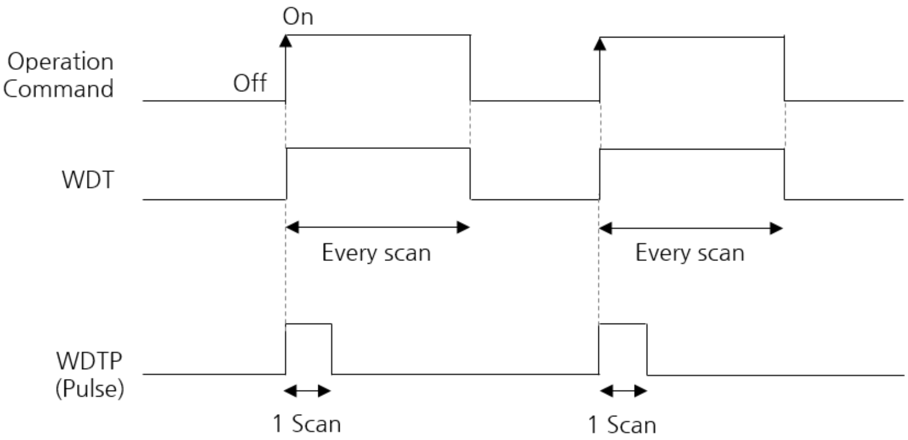


- Multiple WDT(P) instructions can be used during a scan. However, doing so may cause a delay until the output refresh.

- When a scan begins in the sequence program with WDT(P) instructions executed, watchdog timer runs until the scan meets WDT(P) instruction. When the scan meets WDT(P) instruction, watchdog timer is reset. If the setting value of watchdog timer is 30ms, WDT(P) instruction have to be located at the point where the scan time from step 0 to WDT(P) instruction and the scan time from WDT(P) instruction to PEND(or END) do not exceed 30ms. Refer to the figure below for an example.



Execution Condition



Operation Error

“User WDT error” occurs when one of scan time before or after the WDT(P) instruction exceeds the setting value of watchdog timer.

Program Example

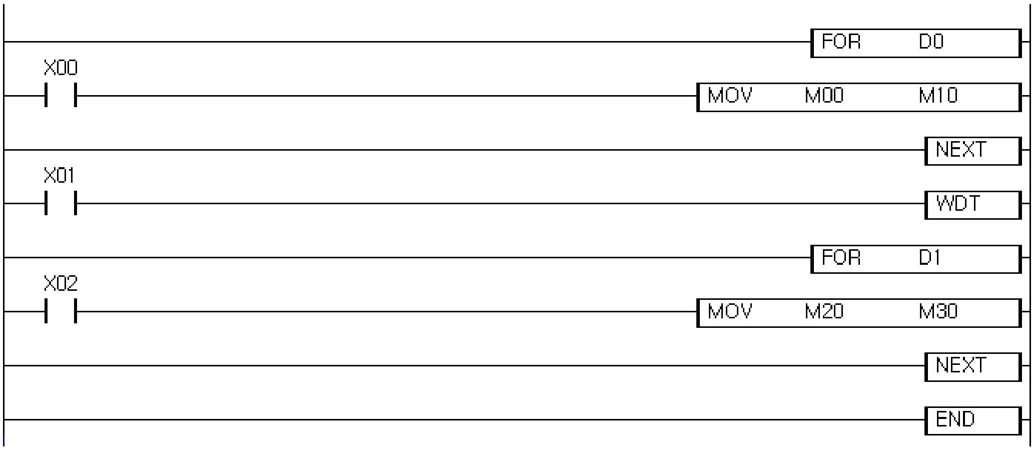
WDT

Setting value of watchdog timer is 10ms.

When “1000” is assigned to D0 and D1 while WDT instruction is not executed, the process between FOR and NEXT repeats 1000 times. The scan time of this sequence program is 17~18ms. “User WDT error” occurs and PLC stops since scan time exceeds setting value of watchdog timer (10ms).

When “1000” is assigned to D0 and D1 with executed WDT instruction, the process between FOR and NEXT repeats 1000 times. However, “User WDT error” does not occur since WDT resets the watchdog timer before scan time exceeds the setting value of watchdog timer.

