2 Application Instruction

2.1 Arithmetic Operation Instructions

2.1.1 ADD, ADDP, DADD, DADDP

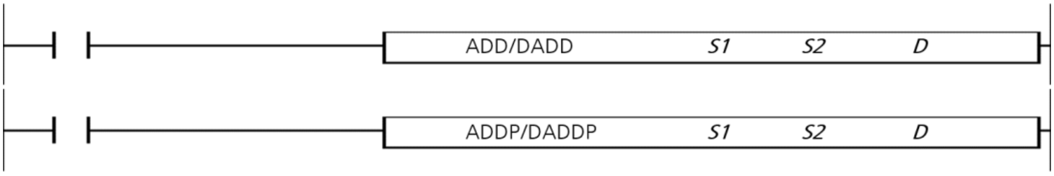
Supported PLC series

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

ADD(P) adds word data *S1* to word data *S2*, and stores the result at *D* in word.
DADD(P) adds double word data *S1* to double word data *S2*, and stores the result at *D* in double word.

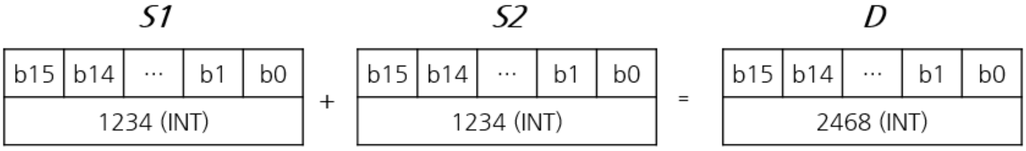
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ADD(P) DADD(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	o	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S1	Head address of the device where word/double word data to be added to is stored.
S2	Head address of the device where word/double word data for adding is stored
D	Head address of the device where the addition result (word/double word) is stored

ADD, ADDP

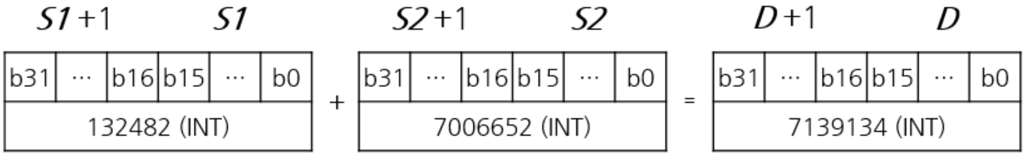
- Adds word data assigned to *S1* to word data assigned to *S2*, and stores the result of word data in the device assigned to *D*.



- At *S1* and *S2* and *D*, a value from -32768 to 32767 can be specified.

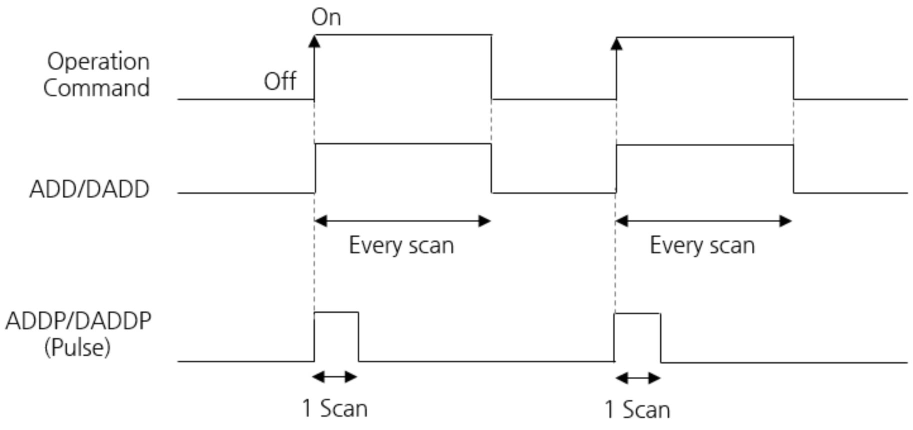
DADD, DADDP

- Adds double word data assigned to *S1* to double word data assigned to *S2*, and stores the result of double word data in the device assigned to *D*.



- At *S1* and *S2* and *D*, a value from -2147483648 to 2147483647 can be specified.

Execution Condition



Operation Error

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)

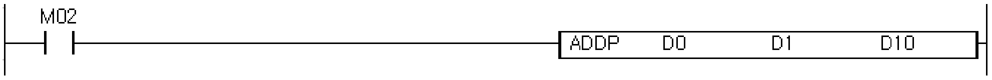
F110 turns ON when the value of the data for adding exceeds the device assigned to *S1* or *S2*.

Program Example

ADDP

The program executes addition of D0 and D1 when operation command M02 is ON, then outputs value to D10.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M02		
ADDP	D0	D1	D10

2.1.2 SUB, SUBP, DSUB, DSUBP

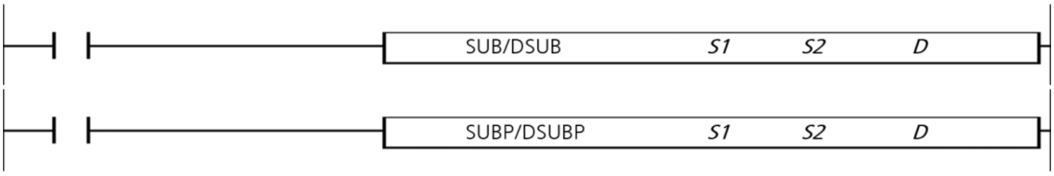
Supported PLC series

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

SUB(P) subtracts word data *S2* from word data *S1*, and stores the result at *D* in word.
DSUB(P) subtracts double word data *S2* from double word data *S1*, and stores the result at *D* in double word.

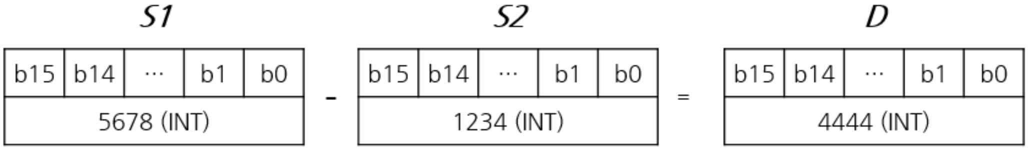
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SUB(P) DSUB(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S1	Head address of the device where word/double word data to be subtracted from is stored.
S2	Head address of the device where word/double data for subtracting is stored.
D	Head address of the device where the subtraction result (word/double word) is stored.

SUB, SUBP

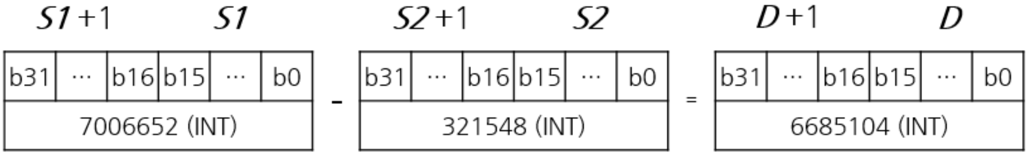
- Subtracts word data assigned to *S2* from word data assigned to *S1*, and stores the result of word data in the device assigned to *D*.



- At *S1* and *S2* and *D*, a value from -32768 to 32767 can be specified.

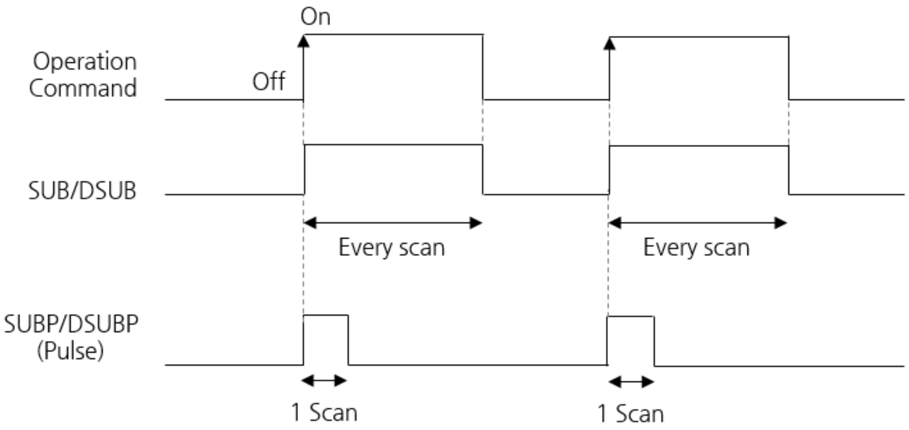
DSUB, DSUBP

- Subtracts double word data assigned to *S2* from double word data assigned to *S1*, and stores the result of double word data in the device assigned to *D*.



- At *S1* and *S2* and *D*, a value from -2147483648 to 2147483647 can be specified.

Execution Condition



Operation Error

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)

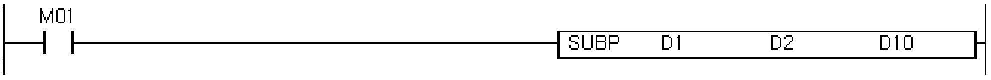
F110 turns ON when the value of the data for subtracting exceeds the device assigned to ***S1*** or ***S2***.

Program Example

SUBP

The program executes subtraction of D2 from D1 when operation command M01 is ON, then stores the result value to D60.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M01		
SUBP	D1	D2	D10

2.1.3 MUL, MULP, DMUL, DMULP

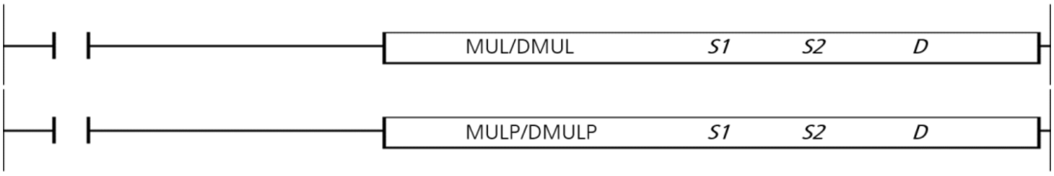
Supported PLC series

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

MUL(P) multiplies word data *S1* and word data *S2*, and stores the result at *D* in double word.
DMUL(P) multiplies double word data *S1* and double word data *S2*, and stores the result at *D* in 4 words.

Introduction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
MUL(P) DMUL(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				

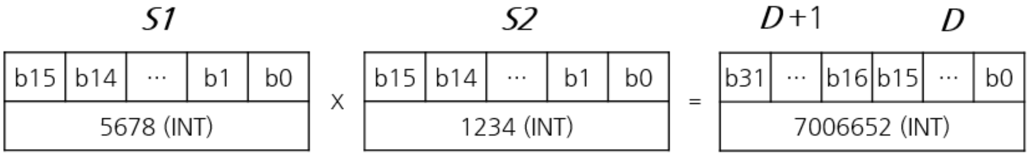


S1	Head address of the device where word/double word data to be multiplied is stored
S2	Head address of the device where word/double word data for multiplying is stored
D	Head address of the device where the multiplication result (double word/4 word) is stored

⚠ DMUL(P) instruction doesn't support device R.

MUL, MULP

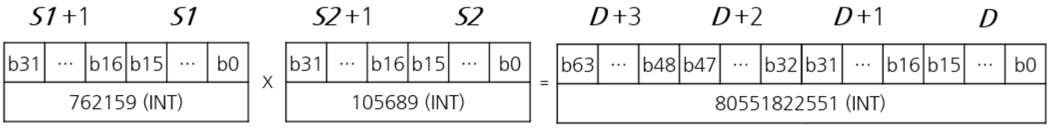
- Multiplies word data assigned to *S1* and word data assigned to *S2*, and stores result of double word data in device assigned to *D*.



- At *S1* and *S2*, a value from -32768 to 32767 can be specified.

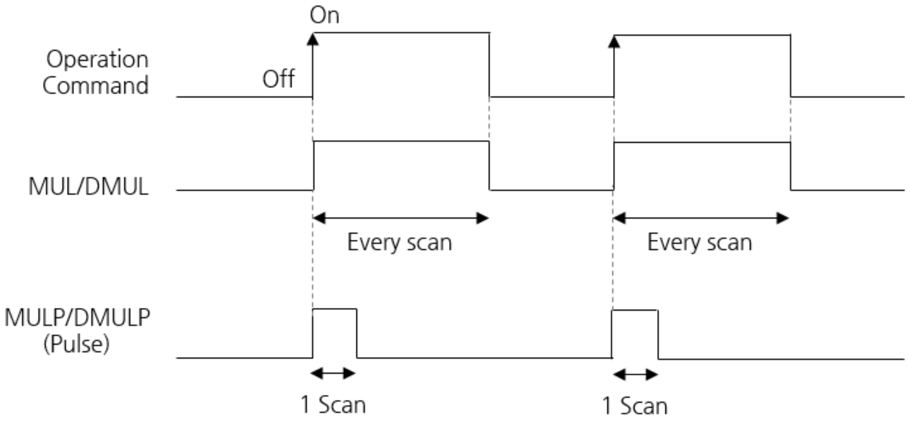
DMUL, DMULP

- Multiplies double word data assigned to *S1* and double word assigned to data *S2*, and stores result of 4-word data in device assigned to *D*.



- At *S1* and *S2*, a value from -2147483648 to 2147483647 can be specified.

Execution Condition



Operation Error

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)

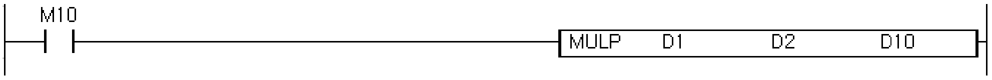
F110 turns ON when the value of the data for multiplying exceeds the device assigned to ***S1*** or ***S2***.

Program Example

MULP

The program executes multiplication of D1 and D2 when operation command M10 is ON, then stores the result in D10 and D11.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M10		
MULP	D1	D2	D10

2.1.4 WMUL, WMULP, DWMUL, DWMULP

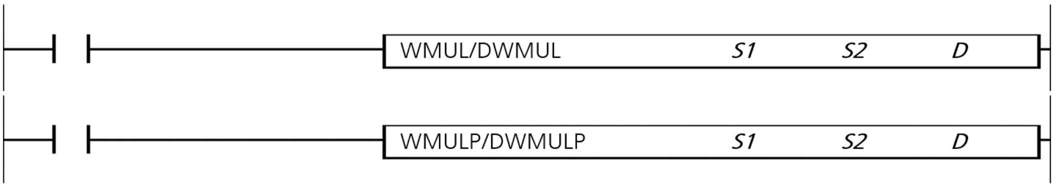
Supported PLC series

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

WMUL(P) multiplies word data *S1* and word data *S2*, and stores the result at *D* in word.
DWMUL(P) multiplies double word data *S1* and double word data *S2,* and stores the result at *D* in double word.

Introduction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
WMUL(P) DWMUL(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				

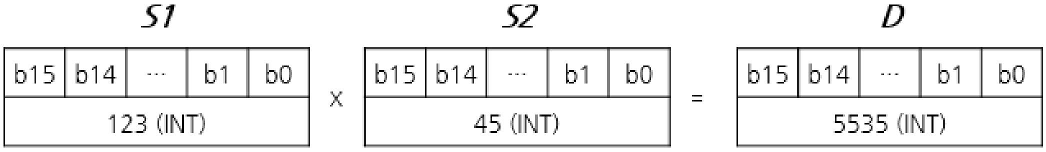


S1	Head address of the device where word/double word data to be multiplied is stored.
S2	Head address of the device where word/double word data for multiplying is stored.
D	Head address of the device where the multiplying is stored. (word/double word data)

⚠ DWMUL(P) instruction doesn't support device R.

WMUL, WMULP

- Multiplies word data assigned to *S1* and word data assigned to *S2*, and stores result of word data in device assigned to *D*.

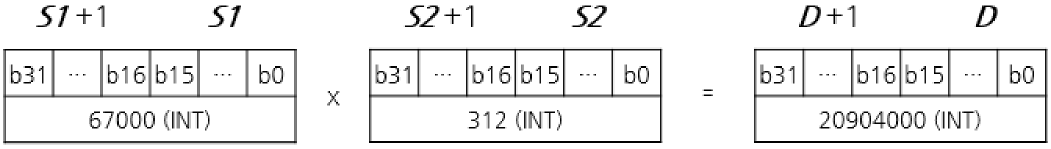


- At *S1* and *S2,* -32768 to 32767 can be specified.

- Unlike MUL(P) instruction, the result value is not stored in size of double word data, but remains in size of word data.

DWMUL, DWMULP

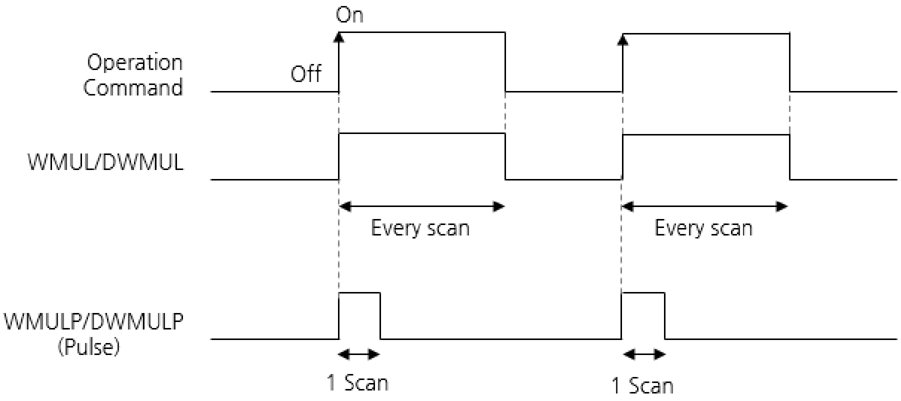
- Multiplies double word data assigned to *S1* and double word data assigned to *S2,* and stores lower word data in device assigned to *D*, upper word data in device assigned to *D* +1.



- At *S1* and *S2,* a value from -2147483648 to 2147483647 can be specified.

- Unlike DMUL(P) instruction, the result value is not stored in size of 4 word data, but remains in size of double word data.

Execution Condition



Operation Error

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)

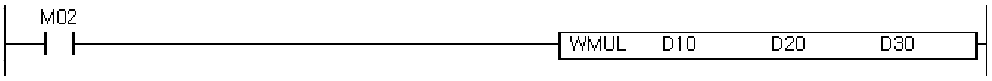
F110 turns ON when the result value exceeds the data range of device D in WMUL(P)/DWMUL(P) instruction.

Program Example

WMUL, WMULP

The program executes multiplication of D10 and D20 when M02 is ON, then store the value in D30.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M02		
WMUL	D10	D20	D30

2.1.5 DIV, DIVP, DDIV, DDIVP

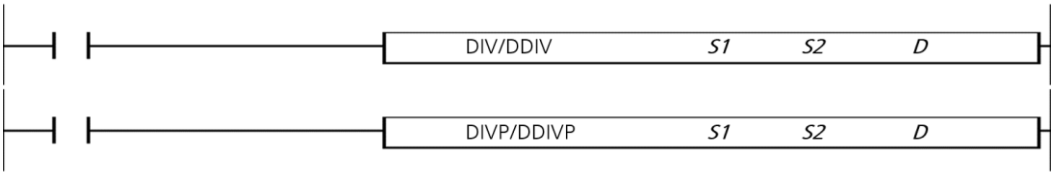
Supported PLC series

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

DIV(P) divides word data *S1* by word data *S2*, and stores the quotient at *D* and the remainder at *D*+1 in word.
DDIV(P) divides double word data *S1* by double word data *S2*, and stores the quotient at *D* and the remainder at *D*+1 in double word.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
DIV(P) DDIV(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				



S1	Head address of the device where word/double word data to be divided is stored
S2	Head address of the device where word/double word data for dividing is stored
D	Head address of the device where the quotient (word/double word) is stored
D+1/D+2	Head address of the device where the remainder (word/double word) is stored

⚠ DDIV(P) instruction doesn't support device R.

DIV, DIVP

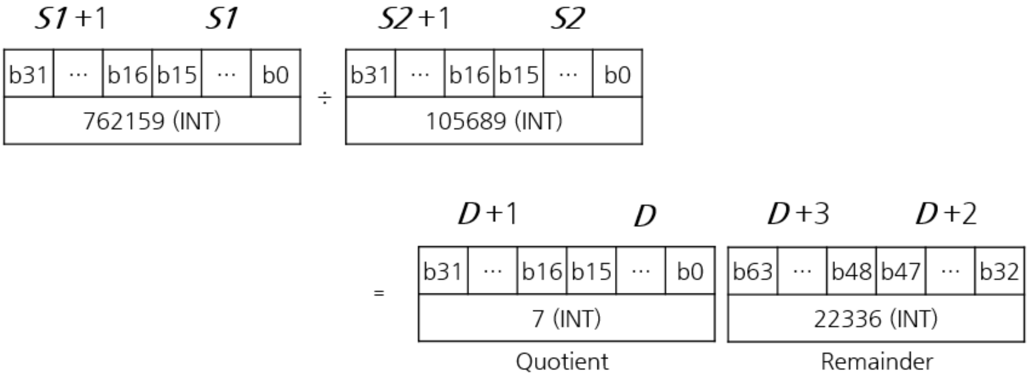
- Divides word data assigned to *S1* by word data assigned to *S2*, and stores the quotient of word data in device assigned to *D* and the remainder of word data at *D*+1.

S1		*S2*		*D* (Quotient)	*D*+1 (Remainder)
b15 | b14 | ··· | b1 | b0		b15 | b14 | ··· | b1 | b0		b15 | b14 | ··· | b1 | b0	b15 | b14 | ··· | b1 | b0
5678 (INT)	÷	1234 (INT)	=	4 (INT)	742 (INT)

- At *S1* and *S2*, a value from -32768 to 32767 can be specified.
- The result of the division operation is stored as double word data, and both the quotient and remainder are stored.

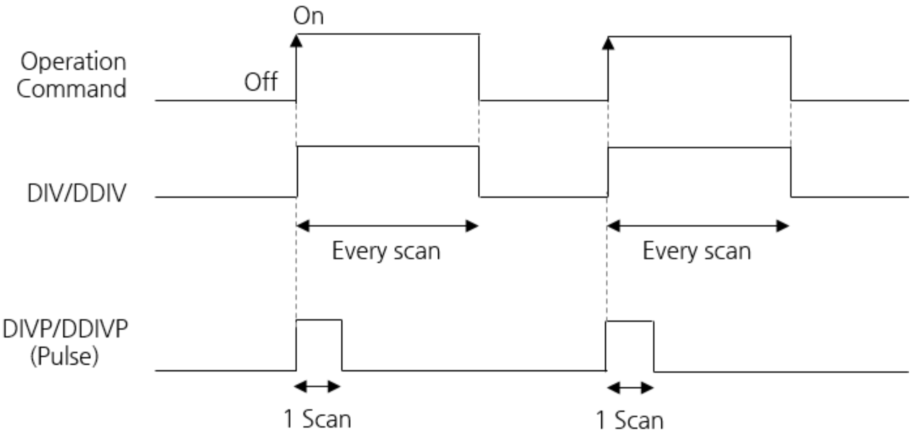
DDIV, DDIVP

- Divides double word data assigned to *S1* by double word data assigned to *S2*, and stores the quotient of double word data in device assigned to *D* and *D*+1, the remainder of double word data in device assigned to *D*+2, *D*+3.



- At *S1* and *S2*, a value from -2147483648 to 2147483648 can be specified.
- The division operation result is stored in 4-word data and both the quotient and remainder are stored.

Execution Condition



Operation Error

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)

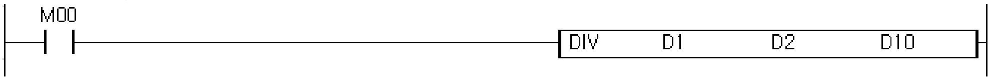
F110 turns ON for 1 scan when divisor is 0. (=When *S2* is 0)

Program Example

DIV

The program executes division of D1 by D2 when operation command M00 is ON, then store the quotient value in D10 and the remainder value in D20.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
DIV	D1	D2	D10

2.1.6 WDIV, WDIVP, DWDIV, DWDIVP

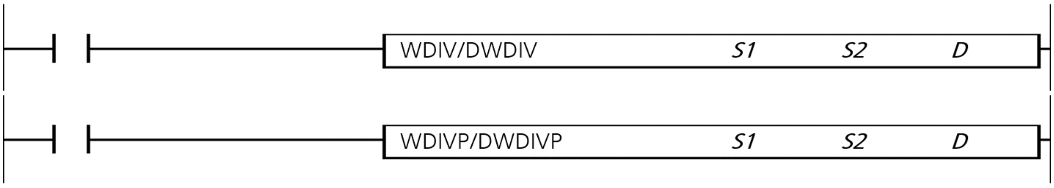
Supported PLC series

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

WDIV(P) divides word data *S1* by word data *S2*, and stores the quotient at *D* in word.
DWDIV(P) divides double word data *S1* by double word data *S2*, and stores the quotient at *D* in double word.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
WDIV(P) DWDIV(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				

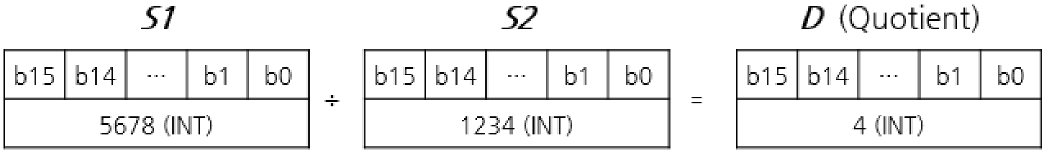


S1	Head address of the device where word/double word data to be divided is stored
S2	Head address of the device where word/double word data for dividing is stored
D	Head address of the device where the quotient (word/double word) is stored

⚠ DWDIV(P) instruction doesn't support device R.

WDIV, WDIVP

- Divides word data assigned to *S1* by word data *S2*, and stores the quotient of word data in device assigned to *D.*

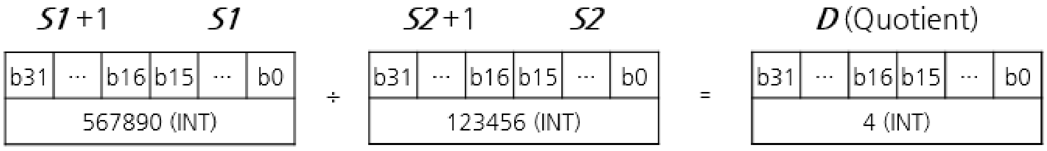


- At *S1* and *S2,* a value from -32768 to 32767 can be specified.

- Unlike DIV(P) instruction, the remainder is not stored when WDIV(P) are executed.

DWDIV, DWDIVP

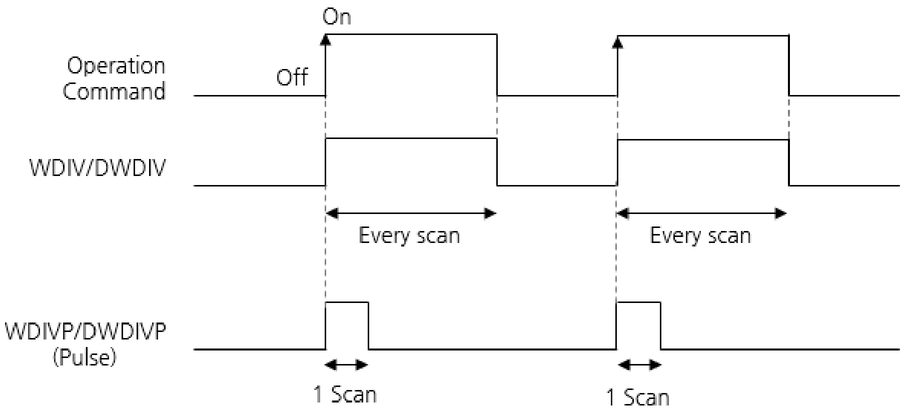
- Divides double word data assigned to *S1* by double word data *S2*, and stores the quotient value of double word data in device assigned to *D.*



- At *S1* and *S2,* a value from -214748648 to 2147483647 can be specified.

- Unlike DDIV(P) instruction, the remainder is not stored when DWDIV(P) are executed.

Execution Condition



Operation Error

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)

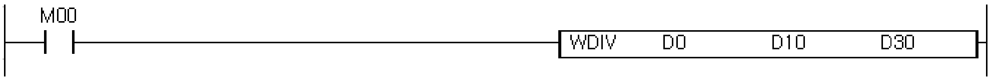
F110 turns ON when the divisor is 0. (When *S2* is 0.)

Program Example

WDIV, WDIVP

The program executes division of D0 by D10 when M00 is ON, then store the quotient value in D30.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
WDIV	D0	D10	D30

2.1.7 WSUM, WSUMP

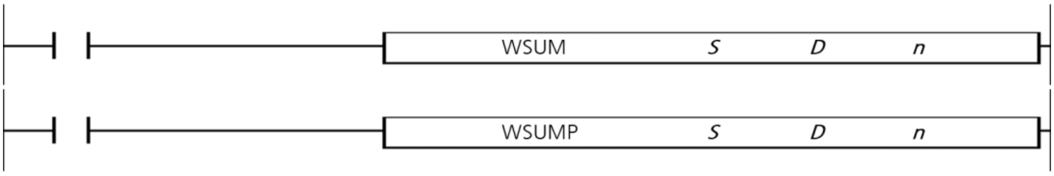
Supported PLC series

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

Function

Adds up *n* word data blocks from the device assigned to *S*. The result is stored in double word data in the device assigned to *D*.

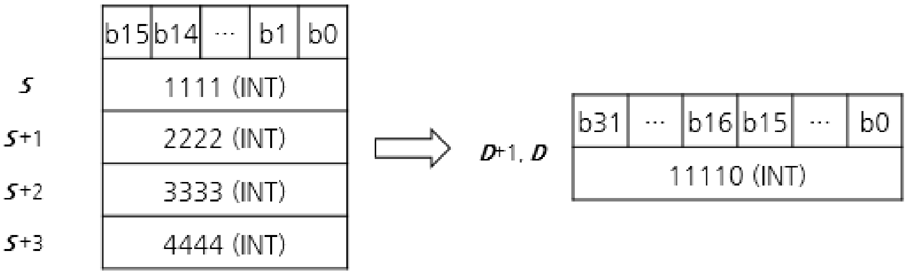
Instruction		Valid device type														Steps	Flag		
		M	X	Y	K	L	F	T	C	Z	R	Q	D	@D	Constant		Error	Zero	Carry
WSUM(P)	*S*	O	O	O	O	O	O	O	O	O	O	O	O	O	-	4	O	-	-
	D	O	O	O	O	O	-	O	O	O	O	O	O	O	-				
	n	O	-	O	O	O	-	O	O	O	O	O	O	O	O				



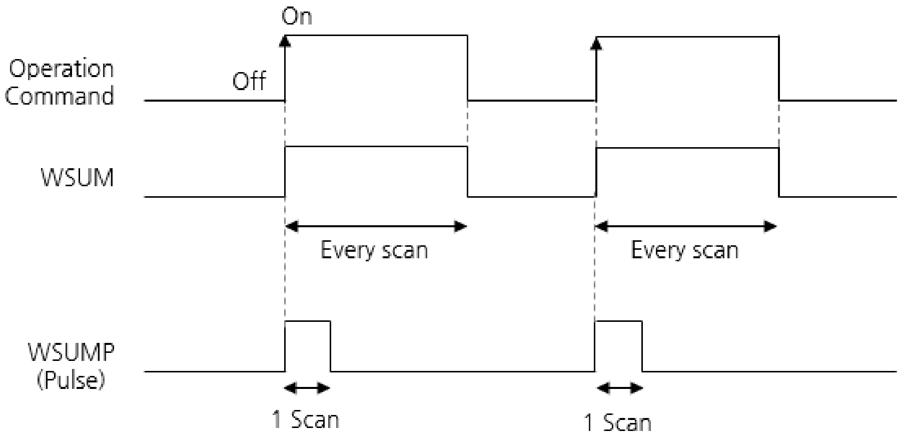
S	Head address of device which counts the total number of word data stored in device
D	Head address of device which stores the result (double word) of operation
n	Number of word data blocks

WSUM, WSUMP

- Adds up *n* word data blocks from the device assigned to *S*, and stores the result value of double word data assigned to *D*.



Execution Condition



Operation Error

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)

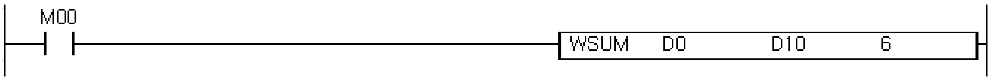
F110 turns ON when the points specified in *n* exceed those of the corresponding device specified *S*.

Program Example

WSUM, WSUMP

Program executes count total word data of D0 to D5 and stores the result value in D10 to D11.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
WSUM	D0	D10	6

2.1.8 BADD, BADDP, DBADD, DBADDP

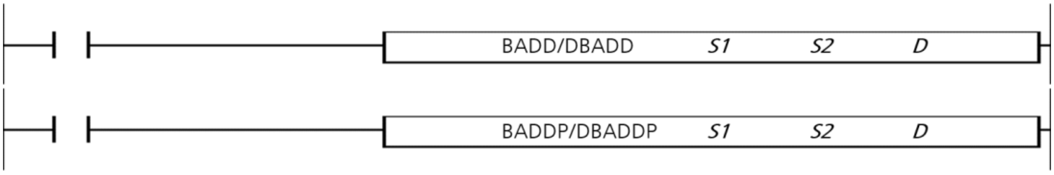
Supported PLC series

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

BADD(P) adds 4-digit BCD data *S1* to 4-digit BCD data *S2,* and stores the result at *D* in 4-digit BCD.
DBADD(P) adds 8-digit BCD data *S1* to 8-digit BCD data *S2,* and stores the result at *D* in 8-digit BCD.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry (NEW)
BADD(P) DBADD(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	O
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S1	Head address of the device where 4/8-digit BCD data to be added to is stored
S2	Head address of the device where 4/8-digit BCD data for adding is stored
D	Head address of the device where the addition operation result in 4/8-digit BCD data will be stored

BADD, BADDP

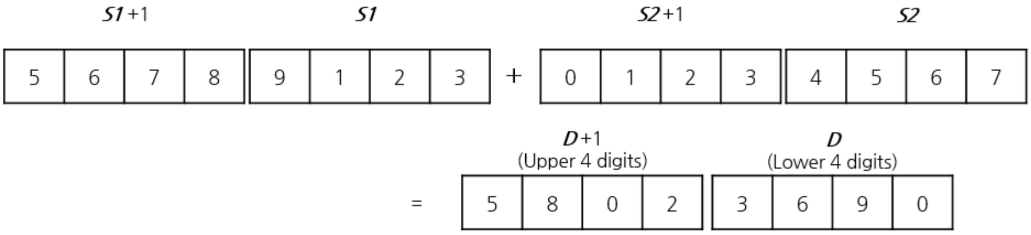
- Adds 4-digit BCD data assigned to *S1* to 4-digit BCD data assigned to *S2,* and stores the result of 4-digit BCD data in the device assigned to *D*.

	S1						*S2*					*D*	
5	6	7	8	+	1	2	3	4	=	6	9	1	2

- 0 to 9999 (4-digit BCD) can be assigned to *S1, S2* and *D*.
- If the result of the addition operation exceeds 9999 (4-digit BCD), the higher digits are ignored.

DBADD, DBADDP

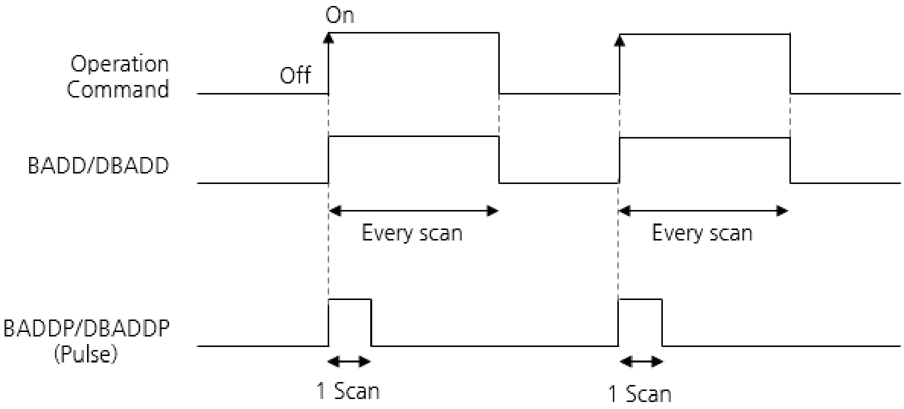
- Adds 8-digit BCD data assigned to *S1* and 8-digit BCD data assigned to *S2*, and stores the result of 8-digit BCD data in the device assigned to *D*.



- 0 to 99999999 (8-digit BCD) can be assigned to *S1, S2* and *D*.

- If the result of the addition operation exceeds 99999999 (8-digit BCD), the higher digits are ignored.

Execution Condition



Operation Error

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)

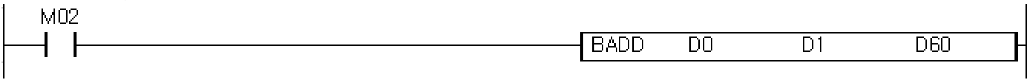
F110 turns ON when *S1* or *S2* is not BCD data in BADD(P)/DBADD(P) instruction.

Program Example

BADD, BADDP

The program executes addition of BCD data D0 and BCD data D1 when operation command M02 is ON, then output the result value at D60.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M02		
BADD	D0	D1	D60

2.1.9 BSUB, BSUBP, DBSUB, DBSUBP

Supported PLC series

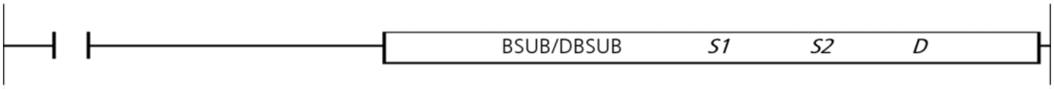
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

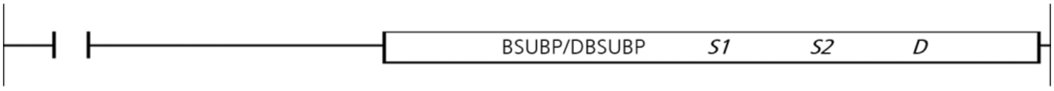
Function

BSUB(P) subtracts 4-digit BCD data *S2* from 4-digit BCD data *S1,* and stores the result at *D* in 4-digit BCD.

DBSUB(P) subtracts 8-digit BCD data *S2* from 8-digit BCD data *S2,* and stores the result at *D* in 8-digit BCD.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero (New)	Carry (New)
BSUB(P) DBSUB(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	O	O
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				



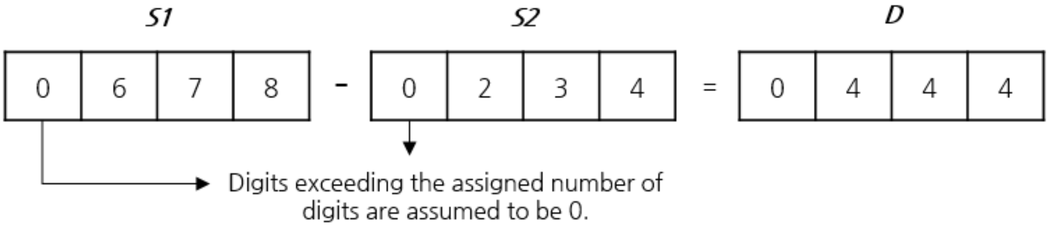


S1	Head address of the device where 4/8-digit BCD data to be subtracted is stored
S2	Head address of the device where 4/8-digit BCD data for subtracting is stored
D	Head address of the device where the subtraction operation result in 4/8-digit BCD data will be stored

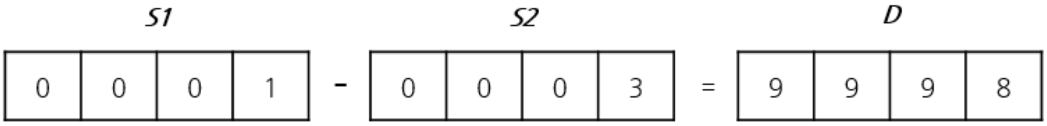
⚠ DBSUB(P) instruction doesn't support device R.

BSUB, BSUBP

- Subtracts 4-digit BCD data assigned to *S2* from 4-digit BCD data assigned to *S1,* and stores the result of 4-digit BCD data in the device assigned to *D*.


S1
S2
D
0
6
7
8
-
0
2
3
4
=
0
4
4
4
Digits exceeding the assigned number of digits are assumed to be 0.

- 0 to 9999 (4-digit BCD) can be assigned to *S1, S2* and *D*.
- If 4-digit BCD data assigned to *S2* is greater than 4-digit BCD data assigned to *S1*, the result is underflowed.



DBSUB, DBSUBP

- Subtracts 8-digit BCD data assigned to *S2* from 8-digit BCD data assigned to *S1,* and stores the result of 8-digit BCD data in the device assigned to *D*.

S1+1 (Upper 4 digits)				*S1* (Lower 4 digits)					*S2*+1 (Upper 4 digits)				*S2* (Lower 4 digits)			
5	6	7	8	9	1	2	3	–	0	1	2	3	4	5	6	7

	D+1 (Upper 4 digits)				*D* (Lower 4 digits)			
=	0	0	0	0	0	0	4	5

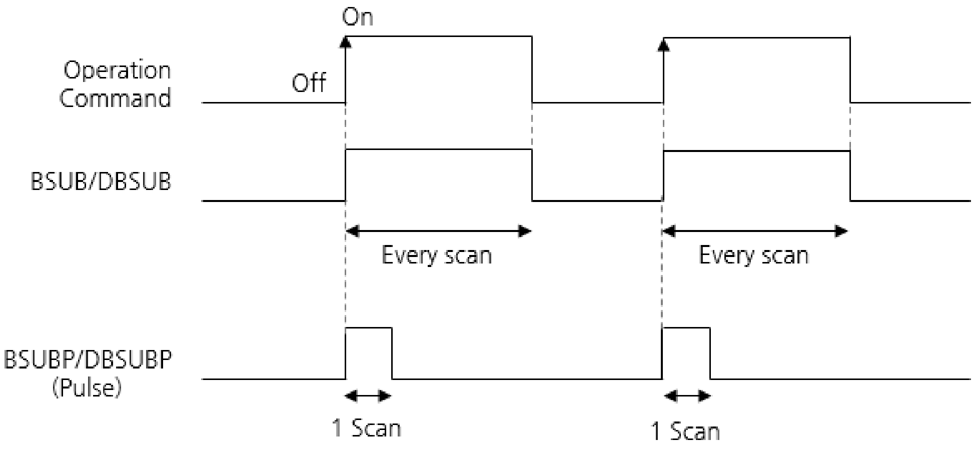
- 0 to 99999999 (8-digit BCD) can be assigned to *S1, S2* and *D*.

- If 8-digit BCD data assigned to *S2* is greater than 8-digit BCD data assigned to *S1*, the underflow occurs.

S1+1 (Upper 4 digits)				*S1* (Lower 4 digits)					*S2*+1 (Upper 4 digits)				*S2* (Lower 4 digits)			
1	2	3	4	5	6	7	8	-	1	2	3	4	5	6	7	9

	D+1 (Upper 4 digits)				*D* (Lower 4 digits)			
=	9	9	9	9	9	9	9	9

Execution Condition



Operation Error

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)

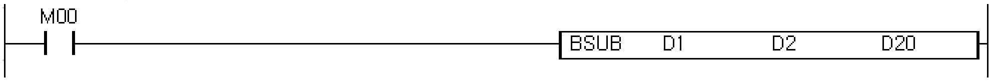
F110 turns ON when *S1* or *S2* is not BCD data in BSUB(P)/DBSUB(P) instruction.

Program Example

BSUB

The program executes subtraction of D2 from D1 when operation command M00 is ON, then output the result value at D20.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
BSUB	D1	D2	D20

2.1.10 BMUL, BMULP, DBMUL, DBMULP

Supported PLC series

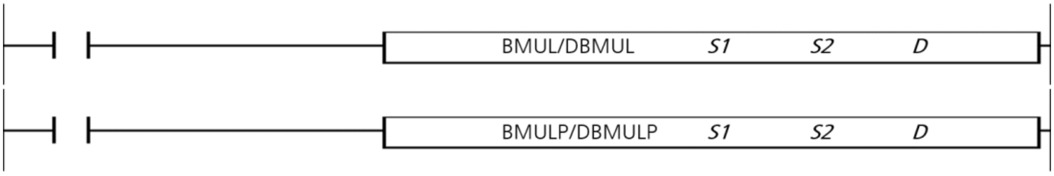
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

BMUL(P) multiplies 4-digit BCD data *S1* and 4-digit BCD data *S2,* and stores the result at *D* in double word.

DBMUL(P) multiplies 8-digit BCD data *S1* and 8-digit BCD data *S2,* and stores the result at *D* in 2 double words.

Introductions		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry (NEW)
BMUL(P) DBMUL(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				

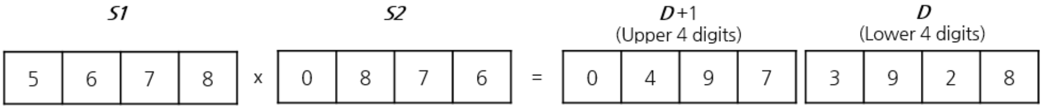


S1	Head address of the device where 4/8-digit data to be multiplied is stored
S2	Head address of the device where 4/8-digit data for multiplying is stored
D	Head address of the double word/2 double word devices where the multiplication operation result will be stored

⚠ DBMUL(P) instruction doesn't support device R.

BMUL, BMULP

- Multiplies 4-digit BCD data assigned to *S1* and 4-digit BCD data assigned to *S2.* The result is stored in the double word device assigned to *D*.
- Stores lower 4 digits in device assigned to *D*, upper 4 digits in device assigned to *D*+1.



- 0 to 9999 (4-digit BCD) can be assigned to *S1* and *S2.*

DBMUL, DBMULP

- Multiplies the 8-digit BCD data assigned to *S1* and 8-digit BCD data assigned to *S2*. The result is stored in the 2 double word device assigned to *D*.

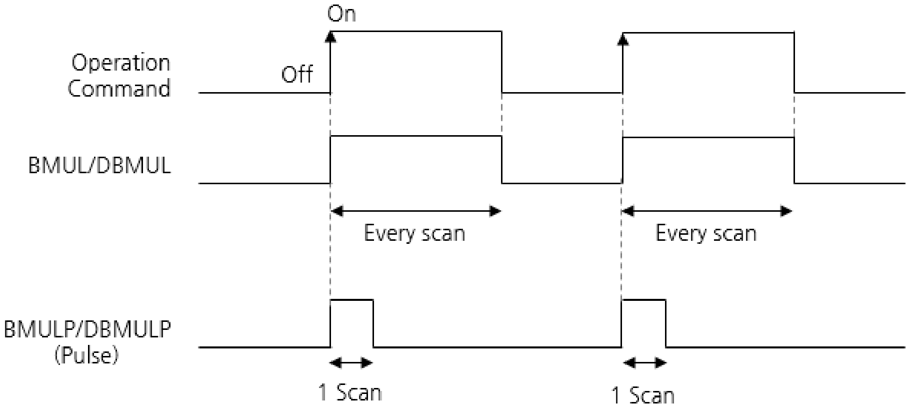
S1+1	*S1*		*S2*+1	*S2*
9 9 9 9	9 9 9 9	x	9 9 9 9	9 9 9 9

	D+3	*D*+2	*D*+1	*D*
=	9 9 9 9	9 9 9 8	0 0 0 0	0 0 0 1

- 0 to 99999999 (8-digit BCD) can be assigned to *S1* and *S2*.

⚠ Results are stored in 2 double word devices. Therefore, be cautious when you assign devices to *D*.

Execution Condition



Operation Error

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)

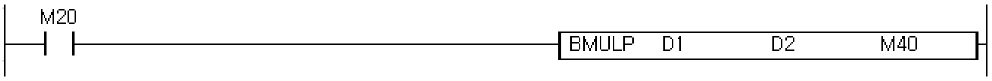
F110 turns ON when the *S1* or *S2* is not BCD data in BMUL(P)/DBMUL(P) instruction.

Program Example

BMULP

The program executes multiplication of BCD data D1 and BCD data D2 when operation command M20 is ON, then store the result value of double word data (lower 4 digits in M40, upper 4 digits in M50).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M20		
BMULP	D1	D2	M40

2.1.11 BDIV, BDIVP, DBDIV, DBDIVP

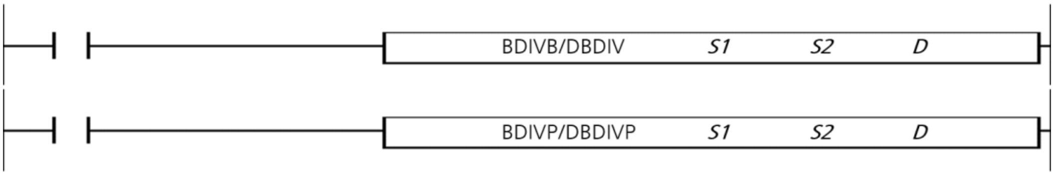
Supported PLC series

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

BDIV(P) divides 4-digit BCD data *S1* by 4-digit BCD data *S2,* and stores the quotient at *D* and the remainder at *D*+1 in 4-digit BCD.
DBDIV(P) divides 8-digit BCD data *S1* by 8-digit BCD data *S2,* and stores the quotient at *D* and the remainder at *D*+1 in 8-digit BCD.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero (New)	Carry
BDIV(P) DBDIV(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	O	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O*	O	O	O	-				

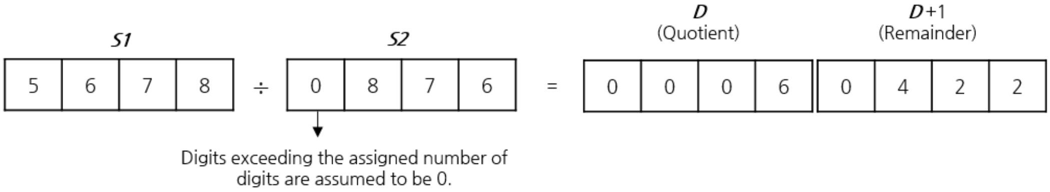


S1	Head address of the device where 4/8-digit data to be divided is stored
S2	Head address of the device where 4/8-digit data for dividing is stored
D	Head address of the device where the quotient (4/8-digit BCD) is stored
D+1	Head address of the device where the remainder (4/8-digit BCD) is stored

⚠ DBDIV(P) instruction doesn't support device R.

BDIV, BDIVP

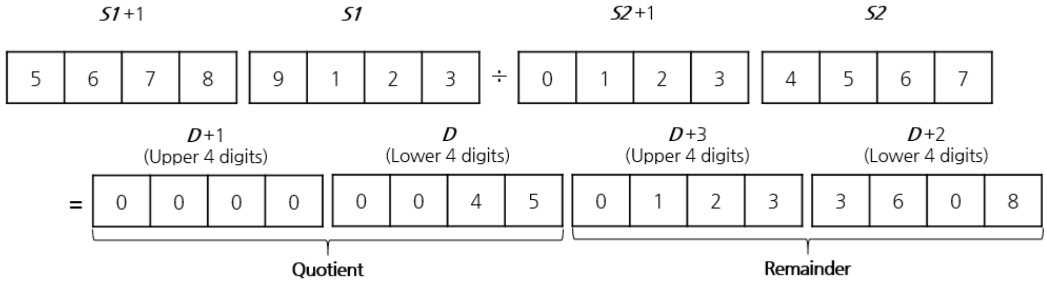
- Divides 4-digit BCD data assigned to *S1* by 4-digit data assigned to *S2,* and stores the quotient in device assigned to *D* and the remainder in device assigned to *D*+1.


S1
5
6
7
8
÷
S2
0
8
7
6
=
D
(Quotient)
0
0
0
6
D+1
(Remainder)
0
4
2
2
Digits exceeding the assigned number of digits are assumed to be 0.

- Uses 32 bits to store quotient and remainder according to the result of the division operation.
  Quotient (4-digit BCD): Stored at the lower 16 bits.
  Remainder (4-digit BCD): Stored at the upper 16 bits.

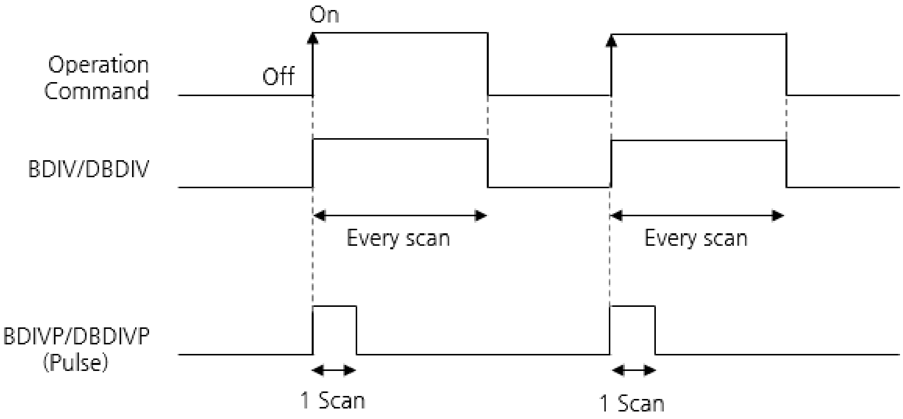
DBDIV, DBDIVP

- Divides BCD 8-digit data assigned to *S1* by BCD 8-digit data assigned to *S2,* and stores the quotient in device assigned to *D* and the remainder in device assigned to *D*+1.



- Uses 64 bits to store quotient and remainder according to the result of the division operation.
  Quotient (8-digit BCD): Stored at the lower 32 bits.
  Remainder (8-digit BCD): Stored at the upper 32 bits.

Execution Condition



Operation Error

Error Flag (F110)

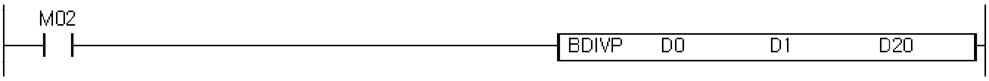
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)
F110 turns ON for 1 scan when divisor is 0. (= When *S2* is 0)

Program Example

BDIVP

The program executes divide BCD data D0 by BCD data D1 when operation command M02 is ON, then store the quotient in D20 and the remainder in D21.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M02		
BDIVP	D0	D1	D20

2.1.12 EADD, EADDP

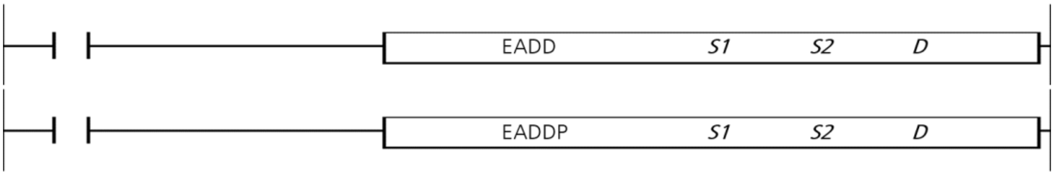
Supported PLC series

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

Function

Adds 32-bit floating-point number assigned to *S1* and 32-bit floating-point number assigned to *S2*, and stores the result data of the operation in the device assigned to *D*.

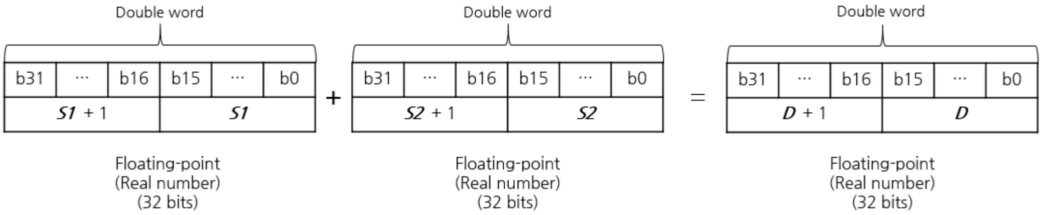
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EADD(P)	*S1*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S1	Data to be added or head address of the device where the data to be added is stored (32-bit floating-point number)
S2	Data for adding or head address of the device where the data for adding is stored (32-bit floating-point number)
D	Head address of the device where the operation result is stored (32-bit floating-point number)

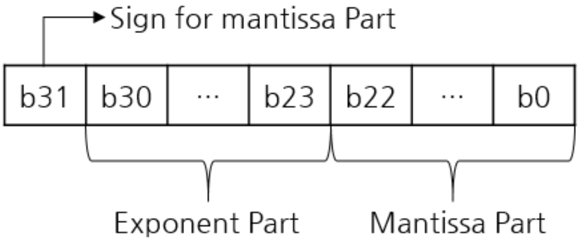
EADD, EADDP

- Adds 32-bit floating-point number assigned to *S1* and 32-bit floating-point number assigned to *S2*. The result is stored in the device assigned to *D*.



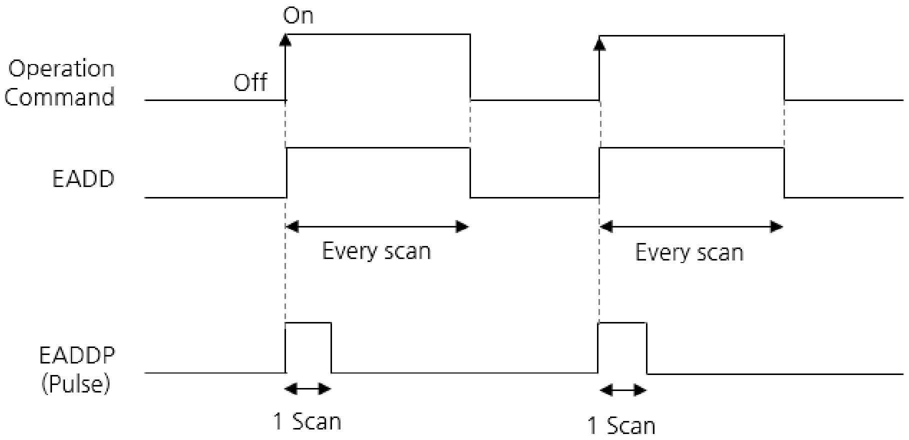
- Values which can be assigned and stored to *S1*, *S2* and *D* are as below:
  $2^{-127} \leq$ | Assigned value (stored value) | $\leq 2^{128}$

- At CICON, you can input 32-bit floating-point number at device memory in following format.



- At exponent part (bit 23 - bit 30), value HFF and H00 are not used.
- Valid value of exponent part is from -126 to 127.
- To express “0” with floating-point, set bit 0 - bit 31 as “0”.

Execution Condition



Operation Error

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)

F110 turns ON when the specified device value is not within the following range:
$2^{-127} \leq |\text{Assigned value (stored value)}| \leq 2^{128}$

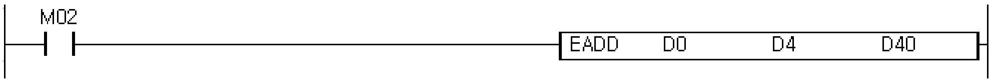
F110 turns ON when the operation result exceeds the following range:
$2^{128} \leq |\text{Operation result}|$

Program Example

EADD

The program adds the 32-bit floating-point number in D0 and the 32-bit floating-point number in D4 when operation command M02 is ON, then the result is stored in D40.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M02		
EADD	D0	D4	D40

2.1.13 ESUB, ESUBP

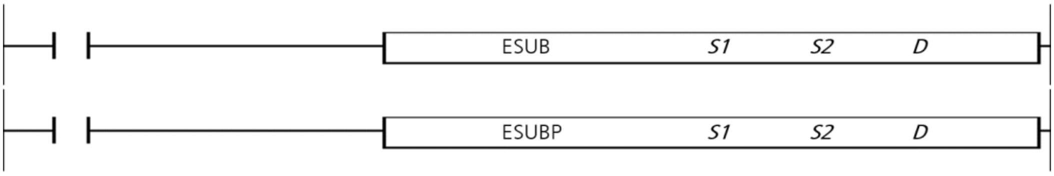
Supported PLC series

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

Function

Subtracts 32-bit floating-point number assigned to *S2* from 32-bit floating-point number assigned to *S1,* and stores the result data of the operation in the device assigned to *D*.

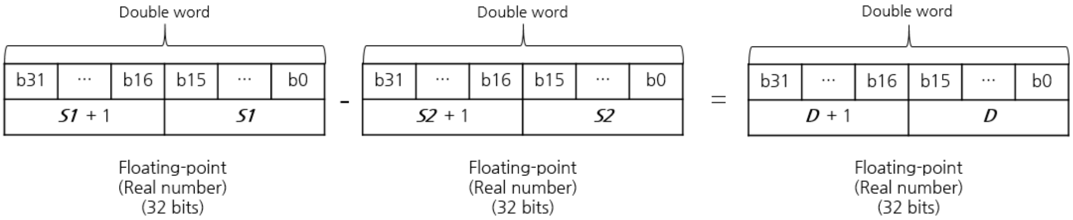
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ESUB(P)	*S1*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S1	Data to be subtracted or head address of the device where the data to be subtracted is stored (32-bit floating-point number)
S2	Data for subtracting or head address of the device where the data for subtracting is stored (32-bit floating-point number)
D	Head address of the device where the operation result is stored (32-bit floating-point number)

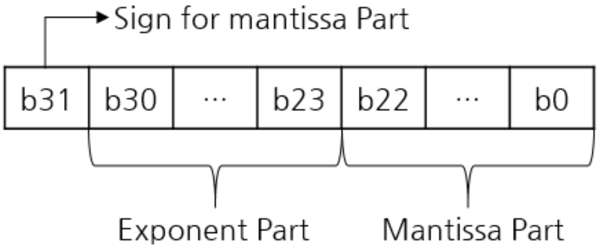
ESUB, ESUBP

- Subtracts 32-bit floating-point number assigned to *S2* from 32-bit floating-point number assigned to *S1,* and stores the result value in the device assigned to *D*.


Double word
b31
…
b16
b15
…
b0
S1 + 1
S1
Floating-point (Real number) (32 bits)
-
Double word
b31
…
b16
b15
…
b0
S2 + 1
S2
Floating-point (Real number) (32 bits)
=
Double word
b31
…
b16
b15
…
b0
D + 1
D
Floating-point (Real number) (32 bits)

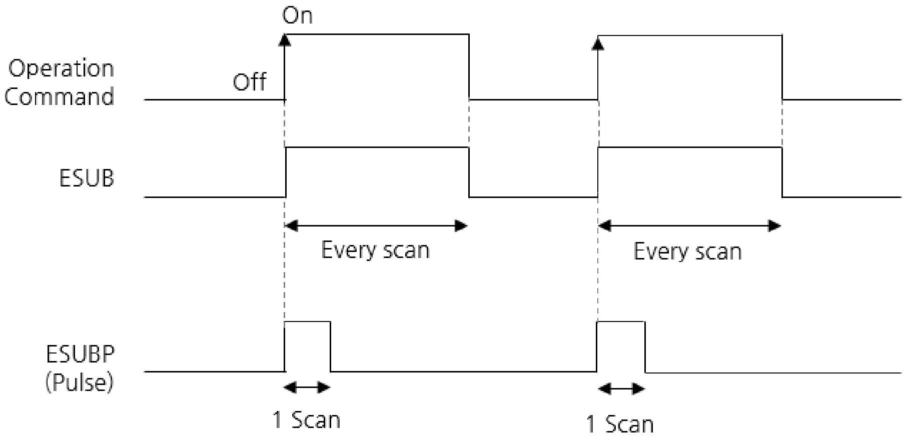
- Values which can be assigned and stored to *S1, S2* and *D* are as below:
  2^{-127} ≤ | Assigned value (stored value) | ≤ 2^{128}

- At CICON, you can input 32-bit floating-point number at device memory in following format.



- At exponent part (bit 23 - bit 30), value HFF and H00 are not used.
- Valid value of exponent part is from -126 to 127.
- To express “0” with floating-point, set bit 0 - bit 31 as “0”.

Execution Condition



Operation Error

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)

F110 turns ON when the specified device value is not within the following range:
$2^{-127} \leq$ | Assigned value (stored value) | $\leq 2^{128}$

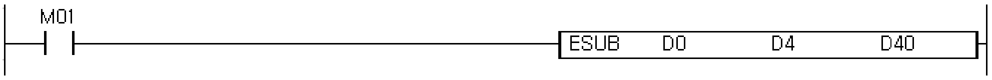
F110 turns ON when the operation result exceeds the following range:
$2^{128} \leq$ | Operation result |

Program Example

ESUB

The program subtracts the 32-bit floating-point number in D4 from the 32-bit floating-point number in D0 when operation command M01 is ON, then the result is stored in D40.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M01		
ESUB	D0	D4	D40

2.1.14 EMUL, EMULP

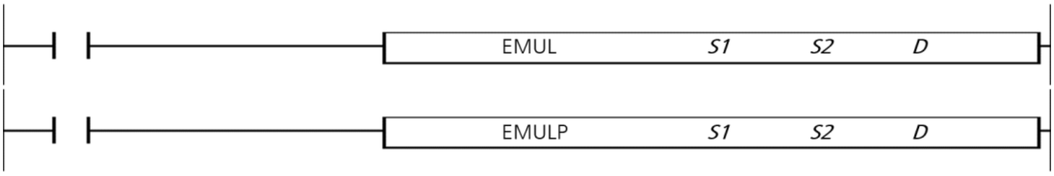
Supported PLC series

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

Function

Multiplies 32-bit floating-point number assigned to *S1* and 32-bit floating-point number assigned to *S2,* and stores the result data of the operation in the device assigned to *D*.

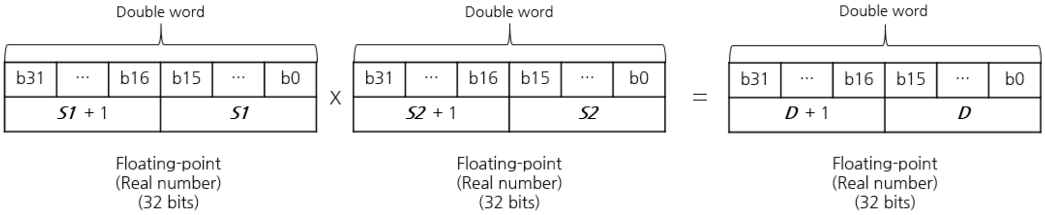
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EMUL(P)	*S1*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S1	Data to be multiplied or head address of the device where the data to be multiplied is stored (32-bit floating-point number)
S2	Data for multiplying or head address of the device where the data for multiplying is stored (32-bit floating-point number)
D	Head address of the device where the operation result is stored (32-bit floating-point number)

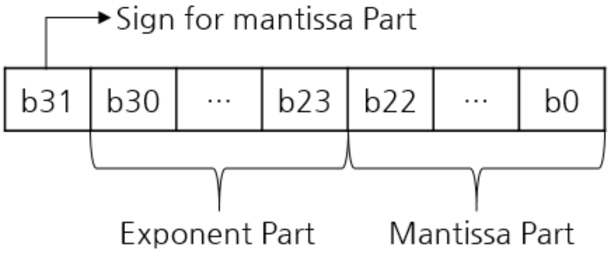
EMUL, EMULP

- Multiplies 32-bit floating-point number assigned to *S1* and 32-bit floating-point number assigned to *S2,* and stores the result in the device assigned to *D*.



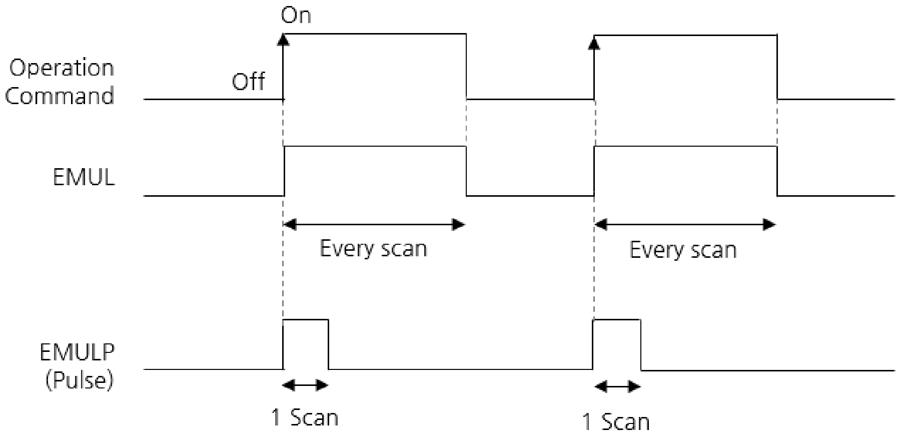
- Values which can be assigned to *S1*, *S2* and *D* and which can be stored are as below: $2^{-127} \leq$ I Assigned value (stored value) I $\leq 2^{128}$

- At CICON, you can input 32-bit floating-point number at device memory in following format.



- At exponent part (bit 23 - bit 30), value HFF and H00 are not used.
- Valid value of exponent part is from -126 to 127.
- To express “0” with floating-point, set bit 0 - bit 31 as “0”.

Execution Condition



Operation Error

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)

F110 turns ON when the specified device value is not within the following range:
$2^{-127} \leq$ | Assigned value (stored value) | $\leq 2^{128}$

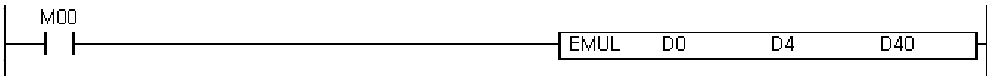
F110 turns ON when the operation result exceeds the following range:
$2^{128} \leq$ | Operation result |

Program Example

EMUL

The program multiplies the 32-bit floating-point number in D0 and the 32-bit floating-point number in D4 when operation command M00 is ON, then the result is stored in D40.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
EMUL	D0	D4	D40

2.1.15 EDIV, EDIVP

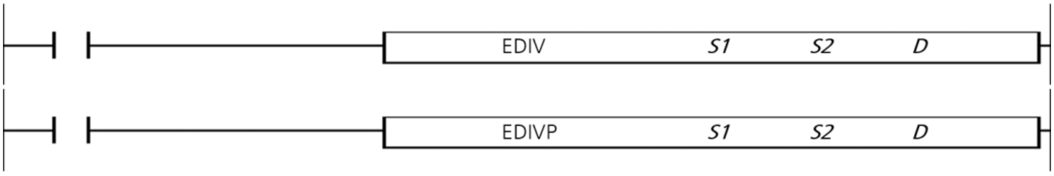
Supported PLC series

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

Function

Divides 32-bit floating-point number assigned to ***S1*** by 32-bit floating-point number assigned to ***S2,*** and stores the result data of the operation in the device assigned to ***D***.

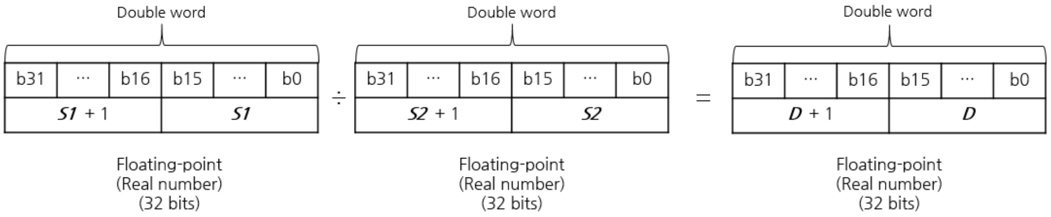
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EDIV(P)	*S1*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-		-	O	-	O	O	O	-				



S1	Data to be divided or head address of the device where the data to be divided is stored (32-bit floating-point number)
S2	Data for dividing or head address of the device where the data for dividing is stored (32-bit floating-point number)
D	Head address of the device where the operation result is stored (32-bit floating-point number)

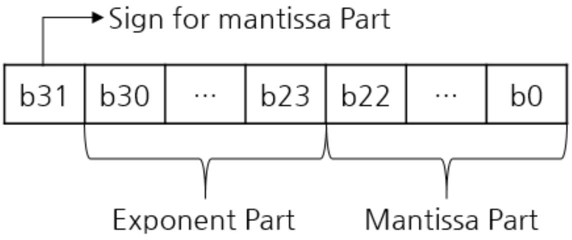
EDIV, EDIVP

- Devides 32-bit floating-point number assigned to ***S1*** by 32-bit floating-point number assigned to ***S2,*** and result is stored in the device assigned to ***D***.



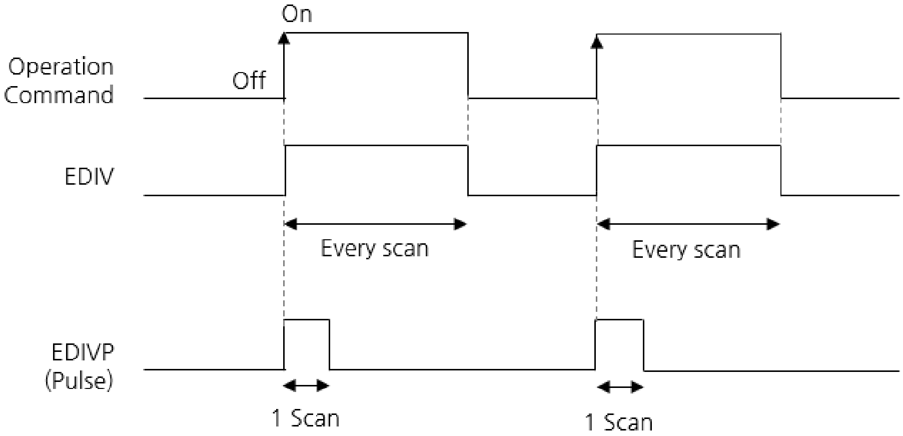
- Values which can be assigned and stored to ***S1, S2*** and ***D*** are as below:
  2^{-127} ≤ l Assigned value (stored value) l ≤ 2^{128}

- At CICON, you can input 32-bit floating-point number at device memory in following format.



- At exponent part (bit 23 - bit 30), value HFF and H00 are not used.
- Valid value of exponent part is from -126 to 127.
- To express “0” with floating-point, set bit 0 - bit 31 as “0”.

Execution Condition



Operation Error

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)

F110 turns ON when divisor is 0. (When *S2* is 0)

F110 turns ON when the specified device value is not within the following range:
$2^{-127} \leq |\text{ Assigned value (stored value) }| \leq 2^{128}$

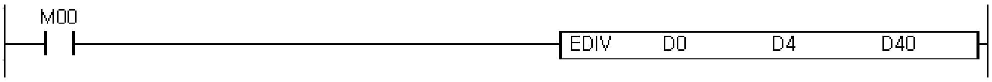
F110 turns ON when the operation result exceeds the following range:
$2^{128} \leq |\text{ Operation result }|$

Program Example

EDIV

Program divides the 32-bit floating-point number in D0 by the 32-bit floating-point number in D4 when operation command M00 is ON, then the result is stored in D40.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M02		
EDIV	D0	D4	D40

2.1.16 INC, INCP, DINC, DINCP

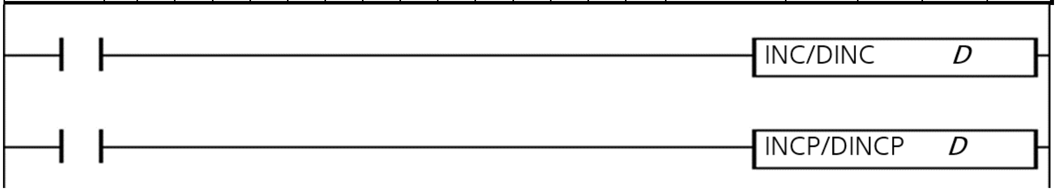
Supported PLC Series

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

Adds 1 to word/double word data in the device assigned to *D*, and stores the increased value of word/double word in the device assigned to *D*.

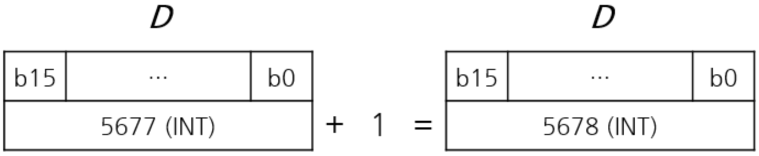
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
INC(P) DINC(P)	*D*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	-	2	O	-	-



D	Head address of word/double word device for INC(P)/DINC(P) operation

INC, INCP

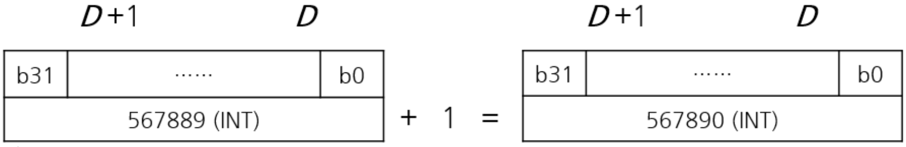
- Adds 1 to word data in the device assigned to *D*.



⚠ When INC, INCP operation is performed at the device assigned to *D*, which contains 32767, the value -32768 is stored at the device assigned to *D*.

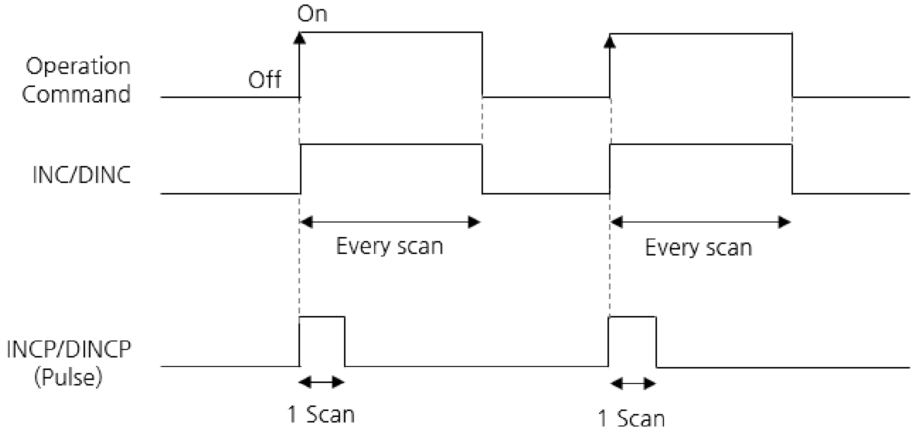
DINC, DINCP

- Adds 1 to double word data in the device assigned to *D*.



⚠ When DINC, DINCP operation is performed for the device assigned to *D*, which contains 2147483647, the value -2147483648 is stored at the device assigned to *D*.

Execution Condition



Operation Error

Error flag (F110)

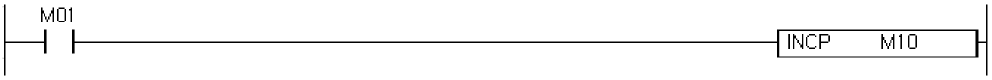
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

INCP

In the program, the value of word device M10 (M10~M1F) will increase by 1 when operation command M01 goes from OFF to ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	M01
INCP	M10

2.1.17 DEC, DECP, DDEC, DDECP

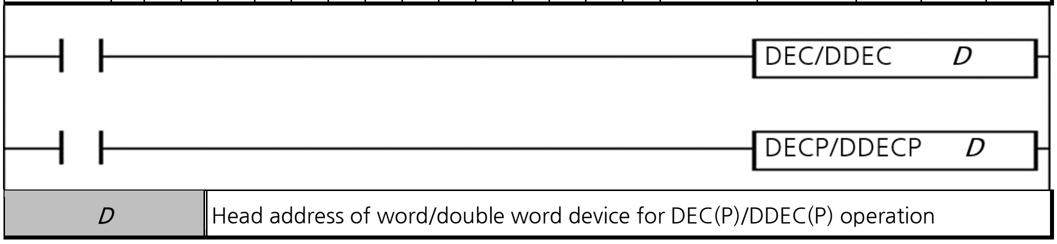
Supported PLC Series

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

Subtracts 1 from word/double word data in the device assigned to *D*, and stores the decreased value of word/double word data in the device assigned to *D*.

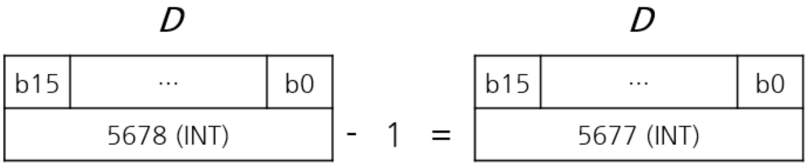
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
DEC(P) DDEC(P)	*D*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-	2	O	-	-



D	Head address of word/double word device for DEC(P)/DDEC(P) operation

DEC, DECP

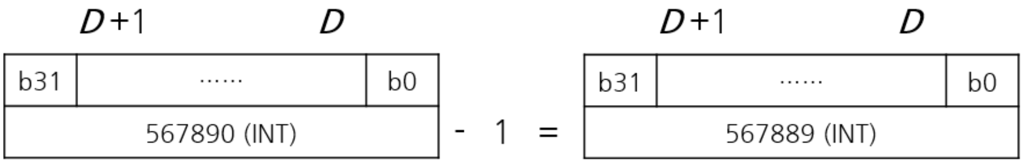
- Subtracts 1 from word data in the device assigned to *D*.



⚠ When DEC, DECP operation is performed at the device assigned to *D*, which contains -32768, the value 32767 is stored at the device assigned to *D*.

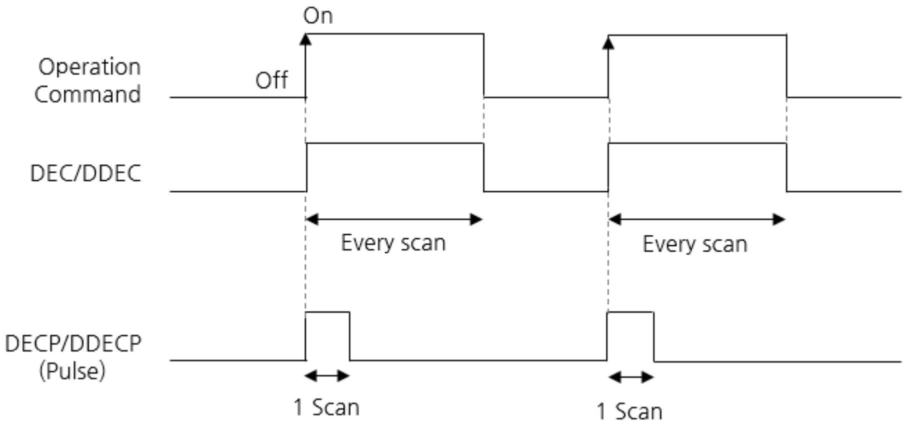
DDEC, DDECP

- Subtracts 1 from double word data in the device assigned to *D*.



⚠ When DDEC, DDECP operation is performed at the device assigned to *D*, which contains -2147483648, the value 2147483647 is stored at the device assigned to *D*.

Execution Condition



Operation Error

Error flag (F110)

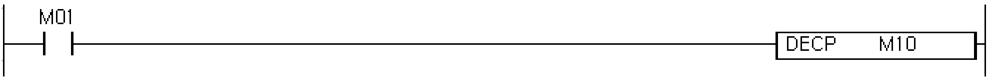
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DECP

In the program, the value of word device M10 (M10~M1F) will decrease by 1 when operation command M01 goes from OFF to ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	M01
DECP	M10

2.1.18 PWR, PWRP

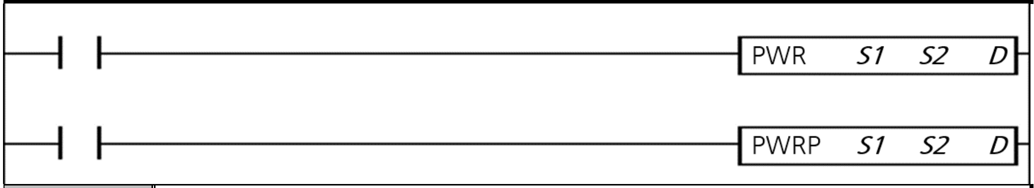
Supported PLC Series

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

Function

PWR(P) instructions calculate base 16-bit INT data stored in word device *S1*, raised to the power of 16-bit BIN data stored in word device *S2*. The result is stored in the double word device *D*.

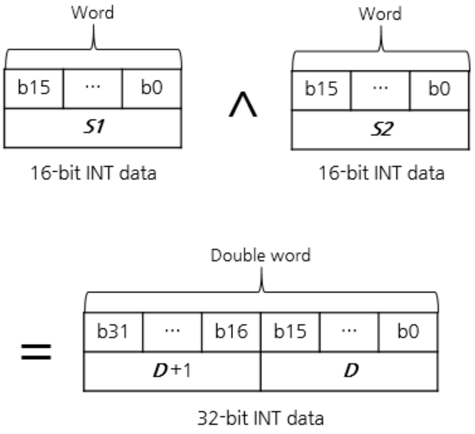
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PWR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O-	-	O	-	O	O	O	-				



S1	Constant or address of word device where base is stored.
S2	Constant or address of word device where exponent is stored.
D	Head address of double word device where the calculation result will be stored.

PWR, PWRP

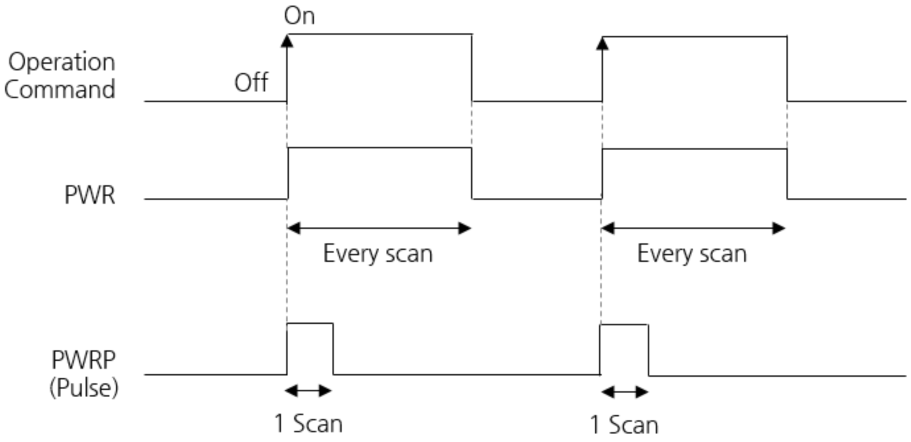
- PWR(P) instructions calculate base 16-bit INT data stored in word device *S1*, raised to the power of 16-bit INT data stored in word device *S2*. The result is stored in the double word device *D*.



- The range of the value assigned to word device *S1* and *S2* is as below:
  -32768 ≤ Value assigned to device *S1*, *S2* ≤ 32767

- The range of the result can be stored in double word device *D* is as below:
  -2147483648 ≤ Value assigned to device *D* ≤ 2147483647
  It means that even if you calculate 32767^{32767}, the result is “2147483647”.

Execution Condition



Operation Error

Error flag (F110)

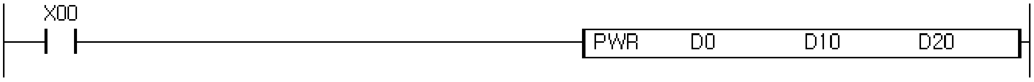
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

PWR

When X00 turns ON, PWR instruction calculates the base stored in D0 raised to the power of the value assigned to D10. Then, the result is stored in double word device D20 (D20~D21).

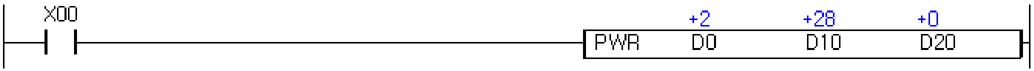
Ladder Diagram (LD)



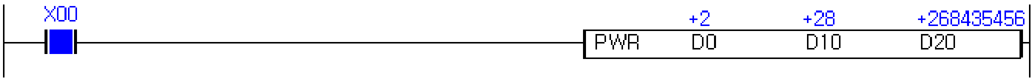
Instruction List (IL)

Instruction	Device		
LD	X00		
PWR	D0	D10	D20

- “2” is assigned to word device D0 as a base. “28” is assigned to word device D10 as an exponent.



- When X00 turns ON, PWR instruction is executed. Then, the instruction calculates 2^{28}. The result “268435456” is stored in double word device D20 (D20~D21).



2.1.19 EPWR, EPWRP

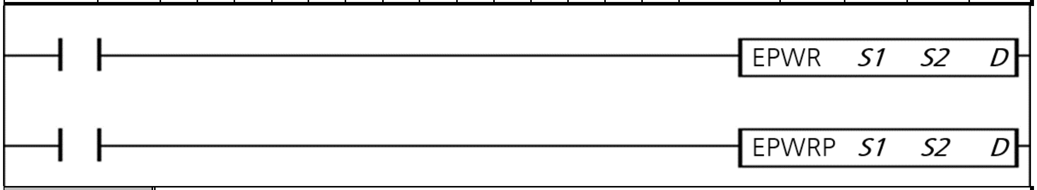
Supported PLC Series

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

Function

EPWR(P) instructions calculate base 32-bit floating-point number stored in double word device *S1* raised to the power of 32-bit floating-point number stored in *S2*. The result is stored in the double word device *D*.

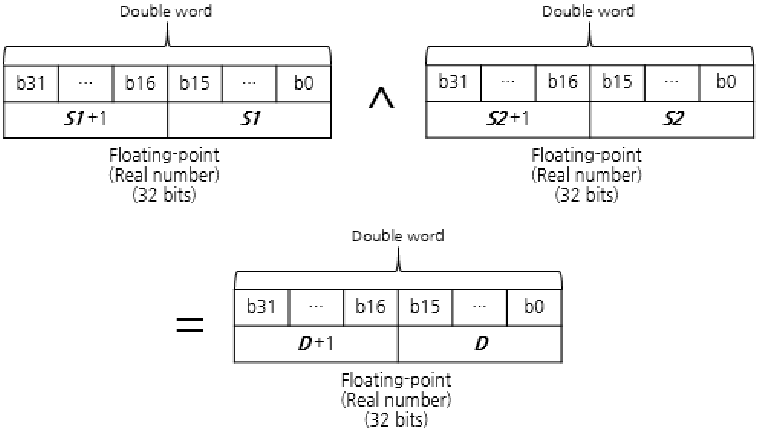
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EPWR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O-	-	O	-	O	O	O	-				



S1	Constant, floating-point number or head address of double word device where base is stored. (32-bit floating-point number)
S2	Constant, floating-point number or head address of double word device where exponent is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation result will be stored. (32-bit floating-point number)

EPWR, EPWRP

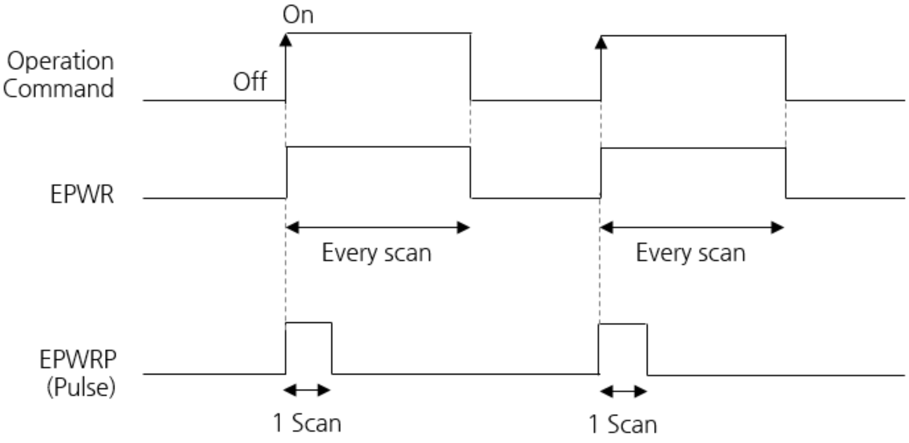
- EPWR(P) instructions calculate base 32-bit floating-point number stored in double word device *S1* raised to the power of 32-bit floating-point number stored in *S2*. The result is stored in the double word device *D*.



- The range of value assigned to double word device *S1, S2* and *D* is as below:
  0, 2^{-127} ≦ Value assigned to *S1, S2* and *D* ≦ 2^{128}

- If the operation result exceeds its range, “1.#INF[1]00” is stored in the device *D*.
- When the value assigned to *S1* is negative and the value assigned to *S2* is non-integer, “1.#QNAN[2]0” is stored in the device *D*.

Execution Condition



Operation Error

Error flag (F110)

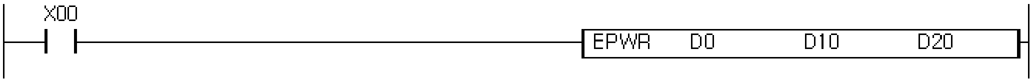
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

EPWR

When X00 turns ON, EPWR instruction calculates the base stored in D0 raised to the power of the value assigned to D10. Then, the result is stored in double word device D20 (D20~D21).

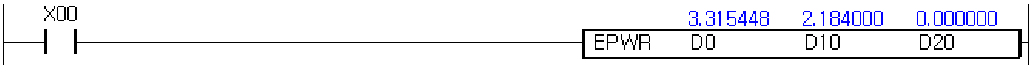
Ladder Diagram (LD)



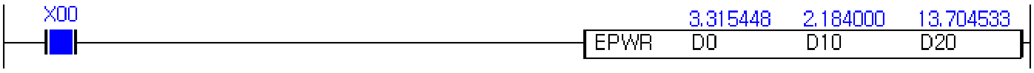
Instruction List (IL)

Instruction	Device		
LD	X00		
EPWR	D0	D10	D20

- “3.315448” is assigned to word device D0 as a base. “2.184000” is assigned to word device D10 as an exponent.



- When X00 turns ON, EPWR instruction is executed. Then, the instruction calculates $3.315448^{2.184000}$. The result “13.704533” is stored in double word device D20 (D20~D21).



[1] INF: Stands for infinity. “1.#INF00” appears when the assigned value cannot be expressed with 32-bit floating-point number.

[2] QNAN: Stands for ‘Quiet Not-a-Number’.

2.1.20 PWM

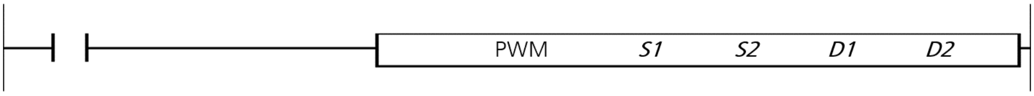
Supported PLC Series

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

Function

PWM instruction outputs pulses set by ON time (*S1*) and the frequency (*S2*) to the output module assigned to *D2*.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PWM	*S1*	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O	5	O	-	-
	S2	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O				
	D1	O	-	O	O	O	-	-	-	-	-	-	-	O	-	-				
	D2	O	-	O	O	O	-	-	-	-	-	-	O	-	-	-				

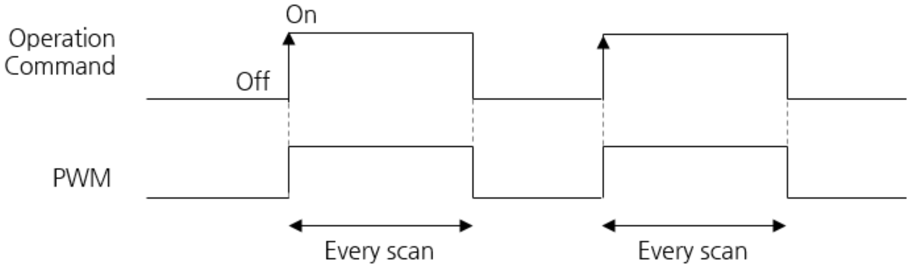


S1	Constant or address of word device where ON time of the pulse to be output is stored. (msec)
S2	Constant or address of word device where frequency of the pulse to be output is stored. (msec)
D1	Head address of 4 word devices where timer value is stored.
D2	Address of bit device to output the pulses.

PWM

- PWM instruction outputs pulses set by ON time (*S1*) and the frequency (*S2*) to the output module assigned to *D2*.
- Do not manipulate the value stored in *D1*, *D1* +1, *D1* +2 and *D1* +3. Timer values are stored in the devices *D1*, *D1* +1, *D1* +2 and *D1* +3 by system.
- The range of value assigned to *S1* and *S2* is as following:
  1 msec ≦ Value assigned to word device *S1* and *S2* ≦ 65535 msec
- The value assigned to *S1* must not exceed the value assigned to *S2*.
- If the values assigned to *S1* and *S2* are equal, the bit device assigned to *D2* is always ON.
- The operation of PWM may be influenced by the scan time since the instruction uses deviation of internal timer.

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)

F110 turns ON for 1 scan under following circumstances:

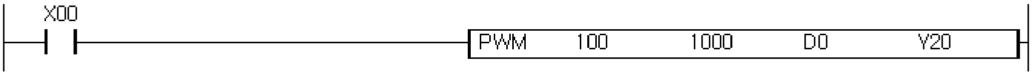
- When the value assigned to *S1* and *S2* is “0”
- When the value assigned to *S1* is greater than *S2.*

Program Example

PWM

When X00 turns ON, PWM instruction is executed. Then, the pulse of 100msec per second is output to Y20. The internal timer value is stored in 4 word devices starting from D0 (D0~D3).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device			
LD	X00			
PWM	100	1000	D0	Y20

2.1.21 RAMP

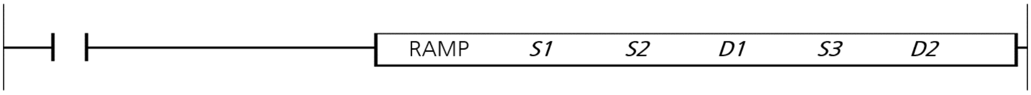
Supported PLC Series

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

Function

RAMP instruction gradually changes the initial value to the final value during the assigned number of execution.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
RAMP	*S1*	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O	6	O	-	-
	S2	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O				
	D1	O	-	O	O	O	-	-	-	-	-	-	-	O	O	-				
	S3	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O				
	D2	O	-	O	O	O	-	-	-	-	-	-	O	O	O	-				



S1	Constant or address of word device where initial value is stored.
S2	Constant or address of word device where final value is stored.
D1	Address of word device where current value is stored.
D1 +1	A word device where the number of execution is stored.
S3	Constant or address of word device where the total number of execution is stored.
D2	Address of bit device to output the operation result.
D2 +1	A bit device to decide the data retention when operation is finished.

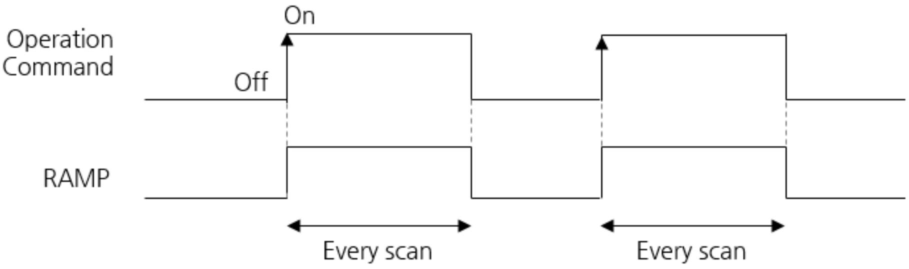
RAMP

- RAMP instruction gradually changes the initial value assigned to *S1* to the final value assigned to *S2* in the number of executions (scans) assigned to *S3*. The current value in the process of value change is stored in *D1*. In *D1* +1, the current number of execution is stored.
- When the current value *D1* reaches the final value *S2*, the bit device assigned to *D2* turns ON.
- If *D2* +1 is ON, the current value stored in *D1* does not change in the following scans.
- If *D2* +1 is OFF, the current value stored in *D1* and *D2* are initialized in the next scan and RAMP instruction operates again.
- The changed value per scan is calculated as below:

$$\text{Changed value per scan} = \frac{(\text{Value assigned to } S2) - (\text{Value assigned to } S1)}{(\text{Value assigned to } S3)}$$

- The range of value assigned to *S1* and *S2* is as below:
  -32768 ≦ Value assigned to *S1* and *S2* ≦ 32767
- The range of value assigned to *S3* is as below:
  1 ≦ Value assigned to *S3* ≦ 32767
- When the value assigned to *S3* is “0”, the instruction does not operate.

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)

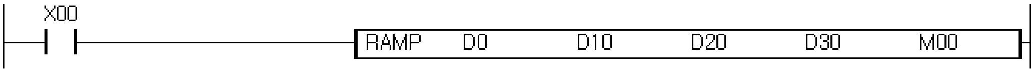
F110 turns ON for 1 scan when the current value did not reach the final value even if the actual execution time has exceeded the value assigned to *S3*.

Program Example

RAMP

When X00 turns ON, RAMP instruction is executed. The initial value is stored in D0. The final value is stored in D10. The current value will be stored in D20 and the number of executions will be stored in D21. The total number of execution is stored in D30. When the current value stored in D20 reaches the final value stored in D10, the bit M00 will be turned ON. If M01 is ON, the RAMP instruction does not operate again from the beginning.

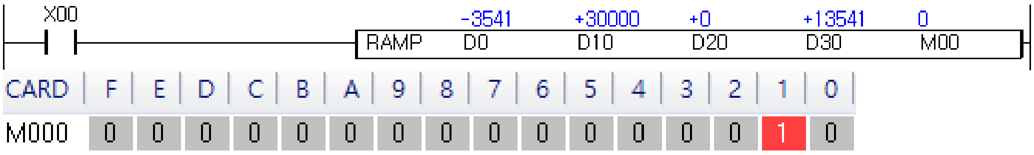
Ladder Diagram (LD)



Instruction List (IL)

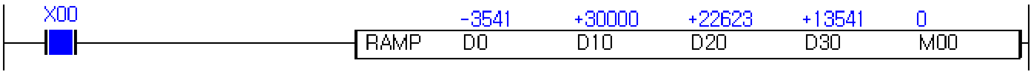
Instruction	Device				
LD	X00				
RAMP	D0	D10	D20	D30	M00

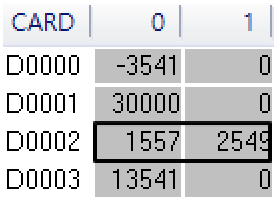
- Initial value “-3541” is stored in D0. Final value “30000” is stored in D10.
- The total number of execution “13541” is stored in D30. It means that when RAMP insturction is executed, initial value “-3541” will be gradually changed into final value “30000” in 13541 scans.
- Since M01 is turned ON, the RAMP operation will be operated only once.



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

- When X00 turns ON, RAMP instruction is executed. Starting from the initial value “-3541” stored in D0, the value is changed gradually in each scan. The current value is stored in D20 and the number of execution is stored in D21.

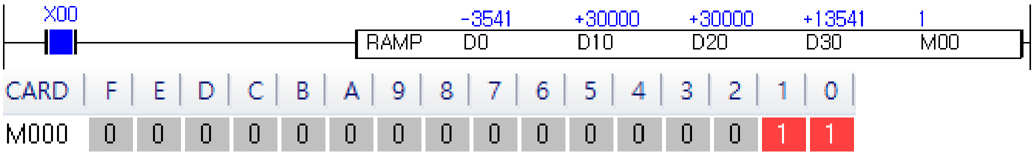




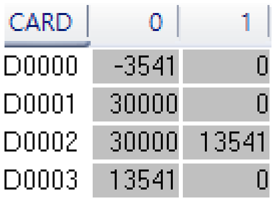
CARD	0	1
D0000	-3541	0
D0001	30000	0
D0002	1557	2549
D0003	13541	0

- Since M01 is ON, the RAMP operation stops and maintains the current value when the current value stored in D20 reaches the final value "30000" stored in D10.

- Since the operation is completed, the bit device M00 turns ON.



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1



CARD	0	1
D0000	-3541	0
D0001	30000	0
D0002	30000	13541
D0003	13541	0

2.1.22 TRAMP

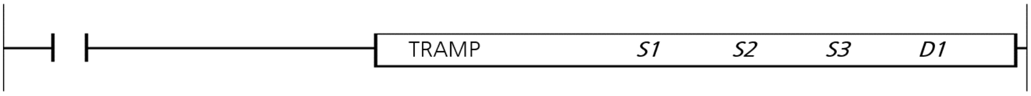
Supported PLC Series

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

Function

TRAMP instruction gradually changes the assigned value during the assigned time value.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
TRAMP	*S1*	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O	5	-	-	-
	S2	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O				
	S3	O	-	O	O	O	-	-	-	-	-	-	-	O	O	O				
	D1	O	-	O	O	O	-	-	-	-	-	-	-	O	O	-				



S1	Constant or address of word device where initial value is stored.
S2	Constant or address of word device where final value is stored.
S3	Constant or address of word device where the time required for the value change is stored.
D1	Address of word device to store the current value.
D1 +4~*D1* +7	2 double word devices for the internal timers. (Reserved)

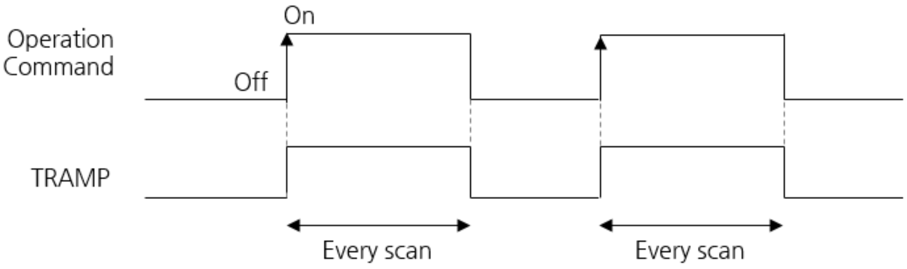
TRAMP

- TRAMP instruction gradually changes the initial value assigned to *S1* to the final value assigned to *S2* according to the time value assigned to *S3*. The current value in the process of value change is stored in *D1*. 2 double word devices starting from *D1* +4 are reserved for the internal timer.
- When the current value *D1* reaches the final value *S2*, the operations stops.
- The current value stored in *D1* is calculated as below:

$$\text{Current value} = \text{Initial value} + \frac{(\text{Final value} - \text{Initial value}) \times \text{Elapsed time}_{(\text{msec})}}{\text{Required time}}$$

- The range of value assigned to *S1, S2* and *S3* is as below:
  -32768 ≦ Value assigned to *S1, S2* and *S3* ≦ 32767
- When the value assigned to *S3* is “0”, the instruction does not operate. In this case, final value is stored in *D1*.
- If the execution condition turns OFF, the latest value is retained in *D1*. However, when the execution condition goes from OFF to ON, TRAMP instruction operates again from the beginning.
- Even if the execution condition is OFF, an internal process is in operation. Therefore, use indirect data register (@D) with caution in this instruction.

Execution Condition



Operation Error

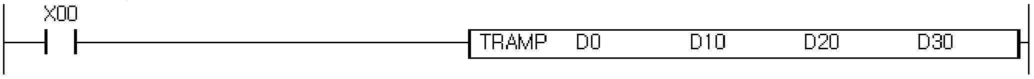
There is no error in this instruction.