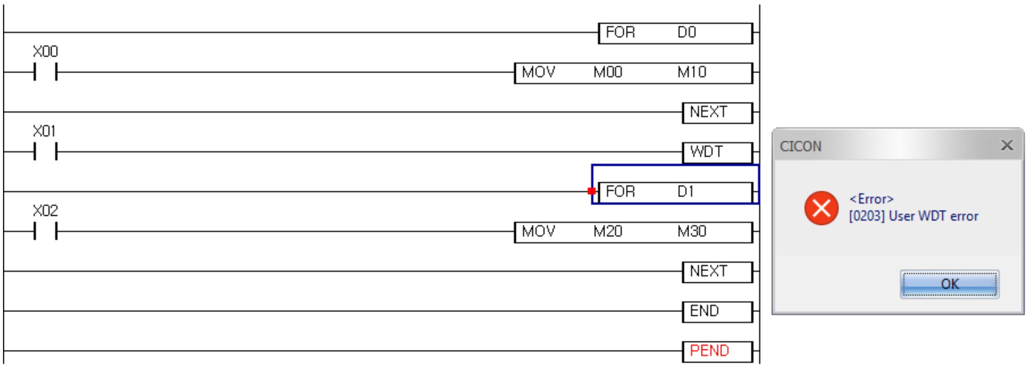
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
FOR	D0	
LD	X00	
MOV	M00	M10
NEXT		
LD	X01	
WDT		
FOR	D1	
LD	X02	
MOV	M20	M30
NEXT		
END		

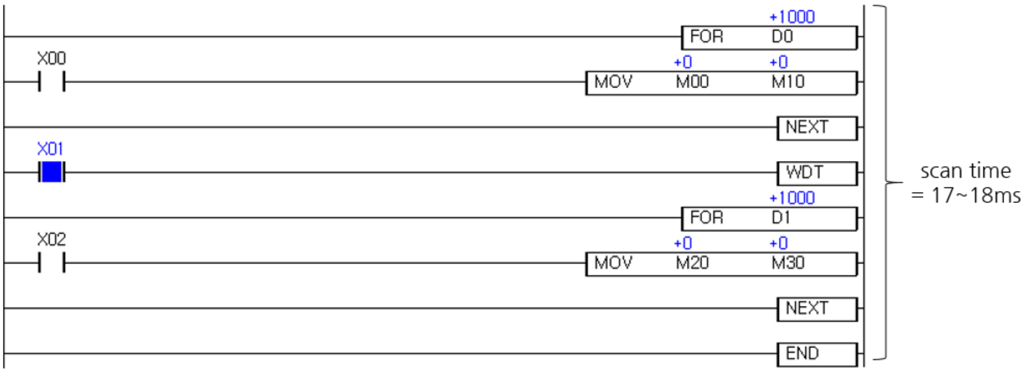
- When “1000” is assigned to D0 and D1 without execution of WDT instruction, scan time becomes 17~18ms. It means that scan time exceeds the setting value of watchdog timer and error occurs. In this case, PLC stops.



<Error>
[0203] User WDT error

- When X01 turns ON, WDT instruction is executed. WDT instruction resets watchdog timer after the first FOR~NEXT instructions. Then, “1000” is assigned to D0 and D1.

- In this case, the scan time for the sequence program will become 17~18ms but it will operate without an error.



1.9.2 STC, CLC

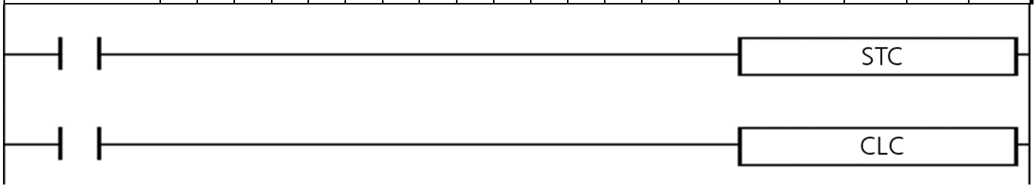
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

STC instruction sets carry flag (F112) when operation condition is ON.
CLC instruction resets carry flag (F112) when operation condition is ON.