Program Example

TRAMP

When X00 turns ON, TRAMP instruction is executed. The initial value is stored in D0. The final value is stored in D10. The required time is stored in D20. Current value will be stored in D30.

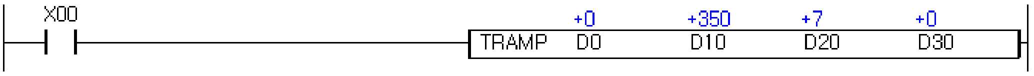
Ladder Diagram (LD)



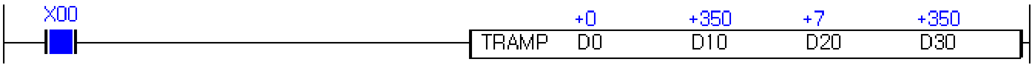
Instruction List (IL)

Instruction	Device			
LD	X00			
TRAMP	D0	D10	D20	D30

- Initial value “0” is stored in D0. Final value “350” is stored in D10.
- Time required for the TRAMP instruction, “7” seconds is stored in D20. It means that when TRAMP insturction is executed, initial value “0” will be gradually changed into final value “350” in 7 seconds.



- When X00 turns ON, TRAMP instruction is executed. Starting from the initial value “0” stored in D0, the value is changed gradually every second.
- After 7 seconds, the operation of TRAMP instruction is finished.



2.2 Comparison Instructions

2.2.1 LD x, LDD x, OR x, ORD x, AND x, ANDD x (x : <, <=, < >, =, >, >=)

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

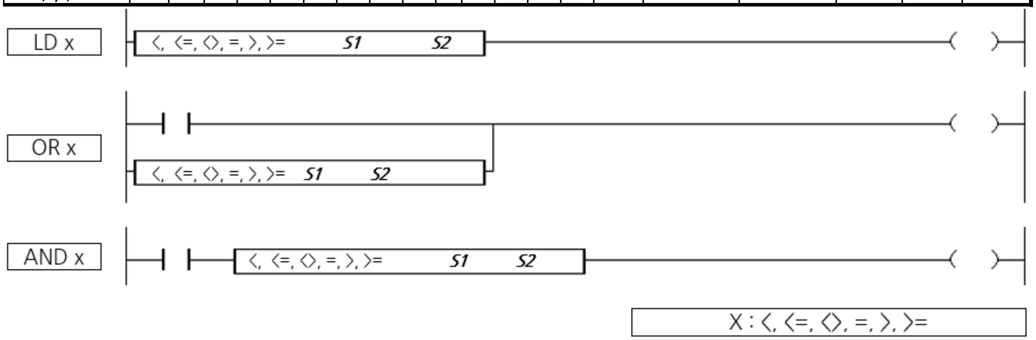
Function

LD x, OR x, AND x (x : <, <=, < >, =, >, >=) instructions compare word data *S1* to word data *S2*.

The comparison result is used as an operation result of 'A' contact.

LDD x, ORD x, ANDD x (x : <, <=, < >, =, >, >=) instructions compare double word data *S1* to double word data *S2*. The comparison result is used as an operation result of 'A' contact.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
<, <=, < >, =, >, >=	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				


LD x
<, <=, <>, =, >, >= S1 S2
OR x
<, <=, <>, =, >, >= S1 S2
AND x
<, <=, <>, =, >, >= S1 S2
X : <, <=, <>, =, >, >=

S1	Constant or address of device where word data to be compared is stored.
S2	Constant or address of device where a word data of comparison criterion is stored.

LD x, OR x, AND x (x : <, <=, < >, =, >, >=)

- LD x, OR x, AND x (x : <, <=, < >, =, >, >=) compares *S1* to *S2*.
- *S1* and *S2* are word data.
- The comparison is operated in INT. Therefore, comparison range of LD x, OR x, AND x (x : <, <=, < >, =, >, >=) is as following:
  -32768 ≦ *S1, S2* ≦ 32767
- When the condition x (x : <, <=, < >, =, >, >=) is true, the operation result is ON.
- If the condition x (x : <, <=, < >, =, >, >=) is false, the operation result is OFF.

(*) Refer to 1.1 Contact Instruction for the differences in structure of LD x, OR x and AND x instructions.

LDD x, ORD x, ANDD x (x : <, <=, < >, =, >, >=)

- LDD x, ORD x, ANDD x (x : <, <=, < >, =, >, >=) compares *S1* to *S2*.
- *S1* and *S2* are double word data.
- The comparison is operated in INT. Therefore, comparison range of LDD x, ORD x, ANDD x (x : <, <=, < >, =, >, >=) is as following:
  -2147483648 ≦ *S1, S2* ≦ 2147483647
- When the condition x (x : <, <=, < >, =, >, >=) is true, the operation result is ON.
- If the condition x (x : <, <=, < >, =, >, >=) is false, the operation result is OFF.

(*) Refer to 1.1 Contact Instruction for the differences in structure of LD x, OR x and AND x instructions.

Condition x	And if *S1* and *S2* are...	Operation result
< (Is less than)	*S1* < *S2*	ON
<= (Is less than or equal to)	*S1* ≦ *S2*	ON
< > (Is different to)	*S1* ≠ *S2*	ON
= (Is equal to)	*S1* = *S2*	ON
> (Is greater than)	*S1* > *S2*	ON
>= (Is greater than or equal to)	*S1* ≧ *S2*	ON

Execution Condition

LD(D) x, OR(D) x, AND(D) x (x : <, <=, < >, =, >, >=) 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 by @D exceeds the range of device D. (The range of device D depends on CPU type)

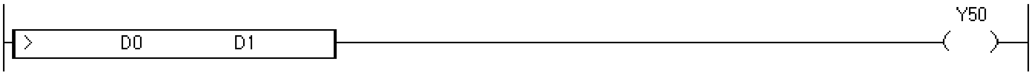
Program Example

LD x (x : <, <=, < >, =, >, >=)

The program operates as following:

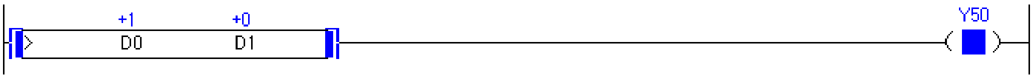
- When the value of D0 is greater than that of D1, Y50 turns ON.
- When the value of D0 is less than or equal to that of D1, Y50 does not turn ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD >	D0	D1
OUT	Y50	

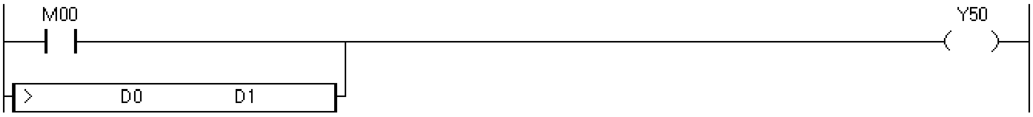


OR x (x : <, <=, < >, =, >, >=)

The program operates as following:

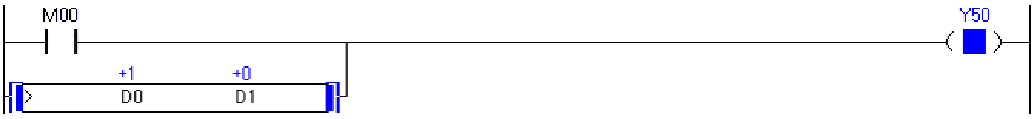
- When M00 is ON or the value of D0 is greater than that of D1, Y50 turns ON.
- When M00 is OFF and the value of D0 is less than or equal to that of D1, Y50 does not turn ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	M00	
OR >	D0	D1
OUT	Y50	

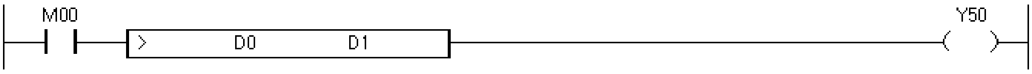


AND x (x : <, <=, < >, =, >, >=)

The program operates as following:

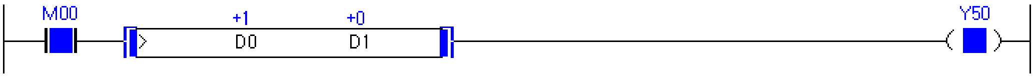
- When M00 is ON and the value of D0 is greater than that of D1, Y50 turns ON.
- When M00 is ON and the value of D0 is less than that of D1, Y50 does not turn ON.
- When M00 is OFF, Y50 does not turn ON regardless of the value of D0.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	M00	
AND >	D0	D1
OUT	Y50	



2.2.2 LDE x, ORE x, ANDE x (x : <, <=, < >, =, >, >=)

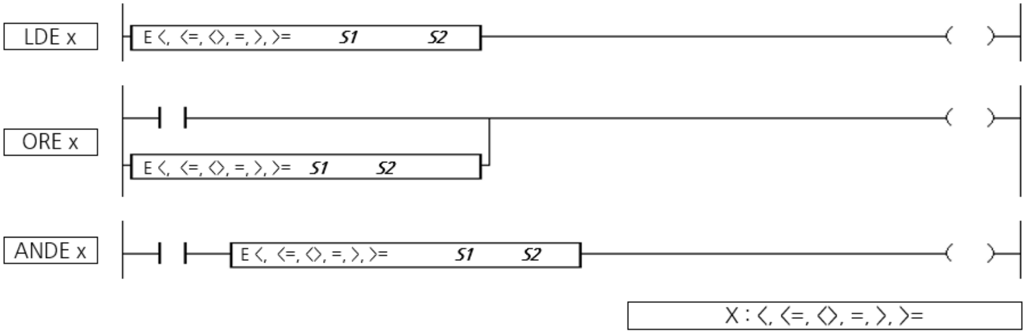
Supported PLC Series

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

Function

LDE x, ORE x, ANDE x (x : <, <=, < >, =, >, >=) instructions compare 32-bit floating-point number *S1* to 32-bit floating-point number *S2*.
The comparison result is used as an operation result of ‘A’ contact.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
E <, <=, < >, =, >, >=	*S1*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	S2	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				

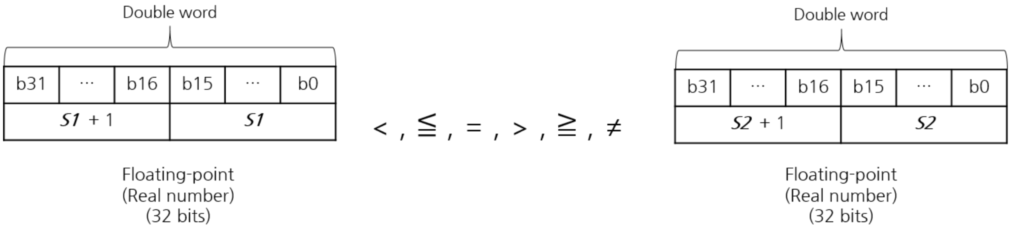


S1	32-bit floating-point number or head address of device where 32-bit floating-point number to be compared is stored.
S2	32-bit floating-point number or head address of device where 32-bit floating-point number of comparison criterion is stored.

LDE x, ORE x, ANDE x (x : <, <=, < >, =, >, >=)

- LDE x, ORE x, ANDE x (x : <, <=, < >, =, >, >=) compare *S1* to *S2*.
- *S1* and *S2* are 32-bit floating-point number.
- The comparison is operated in 32-bit floating-point number. Therefore, comparison range of LDE x, ORE x, ANDE x (x : <, <=, < >, =, >, >=) is as following:
  $2^{-127} \leq | S1, S2 | \leq 2^{128}$
- When the condition x (x : <, <=, < >, =, >, >=) is true, the operation result is ON.
- If the condition x (x : <, <=, < >, =, >, >=) is false, the operation result is OFF.

(*) Refer to 1.1 Contact Instruction for the differences in structure of LD x, OR x and AND x instructions.



Condition x	And if *S1* and *S2* are...	Operation result
< (Is less than)	*S1* < *S2*	ON
<= (Is less than or equal to)	*S1* ≦ *S2*	ON
<> (Is different to)	*S1* ≠ *S2*	ON
= (Is equal to)	*S1* = *S2*	ON
> (Is greater than)	*S1* > *S2*	ON
>= (Is greater than or equal to)	*S1* ≧ *S2*	ON

Execution Condition

LDE x, ORE x, ANDE x (x : <, <=, <>, =, >, >=) 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 by @D exceeds the range of device D. (The range of device D depends on CPU type)

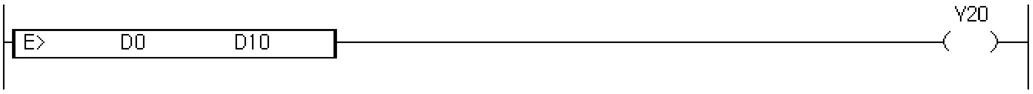
Program Example

LDE x (x : <, <=, <>, =, >, >=)

The program operates as following:

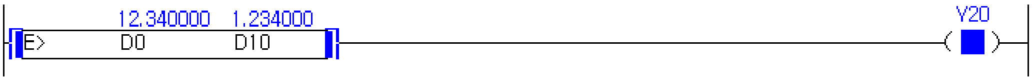
- When the 32-bit floating-point number stored in device D0 (D0, D1) is greater than that of 32-bit floating-point number stored in device D10 (D10, D11), the operation result is ON. Then, Y20 turns ON.
- When the 32-bit floating-point number stored in device D0 (D0, D1) is less than or equal to that of 32-bit floating-point number D10 (D10, D11), the operation result is OFF. Then, Y20 does not turn ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LDE >	D0	D10
OUT	Y20	

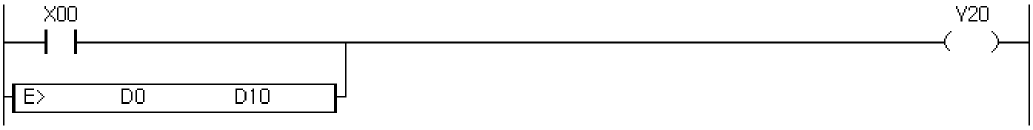


ORE x (x : <, <=, < >, =, >, >=)

The program operates as following:

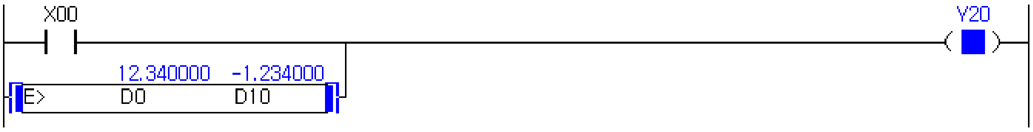
- When X00 is ON or the 32-bit floating-point number stored in device D0 (D0, D1) is greater than that of 32-bit floating-point number stored in device D10 (D10, D11), the operation result is ON. Then, Y20 turns ON.
- When X00 is OFF and the 32-bit floating-point number stored in device D0 (D0, D1) is less than that of 32-bit floating-point number stored in device D10 (D10, D11), the operation result is OFF. Then, Y20 does not turn ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
ORE >	D0	D10
OUT	Y20	

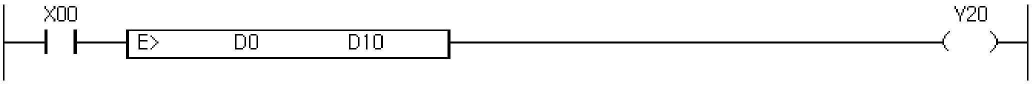


ANDE x (x : <, <=, < >, =, >, >=)

The program operates as following:

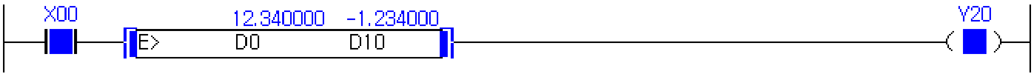
- When X00 is ON and the 32-bit floating-point number stored in device D0 (D0, D1) is greater than that of 32-bit floating-point number stored in device D10 (D10, D11), the operation result is ON. Then, Y20 turns ON.
- When X00 is ON and the value of D0 (D0, D1) is less than that of 32-bit floating-point number stored in device D10 (D10, D11), the operation result is OFF. Then, Y20 does not turn ON.
- When X00 is OFF, Y20 does not turn ON regardless of the comparison result.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
ANDE >	D0	D10
OUT	Y20	



2.2.3 UCMP, UDCMP

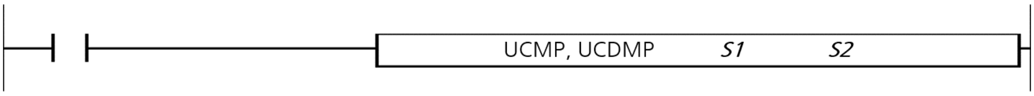
Supported PLC Series

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

UCMP instruction compares word data *S1* to word data *S2*. UCDMP instruction compares double word data *S1* to double word data *S2*. The result is stored in the device F120~F125.

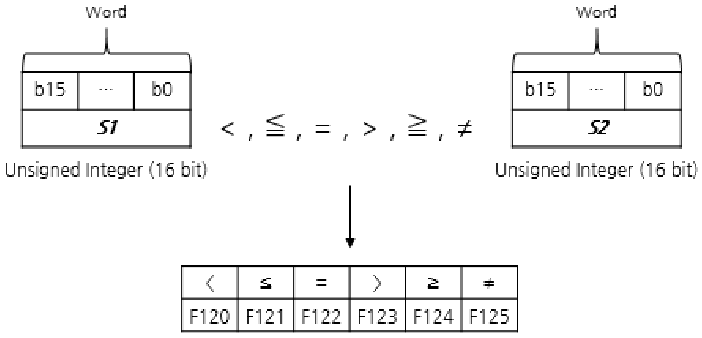
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
UCMP, UDCMP	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S1	Address of word/double device to be compared.
S2	Address of word/double device which is a comparison criterion.

UCMP

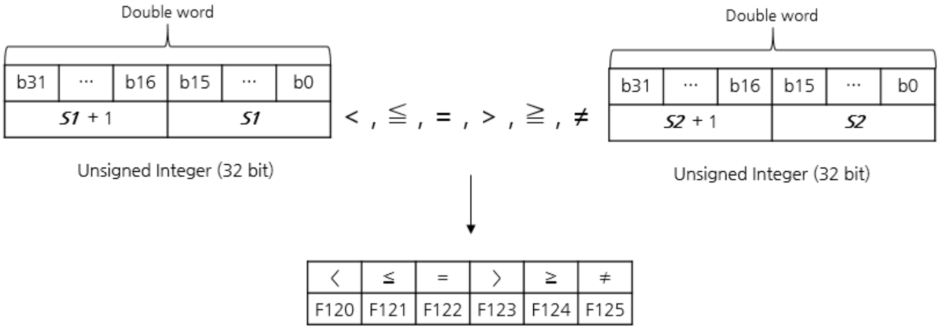
- UCMP compares *S1* to *S2*.
- *S1* and *S2* are word data.
- The comparison is operated in UINT. Therefore, comparison range of UCMP is as following:
  0 ≦ *S1, S2* ≦ 65535
- Comparison result is stored in F120 - F125 as following.



Condition	Address of Flag where result is stored
S1 〈 *S2*	F120
S1 ≦ *S2*	F121
S1 = *S2*	F122
S1 〉 *S2*	F123
S1 ≧ *S2*	F124
S1 ≠ *S2*	F125

UDCMP

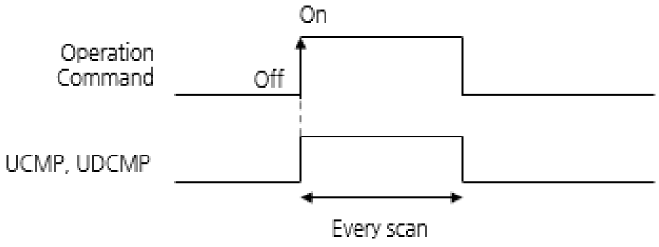
- UDCMP compares *S1* to *S2*.
- *S1* and *S2* are double word data.
- The comparison is operated in UINT. Therefore, comparison range of UDCMP is as following:
  0 ≦ *S1, S2* ≦ 4294967295
- Comparison result is stored in F120 - F125 as following.



〈	≤	=	〉	≥	≠
F120	F121	F122	F123	F124	F125

Condition	Address of Flag where result is stored
S1 〈 *S2*	F120
S1 ≦ *S2*	F121
S1 = *S2*	F122
S1 〉 *S2*	F123
S1 ≧ *S2*	F124
S1 ≠ *S2*	F125

Execution Condition



Operation Error

Error Flag (F110)

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

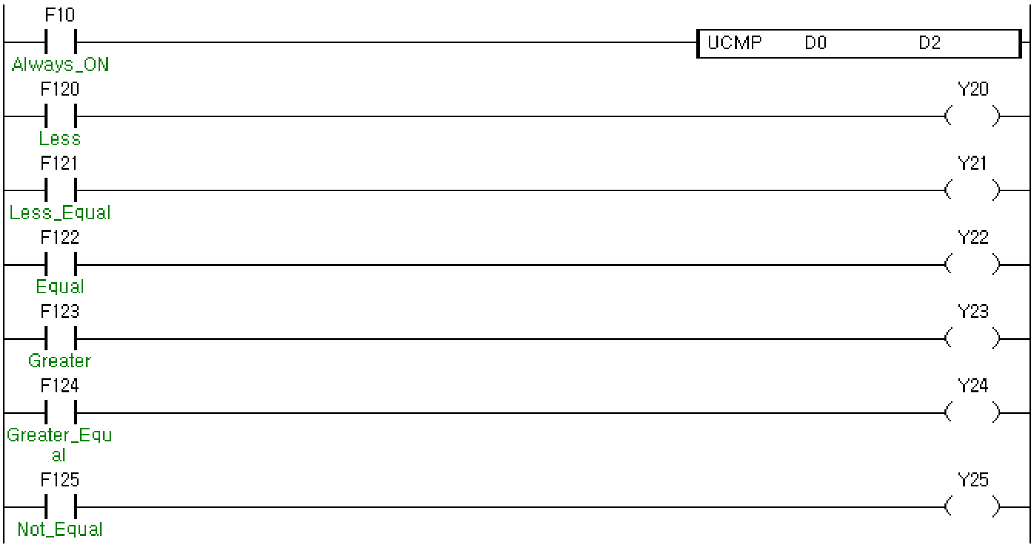
Program Example

UCMP, UDCMP

According to the comparison result, different flags are set as following:

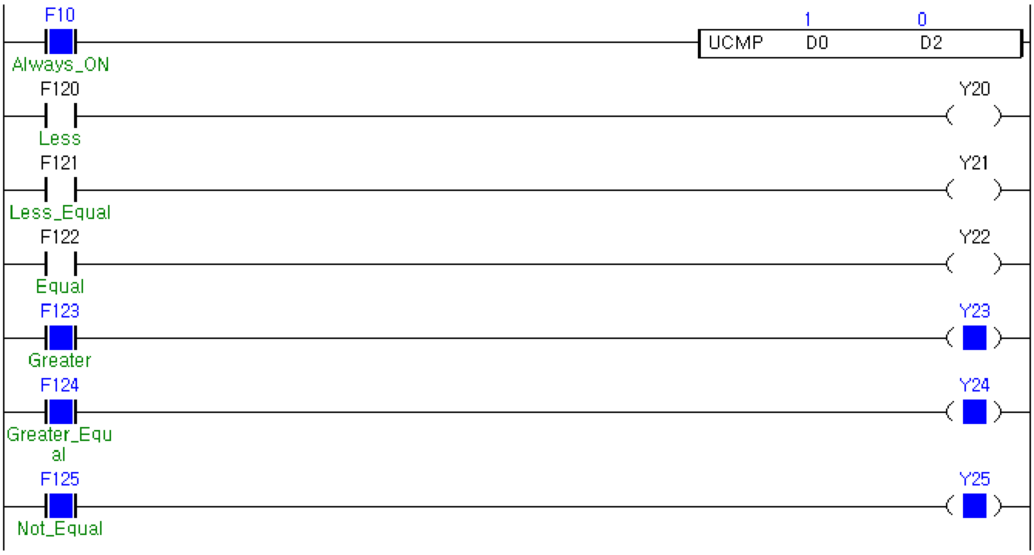
- When the value of D0 is greater than that of D2, F123, F124, F125 are set and Y23, Y24, Y25 turns ON.
- When the value of D0 is less than that of D2, F120, F121, F125 are set and Y20, Y21, Y25 turns ON.
- When the value of D0 is equal to that of D2, F121, F122, F124 are set and Y21, Y22, Y24 turns ON.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	F10	
UCMP	D0	D2
LD	F120	
OUT	Y20	
LD	F121	
OUT	Y21	
LD	F122	
OUT	Y22	
LD	F123	
OUT	Y23	
LD	F124	
OUT	Y24	
LD	F125	
OUT	Y25	


F10
Always_ON
UCMP
1
D0
0
D2
F120
Less
Y20
F121
Less_Equal
Y21
F122
Equal
Y22
F123
Greater
Y23
F124
Greater_Equ
al
Y24
F125
Not_Equal
Y25

2.2.4 BK x, BK x P (x : <, <=, < >, =, >, >=)

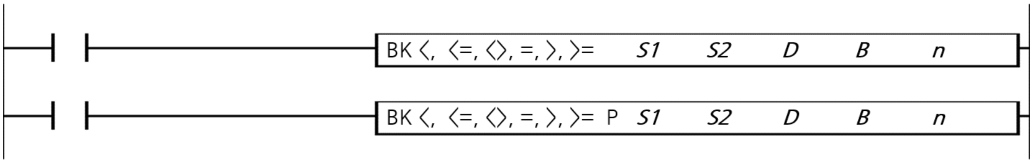
Supported PLC Series

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

BK x, BK x P (x : <, <=, <>, =, >, >=) compare *n* word data blocks starting from *S1* to *n* word data blocks starting from *S2*. The comparison result is stored in the device *D* starting from bit address assigned to *B*.

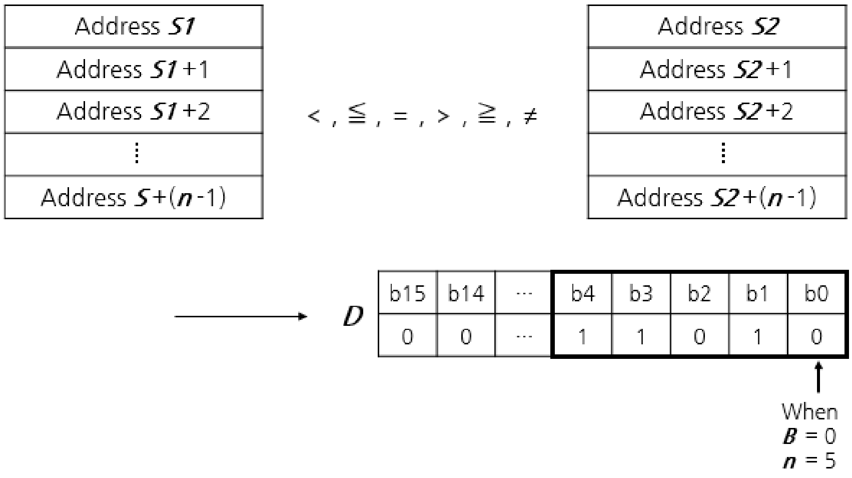
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BK x (P) (x : <, <=, < >, =, >, >=)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	6	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	-				
	D	O	-	O	O	O	-	-	-	-	O	O	O	O	O	-				
	B	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				


BK <, <=, <>, =, >, >= *S1* *S2* *D* *B* *n*
BK <, <=, <>, =, >, >= P *S1* *S2* *D* *B* *n*

S1	Constant or head address of word devices to be compared.
S2	Head address of word devices which are a comparison criterion.
D	Address of word device where the block comparison results are stored.
B	Bit address of word device *D* where the block comparison result is stored.
n	The number of blocks to compare.

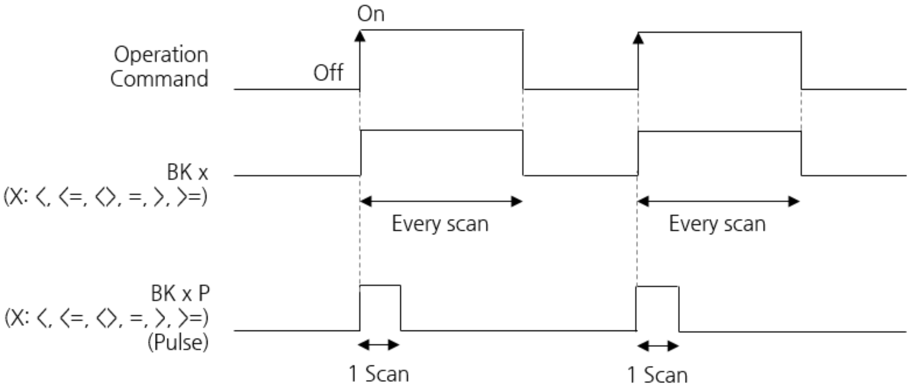
BK x, BK x P (x : <, <=, < >, =, >, >=)

- BK x, BK x P (x : <, <=, < >, =, >, >=) compare *n* word data blocks starting from *S1* to *n* word data blocks starting from *S2*.
- *S1* and *S2* are word data block.
- The number of word data blocks to be compared are assigned to *n*.
- The comparison result is stored in starting from the bit *B* of word device *D*.
- When the number of results exceeds the range of *D*, the results are stored starting from the bit 0 of *D*+1.
  (*) Refer to the Program Example below.
- The comparison is operated in INT. Therefore, range of *S1* and *S2* is as following:
  -32768 ≦ *S1, S2* ≦ 32767
- When the condition x (x : <, <=, < >, =, >, >=) is true, the bit *B* of *D* turns ON.
- When the condition x (x : <, <=, < >, =, >, >=) is false, the bit *B* of *D* turns OFF.
- Each comparison result is stored starting from bit *B* of *D* as following.



Condition x	And if *S1* and *S2* are...	Operation result
〈 (Is less than)	*S1* 〈 *S2*	ON
〈= (Is less than or equal to)	*S1* ≦ *S2*	ON
〈〉 (Is different to)	*S1* ≠ *S2*	ON
= (Is equal to)	*S1* = *S2*	ON
〉 (Is greater than)	*S1* 〉 *S2*	ON
〉= (Is greater than or equal to)	*S1* ≧ *S2*	ON

Execution Condition



Operation Error

Error Flag (F110)

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

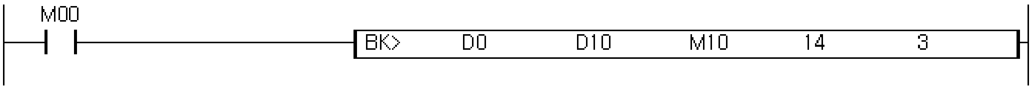
Program Example

BK x, BK x P (x = <, <=, < >, =, >, >=)

This program compares 3 word data blocks starting from D0 to 3 word data blocks starting from D10.
The comparison result is stored in starting from the bit 14 of word device M10. Bit 14 of M10 is M1E.

When M00 is ON, BK> instruction operates.
The instruction compares D0 to D10. Then, the comparison result is stored in M1E.
The instruction compares D1 to D11. Then, the comparison result is stored in M1F.
The instruction compares D2 to D12. Then, the comparison result is stored in M20.

Ladder Diagram (LD)

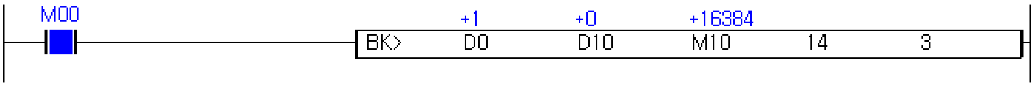


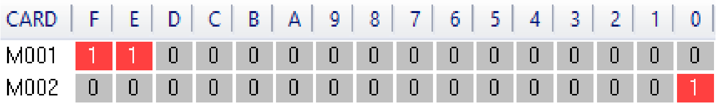
Instruction List (IL)

Instruction	Device				
LD	M00				
BK>	D0	D10	M10	14	3

The program operates as following:

- When M00 turns ON, BK> instruction operates.
- The instruction compares 3 words starting from D0 and 3 words starting from D10.
- The comparison is made between D0 and D10, D1 and D11, D2 and D12. Each comparison result is stored from bit 14 of M10. Bit 14 of M10 is M1E.
- When the values of D0, D1 and D2 are “1” and the values of D10, D11 and D12 are “0”, M1E, M1F and M20 are set.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M001	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

2.2.5 BKCMP x, BKCMP x P (x : <, <=, < >, =, >, >=)

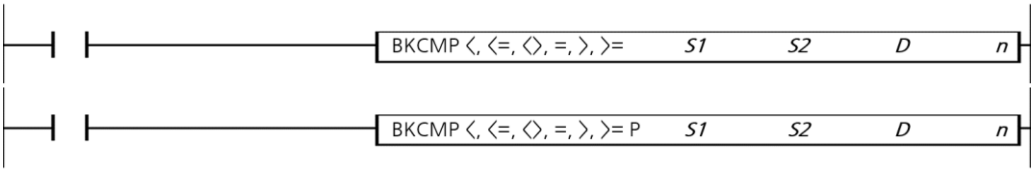
Supported PLC Series

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

BKCMP x, BKCMP x P (x = <, <=, <>, =, >, >=) compare *n* word data starting from word device *S1* to *n* word data starting from word device *S2*. The comparison result is stored in the *n* word devices starting from *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
BKCMP x (P) (x : <, <=, < >, =, >, >=)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	5	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S1	Constant or head address of word devices to be compared.
S2	Head address of word devices which are comparison criteria.
D	Head address of word devices where the block comparison result is stored.
n	The number of blocks to compare.

BKCMP x, BKCMP x P (x = <, <=, < >, =, >, >=)

- BKCMP x, BKCMP x P (x = <, <=, < >, =, >, >=) compare *n* word data blocks starting from *S1* to *n* word data blocks starting from *S2*.
- *S1* and *S2* are word data blocks.
- The number of word data blocks to be compared are assigned to *n*.
- The comparison result is stored in word devices starting from *D*.
- The comparison is operated in INT. Therefore, range of *S1* and *S2* is as following: -32768 ≦ *S1, S2* ≦ 32767
- When the condition x (x = <, <=, < >, =, >, >=) is true, the device *D* is set as “1”.
- When the condition x (x = <, <=, < >, =, >, >=) is false, the device *D* is set as “0”.

If instruction
BKCMP< is executed

S1	12
S1+1	621
S1+2	271
⋮	⋮
S1+(*n*-1)	-321

<

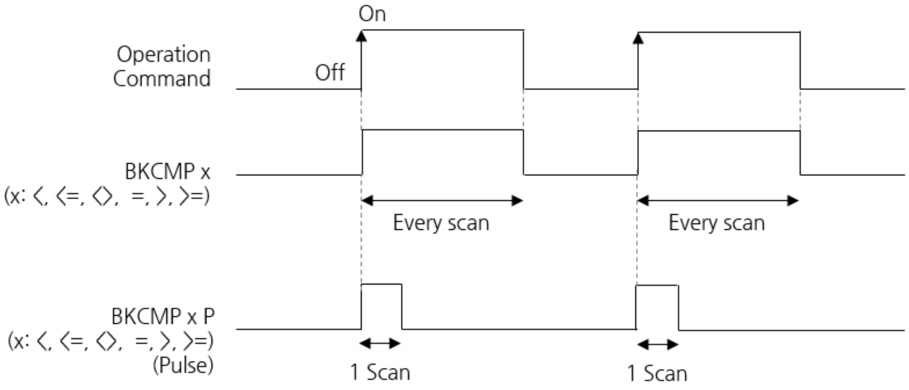
S2	-331
S2+1	5221
S2+2	1064
⋮	⋮
S2+(*n*-1)	-7844

Result →

D	0 (OFF)
D+1	1 (ON)
D+2	1 (ON)
⋮	⋮
D+(*n*-1)	0 (OFF)

Condition x	And if *S1* and *S2* are...	Operation result
〈 (Is less than)	*S1* 〈 *S2*	ON
〈= (Is less than or equal to)	*S1* ≦ *S2*	ON
〈〉 (Is different to)	*S1* ≠ *S2*	ON
= (Is equal to)	*S1* = *S2*	ON
〉 (Is greater than)	*S1* 〉 *S2*	ON
〉= (Is greater than or equal to)	*S1* ≧ *S2*	ON

Execution Condition



Operation Error

Error Flag (F110)

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

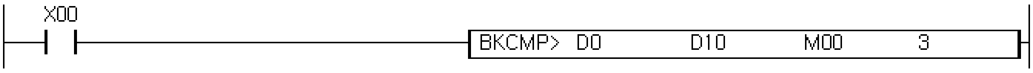
Program Example

BKCMP x, BKCMP x P (x : <, <=, < >, =, >, >=)

This program compares 3 word data blocks starting from D0 (D0~D2) to 3 word data blocks starting from D10 (D10~D12).
The comparison result is stored in 3 word devices starting from M00 (M00~M2F).

When X00 is ON, BKCMP> instruction is executed.
The instruction compares D0 to D10. Then, the comparison result is stored in M00.
The instruction compares D1 to D11. Then, the comparison result is stored in M10.
The instruction compares D2 to D12. Then, the comparison result is stored in M20

Ladder Diagram (LD)

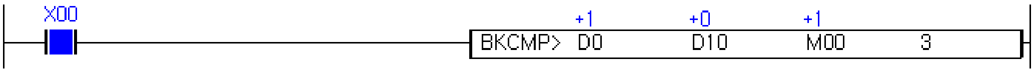


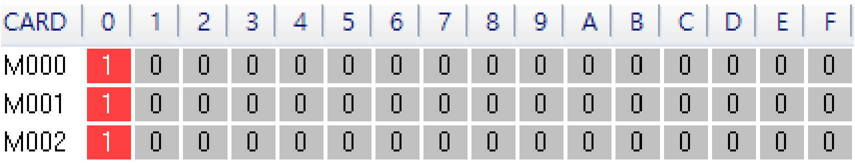
Instruction List (IL)

Instruction	Device			
LD	X00			
BKCMP>	D0	D10	M00	3

The program operates as following:

- The instruction compares 3 words starting from D0 (D0~D2) and 3 word starting from D10 (D10~D12).
- The comparison is made between D0 and D10, D1 and D11, D2 and D12. Each comparison result is stored in M00, M10 and M20.
- The values stored in D0, D1 and D2 are "1" and the values stored in D10, D11 and D12 are "0",
- Since D0, D1 and D2 are greater than D10, D11 and D12, the result "1" is stored in M00, M10 and M20.





CARD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
M000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M001	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M002	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.3 Data Conversion Instructions

2.3.1 NEG, NEGP, DNEG, DNEGP

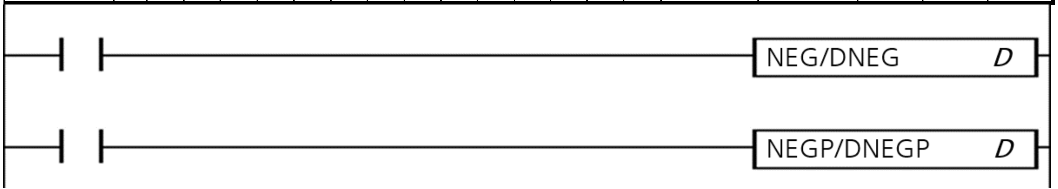
Supported PLC Series

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

NEG(P) converts the sign of the INT data in word device *D* and stores the result in *D*.
DNEG(P) converts the sign of the INT data in double word device *D* and *D+1* and stores the result at *D* and *D+1*.

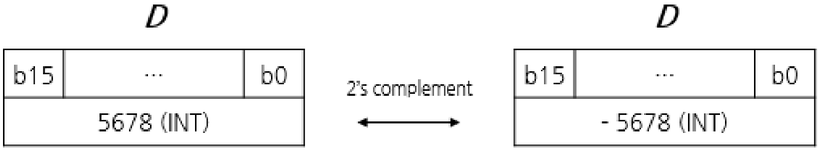
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
NEG(P) DNEG(P)	*D*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-	2	O	-	-



D	Head address of word/double word device where conversion of two's complement will be performed.

NEG, NEGP

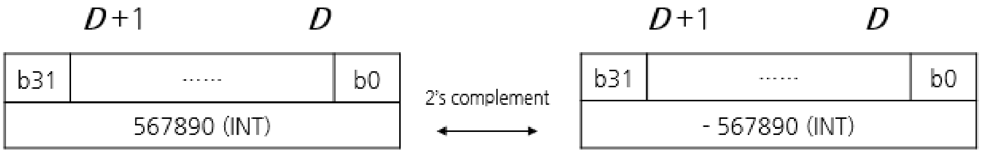
- NEG(P) instruction converts the INT data assigned to word device *D* to its two's complement. The converted data is stored in word device *D*.
- NEG(P) converts the sign of the INT data in word device assigned to *D*.



- NEG(P) instruction is used for converting the positive sign to the negative sign and vice versa.

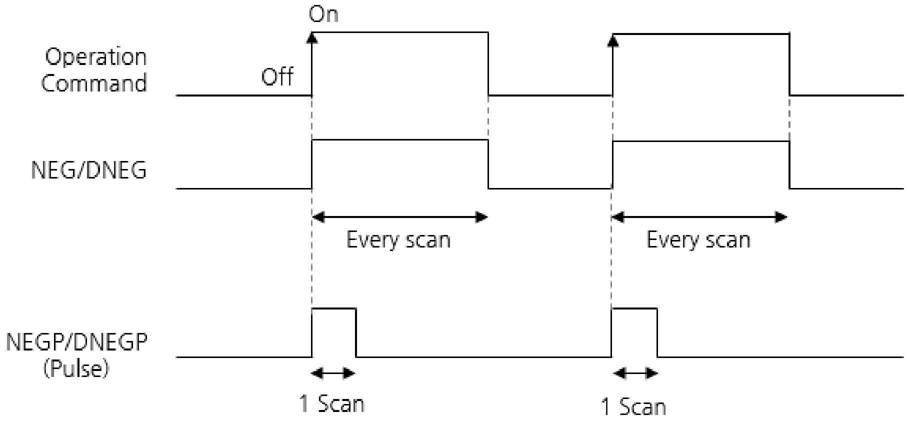
DNEG, DNEGP

- DNEG(P) instruction converts the INT data assigned to double word device *D* and *D*+1 to its two's complement. The converted data is stored in double word device *D* and *D*+1.
- Converts the sign of the INT data in double word device assigned to *D* and *D*+1.



- DNEG(P) instruction is used for converting the positive sign to the negative sign and vice versa.

Execution Condition



⚠ Using pulse contact as execution condition of NEG/DNEG instruction or using NEGP/DNEGP instruction is recommended.

Operation Error

Error flag (F110)

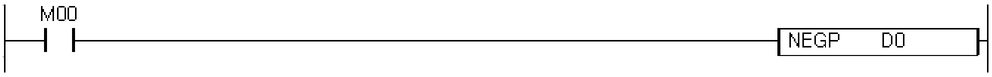
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

NEGP

The program converts the sign of value stored in D0 to positive or negative and store the result at D0.

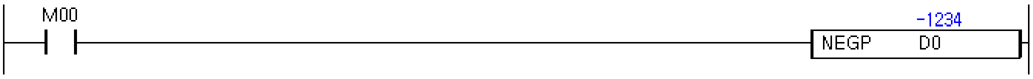
Ladder Diagram (LD)



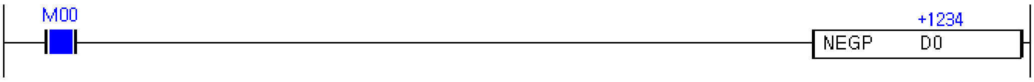
Instruction List (IL)

Instruction	Device
LD	M00
NEGP	D0

- The value assigned to D0 is “-1234”.



- When M00 turns ON, NEGP instruction is executed. Then, the value of D0 is converted to positive once.



2.3.2 BCD, BCDP, DBCD, DBCDP

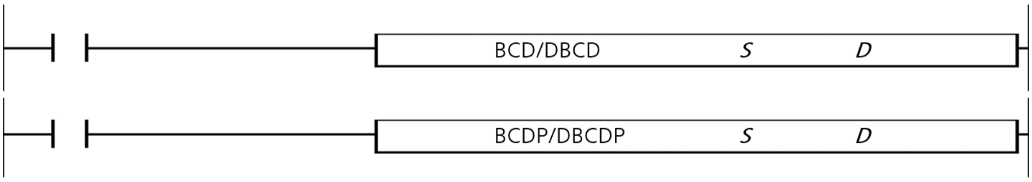
Supported PLC Series

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

BCD(P) instructions convert 16-bit BIN data stored in device *S* into 4-digit BCD data. Then, the conversion result is stored in device *D*.
DBCD(P) instructions convert 32-bit BIN data stored in device *S* into 8-digit BCD data. Then, the conversion result is stored in device *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
BCD(P) DBCD(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Binary or head address of word/double word device where BIN data to be converted is stored.
D	Head address of word/double word device where conversion result (BCD) will be stored.

BCD, BCDP

- BCD(P) instructions convert 16-bit BIN data stored in the device *S* into 4-digit BCD data.
- Converted data is stored in the device *D*.
- When the conversion result (4-digit BCD data) exceeds its range, the error flag (F110) turns ON.
- The range of the conversion result is as following.
  0 ≦ conversion result ≦ 9999

(*) Refer to Appendix n for BCD code table.

When *S*= 9999 (HEX = H270F)

S	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1

⇩

D	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	thousands digit				hundreds digit				tens digit				ones digit			

DBCD, DBCDP

- DBCD(P) instructions convert 32-bit BIN data stored in the device *S* into 8-digit BCD data.
- Converted data is stored in the device *D*.
- When the conversion result (8-digit BCD data) exceeds its range, the error flag (F110) turns ON.
- The range of the conversion result is as following.
  0 ≦ conversion result ≦ 99999999

(*) Refer to Appendix n for BCD code table.

When *S* = 99999999 (HEX = H05F5E0FF)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
S+1	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1

⇩

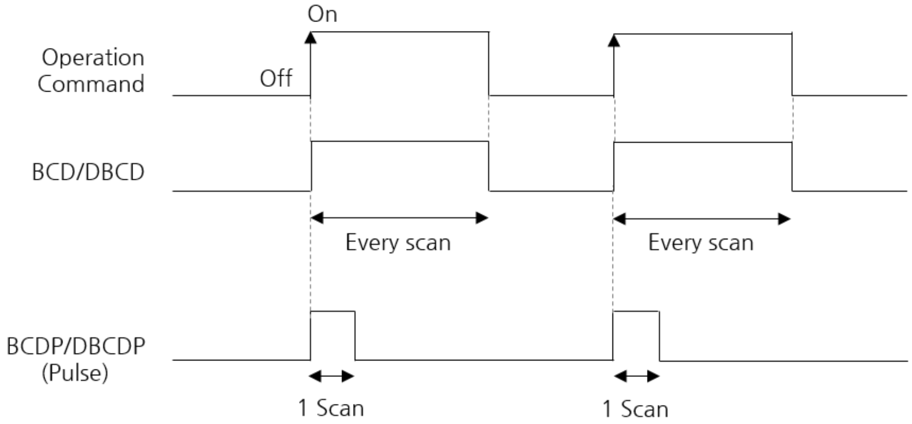
ten millions digit | millions digit | hundred thousands digit | ten thousands digit

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
D+1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

thousands digit | hundreds digit | tens digit | ones digit

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)

F110 turns ON for 1 scan when the conversion result exceeds as following:

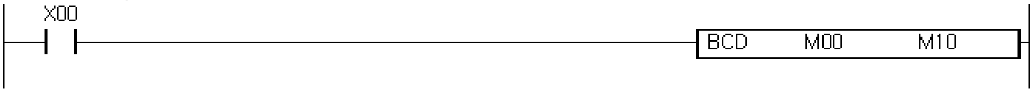
BCD(P): 0 ≦ conversion result ≦ 9999
DBCD(P): 0 ≦ conversion result ≦ 99999999

Program Example

BCD

When operation command X00 is ON, BCD instruction is executed. Then, 16-bit BIN data stored in word device M00 (M00 ~ M0F) is converted into 4-digit BCD data. The conversion result is stored in word device M10 (M10 ~ M1F).

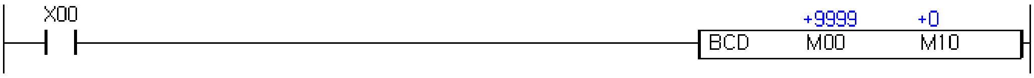
Ladder Diagram (LD)



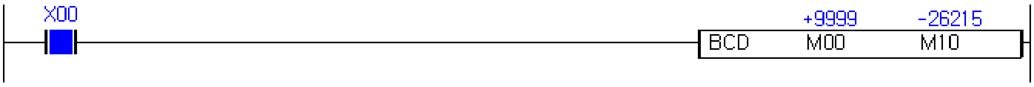
Instruction List (IL)

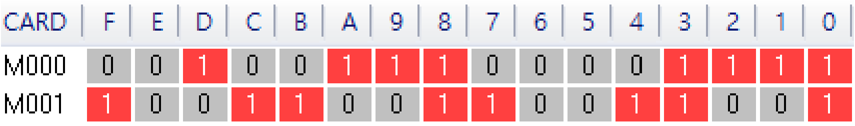
Instruction	Device	
LD	X00	
BCD	M00	M10

- 16-bit BIN data “9999” (= H270F) is assigned to M00.



- When X00 turns ON, BCD instruction is executed. Then, 16-bit BIN data “9999” (= H270F) is converted into 4-digit BCD data “9999” (= H9999). The conversion result is stored in M10 (M10 ~ M1F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1
M001	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

2.3.3 BIN, BINP, DBIN, DBINP

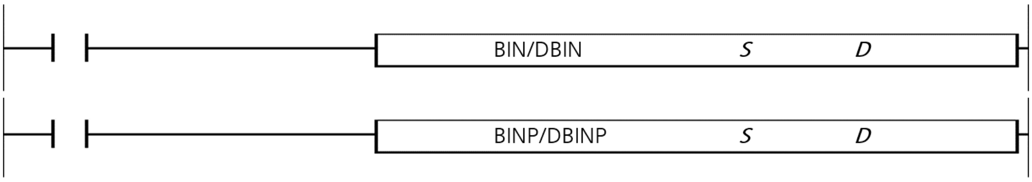
Supported PLC Series

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

BIN(P) instructions convert 4-digit BCD data stored in device *S* into 16-bit UINT data. Then, the conversion result is stored in device *D*.
DBIN(P) instructions convert 8-digit BCD data stored in device *S* into 16-bit UINT data. Then, the conversion result is stored in device *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
BIN(P) DBIN(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				

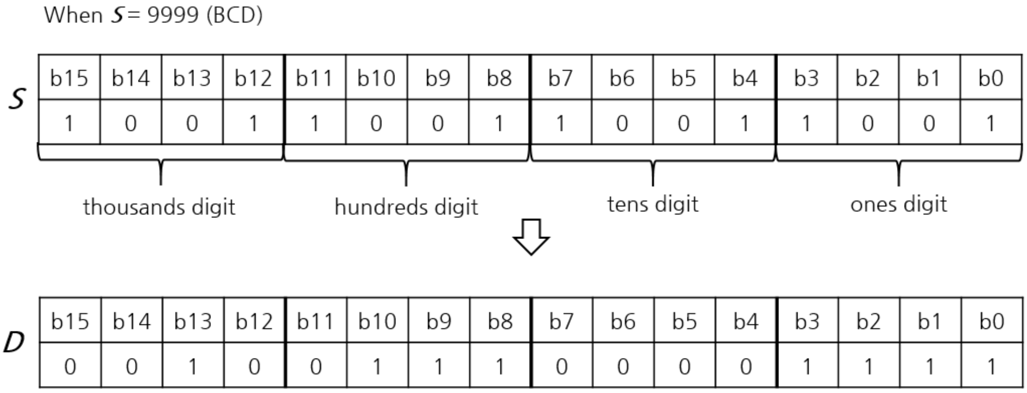


S	BCD or head address of word/double word device where BCD data to be converted is stored.
D	Head address of word/double word device where conversion result (16/32-bit UINT data) will be stored.

BIN, BINP

- BIN(P) instructions convert 4-digit BCD data stored in the device *S* into 16-bit UINT data.
- Converted data is stored in the device *D*.
- When the data to be converted (4-digit BCD data) exceeds its range, The error flag (F110) turns ON.
- The range of 4-digit BCD data and the conversion result is as following.
  0 ≦ 4-digit BCD data ≦ 9999 (H9999)

(*) Refer to Appendix n for BCD code table.


When *S* = 9999 (BCD)

S	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

D	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1

DBIN, DBINP

- DBIN(P) instructions convert 8-digit BCD data stored in the device *S* into 32-bit UINT data.
- Converted data is stored in the device *D*.
- When the data to be converted (8-digit BCD data) exceeds its range, The error flag (F110) turns ON.
- The range of 8-digit BCD data and the conversion result is as following.
  0 ≦ 8-digit BCD data ≦ 99999999 (H99999999)

(*) Refer to Appendix n for BCD code table.

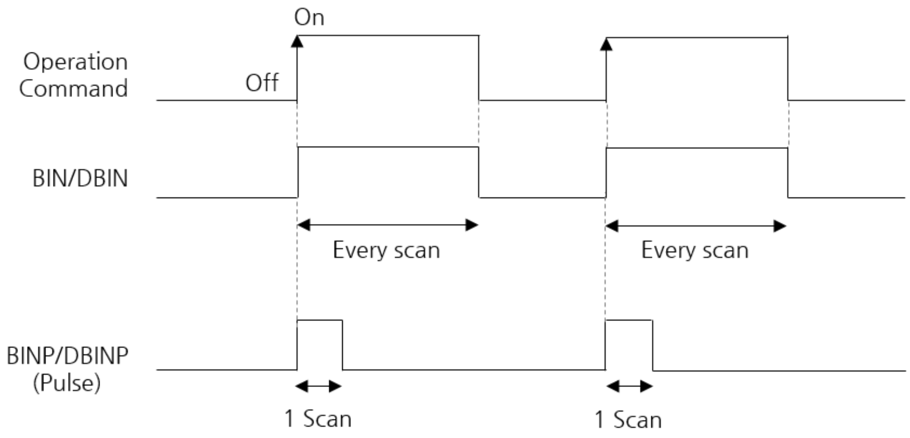
When *S* = 99999999 (BCD)

	ten millions digit				millions digit				hundred thousands digit				ten thousands digit			
S+1	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
S	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	thousands digit				hundreds digit				tens digit				ones digit			

⇩

D+1	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1
D	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1

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)

F110 turns ON for 1 scan when the data to be converted exceeds as following:

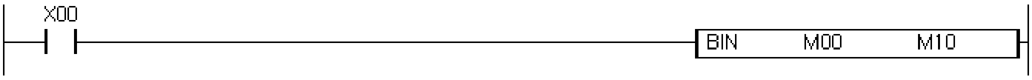
BIN(P): 0 ≦ data to be converted ≦ 9999
DBIN(P): 0 ≦ data to be converted ≦ 99999999

Program Example

BIN

When operation command X00 is ON, BIN instruction is executed. Then, 4-digit BCD data stored in word device M00 (M00 ~ M0F) is converted into 16-bit UINT data. The conversion result is stored in word device M10 (M10 ~ M1F).

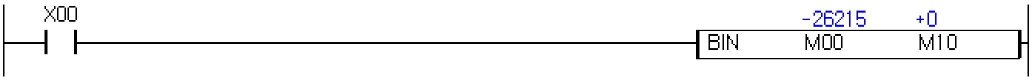
Ladder Diagram (LD)



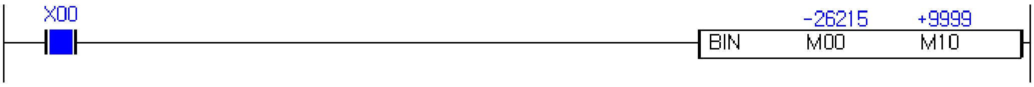
Instruction List (IL)

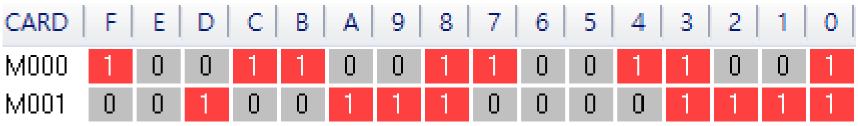
Instruction	Device	
LD	X00	
BIN	M00	M10

- 4-digit BCD data “9999” (= H9999) is assigned to M00.



- When X00 turns ON, BIN instruction is executed. Then, 4-digit BCD data “9999” (= H9999) is converted into 16-bit UINT data “9999” (= H270F). The conversion result is stored in M10 (M10 ~ M1F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
M001	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1

2.3.4 FLT, FLTP, DFLT, DFLTP

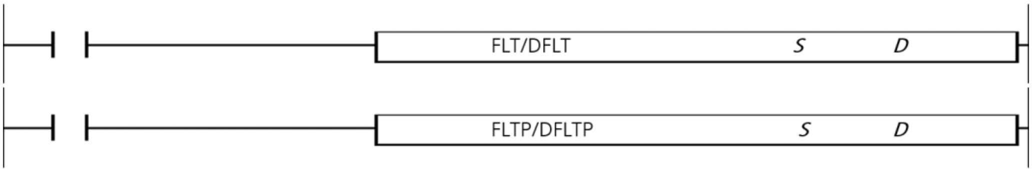
Supported PLC Series

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

Function

FLT(P) instructions convert 16-bit INT data stored in device *S* into 32-bit floating-point number. Then, the conversion result is stored in double word device *D*.
DFLT(P) instructions convert 32-bit INT data stored in device *S* into 32-bit floating-point number. Then, the conversion result is stored in double word device *D*.

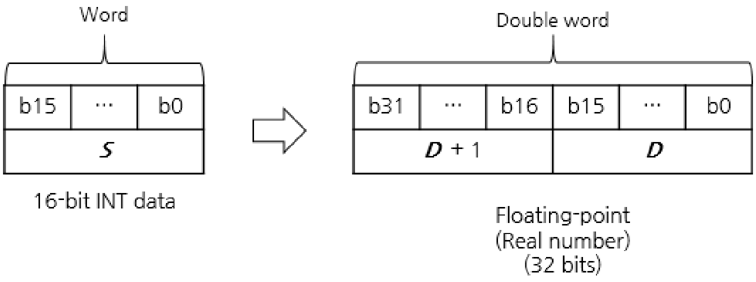
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FLT(P) DFLT(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Constant or head address of word/double word device where data to be converted is stored.
D	Head address of double word device where conversion result (32-bit floating-point number) will be stored.

FLT, FLTP

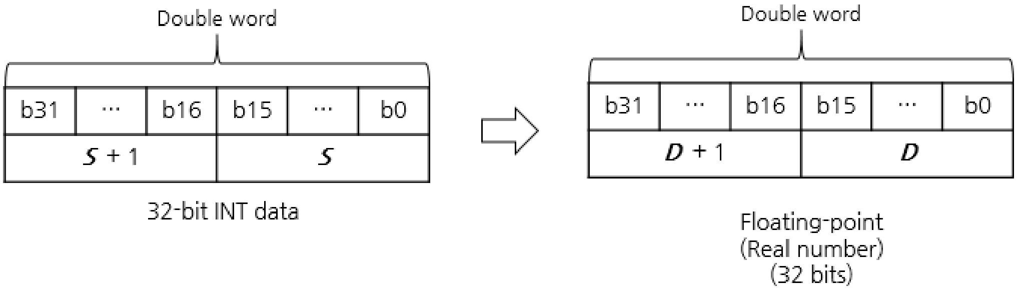
- FLT(P) instructions convert 16-bit INT data stored in the word device *S* into 32-bit floating-point number.
- Converted data is stored in double word device *D*.
- The range of data stored in word device *S* is as following.
  -32768 ≦ assigned value in word device *S* ≦ 32767



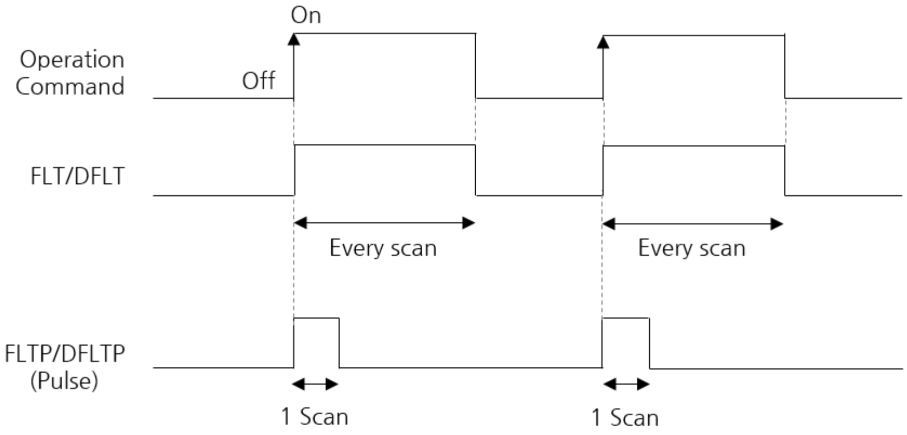
DFLT, DFLTP

- DFLT(P) instructions convert 32-bit BIN data stored in the double word device *S* into 32-bit floating-point number.
- Converted data is stored in double word device *D*.
- The range of data stored in double word device *S* is as following.
  -2147483648 ≦ assigned value in word device *S* ≦ 2147483647
- Significant digits of converted data are 23 bits in binary display. In decimal display, the number of significant digits are approximately 7~8.

- If the integer exceeds 8 digits, upper 23 bits of device *S* are preserved while the lower 9 bits are converted into exponent part of floating-point number. It means that an error may occur on the rightmost digits of conversion result.
  (*) Refer to the program example below.



Execution Condition



Operation Error

Error flag (F110)

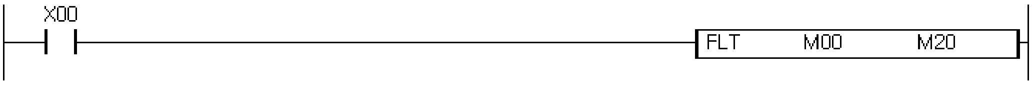
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

FLT

When operation command X00 is ON, FLT instruction is executed. Then, 16-bit INT data stored in word device M00 (M00 ~ M0F) is converted into 32-bit floating-point number. The conversion result is stored in word devices starting from M20 (M20 ~ M3F).

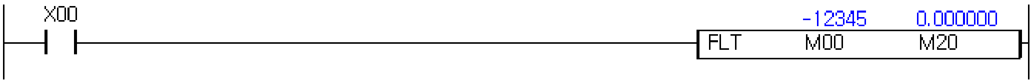
Ladder Diagram (LD)



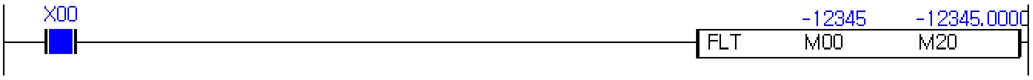
Instruction List (IL)

Instruction	Device	
LD	X00	
FLT	M00	M20

- 16-bit INT data "-12345" is assigned to M00.



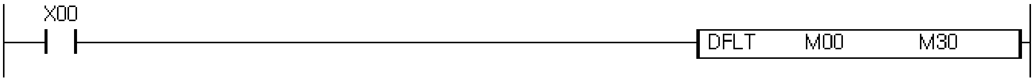
- When X00 turns ON, FLT instruction is executed. Then, 16-bit INT data "-12345" is converted into 32-bit floating-point number "-12345.000000". The conversion result is stored in M20 (M20 ~ M3F).



DFLT

When operation command X00 is ON, DFLT instruction is executed. Then, 32-bit INT data stored in double word device M00 (M00 ~ M1F) is converted into 32-bit floating-point. The conversion result is stored starting from word device M30 (M30 ~ M4F).

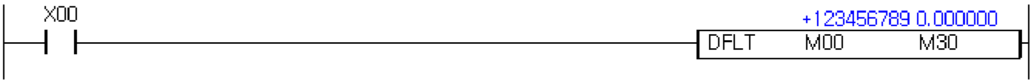
Ladder Diagram (LD)



Instruction List (IL)

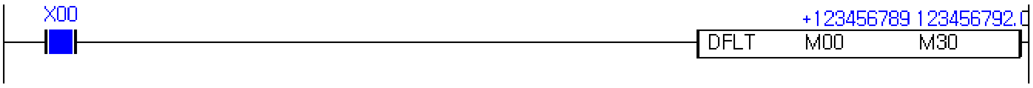
Instruction	Device	
LD	X00	
DFLT	M00	M30

- 32-bit INT data "123456789" is assigned to M00.



- When X00 turns ON, DFLT instruction is executed. Then, 32-bit INT data "123456789" is converted into 32-bit floating-point number "123456792.000000". The conversion result is stored in M30 (M30 ~ M4F).

- The number of significant digits are 7 in 32-bit floating-point number. Therefore, in the conversion result, 7 digits from the leftmost side are significant while following digits have an error.



2.3.5 INT, INTP, DINT, DINTP

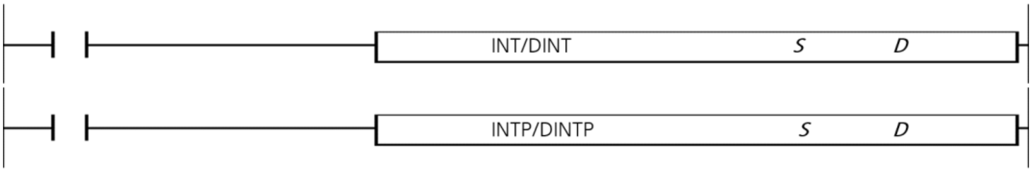
Supported PLC Series

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

Function

INT(P) instructions convert 32-bit floating-point number stored in double word device *S* into 16-bit INT data. Then, the conversion result is stored in word device *D*.
DINT(P) instructions convert 32-bit floating-point number stored in double word device *S* into 32-bit INT data. Then, the conversion result is stored in double word device *D*.

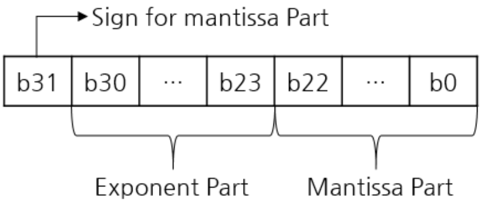
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
INT(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
DINT(P)	*D*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Floating-point number or head address of double word device where 32-bit floating-point number to be converted is stored.
D	Head address of word/double word device where conversion result (INT) will be stored.

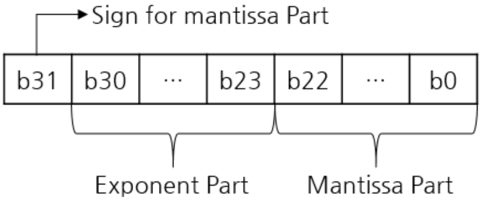
INT, INTP

- INT(P) instructions convert 32-bit floating-point number stored in double word device *S* into 16-bit INT data.
- Converted data is stored in the word device *D*.
- The range of data stored in *S*, *S*+1 is as following.
  -32768 ≦ assigned value in *S*, *S*+1 ≦ 32767
- When the data to be converted exceeds the range of -32768~32767, it overflows. No flag is turned ON.
- When the instruction executes, the 32-bit floating-point number is rounded down to the nearest ones place.
- 32-bit floating-point number is written in following format.

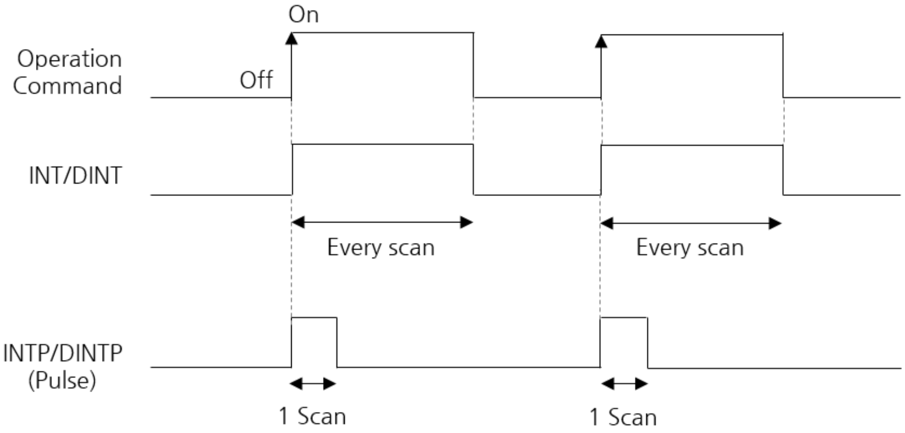


DINT, DINTP

- DINT(P) instructions convert 32-bit floating-point number stored in double word device *S* into 32-bit INT data.
- Converted data is stored in the double word device *D*.
- The range of data stored in *S*, *S*+1 is as following.
  -2147483648 ≦ assigned value in *S*, *S*+1 ≦ 2147483647
- When the data to be converted exceeds the range of -2147483648~2147483647, it overflows. No flag is turned ON.
- When the instruction executes, the 32-bit floating-point number is rounded down to the nearest ones place.
- 32-bit floating-point number is written in following format.



Execution Condition



Operation Error

Error flag (F110)

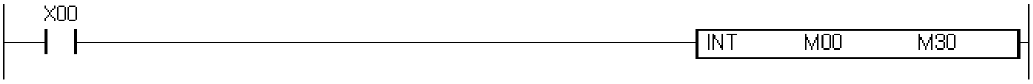
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

INT

When operation command X00 is ON, INT instruction is executed. Then, 32-bit floating-point number stored in double word device M00 (M00 ~ M1F) is converted into 16-bit INT data. The conversion result is stored in word device M30 (M30 ~ M3F).

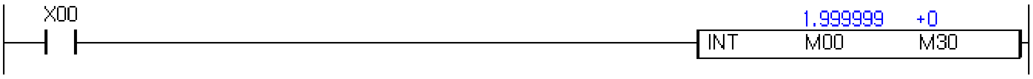
Ladder Diagram (LD)



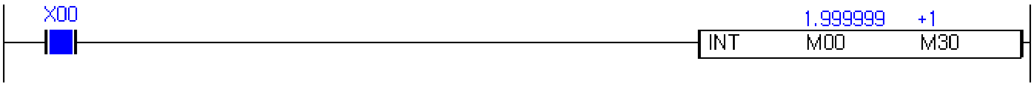
Instruction List (IL)

Instruction	Device	
LD	X00	
INT	M00	M30

- 32-bit floating-point number “1.999999” is assigned to M00.



- When X00 turns ON, INT instruction is executed. Then, 32-bit floating-point number “1.999999” is converted into 16-bit INT data “1”. The conversion result is stored in M30 (M30 ~ M3F).



2.3.6 GRY, GRYP, DGRY, DGRYP

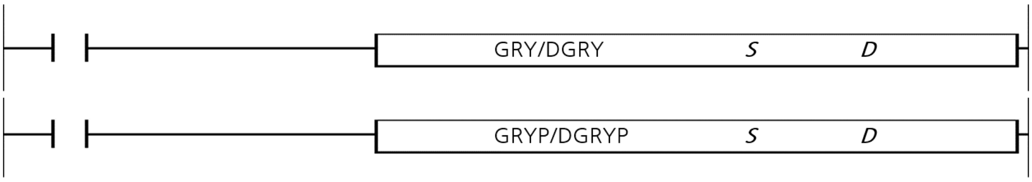
Supported PLC Series

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	-

Function

GRY(P) instructions convert 16-bit BIN data stored in word device *S* into Gray code data. Then, the conversion result is stored in word device *D*.
DGRY(P) instructions convert 32-bit BIN data stored in double word device *S* into Gray code data. Then, the conversion result is stored in double word device *D*.

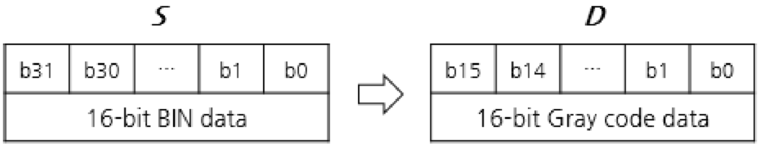
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
GRY(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
DGRY(P)	*D*	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



S	Constant or head address of word/double word device where 16/32-bit BIN data to be converted is stored.
D	Head address of word/double word devices where conversion result (Gray code) will be stored.

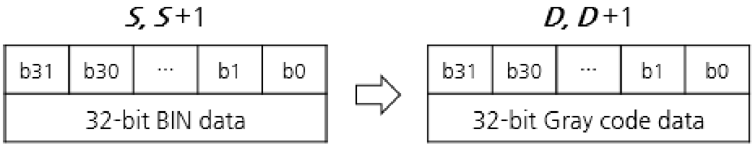
GRY, GRYP

- GRY(P) instructions convert 16-bit BIN data stored in the word device *S* into 16-bit Gray code data.
- Converted data is stored in word device *D*.
- The range of data stored in word device *S* is as following.
  0 ≦ assigned value in word device *S* ≦ 65535
- The conversion result is stored as below.

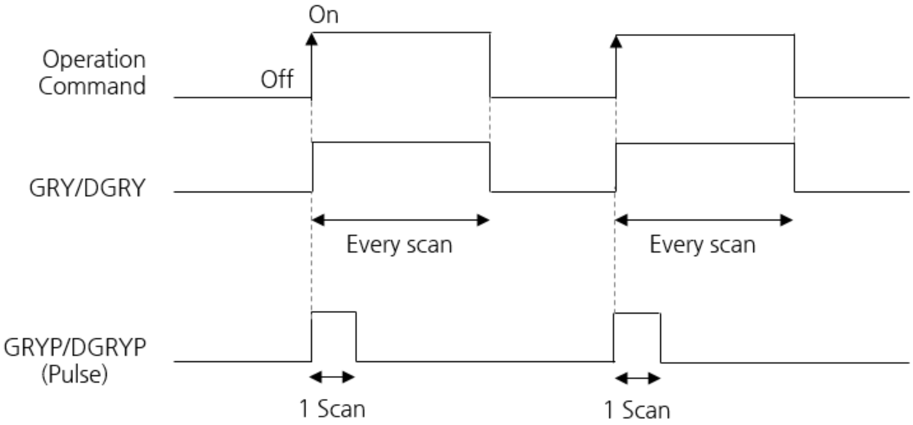


DGRY, DGRYP

- DGRY(P) instructions convert 32-bit BIN data stored in the double word device *S* into 32-bit Gray code data.
- Converted data is stored in double word device *D*.
- The range of data stored in double word device *S* is as following.
  0 ≦ assigned value in word device *S* ≦ 4294967295
- The conversion result is stored as below.



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)

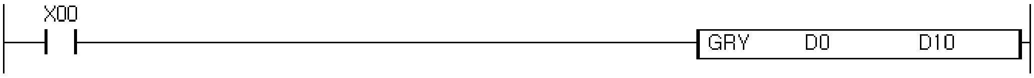
F110 turns ON for 1 scan when the value assigned to *S* is negative.

Program Example

GRY

When operation command X00 is ON, GRY instruction is executed. Then, 16-bit BIN data stored in word device D0 is converted into 16-bit Gray code data. The conversion result is stored starting from word device D10.

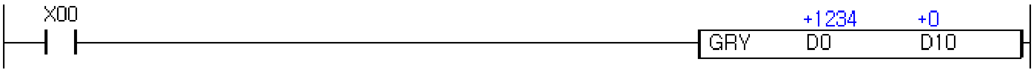
Ladder Diagram (LD)



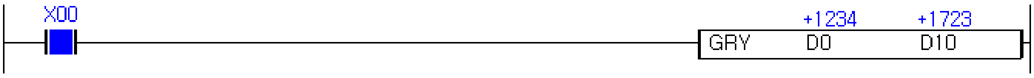
Instruction List (IL)

Instruction	Device	
LD	X00	
GRY	D0	D10

- 16-bit BIN data “1234” is assigned to D0.



- When X00 turns ON, GRY instruction is executed. Then, 16-bit BIN data “1234” is converted into 16-bit Gray code data “1723”. The conversion result is stored in D10.



2.3.7 GBIN, GBINP, DGBIN, DGBINP

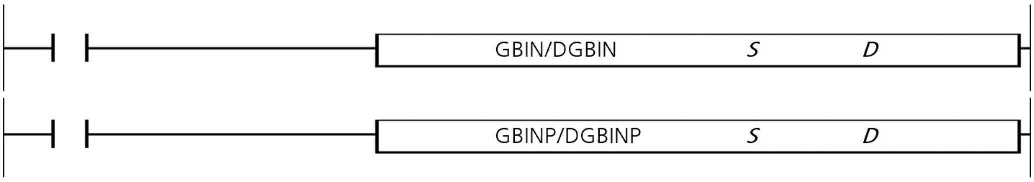
Supported PLC Series

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

GBIN(P) instructions convert 16-bit Gray code data stored in word device *S* into 16-bit INT data. Then, the conversion result is stored in word device *D*.
DGBIN(P) instructions convert 32-bit Gray code data stored in double word device *S* into 32-bit INT data. Then, the conversion result is stored in double word device *D*.

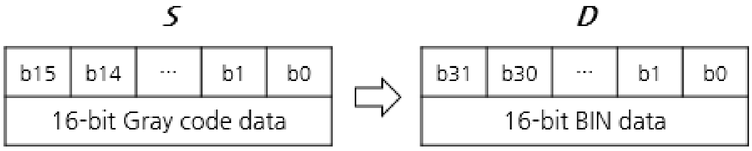
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
GBIN(P) DGBIN(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Head address of word/double word device where Gray code data to be converted is stored.
D	Head address of word/double word device where conversion result (INT) will be stored.

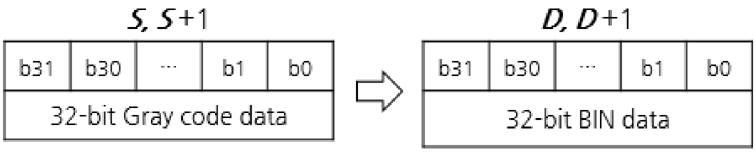
GBIN, GBINP

- GBIN(P) instructions convert 16-bit Gray code data stored in word device *S* into 16-bit INT data.
- Converted data is stored in the word device *D*.
- The range of data stored in word device *S* is as following.
  0 ≦ assigned value in word device *S* ≦ 65535
- The conversion result is stored as below.

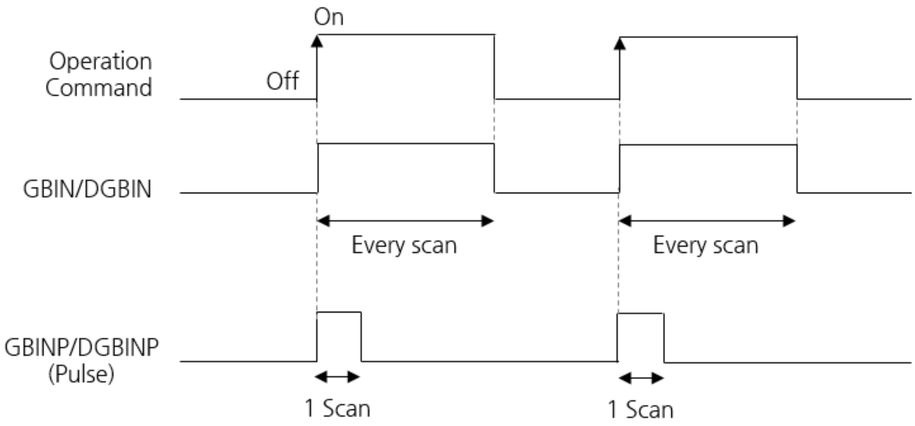


DGBIN, DGBINP

- DGBIN(P) instructions convert 32-bit Gray code data stored in the double word device *S* into 32-bit INT data.
- Converted data is stored in double word device *D*.
- The range of data stored in double word device *S* is as following.
  0 ≦ assigned value in word device *S* ≦ 4294967295
- The conversion result is stored as below.



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)

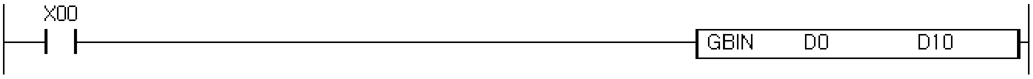
F110 turns ON for 1 scan when the value assigned to *S* is negative.

Program Example

GBIN

When operation command X00 is ON, GBIN instruction is executed. Then, 16-bit Gray code data stored in word device D0 is converted into 16-bit INT data. The conversion result is stored in word device D10.

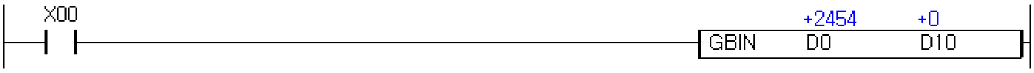
Ladder Diagram (LD)



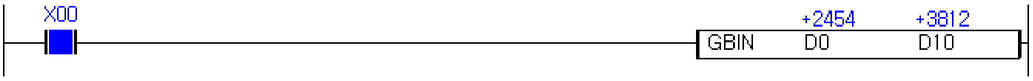
Instruction List (IL)

Instruction	Device	
LD	X00	
GBIN	D0	D10

- 16-bit Gray code data “2454” is assigned to D0.



- When X00 turns ON, GBIN instruction is executed. Then, 16-bit Gray code data “2454” is converted into 16-bit INT data “3812”. The conversion result is stored in D10.



2.4 Data Processing Instructions

2.4.1 MAX, MAXP, DMAX, DMAXP

Supported PLC Series

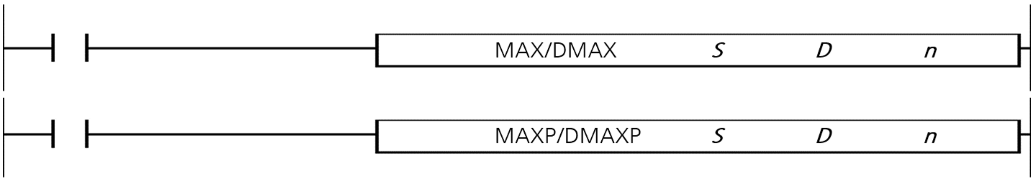
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

MAX(P) searches the maximum value among *n* blocks of word data starting from word device *S*. Then, stores the maximum value in word device *D*.

DMAX(P) searches the maximum value among *n* blocks of double word data starting from double word device *S*. Then, stores the maximum value in double word device *D*.

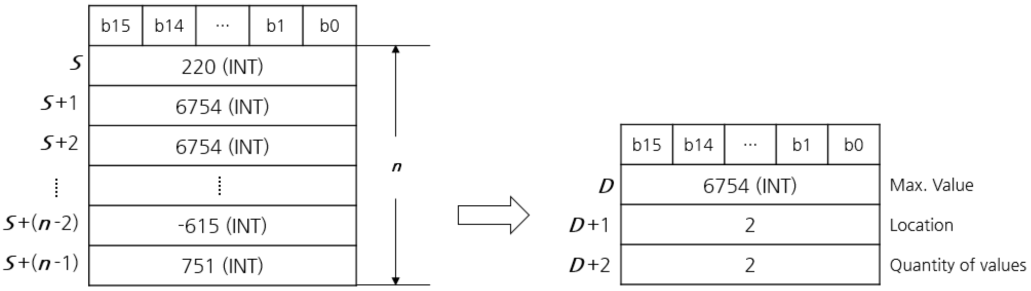
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
MAX(P) DMAX(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S	Head address of the devices to search maximum value.
D	Head address of the devices where searched maximum value will be stored.
D+1/*D*+2	Head address of the word devices where the first location of maximum value data is stored.
D+2/*D*+3	Head address of the word devices where the quantity of maximum value data is stored.
n	The number of word/double word data to be searched, starting from *S*.

MAX, MAXP

- MAX(P) search the maximum value among the *n* blocks of word data, starting from the word device *S*. Then, the searched maximum value is stored in the word device *D*.
- At *S* and *D*, value from -32768 to 32767 can be specified.
- In word device *D*+1, the location of the searched maximum value data is stored. If multiple maximum value data are searched, the first location of the maximum value data is stored. From 1 to *n* can be stored in word device *D*+1.
- At word device *D*+2, the quantity of maximum value data is stored. From 1 to *n* can be stored in word device *D*+2.
- If *n* exceeds the range of 1 ~ 32767, the instruction is not executed.

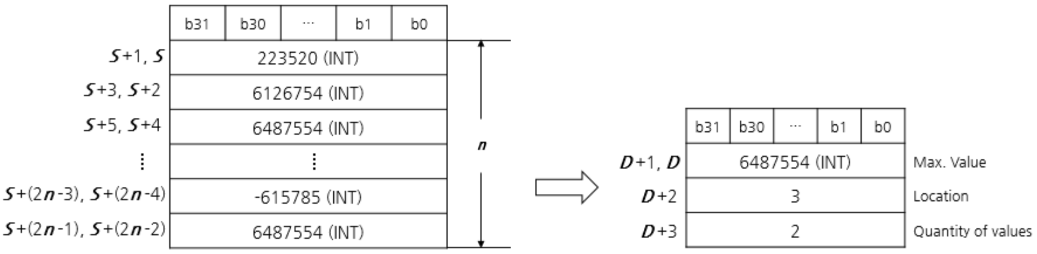


	b15 | b14 | ··· | b1 | b0
S	220 (INT)
S+1	6754 (INT)
S+2	6754 (INT)
⋮	⋮
S+(*n*-2)	-615 (INT)
S+(*n*-1)	751 (INT)

	b15 | b14 | ··· | b1 | b0	
D	6754 (INT)	Max. Value
D+1	2	Location
D+2	2	Quantity of values

DMAX, DMAXP

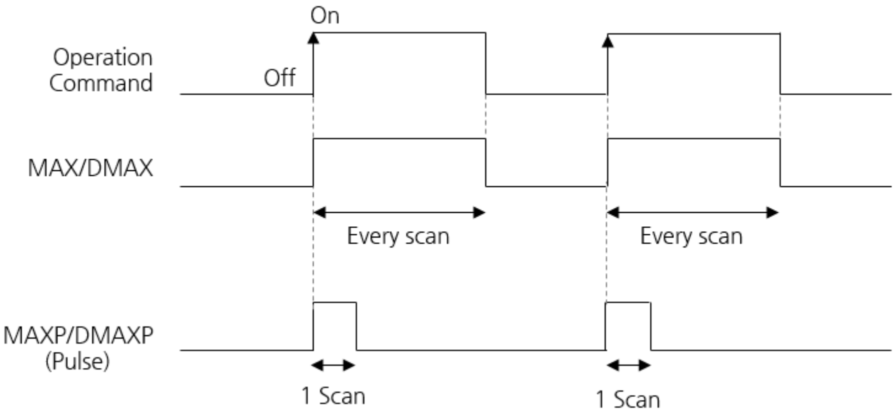
- DMAX(P) search the maximum value among the *n* blocks of double word data, starting from the double word device *S*. Then, the searched value is stored in the double word device *D.*
- At *S* and *D*, value from -2147483648 to 2147483647 can be specified.
- At word device *D* +2, the location of the searched maximum value data is stored. If multiple maximum value data are searched, the first location of the maximum value data is stored. From 1 to *n* can be stored in word device *D* +2.
- At word device *D* +3, the quantity of maximum value data is stored. From 1 to *n* can be stored in word device *D* +3.
- If *n* exceeds the range of 1 ~ 32767, the instruction is not executed.



	b31	b30	···	b1	b0
S+1, *S*	223520 (INT)				
S+3, *S*+2	6126754 (INT)				
S+5, *S*+4	6487554 (INT)				
⋮	⋮				
S+(2*n*-3), *S*+(2*n*-4)	-615785 (INT)				
S+(2*n*-1), *S*+(2*n*-2)	6487554 (INT)				

	b31	b30	···	b1	b0	
D+1, *D*	6487554 (INT)					Max. Value
D+2	3					Location
D+3	2					Quantity of values

Execution Condition



Operation Error

Error Flag (F110)

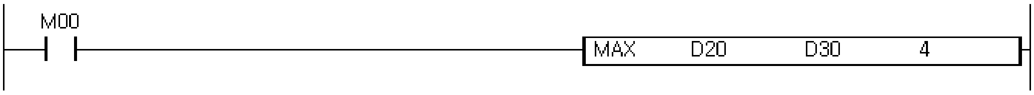
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

MAX

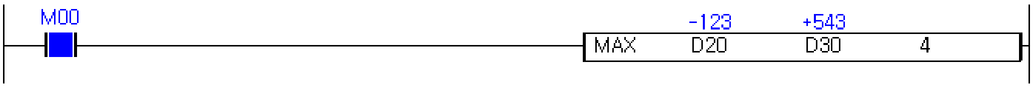
When operation command M00 is ON, MAX instruction is executed and searches for the maximum value in 4 word devices (D20 ~ D23).
Then, MAX instruction stores the maximum value in word device D30, location of maximum value data in word device D31 and the quantity of maximum value data in word device D32.

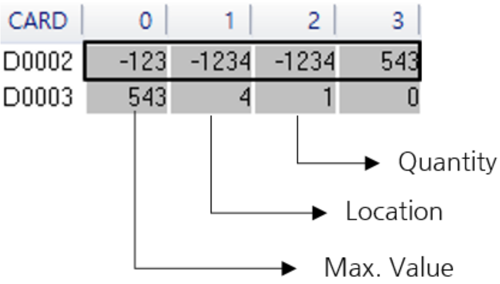
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
MAX	D20	D30	4





CARD	0	1	2	3
D0002	-123	-1234	-1234	543
D0003	543	4	1	0

2.4.2 MIN, MINP, DMIN, DMINP

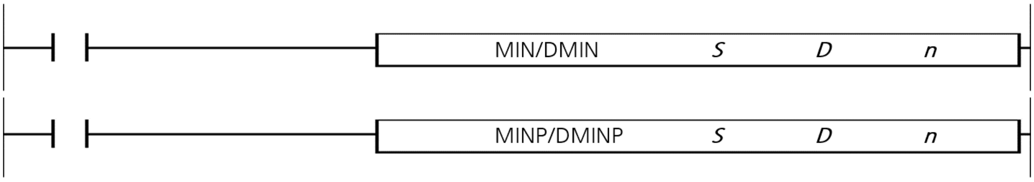
Supported PLC Series

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

MIN(P) searches the minimum value among *n* blocks of word data starting from word device *S*. Then, stores the minimum value in word device *D*.
DMIN(P) searches the minimum value among *n* blocks of double word data starting from double word device *S*. Then, stores the minimum value in double word device *D*.

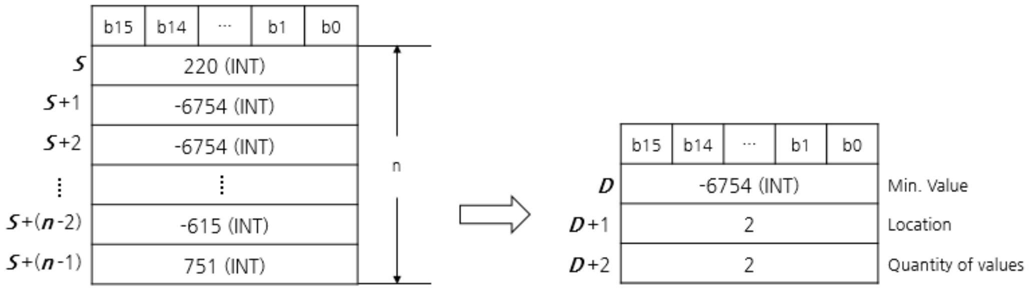
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
MIN(P) DMIN(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S	Head address of the devices to search minimum value.
D	Head address of the devices where searched minimum value will be stored.
D+1/*D*+2	Head address of the word devices where the first location of minimum value data is stored.
D+2/*D*+3	Head address of the word devices where the quantity of minimum value data is stored.
n	The number of word/double word data to be searched, starting from *S*.

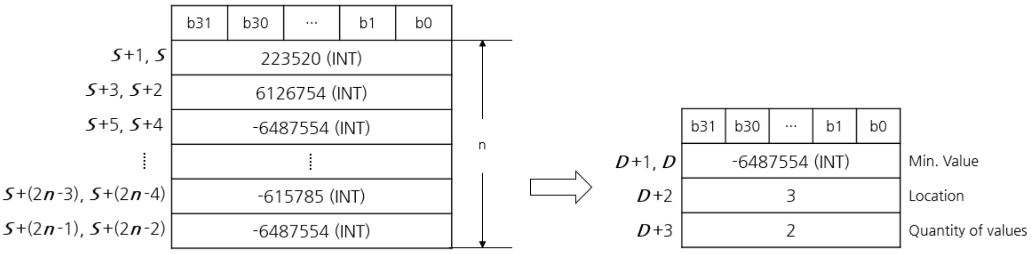
MIN, MINP

- MIN(P) search the minimum value among the *n* blocks of word data, starting from the word device *S*. Then, the searched value is stored in the word device *D*.
- At *S* and *D*, value from -32768 to 32767 can be specified.
- At word device *D*+1, the location of the searched minimum value data is stored. If multiple minimum value data are searched, the first location of the minimum value data is stored. From 1 to *n* can be stored in word device *D*+1.
- At word device *D*+2, the quantity of minimum value data is stored. From 1 to *n* can be stored in word device *D*+2.
- If *n* exceeds the range of 1 ~ 32767, the instruction is not executed.



DMIN, DMINP

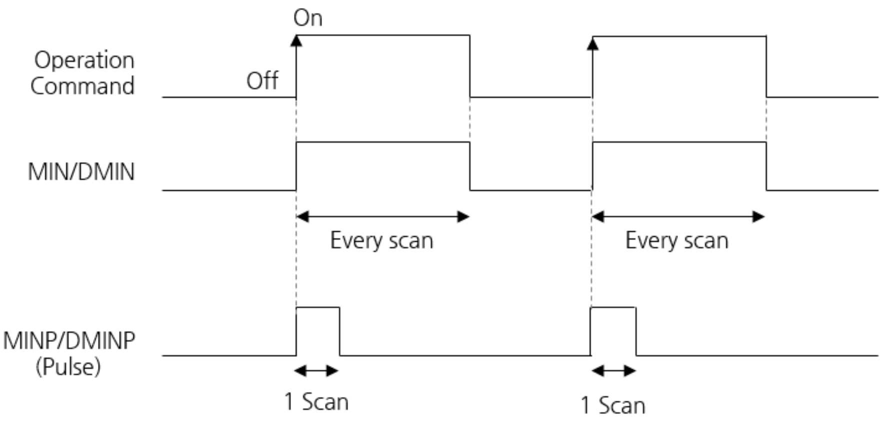
- DMIN(P) search the minimum value among the *n* blocks of double word data, starting from the double word device *S*. Then, the searched value is stored in the double word device *D*.
- At *S* and *D*, value from -2147483648 to 2147483647 can be specified.
- At word device *D* +2, the location of the searched minimum value data is stored. If multiple minimum value data are searched, the first location of the minimum value data is stored. From 1 to *n* can be stored in word device *D* +2.
- At word device *D* +3, the quantity of minimum value data is stored. From 1 to *n* can be stored in word device *D* +3.
- If *n* exceeds the range of 1 ~ 32767, the instruction is not executed.



	b31	b30	···	b1	b0
S+1, *S*	223520 (INT)				
S+3, *S*+2	6126754 (INT)				
S+5, *S*+4	-6487554 (INT)				
⋮	⋮				
S+(2*n*-3), *S*+(2*n*-4)	-615785 (INT)				
S+(2*n*-1), *S*+(2*n*-2)	-6487554 (INT)				

	b31	b30	···	b1	b0	
D+1, *D*	-6487554 (INT)					Min. Value
D+2	3					Location
D+3	2					Quantity of values

Execution Condition



Operation Error

Error Flag (F110)

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

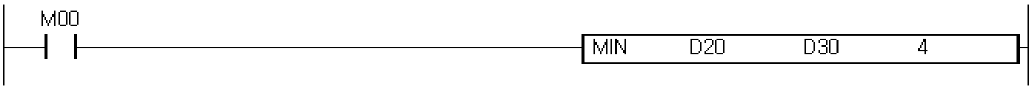
Program Example

MIN

When operation command M00 is ON, MIN instruction is executed and searches for the minimum value in 4 word devices (D20 ~ D23).

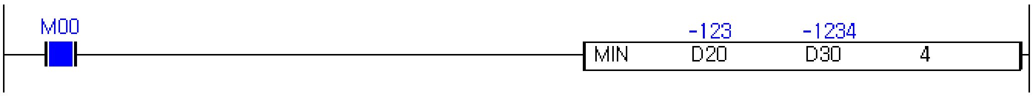
Then, MIN instruction stores the minimum value in word device D30, location of minimum value data in word device D31 and the quantity of minimum value data in word device D32.

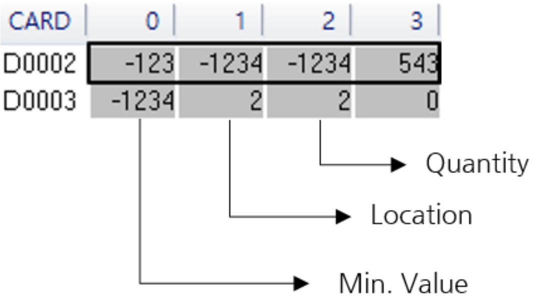
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M00		
MIN	D20	D30	4





CARD	0	1	2	3
D0002	-123	-1234	-1234	543
D0003	-1234	2	2	0

2.4.3 SUM, SUMP, DSUM, DSUMP

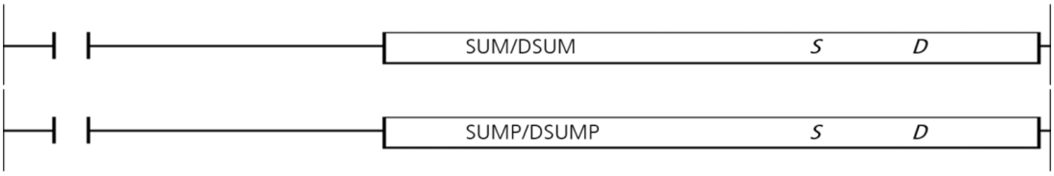
Supported PLC Series

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

SUM(P) and DSUM(P) count the number of bits which are set. Then, instructions store the number of set bits in the assigned device.

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



S	Head address of word/double word device where to count the total number of bits which are set.
D	Head address of word device to store the total number of bits which are set.

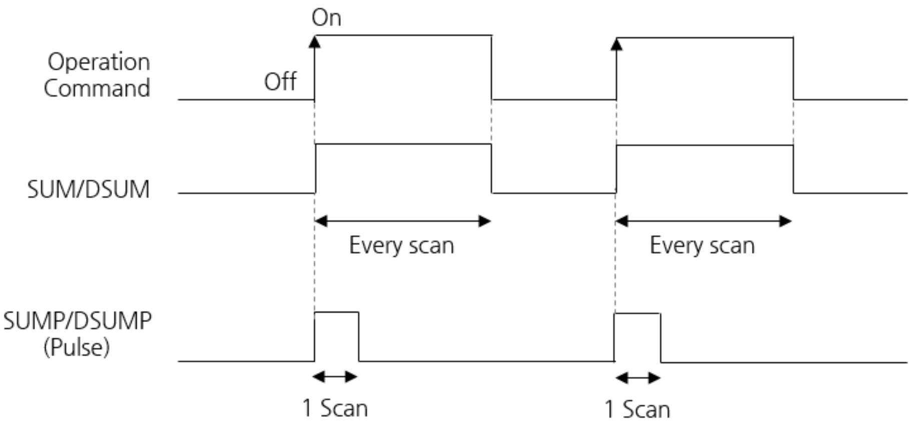
SUM, SUMP

- Counts the total number of bits which are set in a word device. The sum starts from the word device assigned to *S*.
- SUM(P) instruction counts the total number of bits which are set from 16-bit data in the word device assigned to *S*.
- When the instruction executes, the result is stored in the device assigned to *D*.

DSUM, DSUMP

- Counts the total number of bits which are set in a double word device. The sum starts from the double word device assigned to *S*.
- DSUM(P) instruction counts the total number of bits which are set from 32-bit data in the device assigned to *S*.
- When the instruction executes, the result is stored in the device assigned to *D*.

Execution Condition



Operation Error

Error Flag (F110)

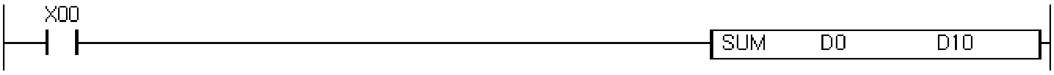
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

SUM, SUMP

When X00 is turned ON, the program counts the total number of bits which are set in D0. The result is stored at D10.

Ladder Diagram

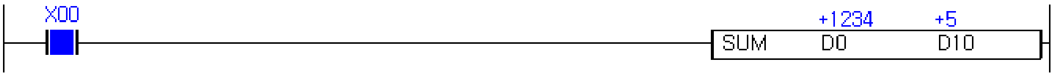


Instruction List

Instruction	Device	
LD	X00	
SUM	D0	D10

The program operates as following:

- Turn X00 ON. When D0 is “1234” (BIN=0000 0100 1101 0010), there are 5 bits which are set. Therefore, the result “5” is stored in D10.



2.4.4 SEG, SEGP

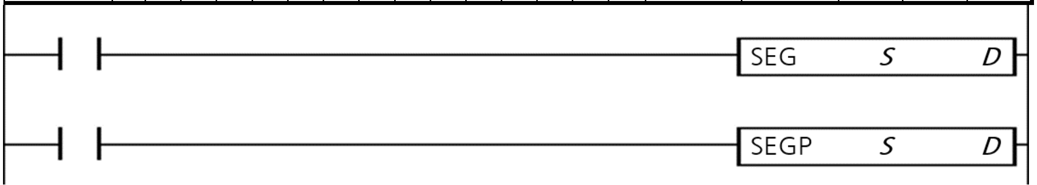
Supported PLC Series

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

SEG(P) instructions convert each digit (4-bit, in total 4 digits) of hexadecimal data stored in word device *S* into 7-segment display code (1 byte, in total 4 bytes = 2 words). Then, the conversion result is stored in double word device *D*.

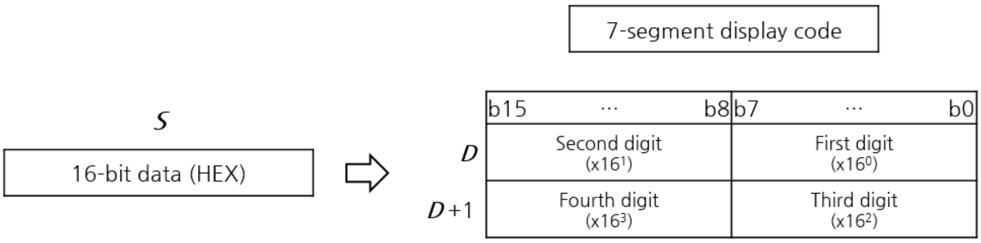
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SEG(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



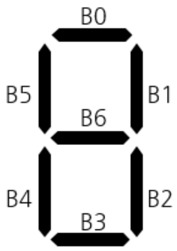
S	Constant or address of word device where hexadecimal data to be converted is stored.
D	Head address of double word device where conversion result (7-segment display code) will be stored.

SEG, SEGP

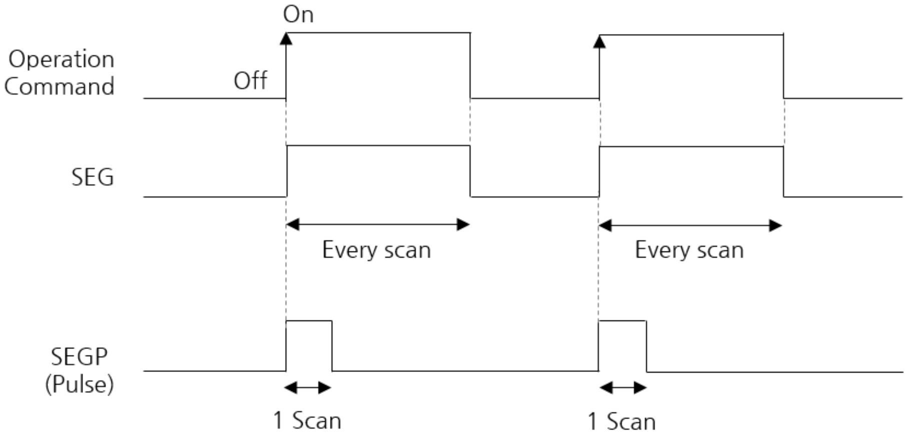
- SEG(P) instructions convert each digit of 16-bit data in hexadecimal notation, stored in the word device *S*, into 7-segment display code.
- Converted data is stored in device *D* and *D*+1.
- The range of data stored in device *S* is as following.
  H0000 ≦ assigned value in device *S* ≦ HFFFF
- It is possible to express arabic numerals (0~9) and alphabets (A~F).
- The 16^0's place of the conversion result is stored in the lower byte of device *D*.



	b15 ··· b8	b7 ··· b0
D	Second digit ($\times 16^1$)	First digit ($\times 16^0$)
D+1	Fourth digit ($\times 16^3$)	Third digit ($\times 16^2$)

S		Configuration of 7 segments	*D*									Display
HEX	Bit pattern		B7	B6	B5	B4	B3	B2	B1	B0	HEX	
0	0000		0	0	1	1	1	1	1	1	3F	0
1	0001		0	0	0	0	0	1	1	0	06	1
2	0010		0	1	0	1	1	0	1	1	5B	2
3	0011		0	1	0	0	1	1	1	1	4F	3
4	0100		0	1	1	0	0	1	1	0	66	4
5	0101		0	1	1	0	1	1	0	1	6D	5
6	0110		0	1	1	1	1	1	0	1	7D	6
7	0111		0	0	1	0	0	1	1	1	27	7
8	1000		0	1	1	1	1	1	1	1	7F	8
9	1001		0	1	1	0	1	1	1	1	6F	9
A	1010		0	1	1	1	0	1	1	1	77	A
B	1011		0	1	1	1	1	1	0	0	7C	b
C	1100		0	0	1	1	1	0	0	1	39	C
D	1101		0	1	0	1	1	1	1	0	5E	d
E	1110		0	1	1	1	1	0	0	1	79	E
F	1111		0	1	1	1	0	0	0	1	71	F

Execution Condition



Operation Error

Error flag (F110)

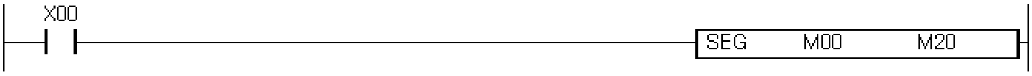
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

SEG

When operation command X00 is ON, SEG instruction is executed. Then, 16-bit data in hexadecimal notation, stored in word device M00 (M00 ~ M0F), is converted into 7-segment display code. The result of conversion is stored in double word device M20 (M20 ~ M3F).