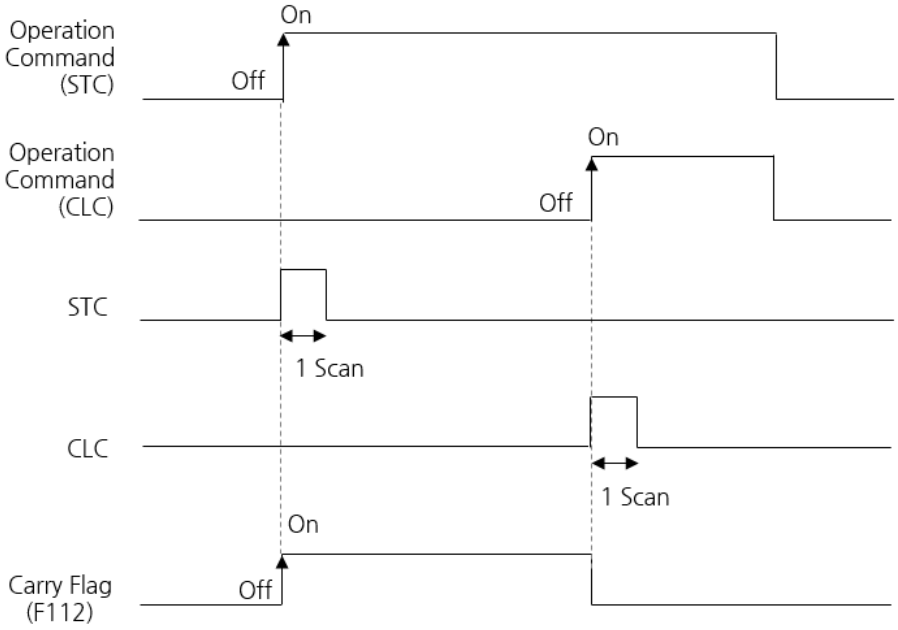
Instruction	Valid Device Type															Steps	Flag		
	M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
STC, CLC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	O



STC, CLC

- STC instruction sets the carry flag (F112) when operation condition is ON.
- CLC instruction clears the carry flag (F112) when operation condition is ON.
- When operation conditions of both STC and CLC isntructions are OFF or ON, there is no change in the carry flag (F112).

Execution Condition



⚠ There is a difference in operation with carry flag between BP, XPnA/1R, CP3A/B/P/U, CP4A~D/U and PLC-S, XPnB/E/F, CPnE/F. When you write a sequence program with other instructions using the carry flag, please be aware.
(*) For more information, please refer to n.n Carry Flag.

Operation Error

There is no operation error in STC, CLC instructions.

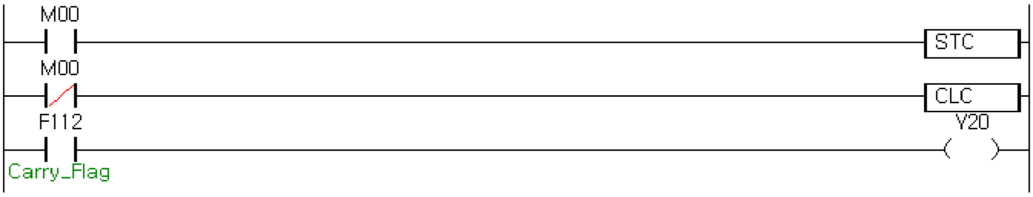
Program Example

STC, CLC

The program operates as following:

- When M00 is ON, STC instruction is executed and sets carry flag (F112). When carry flag is ON, Y20 turns ON.
- When M00 is OFF, CLC instruction is executed and clears carry flag. When carry flag is cleared, Y20 turns OFF.

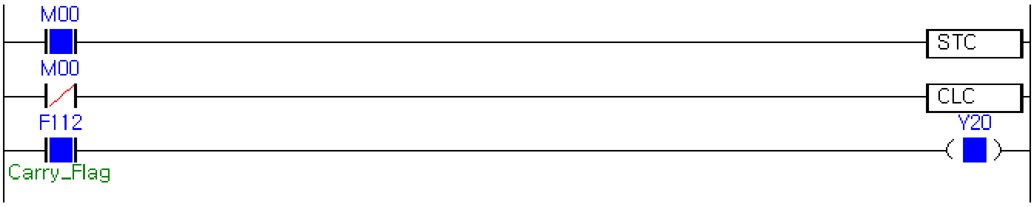
Ladder Diagram (LD)



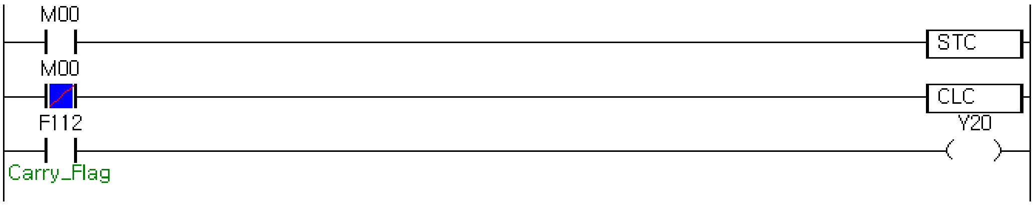
Instruction List (IL)

Instruction	Device
LD	M00
STC	
LDI	M00
CLC	
LD	F112
OUT	Y20

- When M00 is ON, STC instruction sets carry flag (F112). When F112 turns ON, Y20 turns ON.



- When M00 is OFF, CLC instruction clears carry flag (F112). When F112 turns OFF, Y20 turns OFF.



1.9.3 RFS, RFSP

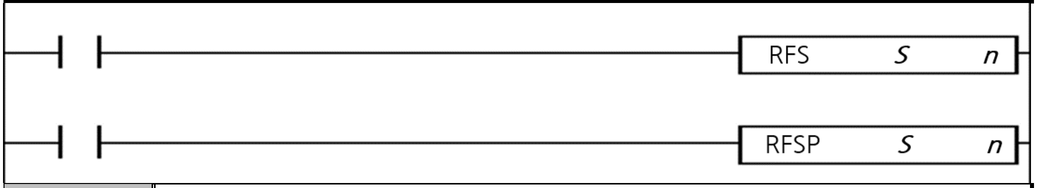
Supported PLC series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

RFS(P) instructions force I/O refresh on the *n* words of I/O devices starting from *S*.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error (New)	Zero	Carry
RFS(P)	*S*	-	O	O	-	-	-	-	-	-	-	-	-	-	-	-	3	O	-	-
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

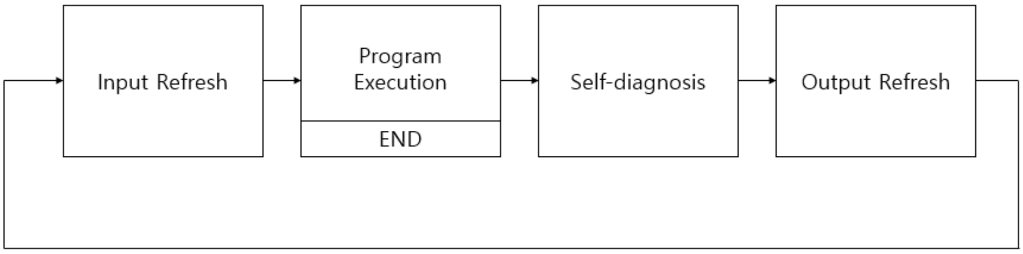


S	Head address of I/O device to be refreshed.
n	The number of I/O devices to refresh. (Word)

⚠ When this instruction is executed with CPU types such as PLC-S, BP, XPnA/1R, CP3A/B/P/U and CP4A~D/U, entire I/O devices are refreshed regardless of the devices and values assigned to operands.

RFS, RFSP

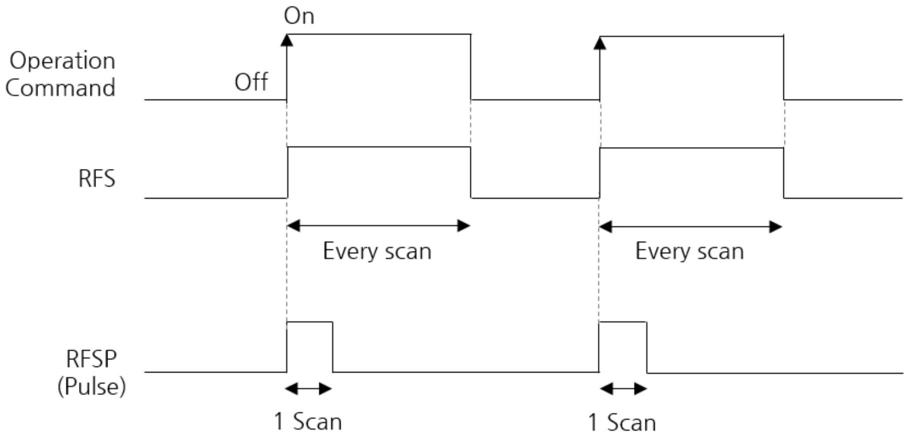
- RFS(P) instructions force I/O refresh during the execution of scan program. The number of I/O devices is specified by *n* words. The specified I/O devices starting from *S* are refreshed.
- Figure below is a scan cycle. When RFS(P) instructions are executed in the progress of scan cycle, I/O refresh is executed forcibly during the program execution.



- Normally external I/O refreshing is conducted after the execution of END instruction, so it is not possible to output a pulse signal to an external actuator during a scan. However, when RFS(P) instruction is executed, corresponding I/O devices are refreshed forcibly during the program execution. It means that a you can output a pulse signal to an external source during a scan.

⚠ RFS(P) instruction operates only on the I/O devices assigned to the local base.