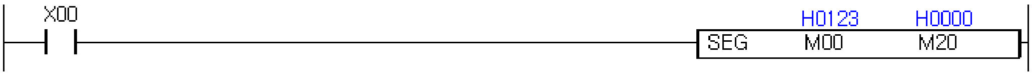
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
SEG	M00	M20

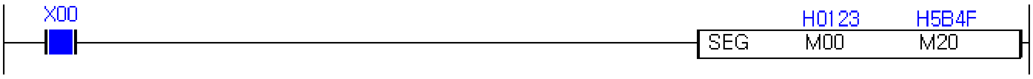
- 16-bit data “H0123” is assigned to M00.

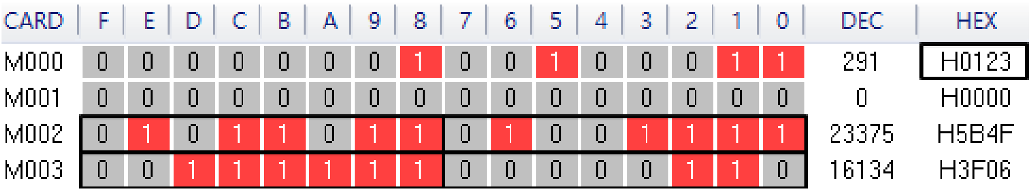


- When X00 turns ON, SEG instruction is executed. Then, 16-bit data “H0123” is convertd into 7-segment display code “0123”.

- The conversion result is stored in M20 (M20 ~ M3F).

- “0” is stored as “H3F” in upper byte of M30 (M38~M3F). “1” is stored as “H06” in lower byte of M30 (M30~M37). “2” is stored as “H5B” in upper byte of M20 (M28~M2F). “3” is stored as “H4F” in lower byte of M20 (M20~M27).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX
M000	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	291	H0123
M001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	H0000
M002	0	1	0	1	1	0	1	1	0	1	0	0	1	1	1	1	23375	H5B4F
M003	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	0	16134	H3F06

2.4.5 DECO, DECOP

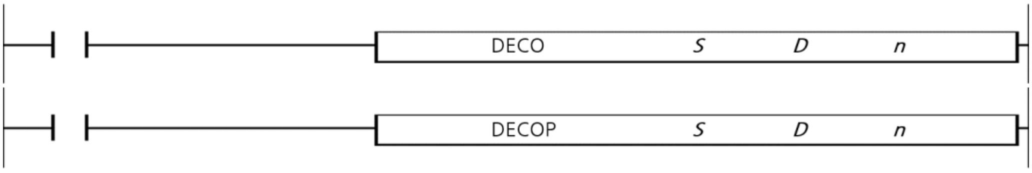
Supported PLC Series

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

DECO(P) instructions read the absolute value of *n* bits which is stored in the word device *S*. Then, the corresponding bits starting from the word device *D* are turned ON and all other bits in the word device *D* are turned 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
DECO(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

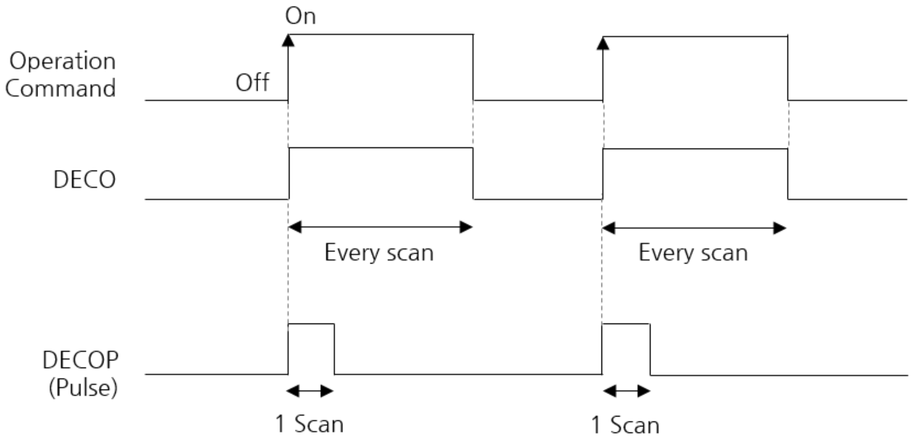


S	Address of word device where data to be decoded is stored.
D	Head address of word devices where decoding result will be stored.
n	The number of bits to be decoded in device *S*. (1~8)

DECO, DECOP

- DECO(P) instructions read the absolute value of *n* bits which is stored in the word device *S*. Then, turns ON the corresponding bit starting from the word device *D*, and turns OFF all other bits in the word device *D*.
- DECO(P) instructions decode maximum 8 bits into 256 bits.
- *n* bits are decoded starting from the device *S*.
- According to the absolute value of specified bits, the corresponding bit starting from device *D* is turned ON.
- *n* can be assigned from 1 to 8. It means that you can use maximum 8 bits starting from the least significant bit of the device *S*.
- When DECO(P) instructions operate, 2^n bits starting from the device *D* are preoccupied.

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)

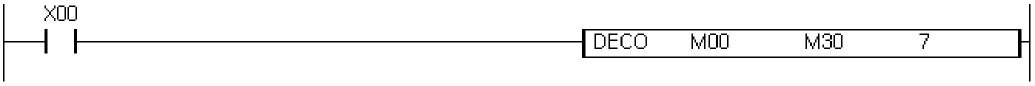
F110 turns ON for 1 scan when the value of *n* exceeds the range of 1~8.

Program Example

DECO

When operation command X00 is ON, DECO instruction is executed. Specified 7 bits, starting from M00 (M00~M06), will be decoded. The instruction reads the absolute value of M00 (M00~M06). Then, the corresponding bit starting from device M30 turns ON.

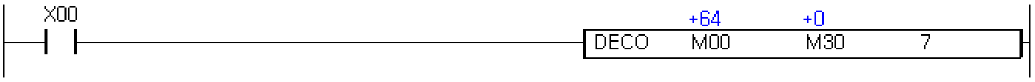
Ladder Diagram (LD)



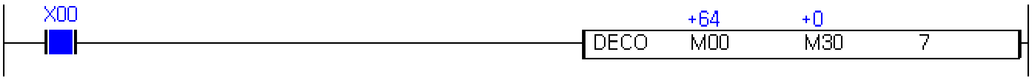
Instruction List (IL)

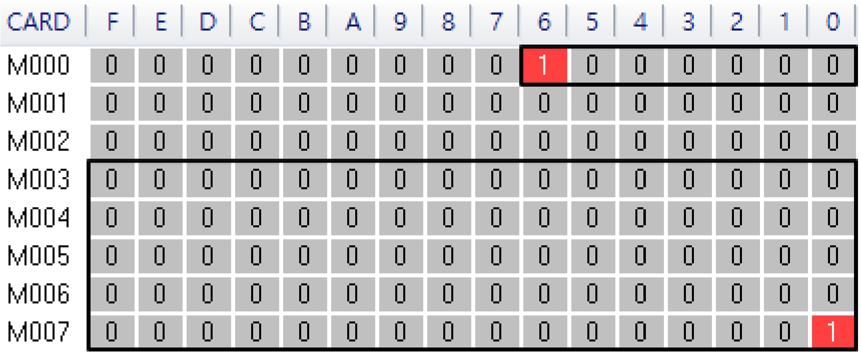
Instruction	Device		
LD	X00		
DECO	M00	M30	7

- "64" assigned to M00 (M00~M06).



- When X00 turns ON, DECO instruction is executed. The instruction reads the absolute value of specified 7 bits starting from M00 (M00~M06). Since the value is "64", bit 64 of M30 turns ON. Bit 64 of M30 is M70.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
M001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M006	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M007	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

2.4.6 ENCO, ENCOP

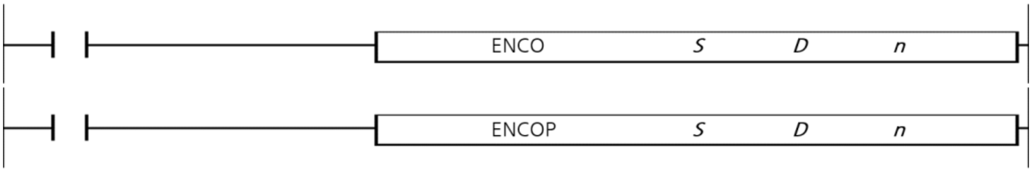
Supported PLC Series

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

ENCO(P) instructions find location of the most significant bit which is set within the specified 2^n bits starting from device *S*. Then, according to the location of the most significant bit which is set, the corresponding location value is stored in device *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
ENCO(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

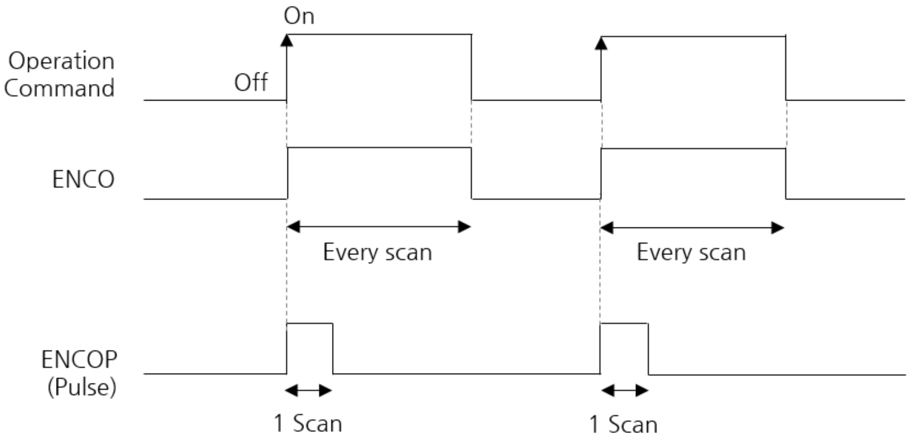


S	Constant or head address of word devices where data to be encoded is stored.
D	Address of word device where encoding result will be stored.
n	The exponent of 2 which defines the number of bits to be encoded in device *S*. (1~8)

ENCO, ENCOP

- ENCO(P) instructions find location of the most significant bit which is set within the specified 2^n bits starting from device *S*. Then, the instructions store the location value of the most significant bit which is set, in device *D* as integer.
- ENCO(P) instructions encode maximum 256 bits (=16 words) into 8 bits.
- 2^n bits are encoded starting from the device *S*.
- According to the location of the most significant bit which is set, the corresponding location value is stored in device *D*.
- *n* can be assigned from 1 to 8. If *n* exceeds its range, the instructions do not operate.

Execution Condition



Operation Error

Error flag (F110)

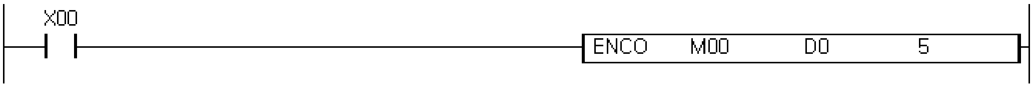
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

ENCO

When operation command X00 is ON, ENCO instruction is executed. The instruction finds the most significant bit, which is set as “1”, within range of 2^5 bits (= 32 bits = 2 words) starting from device M00 (M00~M1F). Then, its location value is encoded. The encoded value is stored in the word device D0.

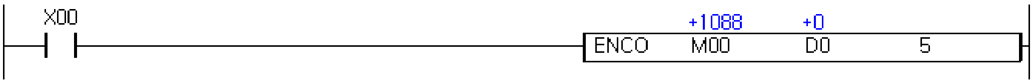
Ladder Diagram (LD)



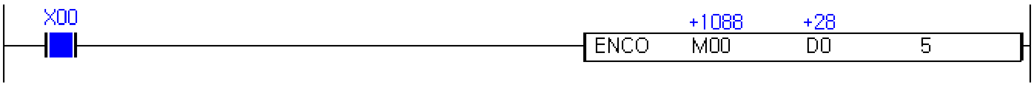
Instruction List (IL)

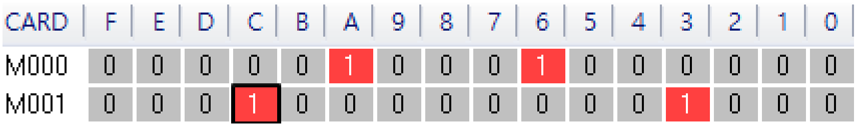
Instruction	Device		
LD	X00		
ENCO	M00	D0	5

- Bit 6, 10, 19 and 28 are set within range of 2^5 bits (= 32 bits = 2 words) starting from device M00 (M06, M0A, M13 and M1C).



- When X00 turns ON, ENCO instruction is executed. Location of the most significant bit which is set is M1C. The location of M1C from M00 is 28. Therefore, “28” is stored in device D0.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
M001	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

2.4.7 DIS, DISP

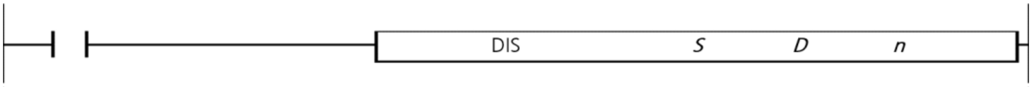
Supported PLC Series

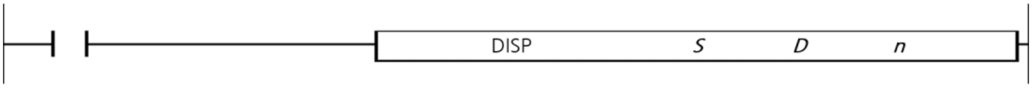
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

DIS(P) instructions split 16-bit data (= 1 word) stored in word device *S* into 4 nibbles (4-bit data). Each nibble is distributed to *n* word devices starting from *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
DIS(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

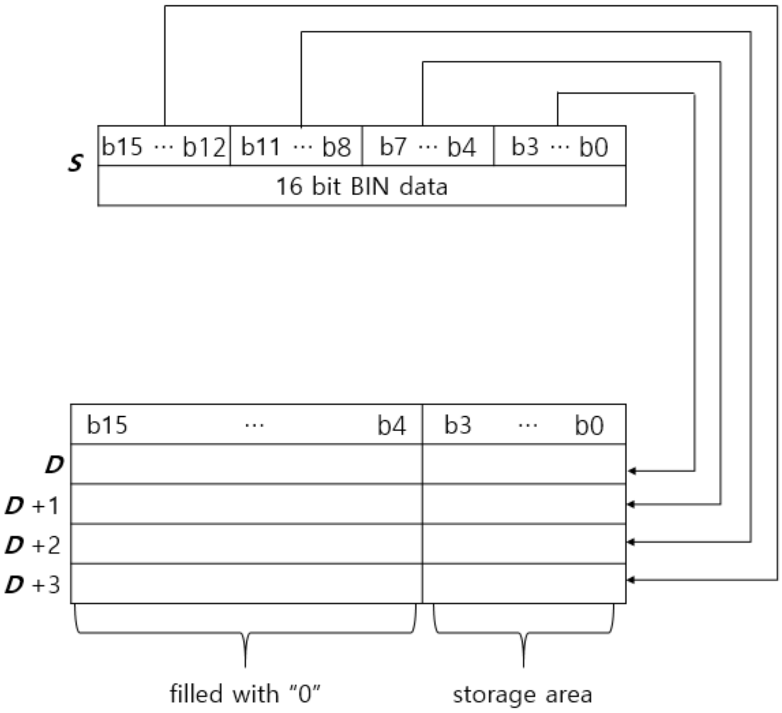




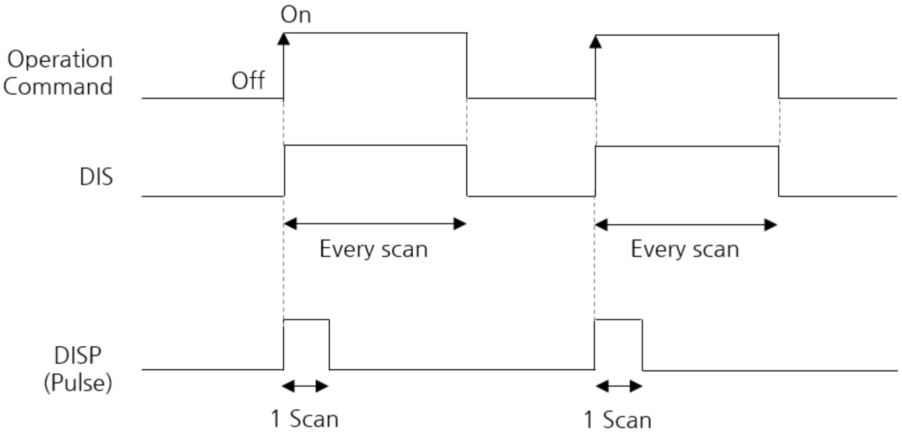
S	Constant or address of word device where data to be distributed is stored.
D	Head address of word devices where distribution result will be stored.
n	The number of 4-bit data to be distributed. (1~4)

DIS, DISP

- DIS(P) instructions split 16-bit data (= 1 word) stored in word device *S* into 4 nibbles.
- The number of nibbles to be distributed is specified by *n*.
- Nibbles starting from the least significant bit of word device *S* are distributed to *n* word devices starting from *D*.
- Distributed data are stored in bit 0 ~ bit 3 of each destination device. Other bits in destination device are filled with “0”.
- *n* can be assigned in range of 1~4. When *n* exceeds its range, the instruction does not operate.
- The least significant nibble stored in word device *S* is distributed to the bit 0 ~ bit 3 of word device *D*.



Execution Condition



Operation Error

Error flag (F110)

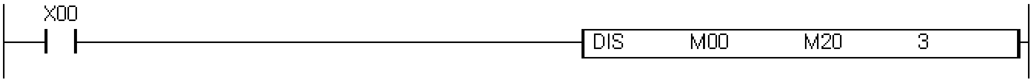
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DIS

When operation command X00 is ON, DIS instruction is executed. Then, 16-bit data stored in word device M00 (M00 ~ M0F), is split into 4 nibbles. The 3 nibbles are distributed to the 3 word devices, starting from M20 (M20~M4F). Distributed data are stored in the bit 0 ~ bit 3 of each word device.

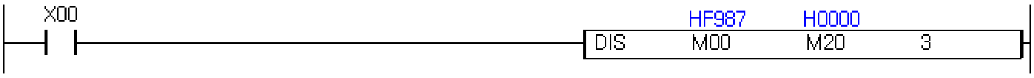
Ladder Diagram (LD)



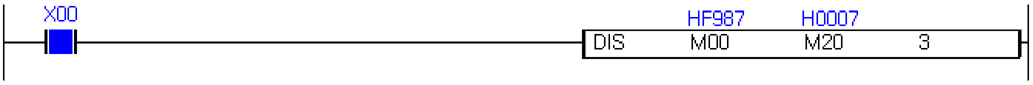
Instruction List (IL)

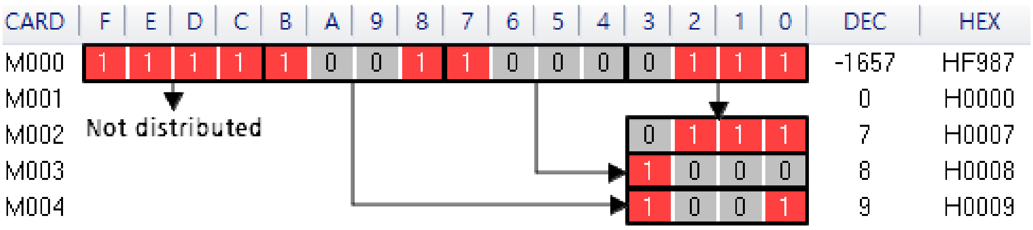
Instruction	Device		
LD	X00		
DIS	M00	M20	3

- 16-bit data “HF987” (1111 1001 1000 0111) is assigned to word device M00 (M00~M0F).



- When X00 turns ON, DIS instruction is executed. Then, 16-bit data “HF989” is split into 4 nibbles. In hexadecimal, the split data are as following:
  “H000F”, “H0009”, “H0008”, “H0007”.
- 3 nibbles starting from the least significant bit are distributed to three word devices starting from M20 (M20~M4F).
- The most significant nibble “HF” in word device M00 (M00~M0F) is not distributed.
- The most significant nibble “H9”, among the 3 nibbles, is stored in device M40 (M40~M4F).
- The second nibble “H8” is stored in device M30 (M30~M3F).
- The least significant nibble “H0007” is stored in device M20 (M20~M2F).
- M24~M2F, M34~M3F, M44~M4F are filled with “0”.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX
M000	1	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1	-1657	HF987
M001																	0	H0000
M002													0	1	1	1	7	H0007
M003													1	0	0	0	8	H0008
M004													1	0	0	1	9	H0009

2.4.8 UNI, UNIP

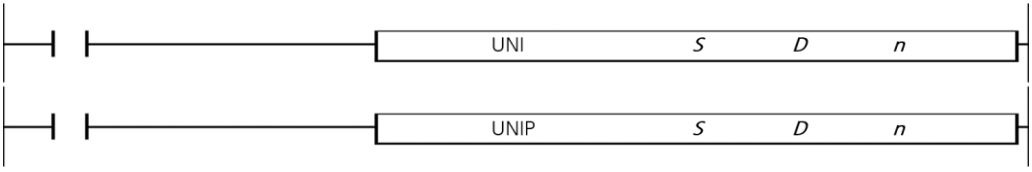
Supported PLC Series

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

UNI(P) instructions link least significant nibbles stored in *n* separate word devices starting from *S*. Linked data are stored in word device *D*.

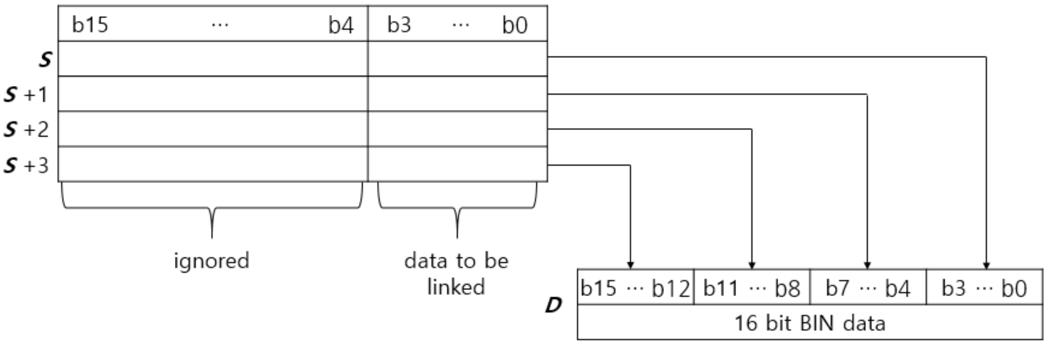
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
UNI(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



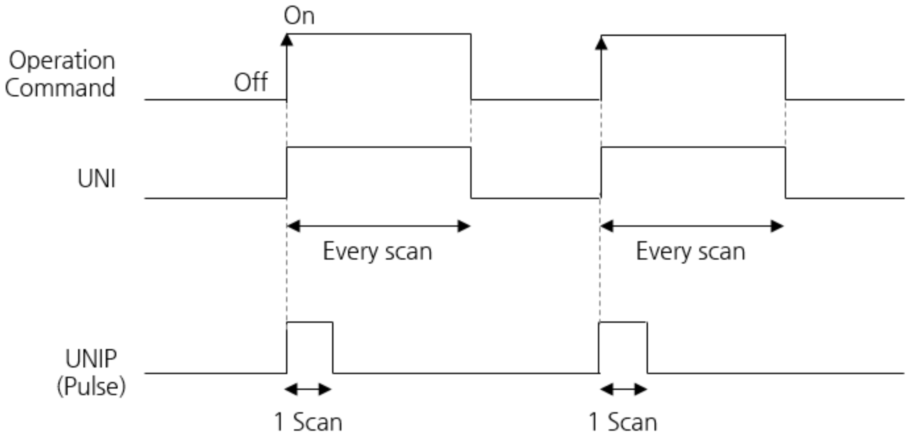
S	Head address of word devices where data (nibble) to be linked is stored.
D	Address of word device where linked data will be stored.
n	The number of data (nibble) to be linked. (1~4)

UNI, UNIP

- UNI(P) instructions link least significant nibbles stored in *n* separate word devices starting from *S*. Linked data are stored in word device *D*.
- The number of nibbles to be linked is specified by *n*.
- The least significant nibble of each *n* word devices, starting from *S*, are linked.
- Linked data is stored in the word device *D*.
- When less than 4 nibbles are linked, the more significant bits of device *D* are filled with "0".
- *n* can be assigned in range of 1~4. When *n* exceeds its range, the instruction does not operate.
- The least significant nibble stored in word device *S* is distributed to the bit 0 ~ bit 3 of word device *D*.



Execution Condition



Operation Error

Error flag (F110)

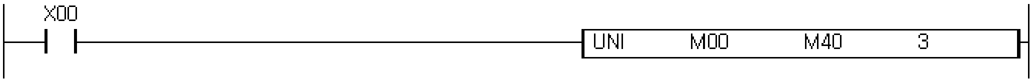
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

UNI

When operation command X00 is ON, UNI instruction is executed. Then, the least significant nibbles stored in 3 word devices starting from M00 (M00 ~ M2F) are linked. Linked data is stored in the word device M40 (M40~M4F).

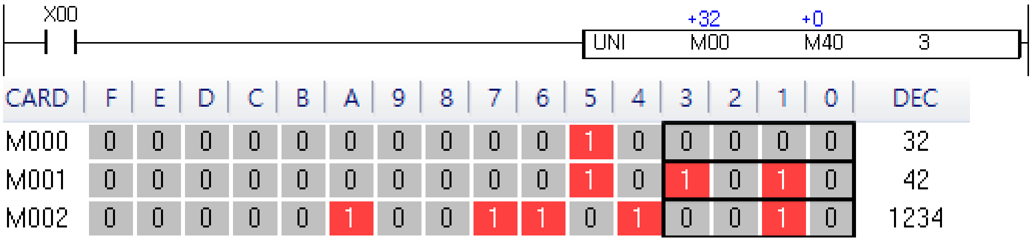
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	X00		
UNI	M00	M40	3

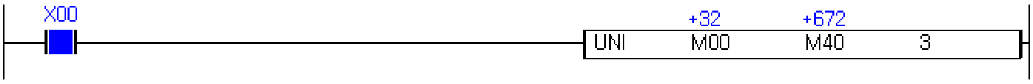
- “32” is assigned to M00 (M00~M0F). “42” is assigned to M10 (M10~M1F). “1234” is assigned to M20 (M20~M2F).

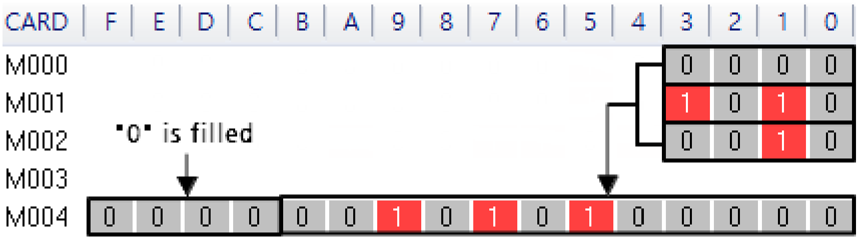


CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC
M000	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	32
M001	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	42
M002	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1234

- When X00 turns ON, UNI instruction is executed. Then, each least significant nibble is linked.

- Since 3 nibbles are linked, “0” are filled in the most significant nibble in word device M40 (M40~M4F). The linked data to be stored in M40 (M40~M4F) is therefore “0000 0000 1010 0010”.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000													0	0	0	0
M001													1	0	1	0
M002													0	0	1	0
M003																
M004	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0

2.4.9 SCL, SCLP, DSCL, DSCLP

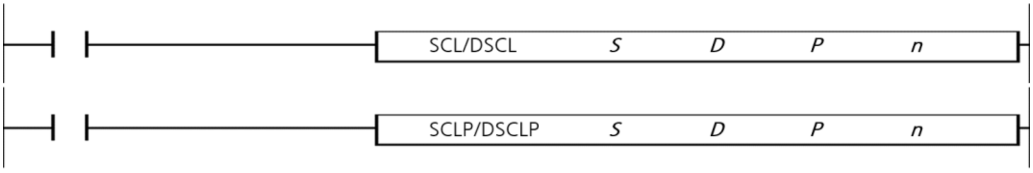
Supported PLC Series

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

SCL(P) instructions execute scaling of word data according to the specified linear function. DSCL(P) instructions execute scaling of double word data according to the specified linear function.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SCL(P) DSCL(P)	*S*	O	O	O	O	O	-	-	-	-	O	O	O	O	O	-	5	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	O	O	O	O	-				
	P	O	-	O	O	O	-	-	-	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S	Head address of word devices where data to be scaled is stored.
D	Head address of word/double word devices where result of data scaling will be stored.
P	Head address of data table where coordinate data to define linear function are stored.
n	Number of data to be scaled.

SCL, SCLP

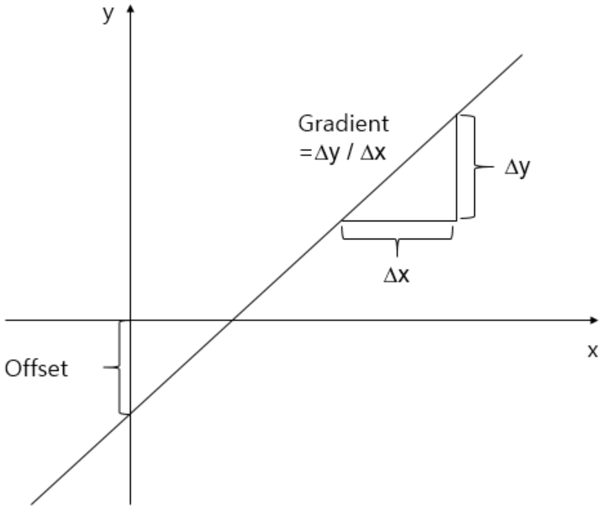
- SCL(P) instructions scale data of *n* words starting from word device *S*. The scaling is executed according to the linear function defined with data assigned to data table *P*. Scaled data are stored in *n* word devices starting from device *D*.
- Data table starting from word device *P* is comprised as below:

P	Range start point of data to be scaled (Start.*S*) ·· x axis
P+1	Range end point of data to be scaled (End.*S*) ·· x axis
P+2	Range start point of scaled data (Start.*D*) ·· y axis
P+3	Range end point of scaled data (End.*D*) ··· y axis

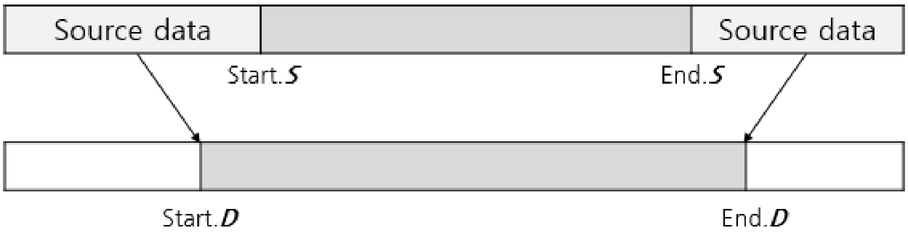
- The equations for the data scaling is as below:

1. Destination (*D*)= Source (*S*) x Gradient + Offset
2. Gradient = (End.*D*-Start.*D*) ÷ (End.*S*-Start.*S*)
3. Offset (Y intercept) = End.*D*-Gradient x End.*S*

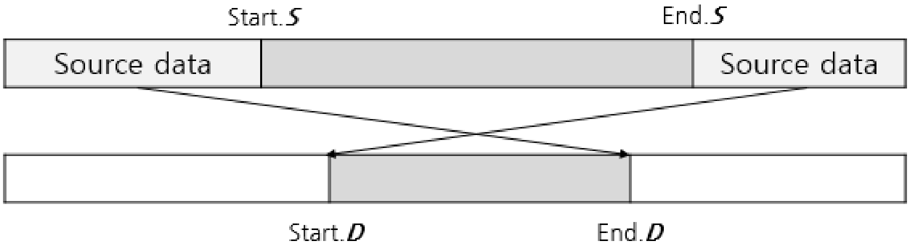
- The graph with the equations above will be drawn as below:



- When the data assigned to *S* exceeds the range between Start.*S* and End.*S*, the data is converted into the value assigned to Start.*D* (When value assigned to *S* is smaller than Start.*S*) or End.*D* (When value assigned to *S* is greater than End.*S*).



- When gradient is negative, the data assigned to *S* is converted into the value assigned to End.*D* (When value assigned to *S* is smaller than Start.*S*) or Start.*D* (When value assigned to *S* is greater than End.*S*)



- The value assigned to *n* must not exceed the range of the corresponding device. If assigned value exceeds the range of device area assigned to *S* and *D*, it may cause PLC CPU STOP.

DSCL, DSCLP

- DSCL(P) instructions scale data of *n* words starting from word device *S*. The scaling is operated according to the linear function specified with coordinated assigned to data table *P*. Scaled data are stored in *n* double word devices starting from device *D*.

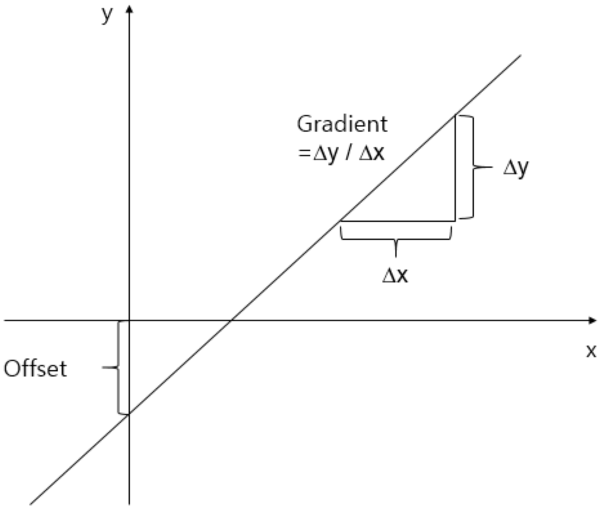
- Data table starting from double word device *P* is comprised as below:

P, *P*+1	Range start point of data to be scaled (Start.*S*) ······································ x axis
P+2, *P*+3	Range end point of data to be scaled (End.*S*) ·· x axis
P+4, *P*+5	Range start point of scaled data (Start.*D*) ·· y axis
P+6, *P*+7	Range end point of scaled data (End.*D*) ·· y axis

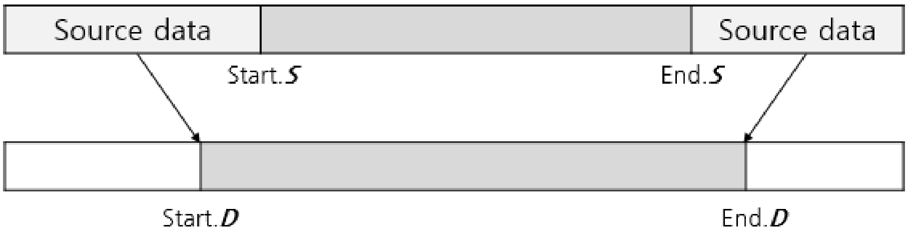
- The equations for the data scaling is as below:

1. Destination (*D*)= Source (*S*) x Gradient + Offset
2. Gradient = (End.*D*-Start.*D*) ÷ (End.*S*-Start.*S*)
3. Offset (Y intercept) = End.*D*-Gradient x End.*S*

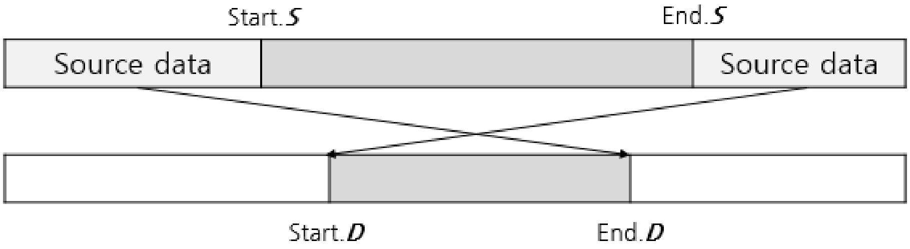
- The graph with the equations above will be drawn as below:



- When the data assigned to *S* exceeds the range between Start.*S* and End.*S*, the data is converted into the value assigned to Start.*D* (When value assigned to *S* is smaller than Start.*S*) or End.*D* (When value assigned to *S* is greater than End.*S*).

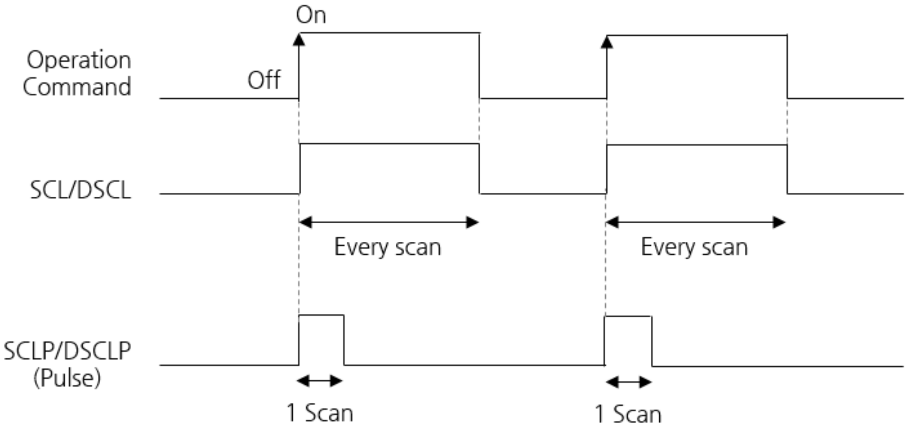


- When gradient is negative, the data assigned to *S* is converted into the value assigned to End.*D* (When value assigned to *S* is smaller than Start.*S*) or Start.*D* (When value assigned to *S* is greater than End.*S*)



- The value assigned to *n* must not exceed the range of the corresponding device. If assigned value exceeds the range of device area assigned to *S* and *D*, it may cause PLC CPU STOP.

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)

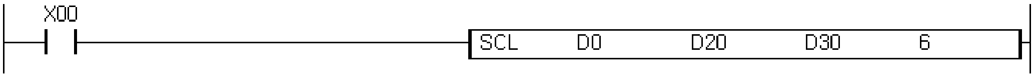
F110 turns ON for 1 scan when (End.*D* - Start.*D*) is divided by 0. (When Start.*S* = End.*S*)

Program Example

SCL

When X00 turns ON, SCL instruction is executed. 6 word data blocks starting from D0 will be scaled. According to the coordinates assigned to D30, a function for scaling is defined. Then, the converted data will be stored in 6 word devices starting from D20.

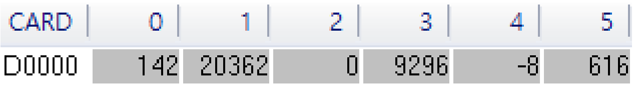
Ladder Diagram (LD)



Instruction List (IL)

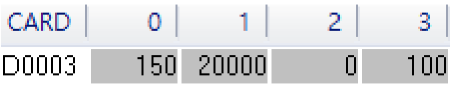
Instruction	Device			
LD	X00			
SCL	D0	D20	D30	6

- Data to be scaled are assigned to D0, D1, D2, D3, D4 and D5 as below.

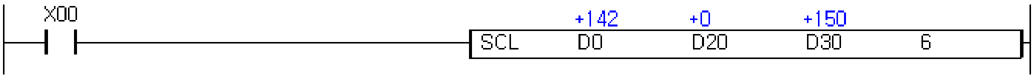


CARD	0	1	2	3	4	5
D0000	142	20362	0	9296	-8	616

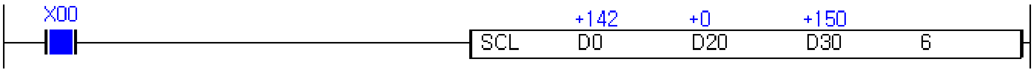
- A linear function is defined by the data stored in D30, D31, D32 and D33. The range of source data is from 150 to 20000. The range of destination data is from 0 to 100.

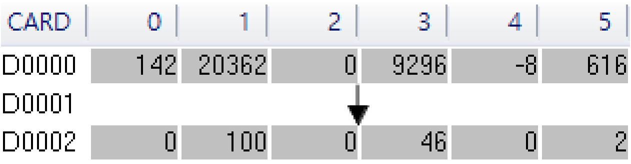


CARD	0	1	2	3
D0003	150	20000	0	100



- When X00 turns ON, SCL instruction is executed. Then, scaled data are stored in D20, D21, D22, D23, D24 and D25.
- Since the data stored in D0, D1, D2 and D4 exceeds the source data, they are converted into the minimum or maximum value of the destination data.





CARD	0	1	2	3	4	5
D0000	142	20362	0	9296	-8	616
D0001						
D0002	0	100	0	46	0	2

2.4.10 ESCL, ESCLP

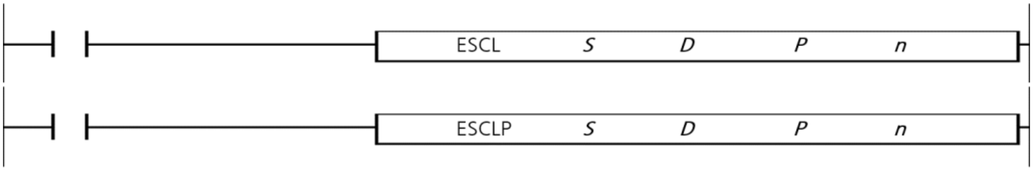
Supported PLC Series

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

Function

ESCL(P) instructions execute scaling of 32-bit floating-point data according to the specified linear function.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ESCL(P)	*S*	O	O	O	O	O	-	-	-	-	O	O	O	O	O	O	5	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	O	O	O	O	O				
	P	O	-	O	O	O	-	-	-	-	O	O	O	O	O	O				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S	Head address of double word devices where data to be scaled is stored. (32-bit floating-point number)
D	Head address of double word devices where result of data scaling will be stored. (32-bit floating-point number)
P	Head address of data table where coordinate data to define linear function are stored. (32-bit floating-point number)
n	Number of data to be scaled.

ESCL, ESCLP

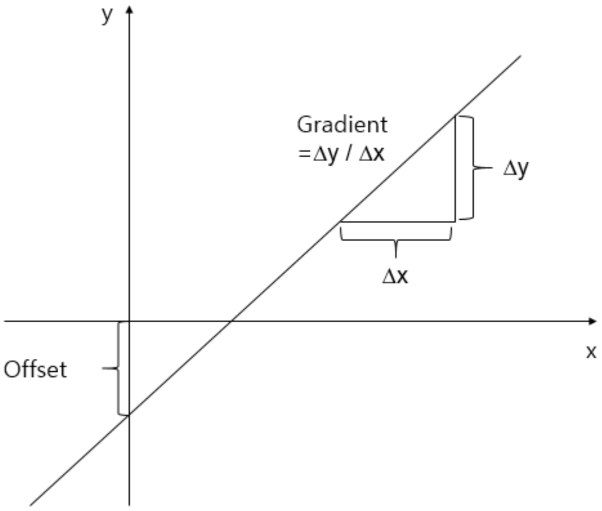
- ESCL(P) instructions execute scalling of *n* 32-bit floating-point number data starting from double word device *S*. The scaling is operated according to the linear function defined with data assigned to data table *P*. Scaled data are stored in *n* double word devices starting from device *D*.
- Data table starting from double word device *P* is comprised as below:

P, *P*+1	Range start point of data to be scaled (Start.*S*) ······································ x axis
P+2, *P*+3	Range end point of data to be scaled (End.*S*) ·· x axis
P+4, *P*+5	Range start point of scaled data (Start.*D*) ·· y axis
P+6, *P*+7	Range end point of scaled data (End.*D*) ·· y axis

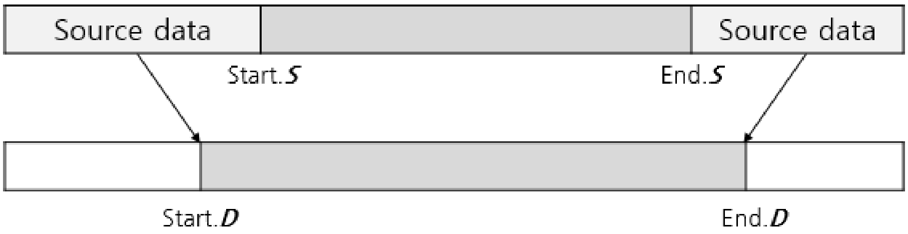
- The equations for the data scaling is as below:

1. Destination (*D*)= Source (*S*) x Gradient + Offset
2. Gradient = (End.*D*-Start.*D*) ÷ (End.*S*-Start.*S*)
3. Offset (Y intercept) = End.*D*-Gradient x End.*S*

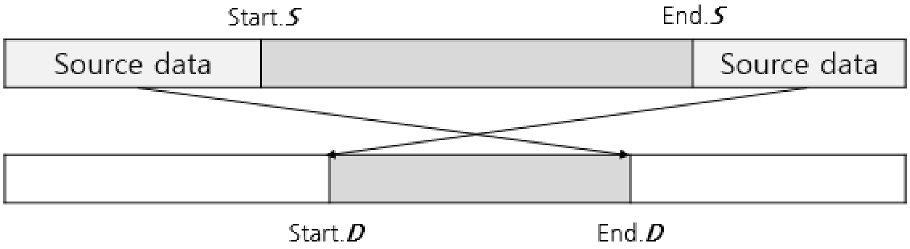
- The graph with the equations above will be drawn as below:



- When the data assigned to *S* exceed the range between Start.*S* and End.*S*, the data is converted into the value assigned to Start.*D* (When value assigned to *S* is smaller than Start.*S*) or End.*D* (When value assigned to *S* is greater than End.*S*).



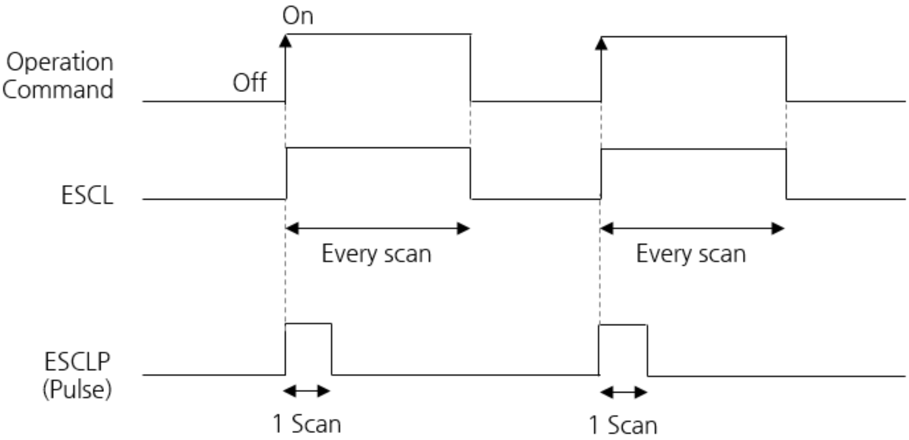
- When gradient is negative, the data assigned to *S* is converted into the value assigned to End.*D* (When value assigned to *S* is smaller than Start.*S*) or Start.*D* (When value assigned to *S* is greater than End.*S*)



- The value assigned to *n* must not exceed the range of the corresponding device. If assigned value exceeds the range of device area assigned to *S* and *D*, it may cause PLC CPU STOP.
- When the range of data to be scaled does not exist (When Start.*S*=End.*S*), "1.#QNAN[3]0" is stored in double word devices starting from *D*.

[3] QNAN: Stands for 'Quiet Not-a-Number'.

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)

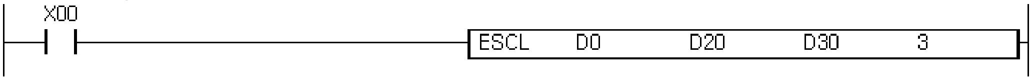
F110 turns ON for 1 scan when (End.*D* - Start.*D*) is divided by 0. (When Start.*S* = End.*S*)

Program Example

ESCL

When X00 turns ON, ESCL instruction is executed. 3 double word data blocks starting from D0 (D0~D5) will be scaled. According to the coordinates assigned to D30, a function for scaling is defined. Then, the converted data will be stored in 3 double word devices starting from D20 (D20~D25).

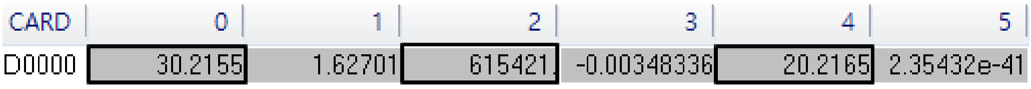
Ladder Diagram (LD)



Instruction List (IL)

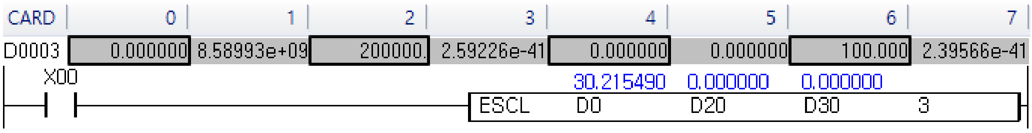
Instruction	Device			
LD	X00			
ESCL	D0	D20	D30	3

- 32-bit floating-point numbers to be scaled are assigned to D0, D2 and D4 as below.



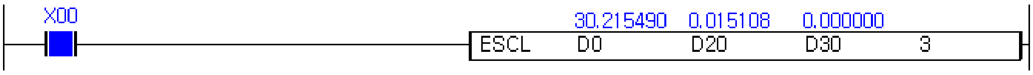
CARD	0	1	2	3	4	5
D0000	30.2155	1.62701	615421	-0.00348336	20.2165	2.35432e-41

- A linear function is defined by the data stored in D30, D32, D34 and D36. The range of source data is from 0 to 200000. The range of destination data is from 0 to 100.

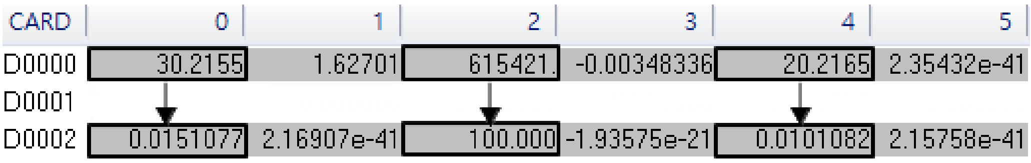


CARD	0	1	2	3	4	5	6	7
D0003	0.000000	8.58993e+09	200000.	2.59226e-41	0.000000	0.000000	100.000	2.39566e-41

- When X00 turns ON, ESCL instruction is executed. Then, scaled data are stored in D20, D22 and D24.
- Since the data stored in D2 exceeds the range of source data, it is converted into the maximum value of the destination data.



```
X00                        30.215490  0.015108  0.000000
─┤■├──────────────────[ESCL   D0        D20       D30      3 ]
```



CARD	0	1	2	3	4	5
D0000	30.2155	1.62701	615421.	-0.00348336	20.2165	2.35432e-41
D0001						
D0002	0.0151077	2.16907e-41	100.000	-1.93575e-21	0.0101082	2.15758e-41

2.4.11 SER, SERP, DSER, DSERP

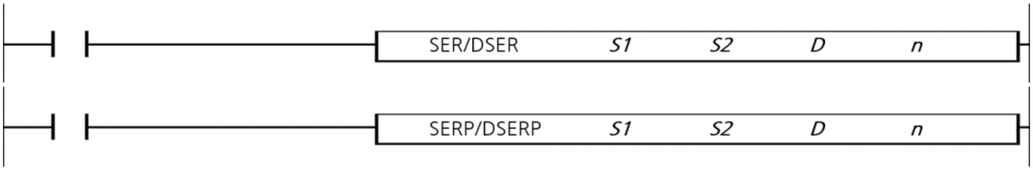
Supported PLC Series

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

SER(P) instructions search for a word data within a range of words.
DSER(P) instructions search for a double word data within a range of double words.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SER(P) DSER(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	5	O	-	-
	S2	O	O	O	-	O	-	O	O	-	O	-	O	O	O	-				
	D	O	-	O	O	O	-	-	-	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



S1	Address of word/double word device where data to find is stored.
S2	Head address of word/double word device where to search the data assigned to *S1*.
D	Head address of word device where the result of data search will be stored.
n	Number of data blocks to search. (*n* ≧1)

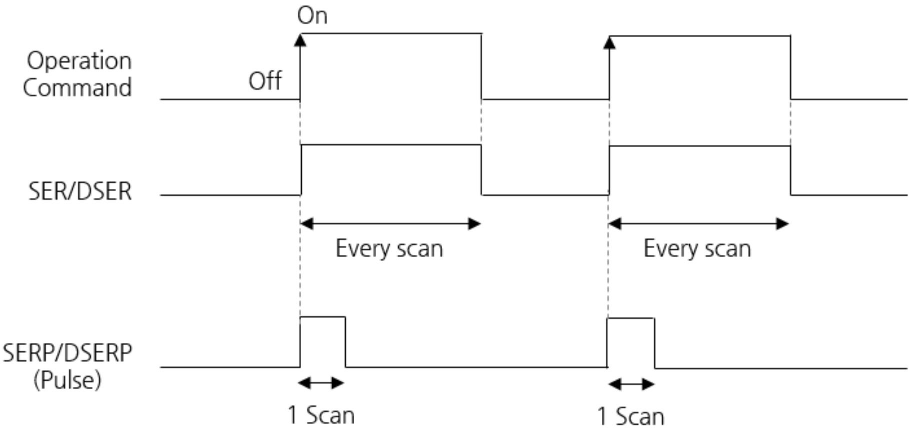
SER, SERP

- SER(P) instructions find word data assigned to *S1* among *n* devices starting from word device *S2*. The result of data search is stored in word devices starting from *D*.
- Only positive numbers can be assigned to *n*.
- After the execution of instruction, the location of first matched word data is stored in *D*. The number of matched data is stored in *D*+1.
- If there is no matched data, “0” is stored in *D* and *D*+1.

DSER, DSERP

- DSER(P) instructions find double word data assigned to *S1* among *n* devices starting from double word device *S2*. The result of data search is stored in double word devices starting from *D*.
- Only positive numbers can be assigned to *n*.
- After the execution of instruction, the location of first matched word data is stored in *D*. The number of matched data is stored in *D*+1.
- If there is no matched data, “0” is stored in *D* and *D*+1.

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)

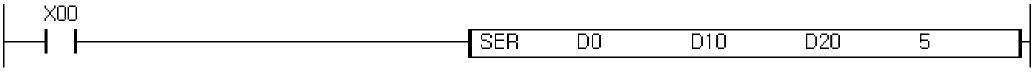
F110 turns ON for1 scan when the value assigned to *n* is “0” or negative.

Program Example

SER

When X00 turns ON, SER instruction is executed. Among 5 word devices starting from D10 (D10~D14), data assigned to D0 will be searched. The location of first matched data will be stored in D20. The number of matched data will be stored in D21.

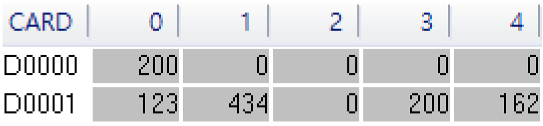
Ladder Diagram (LD)



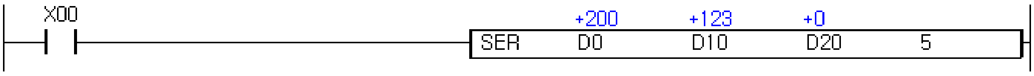
Instruction List (IL)

Instruction	Device			
LD	X00			
SER	D0	D10	D20	5

- The instruction will search for “200 (value assigned to D0)” among the data stored in 5 word devices starting from D10 (D10~D14).

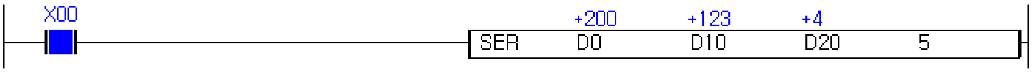


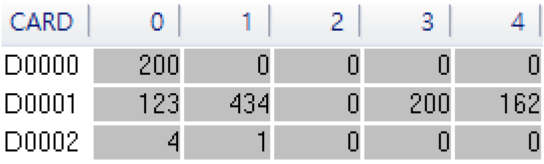
CARD	0	1	2	3	4
D0000	200	0	0	0	0
D0001	123	434	0	200	162



- When X00 turns ON, the intruction is executed. Then, the result is stored in D20 and D21.

- The location of first matched data “4” is stored in D20. The number of matched data “1” is stored in D21.





CARD	0	1	2	3	4
D0000	200	0	0	0	0
D0001	123	434	0	200	162
D0002	4	1	0	0	0

2.4.12 BSER, BSERP

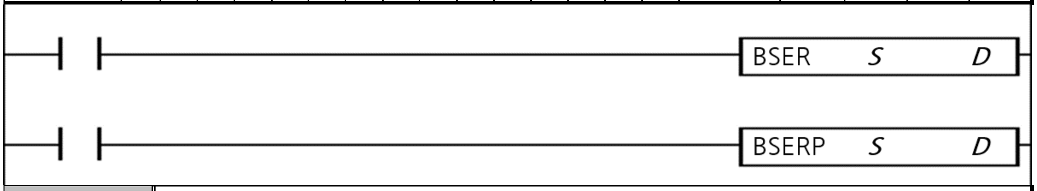
Supported PLC Series

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

BSER(P) instructions search for the most significant bit and the least significant bit which are set in a word device.

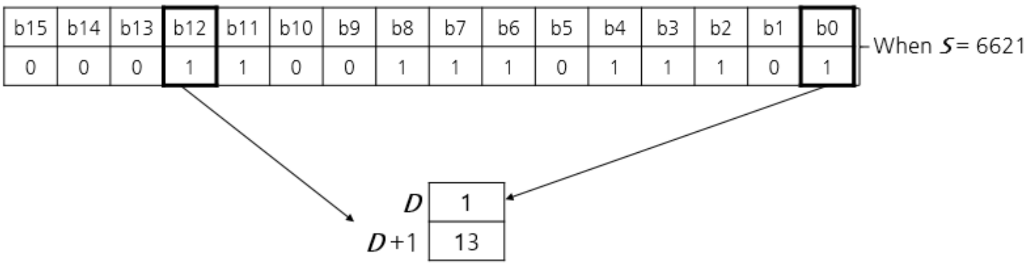
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BSER(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Address of word device to search the location of the most/least significant bit which is set.
D	Head address of word device to store the location of searched data

BSER, BSERP

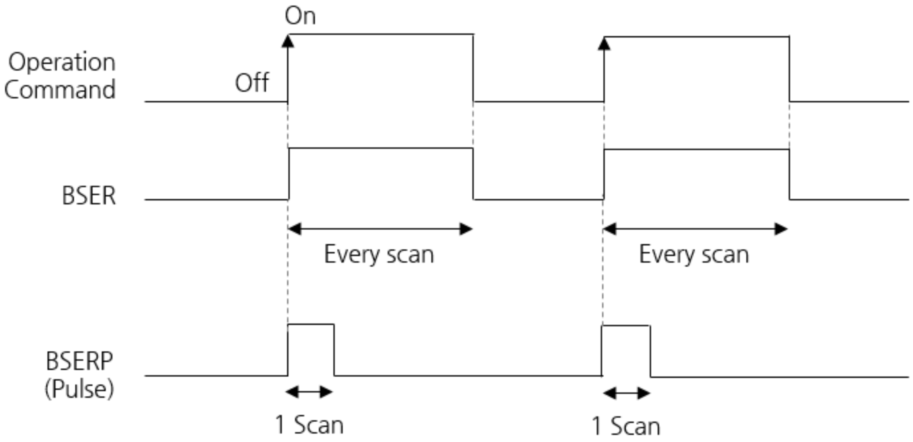
- BSER(P) instructions search for the most significant bit and the least significant bit which are set in a word device assigned to *S*. The result of bit search is stored in word device *D* and *D*+1.
- After the execution of instruction, location of the least significant bit which is set is stored in word device *D*. Location of the most significant bit which is set is stored in word device *D*+1.



b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	1	1	0	0	1	1	1	0	1	1	1	0	1

- If there is no matched data, “0” is stored in *D* and *D*+1.
- If there is only one bit which is set, same value is stored in both word device *D* and *D*+1.

Execution Condition



Operation Error

Error flag (F110)

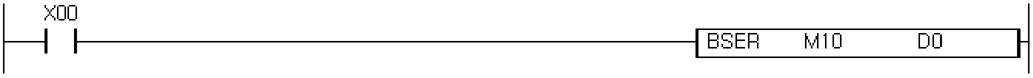
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BSER

When X00 turns ON, BSER instruction is executed. According to the value assigned to M10, the instruction finds the most significant and the least significant bits which are set. Location of the least significant bit which is set is stored in D0. Location of the most significant bit which is set will be stored in D1.

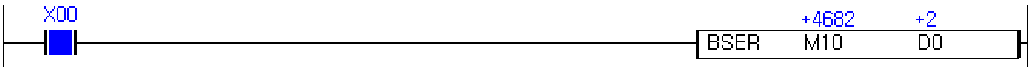
Ladder Diagram (LD)

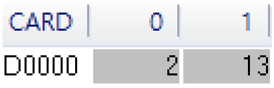


Instruction List (IL)

Instruction	Device	
LD	X00	
BSER	M10	D0

- Value assigned to M10 is “4682” (BIN=0001 0010 0100 1010). The instruction will search for location of the most significant bit and the least significant bit which are set.
- When X00 turns ON, the intruction is executed. Then, the result is stored in D0 and D1.
- Location of the least significant bit which is set, “2” is stored in D0. Location of the most significant bit which is set, “13” is stored in D1.





CARD	0	1
D0000	2	13

2.5 String Process Instructions

2.5.1 BINDA, BINDAP, DBINDA, DBINDAP

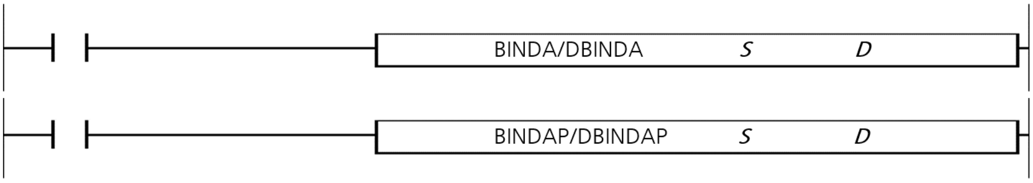
Supported PLC Series

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

BINDA(P) instructions convert 16-bit BIN data in decimal notation, stored in word device *S*, into ASCII code. Then, the conversion result is stored in 3 word devices starting from device *D*. DBINDA(P) instructions convert 32-bit BIN data in decimal notation, stored in double word device *S*, into ASCII code. Then, the conversion result is stored in 6 word devices starting from device *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
BINDA(P) DBINDA(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Constant or head address of word/double word device where 16/32-bit BIN data to be converted is stored.
D	Head address of 3/6 word devices where conversion result (ASCII) will be stored.

BINDA, BINDAP

- BINDA(P) instructions convert each digit of 16-bit BIN data in decimal notation, which is stored in the word device *S*, into ASCII code.
- Converted data is stored in 3 word devices starting from device *D*.
- The range of data stored in word device *S* is as following.
  -32768 ≦ assigned value in word device *S* ≦ 32767
- If the data stored in device *S* is positive, the sign “H20” (= ‘space’ in ASCII code) will be stored in the lower byte of device *D*.
- If the data stored in device *S* is negative, the sign “H2D” (= ‘ - ’ in ASCII code) will be stored in the lower byte of device *D*.
- Ones digit of the conversion result is stored in the upper byte of device *D*+2.
- “H20” will be stored in the destination devices for the leading zeros of valid digits.

(*) Refer to Appendix n for ASCII code table.

- The conversion result is stored as below.

S

16-bit BIN data ⇨

ASCII code

	b15 ··· b8	b7 ··· b0
D	Ten thousands digit (x10^4)	Sign
D+1	Hundreds digit (x10^2)	Thousands digit (x10^3)
D+2	Ones digit (x10^0)	Tens digit (x10^1)

DBINDA, DBINDAP

- DBINDA(P) instructions convert each digit of 32-bit BIN data in decimal notation, which is stored in the double word device *S*, into ASCII code.
- Converted data is stored in 6 word devices starting from the device *D*.
- The range of data stored in word device *S* is as following.
  -2147483648 ≦ assigned value in word device *S* ≦ 2147483647
- If the data stored in device *S* is positive, the sign "H20" (= 'space' in ASCII code) will be stored in the lower byte of device *D*.
- If the data stored in device *S* is negative, the sign "H2D" (= ' - ' in ASCII code) will be stored in the lower byte of device *D*.
- Ones digit of the conversion result is stored in the lower byte of device *D*+5. In the upper byte of word device *D*+5, H00 is stored.
- "H20" will be stored in the destination devices for the leading zeros of valid digits.

(*) Refer to Appendix n for ASCII code table.

- The conversion result is stored as below.

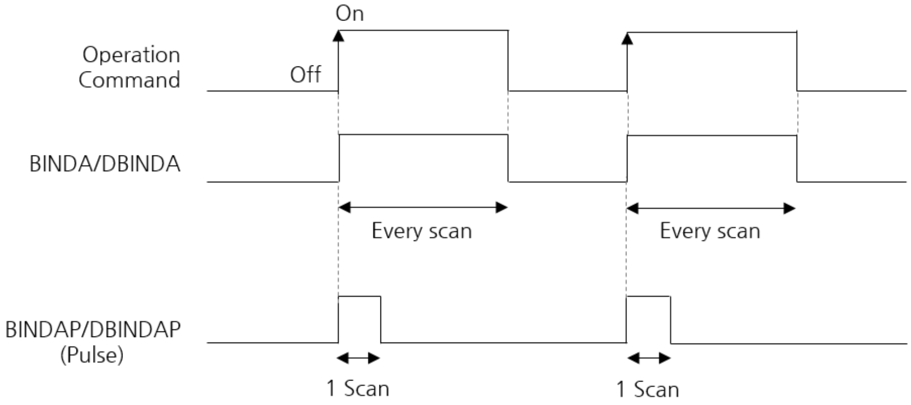
S

32-bit BIN data ⇨

ASCII code

	b15 ··· b8	b7 ··· b0
D	Billions digit (x10^9)	Sign
D+1	Ten millions digit (x10^7)	Hundred millions digit (x10^8)
D+2	Hundred thousands digit (x10^5)	Millions digit (x10^6)
D+3	Thousands digit (x10^3)	Ten thousands digit (x10^4)
D+4	Tens digit (x10^1)	Hundreds digit (x10^2)
D+5	H00	Ones digit (x10^0)

Execution Condition



Operation Error

Error flag (F110)

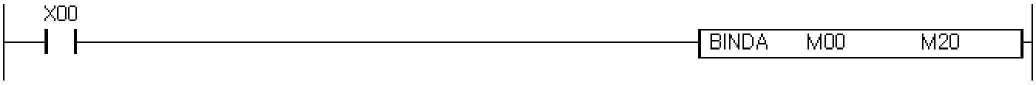
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BINDA

When operation command X00 is ON, BINDA instruction is executed. Then, 16-bit BIN data stored in word device M00 (M00 ~ M0F) is converted into ASCII data. The conversion result is stored in 3 word devices starting from word device M20 (M20 ~ M4F).

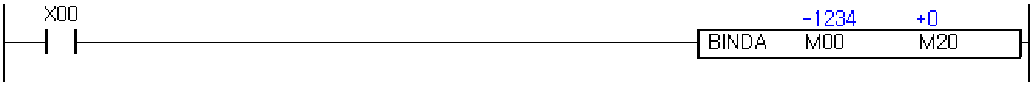
Ladder Diagram (LD)



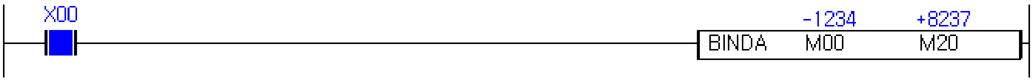
Instruction List (IL)

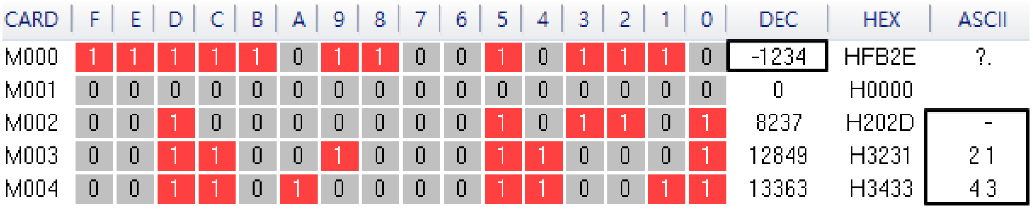
Instruction	Device	
LD	X00	
BINDA	M00	M20

- 16-bit BIN data “-1234” is assigned to M00.



- When X00 turns ON, BINDA instruction is executed. Then, 16-bit BIN data “-1234” is converted into ASCII data “-1234”. The conversion result is stored in 3 word devices starting from M20 (M20 ~ M4F). Since the ten thousands digit is “0”, “H20” is filled at the upper byte of M20 (M28~M2F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX	ASCII
M000	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	-1234	HFB2E	?.
M001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	H0000	
M002	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	8237	H202D	-
M003	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	12849	H3231	2 1
M004	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	13363	H3433	4 3

2.5.2 DABIN, DABINP, DDABIN, DDABINP

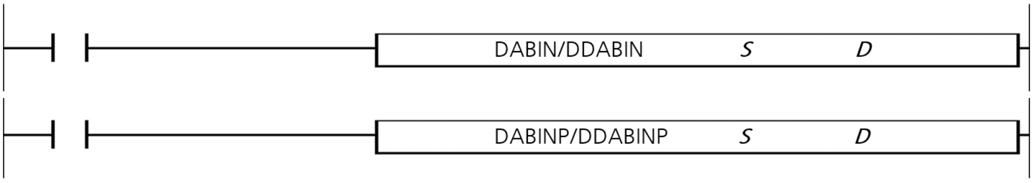
Supported PLC Series

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

DABIN(P) instructions convert ASCII code stored in 3 word devices starting from device *S* into 16-bit INT data. Then, the conversion result is stored in word device *D*.
DDABIN(P) instructions convert ASCII code stored in 6 word devices starting from device *S* into 32-bit INT data. Then, the conversion result is stored in double word device *D*.

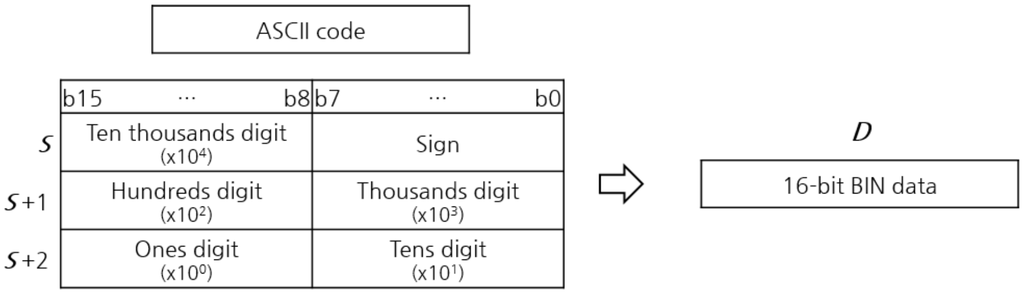
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
DABIN(P) DDABIN(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Head address of 3/6 word devices where ASCII code to be converted is stored.
D	Head address of word/double word device where conversion result (INT) will be stored.

DABIN, DABINP

- DABIN(P) instructions convert decimal number in ASCII code, stored in 3 word devices starting from the device *S*, into 16-bit INT data.
- Converted data is stored in the word device *D*.
- The range of decimal number in ASCII code stored in *S*, *S*+1, *S*+2 is as following.
-32768 ≦ decimal number in ASCII code stored in *S*, *S*+1, *S*+2 ≦ 32767
- When the data to be converted exceeds the range of -32768~32767, it overflows.
- To execute DABIN(P) instructions properly, input ASCII code in following format.



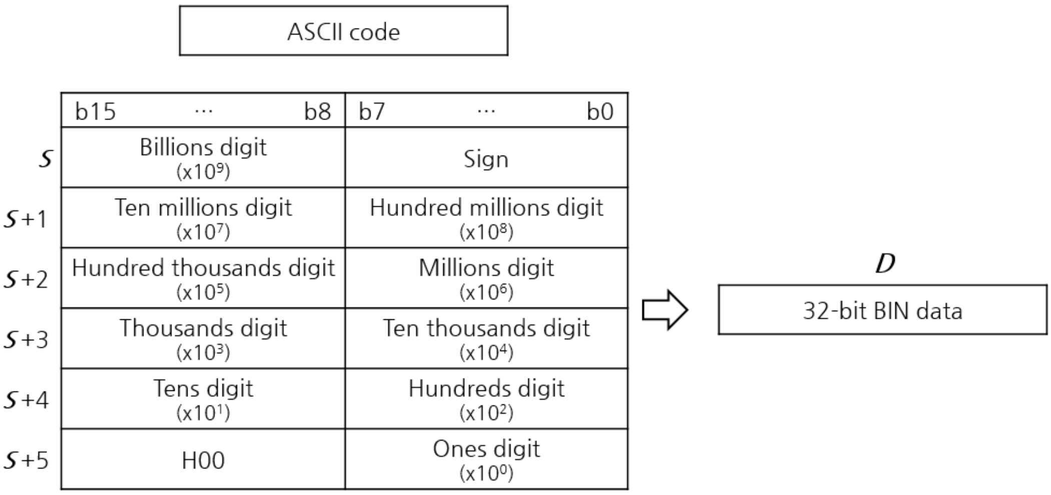
ASCII code

	b15 ··· b8	b7 ··· b0
S	Ten thousands digit (x10^4)	Sign
S+1	Hundreds digit (x10^2)	Thousands digit (x10^3)
S+2	Ones digit (x10^0)	Tens digit (x10^1)

- Each digit has to be written in range of “H30” ~ “H39”. If not, they are processed as “0”.
- At lower byte of device *S*, input “H20” (‘Space’ in ASCII code) for positive number or “H2D” (‘ - ’ in ASCII code) for negative number. “H20” and “H2D” can be omitted. In this case, the ASCII data will be converted into positive number.

DDABIN, DDABINP

- DDABIN(P) instructions convert decimal number in ASCII code, stored in 6 word devices starting from the device *S*, into 32-bit INT data.
- Converted data is stored in the double word device *D*.
- The range of decimal number in ASCII code stored in *S* ~ *S*+5 is as following.
  -2147483648 ≦ decimal number in ASCII code stored in *S* ~ *S*+5 ≦ 2147483647
- When the data to be converted exceeds the range of -2147483648~2147483647, it overflows.
- To execute DDABIN(P) instructions properly, input ASCII code in following format.



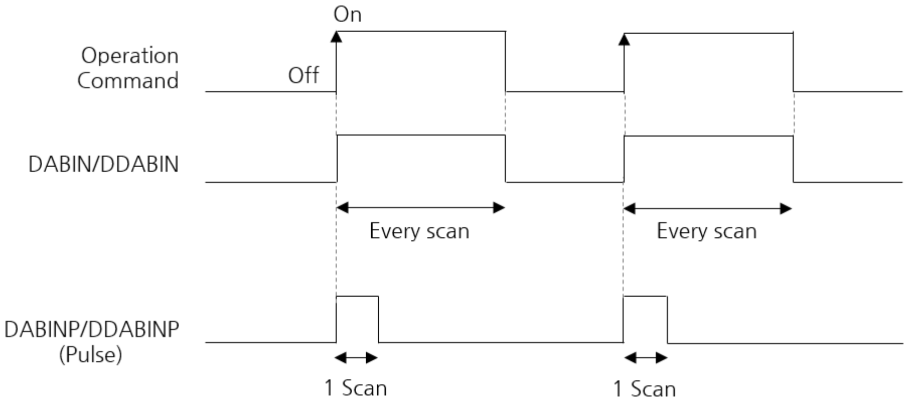
ASCII code

	b15 ··· b8	b7 ··· b0
S	Billions digit $(x10^{9})$	Sign
S+1	Ten millions digit $(x10^{7})$	Hundred millions digit $(x10^{8})$
S+2	Hundred thousands digit $(x10^{5})$	Millions digit $(x10^{6})$
S+3	Thousands digit $(x10^{3})$	Ten thousands digit $(x10^{4})$
S+4	Tens digit $(x10^{1})$	Hundreds digit $(x10^{2})$
S+5	H00	Ones digit $(x10^{0})$

⇨ *D* 32-bit BIN data

- Each digit has to be written in range of "H30" ~ "H39". If not, they are processed as "0".
- At lower byte of device *S*, input "H20" ('Space' in ASCII code) for positive number or "H2D" (' - ' in ASCII code) for negative number. "H20" and "H2D" can be omitted. In this case, the ASCII code will be converted into positive number.

Execution Condition



Operation Error

Error flag (F110)

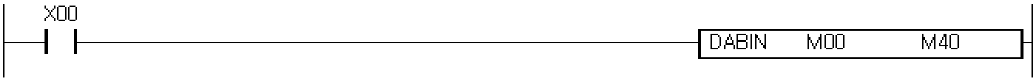
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DABIN

When operation command X00 is ON, DABIN instruction is executed. Then, decimal in ASCII code stored in 3 word devices starting from word device M00 (M00 ~ M2F) is converted into 16-bit BIN data. The conversion result is stored in word device M40 (M40 ~ M4F).

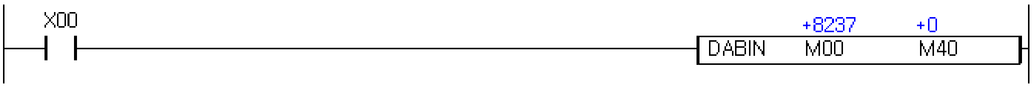
Ladder Diagram (LD)



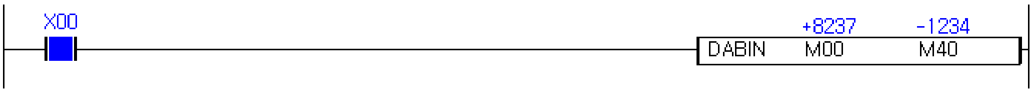
Instruction List (IL)

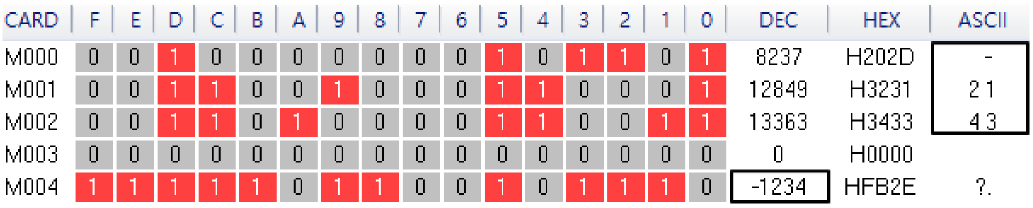
Instruction	Device	
LD	X00	
DABIN	M00	M40

- ASCII data “-1234” is assigned to M00.



- When X00 turns ON, DABIN instruction is executed. Then, ASCII data “-1234” is converted into 16-bit INT data “-1234”. The conversion result is stored in M40 (M40 ~ M4F). The ten thousands digit is written as “H20” but it is processed as “H30”, which is equal to “0” in ASCII code.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX	ASCII
M000	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	8237	H202D	-
M001	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	12849	H3231	2 1
M002	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	13363	H3433	4 3
M003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	H0000	
M004	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	-1234	HFB2E	?.

2.5.3 BINHA, BINHAP, DBINHA, DBINHAP

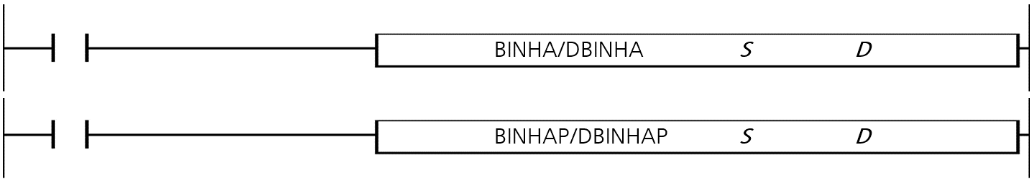
Supported PLC Series

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

BINHA(P) instructions convert 16-bit BIN data in hexadecimal notation, stored in word device *S*, into ASCII code. Then, the conversion result is stored in double word device *D*.
DBINHA(P) instructions convert 32-bit BIN data in hexadecimal notation, stored in double word device *S*, into ASCII code. Then, the conversion result is stored in 4 word devices starting from device *D*.

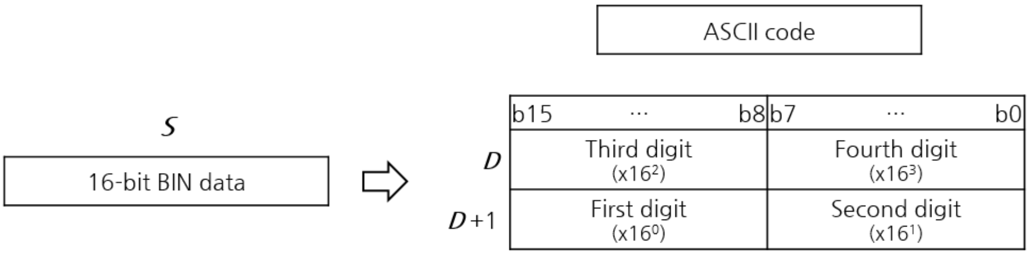
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BINHA(P) DBINHA(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Constant or head address of word/double word device where 16/32-bit BIN data to be converted is stored.
D	Head address of 2/4 word devices where conversion result (ASCII) will be stored.

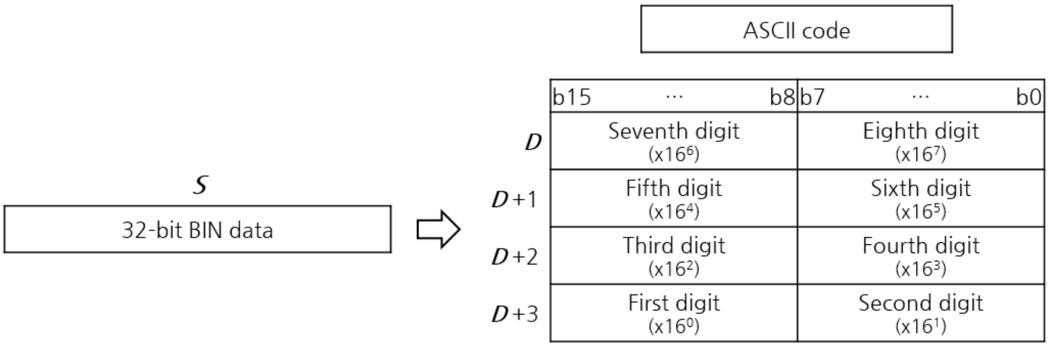
BINHA, BINHAP

- BINHA(P) instructions convert 16-bit BIN data in hexadecimal notation, which is stored in the word device *S*, into ASCII code.
- Converted data is stored in 2 word devices starting from device *D*.
- The range of 16-bit BIN data stored in word device *S* is as following.
  0 ≦ assigned value in word device *S* ≦ HFFFF
- The data stored in device *D* and *D*+1 is 4-digit hexadecimal in ASCII code.
- The first digit of the conversion result is stored in the upper byte of device *D*+1.
- Converted data is stored as below.



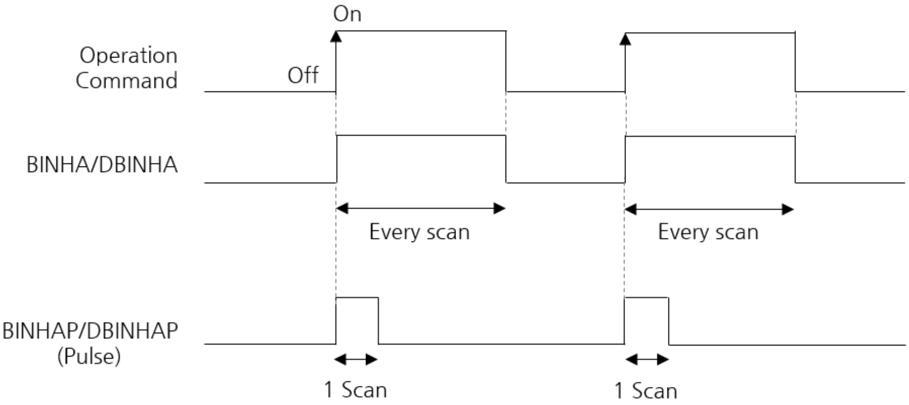
DBINHA, DBINHAP

- DBINHA(P) instructions convert 32-bit BIN data in hexadecimal notation, which is stored in the double word device *S*, into ASCII code.
- Converted data is stored in 4 word devices starting from device *D*.
- The range of data stored in double word device *S* is as following.
  0 ≦ assigned value in double word device *S* ≦ HFFFFFFFF
- The first digit of the conversion result is stored in the upper byte of device *D*+3.
- The data stored in device *D*, *D*+1, *D*+2 and *D*+3 is 8-digit hexadecimal in ASCII code.
- Converted data is stored as below.



	b15 ··· b8	b7 ··· b0
D	Seventh digit (x16^6)	Eighth digit (x16^7)
D+1	Fifth digit (x16^4)	Sixth digit (x16^5)
D+2	Third digit (x16^2)	Fourth digit (x16^3)
D+3	First digit (x16^0)	Second digit (x16^1)

Execution Condition



Operation Error

Error flag (F110)

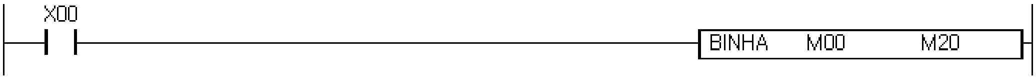
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BINHA

When operation command X00 is ON, BINHA instruction is executed. Then, 16-bit BIN data in hexadecimal notation, which is stored in word device M00 (M00 ~ M0F), is converted into ASCII code. The conversion result is stored in double word device M20 (M20 ~ M3F).

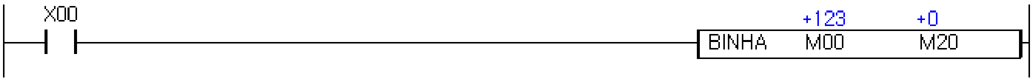
Ladder Diagram (LD)



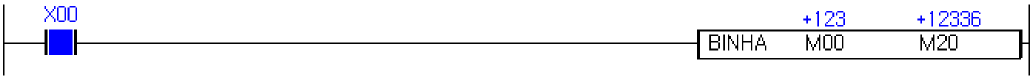
Instruction List (IL)

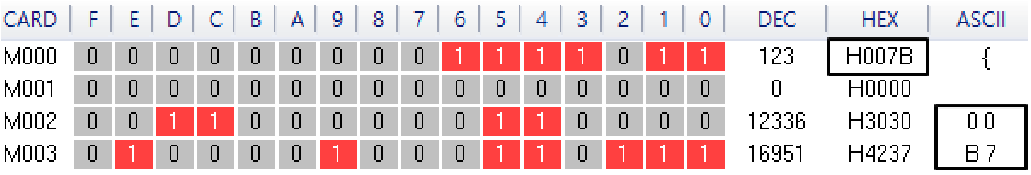
Instruction	Device	
LD	X00	
BINHA	M00	M20

- 16-bit BIN data “123” is assigned to M00.



- When X00 turns ON, BINHA instruction is executed. Then, 16-bit BIN data “123” in hexadecimal notation (= H007B) is converted into ASCII code “007B”. The conversion result is stored in M20 (M20 ~ M3F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX	ASCII
M000	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	123	H007B	{
M001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	H0000	
M002	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	12336	H3030	0 0
M003	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	16951	H4237	B 7

2.5.4 HABIN, HABINP, DHABIN, DHABINP

Supported PLC Series

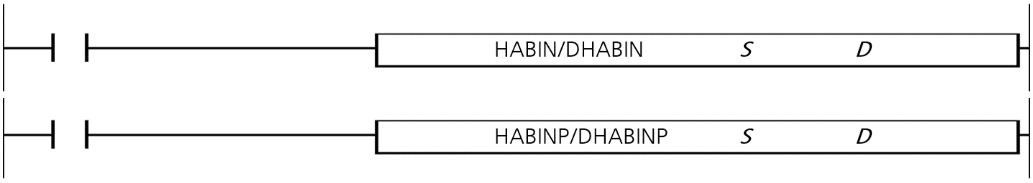
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

HABIN(P) instructions convert ASCII data stored in 2 word devices starting from device *S* into 16-bit BIN data in hexadecimal notation. Then, the conversion result is stored in word device *D*.

DHABIN(P) instructions convert ASCII data stored in 4 word devices starting from device *S* into 32-bit BIN data in hexadecimal notation. Then, the conversion result is stored in double word device *D*.

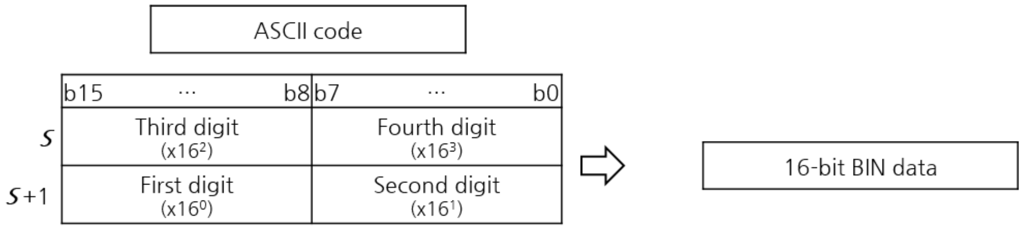
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
HABIN(P) DHABIN(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Head address of 2/4 word devices where ASCII code to be converted is stored.
D	Head address of word/double word device where conversion result (BIN) will be stored.

HABIN, HABINP

- HABIN(P) instructions convert hexadecimal in ASCII code, stored in 2 word devices starting from device *S*, into 16-bit BIN data in hexadecimal notation.
- Converted data is stored in the word device *D*.
- The range of ASCII data stored in *S*, *S*+1 is as following.
  "0" ≦ ASCII data stored in *S*, *S*+1 ≦ "FFFF"
- To execute HABIN(P) instructions properly, input ASCII code in following format.



- Each digit has to be in range of "H30" ~ "H39" or "H41" ~ "H46". If not, they are processed as "0".

DHABIN, DHABINP

- DHABIN(P) instructions convert ASCII code, stored in 4 word devices starting from device *S*, into 32-bit BIN data in hexadecimal notation.
- Converted data is stored in the double word device *D*.
- The range of ASCII data stored in *S* ~ *S*+3 is as following.
  "0" ≦ ASCII code data in *S* ~ *S*+3 ≦ "FFFFFFFF"
- To execute DHABIN(P) instructions properly, input ASCII code in following format.

ASCII code

	b15 ··· b8	b7 ··· b0
S	Seventh digit $(x16^6)$	Eighth digit $(x16^7)$
S+1	Fifth digit $(x16^4)$	Sixth digit $(x16^5)$
S+2	Third digit $(x16^2)$	Fourth digit $(x16^3)$
S+3	First digit $(x16^0)$	Second digit $(x16^1)$

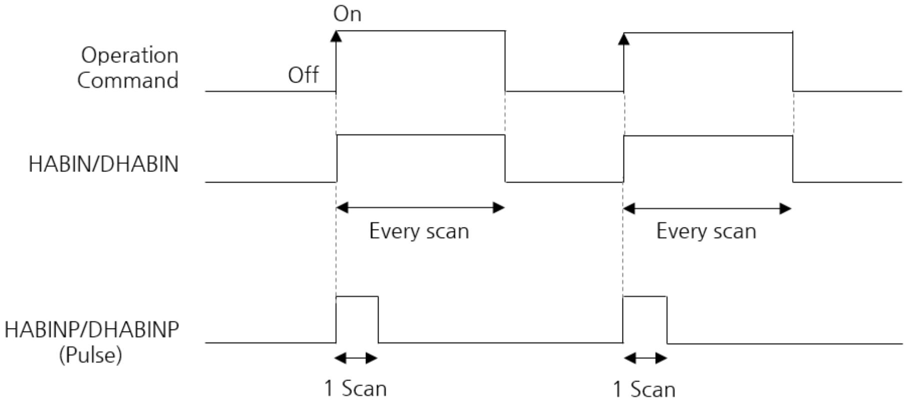
⇨

D, *D*+1

32-bit BIN data

- Each digit has to be in range of "H30" ~ "H39" or "H41" ~ "H46". If not, they are processed as "0".
- From first digit to fourth digit of hexadecimal are stored in device *D*.
- From fifth digit to eighth digit of hexadecimal are stored in device *D*+1.

Execution Condition



Operation Error

Error flag (F110)

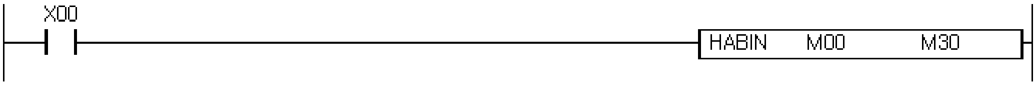
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

HABIN

When operation command X00 is ON, HABIN instruction is executed. Then, ASCII code stored in double word device M00 (M00 ~ M1F) is converted into 16-bit BIN data in hexadecimal notation. The conversion result is stored in word device M30 (M30 ~ M3F).

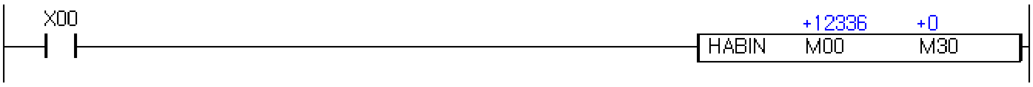
Ladder Diagram (LD)



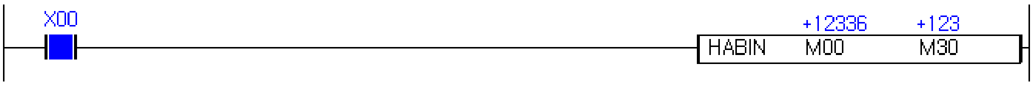
Instruction List (IL)

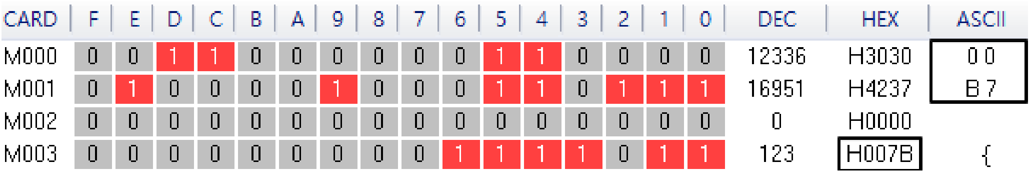
Instruction	Device	
LD	X00	
HABIN	M00	M30

- ASCII data “007B” is assigned to M00.



- When X00 turns ON, HABIN instruction is executed. Then, ASCII data “007B” is converted into 16-bit BIN data “H007B”. The conversion result is stored in M30 (M30 ~ M3F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX	ASCII
M000	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	12336	H3030	0 0
M001	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	16951	H4237	B 7
M002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	H0000	
M003	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	123	H007B	{

2.5.5 BCDDA, BCDDAP, DBCDDA, DBCDDAP

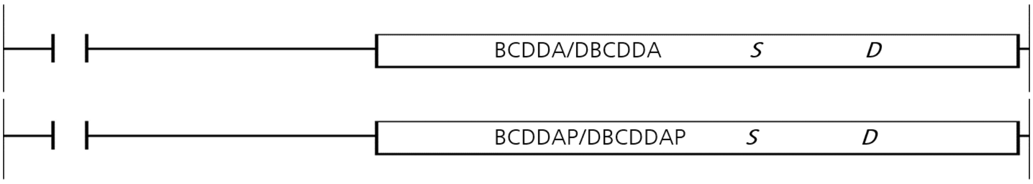
Supported PLC Series

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

BCDDA(P) instructions convert 4-digit BCD data stored in device *S* into ASCII code. Then, the conversion result is stored in 2 word devices starting from device *D*.
DBCDDA(P) instructions convert 8-digit BCD data stored in device *S* into ASCII code. Then, the conversion result is stored in 4 word devices starting from device *D*.

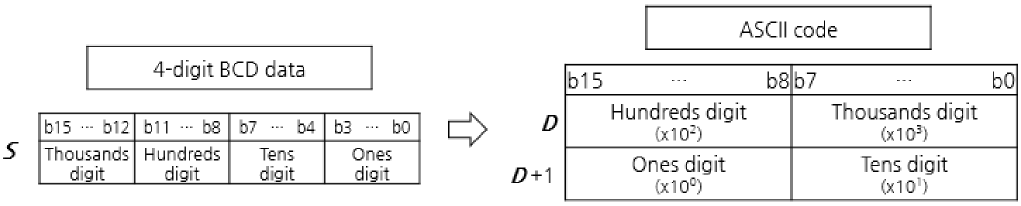
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BCDDA(P) DBCDDA(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



S	Constant or head address of word/double word device where 4/8-digit BCD data to be converted is stored.
D	Head address of 2/4 word device where conversion result (ASCII) will be stored.

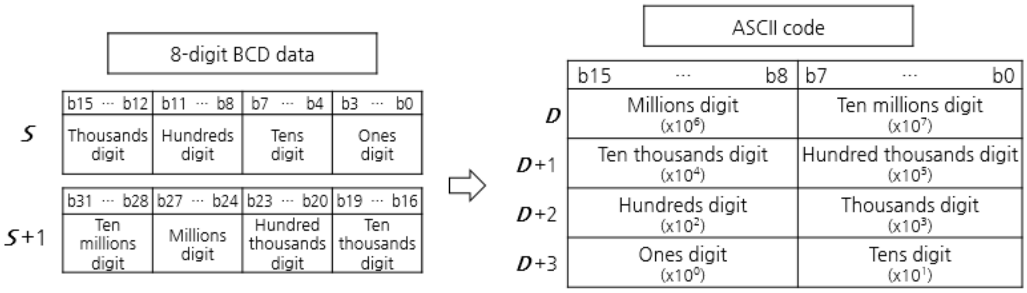
BCDDA, BCDDAP

- BCDDA(P) instructions convert each digit of 4-digit BCD data stored in the device *S* into ASCII code.
- Converted data is stored in device *D* and *D*+1.
- The range of BCD data stored in device *S* is as following.
  0 ≦ assigned value in device *S* ≦ 9999 (H9999)
- Ones digit of the conversion result is stored in the upper byte of device *D*+1.
- “H20” will be stored in the destination device for the leading zeros of valid digits.
- The conversion result is stored as below.



DBCDDA, DBCDDAP

- DBCDDA(P) instructions convert each digit of 8-digit BCD data stored in the device *S* into ASCII code.
- Converted data is stored in 4 word devices starting from device *D*.
- The range of BCD data stored in device *S* is as following.
  0 ≦ assigned value in device *S* ≦ 99999999 (H99999999)
- Ones digit of the conversion result is stored in the upper byte of device *D* +3.
- "H20" will be stored in the destination device for the leading zeros of valid digits.
- The conversion result is stored as below.



8-digit BCD data

	b15 ··· b12	b11 ··· b8	b7 ··· b4	b3 ··· b0
S	Thousands digit	Hundreds digit	Tens digit	Ones digit

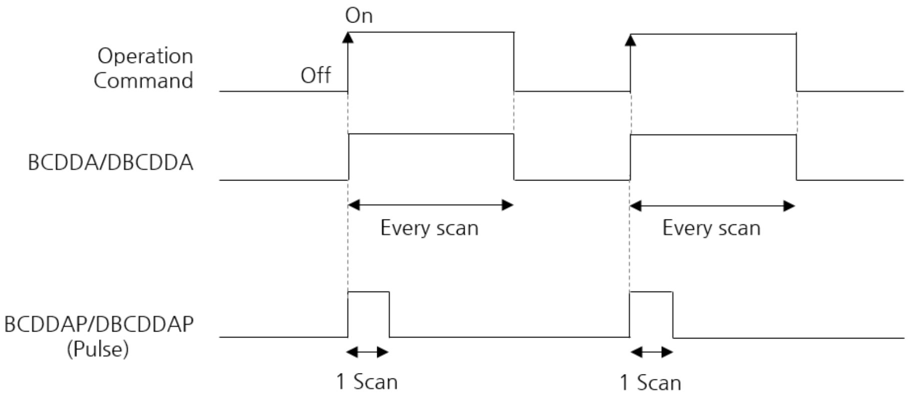
	b31 ··· b28	b27 ··· b24	b23 ··· b20	b19 ··· b16
S+1	Ten millions digit	Millions digit	Hundred thousands digit	Ten thousands digit

⇨

ASCII code

	b15 ··· b8	b7 ··· b0
D	Millions digit (x10^6)	Ten millions digit (x10^7)
D+1	Ten thousands digit (x10^4)	Hundred thousands digit (x10^5)
D+2	Hundreds digit (x10^2)	Thousands digit (x10^3)
D+3	Ones digit (x10^0)	Tens digit (x10^1)

Execution Condition



Operation Error

Error flag (F110)

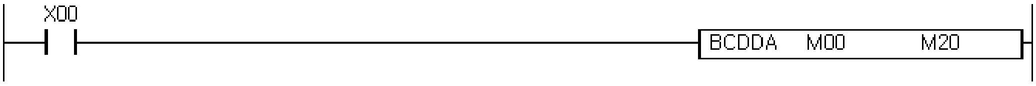
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BCDDA

When operation command X00 is ON, BCDDA instruction is executed. Then, 4-digit BCD data stored in word device M00 (M00 ~ M0F) is converted into ASCII code. The conversion result is stored in 2 word devices starting from word device M20 (M20 ~ M3F).

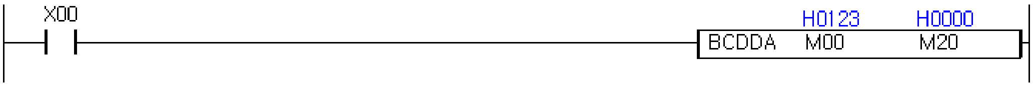
Ladder Diagram (LD)



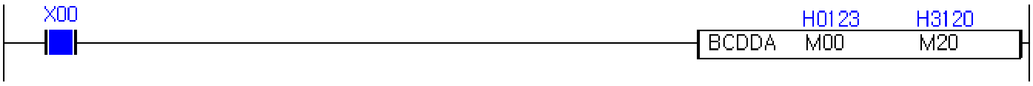
Instruction List (IL)

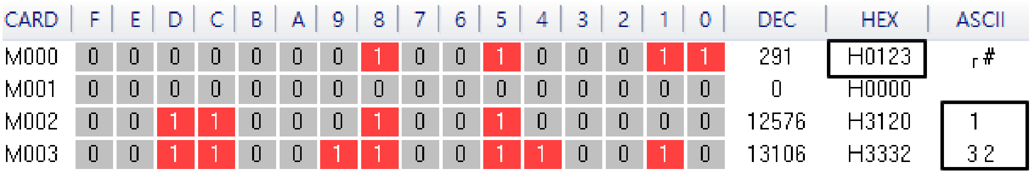
Instruction	Device	
LD	X00	
BCDDA	M00	M20

- 4-digit BCD data “123” (= H0123) is assigned to M00.



- When X00 turns ON, BCDDA instruction is executed. Then, 4-digit BCD data “123” (= H0123) is converted into ASCII code “123”. The conversion result is stored in M20 (M20 ~ M3F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	DEC	HEX	ASCII
M000	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	291	H0123	r#
M001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	H0000	
M002	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	12576	H3120	1
M003	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	13106	H3332	3 2

2.6 Bit Process Instructions

2.6.1 TEST, TESTP, DTEST, DTESTP

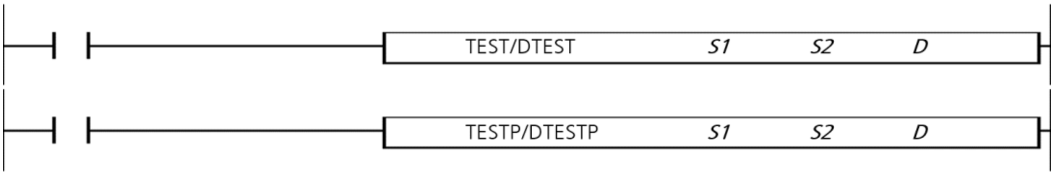
Supported PLC Series

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

TEST(P) and DTEST(P) fetch bit data and store in bit device.

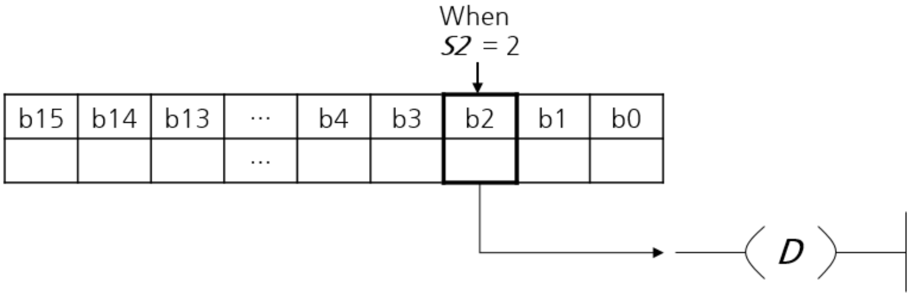
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
TEST(P) DTEST(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	-	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	-	-	-	-	-	-				



S1	Head address of word/double word device to fetch bit data.
S2	Location of bit data to be fetched from the word/double word device assigned to *S1*.
D	Bit device where fetched bit data will be stored.

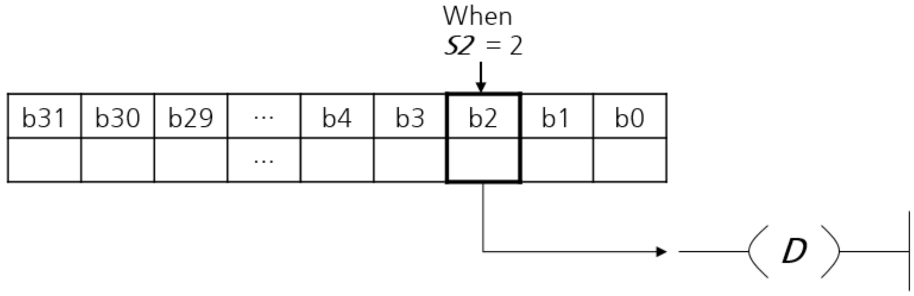
TEST, TESTP

- TEST and TESTP fetch bit data which is located in *S2*. *S2* indicates the location of bit fetched from word device assigned to *S1*. Fetched bit data is stored in bit device assigned to *D*.
- Status of bit device assigned to *D* is OFF when value is “0”. It is ON when value is “1”.
- 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.)

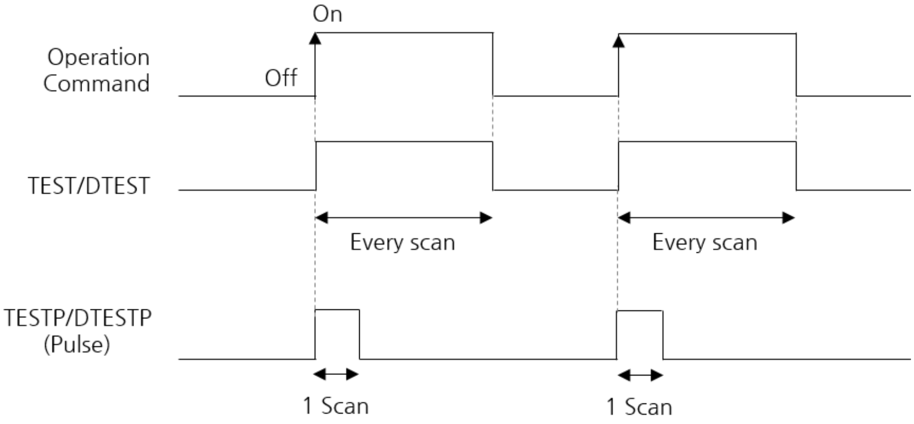


DTEST, DTESTP

- DTEST and DTESP fetches bit data which is located in *S2*. *S2* indicates the location of bit fetched from double word device assigned to *S1* or *S1* +1. Fetched bit data is stored in bit device assigned to *D*.
- Status of bit device assigned to *D* is OFF when value is “0”. It is ON when value is “1”.
- Value of *S2* is valid from 0 to 31. If value of *S2* is greater than or equal to 32, the location of bit is indicated by the remainder of n/32. (e.g. When *S2* is 34, it is equivalent to the case when *S2* is 2.)



Execution Condition



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.

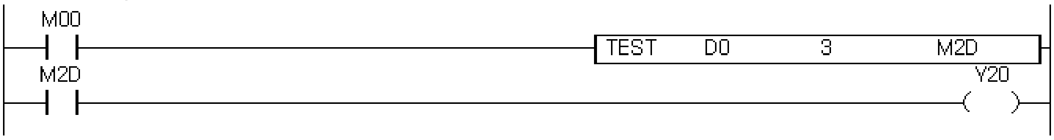
(Range of device D depends on CPU type)

Program Example

TEST, TESTP

When M00 is turned ON, the instruction is executed. M2D is turned ON or OFF according to the bit 3 in the word device D0. You can check the status of M2D through output Y20. When M2D is “1”, Y20 is turned ON.

Ladder Diagram (LD)



Instruction List (IL)

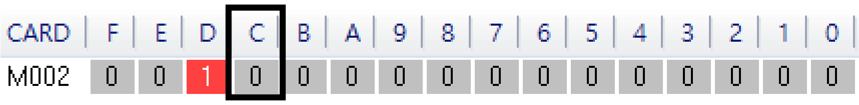
Instruction	Device		
LD	M00		
TEST	D0	3	M2D
LD	M2D		
OUT	Y20		

The program operates as following:

- When D0 is 11 (BIN=0000 0000 0000 1011), M2D is ON.

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
When D0 = 11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

When S2 = 3

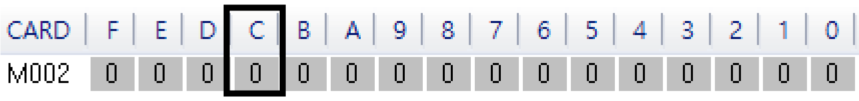


CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M002	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

- When D0 is 16 (BIN=0000 0000 0001 0000), M2D is OFF.

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
When D0 = 16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

When S2 = 3



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2.6.2 BSET, BSETP, BRST, BRSTP

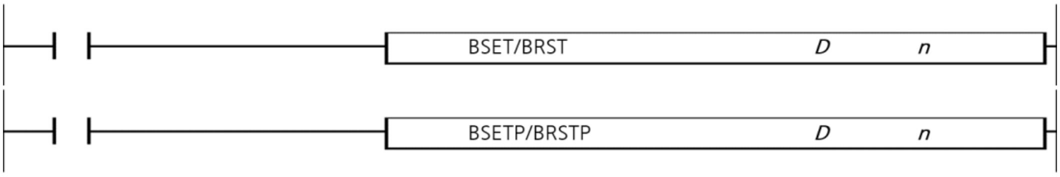
Supported PLC Series

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

BSET, BSETP turn ON the assigned bit. BRST, BRSTP turn OFF the assigned bit.

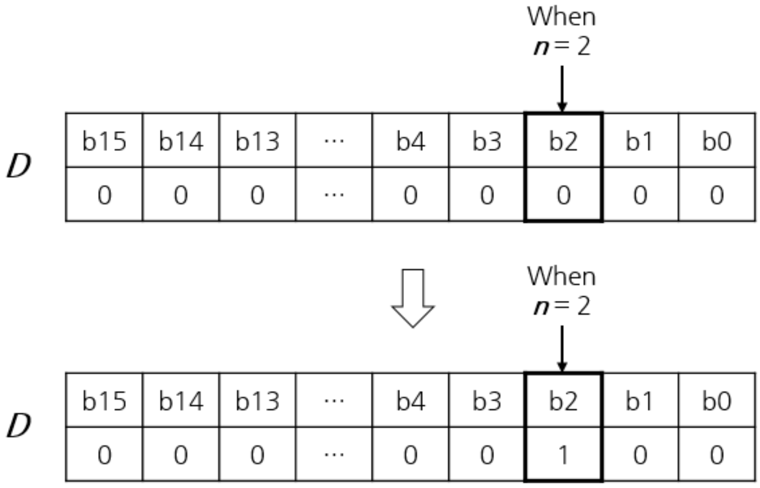
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BSET(P) BRST(P)	*D*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-	3	O	-	-
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



D	Address of word device to set and reset bit.
n	Location of bit data to set and reset.

BSET, BSETP

- BSET and BSETP turn ON the bit *n* of device assigned to *D*.
- *n* is valid from 0 - 15. If it is greater than or equal to 16, the location of bit is indicated by the remainder of *n* /16. (e.g. When *n* is 17, it is equivalent to the case when *n* is 1).
- Bits turned ON by BSET/BSETP can be turned OFF by other instructions.

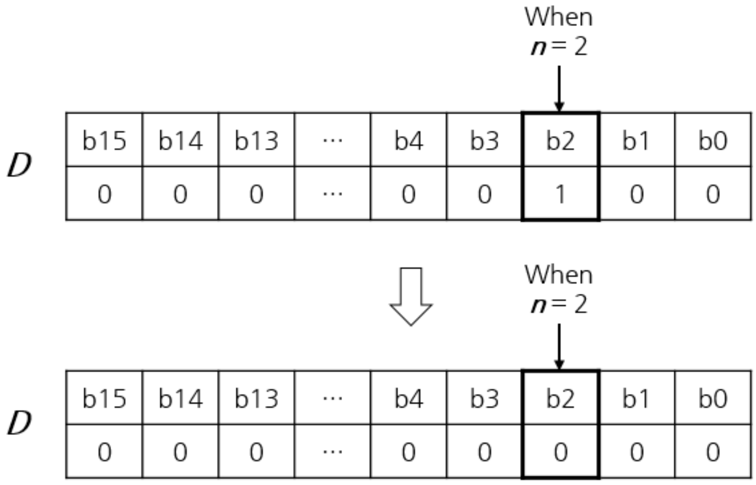


D	b15	b14	b13	···	b4	b3	b2	b1	b0
	0	0	0	···	0	0	0	0	0

D	b15	b14	b13	···	b4	b3	b2	b1	b0
	0	0	0	···	0	0	1	0	0

BRST, BRSTP

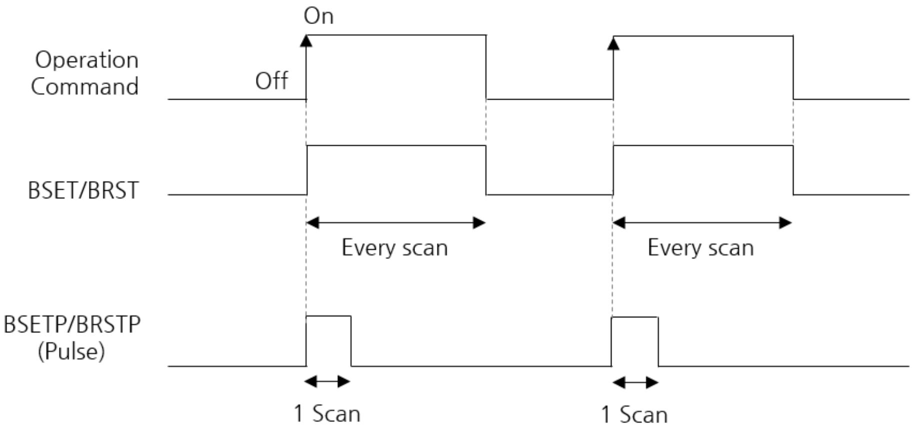
- BRST and BRSTP turn OFF the bit *n* of device assigned to *D*.
- *n* is valid from 0 to 15. If it is greater than or equal to 16, the location of bit is indicated by the remainder of *n* /16. (e.g. When *n* is 17, it is equivalent to the case when *n* is 1).
- Bits turned OFF by BRST/BRSTP can be turned ON by other instructions.



b15	b14	b13	···	b4	b3	b2	b1	b0
0	0	0	···	0	0	1	0	0

b15	b14	b13	···	b4	b3	b2	b1	b0
0	0	0	···	0	0	0	0	0

Execution Condition



Operation Error

Error Flag (F110)

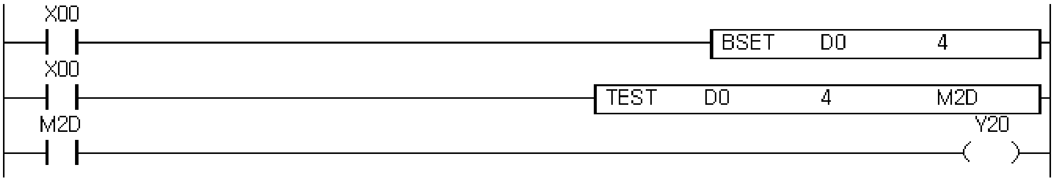
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BSET, BSETP

The following program turns ON bit 4 of word device D0 according to the execution condition. When X00 is ON, bit 4 of D0 is turned ON. To check the status of bit 4 of D0, you can fetch the bit and store it at M3D using TEST instruction. Then check the bit through output Y20.

Ladder Diagram (LD)

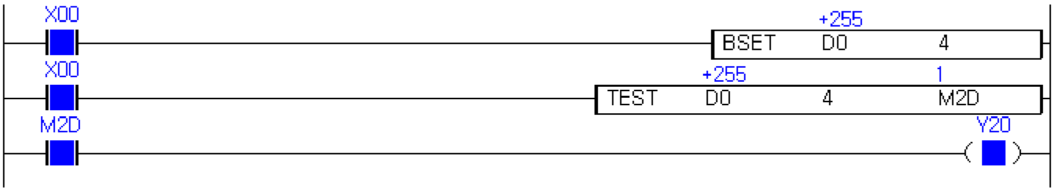


Instruction List (IL)

Instruction	Device		
LD	X00		
BSET	D0	4	
LD	X00		
TEST	D0	4	M2D
LD	M2D		
OUT	Y20		

The program operates as following:

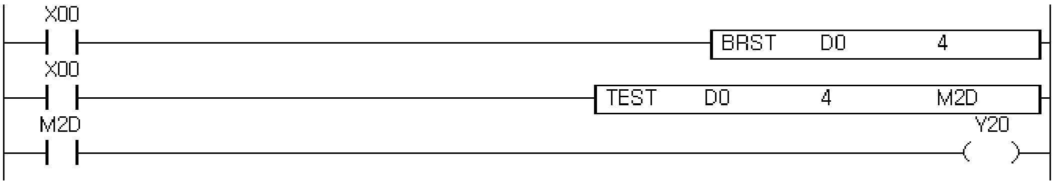
- Input “239” at D0 (BIN= 0000 0000 1110 1111).
- When X00 executes, the bit 4 of D0 will be set. Therefore, value of D0 becomes “255” (BIN=0000 0000 1111 1111).



BRST, BRSTP

The following program turns OFF bit 4 of D0 according to the execution condition. When X00 is ON, bit 4 of D0 is turned OFF. To check the status of bit 4 of D0, you can fetch the bit and store it at M3D using TEST instruction. Then check the bit through output Y20.

Ladder Diagram (LD)

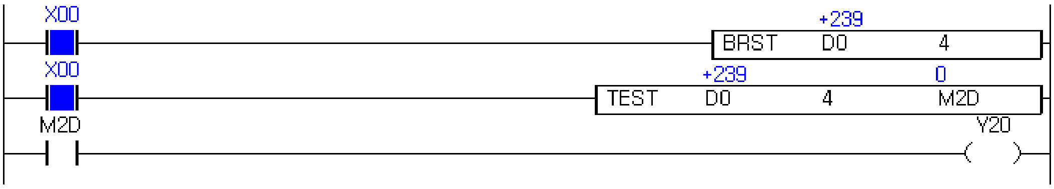


Instruction List (IL)

Instruction	Device		
LD	X00		
BRST	D0	4	
LD	X00		
TEST	D0	4	M2D
LD	M2D		
OUT	Y20		

The program operates as following:

- Input “255” at D0 (BIN= 0000 0000 1111 1111).
- When X00 executes, the bit 4 of device D will be reset. Therefore, value of D0 becomes “239”
  (BIN=0000 0000 1110 1111).



2.7 Data Transfer Instructions

2.7.1 MOV, MOVP, DMOV, DMOVP

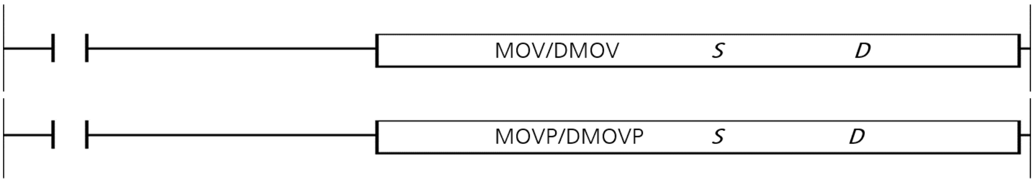
Supported PLC Series

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

MOV(P) instructions transfer constant or word data from word device *S* to word device *D*. DMOV(P) instructions transfer constant or double word data from double word device *S* to double word device *D*.

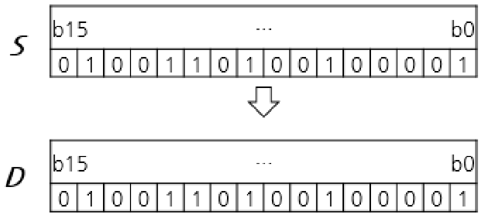
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
MOV(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
DMOV(P)	*D*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Constant or head address of word/double word device where data to be transferred is stored.
D	Head address of word/double word device where transferred word/double word data will be stored.

MOV, MOVP

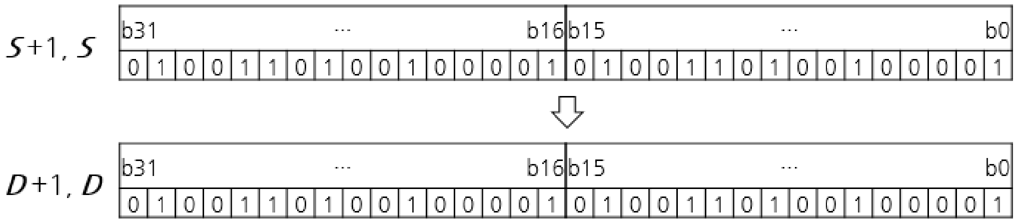
- MOV(P) instructions transfer constant or word data from word device *S* to word device *D*.
- 16-bit data is transferred.



	b15							···								b0
S	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1
D	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1

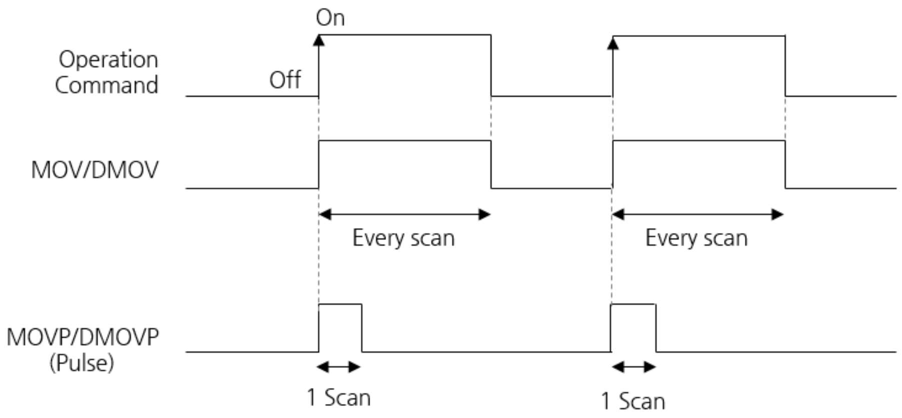
DMOV, DMOVP

- DMOV(P) instructions transfer constant or double word data from double word device *S* to double word device *D*.
- 32-bit data is transferred.



	b31								···							b16	b15								···							b0
S+1, *S*	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1
D+1, *D*	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1

Execution Condition



Operation Error

Error flag (F110)

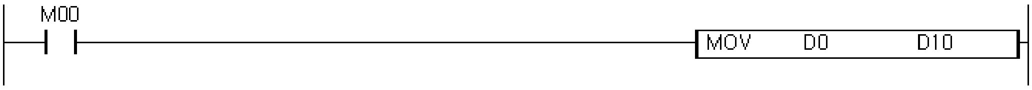
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

MOV

When operation command M00 is ON, MOV instruction is executed. The word data assigned to word device D0 is transferred to word device D10.

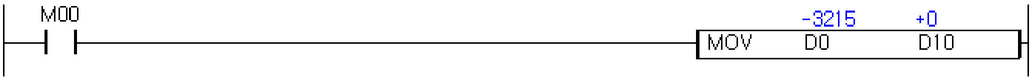
Ladder Diagram (LD)



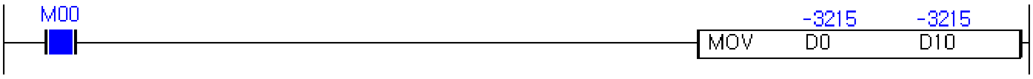
Instruction List (IL)

Instruction	Device	
LD	M00	
MOV	D0	D10

- The value assigned to D0 is “-3215”.



- When M00 turns ON, MOV instruction is executed. Then, the value of D0, “-3215” is transferred to D10.



2.7.2 EMOV, EMOVP

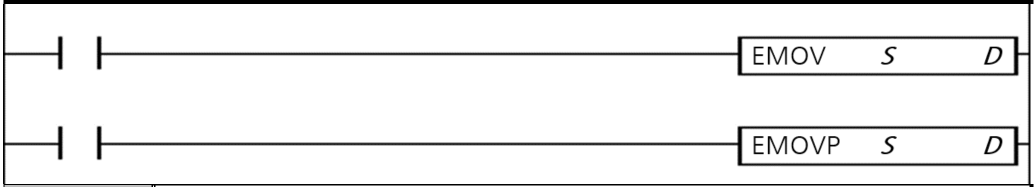
Supported PLC Series

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

Function

EMOV(P) instructions transfer 32-bit floating-point number from device *S* to device *D*.

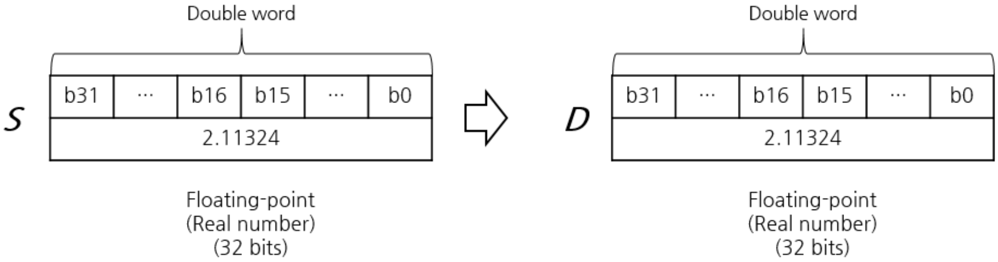
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EMOV(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



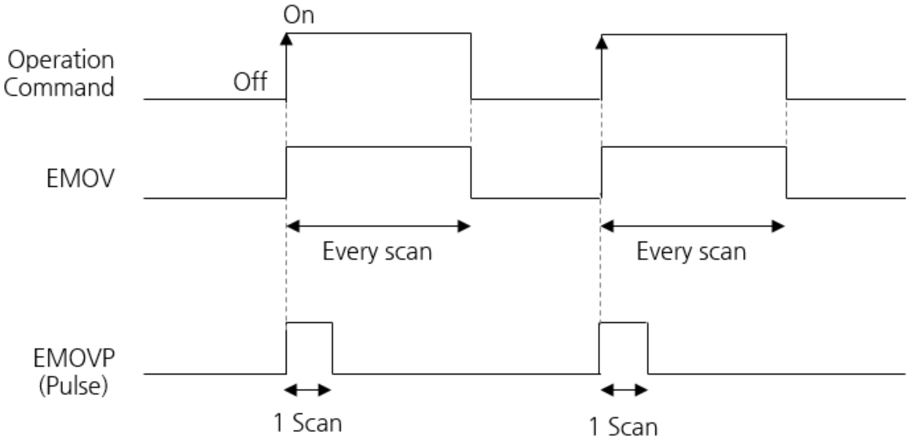
S	32-bit floating-point number or head address of device where data to be transferred is stored.
D	Head address of device where transferred 32-bit floating-point number will be stored.

EMOV, EMOVP

- EMOV(P) instructions transfer 32-bit floating-point number from device *S* to device *D*.
- Values which can be assigned and stored to *S* and *D* are as following.
  $2^{-127} \leq$ | Assigned value (stored value) | $\leq 2^{128}$



Execution Condition



Operation Error

Error Flag (F110)

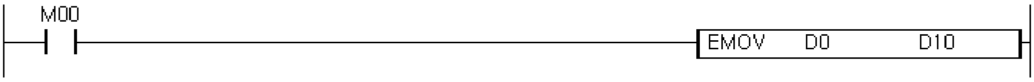
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

EMOV

When operation command M00 is ON, MOV instruction is executed. The 32-bit floating-point number assigned to device D0 is transferred to device D10.

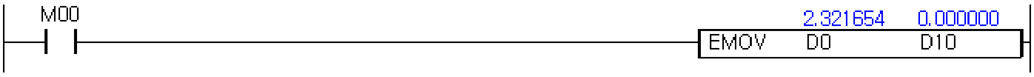
Ladder Diagram (LD)



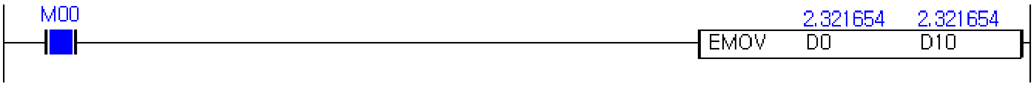
Instruction List (IL)

Instruction	Device	
LD	M00	
EMOV	D0	D10

- The value assigned to D0 is “2.321654”.



- When M00 turns ON, MOV instruction is executed. Then, the value of D0, “2.321654” is transferred to D10.



2.7.3 CML, CMLP, DCML, DCMLP

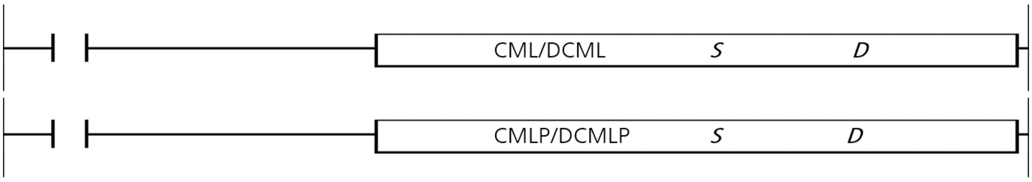
Supported PLC Series

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

CML(P) invert every bit of constant or word data assigned to word device and the inversion result is stored in word device *D*.
DCML(P) invert every bit of constant or double word data assigned to double word device *S* and the inversion result is stored in double word device *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
CML(P) DCML(P)	*S*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S	Constant or head address of word/double word device where data to be inverted is stored.
D	Head address of word/double word device where inversion result will be stored.

CML, CMLP

- CML(P) instructions invert every bit of constant or word data assigned to device *S*. The inversion result is stored in word device *D*.
- 16-bit data is inverted.

	b15							···								b0
S	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1

⇩

	b15							···								b0
D	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	0

DCML, DCMLP

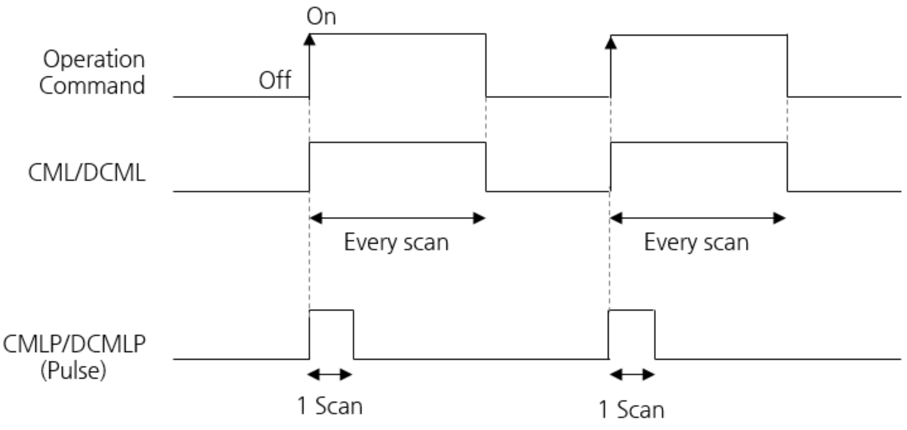
- DCML(P) instructions invert the every bit of constant or double word data assigned to device *S*. The inversion result is stored in double word device *D*.
- 32-bit data is inverted.

	b31							···								b16	b15							···								b0
S+1, *S*	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1

⇩

	b31							···								b16	b15							···								b0
D+1, *D*	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	0

Execution Condition



Operation Error

Error Flag (F110)

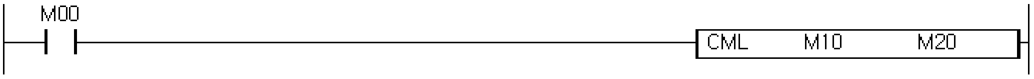
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

CML

When operation command M00 is ON, CML instruction is executed. The word data assigned to word device M10 is inverted. Then, inversion result is stored in word device M20.

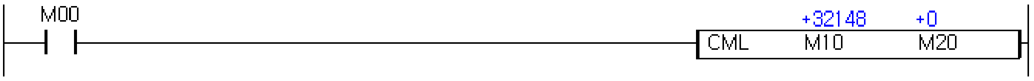
Ladder Diagram (LD)



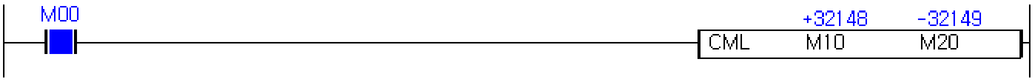
Instruction List (IL)

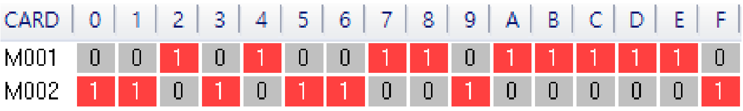
Instruction	Device	
LD	M00	
CML	M10	M20

- The value assigned to M10 is “32148”.



- When M00 is turned ON, CML instruction is executed. Every bit of M10 is inverted. Then, the inversion result is stored in M20.





CARD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
M001	0	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0
M002	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0	1

2.7.4 BMOV, BMOVP

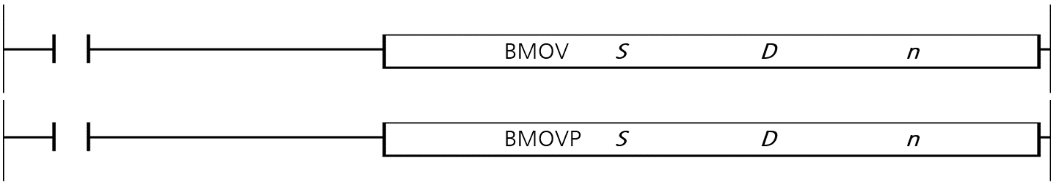
Supported PLC Series

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

BMOV(P) instructions transfer *n* blocks of word data starting from word device *S*. Transferred data are stored in word device starting from *D*.

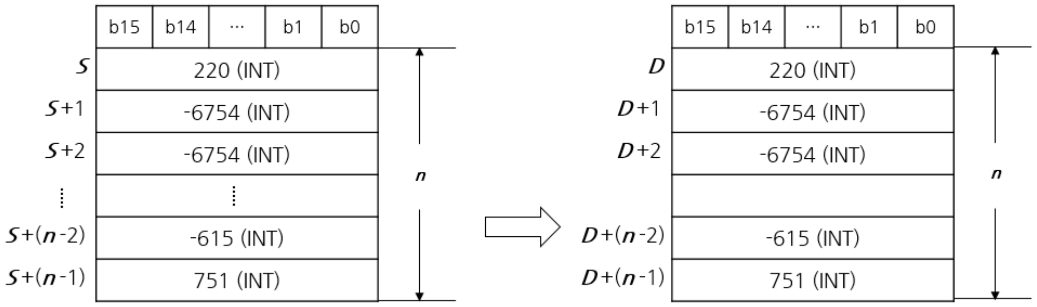
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BMOV(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



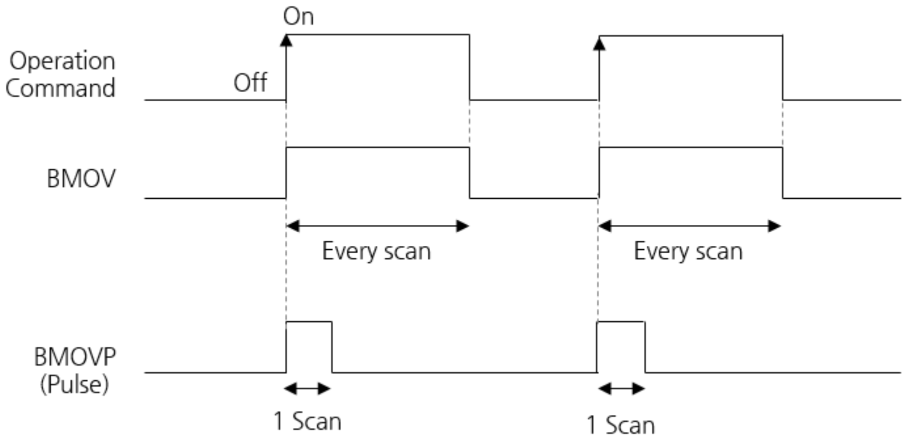
S	Head address of the devices where word data blocks to be transferred.
D	Head address of the devices where transferred word data blocks will be stored.
n	The number of data blocks to be transferred.

BMOV, BMOVP

- BMOV(P) transfer *n* blocks of word data, starting from word device *S* to word device *D*.
- Transferring word data blocks can be operated even in case of an overlap between word devices designated by *S* and *D*.
- Overlap can happen when number of transferred word data blocks are larger than the distance between word devices assigned to *S* and *D*.
- 16-bit data can be assigned to *S* and *D*.
- If *n* exceeds the range of 1 ~ 32767, the instruction is not executed.



Execution Condition



Operation Error

Error Flag (F110)

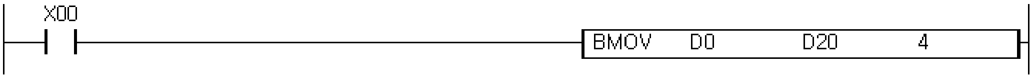
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

BMOV

When operation command X00 is ON, BMOV instruction is executed. The instruction transfers 4 blocks of word data starting from D0 (D0 ~ D3) to D20 (D20 ~ D23).

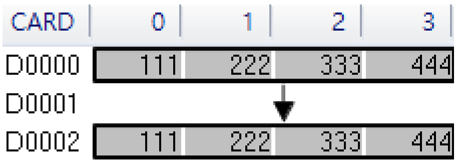
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	X00		
BMOV	D0	D20	4

- When operation command X00 is ON, BMOV instruction is executed. Then, 4 blocks of word data, D0 ~ D3 are transferred to D20 ~ D23.



CARD	0	1	2	3
D0000	111	222	333	444
D0001				
D0002	111	222	333	444

2.7.5 FMOV, FMOVP

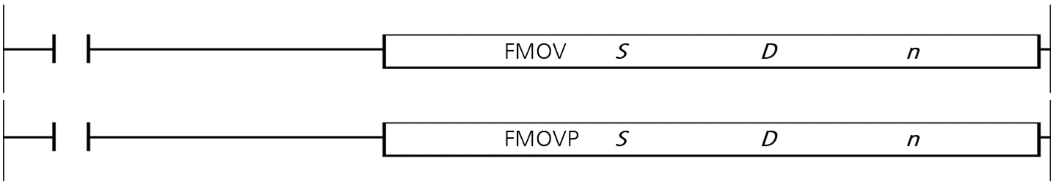
Supported PLC Series

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

FMOV(P) transfer the word data assigned to device *S* to *n* devices starting from word device *D*.

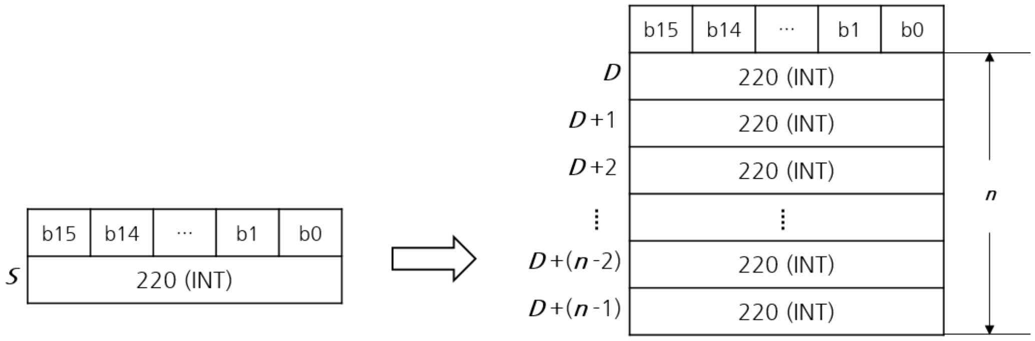
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FMOV(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



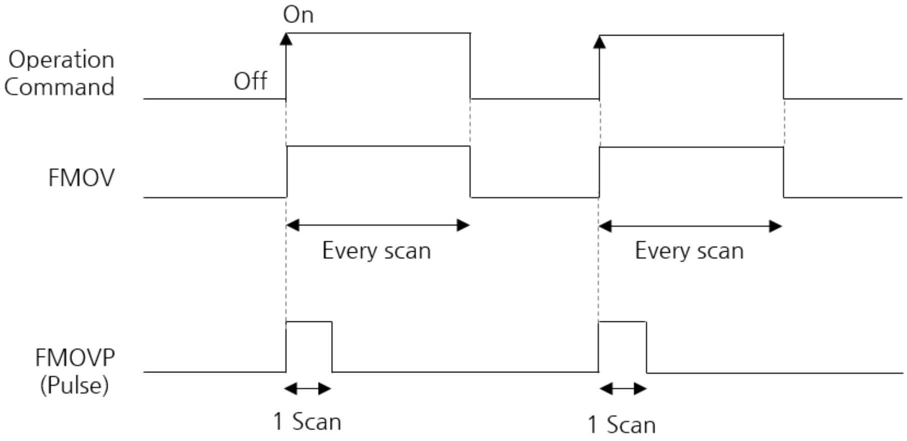
S	Constant or head address of the device where word data to be transferred.
D	Head address of the device where transferred word data will be stored.
n	The number of word devices where word data will be transferred.

FMOV, FMOVP

- FMOV(P) instructions copy the word data assigned to device *S*. Then, copied data is transferred to *n* word devices starting from word device *D*.
- 16-bit data can be assigned to *S* and *D*.
- If *n* exceeds the range of 1 ~ 32767, the instruction is not executed.



Execution Condition



Operation Error

Error Flag (F110)

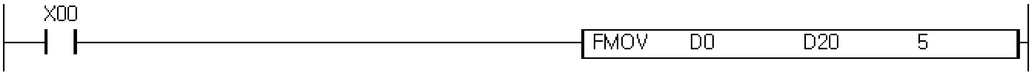
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

FMOV

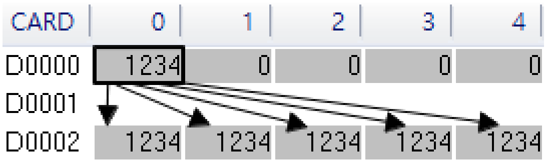
When operation command X00 is ON, FMOV instruction is executed. Then, the value of D0 is copied and transferred to 5 word devices starting from D20 (D20 ~ D24).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	X00		
FMOV	D0	D20	5



CARD	0	1	2	3	4
D0000	1234	0	0	0	0
D0001					
D0002	1234	1234	1234	1234	1234

2.7.6 WBMOV, WBMOVP

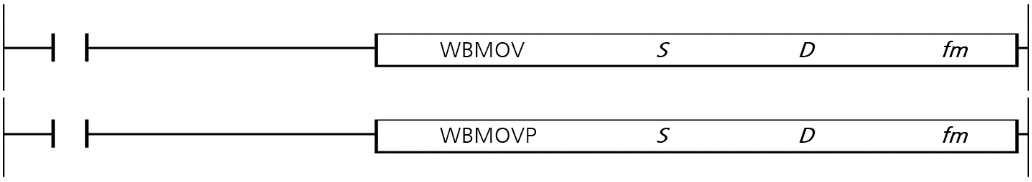
Supported PLC Series

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

WBMOV(P) transfer the specified bit data according to the format assigned to *fm*. The location and number of bits to be transferred are specified by *fm*. Specified bit data are transferred from specified location of word device *S* to specified location of word device *D*.

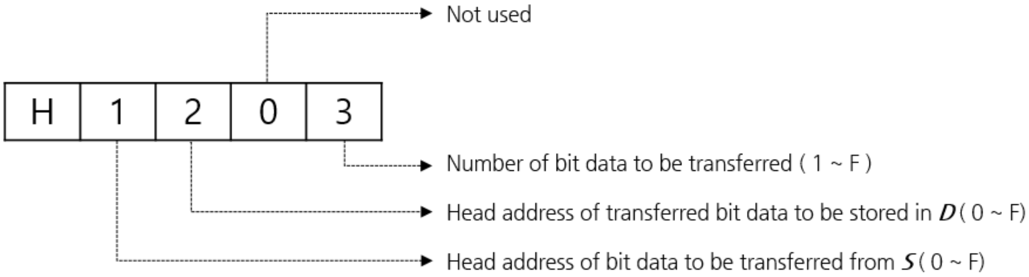
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
WBMOV(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	fm	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

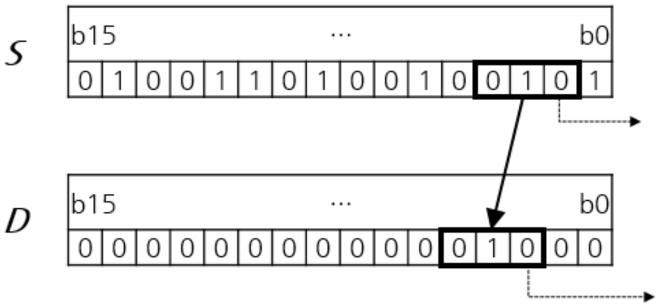


S	Address of the word device to fetch data.
D	Head address of the word device where fetched data will be transferred.
fm	Format to execute WBMOV(P). The format includes the information of head address of bit data in device *S* and *D* and number of bits to be transferred.

WBMOV, WBMOVP

- WBMOV(P) transfer bit data according to the format assigned to *fm*.
- *fm* includes the information of head address of bit data in word device *S* and word device *D*.
- *fm* also includes the information of the number of bit data to be transferred.
- Specified number of bit data, starting from the specified location in word device *S*, are transferred to the specified location in word device *D*.
- *fm* should be written in hexadecimal. The format of *fm* is as following.


Not used
H
1
2
0
3
Number of bit data to be transferred (1 ~ F)
Head address of transferred bit data to be stored in *D* (0 ~ F)
Head address of bit data to be transferred from *S* (0 ~ F)



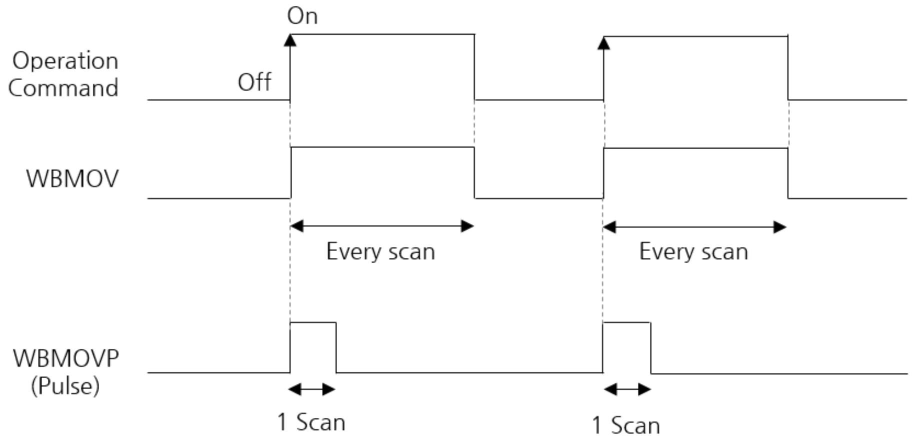
	b15							…								b0
S	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	1
D	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

When ***fm*** = H1203,
3 bits starting from bit 1 of device ***S***
are specified.

When ***fm*** = H1203,
3 bits starting from bit 2 of device ***D***
are specified.

- When the least significant digit in *fm* is 0, the instruction is not executed.
- Overlap can happen when number of transferred bit data are larger than the distance between word devices designated by *S* and *D*.

Execution Condition



Operation Error

Error Flag (F110)

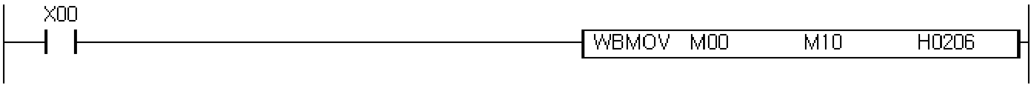
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

WBMOV

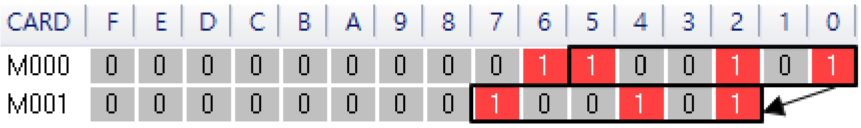
When operation command X00 is ON, WBMOV instruction is executed. Then, bit addresses of source device M00 and destination device M10 are specified. As H0206 is assigned to *fm*, 6 bits starting from bit 0 of word device M00 are transferred to the bit 2 of word device M10. It means that bit data M00 ~ M5 are transferred to M12 ~ M17.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	X00		
WBMOV	M00	M10	H0206



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1
M001	0	0	0	0	0	0	0	0	1	0	0	1	0	1		

2.7.7 BITMOV, BITMOVP

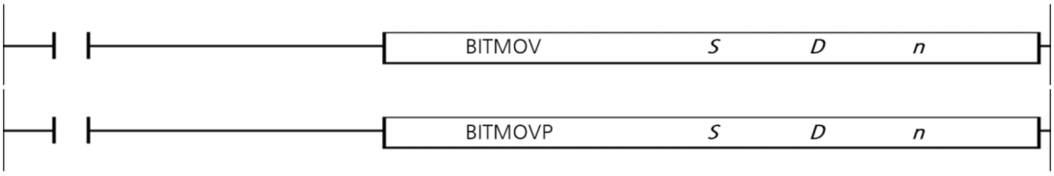
Supported PLC Series

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

BITMOV(P) instructions transfer bits from a device to another device.

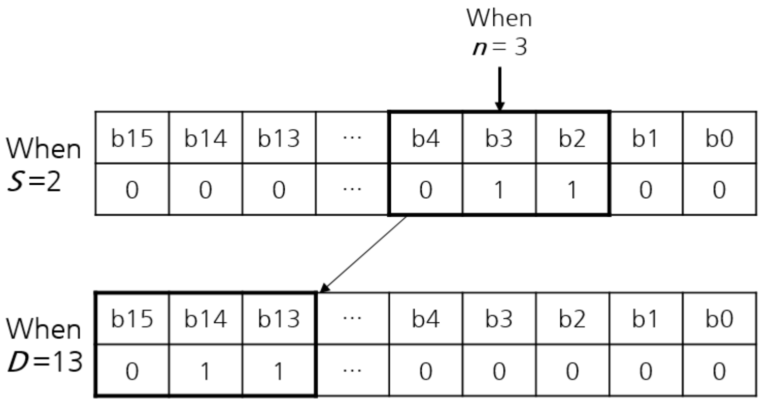
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BITMOV(P)	*S*	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	4	O	-	-
	D	O	-	O	O	O	-	-	-	-	-	-	-	-	-	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



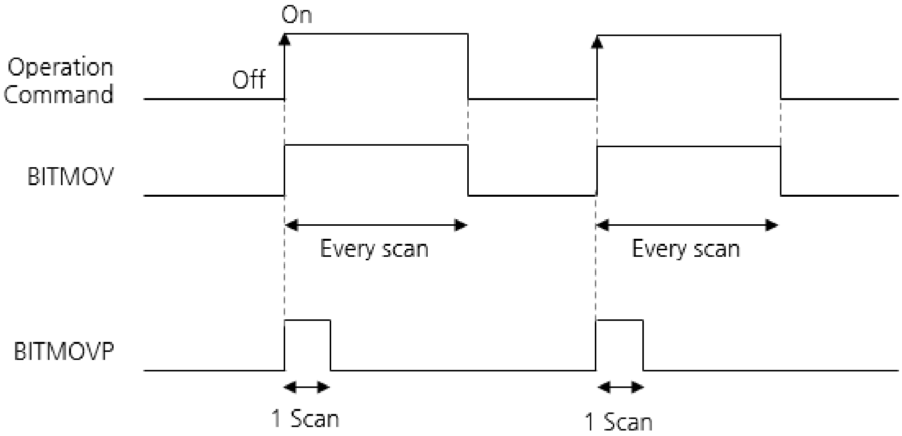
S	Address of bit device where the bit data to be transferred is stored.
D	Address of bit device where the transferred bit data will be stored.
n	The number of bits to transfer from device assigned to *S* to device assigned to *D*. (*n* ≦16)

BITMOV, BITMOVP

- BITMOV(P) specifies the number of bits (*n*) to be transferred. The number of bits starts from the head address of bit device which is assigned to *S*.
- After the execution, specified bits are copied to the bit device starting from the bit address assigned to *D*.
- Bit data transfer can be operated even in case of an overlap between bit devices assigned to *S* and *D*.
- Overlap can happen when number of transferred bits are larger than the distance between bit devices assigned to *S* and *D*.



Execution Condition



Operation Error

Error Flag (F110)

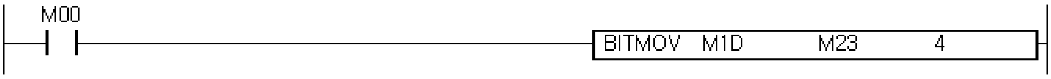
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BITMOV/BITMOVP

This program transfers 4 bits starting from M1D (M1D - M20) to M23 (M23 - M26) when M00 is turned ON.

Ladder Diagram (LD)

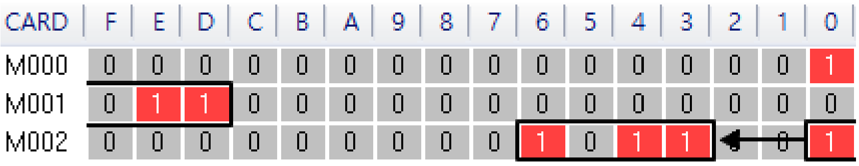


Instruction List (IL)

Instruction	Device		
LD	M00		
BITMOV	M1D	M23	4

The program operates as following:

- Set M1D, M1E and M20. Leave M1F as “0”. When the program executes, these 4 bits are transferred.
- M1D is copied to M23. M1E is copied to M24. M1F is copied to M25. M20 is copied to M26.



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
M001	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
M002	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1

2.7.8 XCH, XCHP, DXCH, DXCHP

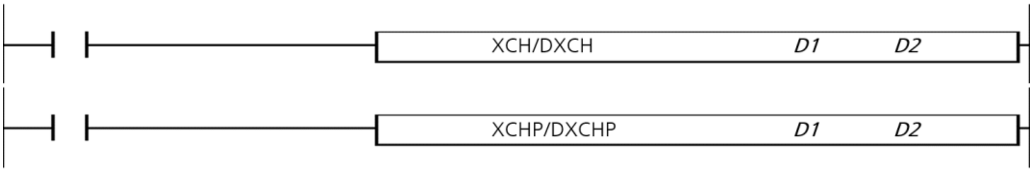
Supported PLC Series

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

XCH(P) instructions exchange word data between word device *D1* and word device *D2*. DXCH(P) instructions exchange double word data between double word device *D1* and double word device *D2*.

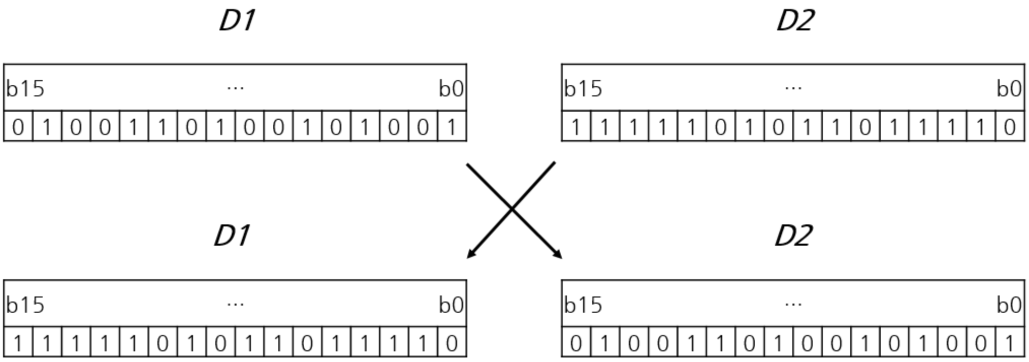
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
XCH(P)	*D1*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-	3	O	-	-
DXCH(P)	*D2*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



D1	Head address of word/double word device where data to be exchanged is stored.
D2	Head address of word/double word device where data to be exchanged is stored.

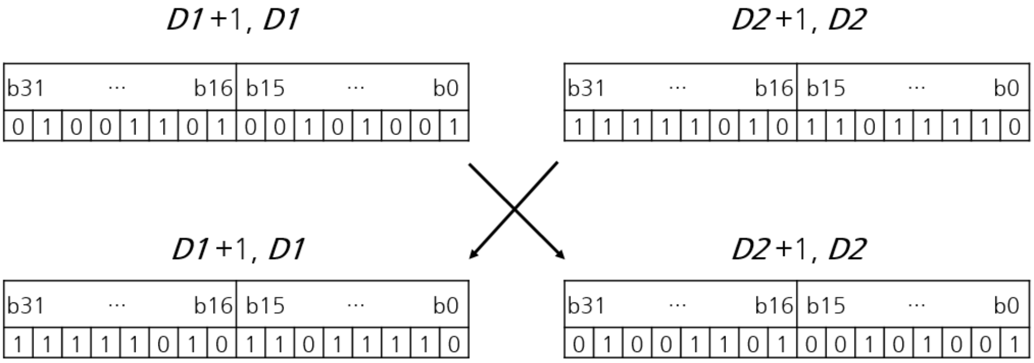
XCH, XCHP

- XCH(P) instructions exchange word data between word device *D1* and word device *D2*.
- 16-bit data is exchanged.



DXCH, DXCHP

- DXCH(P) instructions exchange double word data between double word device *D1* and double word device *D2*.
- 32-bit data is exchanged.



D1+1, *D1*

b31			···				b16	b15			···				b0
0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	1

D2+1, *D2*

b31			···				b16	b15			···				b0
1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	0

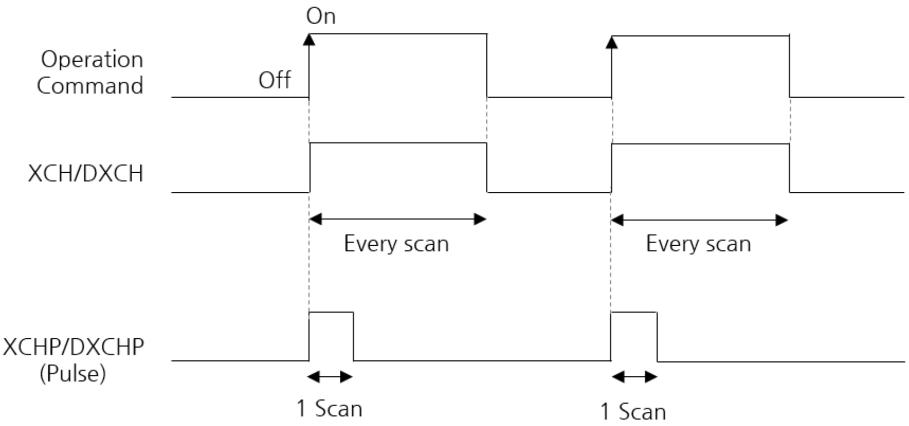
D1+1, *D1*

b31			···				b16	b15			···				b0
1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	0

D2+1, *D2*

b31			···				b16	b15			···				b0
0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	1

Execution Condition



⚠ Using pulse contact as execution condition of XCH/DXCH instruction or using XCHP/DXCHP instruction is recommended.

Operation Error

Error flag (F110)

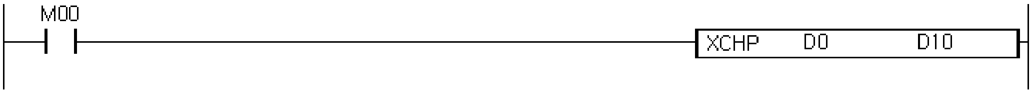
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

XCHP

When operation command M00 is ON, XCHP instruction is executed. The word data assigned to word device D0 and D10 are exchanged once.

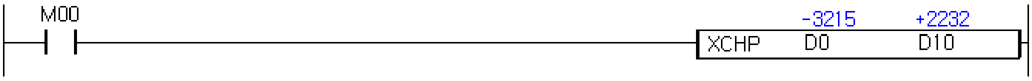
Ladder Diagram (LD)



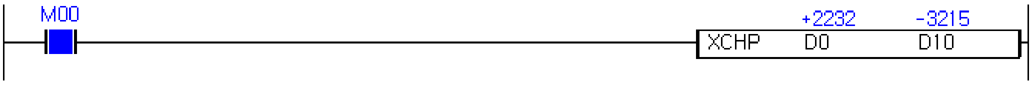
Instruction List (IL)

Instruction	Device	
LD	M00	
XCHP	D0	D10

- The value assigned to D0 is “-3215”. The value assigned to D10 is “2232”.



- When M00 turns ON, XCHP instruction is executed. Then, the value of D0, “-3215” and the value of D10, “2232” are exchaged once.



2.7.9 BXCH, BXCHP

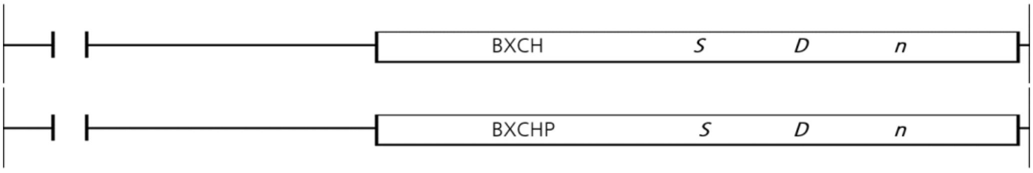
Supported PLC Series

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

BXCH(P) instructions exchange *n* blocks of word data between word device *S* and word device *D*.

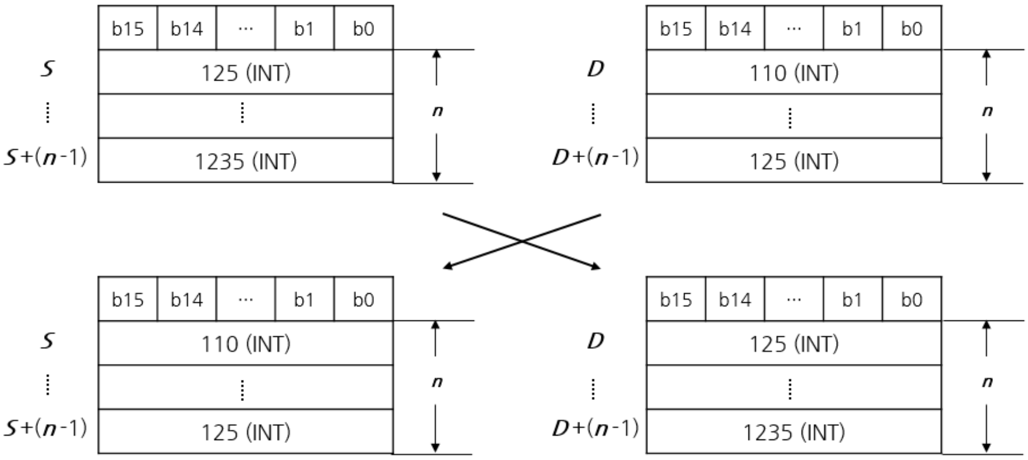
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BXCH(P)	*S*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-	4	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



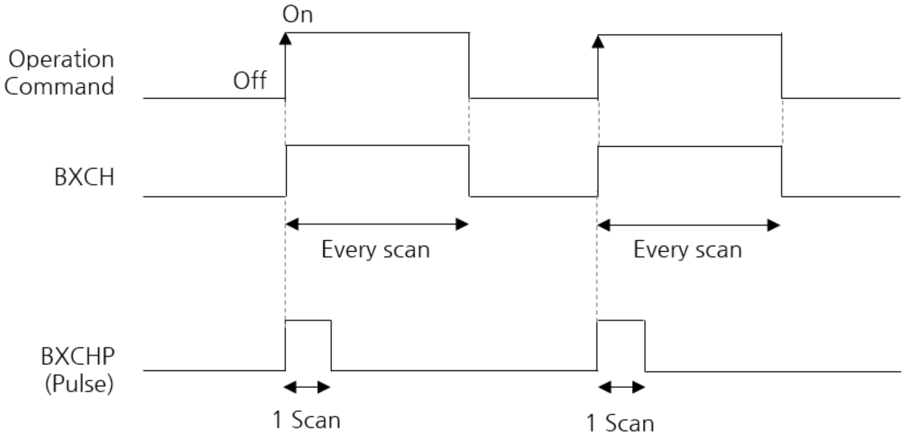
S	Head address of word devices where data to be exchanged is stored.
D	Head address of word devices where data to be exchanged is stored.
n	The number of word data blocks to exchange.

BXCH, BXCHP

- BXCH(P) instructions exchange *n* word data blocks.
- Data from word device *S* to *S*+(*n*-1) and from word device *D* to *D*+(*n*-1) are exchanged.
- 16-bit data is assigned to each word device.



Execution Condition



⚠ Using pulse contact as execution condition of BXCH instruction or using BXCHP instruction is recommended.

Operation Error

Error Flag (F110)

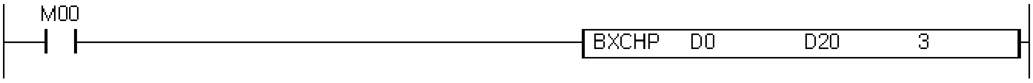
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

BXCHP

When operation command M00 is ON, BXCHP instruction is executed. The 3 word data blocks starting from word device D0 (D0, D1, D2) are exchanged for 3 word data blocks starting from word device D20 (D20, D21, D22) once.

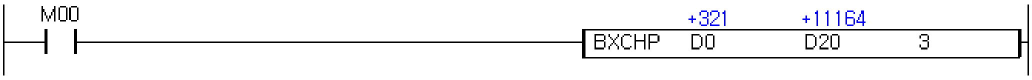
Ladder Diagram (LD)



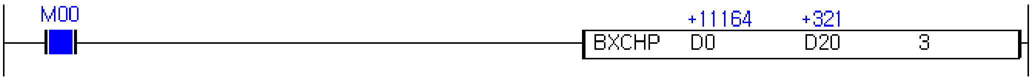
Instruction List (IL)

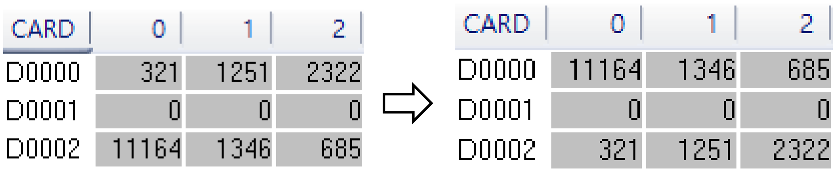
Instruction	Device		
LD	M00		
BXCHP	D0	D20	3

- The value assigned to D0 is “321”, D1 is “1251”, D2 is “2322”.
- The value assigned to D20 is “11164”, D21 is “1346”, D22 is “685”.



- When M00 turns ON, BXCHP instruction is executed. Then, the value of 3 word data blocks starting from D0 (D0, D1, D2) and 3 word data blocks starting from D20 (D20, D21, D22) are exchanged once.





CARD	0	1	2
D0000	321	1251	2322
D0001	0	0	0
D0002	11164	1346	685

⇨

CARD	0	1	2
D0000	11164	1346	685
D0001	0	0	0
D0002	321	1251	2322

2.7.10 SWAP, SWAPP

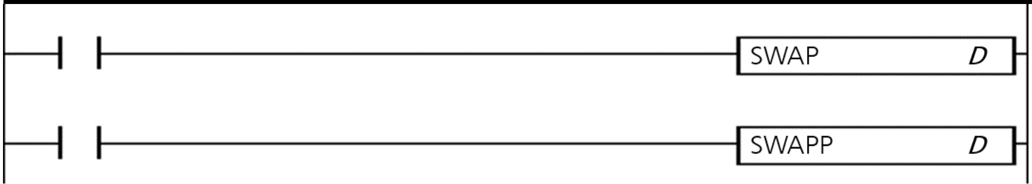
Supported PLC Series

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

SWAP(P) instructions exchange upper byte of word device *D* with lower byte of word device *D* and the result is stored in word device *D*.

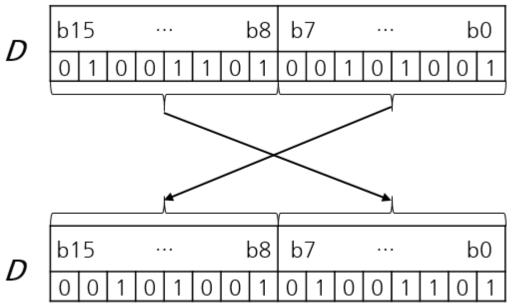
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SWAP(P)	*D*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-



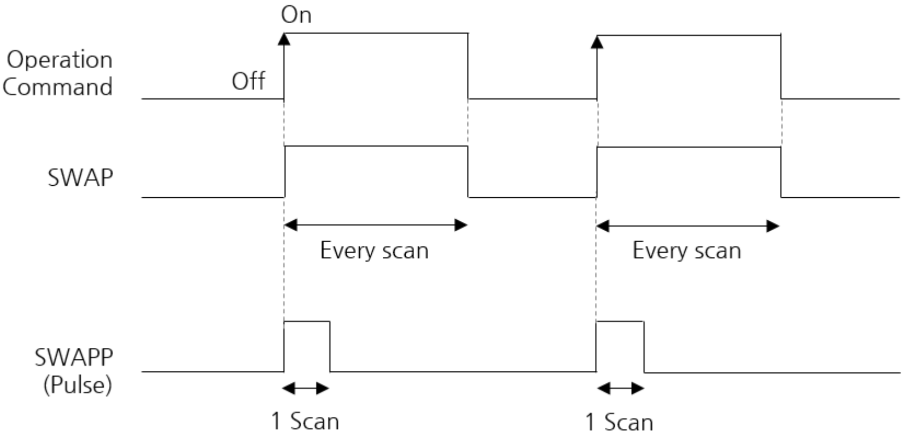
D	Head address of word device where upper byte and lower byte are exchanged and the result is stored.

SWAP, SWAPP

- SWAP(P) instructions exchange upper byte of word device *D* with lower byte of word device *D*.
- The exchange result is stored in word device *D*.



Execution Condition



⚠ Using pulse contact as execution condition of SWAP instruction or using SWAPP instruction is recommended.

Operation Error

Error Flag (F110)

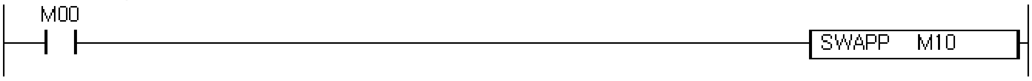
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

SWAPP

When operation command M00 is ON, SWAPP instruction is executed. The upper byte of M10 (M18 ~ M1F) and the lower byte of M10 (M10 ~ M17) are exchanged. Then, exchange result is stored in word device M10.

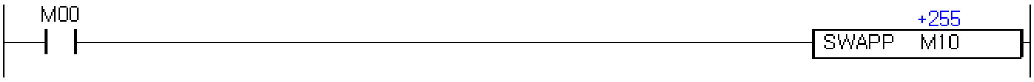
Ladder Diagram (LD)



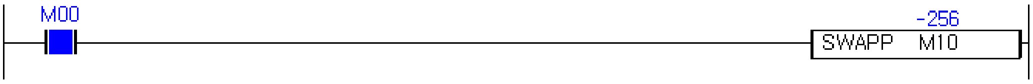
Instruction List (IL)

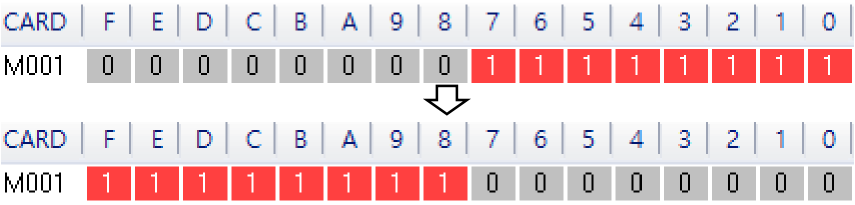
Instruction	Device
LD	M00
SWAPP	M10

- The value assigned to M10 is “255”.



- When M00 is turned ON, SWAPP instruction is executed. Upper byte of M10 (M18 ~ M1F) and lower byte of M10 (M10 ~ M17) are exchanged. Then, the exchange result is stored in M10.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M001	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M001	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

2.8 Data Table Instructions

2.8.1 FIFW, FIFWP

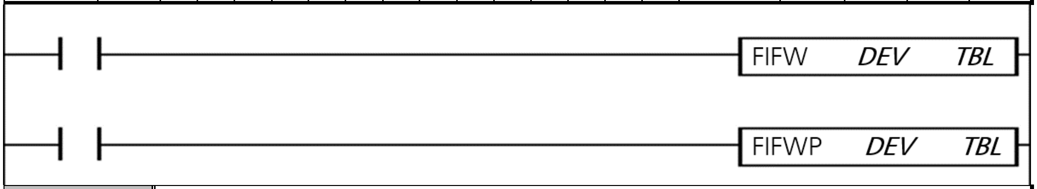
Supported PLC Series

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

FIFW(P) instructions write data to the data table.

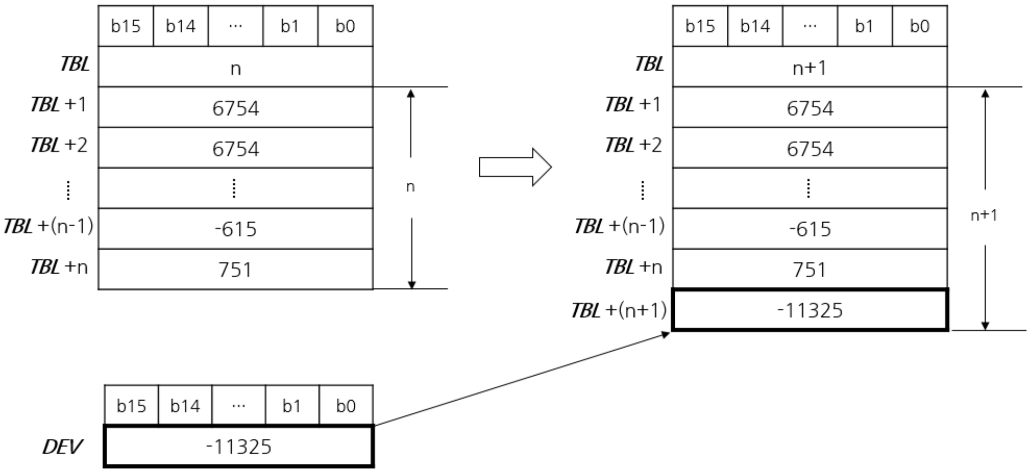
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FIFW(P)	*DEV*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	TBL	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-				



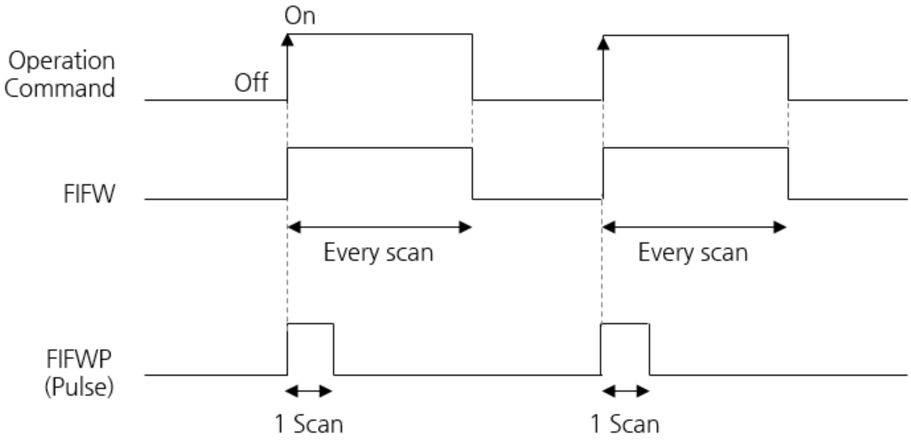
DEV	Address of word device where data to be written into the table is stored.
TBL	Head address of word device where the table is located.

FIFW, FIFWP

- FIFW(P) instructions write the data stored in word device assigned to *DEV* into data table starting from the word device assigned to *TBL*. The data is stored in sequence starting from *TBL* +1.
- In device *TBL*, the number of data blocks in the table is stored.



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)

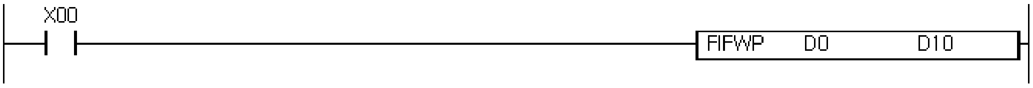
F110 turns ON for 1 scan when the data table exceeds the range of corresponding device.

Program Example

FIFWP

When X00 turns ON, FIFWP instruction writes the data stored in D0 into the data table starting from D10. On the first execution of FIFWP instruction, the data is written into D11. On the second execution, the data is written into D12. Every time the data is added to the table, the value stored in D10 increases by 1.

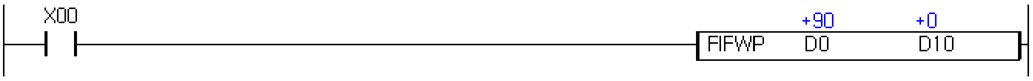
Ladder Diagram (LD)



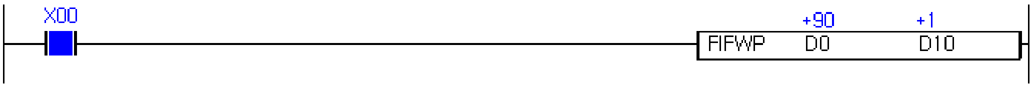
Instruction List (IL)

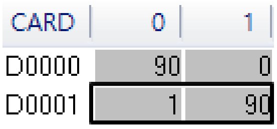
Instruction	Device	
LD	X00	
FIFWP	D0	D10

- Data “90” is assigned to word device D0.



- When X00 turns ON, “90” is written to the device D11. In D10, the number of data blocks (= 1) is stored.





CARD	0	1
D0000	90	0
D0001	1	90

2.8.2 FIFR, FIFRP

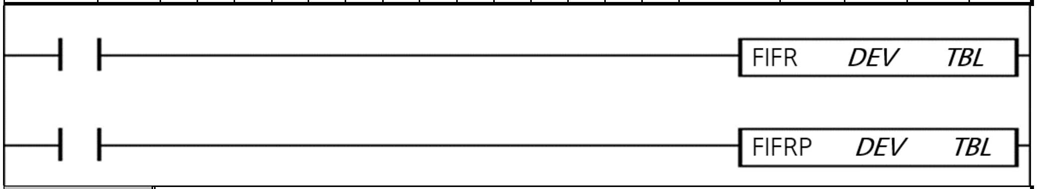
Supported PLC Series

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

FIFR(P) instructions read the oldest data from the data table.

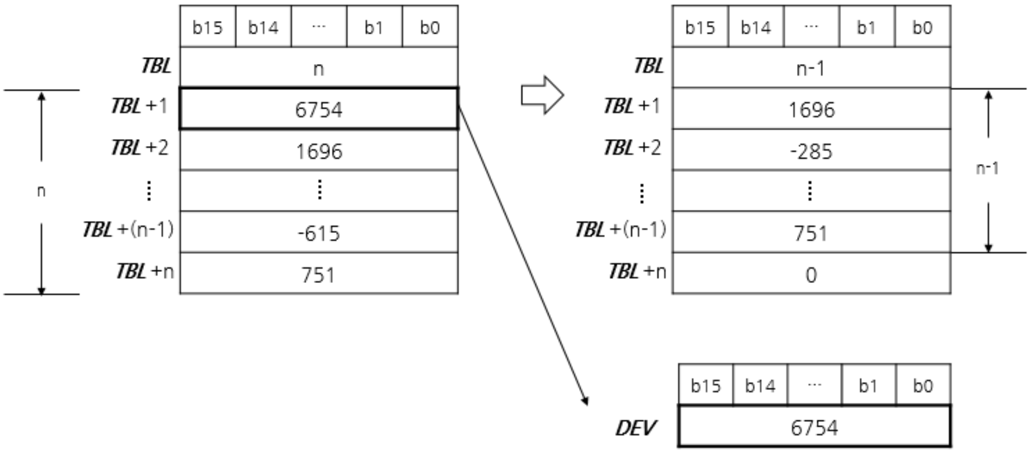
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FIFR(P)	*DEV*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	TBL	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-				



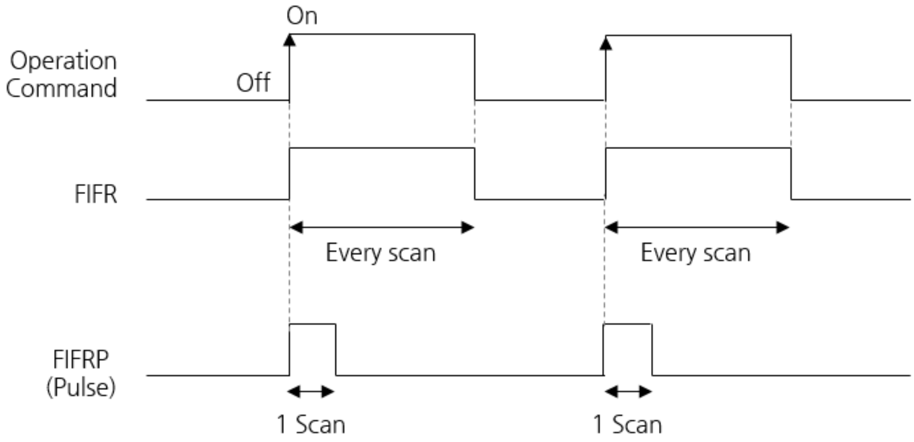
DEV	Address of word device where data read from the table will be stored.
TBL	Head address of word device where the table is located.

FIFR, FIFRP

- FIFR(P) instructions read the oldest data stored in data table starting from the word device assigned to *TBL*. The data read from the *TBL* +1 is stored in the word device assigned to *DEV*.
- After the execution of FIFR(P), the following data blocks shift 1 word in the direction of the least significant device. Then, the value stored in *TBL* decreases by 1.



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)

F110 turns ON for 1 scan when the data table exceeds the range of corresponding device.

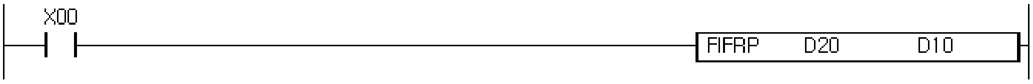
F110 turns ON for 1 scan when there is no data in the data table (When value stored in *TBL* is "0").

Program Example

FIFRP

When X00 turns ON, FIFRP instruction reads the oldest data (D11) in the data table starting from D10. The oldest data (D11) read from the table is stored in D20. Then, the following data blocks are shifted 1 word in the direction of the least significant and the value stored in D10 decreases by 1.

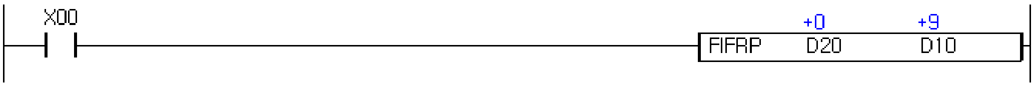
Ladder Diagram (LD)

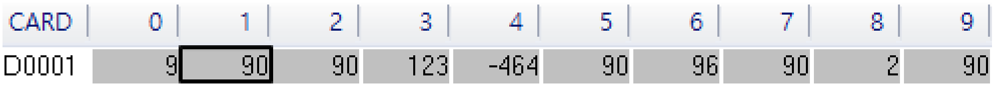


Instruction List (IL)

Instruction	Device	
LD	X00	
FIFRP	D20	D10

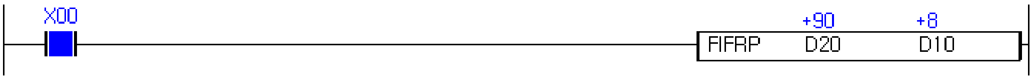
- There are 9 data blocks stored in the data table D10 and the oldest data is "90".

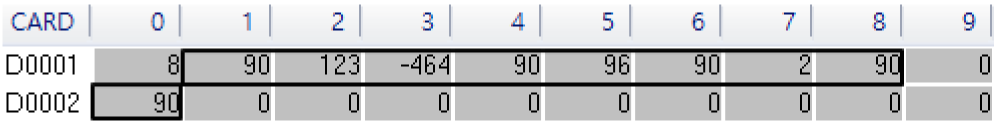




CARD	0	1	2	3	4	5	6	7	8	9
D0001	9	90	90	123	-464	90	96	90	2	90

- When X00 turns ON, FIFRP instruction is executed. Then, the oldest data “90” is stored in D20. The following data blocks (D12~D19) are shifted 1 word in the direction of the least significant device. Then, the value stored in D10 decreases by 1.





CARD	0	1	2	3	4	5	6	7	8	9
D0001	8	90	123	-464	90	96	90	2	90	0
D0002	90	0	0	0	0	0	0	0	0	0

2.8.3 FPOP, FPOPP

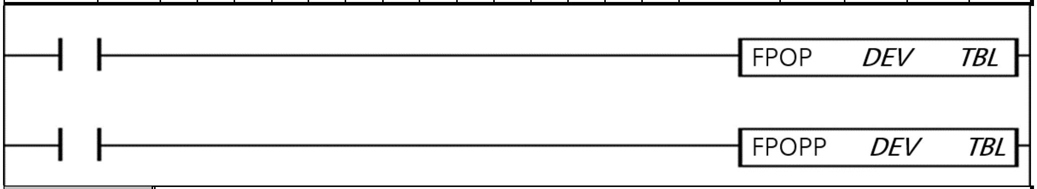
Supported PLC Series

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

FPOP(P) instructions read the latest data from the data table.

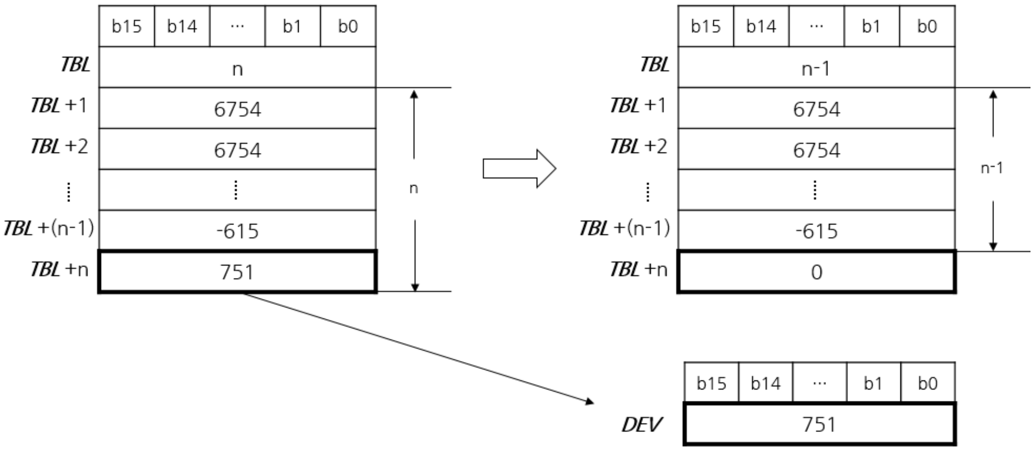
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FPOP(P)	*DEV*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	3	O	-	-
	TBL	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-				



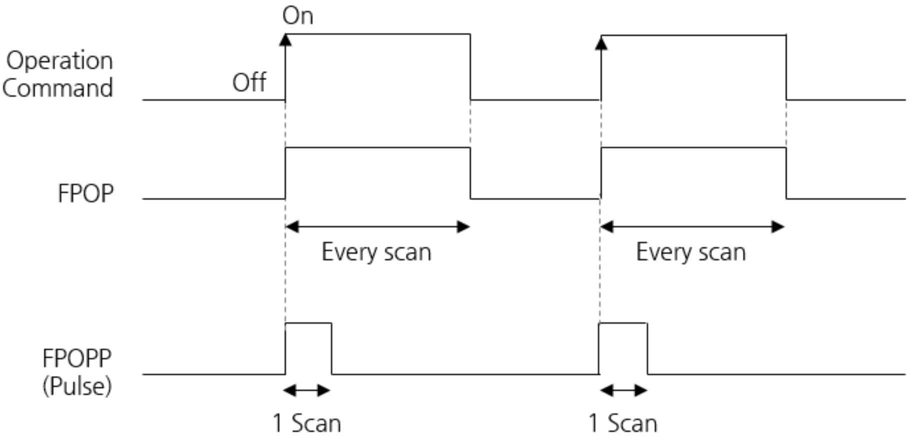
DEV	Address of word device where data read from the table will be stored.
TBL	Head address of word device where the table is located.

FPOP, FPOPP

- FPOP(P) instructions read the latest data stored in data table starting from the word device assigned to *TBL*. The latest data read from the data table is stored in the word device assigned to *DEV*.
- In device assigned to *TBL*, the number of data blocks in the table is stored.
- After the execution of FPOP(P) instruction, “0” is stored in the device where the data is read by the FPOP(P) instruction. Then, the value stored in *TBL* decreases by 1.



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)

F110 turns ON for 1 scan when the data table exceeds the range of corresponding device.

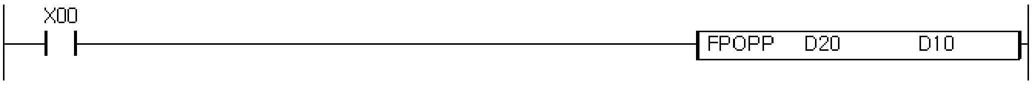
F110 turns ON for 1 scan when there is no data in the data table (When value stored in *TBL* is "0").

Program Example

FPOPP

When X00 turns ON, FPOP instruction reads the latest data in the data table starting from D10. The latest data read from the table is stored in D20. Then, "0" is stored in the device (D19) where latest data was read by the instruction.

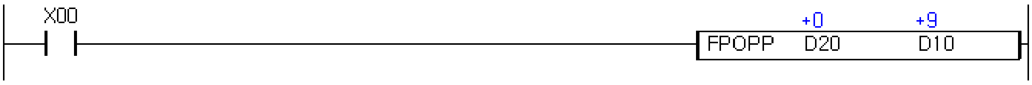
Ladder Diagram (LD)

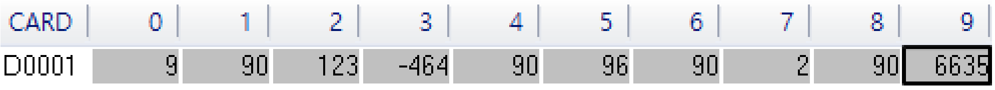


Instruction List (IL)

Instruction	Device	
LD	X00	
FPOPP	D20	D10

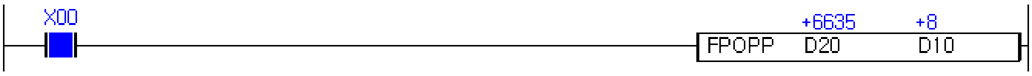
- There are 9 data blocks stored in the data table D10 and the latest data is "6635".

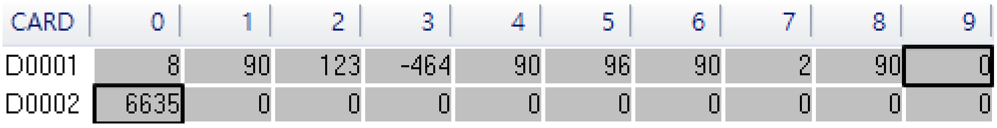




CARD	0	1	2	3	4	5	6	7	8	9
D0001	9	90	123	-464	90	96	90	2	90	6635

- When X00 turns ON, FPOPP instruction is executed. Then, the latest data “6635” is stored in D20. “0” is stored in the table where the latest data existed. Then, the value stored in D10 decreases by 1.





CARD	0	1	2	3	4	5	6	7	8	9
D0001	8	90	123	-464	90	96	90	2	90	0
D0002	6635	0	0	0	0	0	0	0	0	0

2.8.4 FINS, FINSP

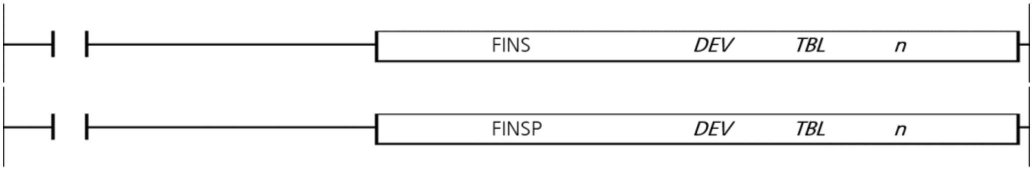
Supported PLC Series

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

FINS(P) instructions insert data into the data table.

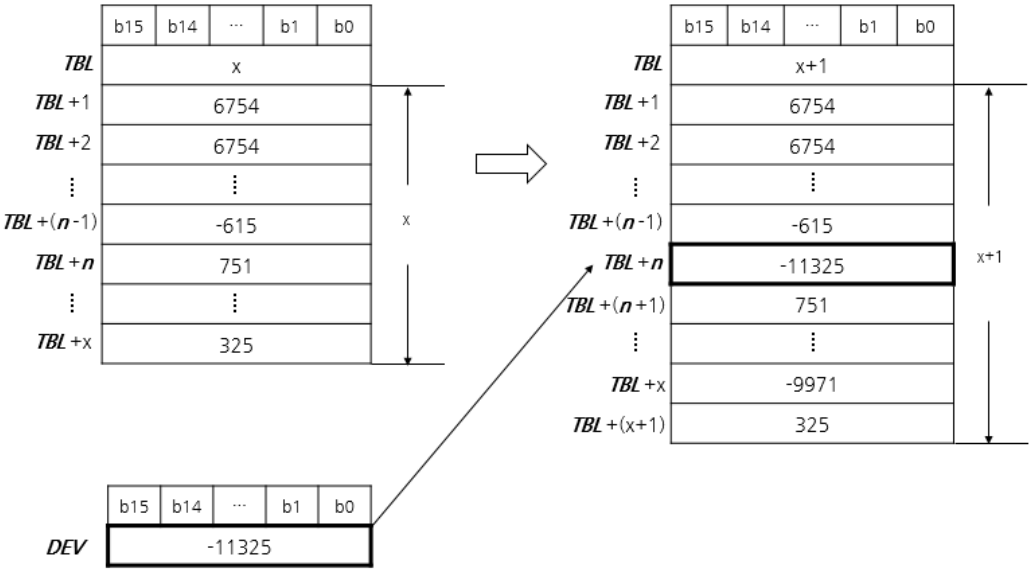
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FINS(P)	*DEV*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	TBL	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



DEV	Address of word device where data to be inserted into the table is stored.
TBL	Head address of word device where the table is located.
n	Location in the table where the data will be inserted. (*n* ≧1)

FINS, FINSP

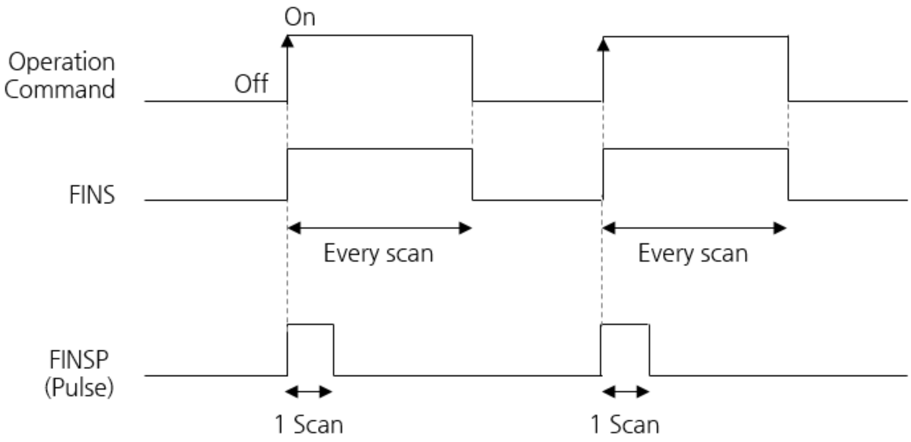
- FINS(P) instructions insert the data stored in word device assigned to *DEV* into the word device *TBL* + *n*. *TBL* + *n* is located in the data table starting from word device *TBL*.



- After the execution of FINS(P), the existing data blocks shift 1 word in the direction of the most significant device. Then, the value stored in *TBL* increases by 1.

- When there is no data in the data table (When value assigned to *TBL* is "0") before the execution of FINS(P) instruction, the instruction can only insert data into the first place (*n* =1).

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)

F110 turns ON for 1 scan when the data table exceeds the range of corresponding device.

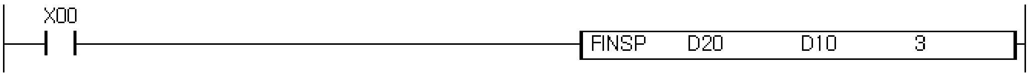
F110 turns ON for 1 scan when there is no data in the data table (When value stored in *TBL* is "0").

Program Example

FINSP

When X00 turns ON, FINSP instruction insert the data stored in D20 into the 3rd device of data table which starts from D10. Then, the existing data blocks are shifted 1 word in the direction of the most significant and the value stored in D10 increases by 1.

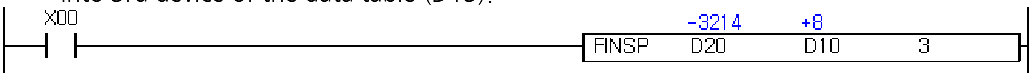
Ladder Diagram (LD)

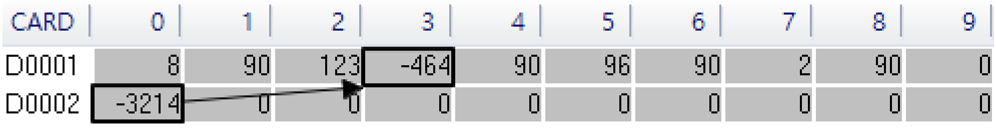


Instruction List (IL)

Instruction	Device		
LD	X00		
FINSP	D20	D10	3

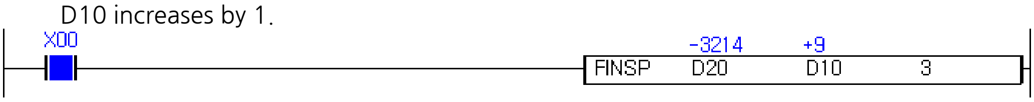
- There are 8 data blocks stored in the data table D10 and the data "-3214" will be inserted into 3rd device of the data table (D13).

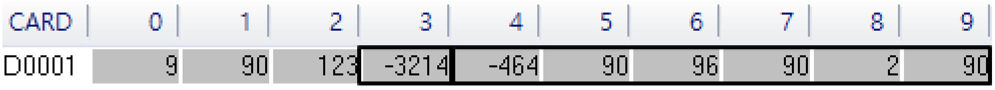




CARD	0	1	2	3	4	5	6	7	8	9
D0001	8	90	123	-464	90	96	90	2	90	0
D0002	-3214	0	0	0	0	0	0	0	0	0

- When X00 turns ON, FINSP instruction is executed. Then, the data “-3214” is inserted into 3rd device located in the data table (D13). The existing data blocks (D13~D18) are shifted 1 word in the direction of the most significant device. Then, the value stored in D10 increases by 1.





CARD	0	1	2	3	4	5	6	7	8	9
D0001	9	90	123	-3214	-464	90	96	90	2	90

2.8.5 FDEL, FDELP

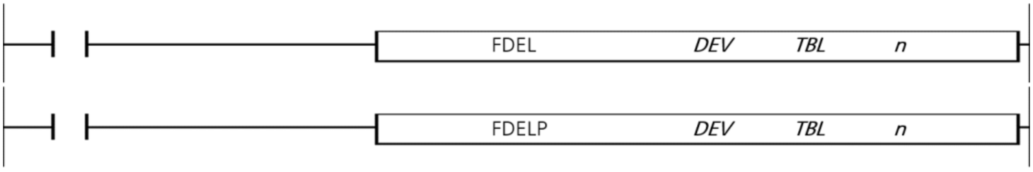
Supported PLC Series

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

FDEL(P) instructions delete data from the data table.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FDEL(P)	*DEV*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	TBL	-	-	-	-	-	-	-	-	-	O	O	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



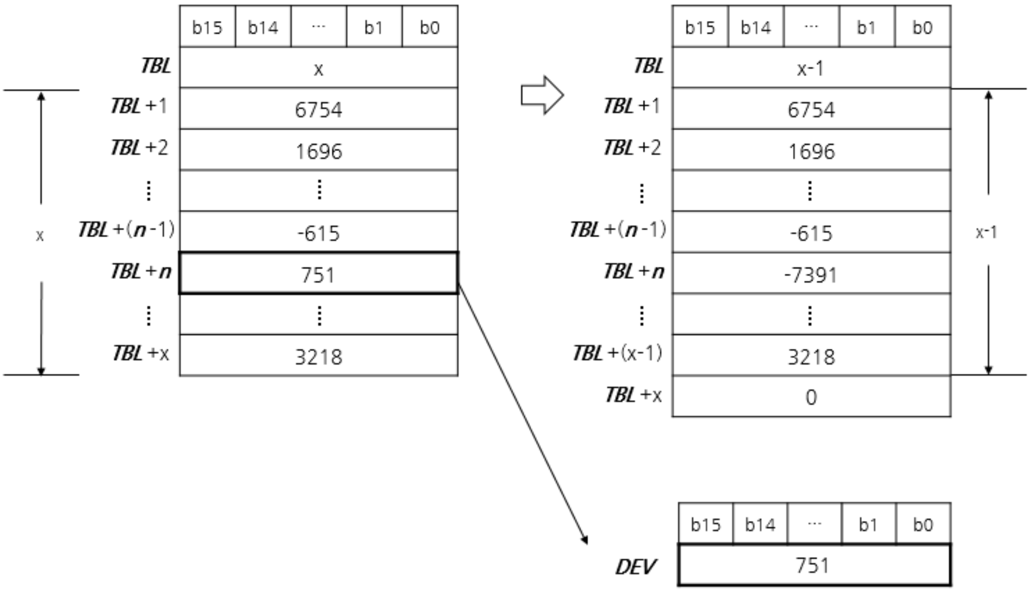
DEV	Address of word device where data deleted from the table will be stored.
TBL	Head address of word device where the table is located.
n	Location in the table where the data will be deleted. (*n* ≧1)

FDEL, FDELP

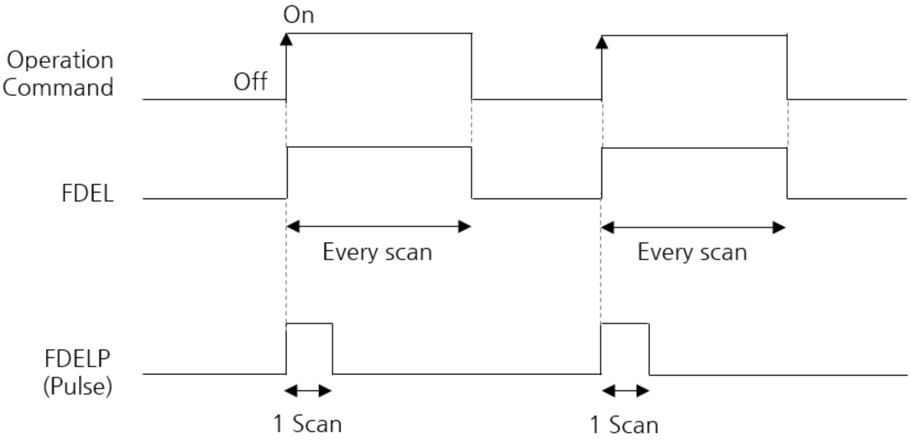
- FDEL(P) instructions delete the data from the word device *TBL* + *n*. *TBL* + *n* is located in the data table starting from word device *TBL*. Deleted data is stored in word device assigned to *DEV*.

- After the execution of FDEL(P), the existing data blocks shift 1 word in the direction of the least significant device. Then, the value stored in *TBL* decreases by 1.

- The range of value assigned to *n* is as following:
  1 ≦ Value assigned to *n* ≦ (Value assigned to *TBL*)+1



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)

F110 turns ON for 1 scan when the data table exceeds the range of corresponding device.

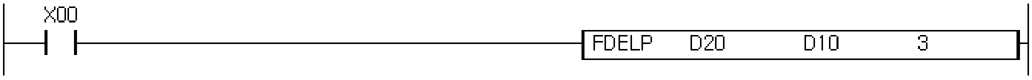
F110 turns ON for 1 scan when there is no data in the data table (When value stored in *TBL* is "0").

Program Example

FDELP

When X00 turns ON, FDELP instruction delete the data stored in the 3rd device of data table which starts from D10. Then, the existing data blocks are shifted 1 word in the direction of the least significant and the value stored in D10 decreases by 1. Deleted data is stored in D20.

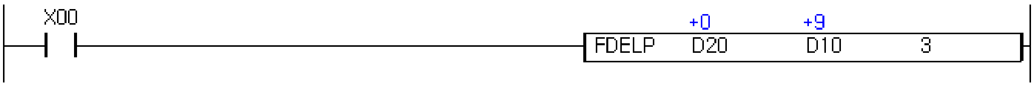
Ladder Diagram (LD)

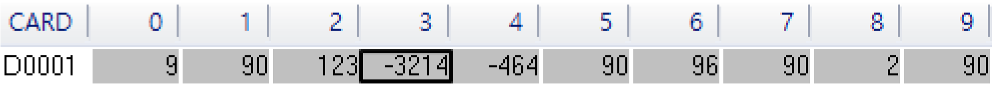


Instruction List (IL)

Instruction	Device		
LD	X00		
FDELP	D20	D10	3

- There are 9 data blocks stored in the data table D10. The data stored in 3rd device of the data table (D13, "-3214") will be deleted.

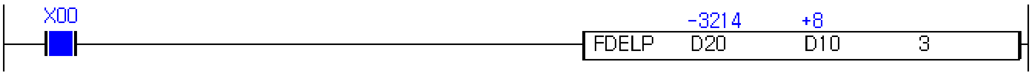


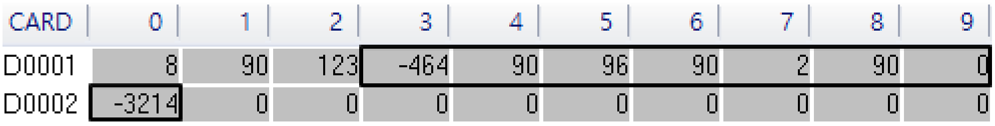


CARD	0	1	2	3	4	5	6	7	8	9
D0001	9	90	123	-3214	-464	90	96	90	2	90

- When X00 turns ON, FDELP instruction is executed. Then, the data "-3214" which is stored in 3rd device located in the data table (D13) is deleted. The following data blocks (D14~D19) are shifted 1 word in the direction of the least significant device. Then, the value stored in D10 decreases by 1.

- Deleted data “-3214” is stored in D20.





CARD	0	1	2	3	4	5	6	7	8	9
D0001	8	90	123	-464	90	96	90	2	90	0
D0002	-3214	0	0	0	0	0	0	0	0	0

2.9 Logical Operation Instructions

2.9.1 WAND, WANDP, DAND, DANDP

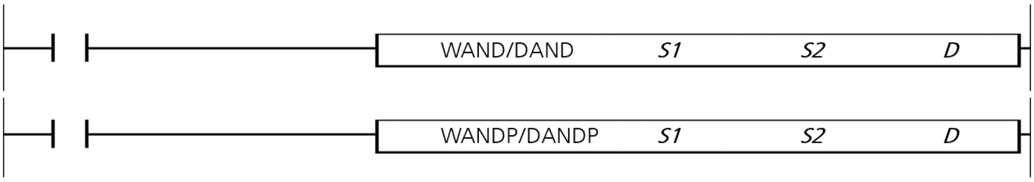
Supported PLC Series

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

WAND(P) instructions perform logical AND operation between word device *S1* and word device *S2*.
DAND(P) instructions perform logical AND operation between double word device *S1* and double word device *S2*.
The result is stored in word/double word device *D*.

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
WAND(P) DAND(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	O*	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				

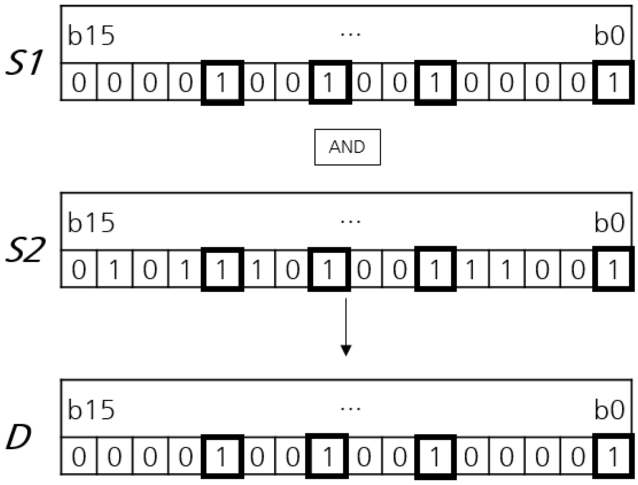


S1	Constant or head address of word/double word device where data for logical AND operation is stored.
S2	Constant or head address of word/double word device where data for logical AND operation is stored.
D	Head address of word/double word device where operation result will be stored.

* Zero flag turns ON only when WAND(P) instruction is operated in CP3A/B/P/U and CP4A~D/U.

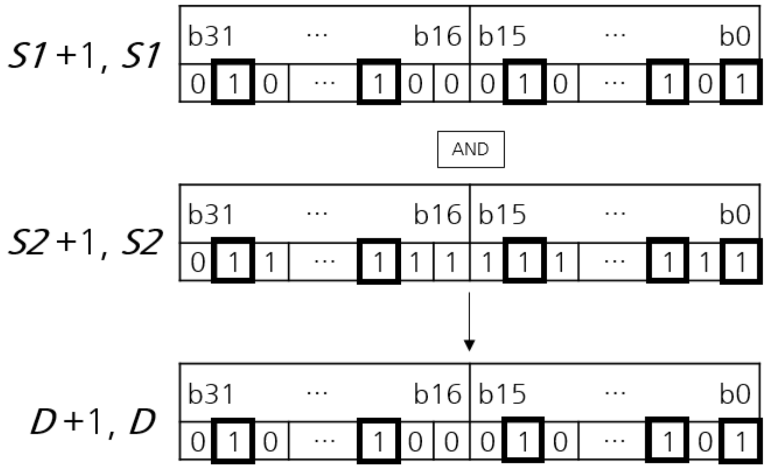
WAND, WANDP

- WAND(P) instructions perform logical AND operation between word device *S1* and word device *S2*.
- Logical AND operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical AND operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits in device *S1* and *S2* are set, corresponding bits in device *D* are set. In another case, bits in device *D* are not set.

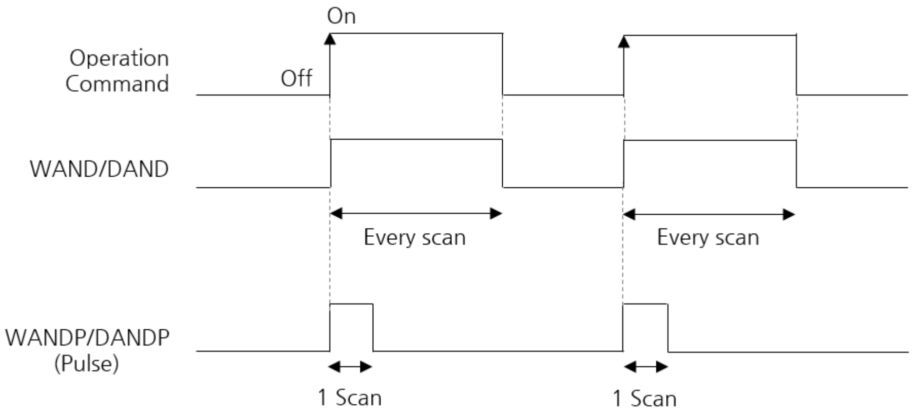


DAND, DANDP

- DAND(P) instructions perform logical AND operation between double word device *S1* and double word device *S2*.
- Logical AND operation is performed between 32-bit data assigned to *S1* and 32-bit data assigned to *S2*.
- Logical AND operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits in device *S1* and *S2* are set, corresponding bits in device *D* are set. In another case, bits in device *D* are not set.