Execution Condition



Operation Error

Error flag (F110)

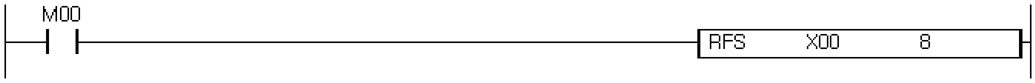
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

RFS

When M00 turns ON, RFS instruction is executed. The RFS instruction refreshes input devices of 8 words starting from X00.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	M00	
RFS	X00	8

1.9.4 FF

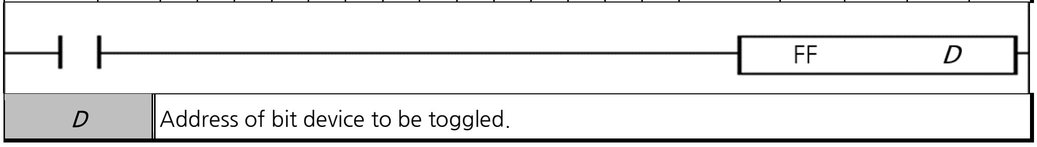
Supported PLC Series

XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	-	-	-

Function

FF instruction toggles the bit device assigned to *D*.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FF	*D*	O	-	O	O	O	-	-	-	-	-	-	O	-	-	-	1	-	-	-



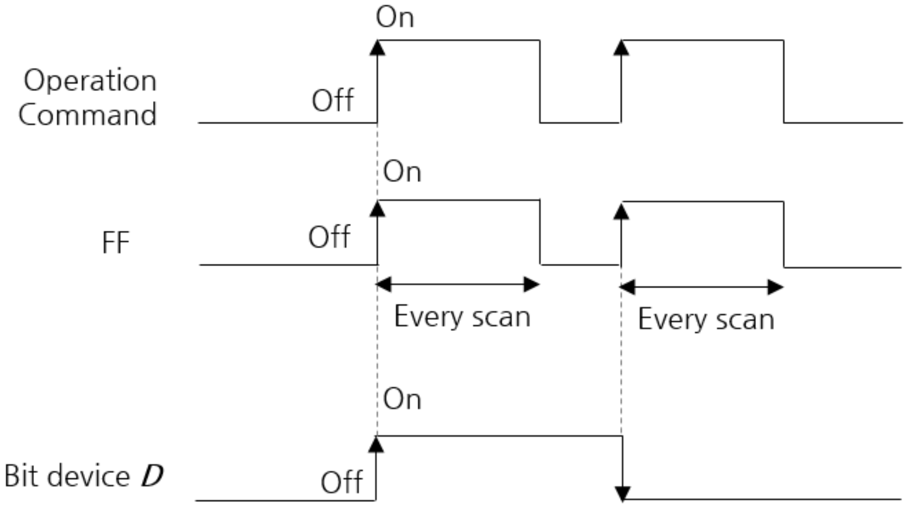
D	Address of bit device to be toggled.

FF

- FF instruction toggles the bit device assigned to *D*.
- Once the instruction is executed, the status of assigned bit device is inversed.

Status of bit device	
Before execution of FF	After execution of FF
OFF	ON
ON	OFF

Execution Condition



Operation Error

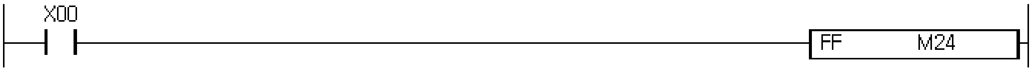
There is no error in this instruction.

Program Example

FF

When X00 turns ON, FF instruction is executed. Then, the bit device M24 turns ON. When X00 turns OFF and ON again, M24 turns OFF.

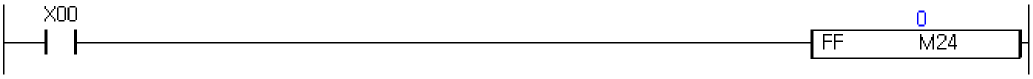
Ladder Diagram (LD)



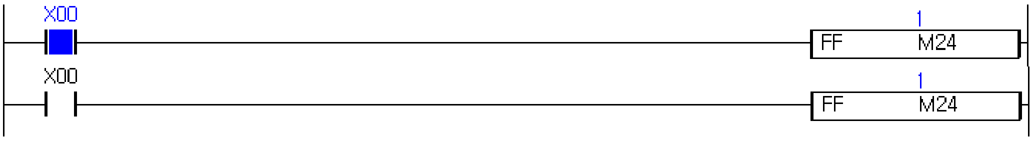
Instruction List (IL)

Instruction	Device
LD	X00
FF	M24

- Before the execution of FF instruction, M24 is set as “0”.



- When X00 turns ON, M24 turns ON. Even if X00 turns OFF, the status of M24 does not change.



- When X00 turns ON again, then M24 turns OFF.