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)

Zero flag (F111)

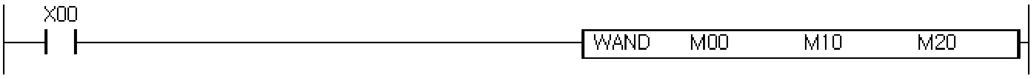
F111 turns ON for 1 scan when the operation result is “0”.
(WAND(P) instruction operated in CP3A/B/P/U and CP4A~D/U only)

Program Example

WAND

When operation command X00 is ON, WAND instruction is executed. The word data assigned to word device M00 (M00 ~ M0F) and M10 (M10 ~ M1F) conduct logical AND operation and the result is stored in M20 (M20 ~ M2F).

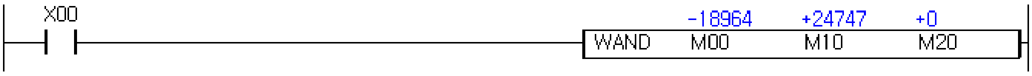
Ladder Diagram (LD)



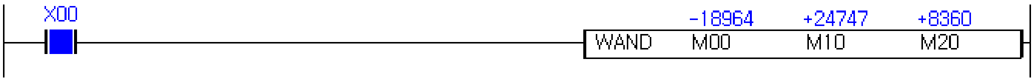
Instruction List (IL)

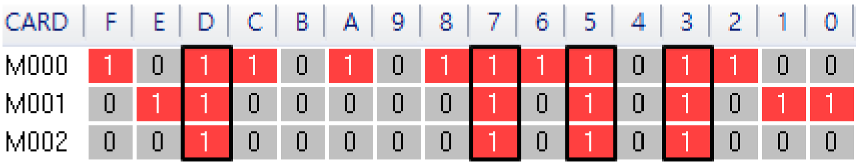
Instruction	Device		
LD	X00		
WAND	M00	M10	M20

- The value assigned to M00 (M00 ~ M0F) is "-18964" (= 1011 0101 1110 1100). The value assigned to M10 (M10 ~ M1F) is "24747" (= 0110 0000 1010 1011).



- When X00 turns ON, WAND instruction is executed. Then, the logical AND operation is conducted on M00 (M00 ~ M0F) and M10 (M10 ~ M1F). The result is stored in M20 (M20 ~ M2F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	1	1	0	1	0	1	1	1	1	0	1	1	0	0
M001	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1
M002	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0

2.9.2 WOR, WORP, DOR, DORP

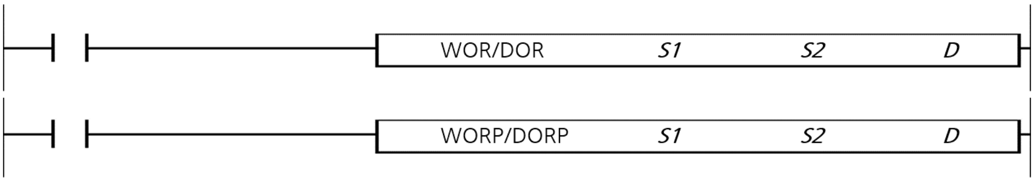
Supported PLC Series

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

WOR(P) instructions perform logical OR operation between word device *S1* and word device *S2*.
DOR(P) instructions perform logical OR operation between double word device *S1* and double word device *S2*.
The result is stored in word/double word device *D*.

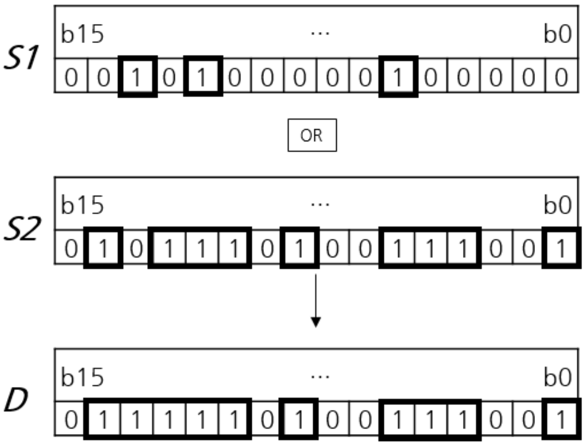
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
WOR(P) DOR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



S1	Constant or head address of word/double word device where data for logical OR operation is stored.
S2	Constant or head address of word/double word device where data for logical OR operation is stored.
D	Head address of word/double word device where operation result will be stored.

WOR, WORP

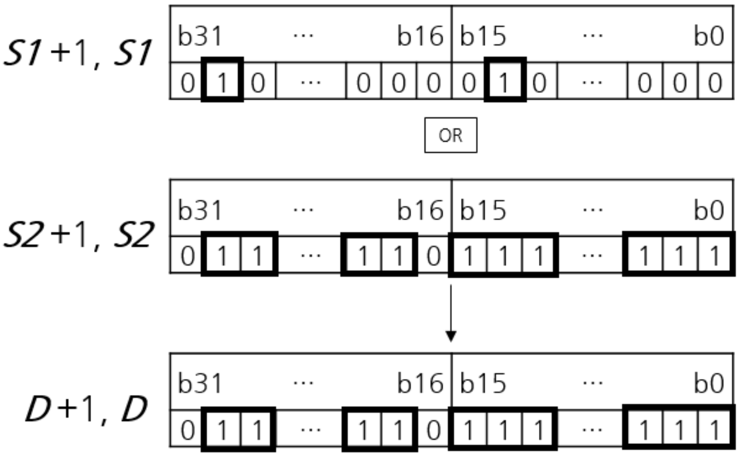
- WOR(P) instructions perform logical OR operation between word device *S1* and word device *S2*.
- Logical OR operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical OR operation is conducted on each bit of device *S1* and *S2*.
- If one of bits in device *S1* and *S2* is set, corresponding bit in device *D* is set. In another case, bits in device *D* are not set.



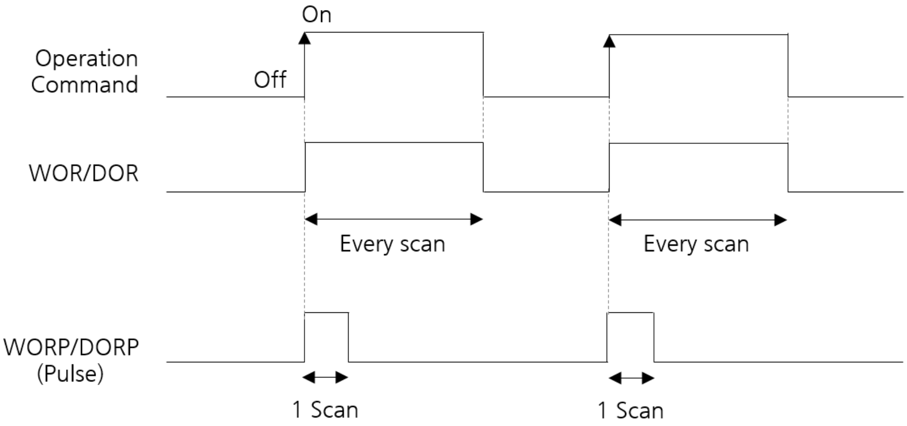
	b15							···								b0
S1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
S2	0	1	0	1	1	1	0	1	0	0	1	1	1	0	0	1
D	0	1	1	1	1	1	0	1	0	0	1	1	1	0	0	1

DOR, DORP

- DOR(P) instructions perform logical OR operation between double word device *S1* and double word device *S2*.
- Logical OR operation is performed between 32-bit data assigned to *S1* and 32-bit data assigned to *S2*.
- Logical OR operation is conducted on each bit of device *S1* and *S2*.
- If one of bits in device *S1* and *S2* is set, corresponding bit in device *D* is set. In another case, bits in device *D* are not set.



Execution Condition



Operation Error

Error Flag (F110)

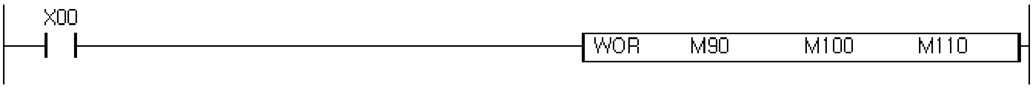
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

WOR

When operation command X00 is ON, WOR instruction is executed. The word data assigned to word device M90 (M90 ~ M9F) and M100 (M100 ~ M10F) conduct logical OR operation and the result is stored in M110 (M110 ~ M11F).

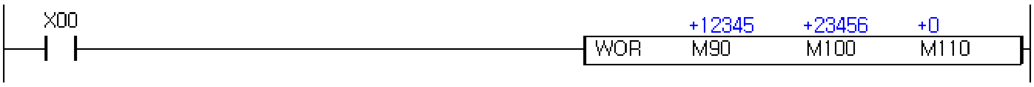
Ladder Diagram (LD)



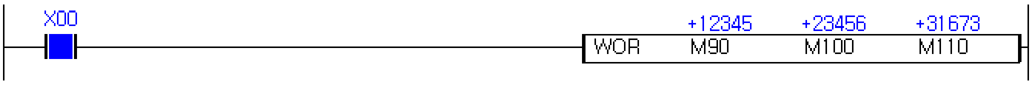
Instruction List (IL)

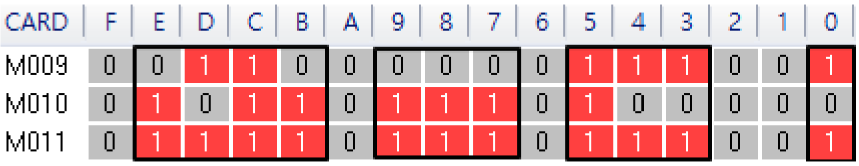
Instruction	Device		
LD	X00		
WOR	M90	M100	M110

- The value assigned to M90 (M90 ~ M9F) is “12345” (= 0011 0000 0011 1001). The value assigned to M100 (M100 ~ M10F) is “23456” (= 0101 1011 1010 0000).



- When X00 turns ON, WOR instruction is executed. Then, the logical OR operation is conducted on M90 (M90 ~ M9F) and M100 (M100 ~ M10F). The result is stored in M110 (M110 ~ M11F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M009	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1
M010	0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0
M011	0	1	1	1	1	0	1	1	1	0	1	1	1	0	0	1

2.9.3 WXOR, WXORP, DXOR, DXORP

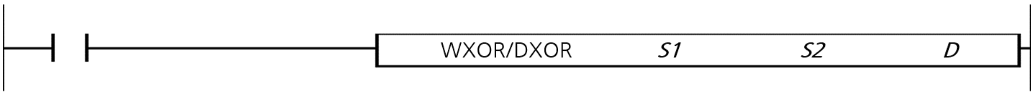
Supported PLC Series

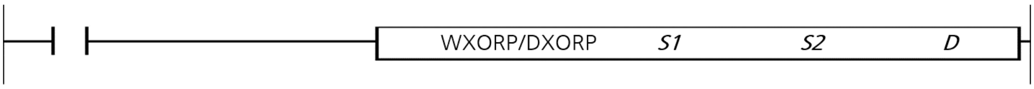
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

WXOR(P) instructions perform logical XOR operation between word device *S1* and word device *S2*.
DXOR(P) instructions perform logical XOR operation between double word device *S1* and double word device *S2*.
The result is stored in word/double word device *D*.

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
WXOR(P) DXOR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	O*	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				



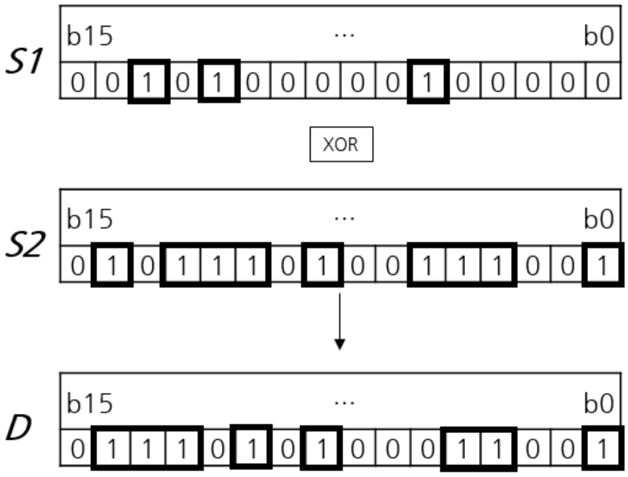


S1	Constant or head address of word/double word device where data for logical XOR operation is stored.
S2	Constant or head address of word/double word device where data for logical XOR operation is stored.
D	Head address of word/double word device where operation result will be stored.

* Zero flag turns ON only when WXOR(P) instruction is operated in CP3A/B/P/U and CP4A~D/U.

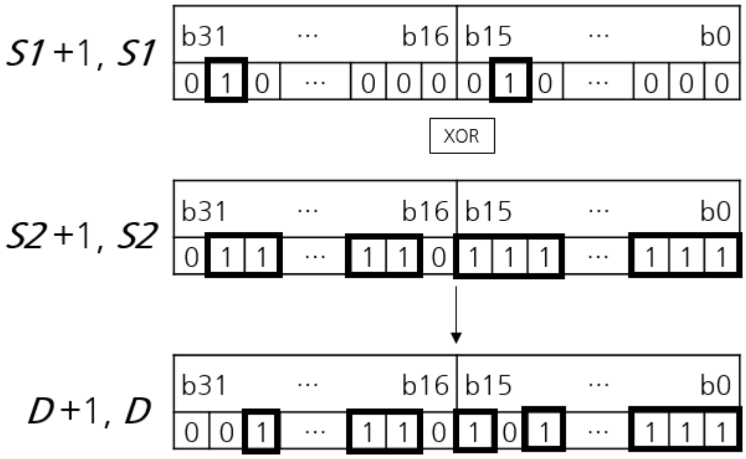
WXOR, WXORP

- WXOR(P) instructions perform logical XOR operation between word device *S1* and word device *S2*.
- Logical XOR operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical XOR operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits are in same status in device *S1* and *S2*, corresponding bits in device *D* are not set.
- If corresponding bits are not in same status in device *S1* and *S2*, corresponding bits in device *D* are set.

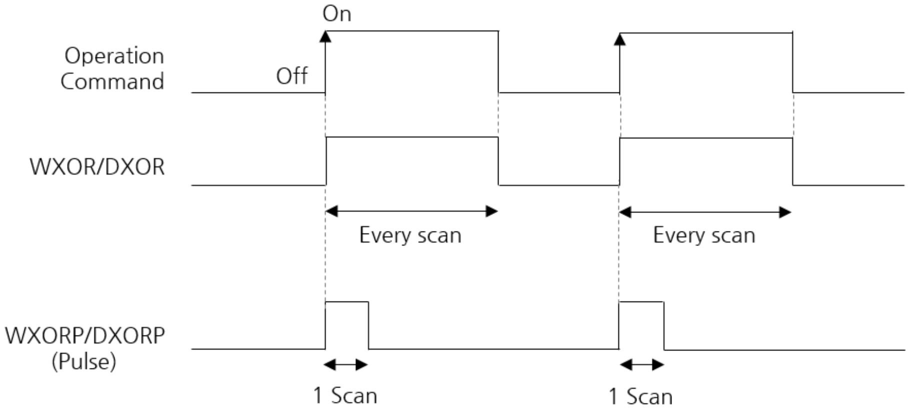


DXOR, DXORP

- DXOR(P) instructions perform logical XOR operation between double word device *S1* and double word device *S2*.
- Logical XOR operation is performed between 32-bit data assigned to *S1* and 32-bit data assigned to *S2*.
- If corresponding bits are in same status in device *S1* and *S2*, corresponding bits in device *D* are not set.
- If corresponding bits are not in same status in device *S1* and *S2*, corresponding bits in device *D* are set.



Execution Condition



Operation Error

Error Flag (F110)

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

Zero flag (F111)

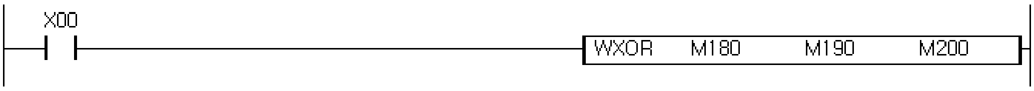
F111 turns ON for 1 scan when the operation result is “0”.
(WXOR(P) instruction operated in CP3A/B/P/U and CP4A~D/U only)

Program Example

WXOR

When operation command X00 is ON, WXOR instruction is executed. The word data assigned to word device M180 (M180 ~ M18F) and M190 (M190 ~ M19F) conduct logical XOR operation and the result is stored in M200 (M200 ~ M20F).

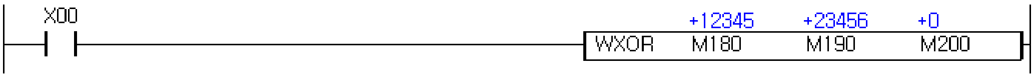
Ladder Diagram (LD)



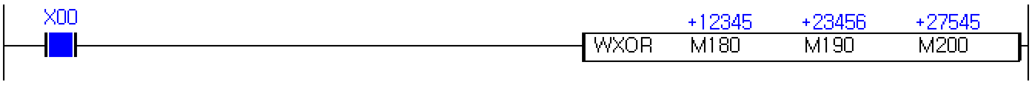
Instruction List (IL)

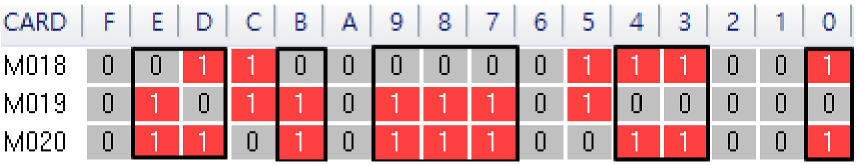
Instruction	Device		
LD	X00		
WXOR	M180	M190	M200

- The value assigned to M180 (M180 ~ M18F) is “12345” (= 0011 0000 0011 1001). The value assigned to M190 (M190 ~ M19F) is “23456” (= 0101 1011 1010 0000).



- When X00 turns ON, WXOR instruction is executed. Then, the logical XOR operation is conducted on M180 (M180 ~ M18F) and M190 (M190 ~ M19F). The result is stored in M200 (M200 ~ M20F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M018	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1
M019	0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0
M020	0	1	1	0	1	0	1	1	1	0	0	1	1	0	0	1

2.9.4 BKAND, BKANDP

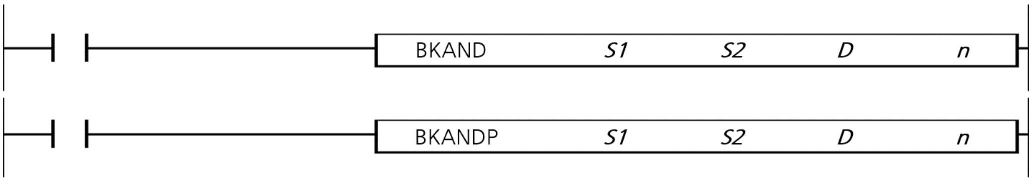
Supported PLC Series

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

BKAND(P) instructions perform logical AND operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
The result is stored in *n* word devices starting from word device *D*.

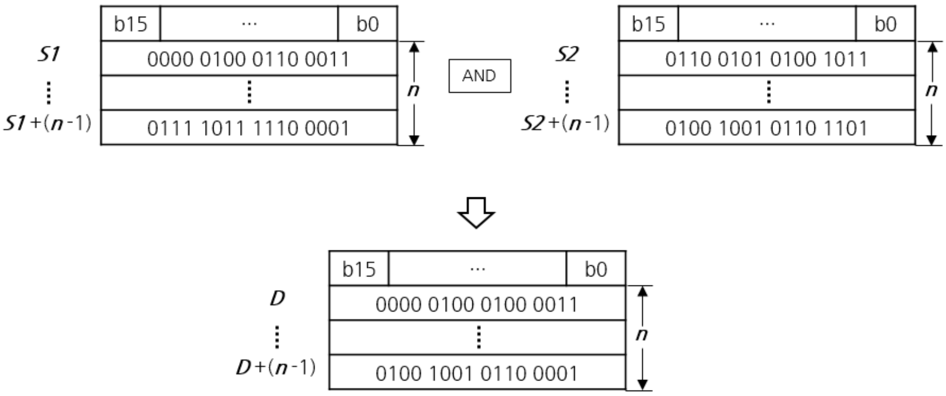
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BKAND(P)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	5	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



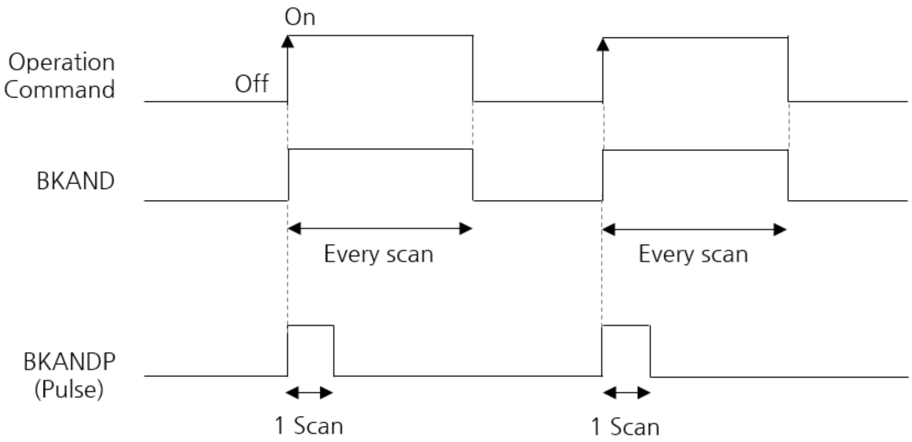
S1	Head address of word devices where data for logical AND operation is stored.
S2	Constant or head address of word devices where data for logical AND operation is stored.
D	Head address of word devices where operation result will be stored.
n	The number of word data blocks to perform logical AND operation.

BKAND, BKANDP

- BKAND(P) instructions perform logical AND operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
- Logical AND operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical AND operation is conducted on each bit of device *S1* and *S2*.
- If both bits in device *S1* and *S2* are set, corresponding bits in device *D* are set. In another case, bits in device *D* are not set.



Execution Condition



Operation Error

Error flag (F110)

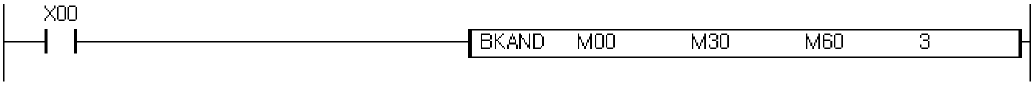
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

BKAND

When operation command X00 is ON, BKAND instruction is executed. Then, 3 word data blocks starting from word device M00 (M00 ~ M2F) and 3 word data blocks starting from M30 (M30 ~ M5F) conduct logical AND operation. The result is stored in 3 word devices starting from M60 (M60 ~ M8F).

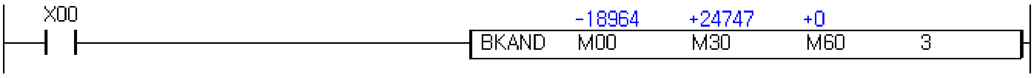
Ladder Diagram (LD)



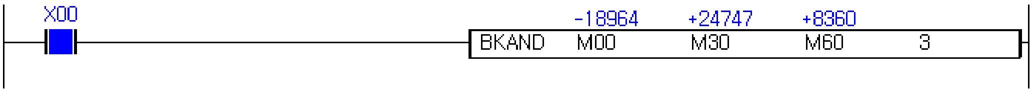
Instruction List (IL)

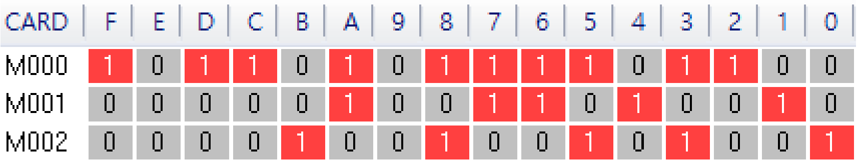
Instruction	Device			
LD	X00			
BKAND	M00	M30	M60	3

- Each value assigned to 3 word devices starting from M00 (M00 ~ M2F) is “-18964” (= 1011 0101 1110 1100), “1234” (= 0000 0100 1101 0010) and “2345” (= 0000 1001 0010 1001).
- Each value assigned to 3 word devices starting from M30 (M30 ~ M5F) is “24747” (= 0110 0000 1010 1011), “4050” (= 0000 1111 1101 0010) and “6625” (= 0001 1001 1110 0001).



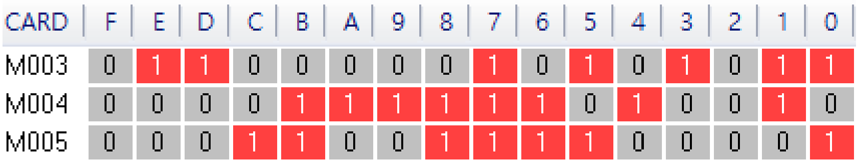
- When X00 turns ON, BKAND instruction is executed. Then, the logical AND operation is conducted on 3 word devices starting from M00 (M00 ~ M2F) and 3 word devices starting from M30 (M30 ~ M5F). The result is stored in 3 word devices starting from M60 (M60 ~ M8F).





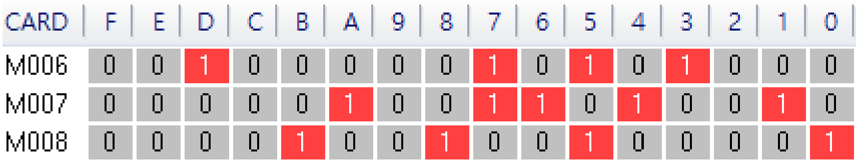
CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	1	1	0	1	0	1	1	1	1	0	1	1	0	0
M001	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0
M002	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

AND



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M003	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1
M004	0	0	0	0	1	1	1	1	1	1	0	1	0	0	1	0
M005	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1

⇩



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M006	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
M007	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0
M008	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1

2.9.5 BKOR, BKORP

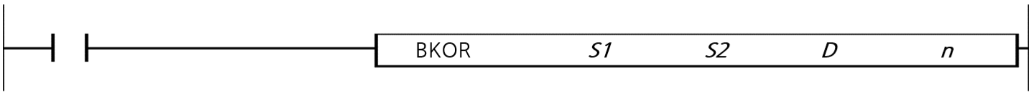
Supported PLC Series

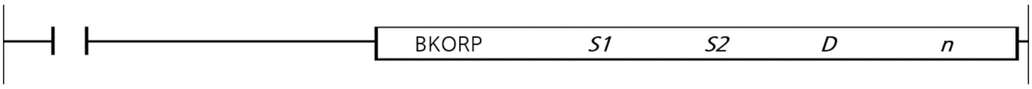
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

BKOR(P) instructions perform logical OR operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
The result is stored in *n* word devices starting from word device *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
BKOR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	5	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

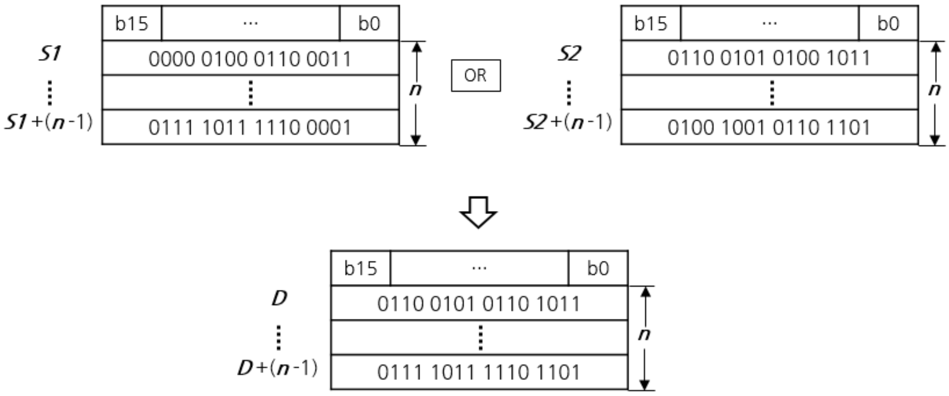




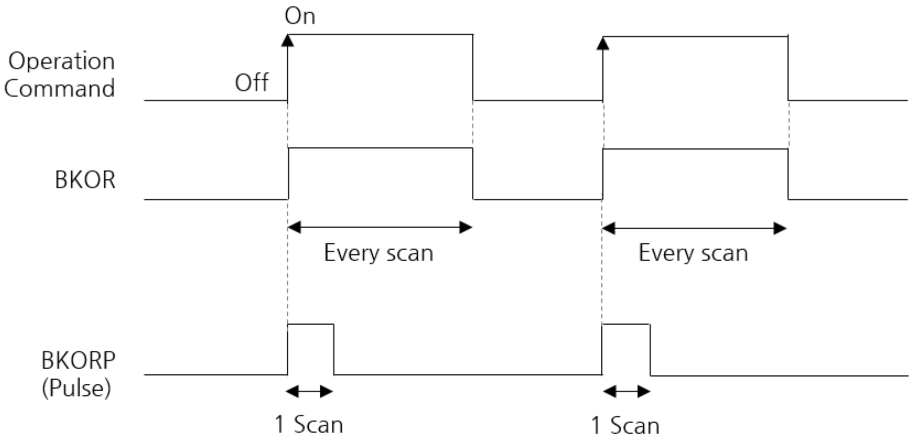
S1	Head address of word devices where data for logical OR operation is stored.
S2	Constant or head address of word devices where data for logical OR operation is stored.
D	Head address of word devices where operation result will be stored.
n	The number of word data blocks to perform logical OR operation.

BKOR, BKORP

- BKOR(P) instructions perform logical OR operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
- Logical OR operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical OR operation is conducted on each bit of device *S1* and *S2*.
- If one of bits is set in device *S1* and *S2*, corresponding bits in device *D* are set. In another case, bits in device *D* are not set.



Execution Condition



Operation Error

Error Flag (F110)

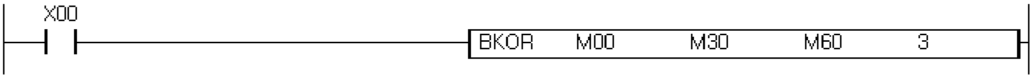
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

BKOR

When operation command X00 is ON, BKOR instruction is executed. Then, 3 word data blocks starting from word device M00 (M00 ~ M2F) and 3 word data blocks starting from M30 (M30 ~ M5F) conduct logical OR operation. The result is stored in 3 word devices starting from M60 (M60 ~ M8F).

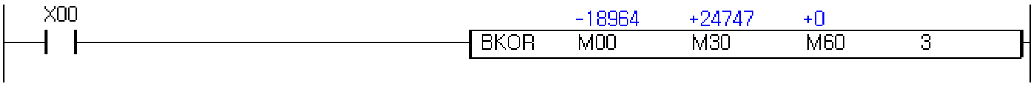
Ladder Diagram (LD)



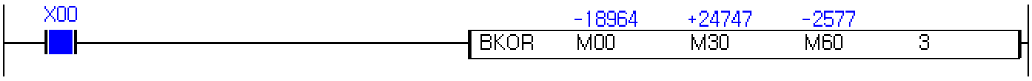
Instruction List (IL)

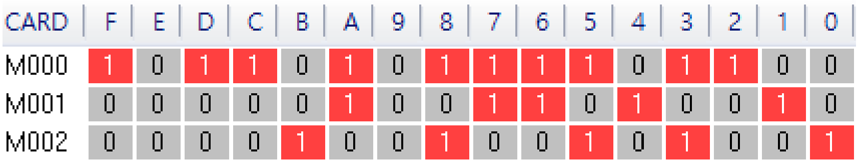
Instruction	Device			
LD	X00			
BKOR	M00	M30	M60	3

- Each value assigned to 3 word devices starting from M00 (M00 ~ M2F) is "-18964" (= 1011 0101 1110 1100), "1234" (= 0000 0100 1101 0010) and "2345" (= 0000 1001 0010 1001).
- Each value assigned to 3 word devices starting from M30 (M30 ~ M5F) is "24747" (= 0110 0000 1010 1011), "4050" (= 0000 1111 1101 0010) and "6625" (= 0001 1001 1110 0001).



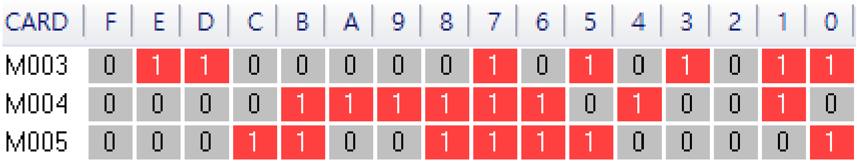
- When X00 turns ON, BKOR instruction is executed. Then, the logical OR operation is conducted on 3 word devices starting from M00 (M00 ~ M2F) and 3 word devices starting from M30 (M30 ~ M5F). The result is stored in 3 word devices starting from M60 (M60 ~ M8F).





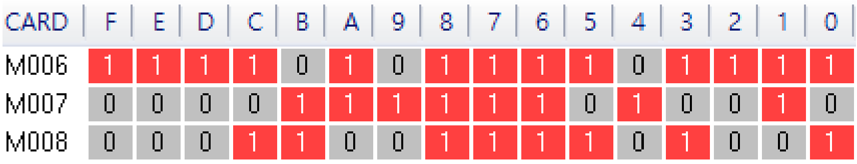
CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	1	1	0	1	0	1	1	1	1	0	1	1	0	0
M001	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0
M002	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

OR



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M003	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1
M004	0	0	0	0	1	1	1	1	1	1	0	1	0	0	1	0
M005	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1

⇩



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M006	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1
M007	0	0	0	0	1	1	1	1	1	1	0	1	0	0	1	0
M008	0	0	0	1	1	0	0	1	1	1	1	0	1	0	0	1

2.9.6 BKXOR, BKXORP

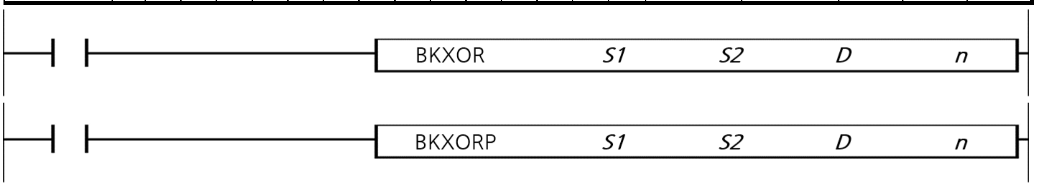
Supported PLC Series

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

BKXOR(P) instructions perform logical XOR operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
The result is stored in *n* word devices starting from word device *D*.

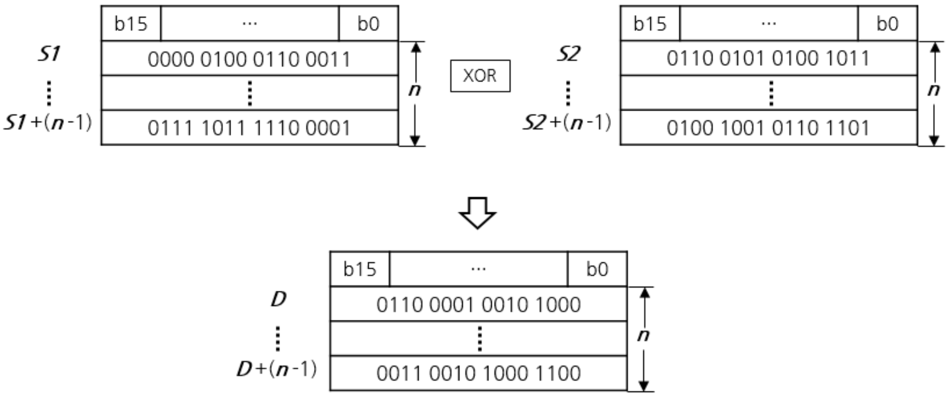
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BKXOR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	5	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



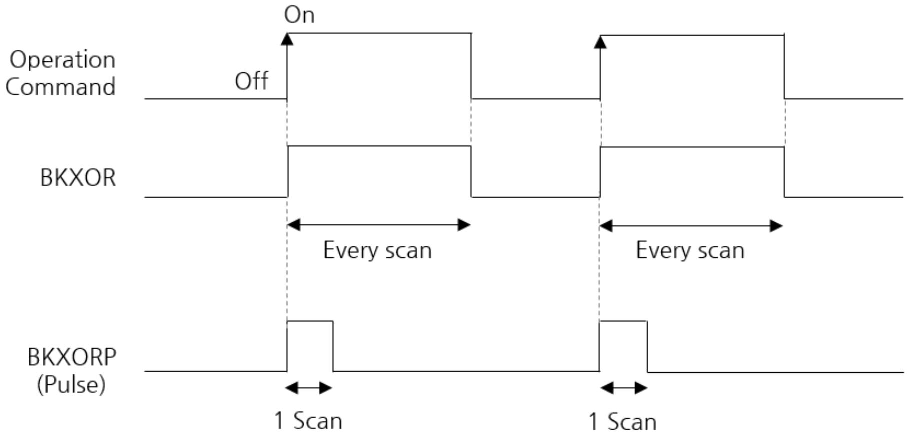
S1	Head address of word devices where data for logical XOR operation is stored.
S2	Constant or head address of word devices where data for logical XOR operation is stored.
D	Head address of word devices where operation result will be stored.
n	The number of word data blocks to perform logical XOR operation.

BKXOR, BKXORP

- BKXOR(P) instructions perform logical XOR operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
- Logical XOR operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical XOR operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits are in same status in device *S1* and *S2*, corresponding bits in device *D* are not set.
- If corresponding bits are not in same status in device *S1* and *S2*, corresponding bits in device *D* are set.



Execution Condition



Operation Error

Error Flag (F110)

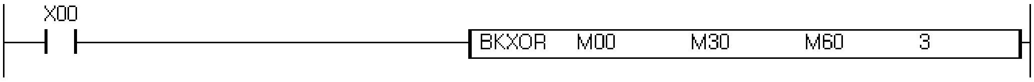
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

BKXOR

When operation command X00 is ON, BKXOR instruction is executed. Then, 3 word data blocks starting from word device M00 (M00 ~ M2F) and 3 word data blocks starting from M30 (M30 ~ M5F) conduct logical XOR operation. The result is stored in 3 word devices starting from M60 (M60 ~ M8F).

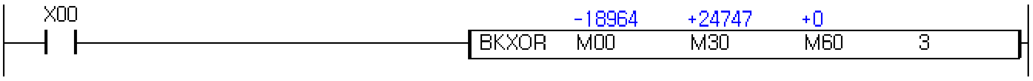
Ladder Diagram (LD)



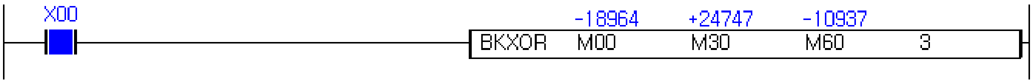
Instruction List (IL)

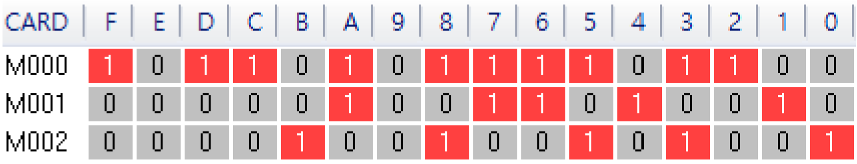
Instruction	Device			
LD	X00			
BKXOR	M00	M30	M60	3

- Each value assigned to 3 word devices starting from M00 (M00 ~ M2F) is “-18964” (= 1011 0101 1110 1100), “1234” (= 0000 0100 1101 0010) and “2345” (= 0000 1001 0010 1001).
- Each value assigned to 3 word devices starting from M30 (M30 ~ M5F) is “24747” (= 0110 0000 1010 1011), “4050” (= 0000 1111 1101 0010) and “6625” (= 0001 1001 1110 0001).



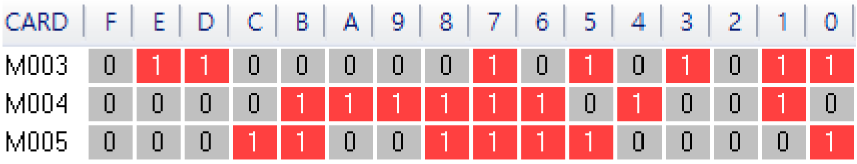
- When X00 turns ON, BKXOR instruction is executed. Then, the logical XOR operation is conducted on 3 word devices starting from M00 (M00 ~ M2F) and 3 word devices starting from M30 (M30 ~ M5F). The result is stored in 3 word devices starting from M60 (M60 ~ M8F).





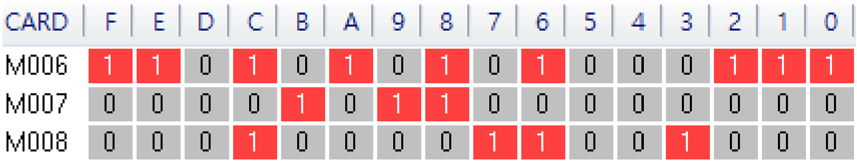
CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	1	1	0	1	0	1	1	1	1	0	1	1	0	0
M001	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0
M002	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

XOR



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M003	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1
M004	0	0	0	0	1	1	1	1	1	1	0	1	0	0	1	0
M005	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1

⇩



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M006	1	1	0	1	0	1	0	1	0	1	0	0	0	1	1	1
M007	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
M008	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0

2.9.7 WXNR, WXNRP, DXNR, DXNRP

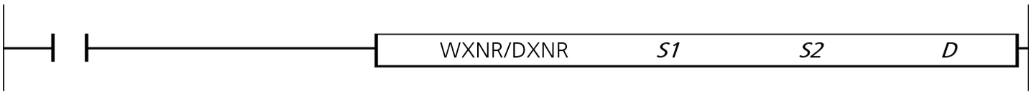
Supported PLC Series

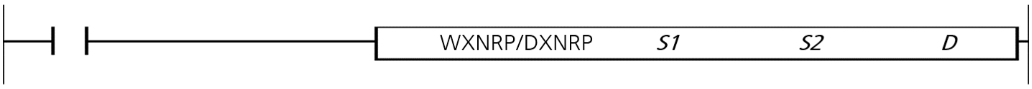
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

WXNR(P) instructions perform logical XNOR operation between word device *S1* and word device *S2*.
DXNR(P) instructions perform logical XNOR operation between double word device *S1* and double word device *S2*.
The result is stored in word/double word device *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
WXNR(P) DXNR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-				

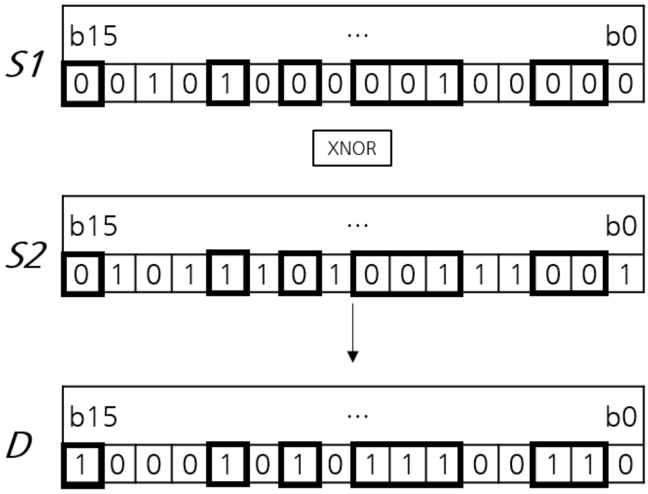




S1	Constant or head address of word/double word device where data for logical XNOR operation is stored.
S2	Constant or head address of word/double word device where data for logical XNOR operation is stored.
D	Head address of word/double word device where operation result will be stored.

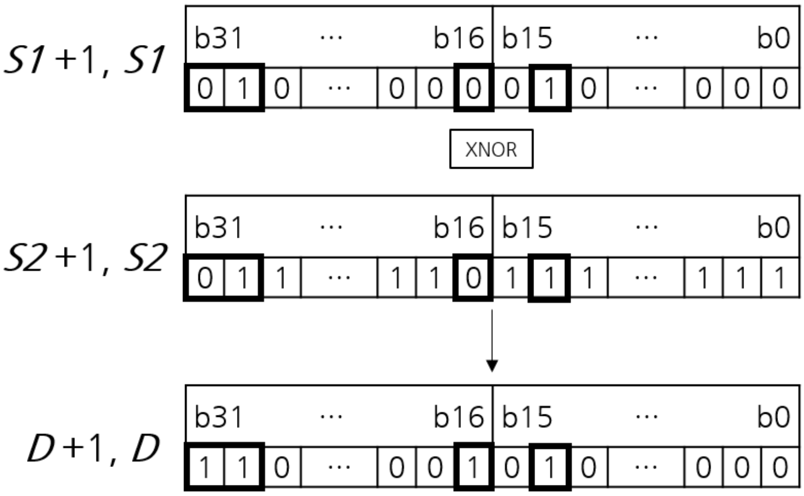
WXNR, WXNRP

- WXNR(P) instructions perform logical XNOR operation between word device *S1* and word device *S2*.
- Logical XNOR operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical XNOR operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits are in same status in device *S1* and *S2*, corresponding bits in device *D* are set.
- If corresponding bits are not in same status in device *S1* and *S2*, corresponding bits in device *D* are not set.

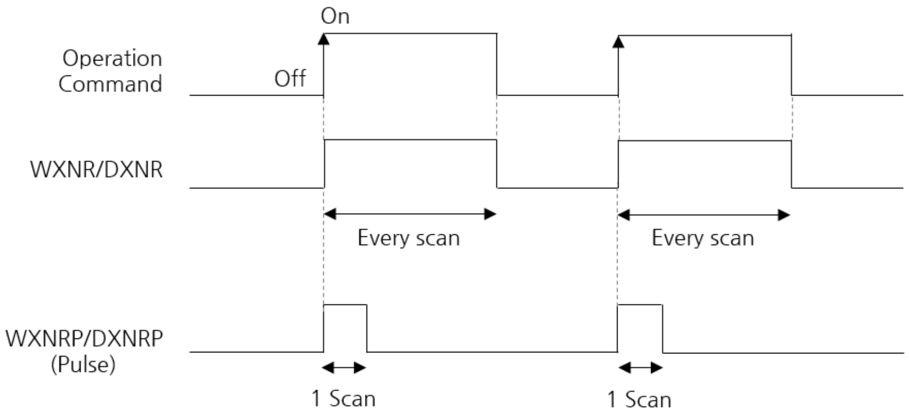


DXNR, DXNRP

- DXNR(P) instructions perform logical XNOR operation between double word device *S1* and double word device *S2*.
- Logical XNOR operation is performed between 32-bit data assigned to *S1* and 32-bit data assigned to *S2*.
- Logical XNOR operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits are in same status in device *S1* and *S2*, corresponding bits in device *D* are set.
- If corresponding bits are not in same status in device *S1* and *S2*, corresponding bits in device *D* are not set.



Execution Condition



Operation Error

Error Flag (F110)

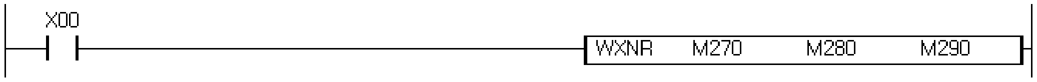
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

WXNR

When operation command X00 is ON, WXNR instruction is executed. The word data assigned to word device M270 (M270 ~ M27F) and M280 (M280 ~ M28F) conduct logical XNOR operation and the result is stored in M290 (M290 ~ M29F).

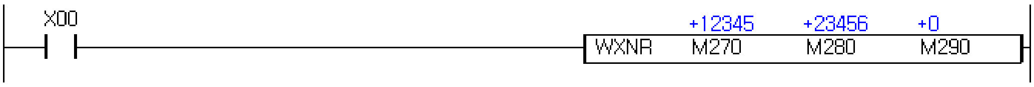
Ladder Diagram (LD)



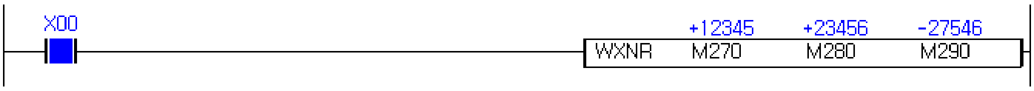
Instruction List (IL)

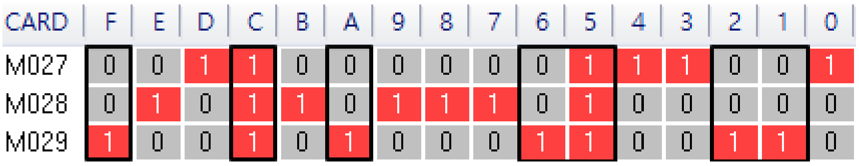
Instruction	Device		
LD	X00		
WXNR	M270	M280	M290

- The value assigned to M270 (M270 ~ M27F) is “12345” (= 0011 0000 0011 1001). The value assigned to M280 (M280 ~ M28F) is “23456” (= 0101 1011 1010 0000).



- When X00 turns ON, WXNR instruction is executed. Then, the logical XNOR operation is conducted on M270 (M270 ~ M27F) and M280 (M280 ~ M28F). The result is stored in M290 (M290 ~ M29F).





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M027	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1
M028	0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0
M029	1	0	0	1	0	1	0	0	0	1	1	0	0	1	1	0

2.9.8 BKXNR, BKXNRP

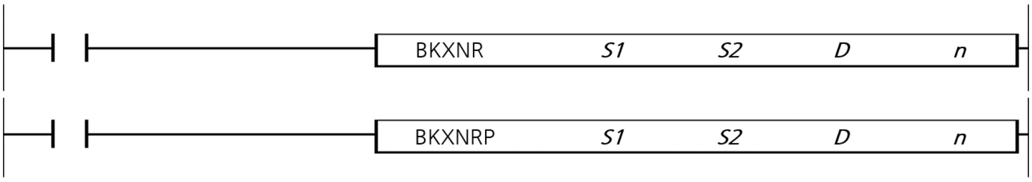
Supported PLC Series

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

BKXNR(P) instructions perform logical XNOR operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
The result is stored in *n* word devices starting from word device *D*.

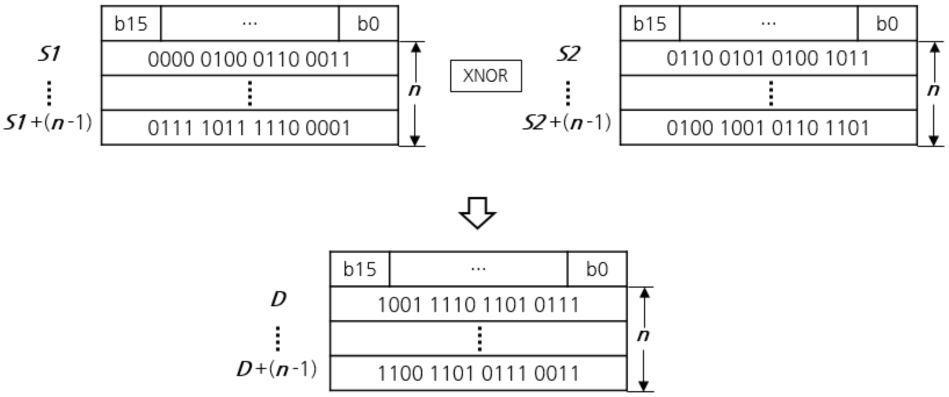
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BKXNR(P)	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	5	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



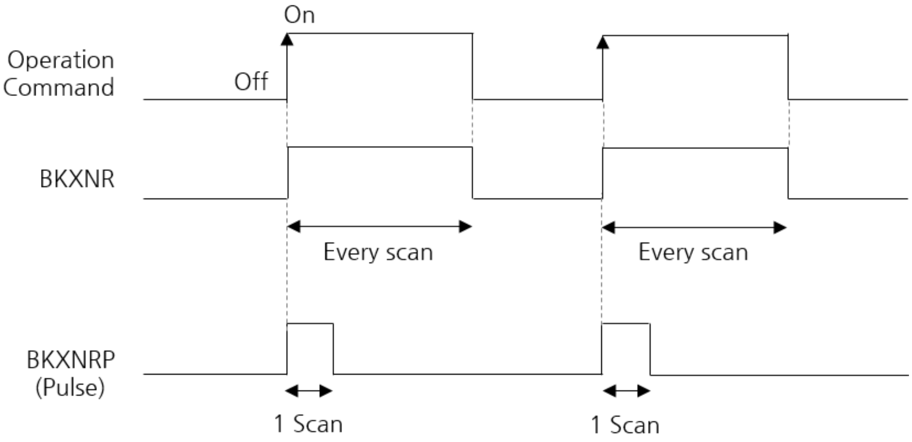
S1	Head address of word devices where data for logical XNOR operation is stored.
S2	Constant or head address of word devices where data for logical XNOR operation is stored.
D	Head address of word devices where operation result will be stored.
n	The number of word data blocks to perform logical XNOR operation.

BKXNR, BKXNRP

- BKXNR(P) instructions perform logical XNOR operation between *n* word data blocks starting from word device *S1* and *n* word data blocks starting from word device *S2*.
- Logical XNOR operation is performed between 16-bit data assigned to *S1* and 16-bit data assigned to *S2*.
- Logical XNOR operation is conducted on each bit of device *S1* and *S2*.
- If corresponding bits are in same status in device *S1* and *S2*, corresponding bits in device *D* are set.
- If corresponding bits are not in same status in device *S1* and *S2*, corresponding bits in device *D* are not set.



Execution Condition



Operation Error

Error Flag (F110)

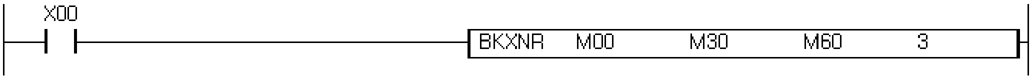
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

BKXNR

When operation command X00 is ON, BKXNR instruction is executed. Then, 3 word data blocks starting from word device M00 (M00 ~ M2F) and 3 word data blocks starting from M30 (M30 ~ M5F) conduct logical XNOR operation. The result is stored in 3 word devices starting from M60 (M60 ~ M8F).

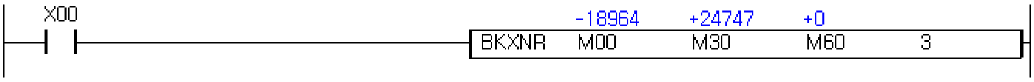
Ladder Diagram (LD)



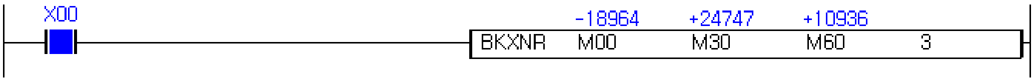
Instruction List (IL)

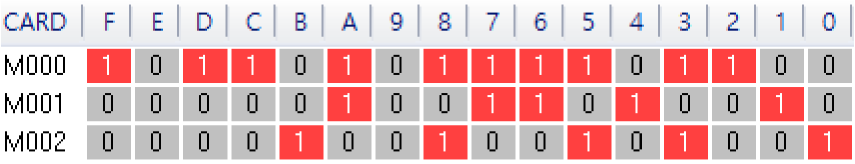
Instruction	Device			
LD	X00			
BKXNR	M00	M30	M60	3

- Each value assigned to 3 word devices starting from M00 (M00 ~ M2F) is "-18964" (= 1011 0101 1110 1100), "1234" (= 0000 0100 1101 0010) and "2345" (= 0000 1001 0010 1001).
- Each value assigned to 3 word devices starting from M30 (M30 ~ M5F) is "24747" (= 0110 0000 1010 1011), "4050" (= 0000 1111 1101 0010) and "6625" (= 0001 1001 1110 0001).



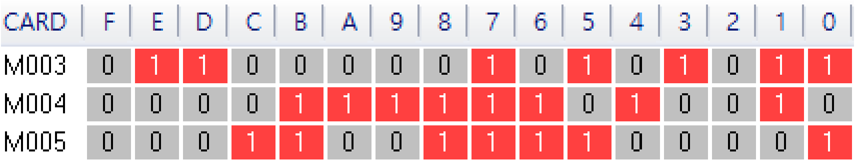
- When X00 turns ON, BKXNR instruction is executed. Then, the logical XNOR operation is conducted on 3 word devices starting from M00 (M00 ~ M2F) and 3 word devices starting from M30 (M30 ~ M5F). The result is stored in 3 word devices starting from M60 (M60 ~ M8F).





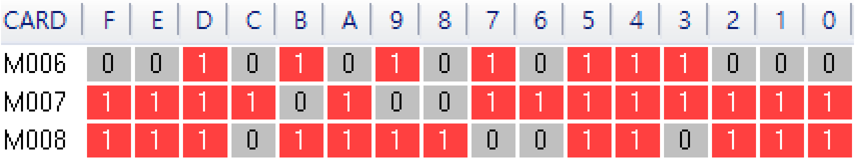
CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	1	1	0	1	0	1	1	1	1	0	1	1	0	0
M001	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0
M002	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

XNOR



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M003	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1
M004	0	0	0	0	1	1	1	1	1	1	0	1	0	0	1	0
M005	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1

⇩



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M006	0	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0
M007	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1
M008	1	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1

2.10 Rotation Instruction

2.10.1 ROR, RORP, DROR, DRORP

Supported PLC Series

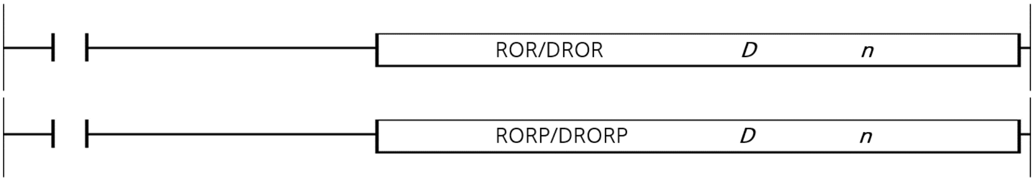
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

ROR(P) instructions rotate 16-bit data of word device *D*, *n* bits to the right. These instructions do not include carry flag.

DROR(P) instructions rotate 32-bit data of double word device *D*, *n* bits to the right. These instructions do not include carry flag.

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

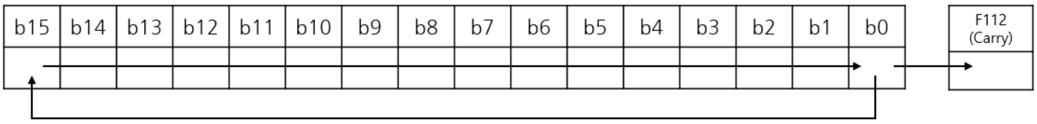


D	Head address of word/double word device where data to be rotated is stored.
n	The number of shifts. (0~15)

* 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. For more information, please refer to "Carry Flag" at [F : Internal flag (Relay)].

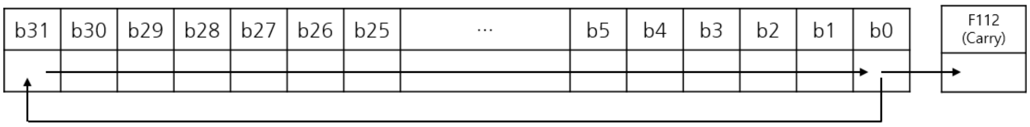
ROR, RORP

- ROR(P) instructions rotate 16-bit data stored in the word device *D* to the right.
- Data is rotated from the most significant bit to the least significant bit in range of word device *D*.
- The least significant bit shifts to carry flag (F112) and the most significant bit in range of word device *D*.
- Carry flag turns ON when "1" is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- When the instruction operates, data stored in the carry flag (F112) is not included in rotation.
- Value of *n* is valid from 0 to 15. If value of *n* is equal to or greater than 16, the number of rotation is specified by the remainder of *n* /16. (e.g. When *n* is 18, it is equivalent to the case when *n* is 2.)

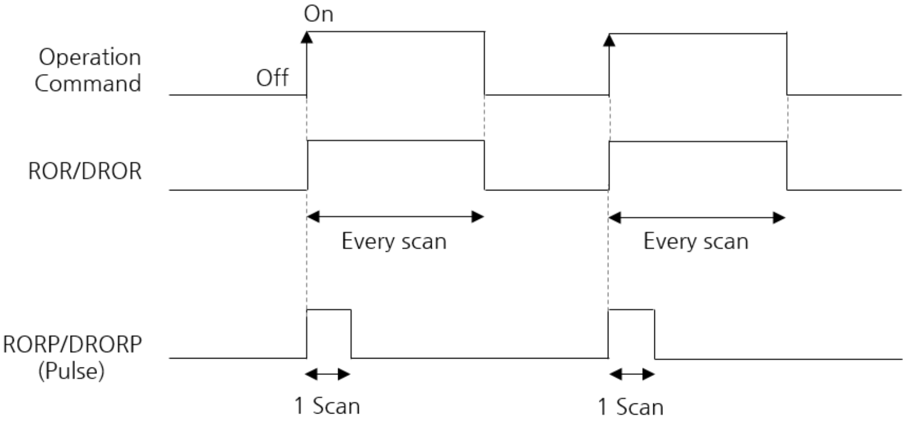


DROR, DRORP

- DROR(P) instructions rotate 32-bit data stored in the double word device *D* to the right.
- Data is shifted from the most significant bit to the least significant bit in range of double word device *D*.
- The least significant bit rotates to carry flag (F112) and the most significant bit in range of double word device *D*.
- Carry flag turns ON when "1" is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- When the instruction operates, data stored in the carry flag (F112) is not included in rotation.
- Value of *n* is valid from 0 to 31. If value of *n* is equal to or greater than 32, the number of rotation is specified by the remainder of *n* /32. (e.g. When *n* is 34, it is equivalent to the case when *n* is 2.)



Execution Condition



Operation Error

Error flag (F110)

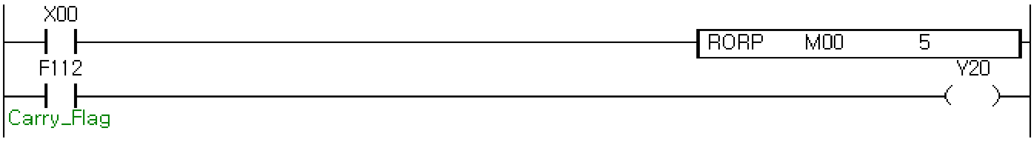
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

RORP

When operation command X00 is ON, RORP instruction is executed. Then, 16-bit data stored in word device M00 (M00 ~ M0F) rotates from the most significant bit to the least significant bit. The data is shifted 5 bits in the direction of least significant bit.
When the carry flag (F112) is ON, Y20 turns ON.

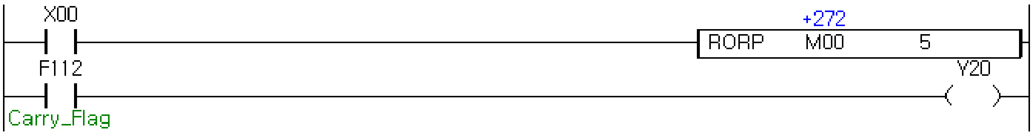
Ladder Diagram (LD)



Instruction List (IL)

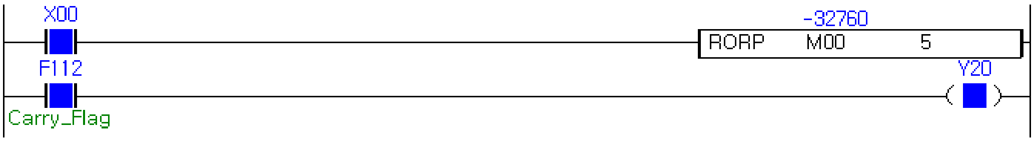
Instruction	Device	
LD	X00	
RORP	M00	5
LD	F112	
OUT	Y20	

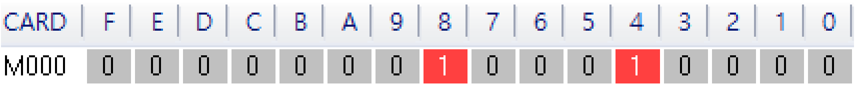
- “272” (= 0000 0001 0001 0000) is assigned to M00.



- When X00 turns ON, RORP instruction is executed. Then, data is shifted 5 bits in the direction of the least significant bit. Bit 4 is shifted to the carry flag (F112) and the most significant bit of word device M00.

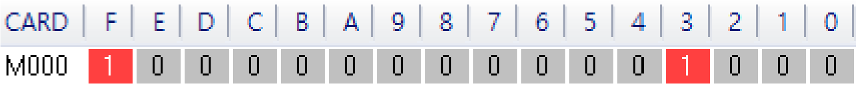
- Since bit 4 is “1”, carry flag is turned ON. Then, Y20 is turned ON.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

⇩



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

2.10.2 ROL, ROLP, DROL, DROLP

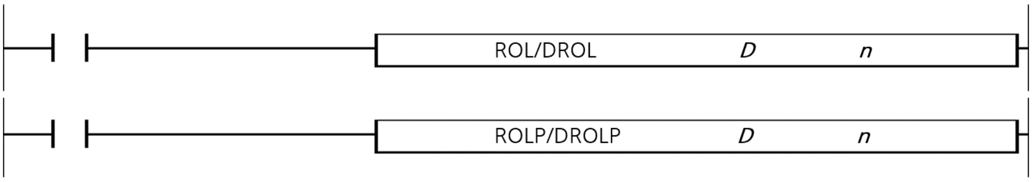
Supported PLC Series

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

ROL(P) instructions rotate 16-bit data of word device *D*, *n* bits to the left. These instructions do not include carry flag.
DROL(P) instructions rotate 32-bit data of double word device *D*, *n* bits to the left. These instructions do not include carry flag.

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

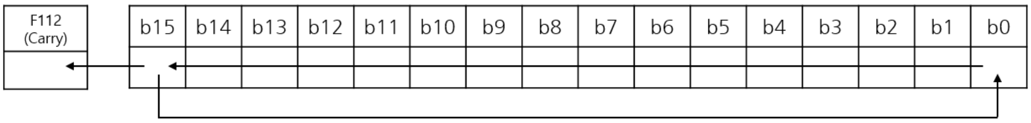


D	Head address of word/double word device where data to be rotated is stored.
n	The number of rotations. (0~15)

* 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. For more information, please refer to “Carry Flag” at [F : Internal flag (Relay)].

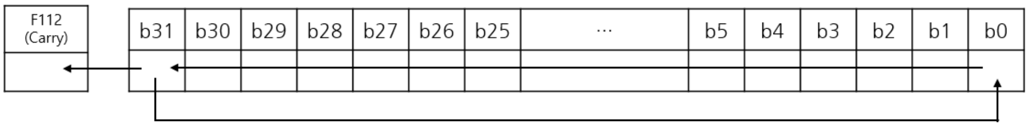
ROL, ROLP

- ROL(P) instructions rotate 16-bit data stored in the word device *D* to the left.
- Data is shifted from the least significant bit to the most significant bit in range of word device *D*.
- The most significant bit shifts to carry flag (F112) and the least significant bit in range of word device *D*.
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- When the instruction operates, data stored in the carry flag (F112) is not included in rotation.
- Value of *n* is valid from 0 to 15. If value of *n* is equal to or greater than 16, the number of rotation is specified by the remainder of *n* /16. (e.g. When *n* is 18, it is equivalent to the case when *n* is 2.)

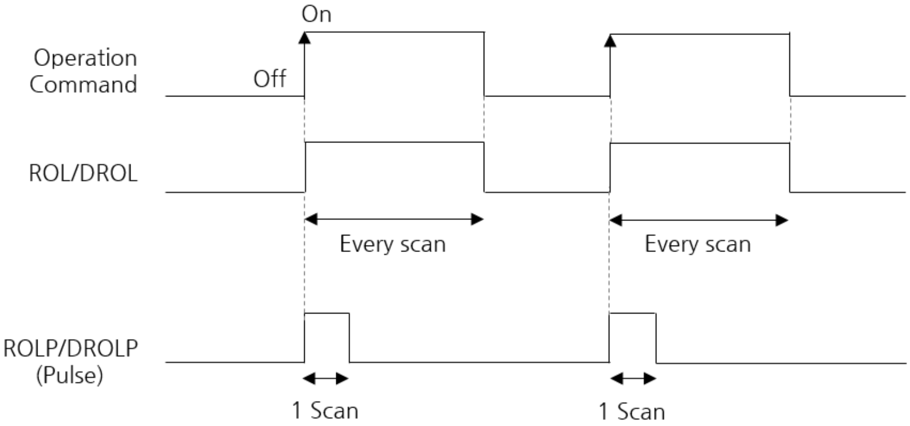


DROL, DROLP

- DROL(P) instructions rotate 32-bit data stored in the double word device *D* to the left.
- Data is shifted from the least significant bit to the most significant bit in range of double word device *D*.
- The most significant bit rotates to carry flag (F112) and the least significant bit in range of double word device *D*.
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- When the instruction operates, data stored in the carry flag (F112) is not included in rotation.
- Value of *n* is valid from 0 to 31. If value of *n* is equal to or greater than 32, the number of rotation is specified by the remainder of *n* /32. (e.g. When *n* is 34, it is equivalent to the case when *n* is 2.)



Execution Condition



Operation Error

Error Flag (F110)

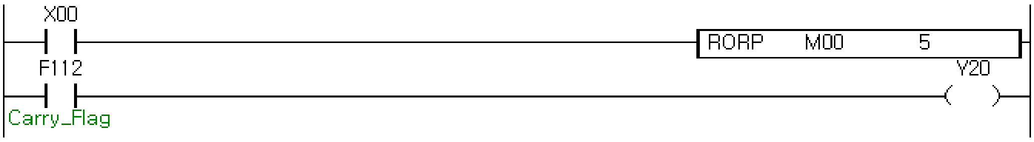
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

ROLP

When operation command X00 is ON, ROLP instruction is executed. Then, 16-bit data stored in word device M00 (M00 ~ M0F) rotates from the least significant bit to the most significant bit. The data is shifted 5 bits in the direction of the most significant bit.
When the carry flag (F112) is ON, Y20 turns ON.

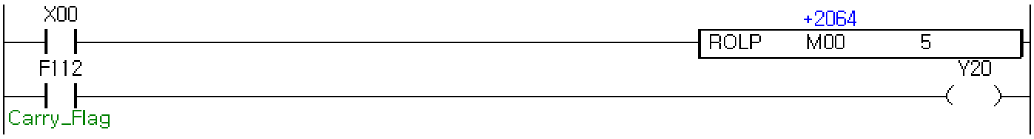
Ladder Diagram (LD)



Instruction List (IL)

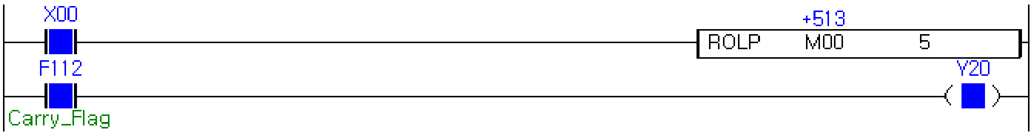
Instruction	Device	
LD	X00	
ROLP	M00	5
LD	F112	
OUT	Y20	

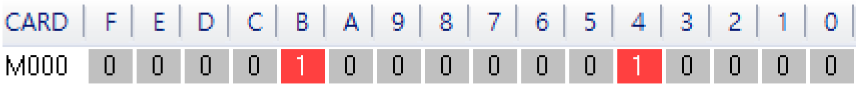
- "2064" (= 0000 1000 0001 0000) is assigned to M00.



- When X00 turns ON, ROLP instruction is executed. Then, data is shifted 5 bits in the direction of the most significant bit. Bit 11 is shifted to the carry flag (F112) and the least significant bit of word device M00.

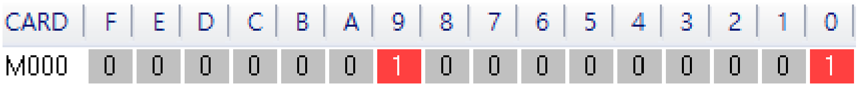
- Since bit 11 is "1", carry flag is turned ON. Then, Y20 is turned ON.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

⇩



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

2.10.3 RCR, RCRP, DRCR, DRCRP

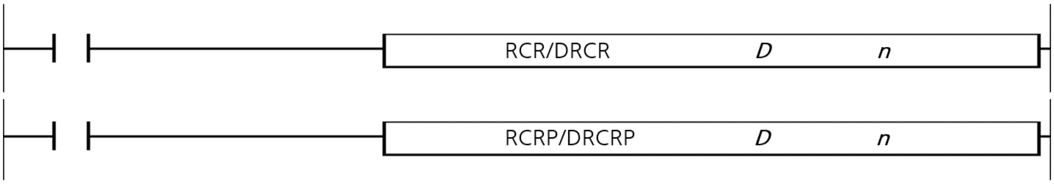
Supported PLC Series

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

RCR(P) instructions rotate 16-bit data of word device *D*, *n* bits to the right. These instructions include carry flag.
DRCR(P) instructions rotate 32-bit data of double word device *D*, *n* bits to the right. These instructions include carry flag.

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

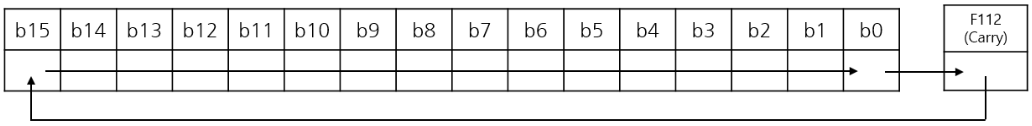


D	Head address of word/double word device where data to be rotated is stored.
n	The number of shifts. (0~15)

* 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. For more information, please refer to "Carry Flag" at [F : Internal flag (Relay)].

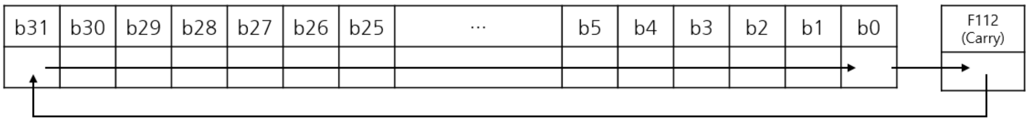
RCR, RCRP

- RCR(P) instructions rotate 16-bit data stored in the word device *D* and carry flag to the right.
- Data is shifted from the most significant bit to the least significant bit in range of word device *D*.
- The least significant bit shifts to carry flag (F112).
- Carry flag turns ON when "1" is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- The bit stored in the carry flag (F112) shifts to the most significant bit.
- Value of *n* is valid from 0 to 15. If value of *n* is equal to or greater than 16, the number of shift is specified by the remainder of *n* /16. (e.g. When *n* is 18, it is equivalent to the case when *n* is 2.)

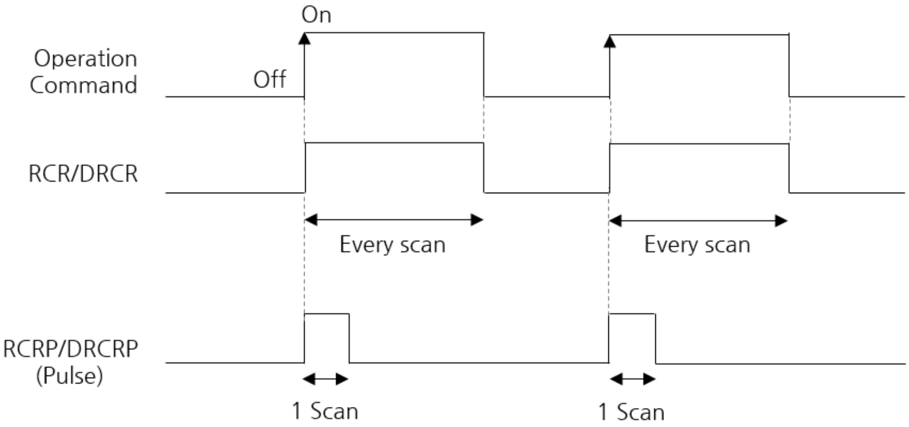


DRCR, DRCRP

- DRCR(P) instructions rotate 32-bit data stored in the double word device *D* and carry flag to the right.
- Data is shifted from the most significant bit to the least significant bit in range of double word device *D*.
- The least significant bit shifts to carry flag (F112).
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- The bit stored in the carry flag (F112) shifts to the most significant bit.
- Value of *n* is valid from 0 to 31. If value of *n* is equal to or greater than 32, the number of shift is specified by the remainder of *n* /32. (e.g. When *n* is 34, it is equivalent to the case when *n* is 2.)



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)

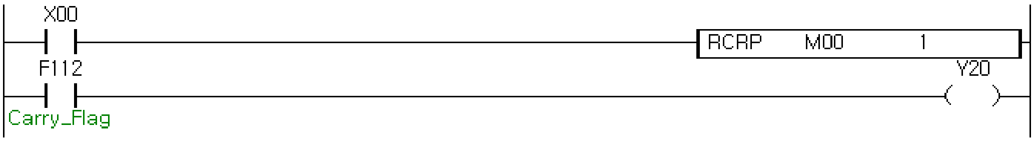
Program Example

RCRP

When operation command X00 is ON, RCRP instruction is executed. Then, 16-bit data stored in word device M00 (M00 ~ M0F) rotates from the most significant bit to the least significant bit including carry flag (F112). The data is shifted 1 bit in the direction of least significant bit. When the carry flag (F112) is ON, Y20 turns ON.

When X00 is turned OFF and ON again, data stored in the carry flag is shifted to the most significant bit.

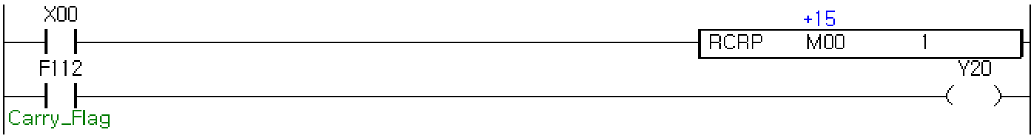
Ladder Diagram (LD)



Instruction List (IL)

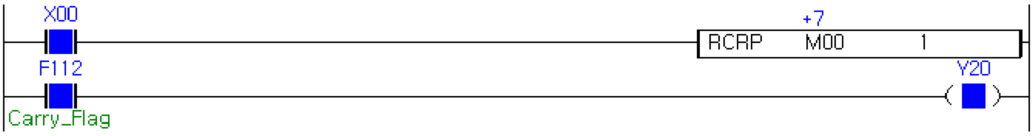
Instruction	Device	
LD	X00	
RCRP	M00	1
LD	F112	
OUT	Y20	

- “15” (= 0000 0000 0000 1111) is assigned to M00.

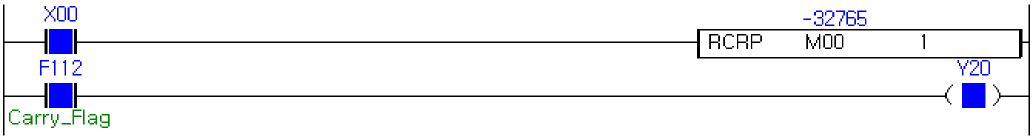


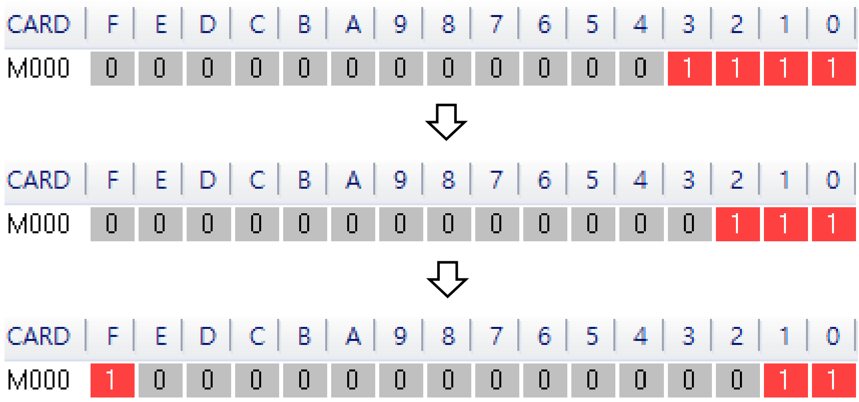
- When X00 turns ON, RCRP instruction is executed. Then, data is shifted 1 bit in the direction of the least significant bit. Bit 0 is shifted to the carry flag (F112).

- Since bit 0 is “1”, carry flag is turned ON. Then, Y20 is turned ON.



- When X00 is turned OFF and turned ON again, the bit stored in the carry flag shifts to the most significant bit.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

2.10.4 RCL, RCLP, DRCL, DRCLP

Supported PLC Series

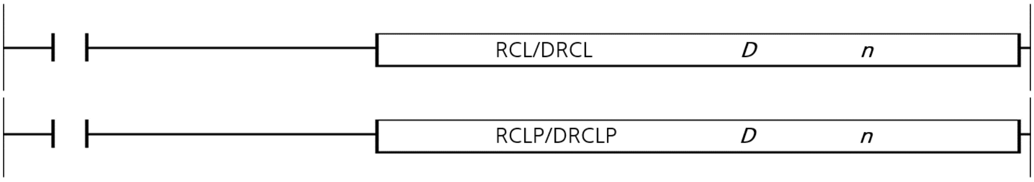
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	O	O	O

Function

RCL(P) instructions rotate 16-bit data of word device *D*, *n* bits to the left. These instructions include carry flag.

DRCL(P) instructions rotate 32-bit data of double word device *D*, *n* bits to the left. These instructions include carry flag.

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

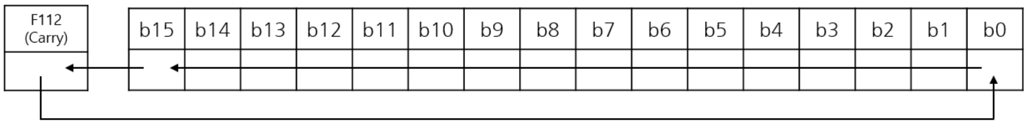


D	Head address of word/double word device where data to be rotated is stored.
n	The number of rotations. (0~15)

* 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. For more information, please refer to “Carry Flag” at [F : Internal flag (Relay)].

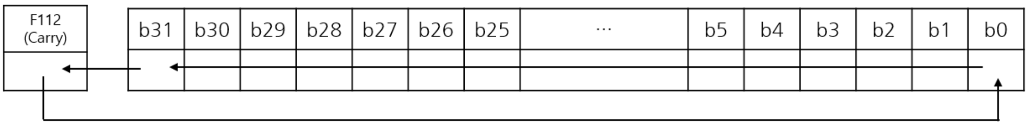
RCL, RCLP

- RCL(P) instructions rotate 16-bit data stored in the word device *D* and carry flag to the left.
- Data is shifted from the least significant bit to the most significant bit in range of word device *D*.
- The most significant bit shifts to carry flag (F112).
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- The bit stored in the carry flag (F112) shifts to the least significant bit.
- Value of *n* is valid from 0 to 15. If value of *n* is equal to or greater than 16, the number of shift is specified by the remainder of *n* /16. (e.g. When *n* is 18, it is equivalent to the case when *n* is 2.)

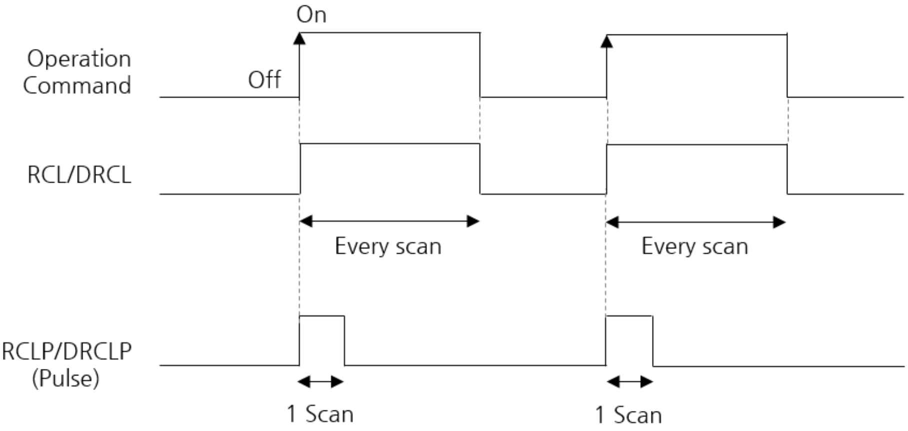


DRCL, DRCLP

- DRCL(P) instructions rotate 32-bit data stored in the double word device *D* and carry flag to the left.
- Data is shifted from the least significant bit to the most significant bit in range of double word device *D*.
- The most significant bit shifts to carry flag (F112).
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- The bit stored in the carry flag (F112) shifts to the least significant bit.
- Value of *n* is valid from 0 to 31. If value of *n* is equal to or greater than 32, the number of shift is specified by the remainder of *n* /32. (e.g. When *n* is 34, it is equivalent to the case when *n* is 2.)



Execution Condition



Operation Error

Error Flag (F110)

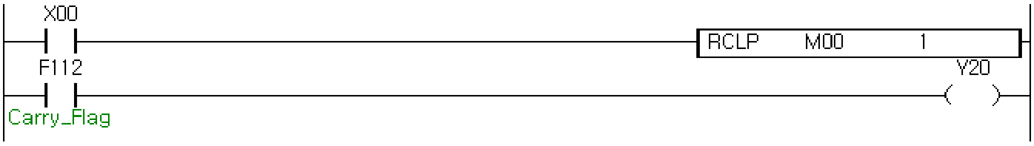
F110 turns ON for 1 scan when the device specified by @D exceeds the range of the device D. (Range of device D depends on CPU type)

Program Example

RCLP

When operation command X00 is ON, RCLP instruction is executed. Then, 16-bit data stored in word device M00 (M00 ~ M0F) rotates from the least significant bit to the most significant bit. The data is shifted 1 bit in the direction of most significant bit including carry flag (F112).
When the carry flag (F112) is ON, Y20 turns ON.
When X00 is turned OFF and ON again, data stored in the carry flag is shifted to the least significant bit.

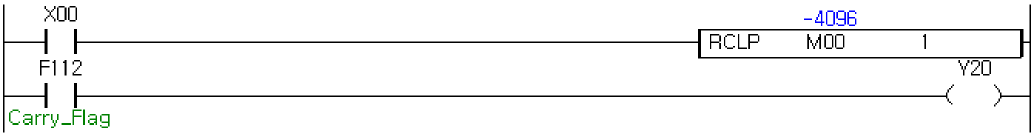
Ladder Diagram (LD)



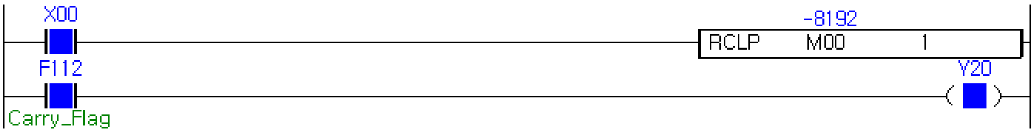
Instruction List (IL)

Instruction	Device	
LD	X00	
RCLP	M00	1
LD	F112	
OUT	Y20	

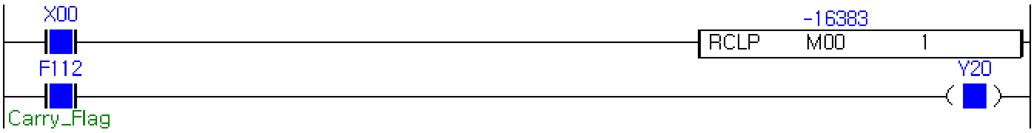
- "-4096" (= 1111 0000 0000 0000) is assigned to M00.

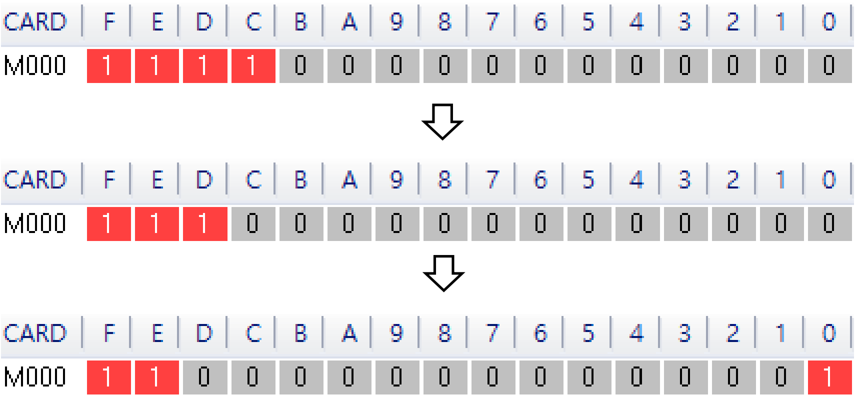


- When X00 turns ON, RCLP instruction is executed. Then, data is rotated 1 bit in the direction of the most significant bit. Bit 15 is shifted to the carry flag (F112).
- Since bit 15 is "1", carry flag is turned ON. Then, Y20 is turned ON.



- When X00 is turned OFF and turned ON again, the bit stored in the carry flag shifts to the least significant bit.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

2.11 Shift Instruction

2.11.1 SFR, SFRP, SFL, SFLP

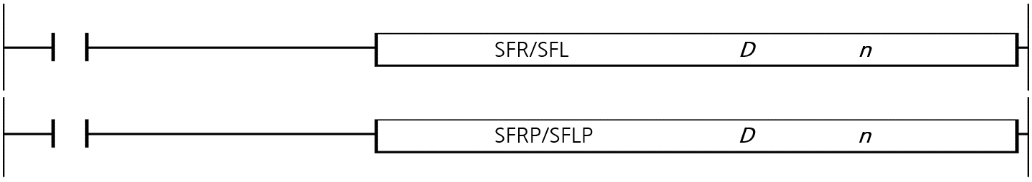
Supported PLC Series

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

SFR(P) instructions shift 16-bit data of word device *D*, *n* bits to the right.
SFL(P) instructions shift 16-bit data of word device *D*, *n* bits to the left.

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



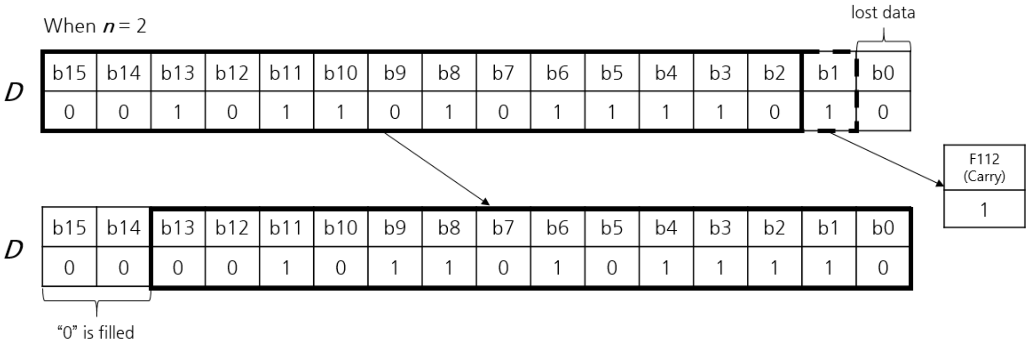
D	Address of word device where data to be shifted is stored.
n	The number of shifts. (0~15)

* 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. For more information, please refer to "Carry Flag" at [F : Internal flag (Relay)].

SFR, SFRP

- SFR(P) instructions shift 16-bit data stored in the word device *D* to the right.
- Data is shifted in the direction of the least significant bit in word device *D*.
- Carry flag turns ON when "1" is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- *n* bits starting from the most significant bit are filled with "0" when the instruction is exeucted.
- Data from bit 0 to bit *n* -2 are lost after the execution of instruction.
- Value of *n* is valid from 0 to 15. If value of *n* is equal to or greater than 16, the number of bits to shift is specified by the remainder of *n* /16. (e.g. When *n* is 18, it is equivalent to the case when *n* is 2.)

⚠ In device T and C, only the value of TC and CC can be shifted.



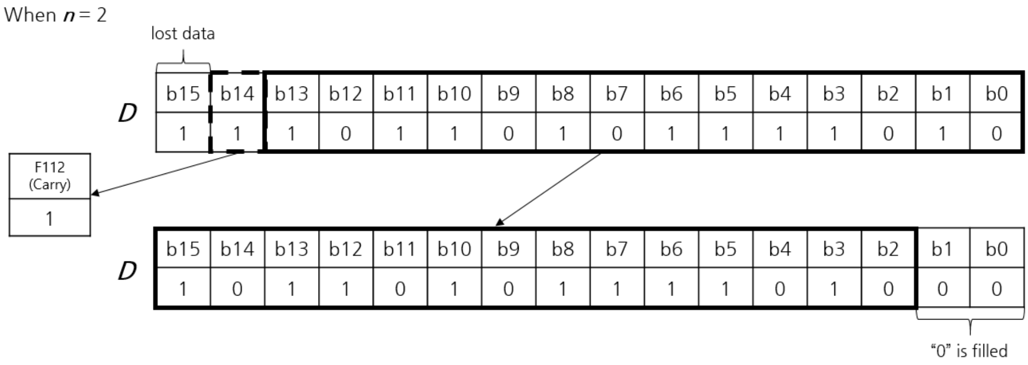
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	1	0	1	1	0	1	0	1	1	1	1	0	1	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	0

SFL, SFLP

- SFL(P) instructions shift 16-bit data stored in the word device *D* to the left.
- Data is shifted in the direction of the most significant bit in word device *D*.
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- *n* bits starting from the least significant bit are filled with “0” when the instruction is exeucted.
- Data of bit 15-(*n* -2) to bit 15 are lost after the execution of instruction.
- Value of *n* is valid from 0 to 15. If value of *n* is equal to or greater than 16, the number of shifts is specified by the remainder of *n* /16. (e.g. When *n* is 18, it is equivalent to the case when *n* is 2.)

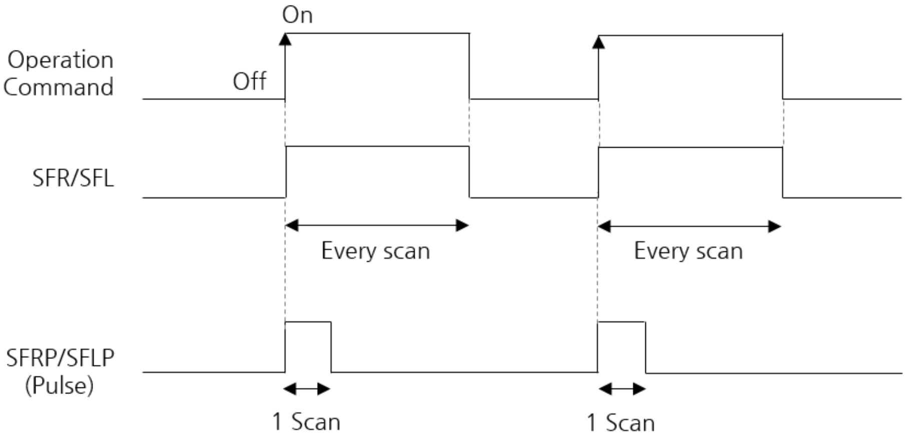
⚠ In device T and C, only the value of TC and CC can be shifted.



b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	0	1	1	0	1	0	1	1	1	1	0	1	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	1	0	1	0	1	1	1	1	0	1	0	0	0

Execution Condition



⚠ Using pulse contact as execution condition of SFR/SFL instruction or using SFRP/SFLP instruction is recommended.

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)

Carry flag (F112)

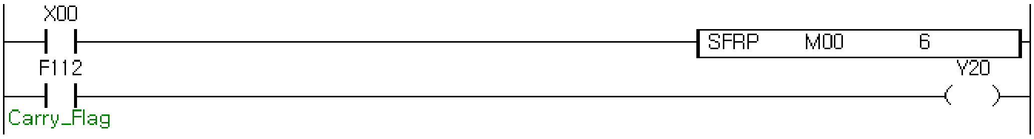
F112 turns ON for 1 scan when operation result overflows.

Program Example

SFRP

When operation command X00 is ON, SFRP instruction is executed. Then, 16-bit data stored in word device M00 (M00 ~ M0F) shifts in the direction of the least significant bit. The data is shifted 6 bits.
The data of bit 5 in word device M00 (M00 ~ M0F) is stored in the carry flag (F112).
When the carry flag (F112) is ON, Y20 turns ON.

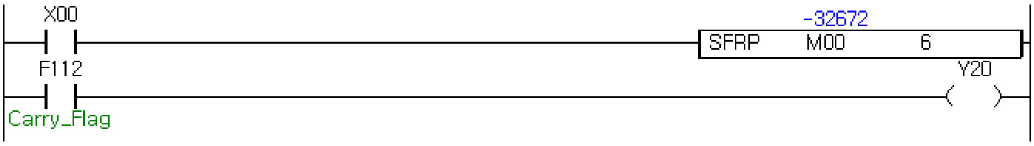
Ladder Diagram (LD)



Instruction List (IL)

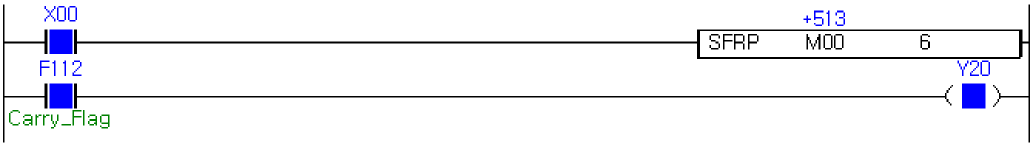
Instruction	Device	
LD	X00	
SFRP	M00	5
LD	F112	
OUT	Y20	

- "-32672" (= 1000 0000 0110 0000) is assigned to M00.

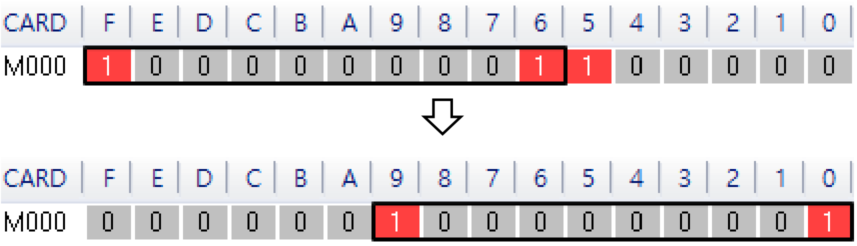


- When X00 turns ON, SFRP instruction is executed. Then, data is shifted 6 bits in the direction of the least significant bit.

- Bit 5 is shifted to the carry flag (F112) and the carry flag is turned ON since bit 5 is "1". Then, Y20 is turned ON.



- 6 bits starting from the most significant bit are filled with "0". (M0A~M0F)



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

2.11.2 BSFR, BSFRP, BSFL, BSFLP

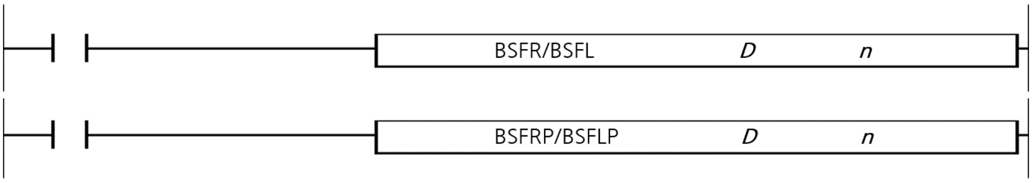
Supported PLC Series

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

BSFR(P) instructions shift *n* -bit data of bit device *D*, 1 bit to the right.
BSFL(P) instructions shift *n* -bit data of bit device *D*, 1 bit to the left.

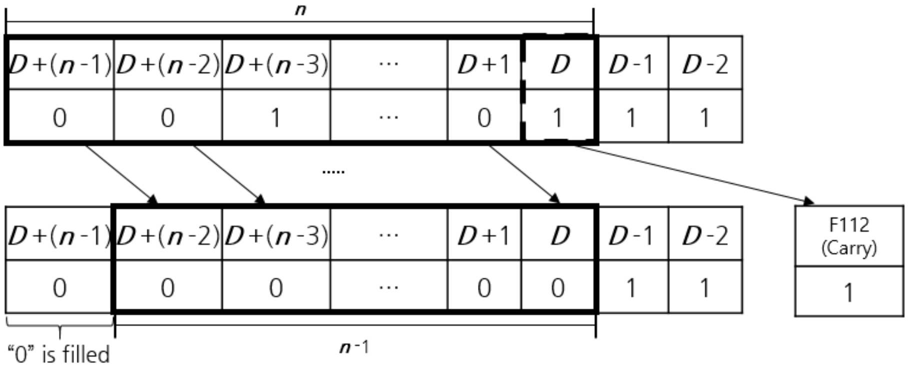
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BSFR(P) BSFL(P)	*D*	O	-	O	O	O	-	-	-	-	-	-	-	-	-	-	3	O	-	O
	n	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



D	Head address of bit device where *n* -bit data to be shifted is stored.
n	The number of bits to shift.

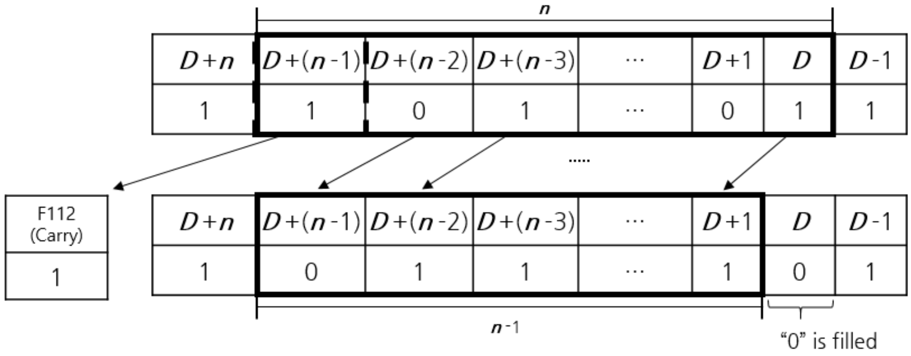
BSFR, BSFRP

- BSFR(P) instructions shift *n* -bit data starting from the bit device *D* to the right.
- Data is shifted 1 bit in the direction of the least significant bit.
- The data located in *D* is shifted to carry flag (F112).
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- After the execution of instruction, the device *D* +(*n* -1) is filled with “0”.
- The range of *n* cannot exceed the size of bit device *D*. The size of bit device *D* depends on the CPU type.

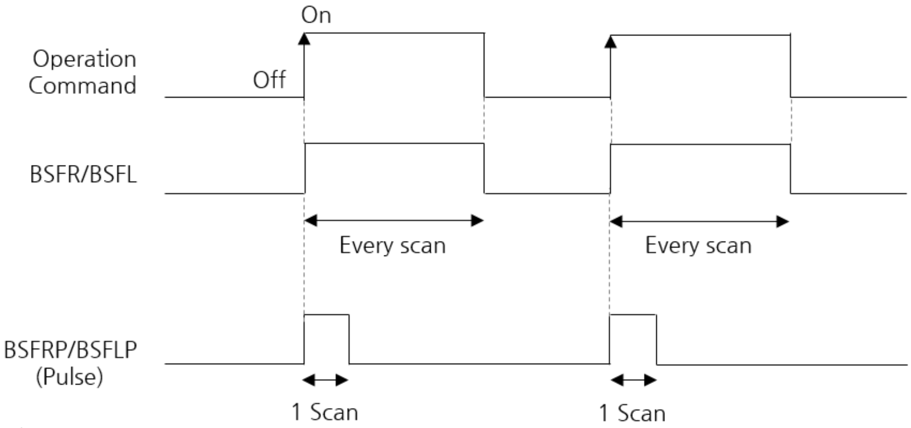


BSFL, BSFLP

- BSFL(P) instructions shift *n* -bit data starting from the bit device *D* to the left.
- Data is shifted 1 bit in the direction of the most significant bit.
- The data located in *D* +(*n* -1) is shifted to carry flag (F112).
- Carry flag turns ON when “1” is shifted into the carry flag (F112). In other cases, carry flag is turned OFF.
- After the execution of instruction, the device *D* is filled with “0”.
- The range of *n* cannot exceed the size of bit device *D*. The size of bit device *D* depends on the CPU type.



Execution Condition



⚠ Using pulse contact as execution condition of BSFR/BSFL instruction or using BSFRP/BSFLP instruction is recommended.

Operation Error

Error Flag (F110)

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

Carry flag (F112)

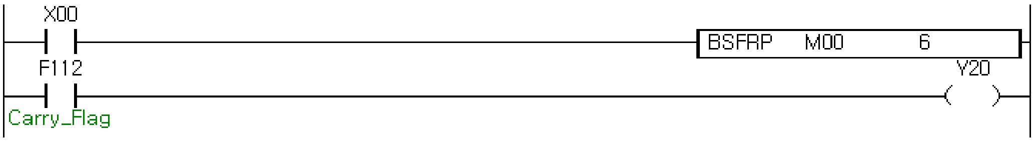
F112 turns ON for 1 scan when operation result overflows.

Program Example

BSFRP

When operation command X00 is ON, BSFRP instruction is executed. Then, 6-bit data starting from bit device M00 (M00 ~ M05) shifts 1 bit in the direction of the least significant bit. The bit data stored in M00 shifts to the carry flag (F112). If M00 is “1”, the carry flag (F112) turns ON. When the carry flag (F112) is ON, Y20 turns ON.

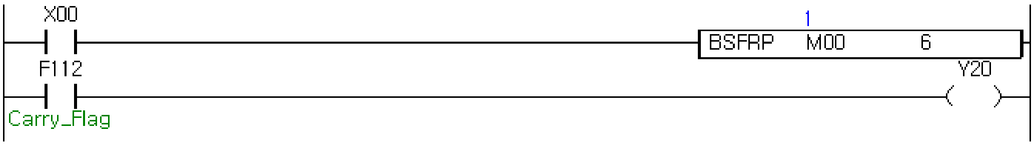
Ladder Diagram (LD)



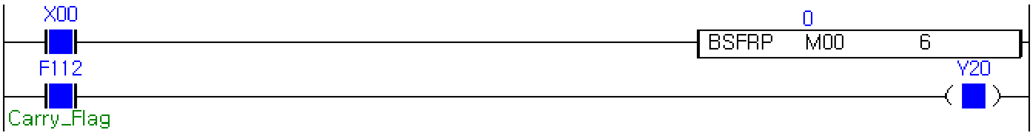
Instruction List (IL)

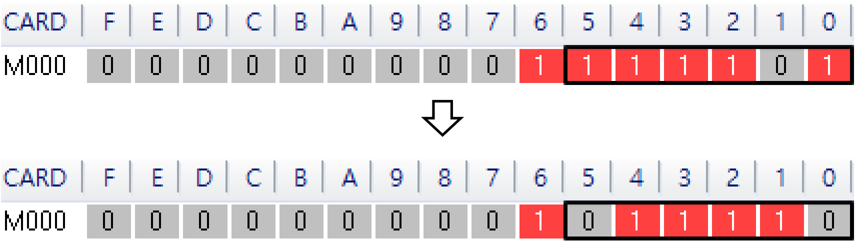
Instruction	Device	
LD	X00	
BSFRP	M00	6
LD	F112	
OUT	Y20	

- “125” (= 0000 0000 0111 1101) is assigned to word device M00.



- When X00 turns ON, BSFRP instruction is executed. Then, data is shifted 1 bit in the direction of the least significant bit. The bit data stored in M00 shifts to the carry flag (F112). If M00 is “1”, the carry flag (F112) turns ON. Then, Y20 is turned ON.





CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1

⇩

CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0

2.11.3 DSFR, DSFRP, DSFL, DSFLP

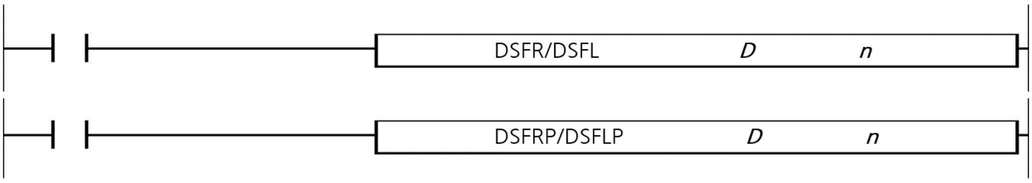
Supported PLC Series

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

DSFR(P) instructions shift *n* word devices starting from word device *D*, 1 word to the right.
DSFL(P) instructions shift *n* word devices starting from word device *D*, 1 word to the left.

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

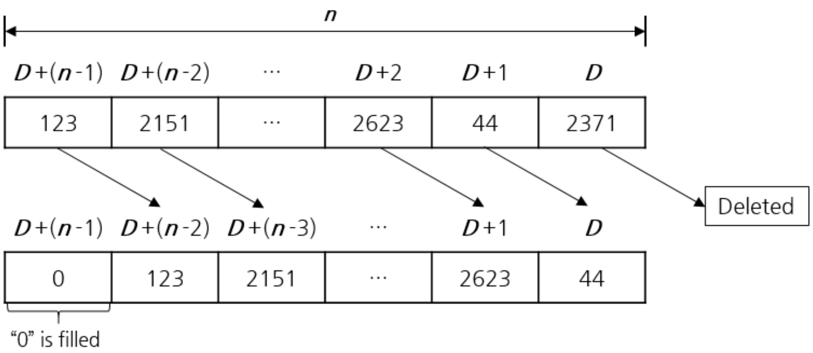


D	Head address of word device where data to be shifted is stored.
n	The number of word devices to shift.

DSFR, DSFRP

- DSFR(P) instructions shift *n*-word data starting from the word device *D* to the right.
- Data is shifted 1 word in the direction of the least significant device.
- After the execution of instruction, the device *D*+(*n*-1) is filled with “0”.
- Data of *D* is lost after the execution of instruction.
- The range of *n* cannot exceed the size of word device *D*. The size of word device *D* depends on the CPU type.

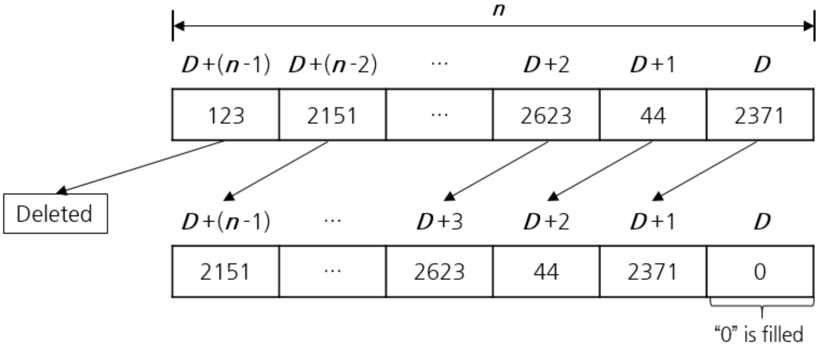
⚠ In device T and C, only the value of TC and CC can be shifted.



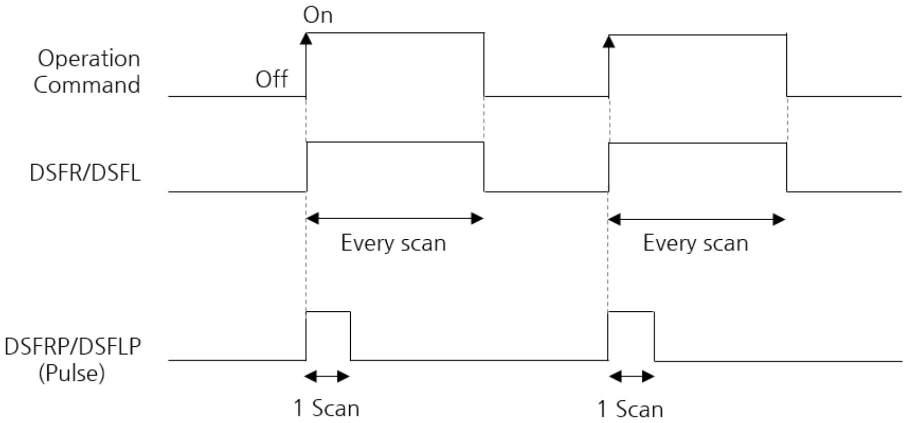
DSFL, DSFLP

- DSFL(P) instructions shift *n* -word data starting from the word device *D* to the left.
- Data is shifted 1 word in the direction of the most significant device.
- After the execution of instruction, the device *D* is filled with "0".
- The range of *n* cannot exceed the size of word device *D*. The size of word device *D* depends on the CPU type.

⚠ In device T and C, only the value of TC and CC can be shifted.



Execution Condition



⚠ Using pulse contact as execution condition of DSFR/DSFL instruction or using DSFRP/DSFLP instruction is recommended.

Operation Error

Error flag (F110)

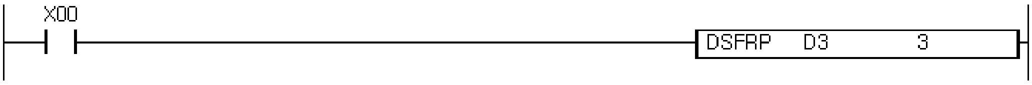
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DSFRP

When operation command X00 is ON, DSFRP instruction is executed. Then, 3 word data stored in word device starting from D3 (D3 ~ D5) shift 1 word in the direction of the least significant device.

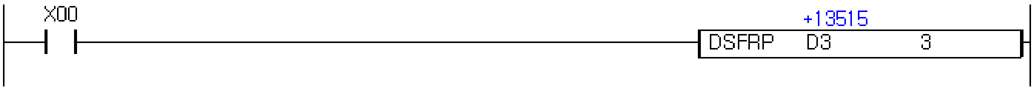
Ladder Diagram (LD)



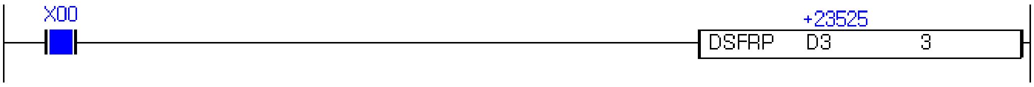
Instruction List (IL)

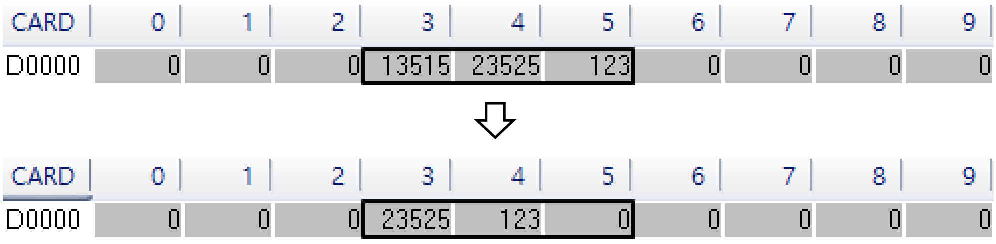
Instruction	Device	
LD	X00	
DSFRP	D3	3

- “13515” is assigned to D3. “23525” is assigned to D4. “123” is assigned to D5.



- When X00 turns ON, DSFRP instruction is executed. Then, data is shifted 1 word in the direction of the least significant device.





CARD	0	1	2	3	4	5	6	7	8	9
D0000	0	0	0	13515	23525	123	0	0	0	0

⇩

CARD	0	1	2	3	4	5	6	7	8	9
D0000	0	0	0	23525	123	0	0	0	0	0

2.12 Clock Instruction

2.12.1 DATERD, DATERDP

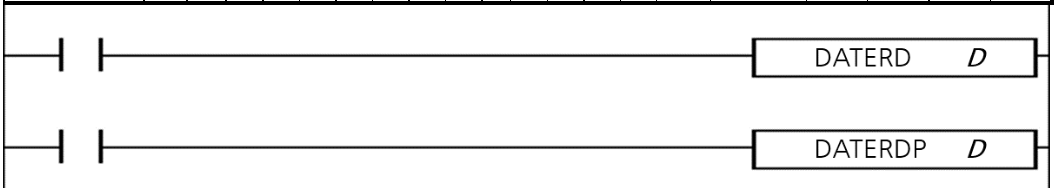
Supported PLC Series

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

DATERD and DATERDP read clock data from RTC (Clock Device) and store in 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
DATERD(P)	*D*	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-	3	O	-	-



D	Head address of word devices where the clock data will be stored.

DATERD, DATERDP

- The instruction reads clock data from RTC (Clock Device).
- The clock data is stored in the word device starting from *D*. Each clock data (year, month, day, hour, minute, second, day of week) is stored in the corresponding word device as below.

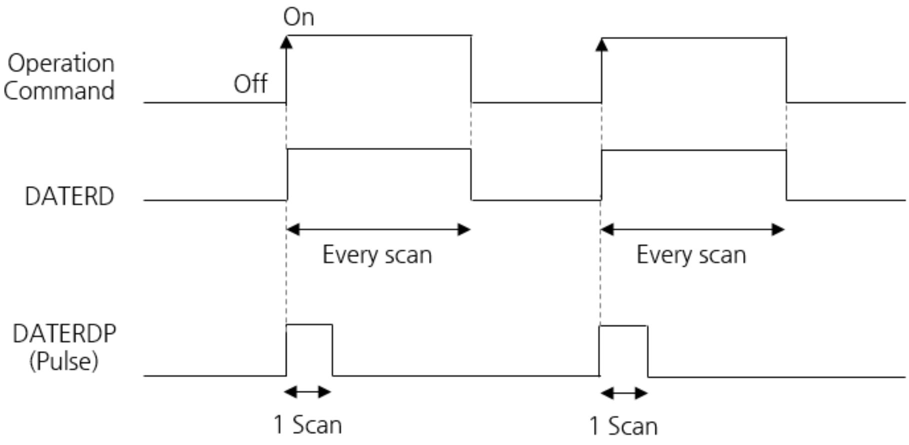
		Data range
D	Year	2000 - 2099
D+1	Month	1 - 12
D+2	Day	1 - 31
D+3	Hour	0 - 23
D+4	Minute	0 - 59
D+5	Second	0 - 59
D+6	Day of week	0 - 6

- The year is indicated in range of 2000-2099
- The day of week is indicated in a range of number 0 to 6. From 0 to 6, each number represents Sunday to Saturday.

0	1	2	3	4	5	6
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

- For leap years, compensation is made automatically.

Execution Condition



Operation Error

Error Flag (F110)

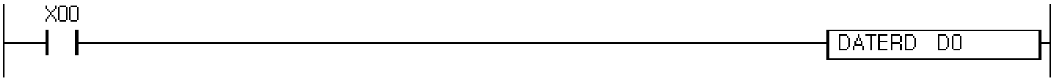
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DATERD, DATERDP

When X00 turns ON, the program reads clock data from RTC (Clock data) and stores the data starting from D0.

Ladder Diagram (LD)

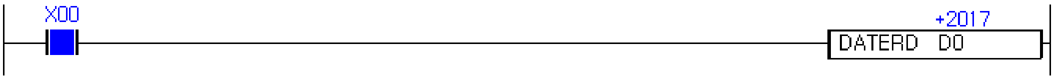


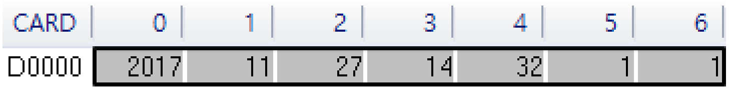
Instruction List (IL)

Instruction	Device
LD	X00
DATERD	D0

The program operates as following:

- When X00 is turned ON, the instruction executes.
- Year data is stored in D0. Month data is stored in D1. Day data is stored in D2. Hour data is stored in D3. Minute data is stored in D4. Second data is stored in D5. Day of week data is stored in D6.





CARD	0	1	2	3	4	5	6
D0000	2017	11	27	14	32	1	1

2.12.2 DATEWR, DATEWRP

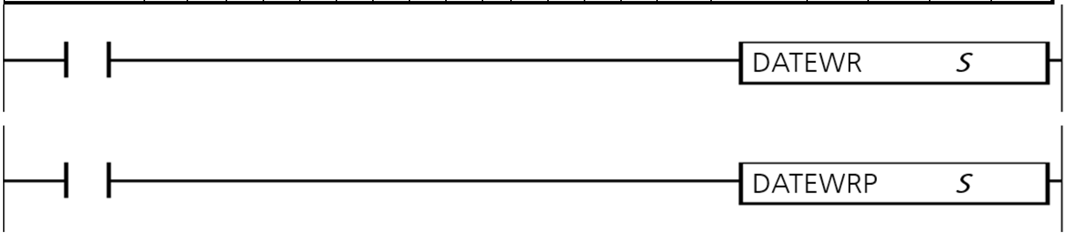
Supported PLC Series

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

DATEWR and DATEWRP write clock data from the device assigned to *S* to RTC (Clock Device).

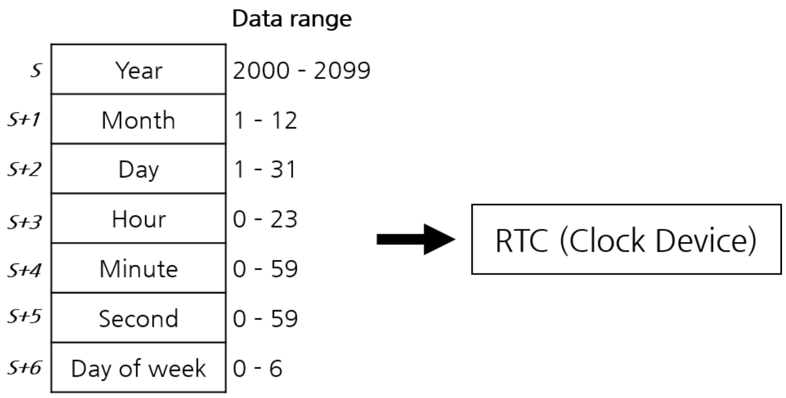
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
DATEWR(P)	*S*	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-	3	O	-	-



S	Head address of word devices where the clock data to be written to RTC is stored.

DATEWR, DATEWRP

- The instruction writes clock data to RTC (Clock Device).
- The clock data to be written to RTC is stored in the word device starting from *S*. Each clock data (year, month, day, hour, minute, second, day of week) is stored in each word device.

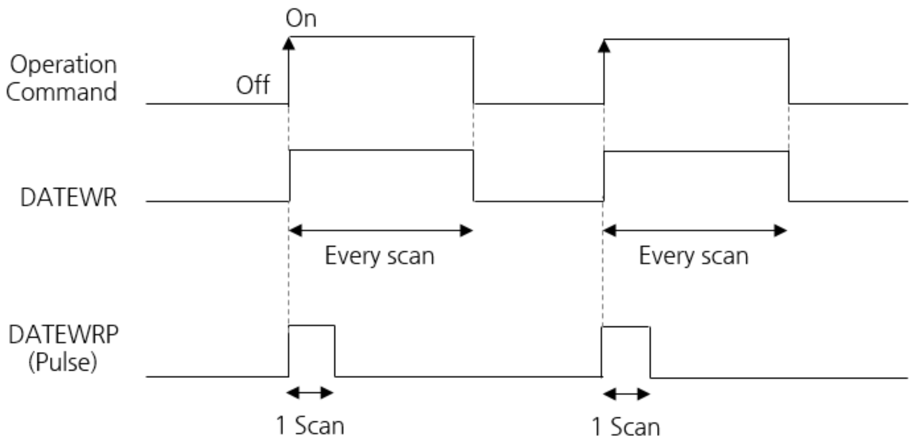


		Data range
S	Year	2000 - 2099
S+1	Month	1 - 12
S+2	Day	1 - 31
S+3	Hour	0 - 23
S+4	Minute	0 - 59
S+5	Second	0 - 59
S+6	Day of week	0 - 6

- Input the year in range of 2000-2099.
- The day of week is indicated as a range of number 0 to 6. From 0 to 6, each number represents Sunday to Saturday.

0	1	2	3	4	5	6
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Execution Condition



Operation Error

Error Flag (F110)

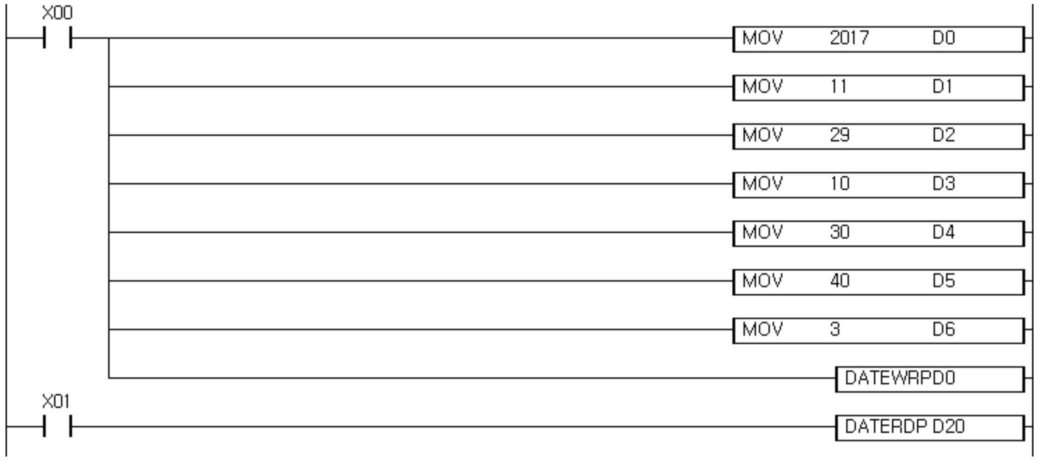
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DATEWR, DATEWRP

When X00 turns ON, the program stores the clock data in D0 - D6. Then writes clock data to RTC (Clock data), starting from D0. After the operation, you can check the result by using DATERD(P) instruction.

Ladder Diagram (LD)

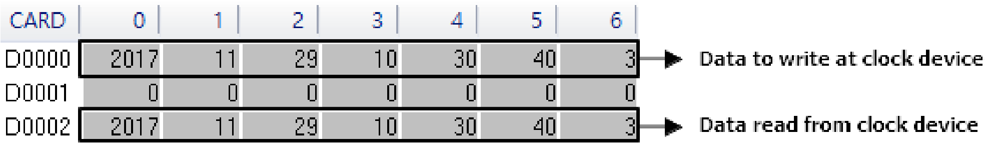


Instruction List (IL)

Instruction	Device	
LD	X00	
MOV	2017	D0
MOV	11	D1
MOV	29	D2
MOV	10	D3
MOV	30	D4
MOV	40	D5
MOV	3	D6
DATEWRP	D0	
LD	X01	
DATERDP	D20	

The program operates as following:

- When X00 is turned ON, each MOV instruction operates.
- Moved data are stored in destination device. Year (2017) in D0, month (11) in D1, day (29) in D2, hour (10) in D3, minute (30) in D4, second (40) in D5 and day of week (3) in D6.
- When X01 is turned ON, the clock data written to the RTC is stored in D20. There might be a few seconds delay since X00 and X01 are turned ON at different moment.



CARD	0	1	2	3	4	5	6
D0000	2017	11	29	10	30	40	3
D0001	0	0	0	0	0	0	0
D0002	2017	11	29	10	30	40	3

Data to write at clock device

Data read from clock device

2.12.3 DATE+, DATE+P

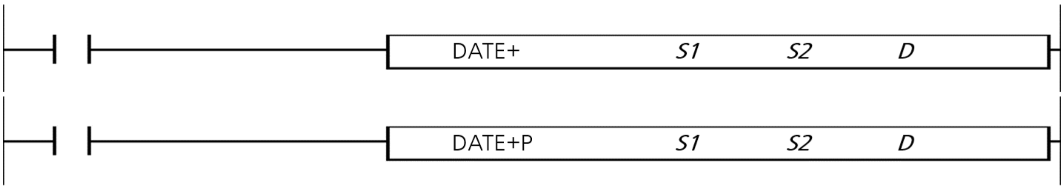
Supported PLC Series

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

DATE+ and DATE+P add a time data assigned to *S2* and time data assigned to *S1.* Result is stored in word 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
DATE+ DATE+P	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



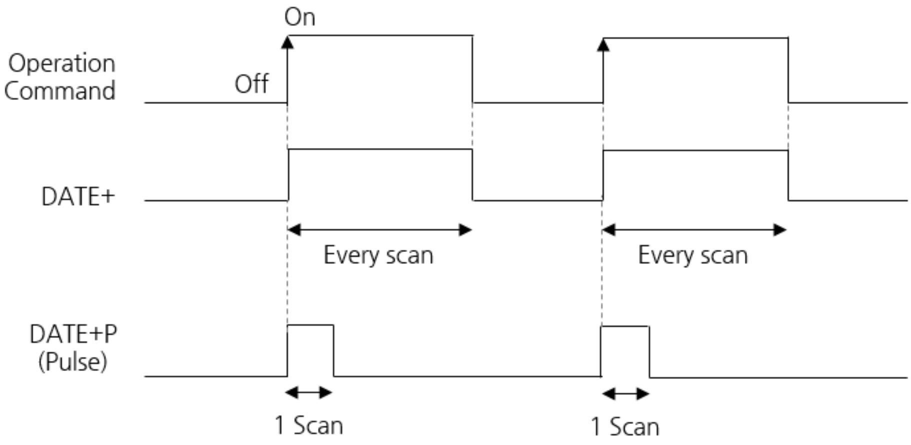
S1	Head address of word devices where the time data to be added is stored.
S2	Head address of word devices where the time data for addition is stored.
D	Head address of word devices where the added time data will be stored.

DATE+, DATE+P

- The instruction adds *S2* to *S1*. *S2* is a time data which is added to *S1. S1* is a time data to be added.
- The added time data is stored in the word device starting from *D*.
- Time data of *S1* and *S2* should exist in order of hour (0 - 23), minute (0 - 59), second (0 - 59).
- If the result of addition exceeds 24 hours, 24 hours will be subtracted from the result.

		Data Range				Data Range				Data Range
S1	Hour	0-23		***S2***	Hour	0-23		***D***	Hour	0-23
S1+1	Minute	0-59	+	***S2+1***	Minute	0-59	=	***D+1***	Minute	0-59
S1+2	Second	0-59		***S2+2***	Second	0-59		***D+2***	Second	0-59

Execution Condition



Operation Error

Error Flag (F110)

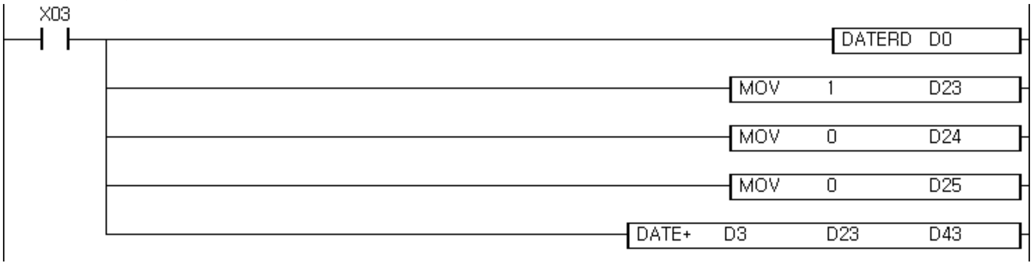
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D.(Range of device D depends on CPU type)

Program Example

DATE+, DATE+P

The program reads the time data from the RTC (Clock Device) when X03 is on. Then adds 1 hour to the data and stores the added time data to D43.

Ladder Diagram (LD)

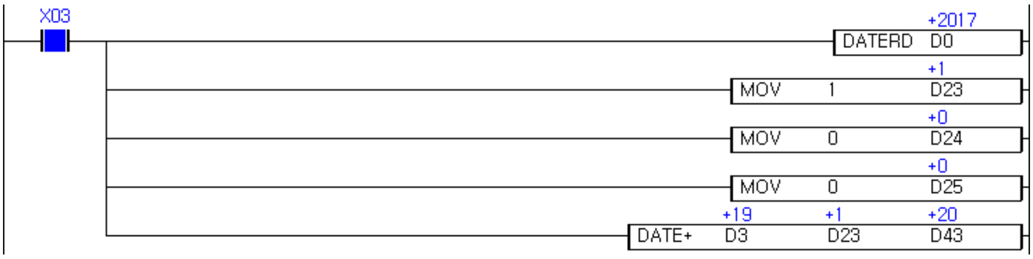


Instruction List (IL)

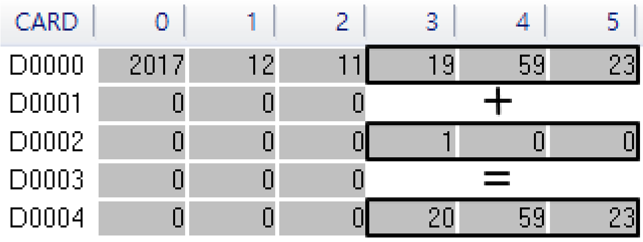
Instruction	Device		
LD	X03		
DATERD	D0		
MOV	1	D23	
MOV	0	D24	
MOV	0	D25	
DATE+	D3	D23	D43

The program operates as following:

- When X03 is ON, the program reads time data from the RTC (Clock Device) and stores it at the device starting from D0. Then the program input “1” in D23, “0” in D24 and D25.
- DATE+ instruction operates and the added time is stored at the device starting from D43.



```
 X03                                                        +2017
─┤■├──┬──────────────────────────────────────────[DATERD   D0      ]
      │                                                     +1
      ├─────────────────────────────────[MOV      1         D23     ]
      │                                                     +0
      ├─────────────────────────────────[MOV      0         D24     ]
      │                                                     +0
      ├─────────────────────────────────[MOV      0         D25     ]
      │                       +19        +1                 +20
      └──────────────────[DATE+   D3         D23            D43     ]
```



CARD	0	1	2	3	4	5
D0000	2017	12	11	19	59	23
D0001	0	0	0		+	
D0002	0	0	0	1	0	0
D0003	0	0	0		=	
D0004	0	0	0	20	59	23

2.12.4 DATE-, DATE-P

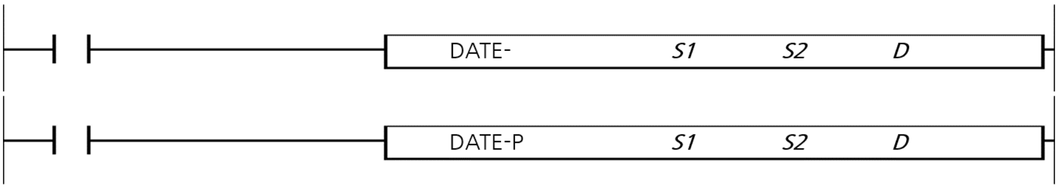
Supported PLC Series

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

DATE- and DATE-P subtract a time data assigned to *S2* from time data assigned to *S1*. The result is stored in a word device.

Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
DATE- DATE-P	*S1*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	4	O	-	-
	S2	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



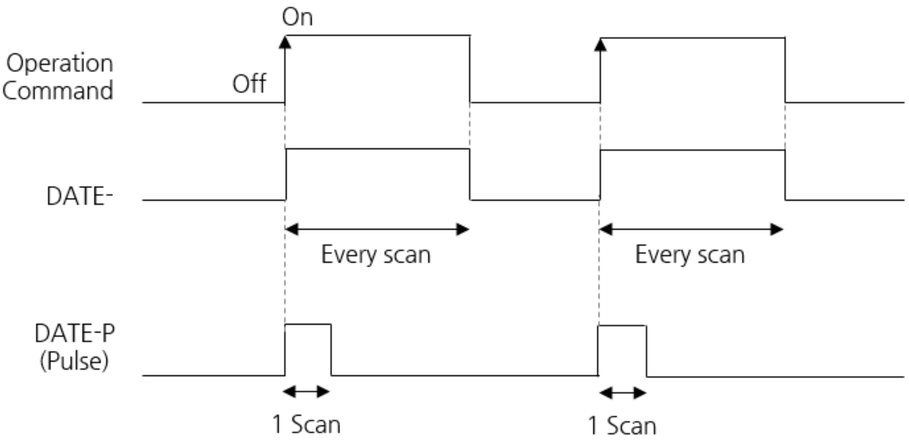
S1	Head address of word devices where the time data to be subtracted is stored.
S2	Head address of word devices where the time data for subtraction is stored.
D	Head address of word devices where the subtracted data will be stored.

DATE-, DATE-P

- The instruction subtracts *S2* from *S1*. *S2* is a time data which is subtracted from *S1*. *S1* is a time data which is subtracted.
- The subtracted time data is stored in the word device starting from *D*.
- Time data of *S1* and *S2* should exist in order of hour (0 - 23), minute (0 - 59), second (0 - 59).
- If subtracted time data is negative, 24 hours will be added to the result.

		Data Range				Data Range				Data Range
S1	Hour	0-23		***S2***	Hour	0-23		***D***	Hour	0-23
S1+1	Minute	0-59	-	***S2+1***	Minute	0-59	=	***D+1***	Minute	0-59
S1+2	Second	0-59		***S2+2***	Second	0-59		***D+2***	Second	0-59

Execution Condition



Operation Error

Error Flag (F110)

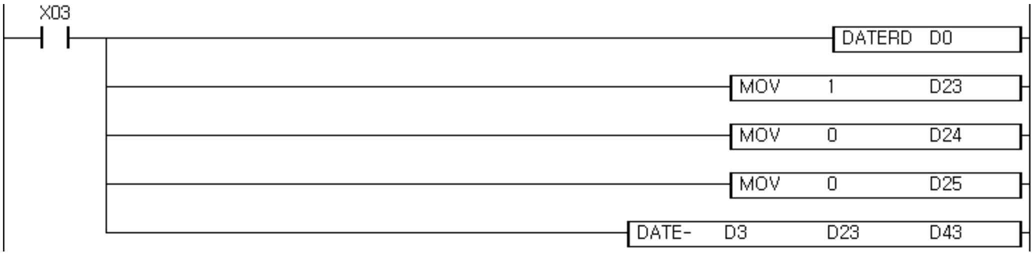
F110 turns ON for 1 scan when the address of device assigned to @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DATE-, DATE-P

The program reads the time data from the RTC (Clock Device) when X03 is on. Then subtracts 1 hour from the data and stores it to the D43.

Ladder Diagram (LD)

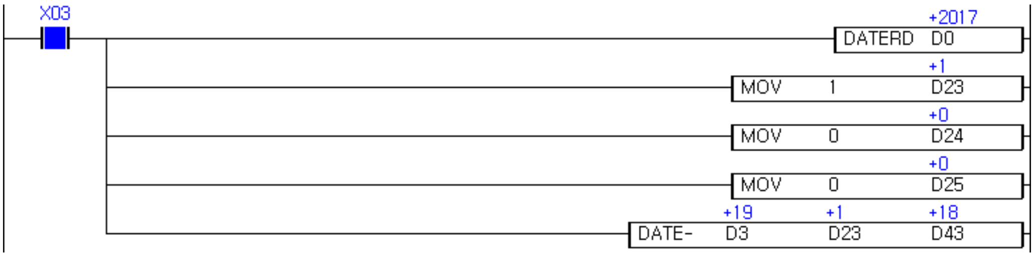


Instruction List (IL)

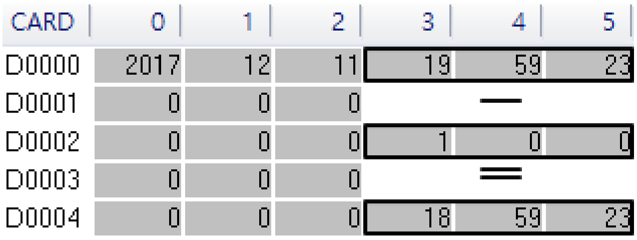
Instruction	Device		
LD	X03		
DATERD	D0		
MOV	1	D23	
MOV	0	D24	
MOV	0	D25	
DATE-	D3	D23	D43

The program operates as following:

- When X03 is ON, the program reads time data from the RTC (Clock Device) and stores it at the device starting from D0. Then the program input "1" in D23 and "0" in D24 and D25.
- DATE- instruction operates and the subtracted time data is stored starting from D43.



```
X03                                  +2017
 |                          DATERD   D0
 |                                   +1
 |                   MOV     1       D23
 |                                   +0
 |                   MOV     0       D24
 |                                   +0
 |                   MOV     0       D25
 |          +19      +1              +18
 |  DATE-   D3       D23             D43
```



CARD	0	1	2	3	4	5
D0000	2017	12	11	19	59	23
D0001	0	0	0		—	
D0002	0	0	0	1	0	0
D0003	0	0	0		=	
D0004	0	0	0	18	59	23

2.12.5 SECOND, SECONDP

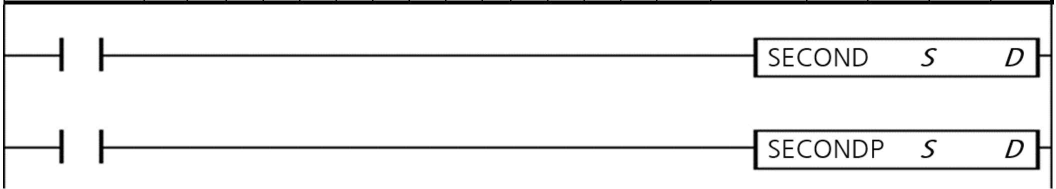
Supported PLC Series

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

SECOND and SECONP converts time data into second-format data.

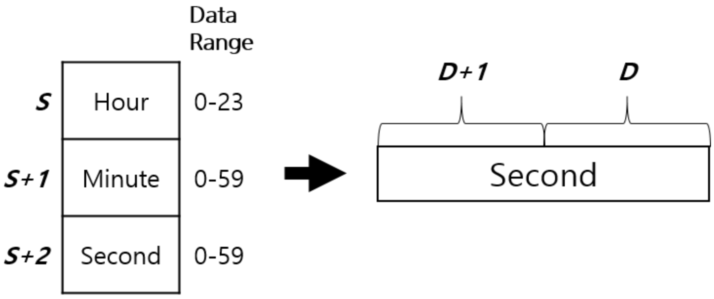
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SECOND(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



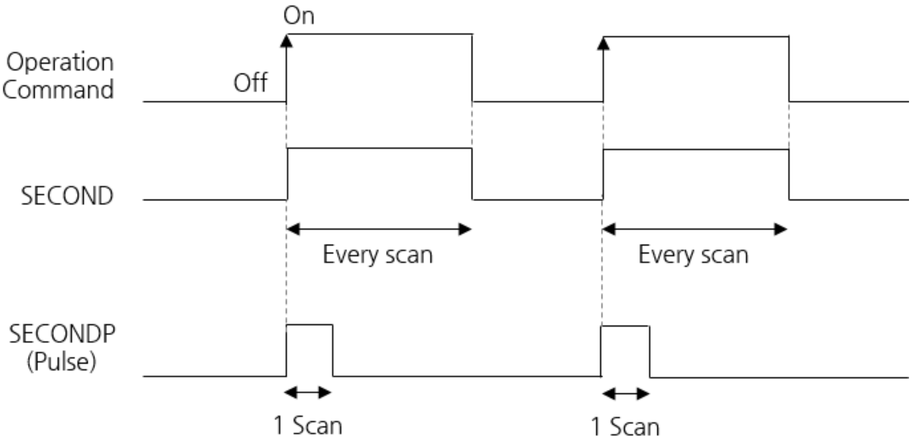
S	Head address of word devices where the time data to be converted is stored.
D	Head address of double word device where the converted time data will be stored.

SECOND, SECONP

- The instruction converts the time data (hour, minute, second) into second-format data.
- Time data is specified by the head address of word devices assigned to *S*.
- After the execution, specified time data is converted into seconds.
- Converted time data in seconds is stored in the double word device assigned to *D*.
- The range of conversion result is from 0 to 86399.



Execution Condition



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)

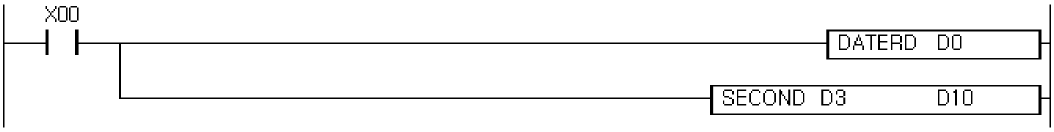
If the value of the time data to convert exceeds the data range, F110 turns ON for 1 scan.

Program Example

SECOND, SECONDP

When X00 is ON, this program converts time data (D3 - D5) into seconds. After the conversion, the result is stored in D10 and D11.

Ladder Diagram (LD)

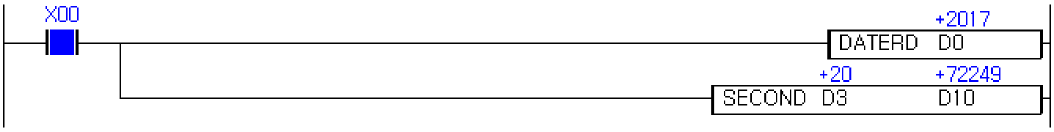


Instruction List (IL)

Instruction	Device	
LD	X00	
DATERD	D0	
SECOND	D3	D10

The program operates as following:

- When X00 is ON, the program reads time data from the RTC (Clock Device) and stores it starting from D0. Then the time data (D3 - D5) is converted into seconds.
- The converted time data is stored in D10 and D11.



2.12.6 HOUR, HOURP

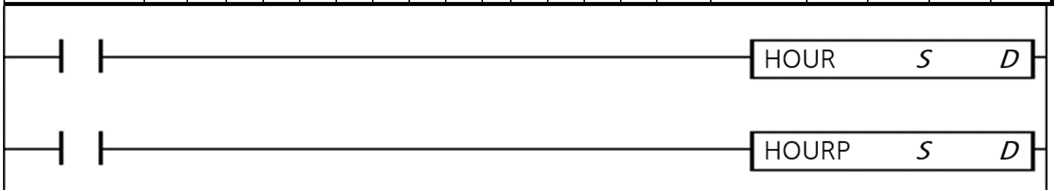
Supported PLC Series

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

HOUR, HOURP convert second-format data into time data (hour, minute, second format).

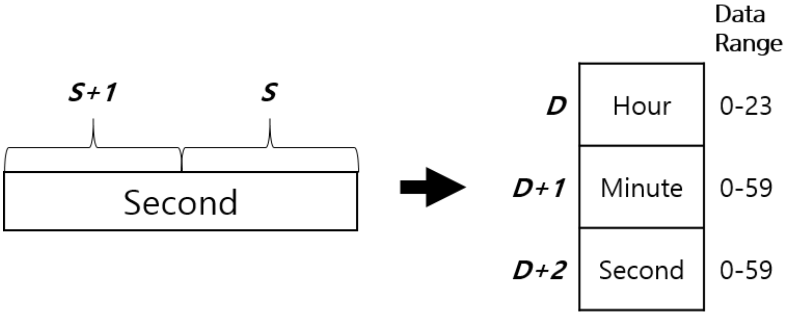
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
HOUR(P)	*S*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	-	3	O	-	-
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				



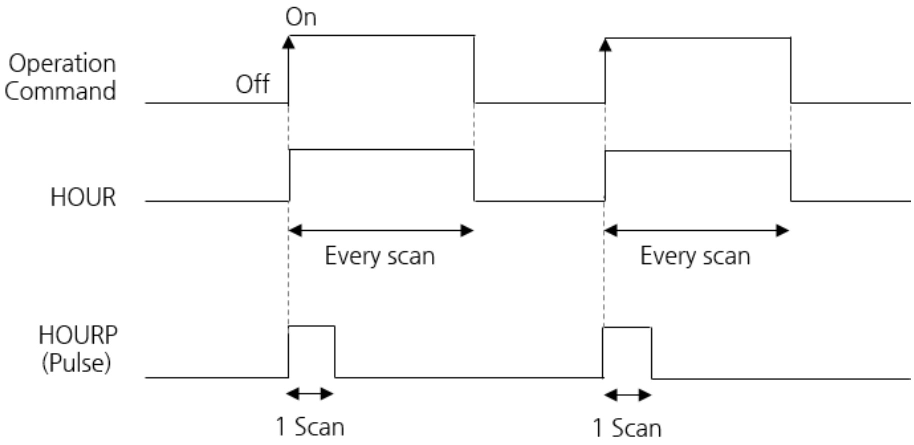
S	Head address of double word device where the time data to be converted is stored.
D	Head address of word devices where the converted time data will be stored.

HOUR, HOURP

- The instruction converts the second-format data into time data (hour, minute, second).
- Second-format data is specified by the head address of double word device assigned to *S*.
- After the execution, specified second-format data is converted into time data.
- Converted time data is stored in the word device assigned to *D*.
- You can input second-format data from 0 to 86399.



Execution Condition



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)

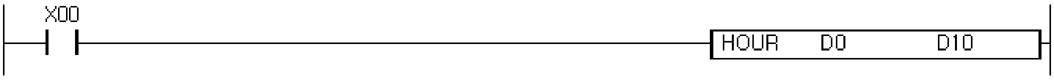
F110 turns ON for 1 scan when the time data to convert exceeds the range of 0~86399.

Program Example

HOUR, HOURP

When X00 is ON, this program converts second-format data D0 into time data. After the conversion, the result is stored starting from D10.

Ladder Diagram (LD)

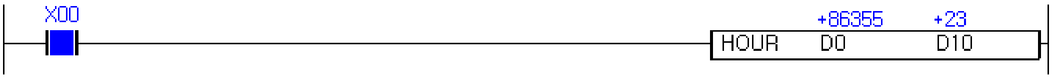


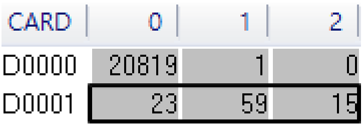
Instruction List (IL)

Instruction	Device	
LD	X00	
HOUR	D0	D10

The program operates as following:

- When X00 is turned ON and D0 and D1 is “86355”, the program converts the second-format data into time data.





CARD	0	1	2
D0000	20819	1	0
D0001	23	59	15

2.13 Buffer Memory Access Instruction

2.13.1 FROM, FROMP, DFRO, DFROP

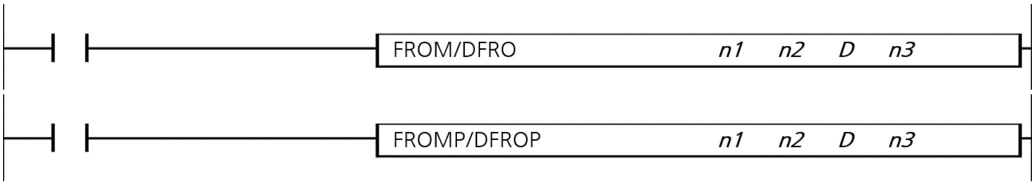
Supported PLC Series

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

FROM(P), DFRO(P) read data from buffer memory of a special module and store the data in the designated device memory.

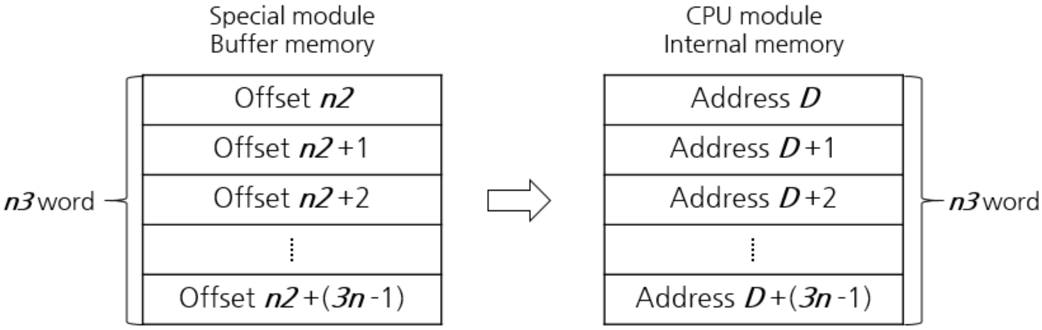
Instruction		Valid device type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FROM(P) DFRO(P)	*n1*	O	O	O	O	O	O	O	O	O	O	O	O	O	-	O	5	O	-	-
	n2	O	O	O	O	O	O	O	O	O	O	O	O	O	-	O				
	D	O	-	O	O	O	-	O	O	-	O	-	O	O	O	-				
	n3	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				



n1	Base and slot number of the special module in hexadecimal
n2	Offset of buffer memory to read word data
D	Head address of word devices where the word data to be read is stored.
n3	The size of word data to read. (0 - 255)

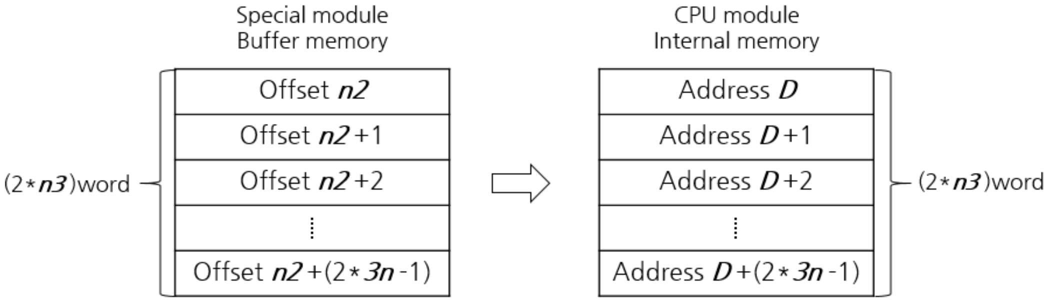
FROM, FROMP

- FROM and FROMP read data in *n3* words starting from offset of buffer memory assigned to *n2*.
- The buffer memory is located at a special module which is assigned to *n1*.
- The data read from the buffer memory is stored in the device assigned to *D* in *n3* words.



DFRO, DFROP

- DFRO and DFROP read data in (2∗*n3*) words starting from offset of buffer memory assigned to *n2*.
- The buffer memory is located at a special module which is assigned to *n1*.
- The data read from the buffer memory is stored in the device assigned to *D* in (2∗*n3*) words.



Assigning *n1* - [Base/Slot]

	Base No.	Slot No.
H	0A	0B

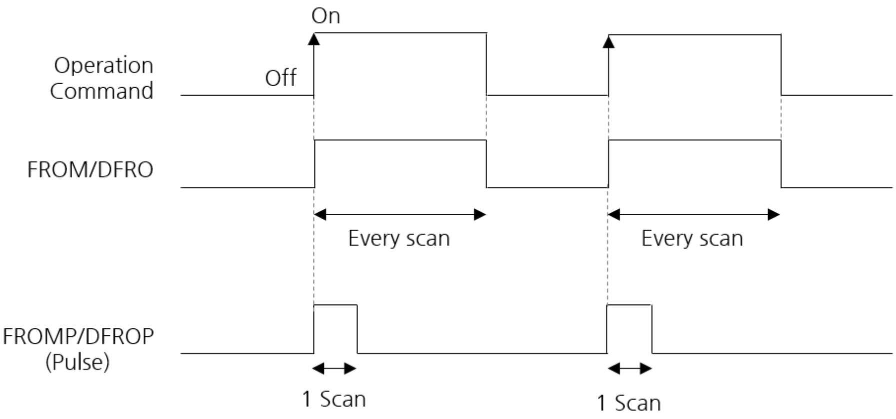
H + [Base No.] + [Slot No.]
H: Stands for hexadecimal.
Base No.: 2 characters in hexadecimal
Slot No.: 2 characters in hexadecimal

Base Number	Slot	S1
Local base	Slot No. 5	H0005 or 5
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 12	H0E0C
16th expansion	Slot No. 10	H100A

Assigning *n2* - [Buffer memory offset]

Offset of data to read which is stored at buffer memory. In other words, *n2* indicates the location of the data to be read. The offset of buffer memory is from 0 to 255.
(∗) Refer to the Buffer Memory Assignment of each special module.

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when:

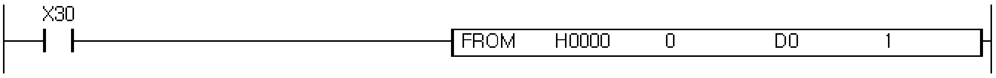
- The address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)
- Access to the special module fails.
- The base and slot number assigned to *n1* is not special module.
- The number of data, *n3,* exceeds the range of device assigned to *D*.

Program Example

FROM, FROMP

This program reads 1 word and stores the data in D0 when X30 is ON. The data is read from buffer memory offset 0 of special module equipped at slot 0 of local base.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device			
LD	X30			
FROM	H0000	0	D0	1

The program operates as following:

- When X30 is ON, the program specifies the location of data. The data is located in buffer memory offset 0 of special module which is equipped at slot 0 of local base.
- The size of the data to be read is 1 word.
- 1-word data which is read from the specified location is stored in D0.

2.13.2 TO, TOP, DTO, DTOP

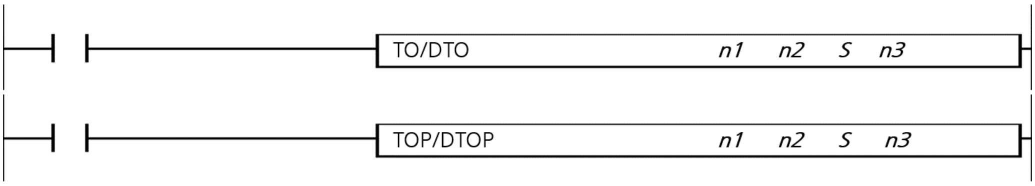
Supported PLC Series

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

TO(P) and DTO(P) write data from internal memory of CPU to buffer memory of a special module.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
TO(P) DTO(P)	*n1*	O	O	O	O	O	O	O	O	O	O	O	O	O	-	O	5	O	-	-
	n2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	S	O	-	O	O	O	O	O	O	-	O	-	O	O	O	O				
	n3	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				

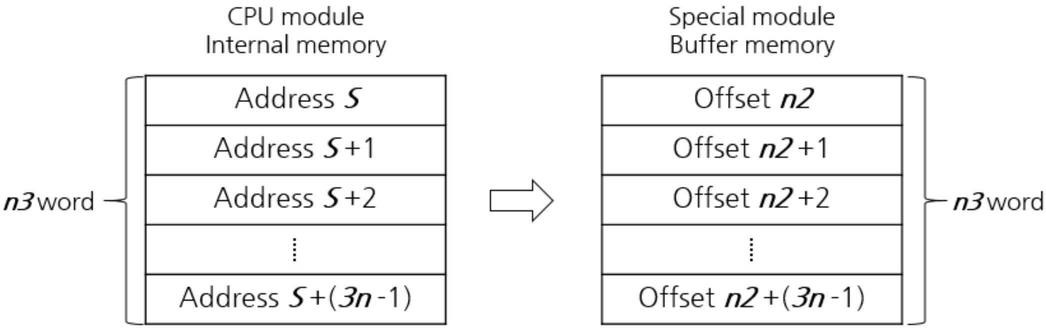


n1	Base and slot number of the special module in hexadecimal.
n2	Offset of buffer memory where the word data will be written
S	Head address of device memory where the word data to be written is stored.
n3	The size of word data to be written. (0 – 255)

TO, TOP

- TO and TOP write *n3* word data to offset of buffer memory assigned to *n2*.
- The buffer memory is located at a special module which is equipped at *n1*.
- The data to be written is stored in the device memory assigned to *S*.

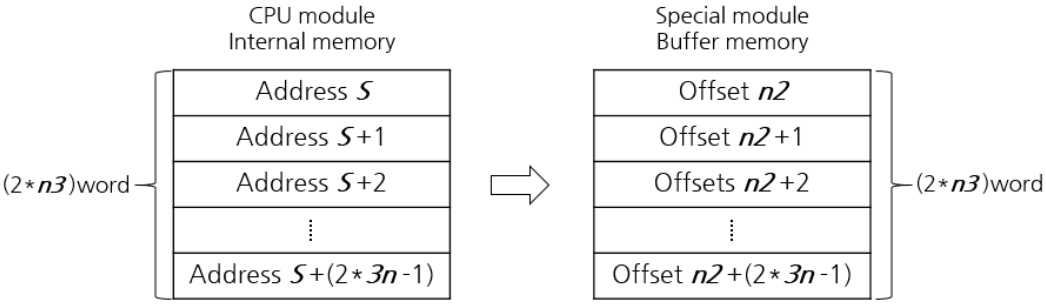
⚠ Using pulse contact as execution condition of TO instruction or using TOP instruction is recommended.



DTO, DTOP

- DTO and DTOP write (2**n3*) word data to offset of buffer memory assigned to *n2*.
- The buffer memory is located at a special module which is equipped at *n1*.
- The data to be written is stored in the device memory assigned to *S*.

⚠ Using pulse contact as execution condition of DTO instruction or using DTOP instruction is recommended.



Assigning *n1* - [Base/Slot]

	Base No.	Slot No.
H	0A	0B

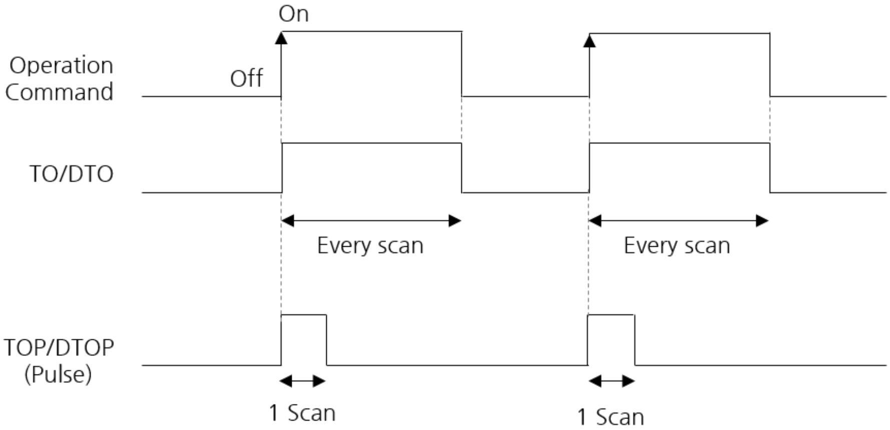
H + [Base No.] + [Slot No.]
H: Stands for hexadecimal.
Base No.: 2 characters in hexadecimal
Slot No.: 2 characters in hexadecimal

Base Number	Slot	*n1*
Local base	Slot No. 5	H0005 or 5
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 12	H0E0C
16th expansion	Slot No. 10	H100A

Assigning *n2* - [Buffer memory offset]

Offset of data to write at the buffer memory. In other words, *n2* indicates the location of the data to be written.
(*) Refer to the Buffer Memory Assignment of each special module.

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when:

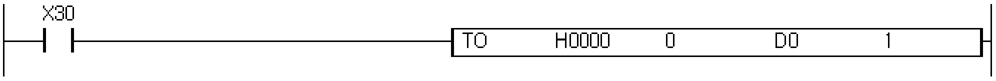
- The address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)
- Access to the special module fails.
- The base and slot number assigned to *n1* is not special module.
- The number of data, *n3,* exceeds the range of device assigned to *S*.

Program Example

TO, TOP

This program writes 1-word data which is located in D0, to the buffer memory when X30 is ON. The data is written to buffer memory offset 0 of special module equipped at slot 0 of local base.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device			
LD	X30			
TO	H0000	0	D0	1

The program operates as following:

- When X30 is ON, the program specifies the location of data. The data is located in D0.
- The size of the data to be written is 1 word.
- 1-word data is written to buffer memory offset 0 of special module which is equipped at slot 0 of local base.

2.14 Data Link Instruction

2.14.1 SND, SNDP

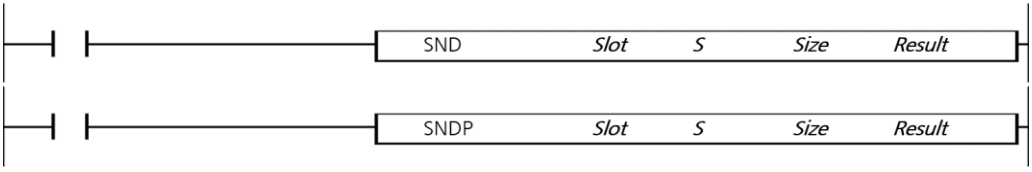
Supported PLC Series

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

SND(P) instructions send a frame starting from *S*, in byte unit assigned to ***size***. ***Slot*** specifies communication module and the port with the assigned value. The result of sending data are stored in the device assigned to ***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
SND(P)	*Slot*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	5	O	-	-
	S	O	O	O	O	O	-	O	O	-	O	-	O	O	O	-				
	Size	O	O	O	O	O	-	O	O	-	O	-	O	O	O	O				
	Result	O	O	O	O	O	-	O	O	-	O	-	O	O	O	-				



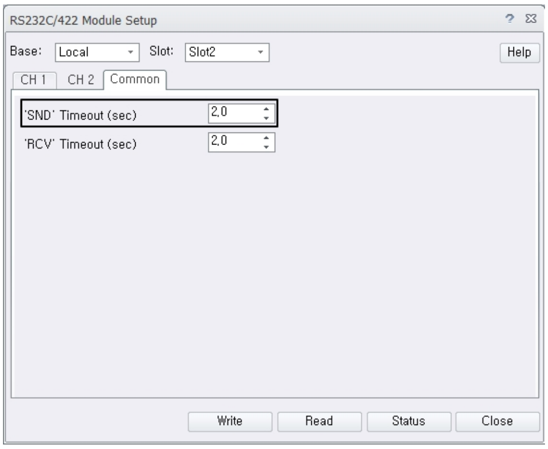
Slot	Constant to specify the slot number of communication module and the port.
S	Head address of word device where data to be sent are stored.
Size	Constant or head address of word device where the data in byte size to be sent is stored.
Result	Address of word device where the result of sending data will be stored.

SND, SDNP

- SND(P) instructions send a frame starting from *S*, in byte unit assigned to ***size***.
- SND(P) instructions support only serial communication. Supported serial communication modules are as following:

CM1 Series	CM3 Series		
SC02A	SP32MDT	SP32MDCF	SP02ERS
SC01A	SP32MDTV	SP16MDR	SP02ERR
SC01B	SP32MDTE	SP16MDRV	SP01ERC
SC02C	SP32MDTF	SP16MDRE	SP02ERSC
SC02CDMA	SP32MDC	SP16MDRF	SP02ERRC
	SP32MDCV	SB16MDT	
	SP32MDCE	SB16MDTV	

- ***Slot*** specifies the communication module and the port with the assigned value.
- The result of sending data is stored in the word device assigned to ***Result***.
- When the instruction is executed, the number of bytes assigned to ***size***, starting from the device *S* are sent.
- When you use SND(P) instructions, you can set timeout at [Online] - [Special Module Setup] - [RS232C/422 Module...] - [Common] tab. Refer to the figure below.



Base: Local
Slot: Slot2
'SND' Timeout (sec) 2.0
'RCV' Timeout (sec) 2.0

Assigning *Slot* - [Base/Slot and communication port]

	Base No.	Communication port	Slot No.
H	0A	1	B

H + [Base No.] + [Communication port] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Communication port: 1 digit in hexadecimal. (0: Channel 1, 1: Channel 2)
Slot No.: 1 digit in hexadecimal

Base Number	Communication port	Slot	*Slot*
Local base	Channel 1	Slot No. 5	H0005
1st expansion	Channel 1	Slot No. 3	H0103
10th expansion	Channel 1	Slot No. 7	H0A07
14th expansion	Channel 2	Slot No. 12	H0D1C
16th expansion	Channel 2	Slot No. 10	H101A

Assigning *S*

Assign the head address of word device where the frame to be sent is stored. The specified frame will be sent.

Assigning *Size*

Decides the size of data to send, starting from the location of frame assigned to *S*.
The range of data length is from 1 byte to 500 bytes (= 250 words).
If *Size* exceeds its range, the instruction will not be executed.
Size can only be assigned in byte unit.

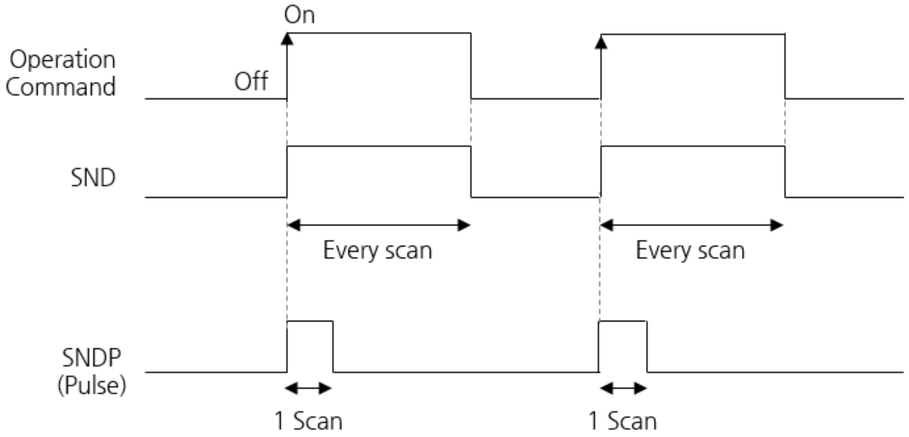
Assigning *Result*

When the frame is sent, the result of data transmission will be stored in the device assigned to *Result*.
Result format is as following:

Bit 0	Transmission completed: ON (1 scan) Transmission failed: ON (Always)
Bit 1	Transmission failed: ON (Always)
Bit 2 - 7	OFF
Bit 8 - F	Error code (0 = No error)

(*) For the information of error code, refer to [Communication module RS232C/422/485] at [Help].

Execution Condition



⚠ You have to use pulse contact as execution condition of SND instruction or use SNDP instruction to prevent data transmission error.

Operation Error

Error flag (F110)

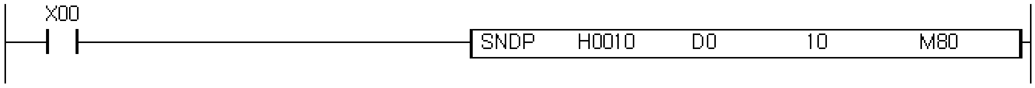
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

SNDP

When operation command X00 is ON, SNDP instruction is executed. Then, 10-byte data (5 words) starting from D0 (D0~D4) is sent. The data is sent through channel 2 of communication module equipped at slot 0 of local base (= H0010). The result of sending data is stored in M80 (M80~M8F).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device			
LD	X00			
SNDP	H0010	D0	10	M80

The program operates as following:

- When X00 is ON, the instruction checks whether communication module is equipped at slot 0 of local base.
- Then, the instruction specifies the location and length of data. The data to be sent is located in D0 ~ D4 (= 10 bytes).
- After the data transmission, the result is stored in M80 (M80~M8F).

2.14.2 RCV, RCVP

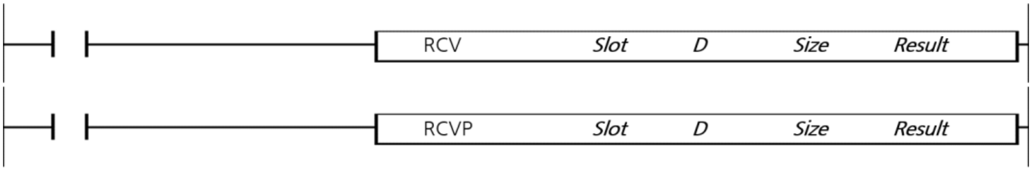
Supported PLC Series

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

RCV(P) instructions receive and store a frame in device starting from *D*, in byte unit assigned to *size*. *Slot* specifies communication module and the port with the assigned value. The result of data reception is stored in the device assigned to *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
RCV(P)	*Slot*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O	5	O	-	-
	D	O	O	O	O	O	-	O	O	-	O	-	O	O	O	-				
	Size	O	O	O	O	O	-	O	O	-	O	-	O	O	O	O				
	Result	O	O	O	O	O	-	O	O	-	O	-	O	O	O	-				



Slot	Constant to specify the slot number of communication module and the port.
D	Head address of word device where received data are stored.
Size	Constant or head address of word device to decide the length of received data in byte size.
Result	Address of word device where the result of data reception will be stored.

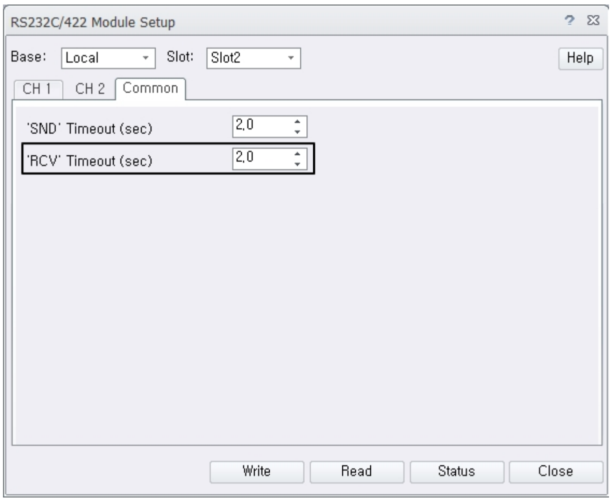
RCV, RCVP

- RCV(P) instructions receive and store a frame in device starting from *D*, in byte unit assigned to *size*.
- RCV(P) instructions support only serial communication. Supported serial communication modules are as following:

CM1 Series	CM3 Series		
SC02A	SP32MDT	SP32MDCF	SP02ERS
SC01A	SP32MDTV	SP16MDR	SP02ERR
SC01B	SP32MDTE	SP16MDRV	SP01ERC
SC02C	SP32MDTF	SP16MDRE	SP02ERSC
SC02CDMA	SP32MDC	SP16MDRF	SP02ERRC
	SP32MDCV	SB16MDT	
	SP32MDCE	SB16MDTV	

- *Slot* specifies the communication module and the port with the assigned value.
- The result of data reception is stored in the device assigned to *Result*.

- When you use RCV(P) instructions, you have to set timeout at [Online] - [Special Module Setup] - [RS232C/422 Module...] - [Common] tab. Refer to the figure below.



Base: Local
Slot: Slot2
'SND' Timeout (sec) 2.0
'RCV' Timeout (sec) 2.0

Assigning *Slot* - [Base/Slot and communication port]

	Base No.	Communication port	Slot No.
H	0A	1	B

H + [Base No.] + [Communication port] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Communication port: 1 digit in hexadecimal. (0: Channel 1, 1: Channel 2)
Slot No.: 1 digit in hexadecimal

Base Number	Communication port	Slot	*Slot*
Local base	Channel 1	Slot No. 5	H0005
1st expansion	Channel 1	Slot No. 3	H0103
10th expansion	Channel 1	Slot No. 7	H0A07
14th expansion	Channel 2	Slot No. 12	H0D1C
16th expansion	Channel 2	Slot No. 10	H101A

Assigning *D*

Assign the head address of word device where the received frame to be stored.

Assigning *Size*

Decides how many data to receive and store the data starting from the device *D*.
The range of data length is from 1 byte to 500 bytes (= 250 words).
If *Size* exceeds its range, the instruction will not be executed.
Size can only be assigned in byte unit.

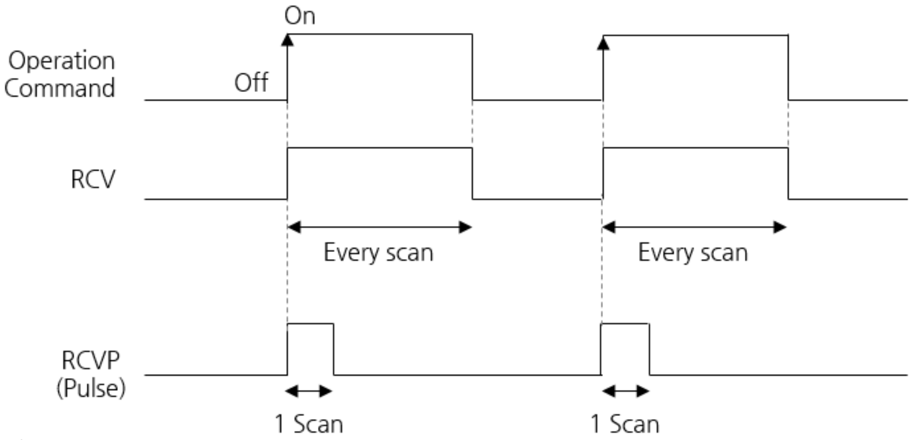
Assigning *Result*

When the frame is received, the result of data reception will be stored in the device assigned to *Result*.
Result format is as following:

Bit 0	Reception completed: ON (1 scan) Reception failed: ON (Always)
Bit 1	Reception failed: ON (Always)
Bit 2 - 7	OFF
Bit 8 - F	Error code (0 = No error)

(*) For the information of error code, refer to [Communication module RS232C/422/485] at [Help].

Execution Condition



⚠ You have to use pulse contact as execution condition of RCV instruction or use RCVP instruction to prevent data reception error.

Operation Error

Error flag (F110)

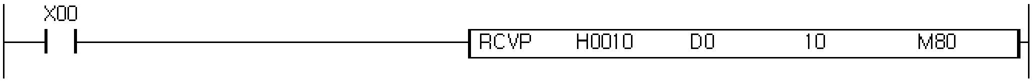
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

RCVP

When operation command X00 is ON, RCVP instruction is executed. 10-byte data (5 words) will be received and stored in the device starting from D0 (D0~D4). The data is received through channel 2 of communication module equipped at slot 0 of local base (= H0010). The result of data reception is stored in M80 (M80~M8F).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device			
LD	X00			
RCVP	H0010	D0	10	M80

The program operates as following:

- When X00 is ON, the instruction checks whether communication module is equipped at slot 0 of local base.
- Then, the instruction specifies the location and length of data. The data will be received in D0 ~ D4 (= 10 bytes).
- After the data transmission, the result is stored in M80 (M80~M8F).

2.14.3 SEND, SENDP

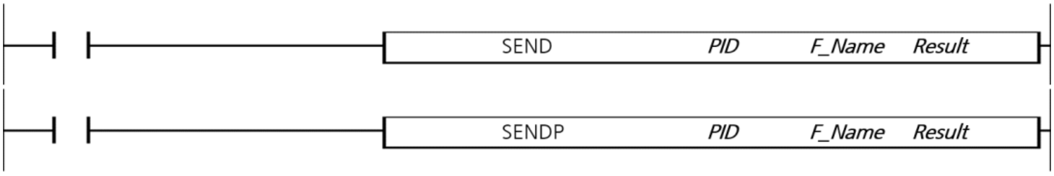
Supported PLC Series

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

SEND(P) instructions send a user defined frame. User defined frame to send is specified by the value assigned to *PID* and *F_Name*. After sending a user defined frame, the result is stored in the word device assigned to *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
SEND(P)	*PID*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	F_Name	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	Result	O	O	O	O	O	-	-	-	-	O	-	O	O	O	-				



PID	Program ID of communication configuration program where frame is stored.
F_Name	Constant or address of word device to specify the frame and communication port(channel).
Result	Address of word device where the result of sending data will be stored.

SEND, SENDP

- SEND(P) instructions send a user defined frame.
- User defined frame is specified by the value assigned to *PID* and *F_Name*.
- When the user defined frames are sent, the result of sending data is stored in the word device assigned to *Result*.
- To use SEND(P) instructions, communication configuration program and TX[4] frames have to exist in the project. If not, data transmission fails.
- SEND(P) instructions support serial (RS232C, RS422, RS485) and ethernet communications. Supported communication modules are as following:

- Ethernet

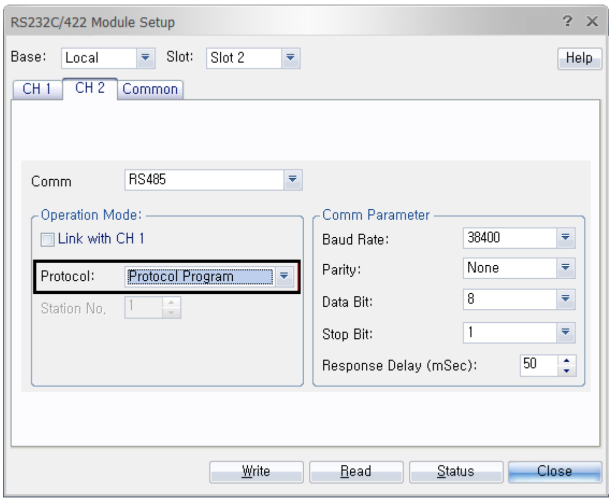
CM1 Series	CM3 Series		
EC01A	SP32MDT	SP32MDCF	SP01EET
EC10A	SP32MDTV	SP16MDR	
EC10B	SP32MDTE	SP16MDRV	
EC10C	SP32MDTF	SP16MDRE	
	SP32MDC	SP16MDRF	
	SP32MDCV	SB16MDT	
	SP32MDCE	SB16MDTV	

[4] TX: Transmit

- Serial

CM1 Series	CM3 Series		
SC02A	SP32MDT	SP32MDCF	SP02ERS
SC01A	SP32MDTV	SP16MDR	SP02ERR
SC01B	SP32MDTE	SP16MDRV	SP01ERC
SC02C	SP32MDTF	SP16MDRE	SP02ERSC
SC02CDMA	SP32MDC	SP16MDRF	SP02ERRC
	SP32MDCV	SB16MDT	
	SP32MDCE	SB16MDTV	

- When you use serial communication, corresponding protocol has to be set up for the module at [Online] - [Special Module Setup] - [RS232C/422 Module...]. Following figure is an example.



RS232C/422 Module Setup

Base: Local Slot: Slot 2

Comm: RS485

Operation Mode:

☐ Link with CH 1

Protocol: Protocol Program

Station No. 1

Comm Parameter

Baud Rate: 38400

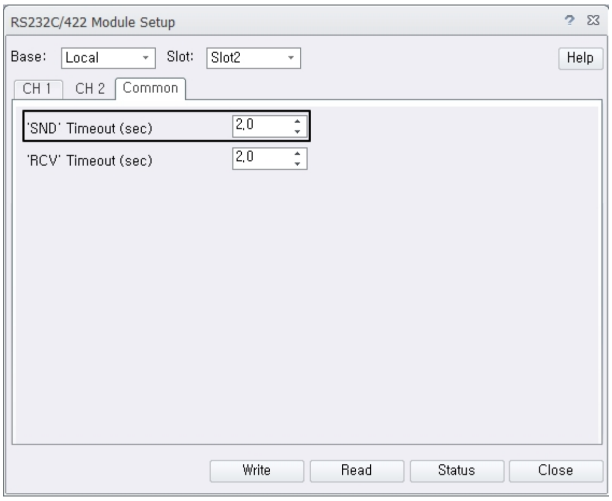
Parity: None

Data Bit: 8

Stop Bit: 1

Response Delay (mSec): 50

- When you use SEND(P) instructions, you can set timeout at [Online] - [Special Module Setup] - [RS232C/422 Module...] - [Common] tab. Refer to the figure below.



RS232C/422 Module Setup

Base: Local Slot: Slot2

'SND' Timeout (sec) 2.0

'RCV' Timeout (sec) 2.0

⚠ Do not use SEND(P) and RECV(P) instructions together within same scan time, in same communication module. Doing so may cause a deterioration.

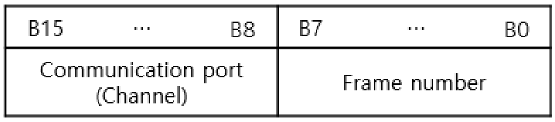
Assigning *PID*

Input constant or head address of word device to specify a program ID of communication configuration program. User defined frame will be sent from the program specified by *PID*.

Assigning *F_Name*

Decides which frame to be sent. Constant or a word device can be assigned. When you assign a frame number to *F_Name*, follow the format below.

Upper byte (Bit 8~15): Communication port (Channel 1: 0, Channel 2: 1, Ethernet: Host ID (0~7))
Lower byte (Bit 0~7): Frame number in communication configuration program.



B15 … B8	B7 … B0
Communication port (Channel)	Frame number

※ When SEND(P) instructions are used with Hybrid Xpanel, upper byte is always "H00" since it uses only channel 1.

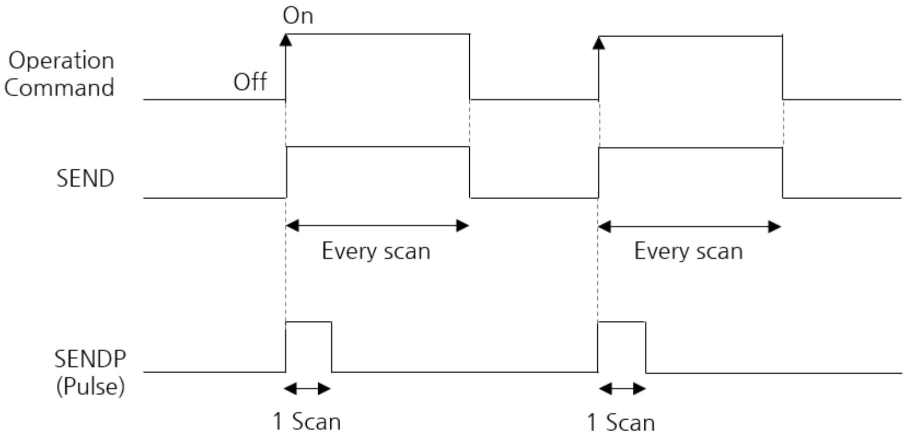
Assigning *Result*

When the frame is sent, the result of data transmission will be stored in the device assigned to *Result*.
Result format is as following:

Bit 0	Transmission completed: ON (1 scan)
Bit 1	Transmission failed: ON (Always)
Bit 2 - 7	OFF
Bit 8 - F	Error code (0 = No error)

(*) For the information of error code, refer to [Communication module RS232C/422/485] or [Ethernet communication] at [Help].

Execution Condition



⚠ You have to use pulse contact as execution condition of SEND instruction or use SENDP instruction to prevent data transmission error.

Operation Error

Error flag (F110)

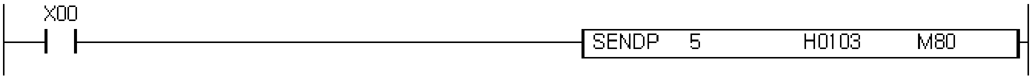
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

SENDP

When operation command X00 is ON, SENDP instruction is executed. Then, frame number “3” in PID 5 will be sent through channel 2. The result of sending data is stored in word device M80 (M80~M8F).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	X00		
SENDP	5	H0103	M80

The program operates as following:

- When X00 is ON, the instruction specifies the frame to be sent. Frame to be sent is located in program ID 5.
- The frame number “3” is sent through the channel 2 (=H0103).
- After the data transmission, the result is stored in M80 (M80~M8F).

2.14.4 RECV, RECVP

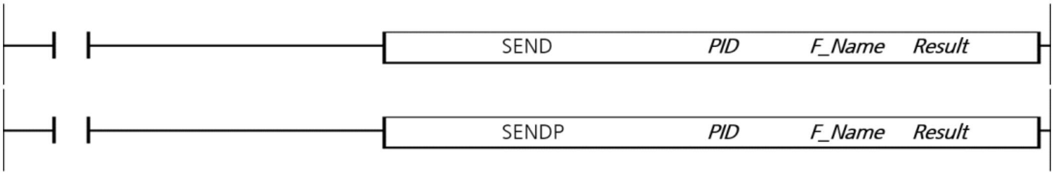
Supported PLC Series

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

RECV(P) instructions receive a user defined frame. User defined frame to receive is specified by the value assigned to *PID* and *F_Name*. When a user defined frame is received, the reception result is stored in the word device assigned to *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
RECV(P)	*PID*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	F_Name	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	Result	O	O	O	O	O	-	-	-	-	O	-	O	O	O	-				



PID	Program ID of communication configuration program where frame is stored.
F_Name	Constant or address of word device to specify the frame and communication port (channel).
Result	Address of word device where the result of data reception will be stored.

RECV, RECVP

- RECV(P) instructions receive a user defined frame.
- A user defined frame to receive are specified by the *PID* and *F_Name*.
- When a user defined frame is received, the result of data reception is stored in the word device assigned to *Result*.
- To use RECV(P) instructions, communication configuration program and RX[5] frames have to exist in the project. If not, data reception fails.
- RECV(P) instructions support serial (RS232C, RS422, RS485) and ethernet communications. Supported communication modules are as following:
  - Ethernet

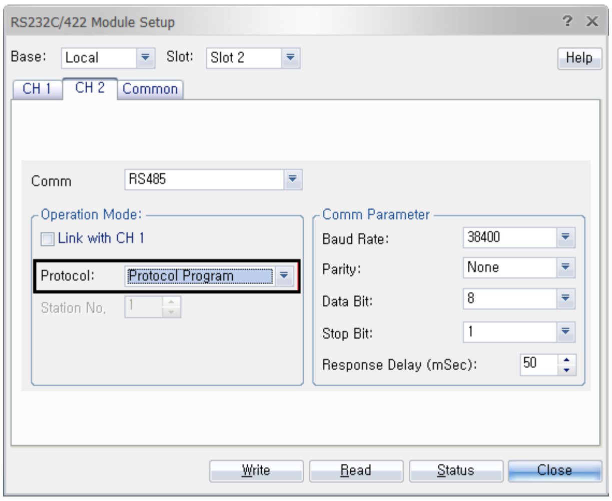
CM1 Series	CM3 Series		
EC01A	SP32MDT	SP32MDCF	SP01EET
EC10A	SP32MDTV	SP16MDR	
EC10B	SP32MDTE	SP16MDRV	
EC10C	SP32MDTF	SP16MDRE	
	SP32MDC	SP16MDRF	
	SP32MDCV	SB16MDT	
	SP32MDCE	SB16MDTV	

[5] RX: Receive

- Serial

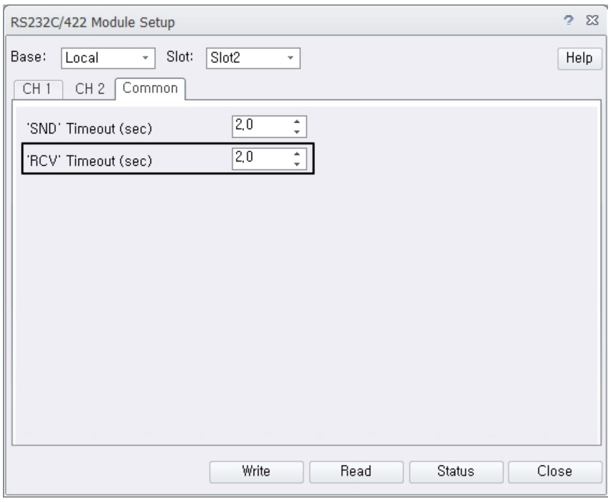
CM1 Series	CM3 Series		
SC02A	SP32MDT	SP32MDCF	SP02ERS
SC01A	SP32MDTV	SP16MDR	SP02ERR
SC01B	SP32MDTE	SP16MDRV	SP01ERC
SC02C	SP32MDTF	SP16MDRE	SP02ERSC
SC02CDMA	SP32MDC	SP16MDRF	SP02ERRC
	SP32MDCV	SB16MDT	
	SP32MDCE	SB16MDTV	

- When you use serial communication, corresponding protocol has to be set up for the module at [Online] - [Special Module Setup] - [RS232C/422 Module...]. Following is an example.



Base: Local
Slot: Slot 2
Comm RS485
□ Link with CH 1
Protocol: Protocol Program
Station No. 1
Baud Rate: 38400
Parity: None
Data Bit: 8
Stop Bit: 1
Response Delay (mSec): 50

- When you use RECV(P) instructions, you have to set timeout at [Online] - [Special Module Setup] - [RS232C/422 Module...] - [Common] tab. Refer to the figure below.



Base: Local
Slot: Slot2
'SND' Timeout (sec) 2.0
'RCV' Timeout (sec) 2.0

⚠ It is not recommended to use RECV(P) and SEND(P) instructions together within same scan time, in same communication module. Not doing so may cause a deterioration.

Assigning *PID*

Input constant or address of word device to specify a program ID of communication configuration program. The user defined frame will be received in the program specified by *PID*.

Assigning *F_Name*

Decides the frame to receive. Constant, hexadecimal or address of a word device can be assigned. When you assign a frame number to *F_Name*, follow the format below.

Upper byte (Bit 8~15): Communication port (Channel 1: 0, Channel 2: 1, Ethernet: Host ID (0~7))
Lower byte (Bit 0~7): Frame number in communication configuration program.

B15 ··· B8	B7 ··· B0
Communication port (Channel)	Frame number

※ When RECV(P) instructions are used with Hybrid Xpanel, upper byte is always "H00" since it uses only channel 1.

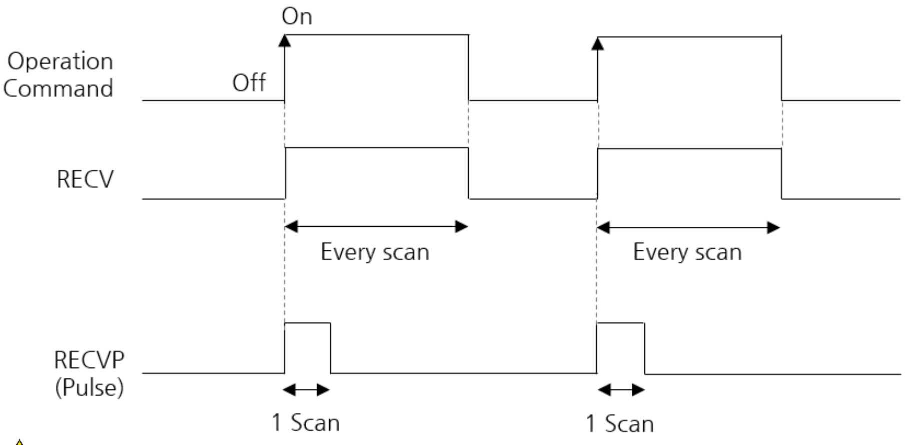
Assigning *Result*

When a frame is received, the result of data reception will be stored in the device assigned to *Result*.
Result format is as following:

Bit 0	Reception completed: ON (1 scan)
Bit 1	Reception failed: ON (Always)
Bit 2 - 7	OFF
Bit 8 - F	Error code (0 = No error)

(*) For the information of error code, refer to [Communication module RS232C/422/485] or [Ethernet communication] at [Help].

Execution Condition



⚠ You have to use pulse contact as execution condition of RECV instruction or use RECVP instruction to prevent data reception error.

Operation Error

Error flag (F110)

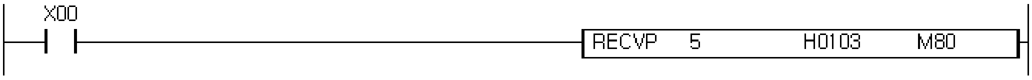
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

RECVP

When operation command X00 is ON, RECVP instruction is executed. The frame number “3” in PID 5 through channel 2. The result of data reception is stored in word device M80 (M80~M8F).

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	X00		
RECVP	5	H0103	M80

The program operates as following:

- When X00 is ON, the instruction specifies the frame to receive. The frame to receive is located in program ID 5.
- The frame number “3” is received through channel 2 (=H0103).
- After the data reception, the result is stored in M80 (M80~M8F).

2.15 Timer Instruction

2.15.1 TON

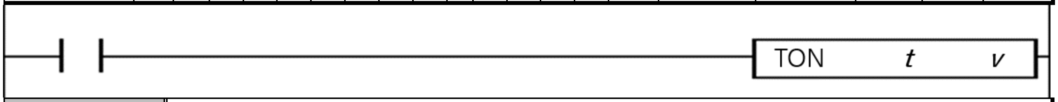
Supported PLC Series

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

TON turns ON a timer contact when the current value reaches a setting value.

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
TON	*t*	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	O	O	-	O				



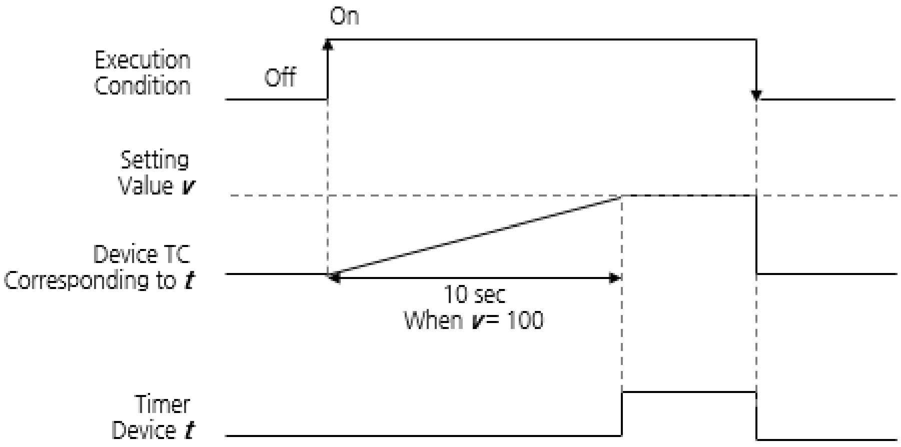
t	Address of timer device T. Device T is a bit device.
v	Setting value of timer.

TON

- Device T assigned to *t* is a bit device. There are device TC and TS to express the detailed value of device T assigned to *t*.
- Device TC is a word device to express the current time value of *t*.
- Device TS is a word device to express the time setting value of *t*. The value of device TS is equal to the value assigned to *v*.
- When execution condition is ON, the value of device TC which corresponds to the timer device *t* increases.
- When the value of device TC which corresponds to the timer device *t* reaches the setting value *v*, the device assigned to *t* turns ON.
- The actual time for timer device *t* is calculated as following:
  Actual time = *v* * time unit (100ms or 10ms)[6]
- From 0 to 65535 can be assigned to *v*.
- When the execution condition turns OFF, the corresponding device TC becomes “0” and the timer turns OFF.
- When the timer contact meets RST instruction, the corresponding device TC becomes “0” and the timer turns OFF.
- The timer contact assigned to *t* can be used as an A or B contact.

[6] You can configure time unit at [Tool] - [PLC Parameter] - [Basic] - [Timer].

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

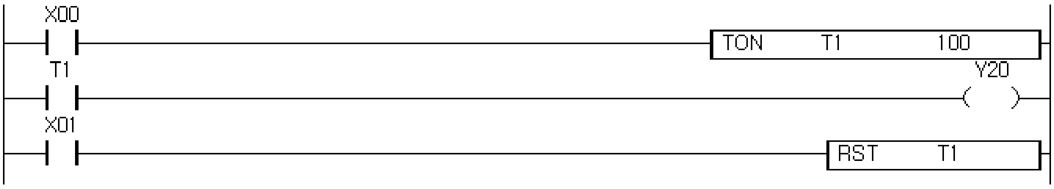
Program Example

TON

When X00 turns ON, the value of TC1 increases. TC1 is a word device which indicates the current time value of T1. When the value of TC1 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)

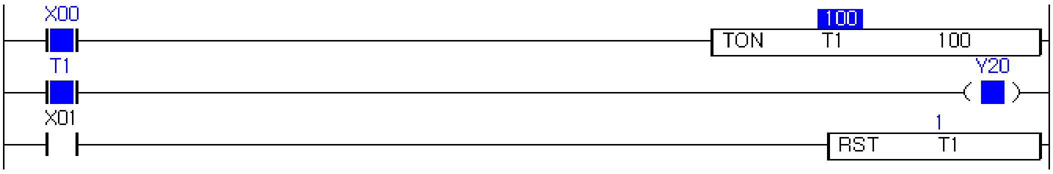


Instruction List (IL)

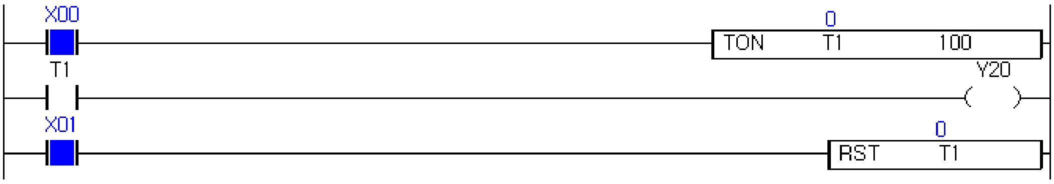
Instruction	Device	
LD	X00	
TON	T1	100
LD	T1	
OUT	Y20	
LD	X01	
RST	T1	

The program operates as following:

- When X00 is ON, the program starts timer device T1. The value of TC1 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 value of TC1 is reset.



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



2.15.2 TOFF

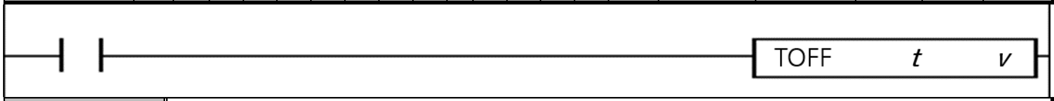
Supported PLC Series

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

TOFF turns ON the timer contact when execution condition is ON. When execution condition is OFF, current time value decreases and turns OFF a timer contact when the current time value becomes “0”.

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
TOFF	*t*	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	O	O	-	O				



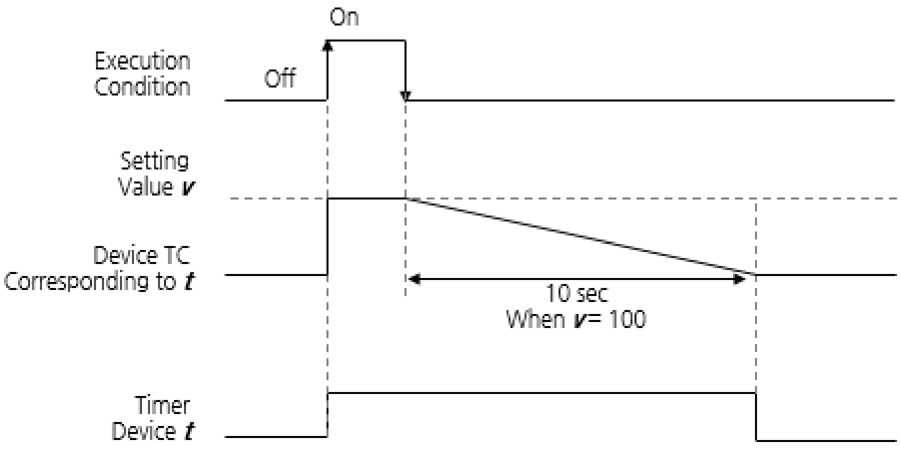
t	Address of timer device T. Device T is a bit device.
v	Setting value of timer.

TOFF

- Device T assigned to *t* is a bit device. There are device TC and TS to express the detailed value of device T assigned to *t*.
- Device TC is a word device to express the current time value of *t*.
- Device TS is a word device to express the time setting value of *t*. The value of device TS is equal to the value assigned to *v*.
- When the execution condition is ON, the value of device TC which corresponds to the timer device *t* becomes the setting value assigned to *v*. Also, the timer contact is turned ON.
- When the execution condition is OFF, the value of TC decreases.
- When the current value becomes “0”, the device assigned by *t* turns OFF.
- If execution condition is turned ON again before the value of TC becomes “0”, the value of TC becomes the setting value *v*.
- The actual time for timer device *t* is calculated as following:
  Actual time = *v* * time unit (100ms or 10ms)[7]
- From 0 to 65535 can be assigned to *v*.
- When the timer contact meets RST instruction, the value of device TC becomes “0” and the timer turns OFF.
- The timer contact assigned to *t* can be used as an A or B contact.

[7] You can configure time unit at [Tool] - [PLC Parameter] - [Basic] - [Timer].

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

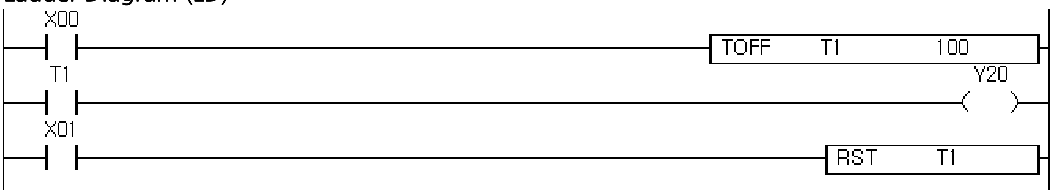
Program Example

TOFF

When X00 turns ON, the value of TC1 becomes the setting value assigned to *v* and T1 turns ON. When T1 turns ON, Y20 turns ON.
When X00 is turned OFF, the current value starts decreasing. When the value of TC1 becomes "0", T1 is turned OFF. When T1 turns OFF, Y20 turns OFF.

If X01 turns ON, RST instruction executes. Therefore, the current value becomes "0" and T1 is turned OFF.

Ladder Diagram (LD)

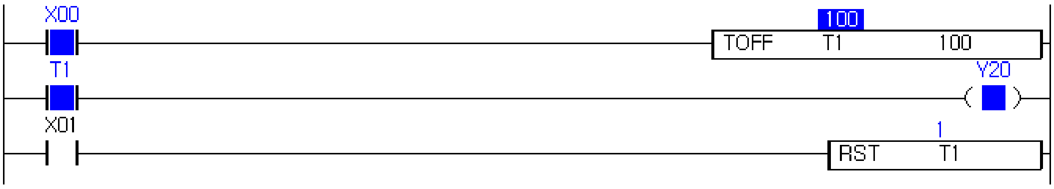


Instruction List (IL)

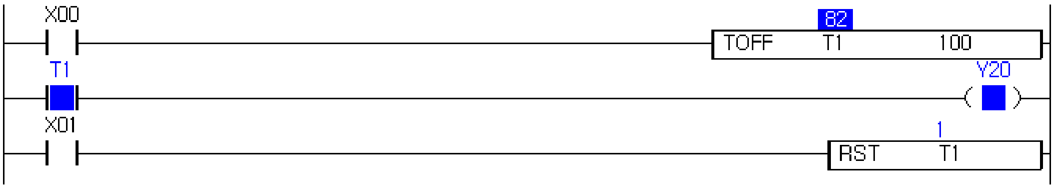
Instruction	Device	
LD	X00	
TOFF	T1	100
LD	T1	
OUT	Y20	
LD	X01	
RST	T1	

The program operates as following:

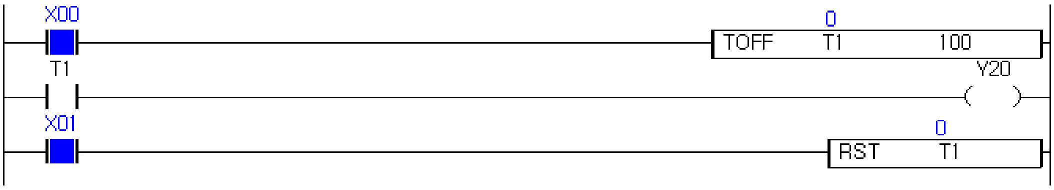
- When X00 is ON, the program sets timer T1. T1 is turned ON then Y20 is turned ON. The value of TC1 becomes “100”. (=10 seconds)



- From the moment X00 is turned OFF, the value of TC1 decreases. When it reaches “0”, T1 and Y20 is turned OFF.



- If X00 is turned ON again before the value of TC1 becomes “0”, the value of TC1 becomes “100”.
- When X01 is ON, RST instruction operates and the value of TC1 becomes “0”.



2.15.3 TMR

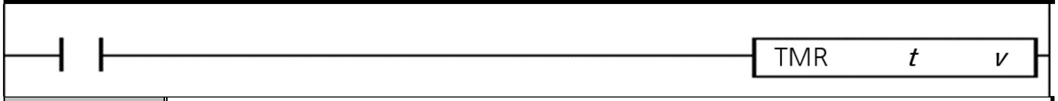
Supported PLC Series

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

TMR turns ON a timer contact when integrated current value reaches the setting value.

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
TMR	*t*	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	O	O	-	O				



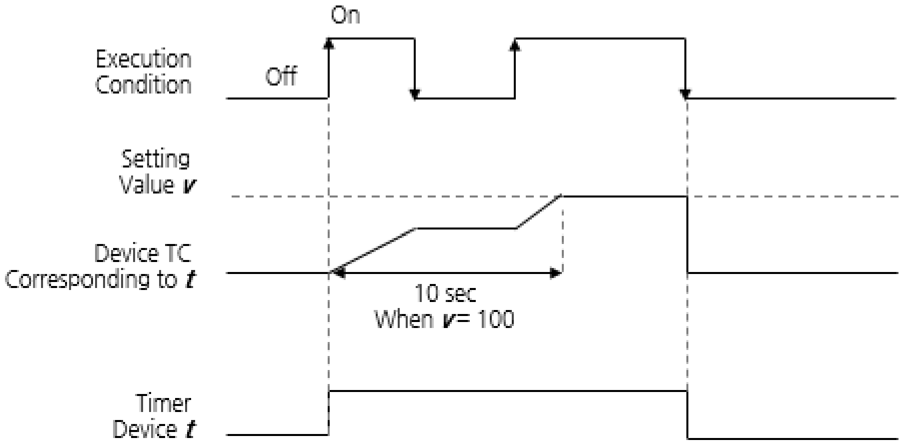
t	Address of timer device T. Device T is a bit device.
v	Setting value of timer.

TMR

- Device T assigned to *t* is a bit device. There are device TC and TS to express the detailed value of device T assigned to *t*.
- Device TC is a word device to express the current time value of *t*.
- Device TS is a word device to express the time setting value of *t*. The value of device TS is equal to the value assigned to *v*.
- When execution condition is ON, the value of device TC which corresponds to the timer device *t* increases.
- Unlike TON instruction, the value of device TC does not change even the execution condition turns OFF. However, to maintain the integrated value in case of power failure, you should use device K which is a latch device by default.
- When the execution condition turns ON again, integrated value of device TC increases.
- When the value of device TC which corresponds to the timer device *t* reaches the setting value *v*, the device assigned by *t* turns ON.
- The actual time for timer device *t* is calculated as following:
  Actual time = *v* * time unit (100ms or 10ms)[8]
- From 0 to 65535 can be assigned to *v*.
- When the timer contact meets RST instruction, the value of device TC becomes “0” and the timer turns OFF.
- The timer contact assigned to *t* can be used as an A or B contact.

[8] You can configure time unit at [Tool] - [PLC Parameter] - [Basic] - [Timer].

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

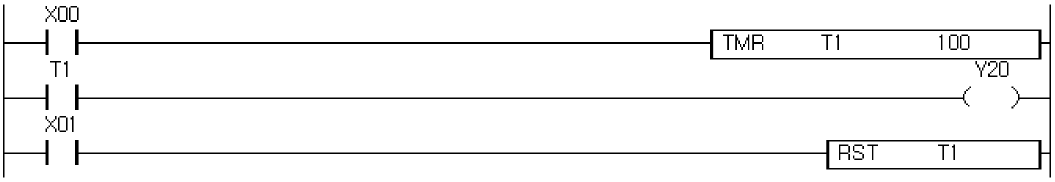
Program Example

TMR

Status of X00 repeats ON and OFF. This program integrates the increased value of TC1 while X00 is ON. When the value of TC1 reaches “100”, T1 turns ON. When T1 is ON, the output Y20 turns ON.

If X01 turns ON, RST instruction executes. Therefore, the value of TC1 becomes “0” and T1 is turned OFF.

Ladder Diagram (LD)

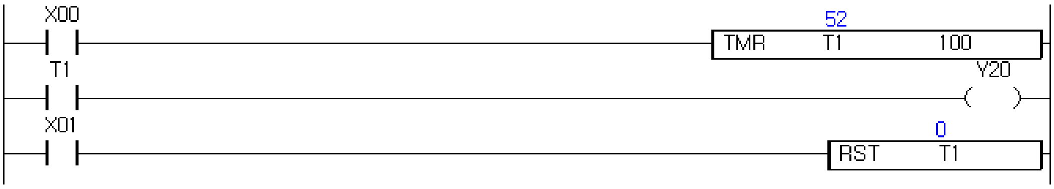


Instruction List (IL)

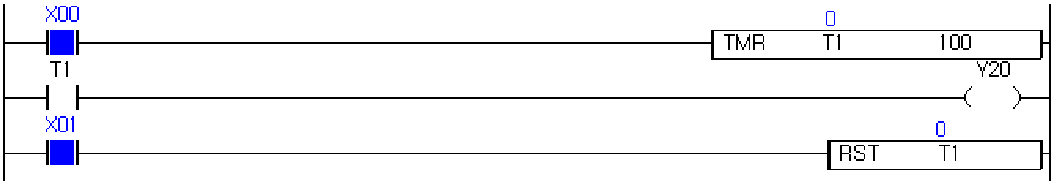
Instruction	Device	
LD	X00	
TMR	T1	100
LD	T1	
OUT	Y20	
LD	X01	
RST	T1	

The program operates as following:

- When X00 is ON, the program starts timer device T1. The value of TC1 increases.
- When X00 is OFF, increment of the current value stops. It does not reset.
- When X00 is ON again and the value of TC1 reaches 100, T1 is turned ON. When T1 is ON, it turns ON Y20.



- When X01 is ON, RST instruction operates and the value of TC1 becomes “0”.



2.15.4 TMON

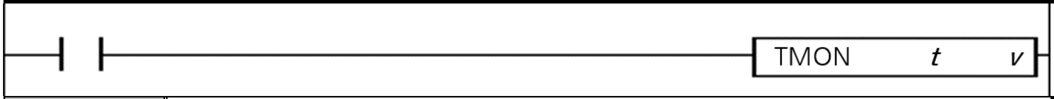
Supported PLC Series

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

TMON turns ON the timer contact when execution condition is ON and current value decreases from setting value. TMON turns OFF a timer contact when the current value becomes “0”.

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
TMON	*t*	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	O	O	-	O				



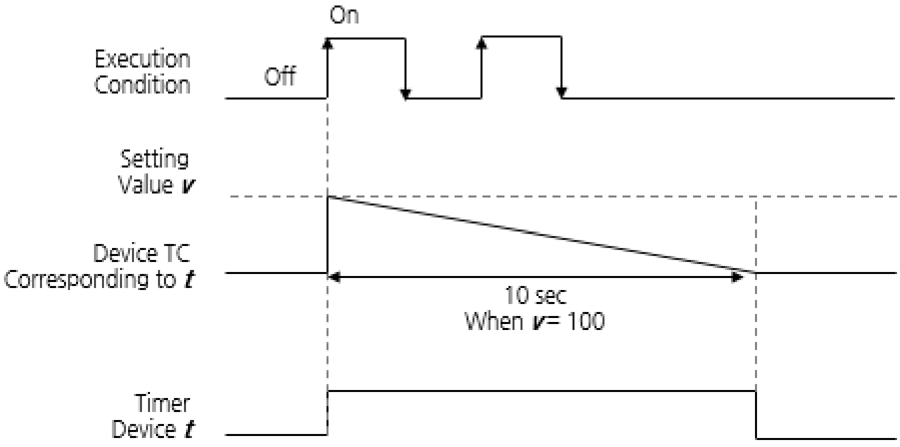
t	Address of timer device T. Device T is a bit device.
v	Setting value of timer.

TMON

- Device T assigned to *t* is a bit device. There are device TC and TS to express the detailed value of device T assigned to *t*.
- Device TC is a word device to express the current time value of *t*.
- Device TS is a word device to express the time setting value of *t*. The value of device TS is equal to the value assigned to *v*.
- When the execution condition is ON, the current value is set as the setting value assigned to *v*. Also, the timer device assigned by *t* is turned ON.
- When the execution condition is ON, the current value of device starts decreasing from the setting value assigned to *v*.
- Unlike TOFF instruction, the value of device TC continues decreasing irrespective of the status of execution condition.
- The actual time for timer device *t* is calculated as following:
  Actual time = *v* * time unit (100ms or 10ms)[9]
- From 0 to 65535 can be assigned to *v*.
- When the value of device TC becomes “0”, the device assigned by *t* turns OFF.
- When the timer contact meets RST instruction, the current value becomes “0” and the timer turns OFF.
- The timer contact assigned to *t* can be used as an A or B contact.

[9] You can configure time unit at [Tool] - [PLC Parameter] - [Basic] - [Timer].

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

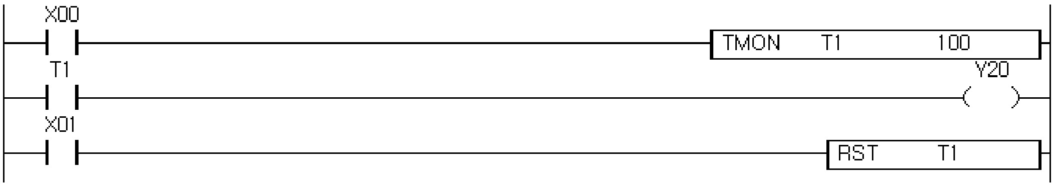
Program Example

TMON

When X00 turns ON, the value of TC1 becomes the setting value assigned to *v* and T1 turns ON. When T1 turns ON, Y20 turns ON.
The value of TC1 starts decreasing from the moment X00 is turned ON. Decrement does not stop even X00 is turned OFF.
When the value of TC1 becomes "0", T1 is turned OFF. When T1 turns OFF, Y20 turns OFF.

If X01 turns ON, RST instruction executes. Therefore, the value of TC1 becomes "0" and T1 is turned OFF.

Ladder Diagram (LD)

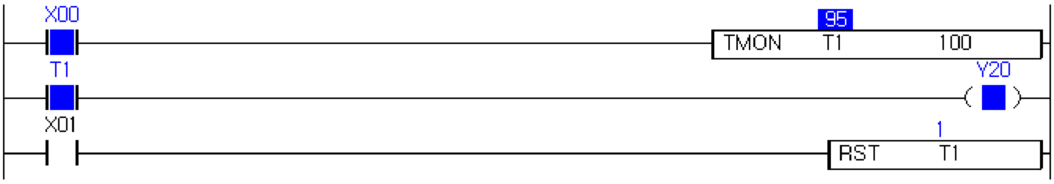


Instruction List (IL)

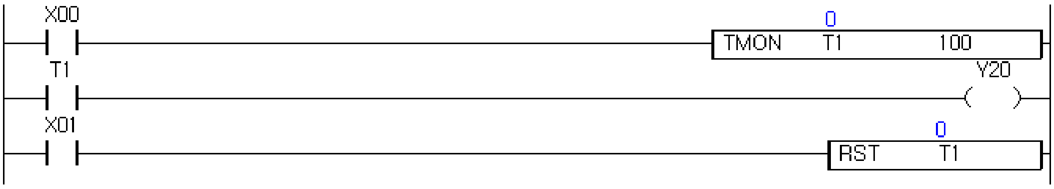
Instruction	Device	
LD	X00	
TMON	T1	100
LD	T1	
OUT	Y20	
LD	X01	
RST	T1	

The program operates as following:

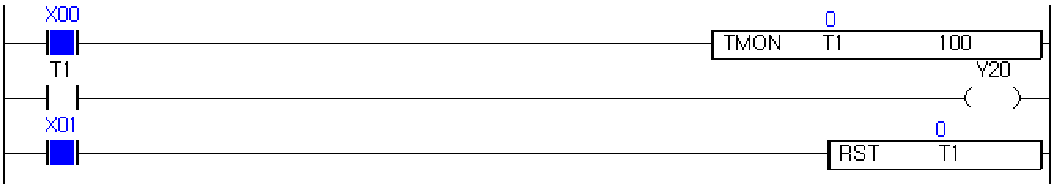
- When X00 is turned ON, the program starts timer device T1. The value of TC1 decreases from the setting value.
- Even X00 is turned OFF, decrement of the current value does not stop.



- When the value of TC1 becomes "0", T1 is turned OFF. Then Y20 is turned OFF.



- When X01 is ON, RST instruction operates and the value of TC1 becomes "0".



2.15.5 TRTG

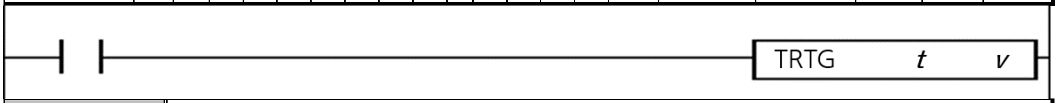
Supported PLC Series

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

TRTG turns ON the timer contact when execution condition is ON and current value decreases from setting value. When execution condition is turned OFF and turned ON again, operations starts over from the beginning. TRTG turns OFF a timer contact when the current value becomes "0".

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
TRTG	*t*	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	O	O	-	O				



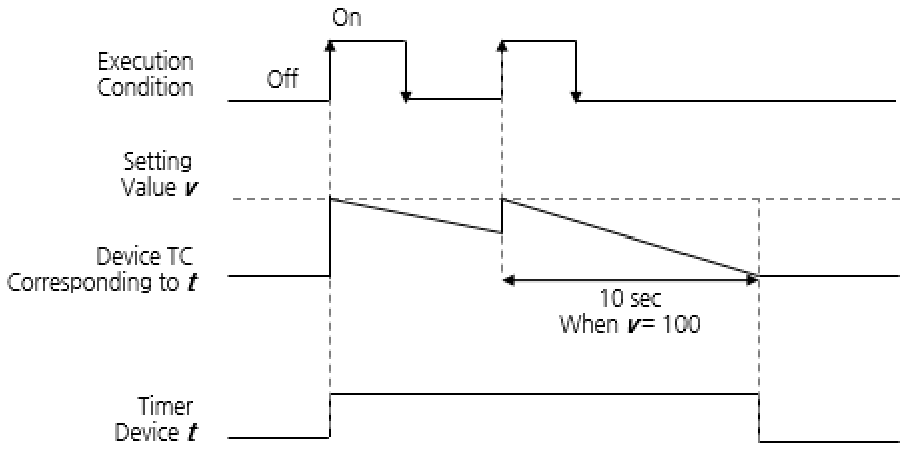
t	Address of timer device T. Device T is a bit device.
v	Setting value of timer.

TRTG

- Device T assigned to *t* is a bit device. There are device TC and TS to express the detailed value of device T assigned to *t*.
- Device TC is a word device to express the current time value of *t*.
- Device TS is a word device to express the time setting value of *t*. The value of device TS is equal to the value assigned to *v*.
- When the execution condition is ON, the value of device TC which corresponds to the timer device *t* becomes the setting value assigned to *v*. Also, the timer device assigned by *t* is turned ON.
- The actual time for timer device *t* is calculated as following:
  Actual time = *v* * time unit (100ms or 10ms)[10]
- From 0 to 65535 can be assigned to *v*.
- Unlike TMON instruction, if the execution condition is turned OFF and ON again, decrement starts over from the beginning.
- When the value of device TC becomes "0", the device assigned by *t* turns OFF.
- When the timer contact meets RST instruction, the current value becomes "0" and the timer turns OFF.
- The timer contact assigned to *t* can be used as an A or B contact.

[10] You can configure time unit at [Tool] - [PLC Parameter] - [Basic] - [Timer].

Execution Condition



Operation Error

Error Flag (F110)

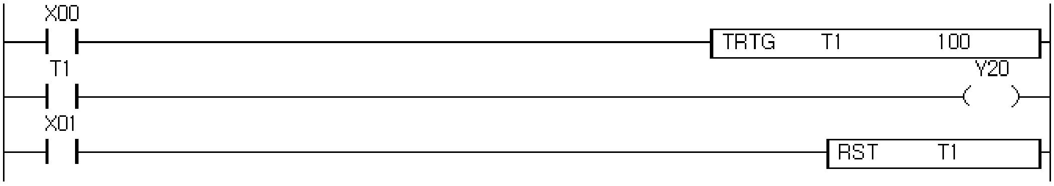
F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

Program Example

TRTG

When X00 turns ON, the value of TC1 becomes the setting value assigned to *v* and T1 turns ON. When T1 turns ON, Y20 turns ON.
The value of TC1 starts decreasing from the moment X00 is turned ON. If X00 is turned OFF and ON again, the decrement starts over from the beginning.
When the value of TC1 becomes "0", T1 is turned OFF. When T1 turns OFF, Y20 turns OFF.
If X01 turns ON, RST instruction executes. Therefore, the value of TC1 becomes "0" and T1 is turned OFF.

Ladder Diagram (LD)

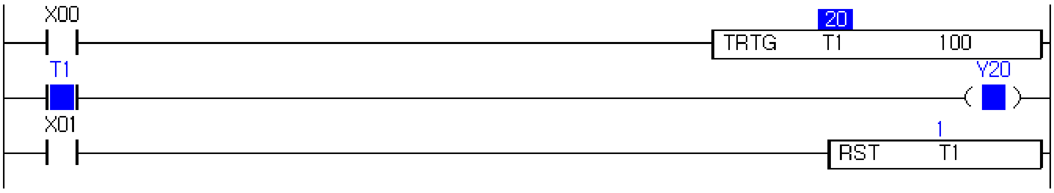


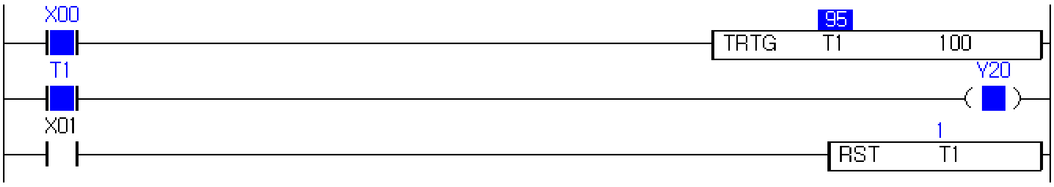
Instruction List (IL)

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

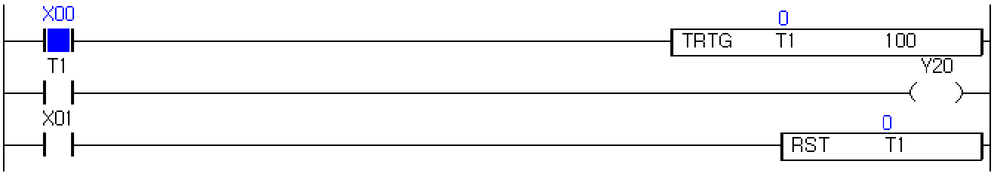
The program operates as following:

- When X00 is turned ON, the program sets timer device T1. The value of TC1 decreases from the setting value.
- If X00 is turned OFF and ON again, decrement of the value of TC1 starts over from the beginning.

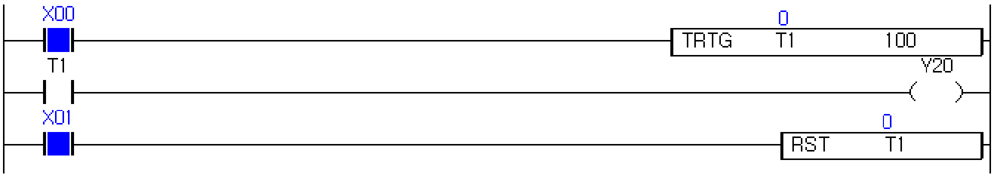




- When the value of TC1 becomes “0”, T1 and Y20 are turned OFF.



- If X01 turns ON, RST instruction operates and the value of TC1 becomes “0”.



2.16 Counter Instruction

2.16.1 CTU

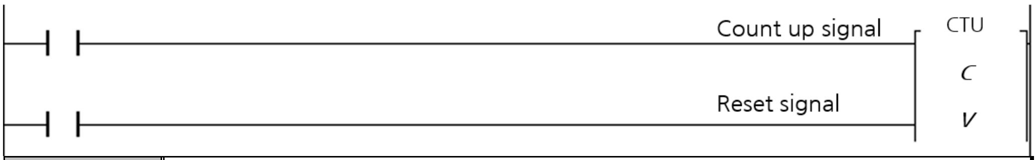
Supported PLC Series

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

CTU counts the number of ON status of execution condition and turns ON a counter contact when the current value reaches a setting value.

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
CTU	*C*	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	3	O	-	-
	V	-	-	-	-	-	-	-	-	-	O	-	-	O	-	O				



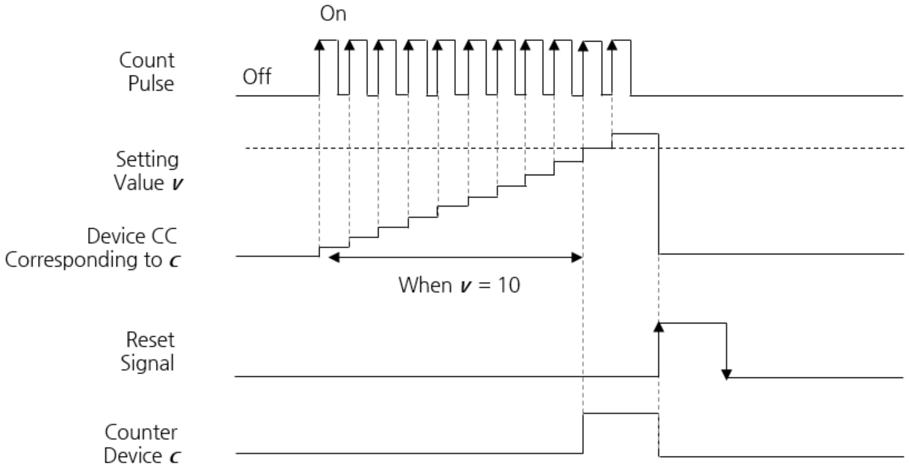
C	Address of counter device C. Device C is a bit device.
V	Setting value of counter.

CTU

- Device C assigned to *C* is a bit device. There are device CC and CS to express the detailed value of device C assigned to *C*.
- Device CC is a word device to express the current count value of *C*.
- Device CS is a word device to express the count setting value of *C*. The value of device CS is equal to the value assigned to *V*.
- Every time the execution condition turns ON, the value of device CC assigned by *C* is increased by 1.
- When value of device CC reaches the setting value assigned to *V*, the counter contact is turned ON.
- From 0 to 65535 can be assigned to *V*.
- Even the value of device CC reaches setting value, CTU can count up until 65535.
- When the value of device CC reaches 65535, the value does not increase anymore.
- When counter contact meets reset signal[11], the value of device CC becomes “0” and the counter turns OFF.
- When counter contacts meet RST instruction, the contact is turned OFF but the value of device CC is maintained.
- The counter contact assigned to *C* can be used as an A or B contact.

[11] The reset signal is different to RST instruction.

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

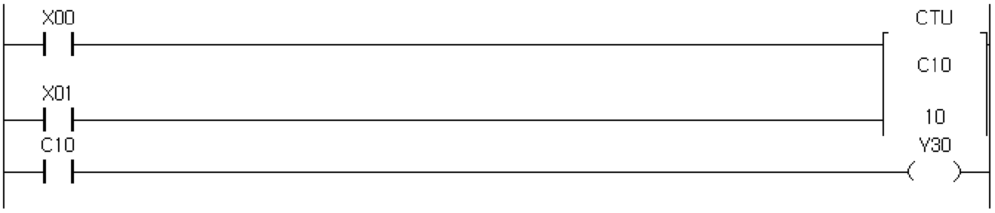
F110 turns ON for 1 scan when the value of device CC exceeds the range of 0 - 65535.

Program Example

CTU

Count up with X00. When X00 turns ON and OFF 10 times and value of CC10 reaches the setting value “10”, output Y30 is turned ON. CC10 is a current count value of C10.
When X01 is turned ON, the value of CC10 becomes “0” and turns OFF. Then, Y30 turns OFF.

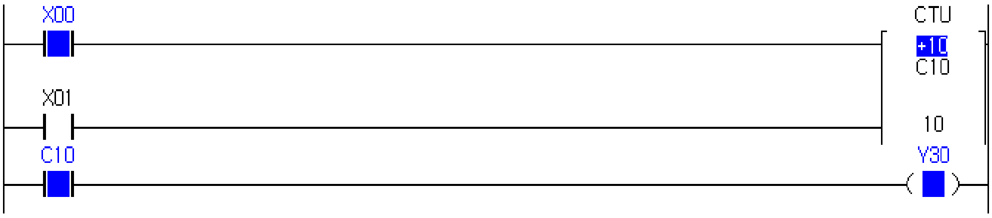
Ladder Diagram (LD)



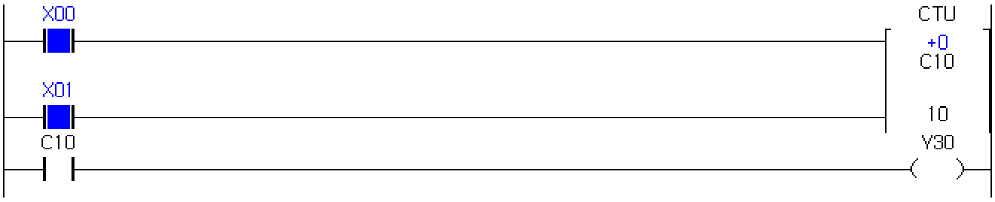
Instruction List (IL)

Instruction	Device	
LD	X00	
LD	X01	
CTU	C10	10
LD	C10	
OUT	Y30	

- The counter counts up with X00. When the value of CC10 reaches “10”, C10 is turned ON. Then Y30 turns ON.



- When X01 turns ON, CC10 is reset. Then C10 and Y30 turns OFF.



2.16.2 CTD

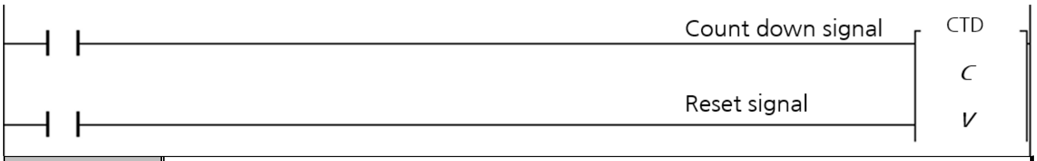
Supported PLC Series

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

CTD counts down the number of ON status of execution condition and turns ON a counter contact when the current value becomes “0”.

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
CTD	*c*	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	-	O	-	O				



c	Address of counter device C. Device C is a bit device.
v	Setting value of counter.

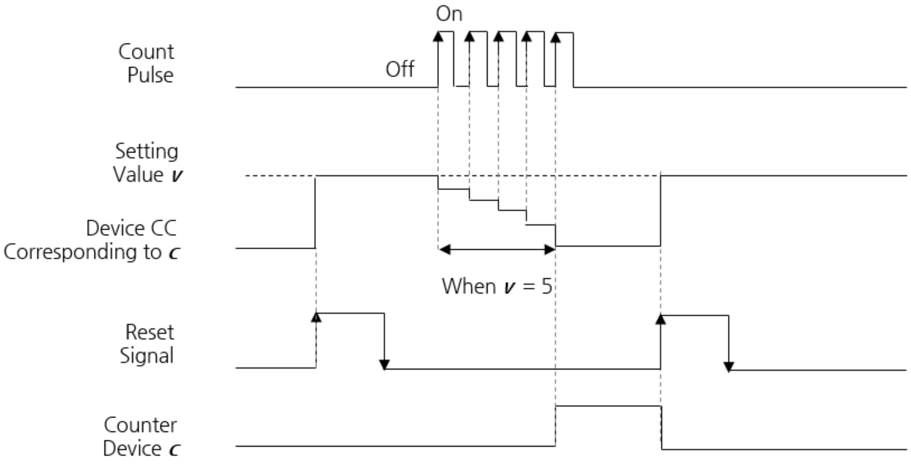
CTD

- Device C assigned to *c* is a bit device. There are device CC and CS to express the detailed value of device C assigned to *c*.
- Device CC is a word device to express the current count value of *c*.
- Device CS is a word device to express the count setting value of *c*. The value of device CS is equal to the value assigned to *v*.
- The current value of device assigned to *c* is set by the setting value assigned to *v*.
- Every time the execution condition turns ON, the value of device CC is decreased by 1.
- From 0 to 65535 can be assigned to *v*.
- When the value of device CC becomes “0”, the counter contact is turned ON.
- CTD can count down until “0”.
- When the value of device CC reaches “0”, the value does not decrease anymore.
- When counter contact meets reset signal[12], the counter contact is turned OFF and the value of device CC becomes the setting value.
- When counter contacts meet RST instruction, the contact is turned OFF but the value of device CC is maintained.
- The counter contact assigned to *c* can be used as an A or B contact.

⚠ At the first operation, the setting value is not assigned to the counter. You should turn ON the reset signal to set setting value as the value of device CC.

[12] The reset signal is different to RST instruction.

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

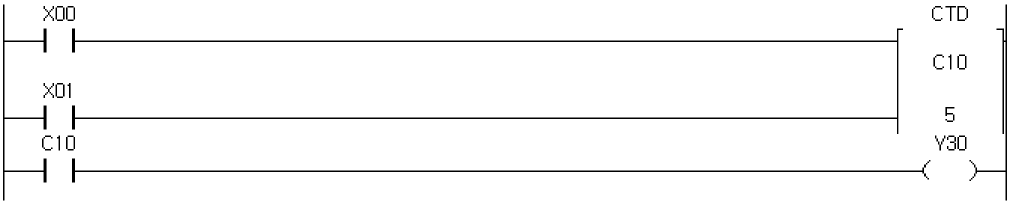
F110 turns ON for 1 scan when the value of device CC exceeds the range of 0 - 65535.

Program Example

CTD

Count down with X00. When X00 turns ON and OFF 5 times and value of CC10 decreases from the setting value "5". When value of CC10 becomes "0", the output Y30 turns ON. When X01 is turned ON, the value of CC10 is reset and turned OFF. Then, Y30 turns OFF.

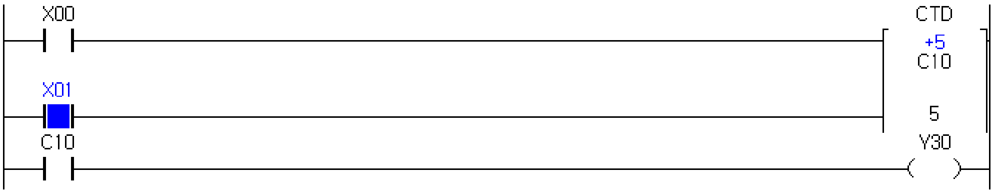
Ladder Diagram (LD)



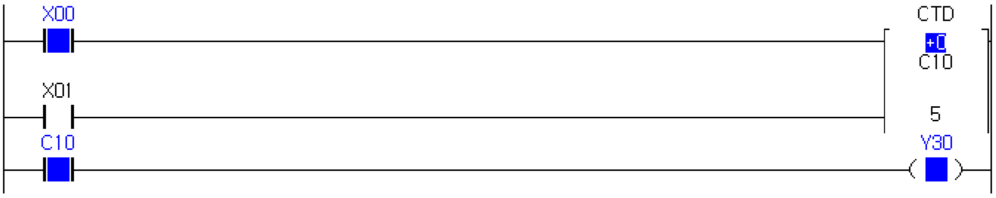
Instruction List (IL)

Instruction	Device	
LD	X00	
LD	X01	
CTU	C10	5
LD	C10	
OUT	Y30	

- Before executing CTD instruction, set setting value as value of CC10 by turning ON X01.



- Turn X00 ON and OFF. When the value of CC10 is decreased and it reaches "0", the counter output C10 turns ON.



- When X01 is turned ON, it starts over from the beginning.

2.16.3 CTUD

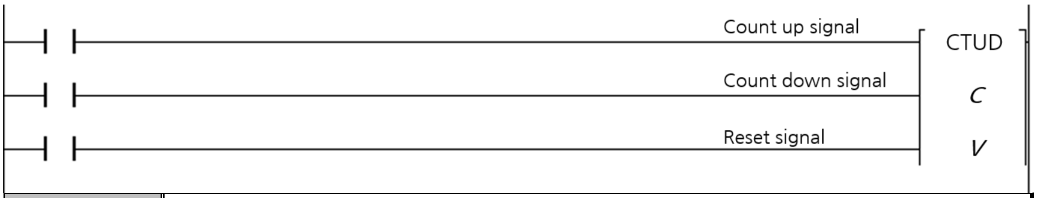
Supported PLC Series

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

CTUD counts up and down the number of ON status of execution condition and turns ON a counter contact when the current value reaches a setting value.

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
CTUD	*c*	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	-	O	-	O				



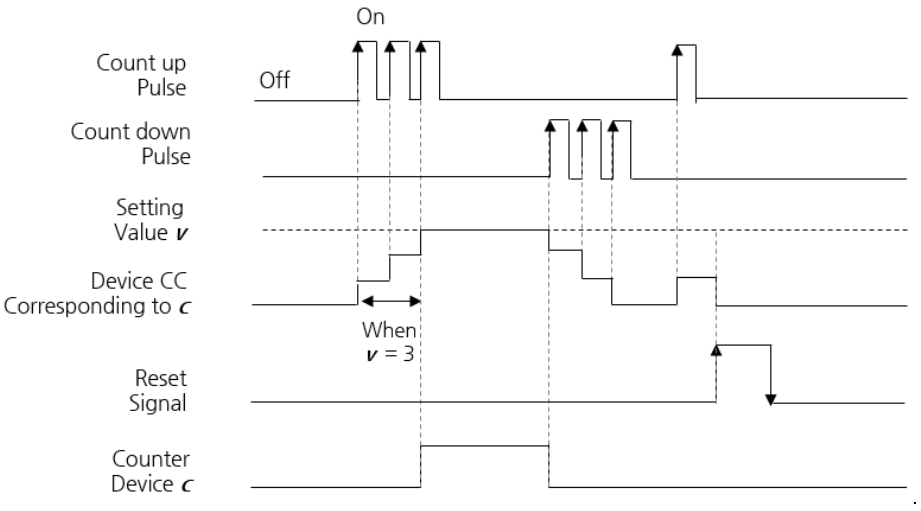
c	Address of counter device C. Device C is a bit device.
v	Setting value of counter.

CTUD

- Device C assigned to *c* is a bit device. There are device CC and CS to express the detailed value of device C assigned to *c*.
- Device CC is a word device to express the current count value of *c*.
- Device CS is a word device to express the count setting value of *c*. The value of device CS is equal to the value assigned to *v*.
- Every time the execution condition of count up signal turns ON, the current value of device assigned by *c* is increased by 1.
- Every time the execution condition of count down signal turns ON, the current value of device assigned by *c* is decreased by 1.
- When value of device CC reaches the setting value assigned to *v*, the counter contact is turned ON.
- CTUD can count up until 65535 and count down until 0.
- From 0 to 65535 can be assigned to *v*.
- When counter contact meets reset signal[13], the counter contact is turned OFF and the value of device CC becomes “0”.
- When counter contacts meet RST instruction, the contact is turned OFF but the value of device CC is maintained.
- If up and down pulse is turned ON simultaneously, the current value does not change.
- The counter contact assigned to *c* can be used as an A or B contact.

[13] The reset signal is different to RST instruction.

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (Device Z is used in Function Block implementation. The range of device Z depends on CPU type.)

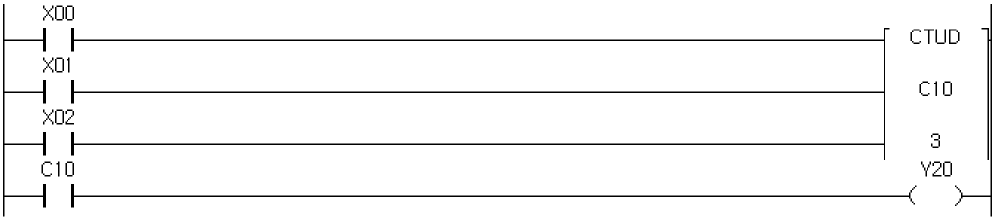
F110 turns ON for 1 scan when the value of device CC exceeds the range of 0 - 65535.

Program Example

CTUD

The program counts up with rising pulse input X00. The program counts down with rising pulse input X01. When the value of CC10 reaches or exceeds the setting value, output Y20 is turned ON.
When X02 is turned ON, the value of CC10 is reset. C10 is turned OFF. Then, Y20 turns OFF.

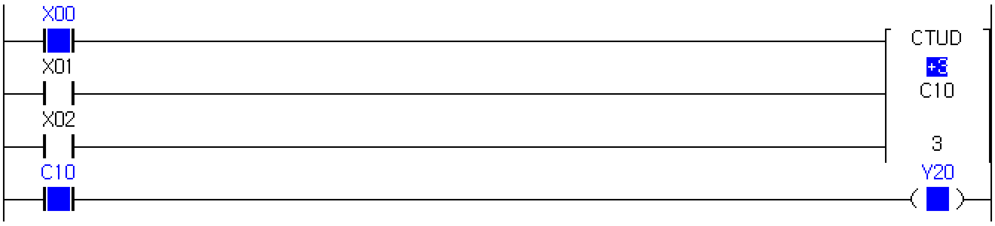
Ladder Diagram (LD)



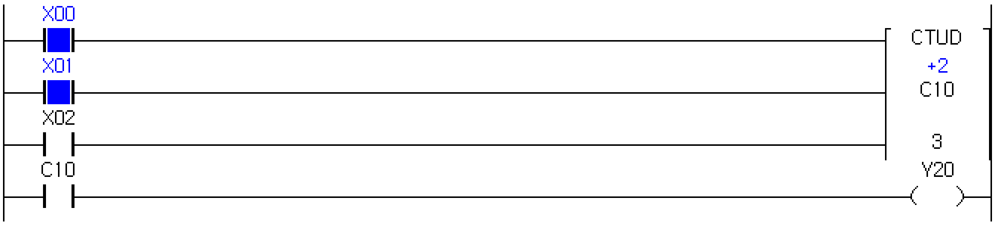
Instruction List (IL)

Instruction	Device	
LD	X00	
LD	X01	
LD	X02	
CTUD	C10	3
LD	C10	
OUT	Y20	

- When X00 turns ON and OFF repeatedly, the value of CC10 increases with every rising pulse input. When current value reaches or exceeds "3", the counter output C10 is turned ON. Then output Y20 is turned ON.



- When X01 is turned ON and the value of CC10 is less than setting value, the counter output C10 is turned OFF. Then output Y20 is turned OFF.



- When X02 is turned ON, C10 is reset and Y20 turns OFF.

2.16.4 CTR

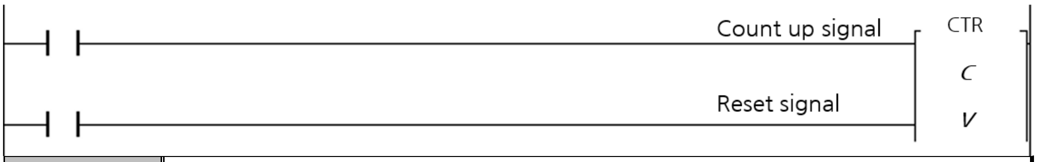
Supported PLC Series

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

CTR counts the number of ON status of execution condition and turns ON a counter contact when the current value reaches a setting value.

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
CTR	*c*	-	-	-	-	-	-	-	O	-	-	-	-	-	-	-	3	O	-	-
	v	-	-	-	-	-	-	-	-	-	O	-	-	O	-	O				



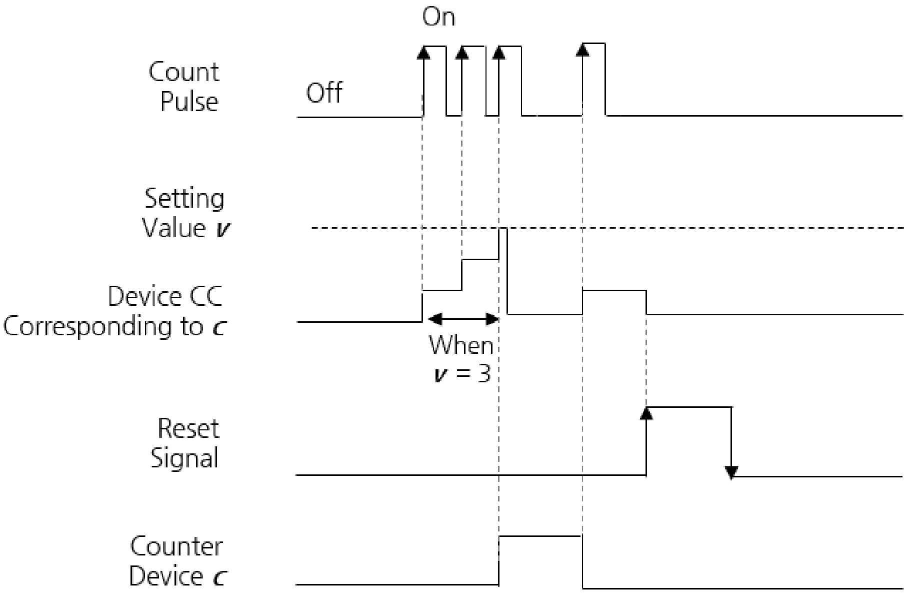
c	Address of counter device C. Device C is a bit device.
v	Setting value of counter.

CTR

- Device C assigned to *c* is a bit device. There are device CC and CS to express the detailed value of device C assigned to *c*.
- Device CC is a word device to express the current count value of *c*.
- Device CS is a word device to express the count setting value of *c*. The value of device CS is equal to the value assigned to *v*.
- Every time the execution condition turns ON, the value of device CC is increased by 1.
- When value of device CC reaches setting value assigned to *v*, the counter contact is turned ON.
- When value of device CC reaches the setting value, the value becomes “0” and the count starts over from the beginning.
- From 0 to 65535 can be assigned to *v*.
- When counter contact meets reset signal[14], the counter contact is turned OFF and the value of device CC becomes “0”.
- When counter contact meets RST instruction, the counter contact does not turn ON even the current value reaches setting value.
- The counter contact assigned to *c* can be used as an A or B contact.

[14] The reset signal is different to RST instruction.

Execution Condition



Operation Error

Error Flag (F110)

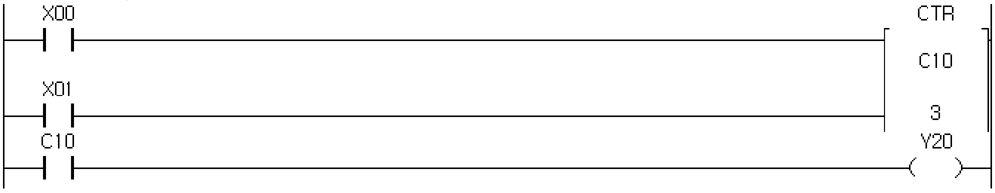
F110 turns ON for 1 scan when the address of device assigned by Z exceeds the range of device. (When device Z is used for Function Block implementation, the valid range of device is Z128 - Z1023.)

Program Example

CTR

The program counts up with rising pulse input X00. When the value of CC10 reaches the setting value, C10 turns ON then output Y20 is turned ON. When the value of CC10 exceeds setting value, the program starts over from the beginning.
When X01 is turned ON, the value of CC10 is reset. C10 is turned OFF. Then, Y20 turns OFF.

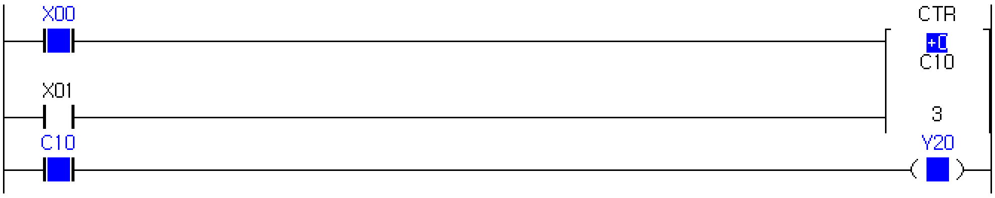
Ladder Diagram (LD)



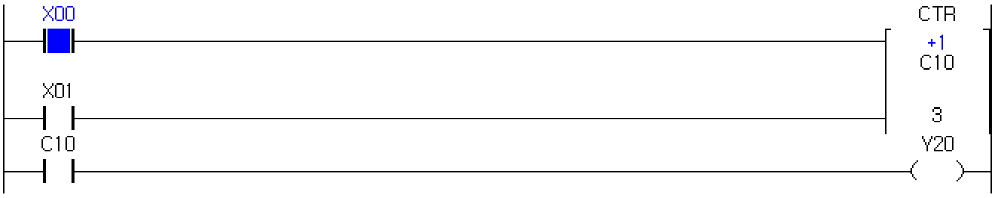
Instruction List (IL)

Instruction	Device	
LD	X00	
LD	X01	
CTR	C10	3
LD	C10	
OUT	Y20	

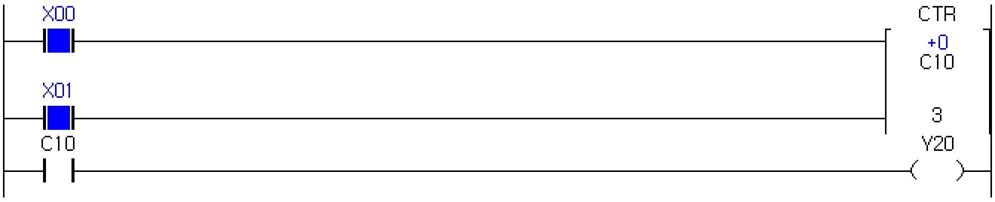
- When X00 turns ON and OFF repeatedly, the program counts up with rising pulse input. When the value of CC10 reaches “3”, C10 is turned ON and value of CC10 becomes “0”. Then output Y20 is turned ON.



- If the value of CC10 exceeds “3”, the output C10 and Y20 are turned OFF.



- When X01 is turned ON, the value of CC10 is reset. C10 is turned OFF. Then, Y20 is turned OFF.



2.17 Special Function Instructions

2.17.1 SIN, SINP

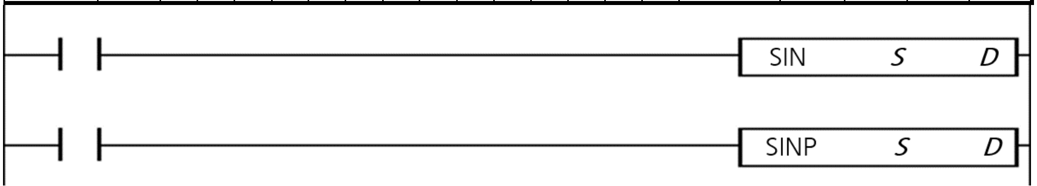
Supported PLC Series

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

Function

SIN(P) instructions calculate the sine of the angle data in radian units (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

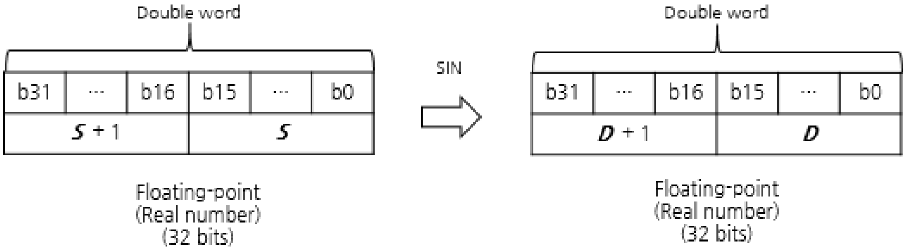
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SIN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Angle data in radian units or head address of double word device where angle data in radian units to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation (Sine) result will be stored.

SIN, SINP

- SIN(P) instructions calculate the sine of the angle data in radian units (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.



- The value stored in double word device *S* has to be in radian unit.

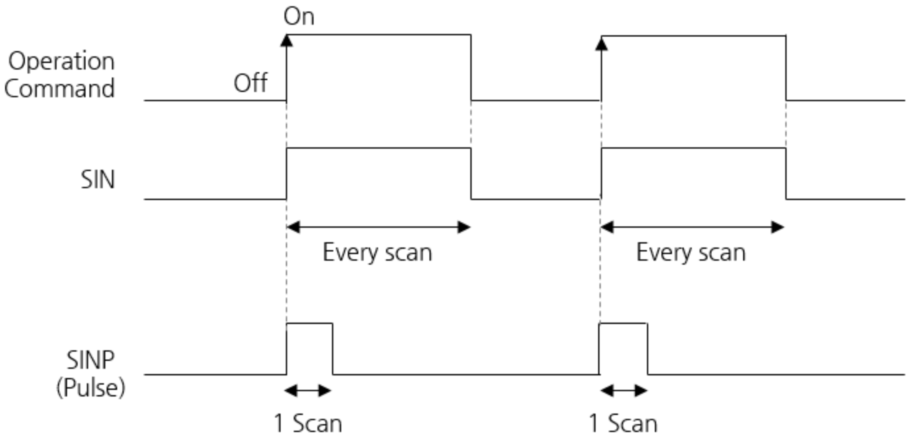
Radian = (Degree × $\frac{\pi}{180}$)

- Value which can be assigned to device *S* is as following.
  0, $2^{-127} \leq$ | Value assigned to device *S* | $\leq 2^{128}$
- When the value assigned to *S* exceeds the range and the instruction is executed, the instruction returns '-1.#QNAN[15]0' in device *D*.

(*) For information on conversion between degrees and radians, refer to RAD or DEG instruction.

[15] QNAN: Stands for 'Quiet Not-a-Number'.

Execution Condition



Operation Error

Error flag (F110)

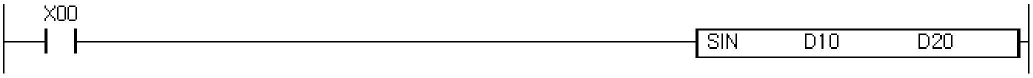
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

SIN

When X00 turns ON, SIN instruction calculates sine of the angle data in radian units stored in D10. Then, the result is stored in D20.

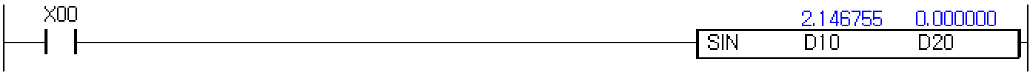
Ladder Diagram (LD)



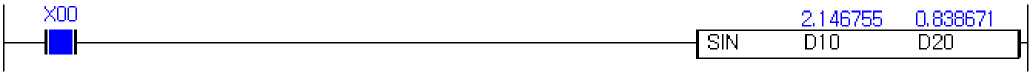
Instruction List (IL)

Instruction	Device	
LD	X00	
SIN	D10	D20

- Angle data in radian unit “2.146755” is assigned to double word device D10 (D10~D11).



- When X00 turns ON, SIN instruction is executed. Then, the result is stored in double word device D20 (D20~D21).



2.17.2 COS, COSP

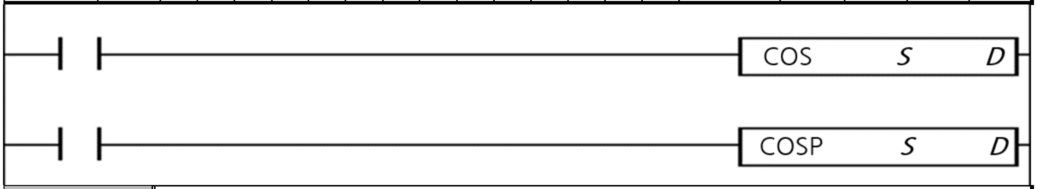
Supported PLC Series

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

Function

COS(P) instructions calculate the cosine of the angle data in radian units (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

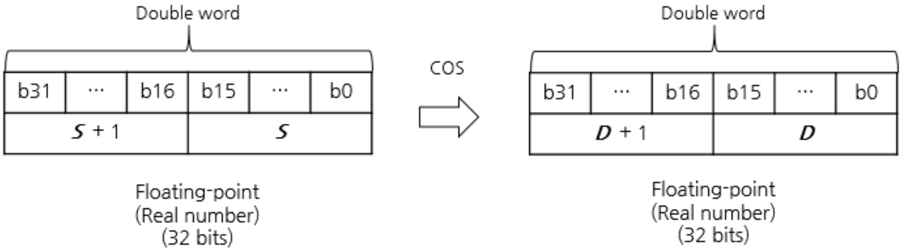
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
COS(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Angle data in radian units or head address of double word device where angle data in radian units to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation (Cosine) result will be stored.

COS, COSP

- COS(P) instructions calculate the cosine of the angle data in radian units (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.



- The value stored in double word device *S* has to be in radian unit.
  Radian = (Degree × $\frac{\pi}{180}$)

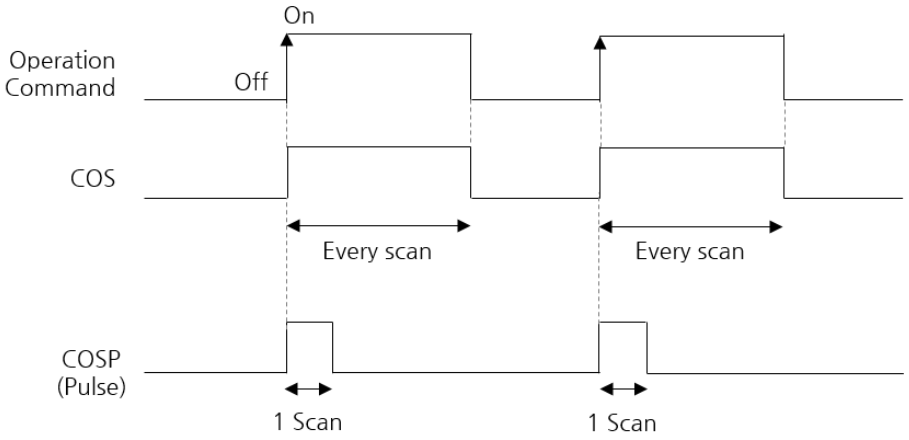
- Value which can be assigned to device *S* is as following.
  0, $2^{-127} \leq$| Value assigned to device *S* |$\leq 2^{128}$

- When the value assigned to *S* exceeds the range and the instruction is executed, the instruction returns ‘-1.#QNAN[16]0’ in device *D*.

(*) For information on conversion between degrees and radians, refer to RAD or DEG instruction.

[16] QNAN: Stands for ‘Quiet Not-a-Number’.

Execution Condition



Operation Error

Error flag (F110)

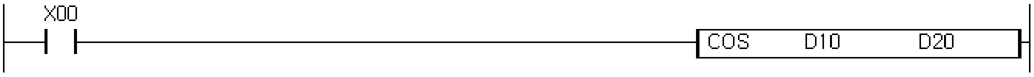
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

COS

When X00 turns ON, COS instruction calculates cosine of the angle data in radian units stored in D10. Then, the result is stored in D20.

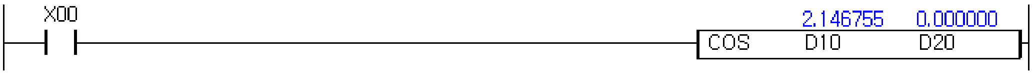
Ladder Diagram (LD)



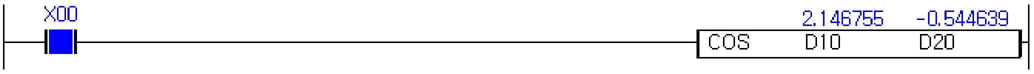
Instruction List (IL)

Instruction	Device	
LD	X00	
COS	D10	D20

- Angle data in radian unit “2.146755” is assigned to double word device D10 (D10~D11).



- When X00 turns ON, COS instruction is executed. Then, the result is stored in double word device D20 (D20~D21).



2.17.3 TAN, TANP

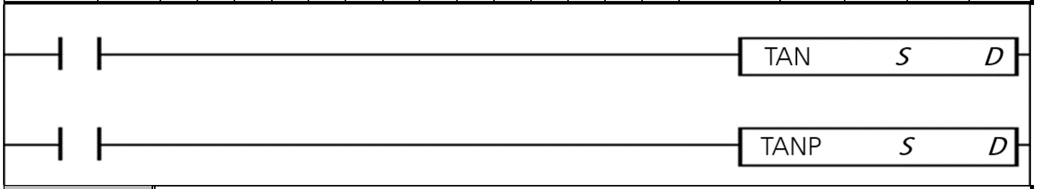
Supported PLC Series

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

Function

TAN(P) instructions calculate the tangent of the angle data in radian units (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

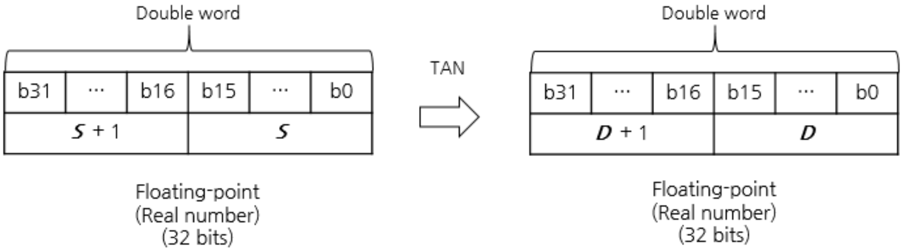
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
TAN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Angle data in radian units or head address of double word device where angle data in radian units to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation (Tangent) result will be stored.

TAN, TANP

- TAN(P) instructions calculate the tangent of the angle data in radian units (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.



- The value stored in double word device *S* has to be in radian unit.

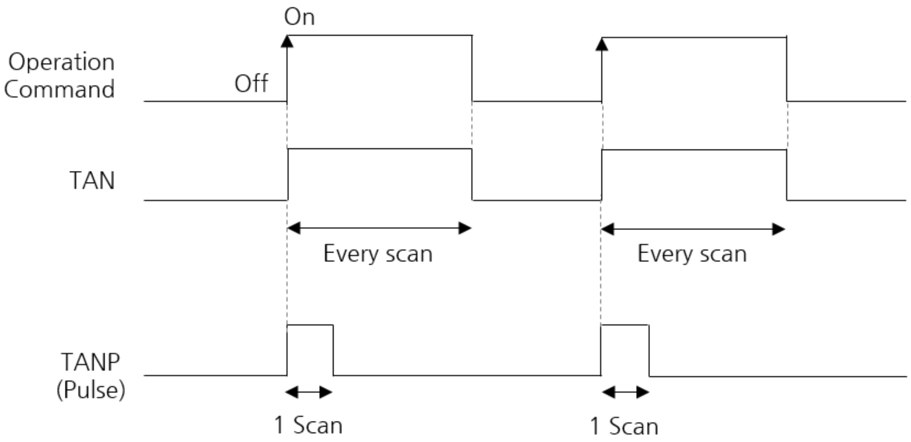
Radian = (Degree × $\frac{\pi}{180}$)

- PLC CPU rounds the radian data from the sixth decimal point. It means that even if you assign 0.5π or 1.5π to device *S*, the instruction operates without an error.
- Value which can be assigned to device *S* is as following.
  0, 2^{-127} ≤| Value assigned to device *S* |≤ 2^{128}
- When the value assigned to *S* exceeds the range and the instruction is executed, the instruction returns '-1.#QNAN[17]0' in device *D*.

(*) For information on conversion between degrees and radians, refer to RAD or DEG instruction.

[17] QNAN: Stands for 'Quiet Not-a-Number'.

Execution Condition



Operation Error

Error flag (F110)

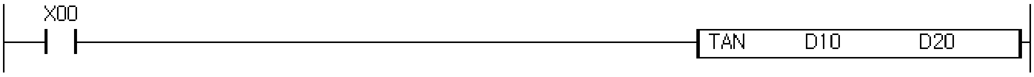
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

TAN

When X00 turns ON, TAN instruction calculates tangent of the angle data in radian units stored in D10. Then, the result is stored in D20.

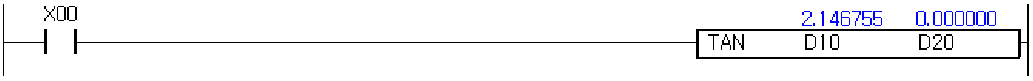
Ladder Diagram (LD)



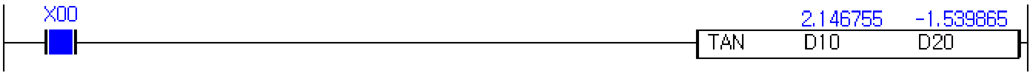
Instruction List (IL)

Instruction	Device	
LD	X00	
TAN	D10	D20

- Angle data in radian unit “2.146755” is assigned to double word device D10 (D10~D11).



- When X00 turns ON, TAN instruction is executed. Then, the result is stored in double word device D20 (D20~D21).



2.17.4 ASIN, ASINP

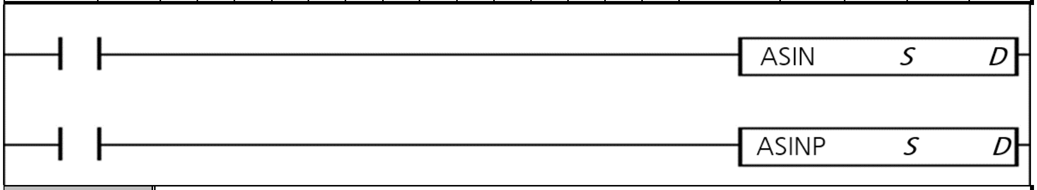
Supported PLC Series

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

Function

ASIN(P) instructions calculate the inverse sine of the SIN data (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

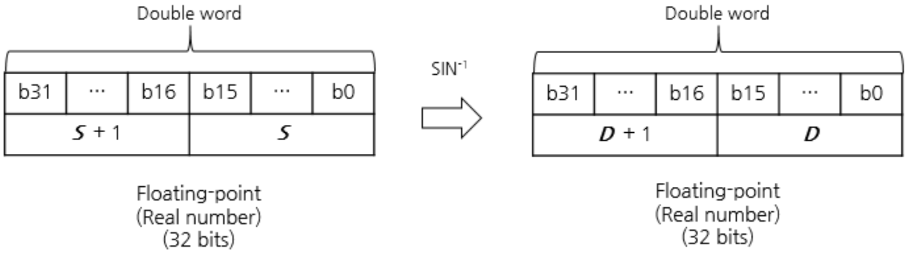
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ASIN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	SIN data or head address of double word device where SIN data to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation (Inverse sine) result will be stored.

ASIN, ASINP

- ASIN(P) instructions calculate the inverse sine of the SIN data (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

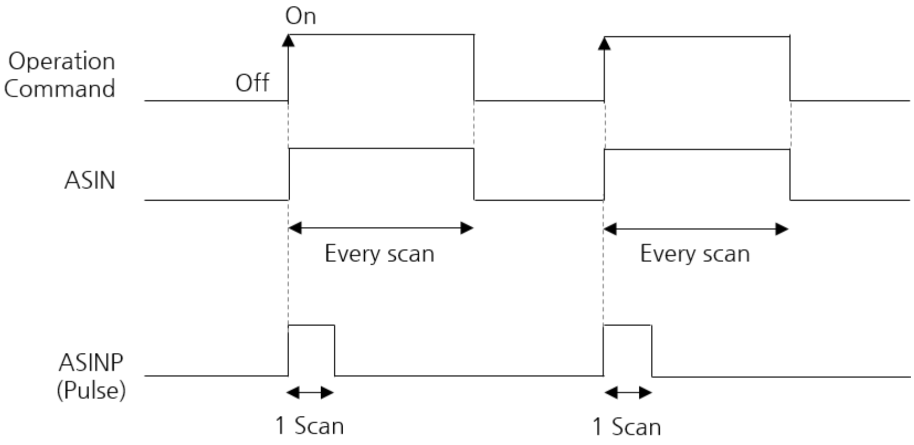


- The value stored in double word device *S* has to be SIN data.
  -1.0 ≤ Value assigned to *S* ≤ 1.0
- When the value assigned to *S* exceeds the range and the instruction is executed, the instruction returns '-1.#QNAN[18]0' in device *D*.
- The calculation result stored in double word device *D* is angle data in radian units.

(*) For information on conversion between degrees and radians, refer to RAD or DEG instruction.

[18] QNAN: Stands for 'Quiet Not-a-Number'.

Execution Condition



Operation Error

Error flag (F110)

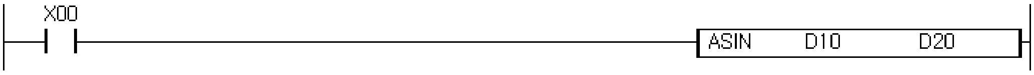
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

ASIN

When X00 turns ON, ASIN instruction calculates inverse sine of the SIN data stored in D10. Then, the result is stored in D20, in radian units.

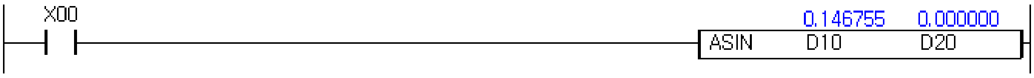
Ladder Diagram (LD)



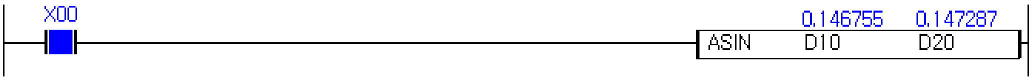
Instruction List (IL)

Instruction	Device	
LD	X00	
ASIN	D10	D20

- SIN data “0.146755” is assigned to double word device D10 (D10~D11).



- When X00 turns ON, ASIN instruction is executed. Then, the result is stored in double word device D20 (D20~D21).



2.17.5 ACOS, ACOSP

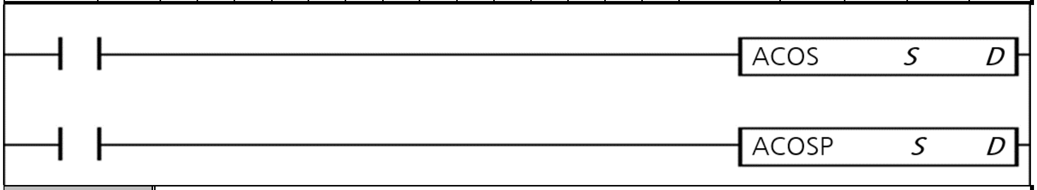
Supported PLC Series

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

Function

ACOS(P) instructions calculate the inverse cosine of the COS data (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

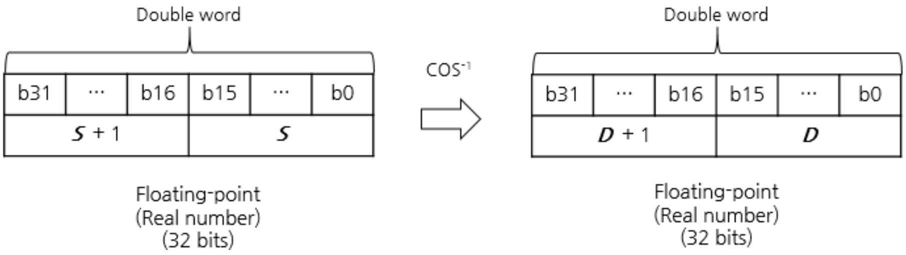
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ACOS(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	COS data or head address of double word device where COS data to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation (Inverse cosine) result will be stored.

ACOS, ACOSP

- ACOS(P) instructions calculate the inverse cosine of the COS data (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

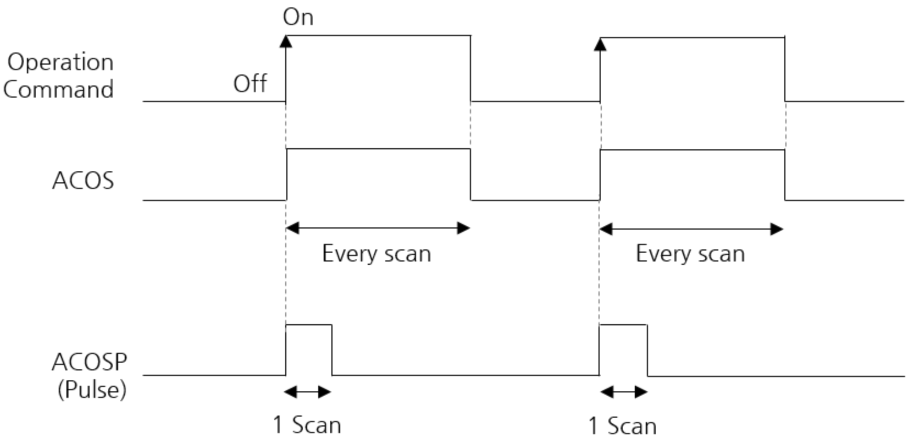


- The value stored in double word device *S* has to be COS data.
  -1.0 ≤ Value assigned to *S* ≤ 1.0
- When the value assigned to *S* exceeds the range and the instruction is executed, the instruction returns '-1.#QNAN[19]0' in device *D*.
- The calculation result stored in double word device *D* is angle data in radian units.

(*) For information on conversion between degrees and radians, refer to RAD or DEG instruction.

[19] QNAN: Stands for 'Quiet Not-a-Number'.

Execution Condition



Operation Error

Error flag (F110)

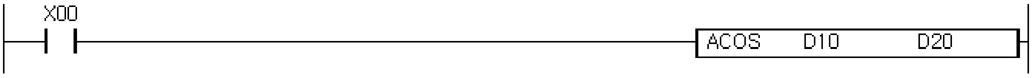
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

ACOS

When X00 turns ON, ACOS instruction calculates inverse cosine of the COS data stored in D10. Then, the result is stored in D20.

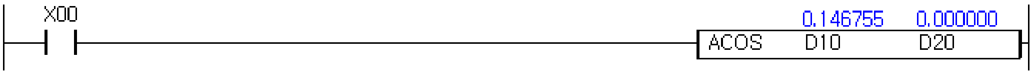
Ladder Diagram (LD)



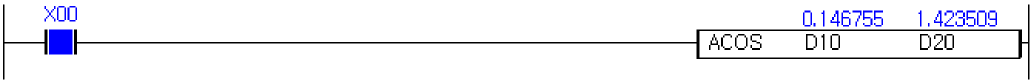
Instruction List (IL)

Instruction	Device	
LD	X00	
ACOS	D10	D20

- COS data “0.146755” is assigned to double word device D10 (D10~D11).



- When X00 turns ON, ACOS instruction is executed. Then, the result is stored in double word device D20 (D20~D21).



2.17.6 ATAN, ATANP

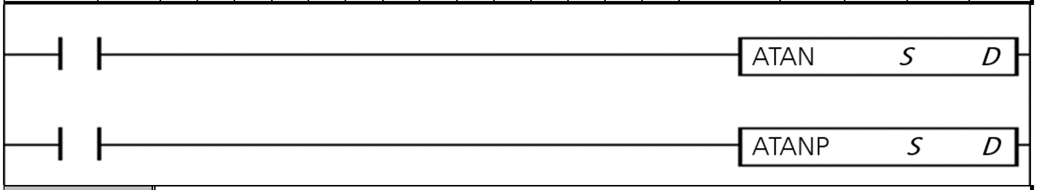
Supported PLC Series

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

Function

ATAN(P) instructions calculate the inverse tangent of the TAN data (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

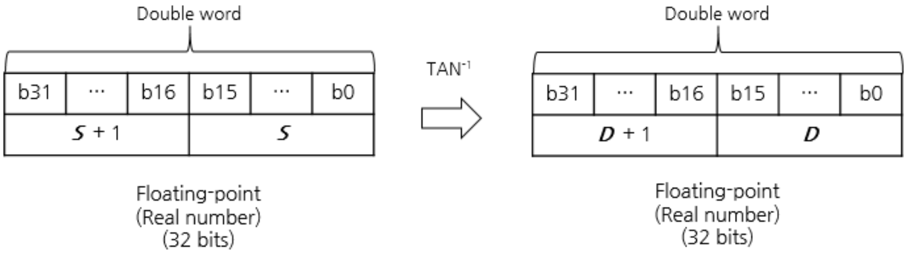
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
ATAN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	TAN data or head address of double word device where TAN data to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation (Inverse tangent) result will be stored.

ATAN, ATANP

- ATAN(P) instructions calculate the inverse tangent of the TAN data (32-bit floating-point number) stored in double word device *S*. The result is stored in the double word device *D*.

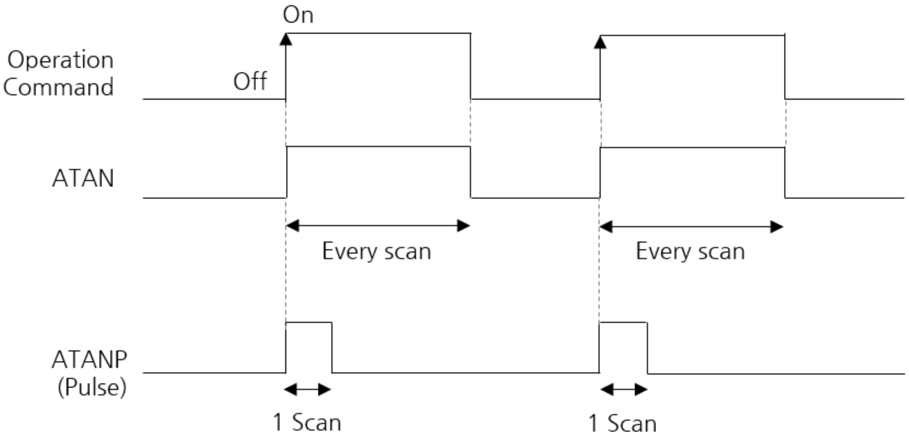


- The value stored in double word device *S* has to be TAN data.
  0, $2^{-127} \leq$ | Value assigned to *S* | $\leq 2^{128}$
- When the value assigned to *S* exceeds the range and the instruction is executed, the instruction returns '-1.#QNAN[20]0' in device *D*.
- The calculation result stored in double word device *D* is angle data in radian units.

(*) For information on conversion between degrees and radians, refer to RAD or DEG instruction.

[20] QNAN: Stands for 'Quiet Not-a-Number'.

Execution Condition



Operation Error

Error flag (F110)

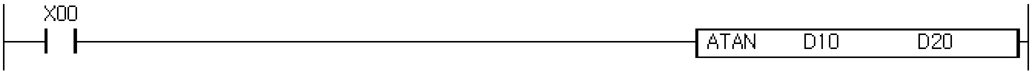
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

ATAN

When X00 turns ON, ATAN instruction calculates inverse tangent of the TAN data stored in D10. Then, the result is stored in D20.

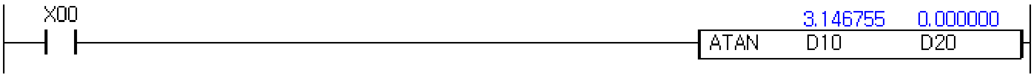
Ladder Diagram (LD)



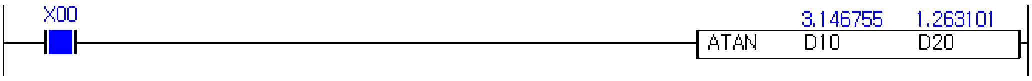
Instruction List (IL)

Instruction	Device	
LD	X00	
ATAN	D10	D20

- TAN data “3.146755” is assigned to double word device D10 (D10~D11).



- When X00 turns ON, ATAN instruction is executed. Then, the result is stored in double word device D20 (D20~D21).



2.17.7 BSIN, BSINP

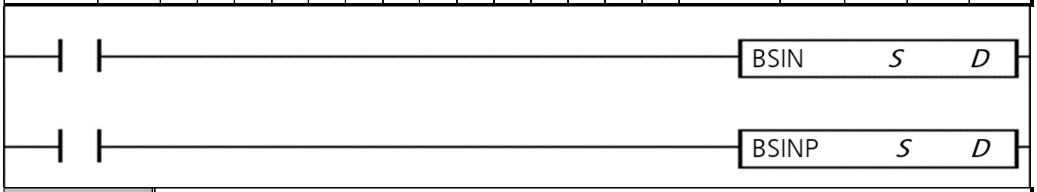
Supported PLC Series

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

Function

BSIN(P) instructions calculate the sine of the angle data in degree units (4-digit BCD) stored in word device *S*. The result is stored in the 3 word devices starting from *D*.

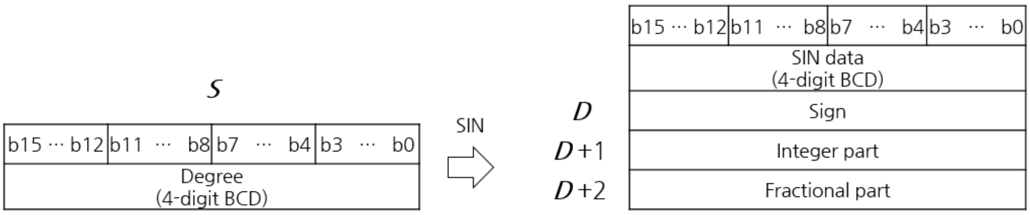
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BSIN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Angle data in degree units or address of word device where angle data in degree units to be calculated is stored. (4-digit BCD)
D	Head address of 3 word devices where the calculation (Sine) result will be stored.

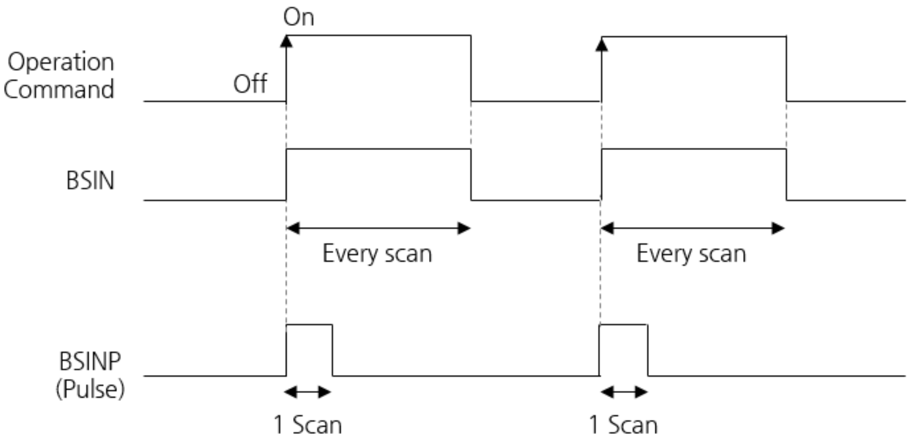
BSIN, BSINP

- BSIN(P) instructions calculate the sine of the angle data in degree units (4-digit BCD) stored in word device *S*. The result is stored in the 3 word devices starting from *D*.



- The value stored in double word device *S* has to be in degree unit.
- The range of BCD data assigned to word device *S* is as below:
  0 ≦ Value assigned to *S* ≦ 360
- If the value assigned to *S* is equal to or greater than 360, the angle data in degree unit is specified by the remainder of (value assigned to *S*)/360. (e.g. When value assigned to *S* is 362, it is equivalent to the case when the value assigned to *S* is 2.)
- When the calculation result is positive, “0” is stored in word device *D*. If the result is negative, “1” is stored in word device *D*.
- The calculation result is stored in BCD data. The range of result is as below:
  -1.0000 ≦ Calculation result ≦ 1.0000
- The calculation result is rounded off from the fifth decimal place.

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)

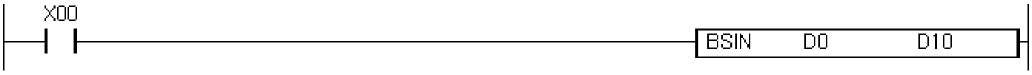
F110 turns ON for 1 scan when the value assigned to *S* is not BCD data.

Program Example

BSIN

When X00 turns ON, BSIN instruction calculates sine of the angle data in degree units stored in D0. Then, the result is stored in 3 word devices starting from D10 (D10~D12).

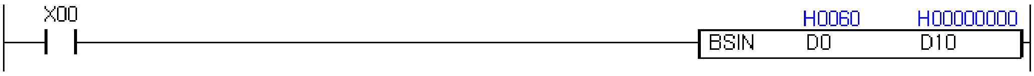
Ladder Diagram (LD)



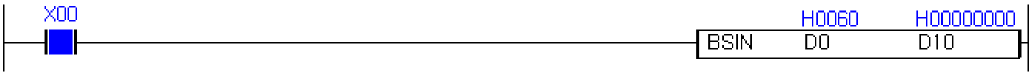
Instruction List (IL)

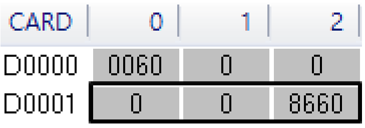
Instruction	Device	
LD	X00	
BSIN	D0	D10

- Angle data in degree unit “60” is assigned to word device D0.



- When X00 turns ON, BSIN instruction is executed. Then, the result “0.8660” is stored in 3 word devices starting from D10 (D10~D12). As result is positive, “0” is stored in D10. The integer part “0” is stored in D11. The fractional part “8660” is stored in D12.





CARD	0	1	2
D0000	0060	0	0
D0001	0	0	8660

2.17.8 BCOS, BCOSP

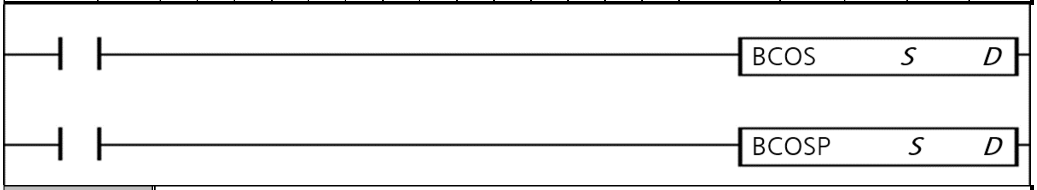
Supported PLC Series

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

Function

BCOS(P) instructions calculate the cosine of the angle data in degree units (4-digit BCD) stored in word device *S*. The result is stored in the 3 word devices starting from *D*.

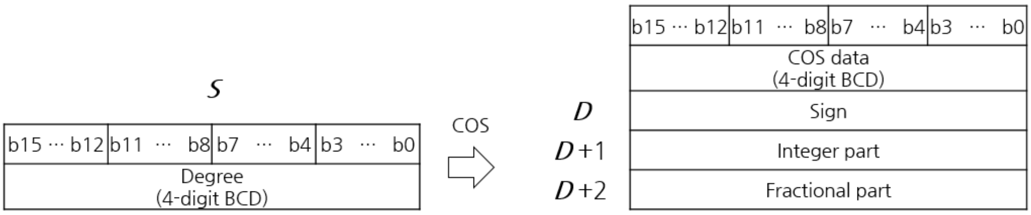
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BCOS(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Angle data in degree units or address of word device where angle data in degree units to be calculated is stored. (4-digit BCD)
D	Head address of 3 word devices where the calculation (Cosine) result will be stored.

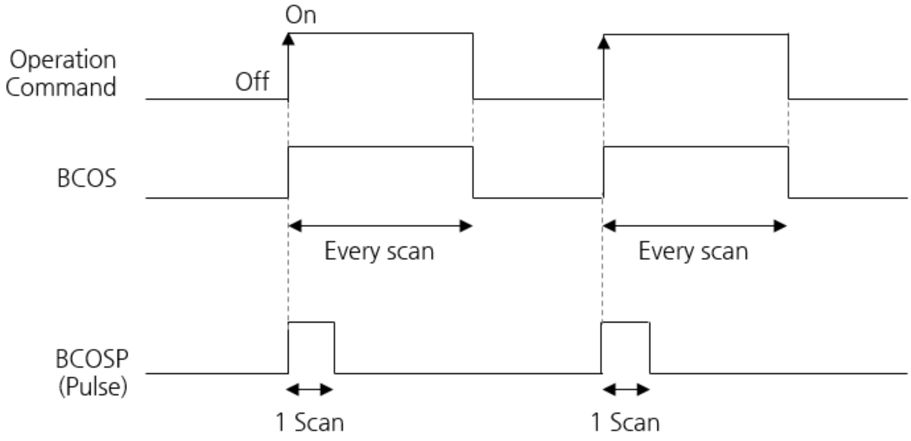
BCOS, BCOSP

- BCOS(P) instructions calculate the cosine of the angle data in degree units (4-digit BCD) stored in word device *S*. The result is stored in the 3 word devices starting from *D*.


S
b15 ··· b12 b11 ··· b8 b7 ··· b4 b3 ··· b0
Degree
(4-digit BCD)
COS
D
D+1
D+2
b15 ··· b12 b11 ··· b8 b7 ··· b4 b3 ··· b0
COS data
(4-digit BCD)
Sign
Integer part
Fractional part

- The value stored in double word device *S* has to be in degree unit.
- The range of BCD data assigned to word device *S* is as below:
  0 ≦ Value assigned to *S* ≦ 360
- If the value assigned to *S* is equal to or greater than 360, the angle data in degree unit is specified by the remainder of (value assigned to *S*)/360. (e.g. When value assigned to *S* is 362, it is equivalent to the case when the value assigned to *S* is 2.)
- When the calculation result is positive, "0" is stored in word device *D*. If the result is negative, "1" is stored in word device *D*.
- The calculation result is stored in BCD data. The range of result is as below:
  -1.0000 ≦ Calculation result ≦ 1.0000
- The calculation result is rounded off from the fifth decimal place.

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)

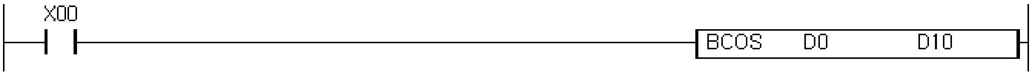
F110 turns ON for 1 scan when the value assigned to *S* is not BCD data.

Program Example

BCOS

When X00 turns ON, BCOS instruction calculates cosine of the angle data in degree units stored in D0. Then, the result is stored in 3 word devices starting from D10 (D10~D12).

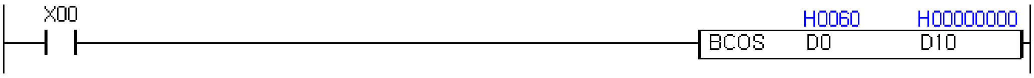
Ladder Diagram (LD)



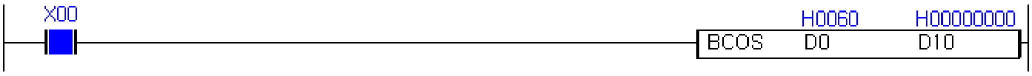
Instruction List (IL)

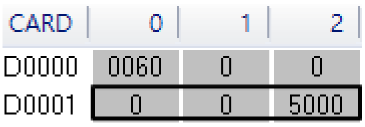
Instruction	Device	
LD	X00	
BCOS	D0	D10

- Angle data in degree unit “60” is assigned to word device D0.



- When X00 turns ON, BCOS instruction is executed. Then, the result “0.5000” is stored in 3 word devices starting from D10 (D10~D12). As result is positive, “0” is stored in D10. The integer part “0” is stored in D11. The fractional part “5000” is stored in D12.





CARD	0	1	2
D0000	0060	0	0
D0001	0	0	5000

2.17.9 BTAN, BTANP

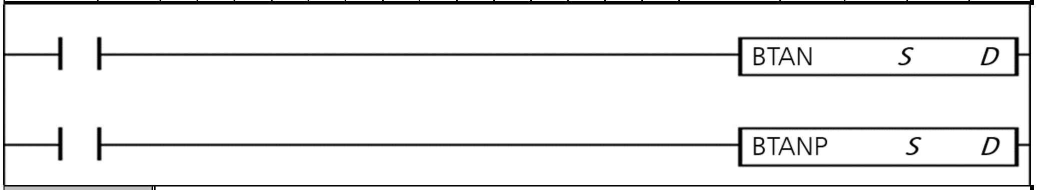
Supported PLC Series

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

Function

BTAN(P) instructions calculate the tangent of the angle data in degree units (4-digit BCD) stored in word device *S*. The result is stored in the 3 word devices starting from *D*.

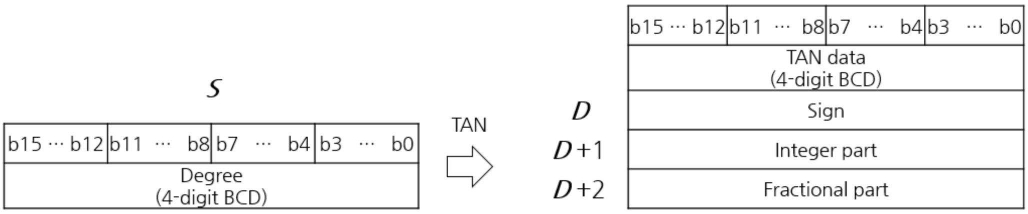
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BTAN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Angle data in degree units or address of word device where angle data in degree units to be calculated is stored. (4-digit BCD)
D	Head address of 3 word devices where the calculation (Tangent) result will be stored.

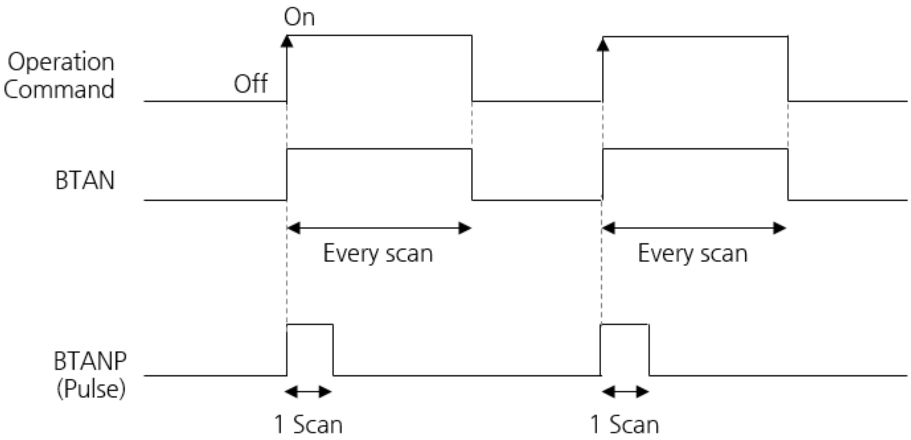
BTAN, BTANP

- BTAN(P) instructions calculate the tangent of the angle data in degree units (4-digit BCD) stored in word device *S*. The result is stored in the 3 word devices starting from *D*.


S
b15 ··· b12 b11 ··· b8 b7 ··· b4 b3 ··· b0
Degree
(4-digit BCD)
TAN
D
D+1
D+2
b15 ··· b12 b11 ··· b8 b7 ··· b4 b3 ··· b0
TAN data
(4-digit BCD)
Sign
Integer part
Fractional part

- The value stored in double word device *S* has to be in degree unit.
- The range of BCD data assigned to word device *S* is as below:
  0 ≦ Value assigned to *S* ≦ 360
- If the value assigned to *S* is equal to or greater than 360, the angle data in degree unit is specified by the remainder of (value assigned to *S*)/360. (e.g. When value assigned to *S* is 362, it is equivalent to the case when the value assigned to *S* is 2.)
- When the calculation result is positive, “0” is stored in word device *D*. If the result is negative, “1” is stored in word device *D*.
- The calculation result is stored in BCD data. The range of result is as below:
  -52.2900 ≦ Calculation result ≦ 27.2900
- The calculation result is rounded off from the fifth decimal place.

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)

F110 turns ON for 1 scan when the value assigned to *S* is not BCD data.

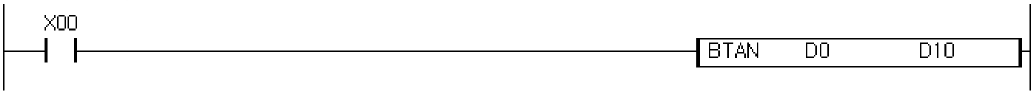
F110 turns ON for 1 scan when the value assigned to *S* is “90” or “270”.

Program Example

BTAN

When X00 turns ON, BTAN instruction calculates tangent of the angle data in degree units stored in D0. Then, the result is stored in 3 word devices starting from D10 (D10~D12).

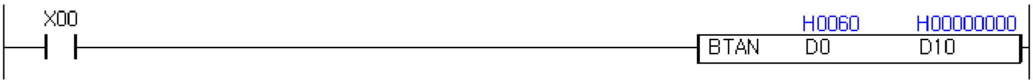
Ladder Diagram (LD)



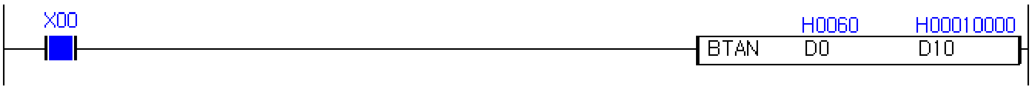
Instruction List (IL)

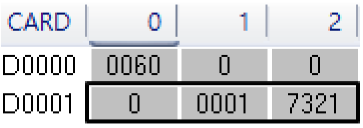
Instruction	Device	
LD	X00	
BTAN	D0	D10

- Angle data in degree unit “60” is assigned to word device D0.



- When X00 turns ON, BTAN instruction is executed. Then, the result “1.7321” is stored in 3 word devices starting from D10 (D10~D12). As result is positive, “0” is stored in D10. The integer part “1” is stored in D11. The fractional part “7321” is stored in D12.





CARD	0	1	2
D0000	0060	0	0
D0001	0	0001	7321

2.17.10 BASIN, BASINP

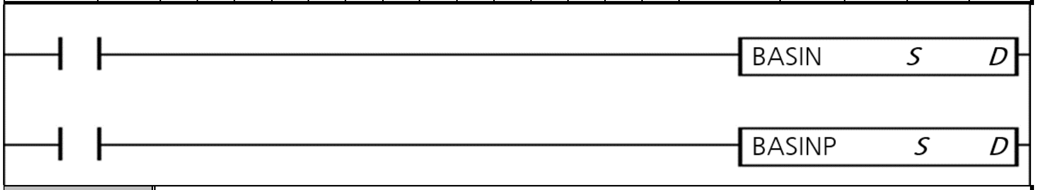
Supported PLC Series

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

Function

BASIN(P) instructions calculate the inverse sine of the SIN data (4-digit BCD) stored in 3 word devices starting from *S*. The result is stored in the double word device *D*.

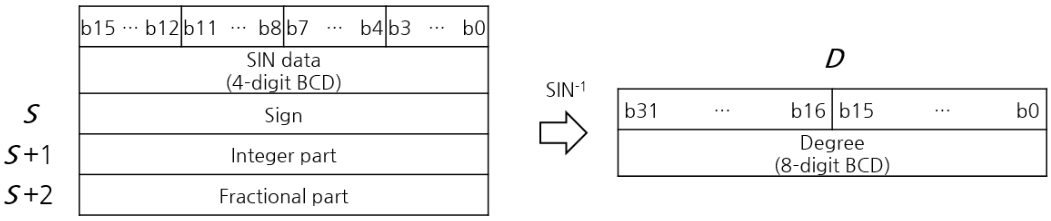
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BASIN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	SIN data or head address of 3 word devices where SIN data to be calculated is stored. (4-digit BCD)
D	Head address of double word device where the calculation (Inverse sine) result will be stored.

BASIN, BASINP

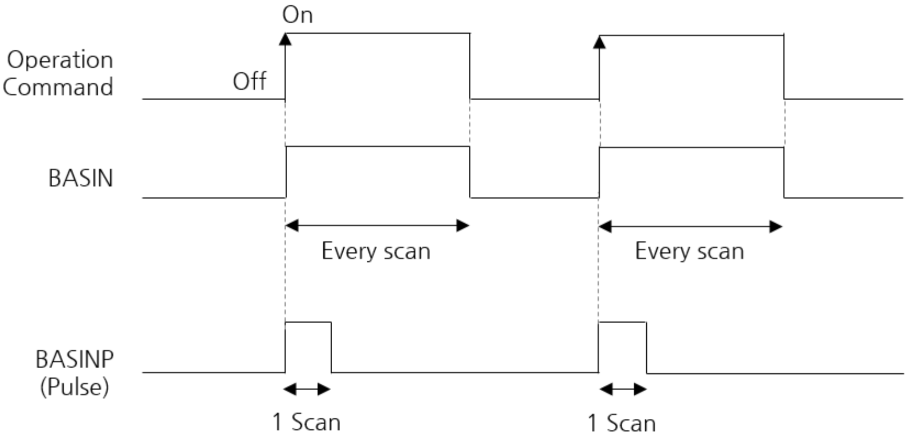
- BASIN(P) instructions calculate the inverse sine of the SIN data (4-digit BCD) stored in 3 word devices starting from *S*. The result is stored in the double word device *D*.



- The value stored in 3 word devices starting from *S* has to be SIN data.
  The range of SIN data is from -1.0000 to 1.0000
- When SIN data to be calculated is positive, assign “0” to *S*. If negative, assign “1” or greater value to *S*.
- Assign integer part of SIN data to *S*+1. Assign fractional part of SIN data to *S*+2. Values assigned to *S*+1 and *S*+2 cannot exceed 1.0000.
- The calculation result stored in double word device *D* is in degree units.
- Frantional part of the result is rounded.
- The BCD values stored in *D* are from 0 to 90 degrees and from 270 to 360 degrees.

(*) For information on conversion between radians and degrees, refer to DEG or RAD instruction.

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)

F110 turns ON for 1 scan when the value assigned to *S*, *S*+1, *S*+2 is not BCD data.

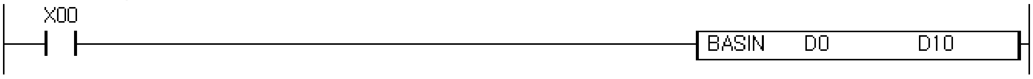
F110 turns ON for 1 scan when the value assigned to *S*, *S*+1, *S*+2 exceeds following range:
-1.0000 ≦ Assigned value ≦ 1.0000

Program Example

BASIN

When X00 turns ON, BASIN instruction calculates inverse sine of the SIN data stored in D0(D0~D2). Then, the result is stored in D10 (D10~D11), in degree units.

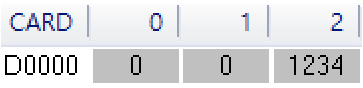
Ladder Diagram (LD)



Instruction List (IL)

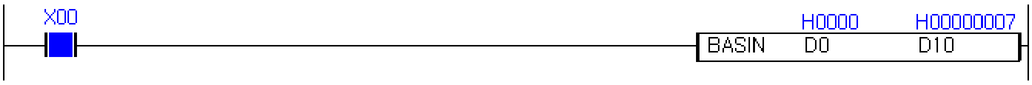
Instruction	Device	
LD	X00	
BASIN	D0	D10

- SIN data “0.1234” is assigned to 3 devices starting from D0. Positive sign “0” is assigned to D0. Integer part “0” is assigned to D1. Fractional part “1234” is assigned to D2.



CARD	0	1	2
D0000	0	0	1234

- When X00 turns ON, BASIN instruction is executed. Then, the result “7” is stored in double word device D10 (D10~D11).



2.17.11 BACOS, BACOSP

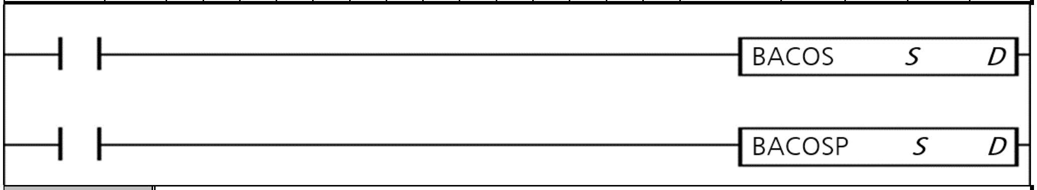
Supported PLC Series

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

Function

BACOS(P) instructions calculate the inverse cosine of the COS data (4-digit BCD) stored in 3 word devices starting from *S*. The result is stored in the double word device *D*.

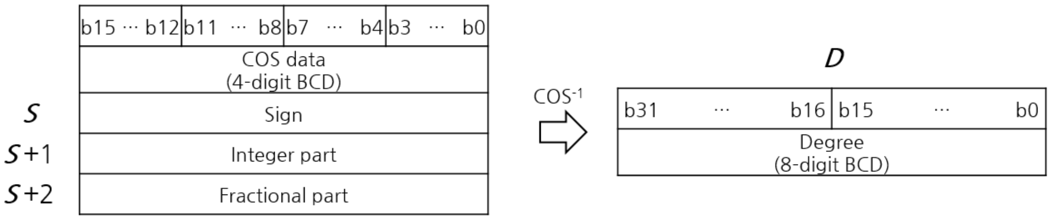
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BACOS(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	COS data or head address of 3 word devices where COS data to be calculated is stored. (4-digit BCD)
D	Head address of double word device where the calculation (Inverse cosine) result will be stored.

BACOS, BACOSP

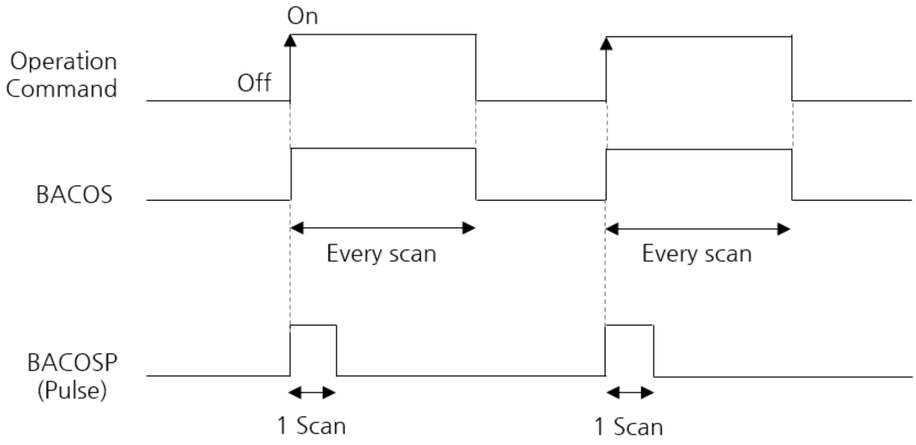
- BACOS(P) instructions calculate the inverse cosine of the COS data (4-digit BCD) stored in 3 word devices starting from *S*. The result is stored in the double word device *D*.



- The value stored in 3 word devices starting from *S* has to be COS data. The range of COS data is from -1.000 to 1.0000.
- When COS data to be calculated is positive, assign “0” to *S*. If negative, assign “1” or greater value to *S*.
- Assign integer part of COS data to *S*+1. Assign fractional part of COS data to *S*+2. Values assigned to *S*+1 and *S*+2 cannot exceed 1.0000.
- The calculation result stored in double word device *D* is in degree units.
- Frantional part of the result is rounded.
- The BCD values stored in *D* are from 0 to 180 degrees.

(*) For information on conversion between radians and degrees, refer to DEG or RAD instruction.

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)

F110 turns ON for 1 scan when the value assigned to *S*, *S*+1, *S*+2 is not BCD data.

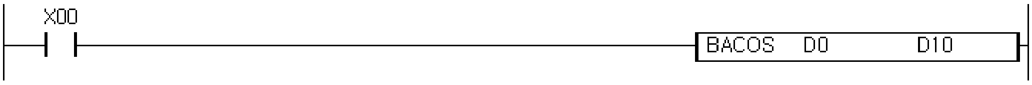
F110 turns ON for 1 scan when the value assigned to *S*, *S*+1, *S*+2 exceeds following range: -1.0000 ≦ Assigned value ≦ 1.0000

Program Example

BACOS

When X00 turns ON, BACOS instruction calculates inverse cosine of the COS data stored in D0(D0~D2). Then, the result is stored in D10 (D10~D11), in degree units.

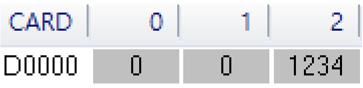
Ladder Diagram (LD)



Instruction List (IL)

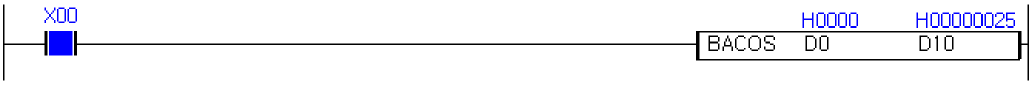
Instruction	Device	
LD	X00	
BACOS	D0	D10

- COS data “0.1234” is assigned to 3 devices starting from D0. Positive sign “0” is assigned to D0. Integer part “0” is assigned to D1. Fractional part “1234” is assigned to D2.



CARD	0	1	2
D0000	0	0	1234

- When X00 turns ON, BACOS instruction is executed. Then, the result “25” is stored in double word device D10 (D10~D11).



2.17.12 BATAN, BATANP

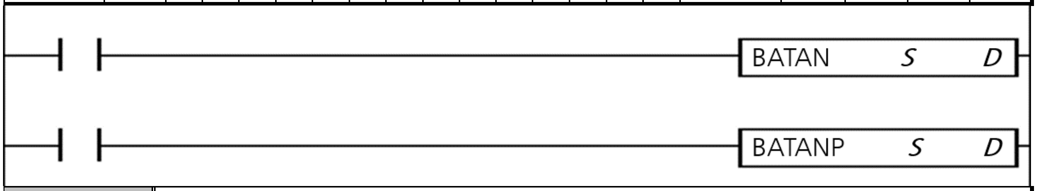
Supported PLC Series

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

Function

BATAN(P) instructions calculate the inverse tangent of the TAN data (4-digit BCD) stored in 3 word devices starting from *S*. The result is stored in the double word device *D*.

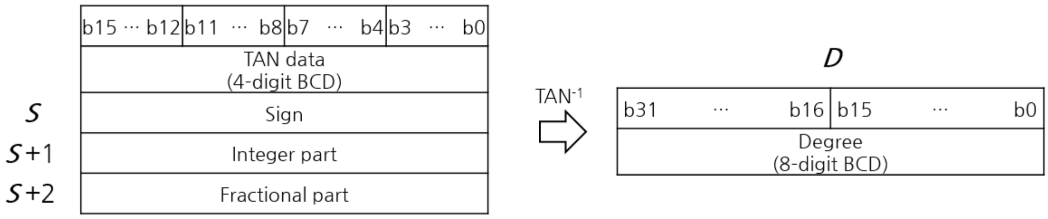
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BATAN(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	TAN data or head address of 3 word devices where TAN data to be calculated is stored. (4-digit BCD)
D	Head address of double word device where the calculation (Inverse tangent) result will be stored.

BATAN, BATANP

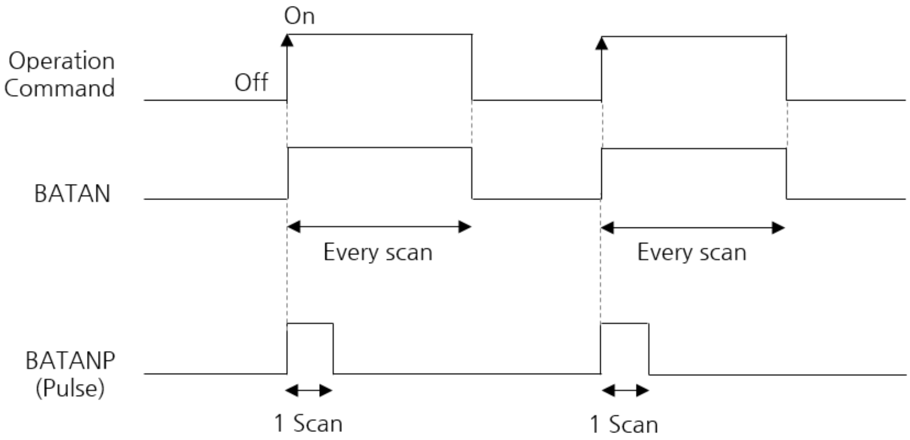
- BATAN(P) instructions calculate the inverse tangent of the TAN data (4-digit BCD) stored in 3 word devices starting from *S*. The result is stored in the double word device *D*.



- The value stored in 3 word devices starting from *S* has to be TAN data. The range of TAN data is from -9999.9999 to 9999.9999.
- When TAN data to be calculated is positive, assign “0” to *S*. If negative, assign “1” or greater value to *S*.
- Assign integer part of TAN data to *S*+1. Assign fractional part of TAN data to *S*+2. Values assigned to *S*+1 and *S*+2 cannot exceed 9999.9999.
- The calculation result stored in double word device *D* is in degree units.
- Frantional part of the result is rounded.
- The BCD values stored in *D* are from 0 to 90 degrees and from 270 to 360 degrees.

(*) For information on conversion between radians and degrees, refer to DEG or RAD instruction.

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)

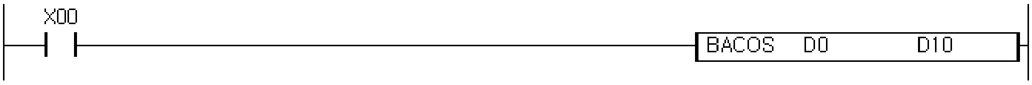
F110 turns ON for 1 scan when the value assigned to *S*, *S*+1, *S*+2 is not BCD data.

Program Example

BATAN

When X00 turns ON, BATAN instruction calculates inverse sine of the TAN data stored in D0(D0~D2). Then, the result is stored in D10 (D10~D11), in degree units.

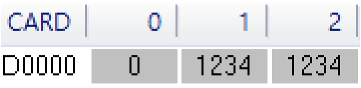
Ladder Diagram (LD)



Instruction List (IL)

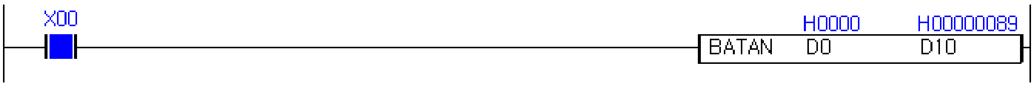
Instruction	Device	
LD	X00	
BATAN	D0	D10

- TAN data “1234.1234” is assigned to 3 devices starting from D0. Positive sign “0” is assigned to D0. Integer part “1234” is assigned to D1. Fractional part “1234” is assigned to D2.



CARD	0	1	2
D0000	0	1234	1234

- When X00 turns ON, BATAN instruction is executed. Then, the result “89” is stored in double word device D10 (D10~D11).



2.17.13 RAD, RADP

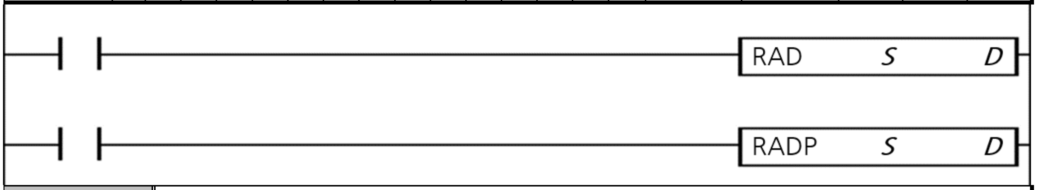
Supported PLC Series

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

Function

RAD(P) instructions convert angle data (°) stored in double word device *S* into radian unit. Then, the conversion result is stored in double word device *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
RAD(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



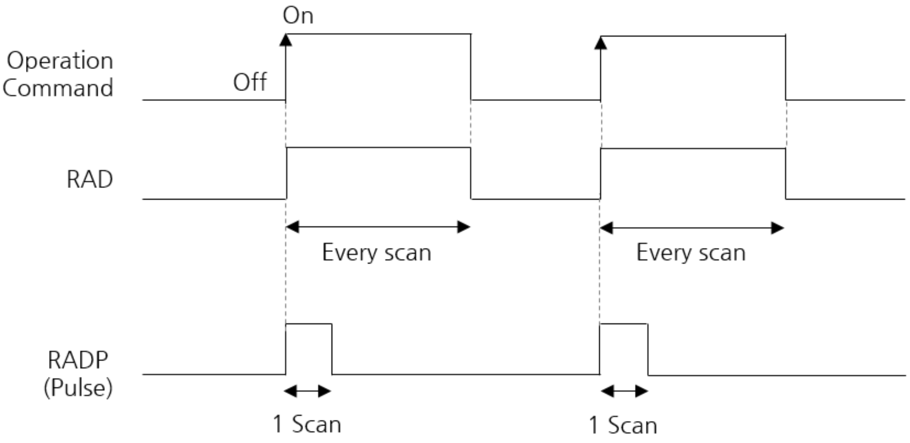
S	Angle or head address of double word device where angle to be converted is stored. (32-bit floating-point number)
D	Head address of double word device where conversion result (Radian unit) will be stored.

RAD, RADP

- RAD(P) instructions convert angle data (32-bit floating-point number) stored in double word device *S* into radian data (32-bit floating-point number).
- Converted data is stored in the double word device *D*.
- Data is converted according to the following equation:

  Radian = Degree $\times \frac{\pi}{180}$
- Values which can be assigned to device *S* is as following.
  -360° ≦ value assigned to device *S* ≦ 360°

Execution Condition



Operation Error

Error flag (F110)

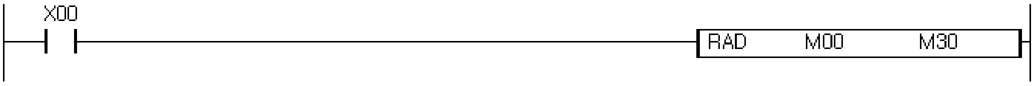
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

RAD

When operation command X00 is ON, RAD instruction is executed. Then, angle data stored in double word device M00 (M00 ~ M1F) is converted into radian unit. The conversion result is stored in double word device M30 (M30 ~ M4F).

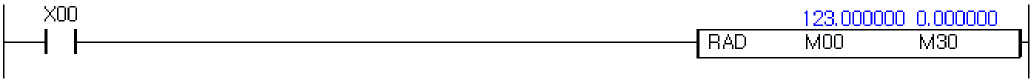
Ladder Diagram (LD)



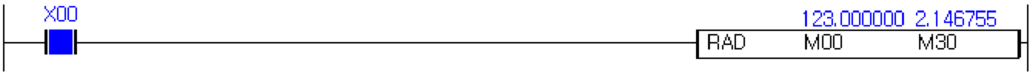
Instruction List (IL)

Instruction	Device	
LD	X00	
RAD	M00	M30

- Angle data in 32-bit floating-point number “123.000000” is assigned to M00.



- When X00 turns ON, RAD instruction is executed. Then, angle data “123.000000” is converted into “2.146755” in radian unit. The conversion result is stored in M30 (M30 ~ M4F).



2.17.14 DEG, DEGP

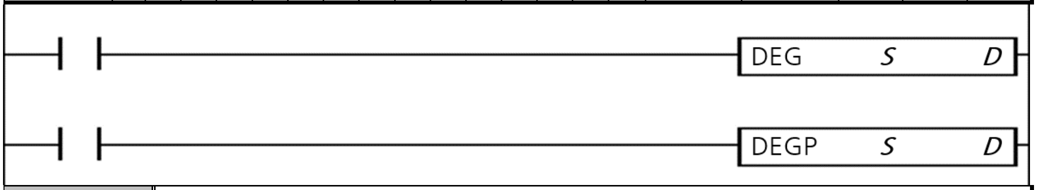
Supported PLC Series

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

Function

DEG(P) instructions convert radian data stored in double word device *S* into angle data (°). Then, the conversion result is stored in double word device *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
DEG(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	-	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



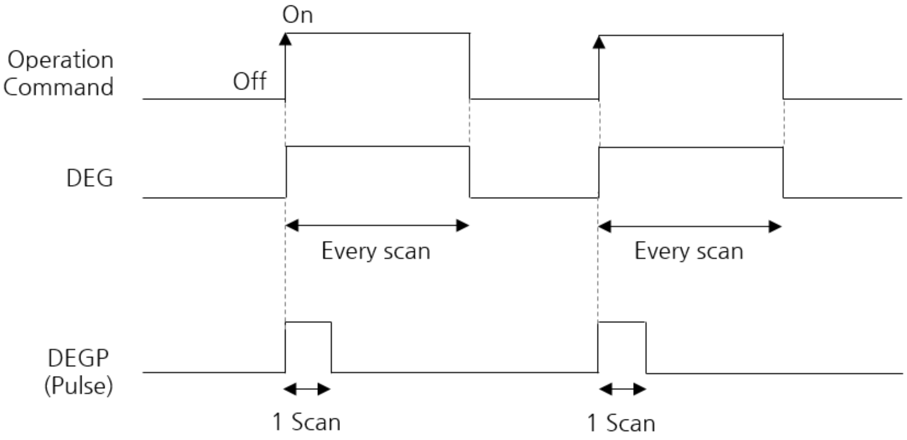
S	Radian or head address of double word device where radian data to be converted is stored. (32-bit floating-point number)
D	Head address of double word device where conversion result (angle data) will be stored.

DEG, DEGP

- DEG(P) instructions convert radian data (32-bit floating-point number) stored in double word device *S* into angle data (32-bit floating-point number).
- Converted data is stored in the double word device *D*.
- Data is converted according to the following equation:

  Degree = Radian $\times \frac{\mathbf{180}}{\boldsymbol{\pi}}$
- Values which can be assigned to device *S* is as following.
  -6.28 ≤ value assigned to device *S* ≤ 6.28

Execution Condition



Operation Error

Error flag (F110)

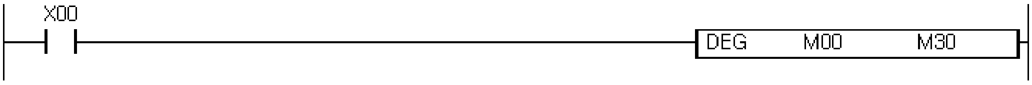
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

DEG

When operation command X00 is ON, DEG instruction is executed. Then, angle data stored in double word device M00 (M00 ~ M1F) is converted into angle data (°). The conversion result is stored in double word device M30 (M30 ~ M4F).

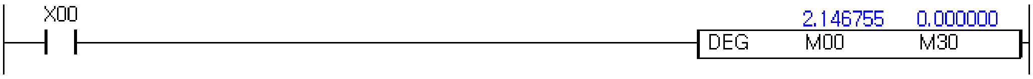
Ladder Diagram (LD)



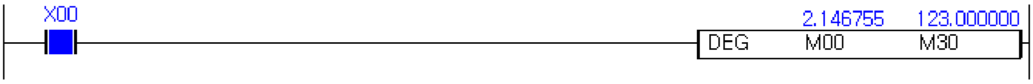
Instruction List (IL)

Instruction	Device	
LD	X00	
DEG	M00	M30

- Radian data in 32-bit floating-point number “2.146755” is assigned to M00.



- When X00 turns ON, DEG instruction is executed. Then, radian data “2.146755” is converted into “123.000000” in degree unit. The conversion result is stored in M30 (M30 ~ M4F).



2.17.15 SQR, SQRP

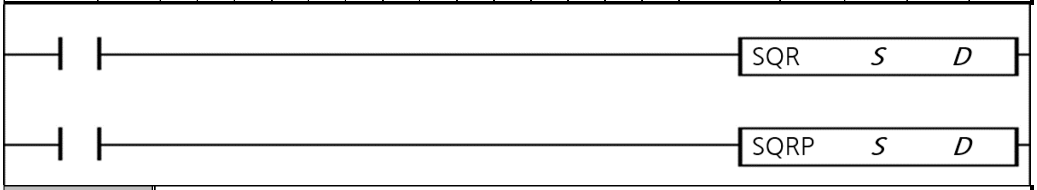
Supported PLC Series

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

Function

SQR(P) instructions calculate the square root of the 32-bit floating-point number stored in double word device *S*. The result is stored in the double word device *D*.

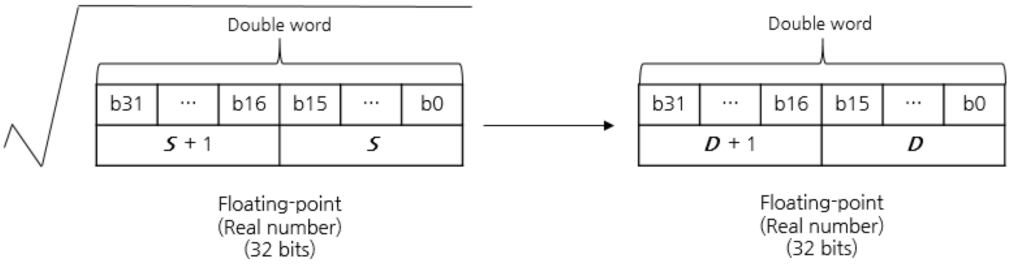
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
SQR(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Constant, floating-point number or head address of double word device where data to be calculated (square root) is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation result will be stored.

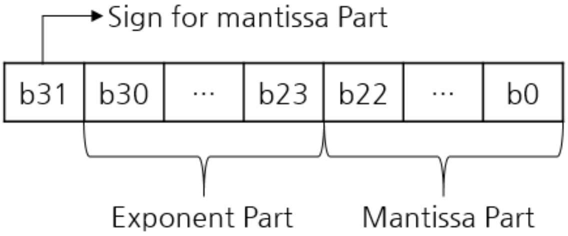
SQR, SQRP

- SQR(P) instructions calculate the squre root of the 32-bit floating-point number stored in double word device *S*. The result is stored in the double word device *D*.



- The range of value assigned to double word device *S* and the operation result is as below:
  0, 2^{-127} ≦ Value assigned to *S* ≦ 2^{128}

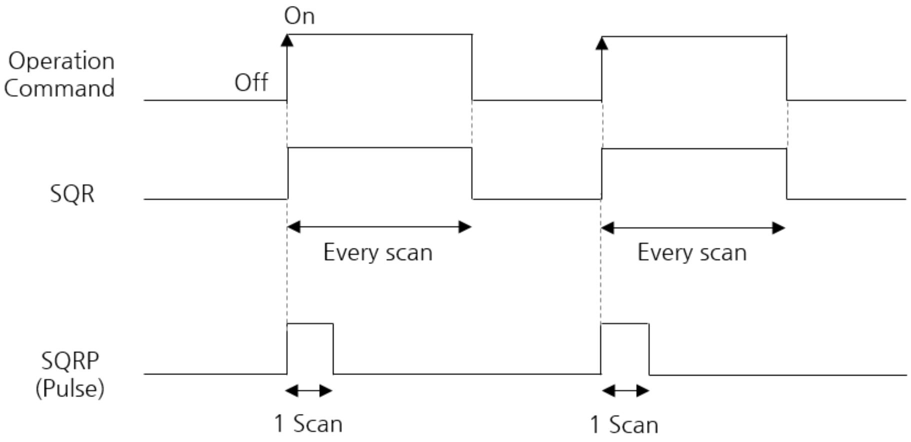
- If the operation result exceeds its range, “1.#INF[21]00” is stored in the device *D*.



- Only positive numbers can be assigned to double word device *S*. It means that bit 31 of double word device *S* always has to be “0”.

[21] INF: Stands for infinity. “1.#INF00” appears when the assigned value cannot be expressed with 32-bit floating-point number.

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)

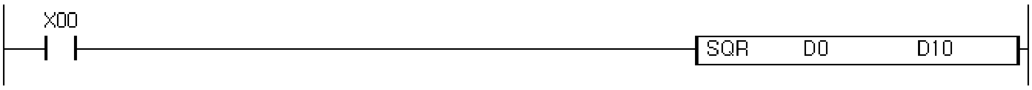
F110 turns ON for 1 scan when the value assigned to *S* is negative.

Program Example

SQR

When X00 turns ON, SQR instruction calculates the square root of the 32-bit floating-point number stored in D0 (D0~D1). Then, the result is stored in double word devices starting from D10 (D10~D11).

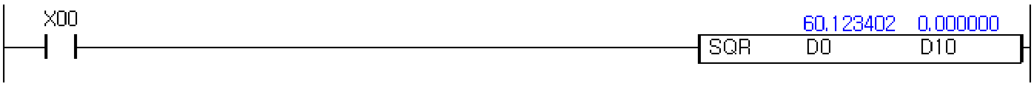
Ladder Diagram (LD)



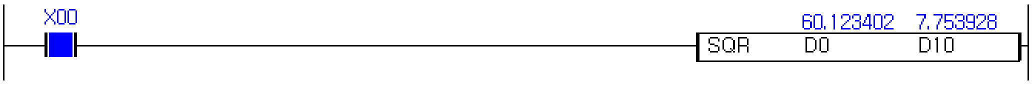
Instruction List (IL)

Instruction	Device	
LD	X00	
SQR	D0	D10

- 32-bit floating-point number “60.123402” is assigned to double word device D0 (D0~D1).



- When X00 turns ON, SQR instruction is executed. Then, the result “7.753928” is stored in double word device D10 (D10~D11).



2.17.16 BSQR, BSQRP, BDSQR, BDSQRP

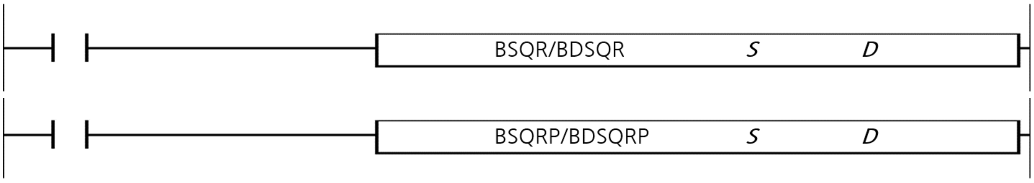
Supported PLC Series

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

Function

BSQR(P) instructions calculate the square root of the 4-digit BCD stored in word device *S*. The result is stored in the double word device *D*.
BDSQR(P) instructions calculate the square root of the 8-digit BCD stored in double word device *S*. The result is stored in the double word device *D*.

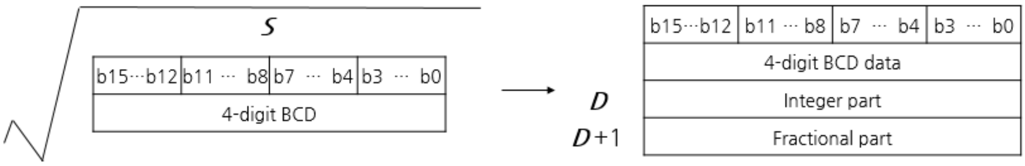
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
BSQR(P) BDSQR(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Constant or head address of word/double word device where data to be calculated (square root) is stored.
D	Head address of double word device where the calculation result will be stored.

BSQR, BSQRP

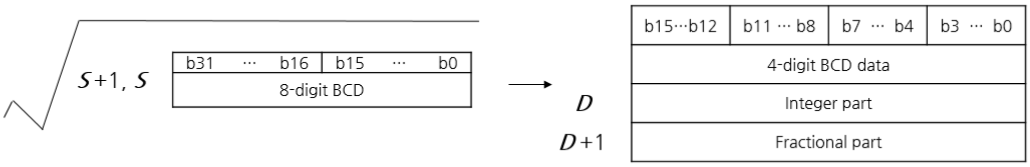
- BSQR(P) instructions calculate the squre root of the 4-digit BCD stored in word device *S*. The result is stored in the double word device *D*.



- In device *D*, integer part of the calculation result is stored as 4-digit BCD. In device *D*+1, fractional part of the result is stored as 4-digit BCD.
- The range of value assigned to word device *S, D* and *D*+1 is as below:
  0 ≦ Value assigned to *S, D* and *D*+1 ≦ 9999
- The calculation result is rounded off from the fifth decimal place.

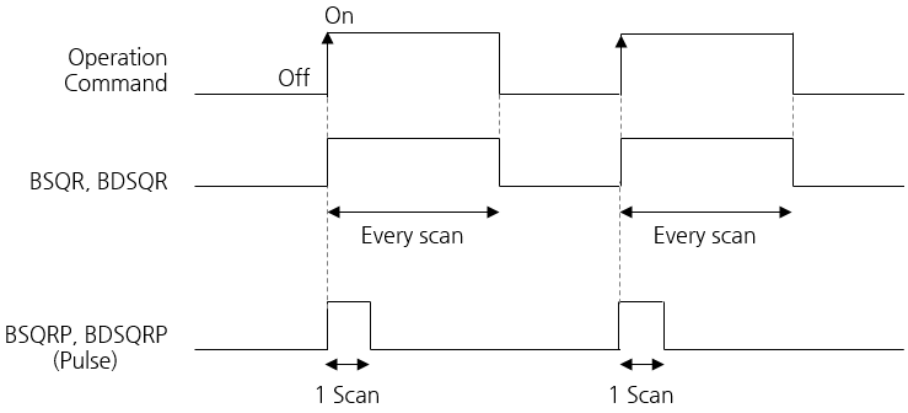
BDSQR, BDSQRP

- BDSQR(P) instructions calculate the squre root of the 8-digit BCD stored in double word device *S*. The result is stored in the double word device *D*.



- In device *D*, integer part of the calculation result is stored as 4-digit BCD. In device *D*+1, fractional part of the result is stored as 4-digit BCD.
- The range of value assigned to word device *S, D* and *D*+1 is as below:
  0 ≦ Value assigned to *S* ≦ 99999999
  0 ≦ Value assigned to *D* and *D*+1 ≦ 9999
- The calculation result is rounded off from the fifth decimal place.

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)

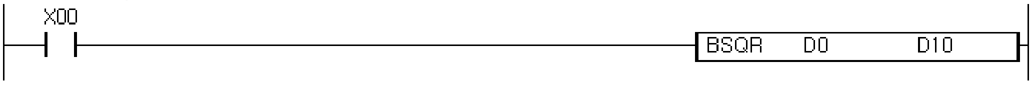
F110 turns ON for 1 scan when the value assigned to *S* is negative.

Program Example

BSQR

When X00 turns ON, BSQR instruction calculates the square root of the 4-digit BCD stored in D0. Then, the result is stored in double word devices starting from D10 (D10~D11).

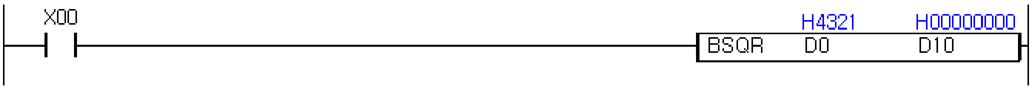
Ladder Diagram (LD)



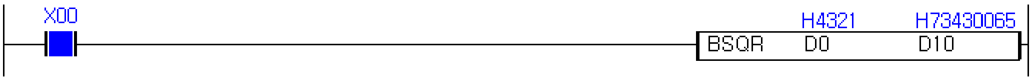
Instruction List (IL)

Instruction	Device	
LD	X00	
BSQR	D0	D10

- 4-digit BCD “4321” is assigned to word device D0.



- When X00 turns ON, BSQR instruction is executed. Then, the result “65.7343” is stored in double word device D10 (D10~D11).



2.17.17 EXP, EXPP

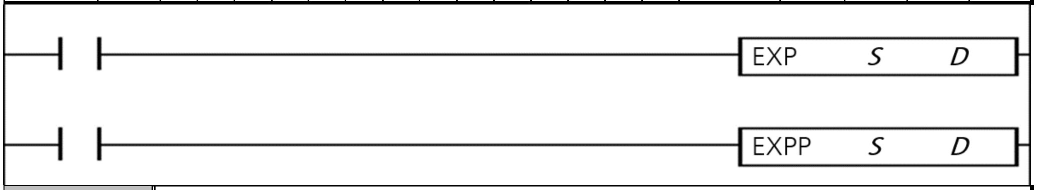
Supported PLC Series

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

Function

EXP(P) instructions calculate the natural exponential (base e) raised to the power of 32-bit floating-point number stored in double word device *S*. The result is stored in the double word device *D*.

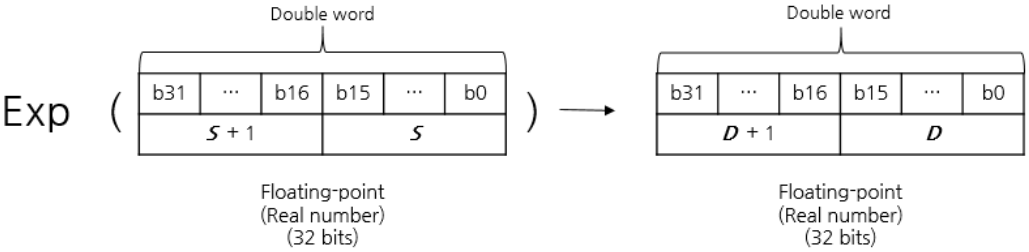
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
EXP(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Constant, floating-point number or head address of double word device where exponent of the exponential constant e is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation result will be stored.

EXP, EXPP

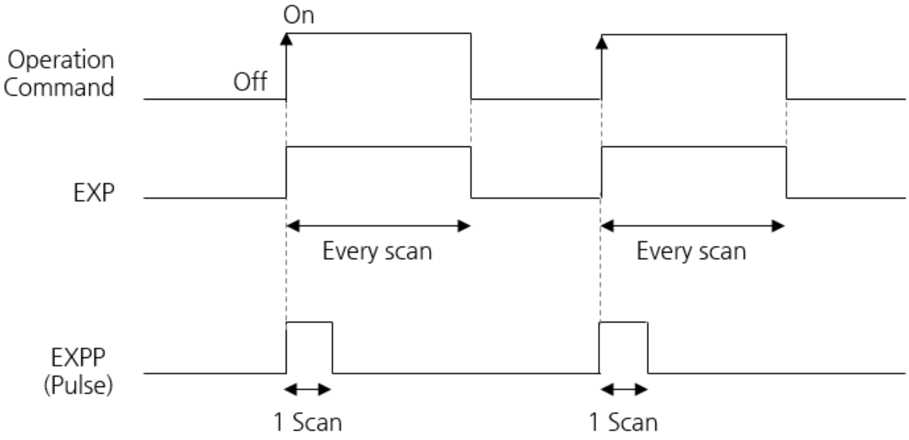
- EXP(P) instructions calculate the natural exponential (base e) raised to the power of 32-bit floating-point number stored in double word device *S*. The result is stored in the double word device *D*.



- EXP(P) instructions take the format of e^{S}.
- The range of the value assigned to double word device *S* and *D* is as below: 2^{-127} ≤| Value assigned to device *S*, *D* |≤ 2^{128}
- If the operation result exceeds its range, "1.#INF[22]00" is stored in the device *D*.
- In the natural exponential calculation, PLC CPU regards constant e as "2.71828".

[22] INF: Stands for infinity. "1.#INF00" appears when the assigned value cannot be expressed with 32-bit floating-point number.

Execution Condition



Operation Error

Error flag (F110)

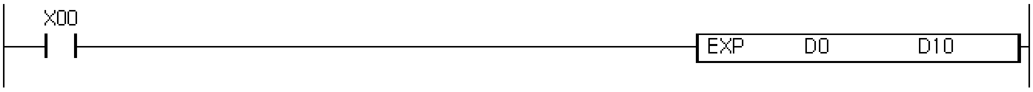
F110 turns ON for 1 scan when the address of device specified by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

EXP

When X00 turns ON, EXP instruction calculates the natural exponential of the 32-bit floating-point number stored in D0 (D0~D1). Then, the result is stored in double word device D10 (D10~D11).

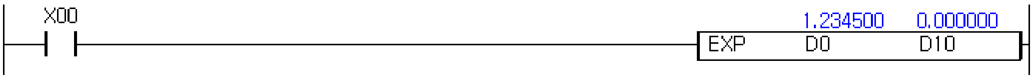
Ladder Diagram (LD)



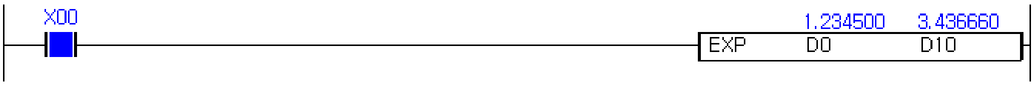
Instruction List (IL)

Instruction	Device	
LD	X00	
EXP	D0	D10

- “1.234500” is assigned to double word device D0.



- When X00 turns ON, EXP instruction is executed. Then, the instruction calculates $e^{1.2345}$. The result “3.436660” is stored in double word device D10 (D10~D11).



2.17.18 LOG, LOGP

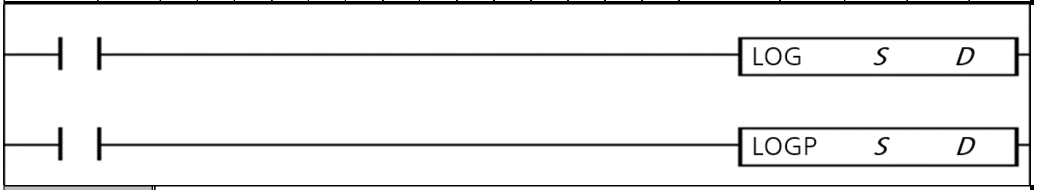
Supported PLC Series

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

Function

LOG(P) instructions calculate the natural logarithm of the 32-bit floating-point number stored in double word device *S*. The result is stored in the double word device *D*.

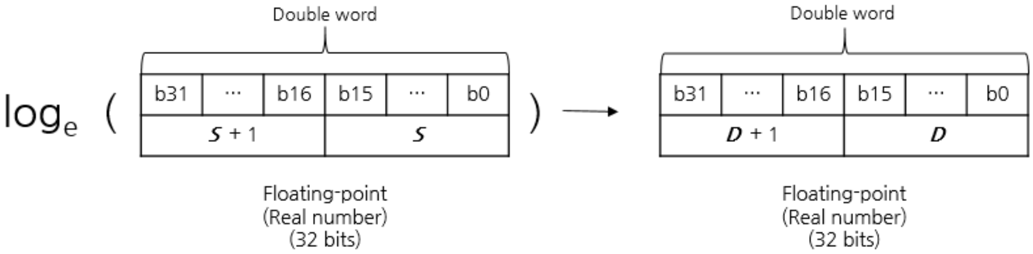
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
LOG(P)	*S*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	-
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



S	Constant, floating-point number or head address of double word device where data to be calculated is stored. (32-bit floating-point number)
D	Head address of double word device where the calculation result will be stored.

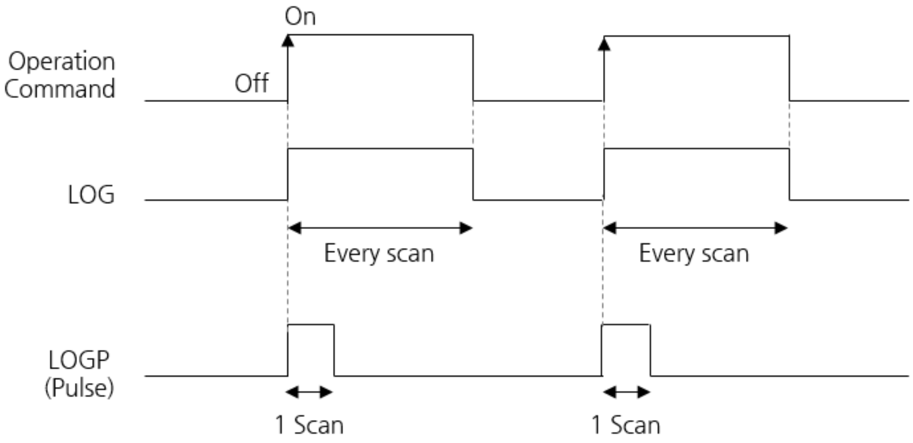
LOG, LOGP

- LOG(P) instructions calculate the natural logarithm of the 32-bit floating-point number stored in double word device *S*. The result is stored in the double word device *D*.



- The range of the value assigned to double word device *S* and *D* is as below:
  2^{-127} ≤| Value assigned to device *S*, *D* |≤ 2^{128}
- In the natural exponential calculation, PLC CPU regards constant e as “2.71828”.

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)

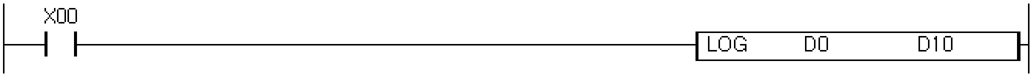
F110 turns ON for 1 scan when the value assigned to *S* is negative or "-0".

Program Example

LOG

When X00 turns ON, LOG instruction calculates logarithm of the 32-bit floating-point number stored in D0(D0~D1). Then, the result is stored in D10 (D10~D11).

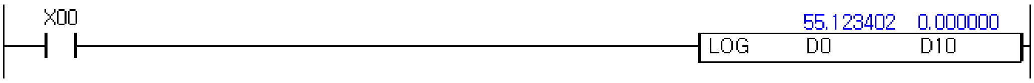
Ladder Diagram (LD)



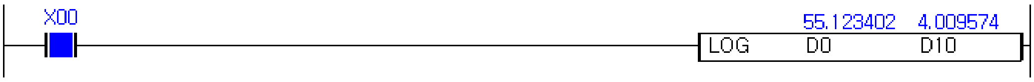
Instruction List (IL)

Instruction	Device	
LD	X00	
LOG	D0	D10

- "55.123402" is assigned to double word device D0 (D0~D1).



- When X00 turns ON, LOG instruction is executed. Then, the result "4.009574" is stored in double word device D10 (D10~D11).



2.17.19 RND, RNDP

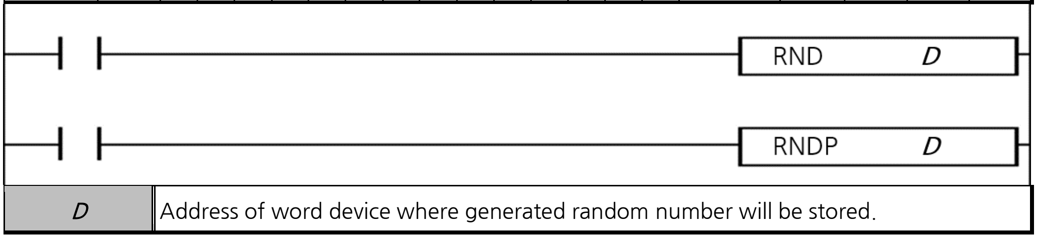
Supported PLC Series

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

RND(P) instructions generate random number and store the number in word device *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
RND(P)	*D*	O	-	O	O	O	-	O	O	-	O	O	O	O	O	-	2	O	-	-

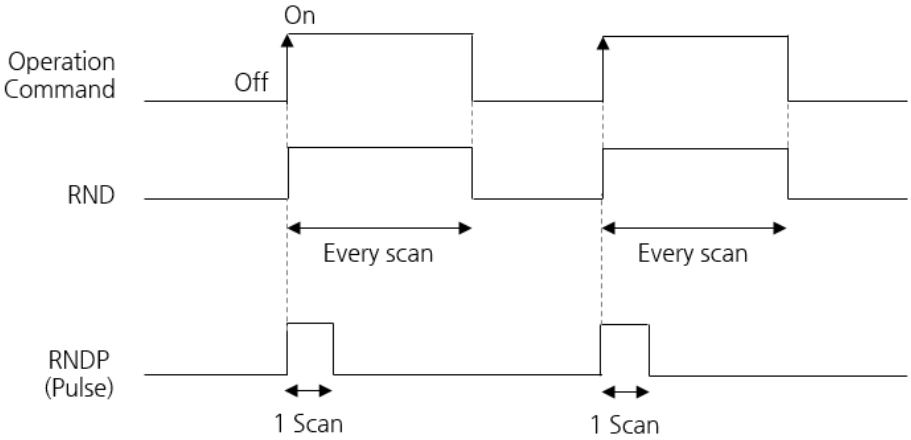


D	Address of word device where generated random number will be stored.

RND, RNDP

- RND(P) instructions generate random number and store the number in the word device *D*.
- The range of random number is from 0 to 32767.
- Generation of random number is on basis of the internal calculation formula.

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)

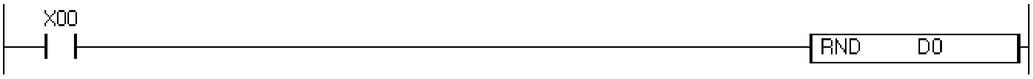
Program Example

RND

Following program example operates as following:

- When X00 turns ON, RND instruction is executed.
- Random numbers are generated and stored in D0.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
RND	D0

2.17.20 SRND, SRNDP

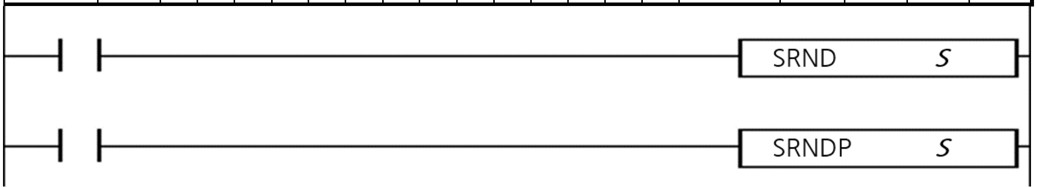
Supported PLC Series

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

Function

SRND(P) instructions change the pattern of random number generator based on the value assigned to word device *S*.

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

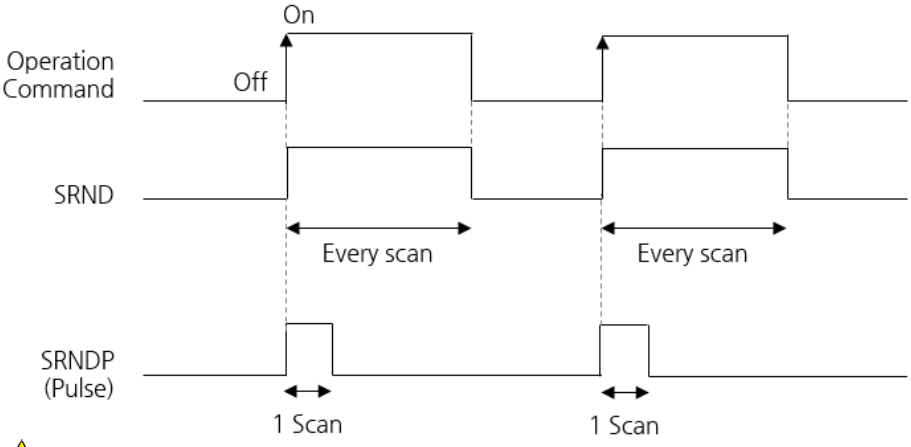


S	Address of word device where random number sequence data is stored. (16-bit BIN data)

SRND, SRNDP

- SRND(P) instructions change the pattern of random number generator based on the value assigned to word device *S*.
- You can assign 16-bit BIN data to the device *S*.

Execution Condition



⚠ Using pulse contact as execution condition of SRND instruction or using SRNDP instruction is recommended.

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)

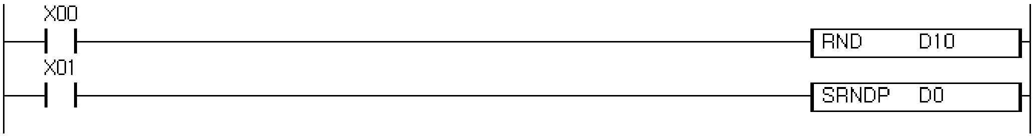
Program Example

SRNDP

Following program example operates as following:

- When X00 turns ON, RND instruction is executed.
- A random number is stored in D10.
- When X01 turns ON, SRNDP instruction is executed.
- According to the value assigned to D0, pattern of the RND instruction changes.
- Random numbers with different pattern are generated and stored in D10.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
RND	D10
LD	X01
SRNDP	D0

2.18 Special Module Access Instruction

2.18.1 FREAD, FREADP

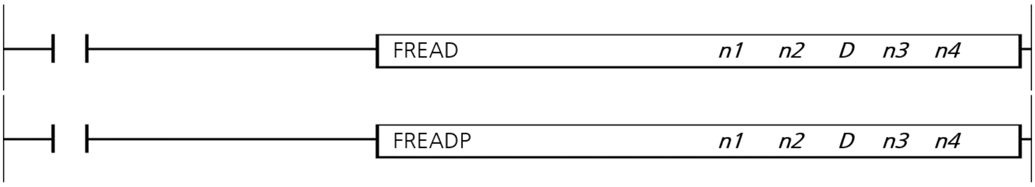
Supported PLC Series

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

Function

FREAD(P) read file data from system memory of a special module and store the data in the internal memory of CPU.

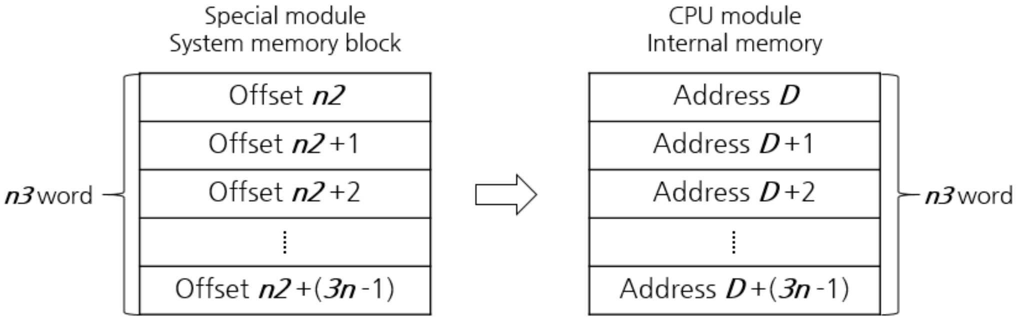
Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FREAD(P)	*n1*	O	O	O	O	O	O	O	O	O	O	O	O	O	-	O	6	O	-	-
	n2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	O	O	O	O	O	-	-	-	-	O	-	O	O	O	-				
	n3	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	n4	O	-	O	O	O	-	-	-	-	-	-	O	O	O	-				



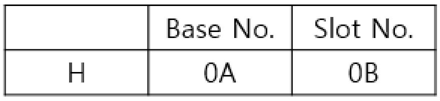
n1	Base and slot number of the special module in hexadecimal.
n2	Offset of file data to be read.
D	Head address of word devices where the file data will be stored.
n3	Size of data to be read (0 - 255).
n4	Head address of word devices to store the operation result. The result is expressed by flag instruction.

FREAD, FREADP

- FREAD and FREADP read data in *n3* words from the file data designated by offset *n2*.
- The file data is a system memory block of a special module which is equipped at *n1*.
- The read data is stored in the device memory assigned to *D* in *n3* words.
- Result of the operation is stored at word device assigned to *n4*.



Assigning *n1* - [Base/Slot]



	Base No.	Slot No.
H	0A	0B

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal.
Base No.: 2 characters in hexadecimal
Slot No.: 2 characters in hexadecimal

If Base No. is assigned to HFF, it means 'file'.
If Base No. is assigned to HFF, assign PID of the file at Slot No.

Base Number	Slot	S1
Local base	Slot No. 5	H0005 or 5
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 12	H0E0C
16th expansion	Slot No. 10	H100A

Assigning *n2* - [Offset of file data]

File data to read is a block of system memory. In other words, offset *n2* indicates the location of the data to be read.
(*) Refer to the System Data Memory of Positioning Module Manual.

Assigning *n3* - [Size]

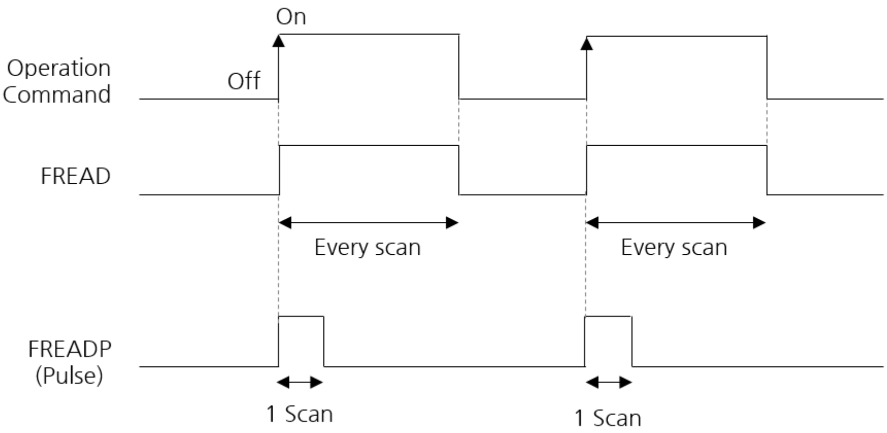
Decides how many data to read from the offset of file data assigned to *n2*.
Range of *n3* is from 0 to 255. If it exceeds the range, an error occurs.
If *n3* is "0", the instruction is not operated.

Assigning *n4*

The device assigned to *n4* is set as "1" when the instruction operates. When the operation ends, the result is stored at this device.
Each bit has meanings as following:

Bit No.	Description
0	Operating instruction (1), Operation completed (0)
1	Operation error
2 - 7	0 (not used)
8 - 15	[Error Code] H00: No Error H01: Capacity of 1 scan process is exceeded (Max. 32 File read/write is possible at 1 scan) H02: [Data] area overflowed H03: [Offset] assignment error

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when:

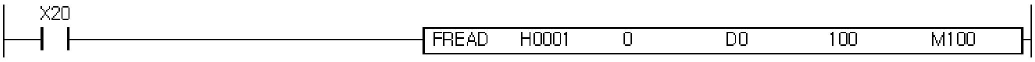
- The address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)
- Access to the special module fails.
- The base and slot number assigned to *n1* is not special module.
- Offset of file data assigned to *n3* exceeds the range of 0 - 255.

Program Example

FREAD, FREADP

This program reads 100 words and stores the data in D0 - D99 when X20 is ON. The data is read starting from file data offset 0 of special module which is equipped at slot 1 of local base. The operation result is stored in M100.

Ladder Diagram (LD)

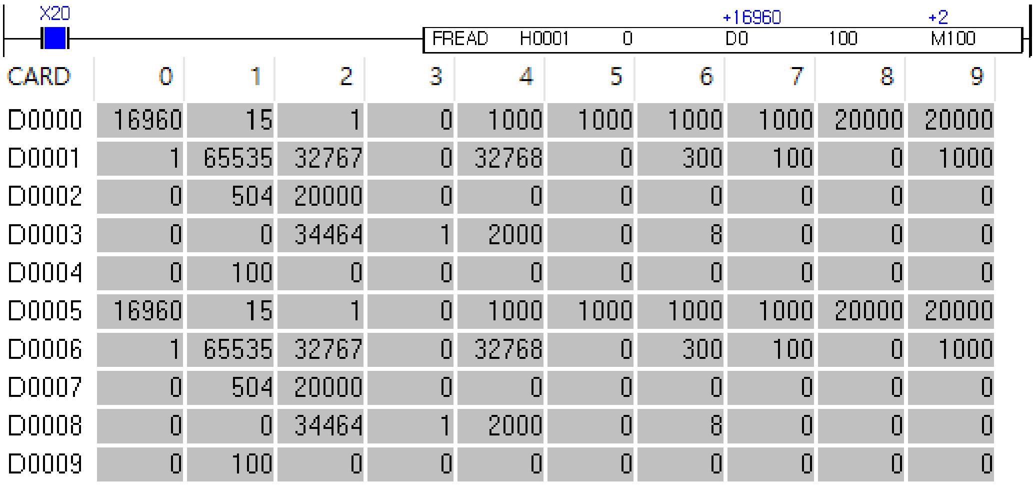


Instruction List (IL)

Instruction	Device				
LD	X20				
FREAD	H0001	0	D0	100	M100

The program operates as following:

- When X20 is ON, the program specifies the location of data. The starting number of data to read is file data offset 0 of special module which is equipped at slot 1 of local base.
- The size of the data to be read is 100 word.
- 100-word data which is read from the specified location is stored in the device starting from D0 to D99.
- The operation result is stored in M100.



CARD	0	1	2	3	4	5	6	7	8	9
D0000	16960	15	1	0	1000	1000	1000	1000	20000	20000
D0001	1	65535	32767	0	32768	0	300	100	0	1000
D0002	0	504	20000	0	0	0	0	0	0	0
D0003	0	0	34464	1	2000	0	8	0	0	0
D0004	0	100	0	0	0	0	0	0	0	0
D0005	16960	15	1	0	1000	1000	1000	1000	20000	20000
D0006	1	65535	32767	0	32768	0	300	100	0	1000
D0007	0	504	20000	0	0	0	0	0	0	0
D0008	0	0	34464	1	2000	0	8	0	0	0
D0009	0	100	0	0	0	0	0	0	0	0

2.18.2 FWRITE, FWRITEP

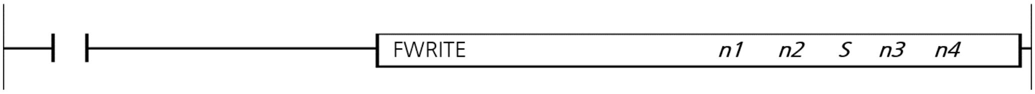
Supported PLC Series

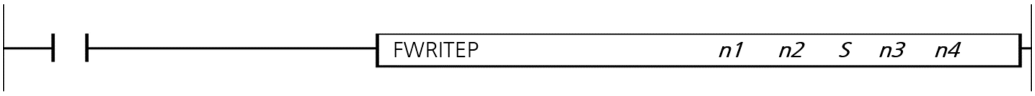
XPnF	CPnF	XPnE	CPnE	XPnB	PLC-S	BP	XPnA/1R	CP3A/B/P/U CP4A~D/U
O	O	O	O	O	O	-	O	O

Function

FWRITE(P) write file data from internal memory of CPU to system memory of a special module.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
FWRITE(P)	*n1*	O	O	O	O	O	O	O	O	O	O	O	O	O	-	O	6	O	-	-
	n2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	-				
	n3	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	n4	O	-	O	O	O	-	-	-	-	-	-	O	O	O	-				

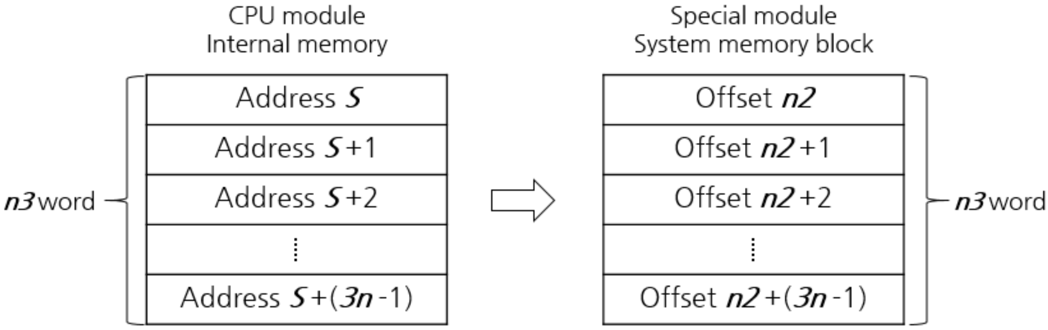




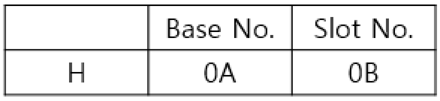
n1	Base and slot number of the special module in hexadecimal
n2	Offset of system memory where file data will be written.
S	Head address of word devices where the file data to be written is stored.
n3	Size of file data to be written (0 - 255).
n4	Head address of word devices to store the operation result. The result is expressed by flag instruction.

FWRITE, FWRITEP

- FWRITE and FWRITEP write file data in *n3* words to the system memory assigned by offset *n2*.
- The location where file data to be written is a special module which is equipped at *n1*.
- The data to be written is stored in the device memory assigned to *S* in *n3* words.
- Result of the operation is stored at word device assigned to *n4*.



Assigning *n1* - [Base/Slot]



	Base No.	Slot No.
H	0A	0B

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal.
Base No.: 2 characters in hexadecimal
Slot No.: 2 characters in hexadecimal

If Base No. is assigned to HFF, it means 'file'.
If Base No. is assigned to HFF, assign PID of the file at Slot No.

Base Number	Slot	S1
Local base	Slot No. 5	H0005 or 5
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 12	H0E0C
16th expansion	Slot No. 10	H100A

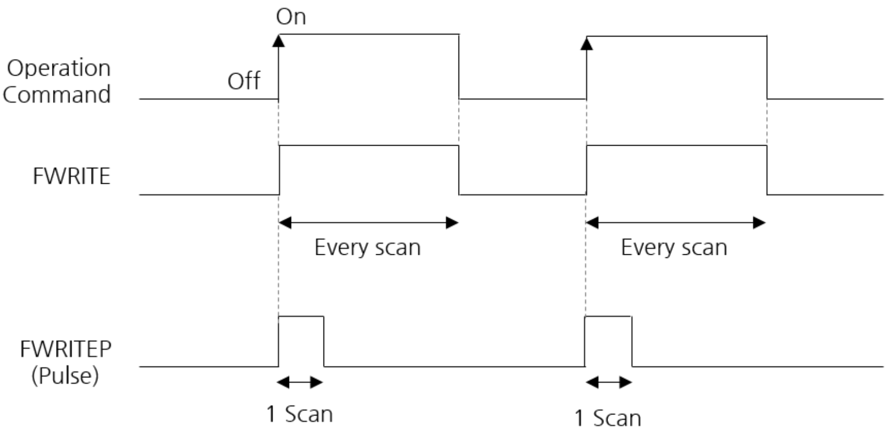
Assigning *n2* - [Offset of file data]
Offset of system memory where file data will be written. In other words, *n2* indicates the location of the data to be written.
(*) Refer to the System Data Memory of Positioning Module Manual.

Assigning *n3* - [Size]
Decides how many data to read from offset of file data assigned by *n2*.
Range of *n3* is from 0 to 255. If it exceeds the range, an error occurs.
If *n3* is "0", the instruction is not operated.

Assigning *n4*
The device assigned to *n4* is set as "1" when the instruction operates. When the operation ends, the result is stored at this device.
Each bit has meanings as following:

Bit No.	Description
0	Operating instruction (1), Operation completed (0)
1	Operation error
2 - 7	0 (not used)
8 - 15	[Error Code] H00: No Error H01: Capacity of 1 scan process is exceeded (Max. 32 File read/write is possible at 1 scan) H02: [Data] area overflowed H03: [Offset] assignment error

Execution Condition



Operation Error

Error Flag (F110)

F110 turns ON for 1 scan when:

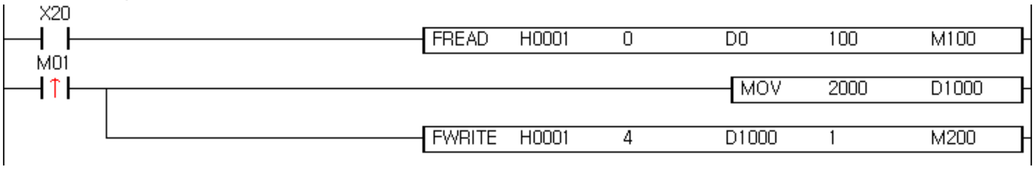
- The address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)
- Access to the special module fails.
- The base and slot number assigned to *n1* is not special module.
- Offset of file data assigned to *n3* exceeds the range of 0 - 255.

Program Example

FWRITE, FWRITEP

This program reads 100-word data from the special module and writes 1-word data which is located in D1000 to the file data offset 4 of special module equipped at slot 1 of local base. The operation result is stored in M200.

Ladder Diagram (LD)

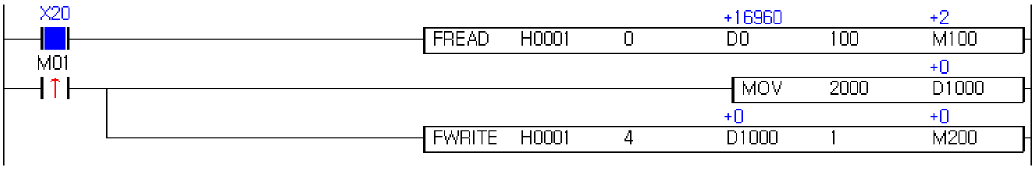


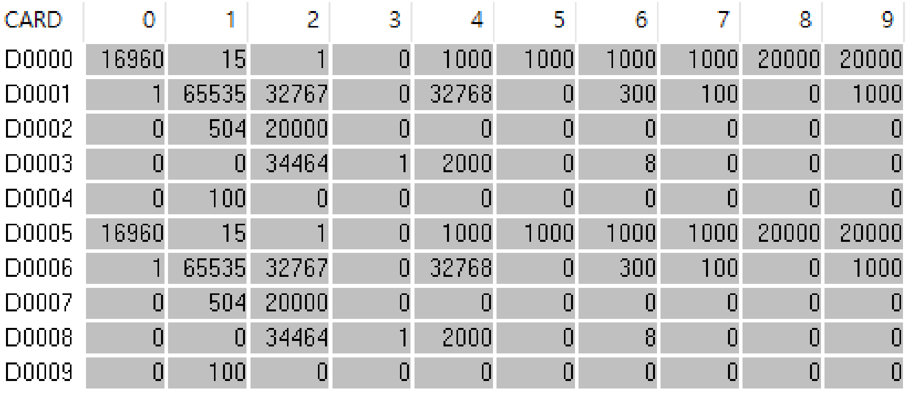
Instruction List (IL)

Instruction	Device				
LD	X20				
FREAD	H0001	0	D0	100	M100
LDP	M01				
MOV	2000		D1000		
FWRITE	H0103	0	D1000	5	M00

The program operates as following:

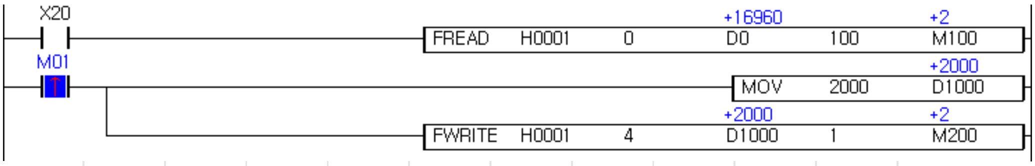
- When X20 is ON, the program specifies the location of data. The starting number of data to read is file data offset 0 of special module which is equipped at slot 1 of local base.
- The size of the data to be read is 100 word.
- 100-word data which is read from the specified location is stored in the device starting from D0 to D99.
- The operation result is stored in M100.

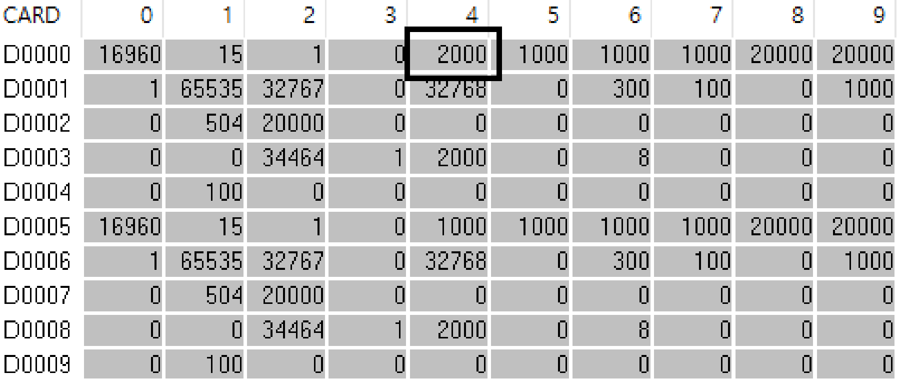




CARD	0	1	2	3	4	5	6	7	8	9
D0000	16960	15	1	0	1000	1000	1000	1000	20000	20000
D0001	1	65535	32767	0	32768	0	300	100	0	1000
D0002	0	504	20000	0	0	0	0	0	0	0
D0003	0	0	34464	1	2000	0	8	0	0	0
D0004	0	100	0	0	0	0	0	0	0	0
D0005	16960	15	1	0	1000	1000	1000	1000	20000	20000
D0006	1	65535	32767	0	32768	0	300	100	0	1000
D0007	0	504	20000	0	0	0	0	0	0	0
D0008	0	0	34464	1	2000	0	8	0	0	0
D0009	0	100	0	0	0	0	0	0	0	0

- When M01 is ON, the program transfers “2000” to D1000.
- FWRITE instruction specifies the location of data to write at special module. The data is located in D1000.
- The size of the data to be written is 1 word.
- 1-word data is written to file data offset 4 of special module which is equipped at slot 1 of local base.
- The operation result is stored in M200.





CARD	0	1	2	3	4	5	6	7	8	9
D0000	16960	15	1	0	2000	1000	1000	1000	20000	20000
D0001	1	65535	32767	0	32768	0	300	100	0	1000
D0002	0	504	20000	0	0	0	0	0	0	0
D0003	0	0	34464	1	2000	0	8	0	0	0
D0004	0	100	0	0	0	0	0	0	0	0
D0005	16960	15	1	0	1000	1000	1000	1000	20000	20000
D0006	1	65535	32767	0	32768	0	300	100	0	1000
D0007	0	504	20000	0	0	0	0	0	0	0
D0008	0	0	34464	1	2000	0	8	0	0	0
D0009	0	100	0	0	0	0	0	0	0	0

2.19 Other Instructions

2.19.1 HSC

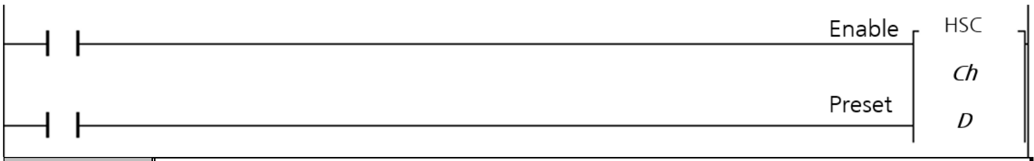
Supported PLC Series

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

Function

HSC instruction counts up with the input pulses from the terminal of the channel assigned to *Ch*. The number of counts are stored in double word device *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
HSC	*Ch*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	`
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				

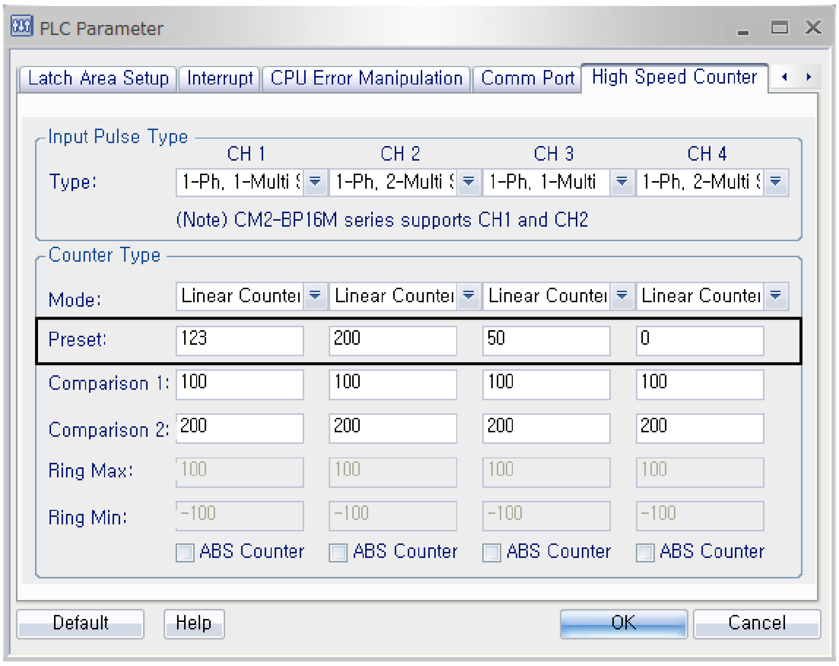


Ch	Constant or word device to specify the channel for the HSC instruction. (1~4)
D	Double word device where the number of counts will be cumulated.

HSC

- When enable signal turns ON, HSC counts up the pulses from terminal of the channel assigned to *Ch*. The number of counts is cumulated in the double word device assigned to *D*.
- HSC instruction only counts up. Count down is not available with this instruction.
- Range of the number of counts is as following:
  -2147483648 ≦ the number of counts ≦ 2147483647
- When enable signal is OFF, the number of counts does not increase in any cases. When enable signal goes from OFF to ON, and if there are input pulses from terminal of the channel assigned to *Ch*, the number stored in device *D* increases from the latest value.
- When preset signal turns ON, internally preset value replaces the value stored in *D*.

- To set the preset value, go to [Tool] - [Parameter] - 'High speed counter' tab.



PLC Parameter

Latch Area Setup | Interrupt | CPU Error Manipulation | Comm Port | High Speed Counter

Input Pulse Type

	CH 1	CH 2	CH 3	CH 4
Type:	1-Ph, 1-Multi !	1-Ph, 2-Multi !	1-Ph, 1-Multi	1-Ph, 2-Multi !

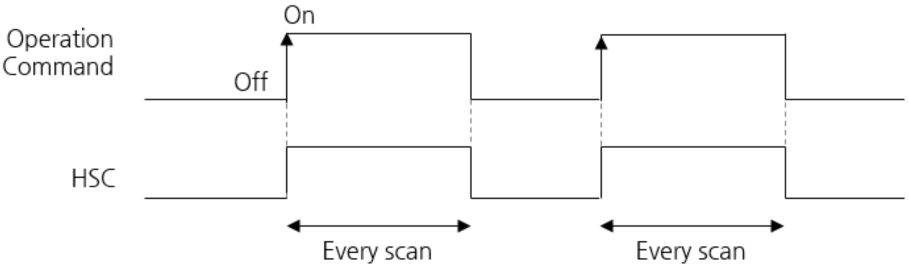
(Note) CM2-BP16M series supports CH1 and CH2

Counter Type

Mode:	Linear Counter	Linear Counter	Linear Counter	Linear Counter
Preset:	123	200	50	0
Comparison 1:	100	100	100	100
Comparison 2:	200	200	200	200
Ring Max:	100	100	100	100
Ring Min:	-100	-100	-100	-100
	□ ABS Counter	□ ABS Counter	□ ABS Counter	□ ABS Counter

Default | Help | OK | Cancel

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)

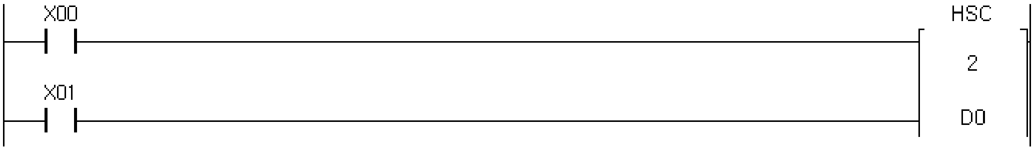
Program Example

HSC

The program operates as following:

- When X00 turns ON, high-speed counter is enabled.
- After the enable signal (X00) turns ON, input pulses from the channel 2 are counted up. The number of counts are stored in D0.
- When the enable signal (X00) turns OFF, the value stored in D0 does not increase.
- When the preset signal (X01) turns ON, preset value is stored in D0.
- When the enable signal (X00) turns ON again, the count starts from the value stored in D0.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
LD	X01	
HSC	2	D0

2.19.2 HSCSW

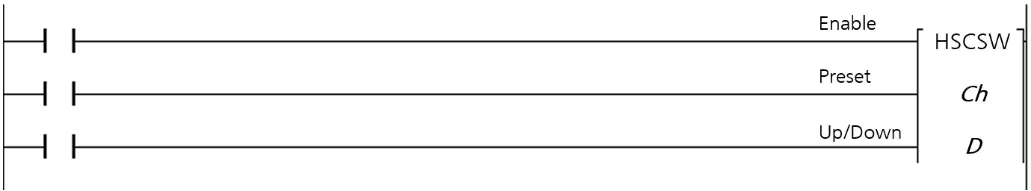
Supported PLC Series

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

Function

HSCSW instruction counts up and down with the input pulses from the terminal of the channel assigned to *Ch*. The number of counts are stored in double word device *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
HSCSW	*Ch*	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O	3	O	-	`
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				

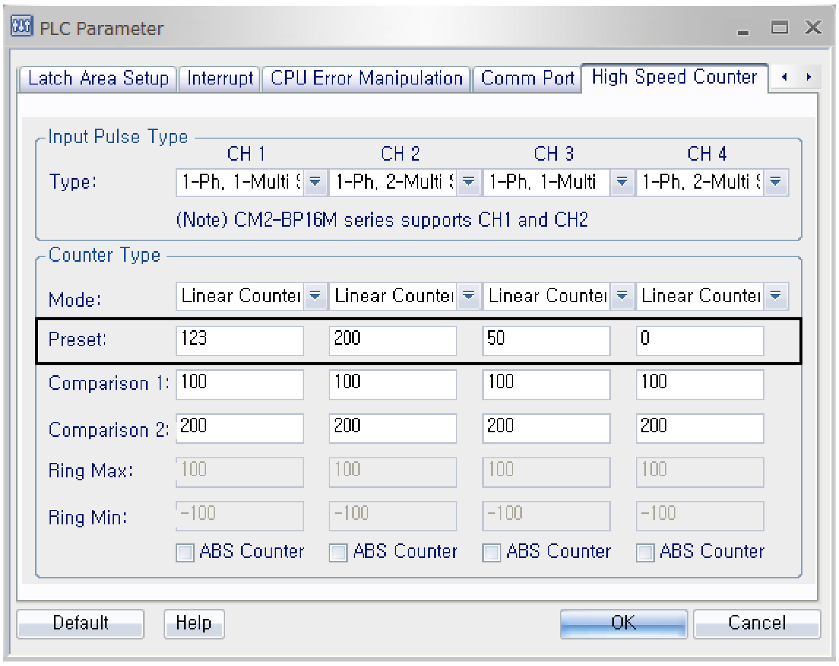


Ch	Constant or word device to specify the channel for the HSCSW instruction. (1~4)
D	Double word device where the number of counts will be cumulated.

HSCSW

- When enable signal turns ON, HSCSW counts up and down with the pulses from terminal of the channel assigned to *Ch*. The number of counts is cumulated in the double word device assigned to *D*.
- Range of the number of counts is as following:
  -2147483648 ≦ the number of counts ≦ 2147483647
- When enable signal is OFF, the number of counts does not increase in any cases. When enable signal goes from OFF to ON, and if there are input pulses from terminal of the channel assigned to *Ch*, the number stored in device *D* increases from the latest value.
- When preset signal turns ON, internally preset value replaces the value stored in *D*.

- To set the preset value, go to [Tool] - [Parameter] - 'High speed counter' tab.



Input Pulse Type

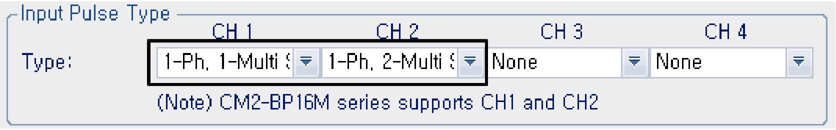
	CH 1	CH 2	CH 3	CH 4
Type:	1-Ph, 1-Multi S	1-Ph, 2-Multi S	1-Ph, 1-Multi	1-Ph, 2-Multi S

(Note) CM2-BP16M series supports CH1 and CH2

Counter Type

	CH 1	CH 2	CH 3	CH 4
Mode:	Linear Counter	Linear Counter	Linear Counter	Linear Counter
Preset:	123	200	50	0
Comparison 1:	100	100	100	100
Comparison 2:	200	200	200	200
Ring Max:	100	100	100	100
Ring Min:	-100	-100	-100	-100
	□ ABS Counter	□ ABS Counter	□ ABS Counter	□ ABS Counter

- The instruction operates only when input of 1 phase 1 multiplication SW or 1 phase 2 multiplication SW is assigned to the channel. Refer to the figure below.



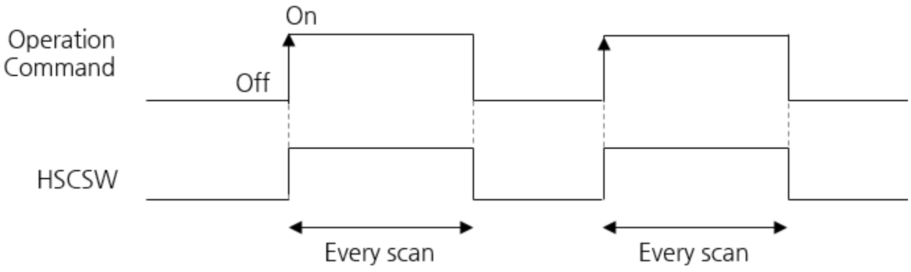
Input Pulse Type

	CH 1	CH 2	CH 3	CH 4
Type:	1-Ph, 1-Multi S	1-Ph, 2-Multi S	None	None

(Note) CM2-BP16M series supports CH1 and CH2

- When Up/Down signal is ON, the instruction counts down. When Up/Down signal is OFF, the instruction counts up.

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)

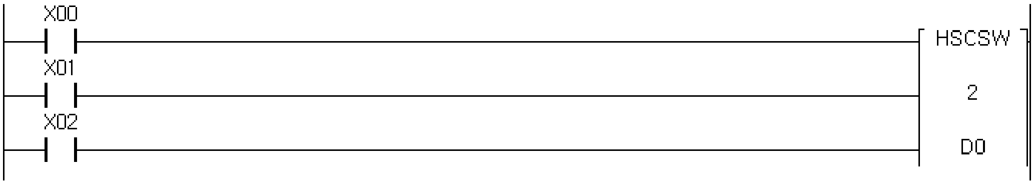
Program Example

HSCSW

The program operates as following:

- When X00 turns ON, high-speed counter is enabled.
- After the enable signal (X00) turns ON, input pulses from the channel 2 are counted up. The number of counts are stored in D0.
- When the enable signal (X00) turns OFF, the value stored in D0 does not increase.
- When the preset signal (X01) turns ON, preset value is stored in D0.
- When the enable signal (X00) turns ON again, the counting starts from the value stored in D0.
- When the Up/Down signal (X02) turns ON, the instruction counts down. When X02 is OFF, the instruction counts up.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
LD	X01	
LD	X02	
HSC	2	D0

2.19.3 ATV

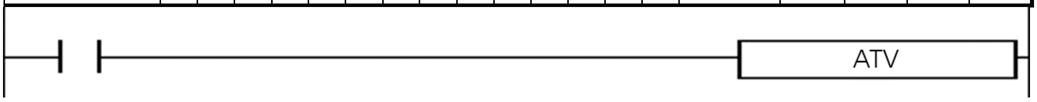
Supported PLC Series

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

Function

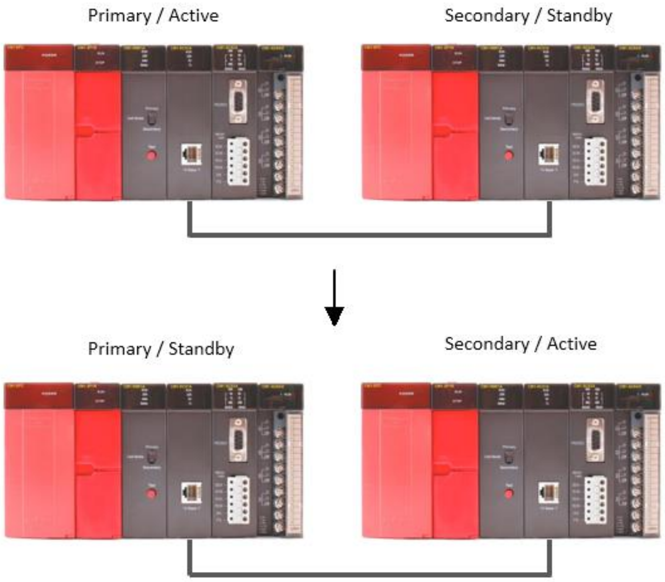
ATV instruction switches the mode (Standby(Backup)/Active) between primary and secondary CPU of the redundant system.

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



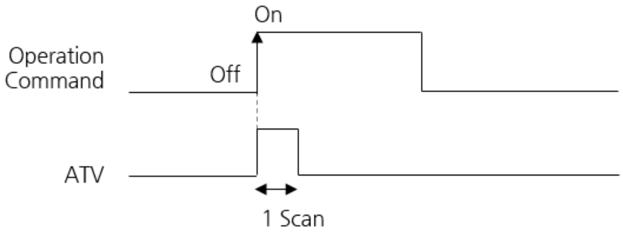
ATV

- ATV instruction switches the mode (Standby(Backup)/Active) between primary and secondary CPU of the redundant system.
- Once the instruction is executed, the activated system will be switched into a standby (backup) mode. Then, the other system which was in standby (backup) mode will be switched into active mode.
- The instruction is operated only once when the execution condition is ON.



(*) For more information redundant system, please refer to n.n Redundant System.

Execution Condition



Operation Error

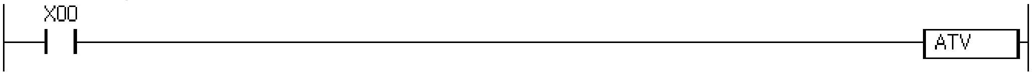
There is no error flag in this instruction.

Program Example

ATV

When X00 turns ON, ATV instruction is executed. Then, the status is switched between active system and standby (backup) system.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device
LD	X00
ATV	

2.19.4 DUTY

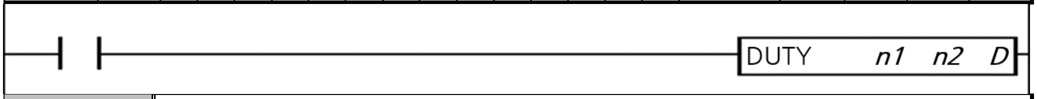
Supported PLC Series

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

DUTY instruction generates the timing pulse by customizing user clock.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
DUTY	*n1*	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O	4	O	-	-
	n2	O	O	O	O	O	O	O	O	-	O	O	O	O	O	O				
	D	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-				

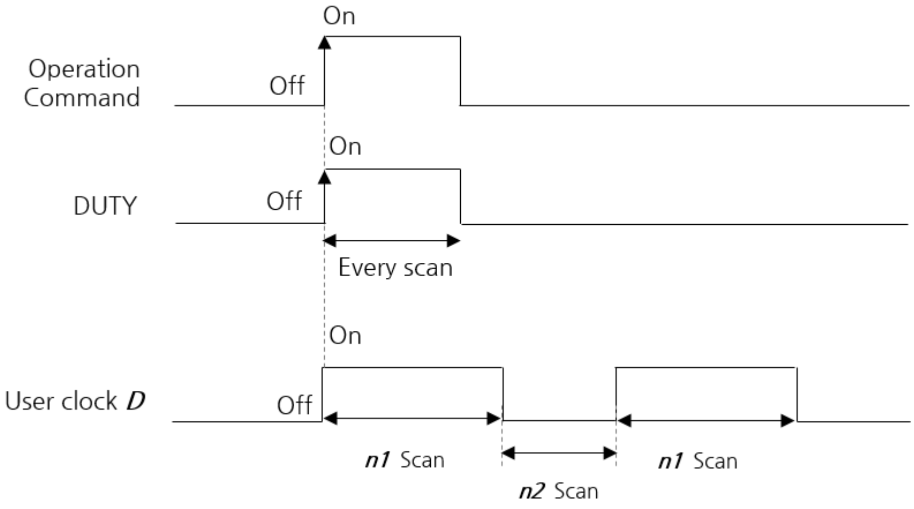


n1	Constant or address of word device to assign the number of scans for user clock ON
n2	Constant or address of word device to assign the number of scans for user clock OFF
D	Address of user clock (F100~F107)

DUTY

- DUTY instruction sets the user clock device assigned to *D*. The duration of the user clock ON and OFF is configured by the value assigned to *n1* and *n2*. The user clock turns ON for *n1* scans and turns OFF for *n2* scans.
- Once the instruction is executed, the assigned user clock turns ON and OFF continuously even if the execution condition is OFF.
- Only positive numbers can be assigned to *n1* and *n2*. The range of value assigned to *n1* and *n2* is as below:
  1 ≦ Value assigned to *n1* and *n2* ≦ 32767
- You can only assign bit devices from F100 to F107 to *D*.
- When the value assigned to *n1* and *n2* changes while the execution condition is ON, the change is immediately applied to the user clock assigned to *D*.

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)

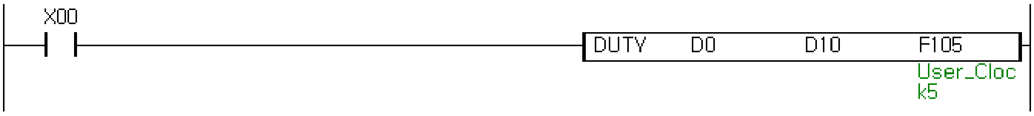
F110 turns ON for 1 scan when the value assigned to *n1* and *n2* is negative.

Program Example

DUTY

When X00 turns ON, DUTY instruction is executed. Then, the user clock F105 turns ON for 300 scans (value assigned to D0) and turns OFF for 500 scans (value assigned to D10).

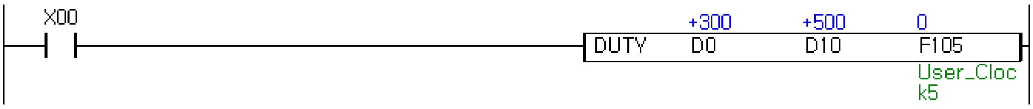
Ladder Diagram (LD)



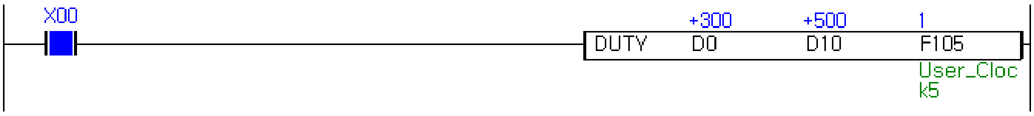
Instruction List (IL)

Instruction	Device		
LD	X00		
DUTY	D0	D10	F105

- "300" is assigned to D0. "500" is assigned to D10.



- When X00 turns ON, F105 stays ON for 300 scans. Then, F105 stays OFF for 500 scans.



2.19.5 IOEXC, IOEXCP

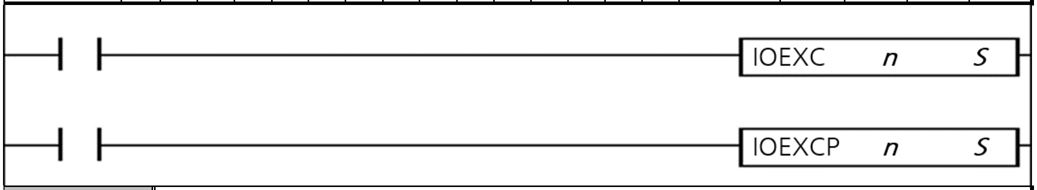
Supported PLC Series

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

Function

IOEXC(P) instructions are used for replacement of I/O module.

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



n	Constant or address of base and slot number where I/O module to be replaced is equipped
S	Constant or address of word device to enable/disable the online I/O module replacement. (0 or 1)

IOEXC, IOEXCP

- IOEXC(P) instructions set the status of online I/O module replacement with the value assigned to *S*. The I/O module to be replaced is equipped at *n*.
- The location of I/O module is specified by the value assigned to *n*.
- Assign "1" to *S* so that you can replace the I/O module specified by *n*. Assign "0" to *S* when you have finished the replacement.
- This instruction is only for the I/O modules and special modules that does not require to download module setups and special programs. (e.g. Analog module)
- Go to [Online] - [Enable/Disable Module...] for online module replacement of special modules.

Assigning *n*

	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]

H: Stands for hexadecimal

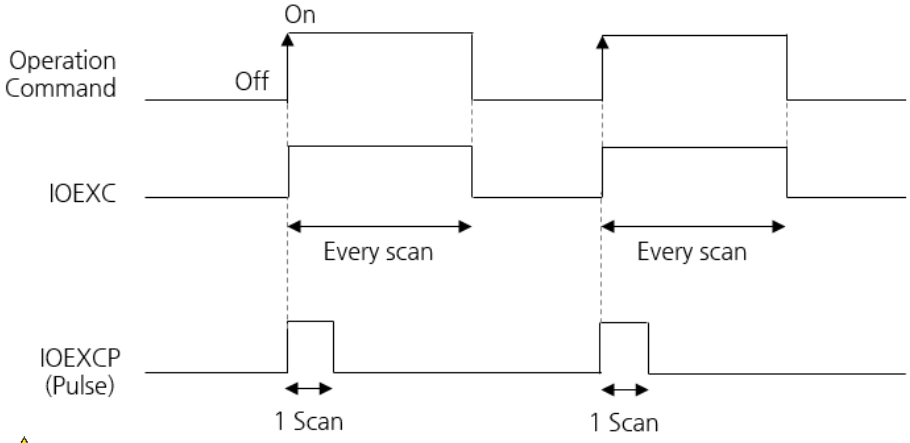
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H1F)

Slot No.: 2 digits in hexadecimal (H00~H0B)

⚠ Base No. and Slot No. could be different according to the PLC CPU and base type.

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

Execution Condition



⚠ Using pulse contact as execution condition of IOEXC instruction or using IOEXCP instruction is recommended.

Operation Error

There is no error flag in this instruction.

System may shut down if I/O module is not equipped properly. Equip the I/O module properly before you assign “0” to *S*.

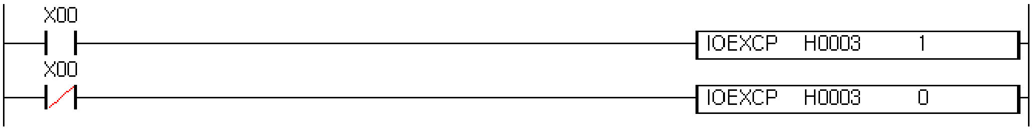
Do not assign “0” to *S* when error LED is ON. Doing so may cause system malfunction. Separate the module from the base and mount again to turn OFF the error LED.

Program Example

IOEXCP

When X00 turns ON, first IOEXCP instruction is executed. Online I/O device replacement feature will be enabled. Then, an I/O module equipped at slot number 3 of local base can be exchanged. When X00 turns OFF, second IOEXCP instruction is executed. Online I/O device replacement feature will be disabled. Then, the replacement procedure is finished.

Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device	
LD	X00	
IOEXCP	H0003	1
LDI	X00	
IOEXCP	H0003	0

- Turn ON X00 before exchanging the I/O module equipped at slot 3 of local base. IOEXCP instruction is executed and online I/O module replacement feature is enabled.
- Then, dismount the I/O module from slot 3 of local base.
- Mount a new I/O module on slot 3 of local base.
- If error LED is ON or I/O module is not mounted properly, dismount and mount the module again.
- If error LED is off and I/O module is mounted properly, turn X00 OFF. Then, online I/O module replacement feature is disabled.

2.20 Load Cell Module Dedicated Instructions

2.20.1 WGBATCH

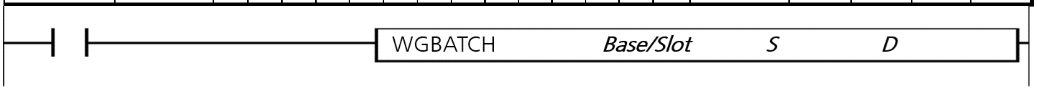
Supported PLC Series

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

Function

WGBATCH instruction executes an embedded batch 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
WGBATCH	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



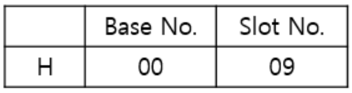
Base/Slot	Constant or address of base and slot number where load cell module is equipped.
S	Constant or address of word device where channel number is stored. (1~4)
D	Address of word device where operation result will be stored.

WGBATCH

- WGBATCH instruction executes a batch program embedded in the load cell module equipped at *Base/Slot*. The batch program is executed on the channel number assigned to *S*. Then, the operation result is stored in word device *D*.
- *Base/Slot* specifies the load cell module with the assigned value.
- Embedded batch program is executed with the channel number assigned to *S*.
- After the execution of instruction, the operation result is stored in word device *D*.
- Only CM1 CPUs with load cell module can use WGBATCH instruction. Supported load cell modules are as below:

Module	Channels
CM1-WG02A	2 Channels
CM1-WG02C	2 Channels
CM1-WG02D	2 Channels
CM1-WG02E	2 Channels
CM1-WG04A	4 Channels

Assigning *Base/Slot*



	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Slot No.: 2 digits in hexadecimal (H00~H0B)

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

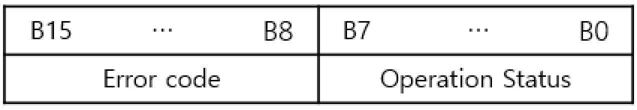
Assigning *S* - [Channel number]

Assign constant or an address of word device where channel number is stored. According to the module, you can assign value in range of 1~4.
The number of channels are as below:

Module	Channels
CM1-WG02A	2 Channels
CM1-WG02C/D/E	2 Channels
CM1-WG04A	4 Channels

Assigning *D* - [Operation result and error code]

Assign an address of word device to store the operation status of WGBATCH. Operation status and error codes are stored in the word device *D*.
Result format is as following:



B15 ··· B8	B7 ··· B0
Error code	Operation Status

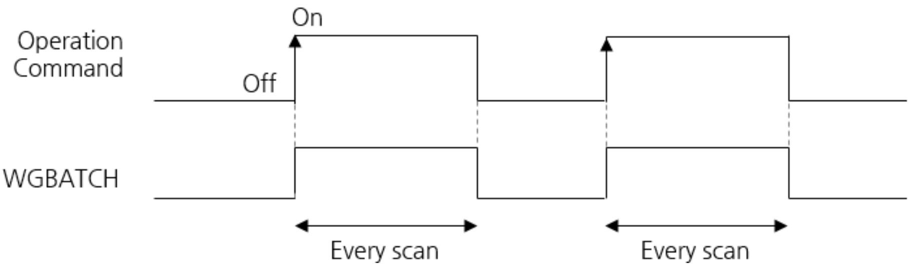
Upper byte: Error code will be stored as listed below.

H00	No error.
H01	Value assigned to *Base/Slot* is invalid.
H01	Module is busy

Lower byte: Operation status will be stored as listed below.

Bit 0	The flag sets when the instruction is in process. The flag resets when the process is completed.
Bit 1	The flag sets when the process is completed.
Bit 2	The flag sets when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

Execution Condition



Operation Error

Error flag (F110)

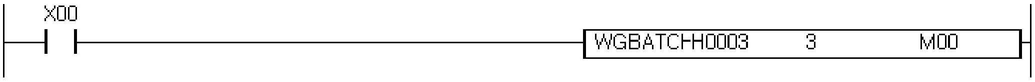
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

WGBATCH

When X00 turns ON, WGBATCH instruction is executed. The load cell module is located on slot number 3 of local base (=H0003). The instruction executes embedded batch program with channel no. 3. When the operation finishes, the result is stored in M00.

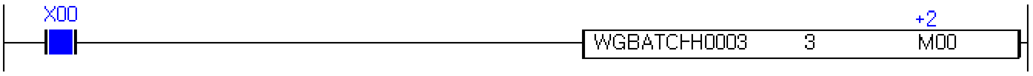
Ladder Diagram (LD)



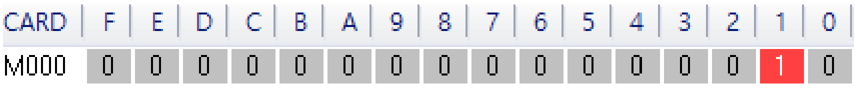
Instruction List (IL)

Instruction	Device		
LD	X00		
WGBATCH	H0003	3	M00

- When X00 turns ON, WGBATCH instruction is executed. The instruction executes a batch program with channel no.3, embedded in the load cell module equipped at slot number 3 of local base (=H0003).
- The operation result is stored in M00 (M00~M0F).



- Since the operation is finished, bit 1 is set. There is no error occurred in this operation.



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

2.20.2 WGSTOP

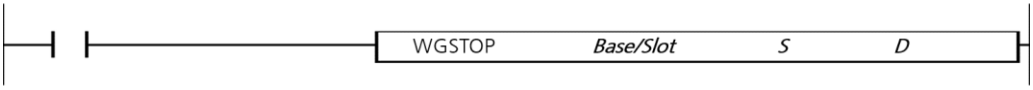
Supported PLC Series

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

Function

WGSTOP instruction forcibly terminates the embedded batch 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
WGSTOP	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



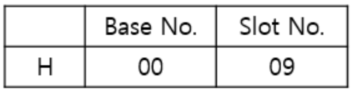
Base/Slot	Constant or address of base and slot number where load cell module is equipped.
S	Constant or address of word device where channel number is stored. (1~4)
D	Address of word device where operation result will be stored.

WGSTOP

- WGSTOP instruction forcibly terminates the batch program embedded in the load cell module equipped at *Base/Slot*. The batch program executed on the channel number assigned to *S* will be terminated. Then, the operation result is stored in word device *D*.
- *Base/Slot* specifies the load cell module with the assigned value.
- Embedded batch program will be terminated on the channel number assigned to *S*.
- After the execution of instruction, the operation result is stored in word device *D*.
- Only CM1 CPUs with load cell module can use WGSTOP instruction. Supported load cell modules are as below:

Module	Channels
CM1-WG02A	2 Channels
CM1-WG02C	2 Channels
CM1-WG02D	2 Channels
CM1-WG02E	2 Channels
CM1-WG04A	4 Channels

Assigning *Base/Slot*



	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Slot No.: 2 digits in hexadecimal (H00~H0B)

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

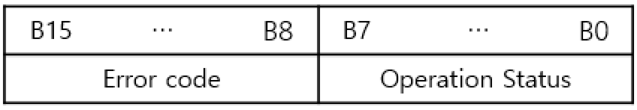
Assigning *S* - [Channel number]

Assign constant or an address of word device where channel number is stored. According to the module, you can assign value in range of 1~4.
The number of channels are as below:

Module	Channels
CM1-WG02A	2 Channels
CM1-WG02C/D/E	2 Channels
CM1-WG04A	4 Channels

Assigning *D* - [Operation result and error code]

Assign an address of word device to store the operation status of WGSTOP. Operation status and error codes are stored in the word device *D*.
Result format is as following:



B15 … B8	B7 … B0
Error code	Operation Status

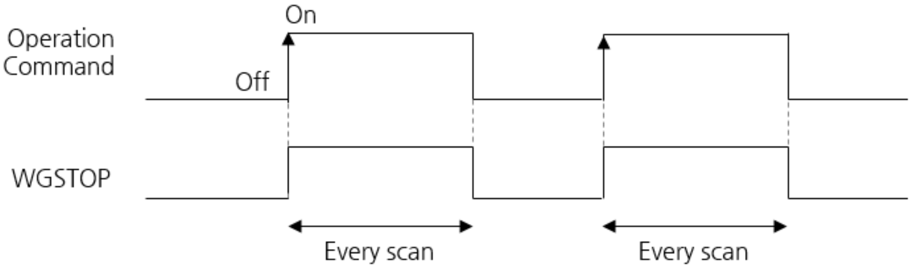
Upper byte: Error code

H00	No error.
H01	Value assigned to *Base/Slot* is invalid.
H01	Module is busy

Lower byte: Operation Status

Bit 0	The flag sets when the instruction is in process. The flag resets when the process is completed.
Bit 1	The flag sets when the process is completed.
Bit 2	The flag sets when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

Execution Condition



Operation Error

Error flag (F110)

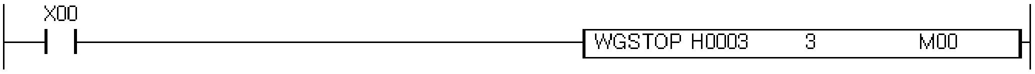
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

WGSTOP

When X00 turns ON, WGSTOP instruction is executed. The load cell module is located on slot number 3 of local base (=H0003). The instruction forcibly terminates embedded batch program of channel no. 3. When the operation finishes, the result is stored in M00.

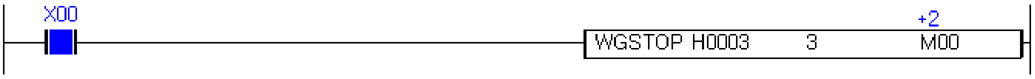
Ladder Diagram (LD)



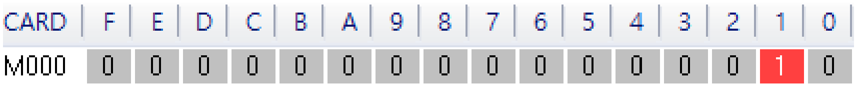
Instruction List (IL)

Instruction	Device		
LD	X00		
WGSTOP	H0003	3	M00

- When X00 turns ON, WGSTOP instruction is executed. The instruction terminates a batch program of channel no.3, embedded in the load cell module equipped at slot number 3 of local base (=H0003).
- The operation result is stored in M00 (M00~M0F).



- Since the operation is finished, bit 1 is set. There is no error occurred in this operation.



CARD	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
M000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

2.21 Positioning Module Dedicated Instructions

2.21.1 PSTRTn (n: 1~2)

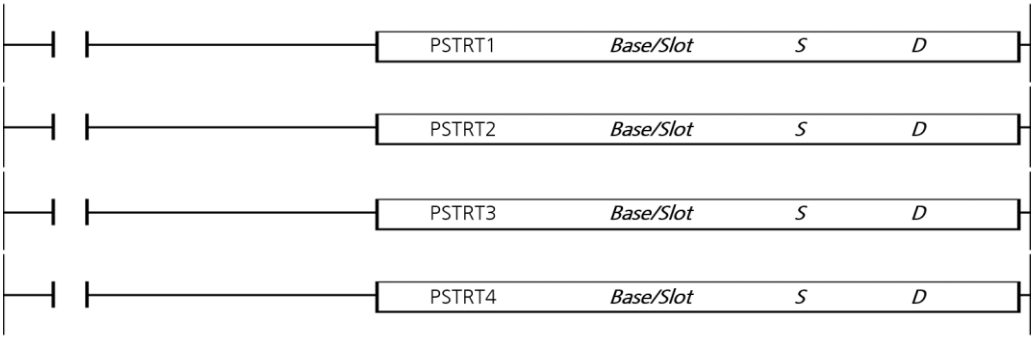
Supported PLC Series

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

Function

PSTRTn instructions control the position of designated axis.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PSTRTn (n:1~2)	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



Base/Slot	Constant or address of base and slot number where positioning module is equipped.
S	Constant or address of word device where control data is stored.
D	Address of word device where operation status will be stored.

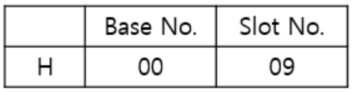
PSTRT1, PSTRT2

- PSTRTn (n: 1~2) instructions start positioning of the designated axis 'n' with the positioning module equipped at *Base/Slot*, according to the control data stored in word device *S*. Then, the operation status is stored in word device *D*.
- *Base/Slot* specifies the positioning module with the assigned value.
- Positioning is operated with the value assigned to word device *S*.
- After the execution of instruction, the status of positioning operation is stored in word device *D*.
- Instructions operates with axes as following:
  PSTRT1: Axis 1
  PSTRT2: Axis 2
- PSTRTn instruction supports only positioning modules as below:

CM1 Series	CM3 Series
PS02A	SP16Mxx
	SP32Mxx

- CM1-PS08N is not supported.

Assigning *Base/Slot*



	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]

H: Stands for hexadecimal

Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)

Slot No.: 2 digits in hexadecimal (H00~H0B)

⚠ In case of PLCS, assign H0000 to *Base/Slot.*

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

Assigning *S* - [Positioning control data]

Assign constant or an address of word device where control data is stored.

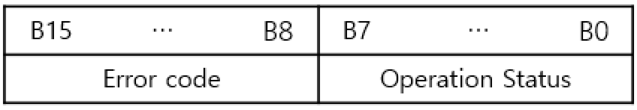
Control data are as below:

Control Data	CM1-PS02A	CM3-SP16M / CM3-SP32M
Positioning data no.	1~600	1~30
Machine OPR[23]	9001	9001
Fast OPR	9002	9002
Change current coordinate value	9003	-
Start multiple axis simultaneously	9004	-

Assigning *D* - [Operation status and error code]

Assign an address of word device to store the operation status of positioning. Operation status and error codes are stored in the word device *D*.

Result format is as following:



B15 … B8	B7 … B0
Error code	Operation Status

Upper byte: Error code

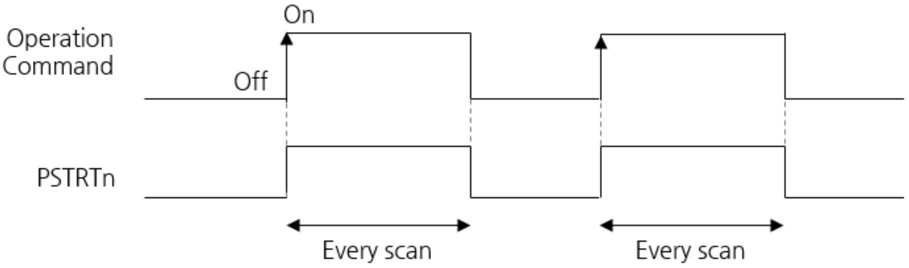
H00	No error
H01	The command is not identified.
H10	The parameter (Positioning control data) is mistyped.
H80	Axis error
HFF	The instruction is not supported.

Lower byte: Operation Status

Bit 0	The flag is set when the instruction is in process.
Bit 1	The flag is set when the process is completed.
Bit 2	The flag is set when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

[23] OPR: OPR stands for Original point return.

Execution Condition



⚠ Using pulse contact as execution condition of PSTRTn instruction is recommended.

Operation Error

Error flag (F110)

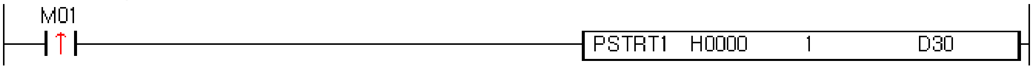
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

PSTRT1

When M01 turns ON, PSTRT1 instruction is executed. The positioning module is located on slot number 0 of local base (=H0000). The instruction uses positioning data no. 1. When the positioning operation finishes, the result is stored in D30.

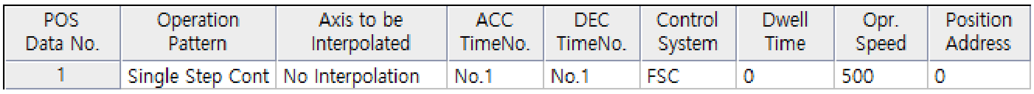
Ladder Diagram (LD)



Instruction List (IL)

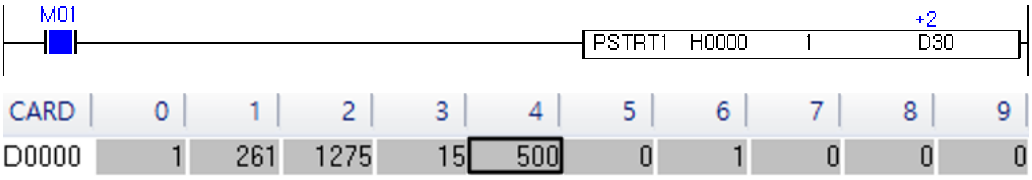
Instruction	Device		
LDP	M01		
PSTRT1	H0000	1	D30

- Positioning data no.1 is configured as below.

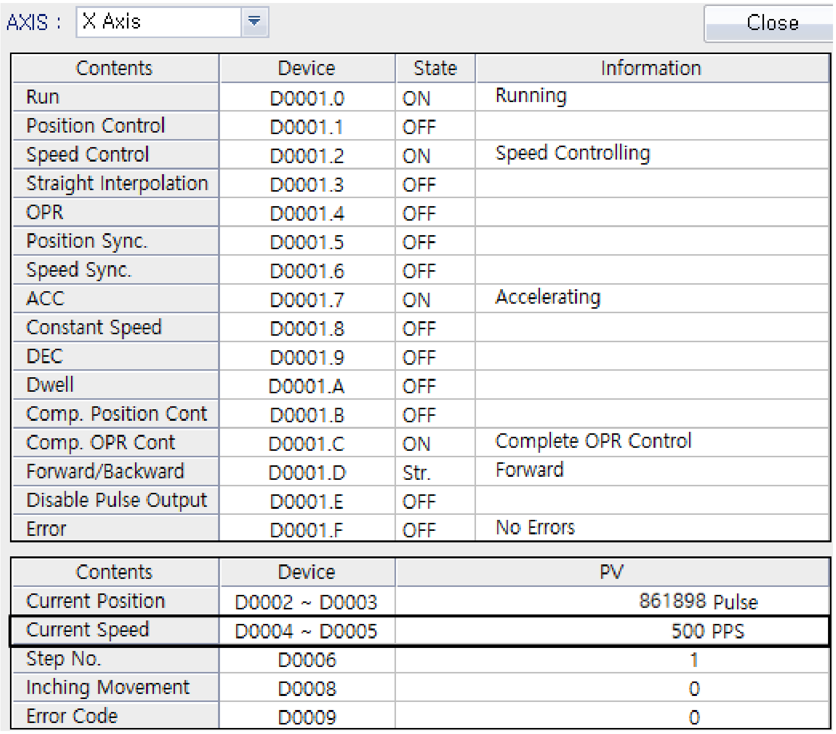


POS Data No.	Operation Pattern	Axis to be Interpolated	ACC TimeNo.	DEC TimeNo.	Control System	Dwell Time	Opr. Speed	Position Address
1	Single Step Cont	No Interpolation	No.1	No.1	FSC	0	500	0

- When M01 turns ON, axis 1 of positioning module uses positioning data no.1. The operation status is stored in D30. Since the operation is finished, bit 1 is set. There is no error occurred in this operation.



CARD	0	1	2	3	4	5	6	7	8	9
D0000	1	261	1275	15	500	0	1	0	0	0



AXIS : X Axis

Contents	Device	State	Information
Run	D0001.0	ON	Running
Position Control	D0001.1	OFF	
Speed Control	D0001.2	ON	Speed Controlling
Straight Interpolation	D0001.3	OFF	
OPR	D0001.4	OFF	
Position Sync.	D0001.5	OFF	
Speed Sync.	D0001.6	OFF	
ACC	D0001.7	ON	Accelerating
Constant Speed	D0001.8	OFF	
DEC	D0001.9	OFF	
Dwell	D0001.A	OFF	
Comp. Position Cont	D0001.B	OFF	
Comp. OPR Cont	D0001.C	ON	Complete OPR Control
Forward/Backward	D0001.D	Str.	Forward
Disable Pulse Output	D0001.E	OFF	
Error	D0001.F	OFF	No Errors

Contents	Device	PV
Current Position	D0002 ~ D0003	861898 Pulse
Current Speed	D0004 ~ D0005	500 PPS
Step No.	D0006	1
Inching Movement	D0008	0
Error Code	D0009	0

2.21.2 PFWRT

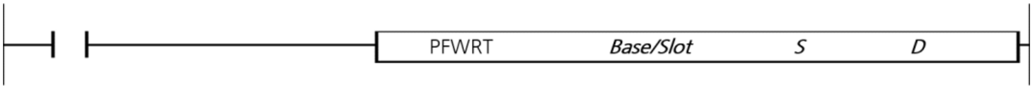
Supported PLC Series

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

Function

PFWRT instruction writes parameters and positioning data to flash memory.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PFWRT	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



Base/Slot	Constant or address of base and slot number where positioning module is equipped.
S	Constant or address of word device where control data type is stored. (*S* =0~2)
D	Address of word device where operation status will be stored.

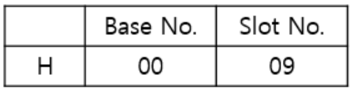
PFWRT

- PFWRT instruction writes parameters and/or positioning data to flash memory, from buffer memory of positioning module equipped at *Base/Slot*. The data to write is decided by the value assigned to word device *S*. The operation status will be stored in the word device *D*.
- *Base/Slot* specifies the positioning module with the assigned value.
- According to the value assigned to word device *S,* different data are written to flash memory.
- After the execution of instruction, the operation status is stored in word device *D*.
- Execution of the instruction is limited up to 25 times after the power ON. To execute the instruction more than 25 times, you have to turn the power OFF and ON again.
- PFWRT instruction supports only positioning modules as below:

CM1 Series	CM3 Series
PS02A	SP16Mxx
	SP32Mxx

- CM1-PS08N is not supported.

Assigning *Base/Slot*



	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Slot No.: 2 digits in hexadecimal (H00~H0B)

⚠ In case of PLCS, assign H0000 to *Base/Slot.*

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

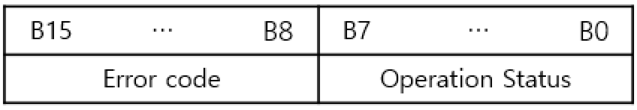
Assigning *S* - [Data to write on flash memory]

Assign constant or an address of word device to select which data to write on flash memory. The range of value assigned to the *S* is from 0 to 2.
Data to write on the flash memory is decided as below:

S=0	Stores both parameters and positioning data to flash memory.
S=1	Stores only parameters to flash memory.
S=2	Stores only positioning data to flash memory.

Assigning *D* - [Operation status and error code]

Assign an address of word device to store the operation status of positioning. Operation status and error codes are stored in the word device *D*.
Result format is as following:



B15 … B8	B7 … B0
Error code	Operation Status

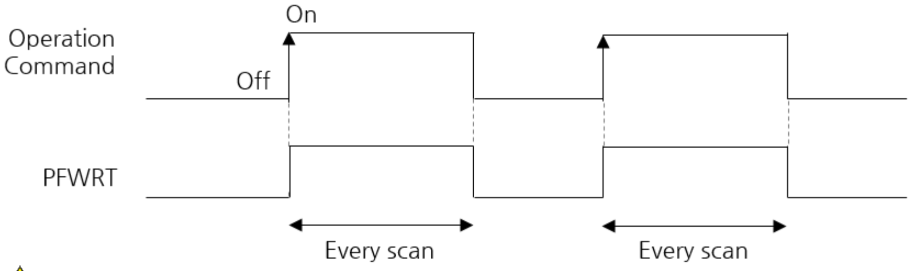
Upper byte: Error code

H00	No error
H01	Writing parameters and positioning data to the flash memory has failed.
H02	Writing positioning data to the flash memory has failed.
H03	Writing parameters to the flash memory has failed.
H04	PFWRT instruction is operated more than 25 times after power ON.
H10	The parameter is mistyped.
HFF	The instruction is not supported.

Lower byte: Operation Status

Bit 0	The flag is set when the instruction is in process.
Bit 1	The flag is set when the process is completed.
Bit 2	The flag is set when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

Execution Condition



⚠ Using pulse contact as execution condition of PFWRT instruction is recommended.

Operation Error

Error flag (F110)

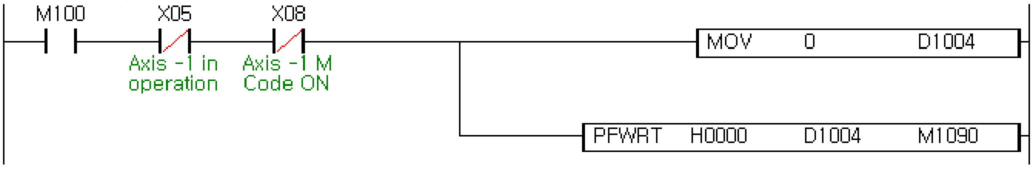
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

PFWRT

When M100 turns ON, PFWRT instruction is executed. The positioning module is located on slot number 0 of local base (=H0000). The parameters and positioning data stored in buffer memory will be written to flash memory. Then, the status of operation is stored in M1090.

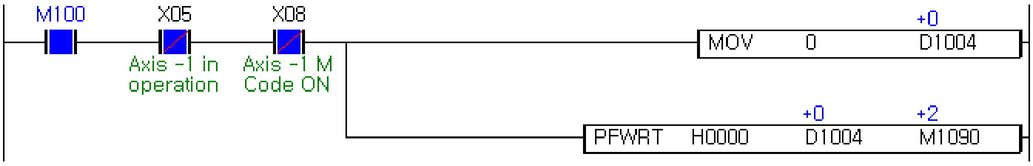
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LD	M100		
ANDI	X05		
ANDI	X08		
MOV	0	D1004	
PFWRT	H0000	D1004	M1090

- When M100, X05 and X08 are turned ON, 0 is stored in D1004 and PFWRT instruction is executed.



- Since “0” is stored in D1004, both parameters and positioning data are written to the flash memory.

- In word device M1090, bit 2 is turned ON. It means that the operation of instruction is completed.

2.21.3 POSCTRL

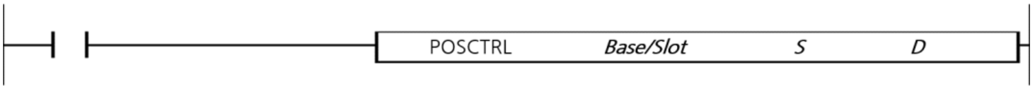
Supported PLC Series

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

Function

POSCTRL instruction operates positioning with parameters assigned to the word 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	Zero	Carry
POSCTRL	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



Base/Slot	Constant or address of base and slot number where positioning module is equipped.
S	Head address of word devices where control data is stored.
D	Address of word device where operation status will be stored.

POSCTRL

- POSCTRL instruction operates positioning with positioning module equipped at *Base/Slot*. The instruction uses values (parameters) assigned to word devices starting from *S*. The operation status will be stored in the word device *D*.
- *Base/Slot* specifies the positioning module with the assigned value.
- The positioning is operated according to the values (parameters) assigned to word devices starting from *S*. At device *S*, an axis to be operated is designated. Control code is assigned to *S*+1. According to the value assigned to *S*+1, different parameters are assigned to the word devices starting from *S*+2.
- After the execution of instruction, the operation status is stored in word device *D*.
- PFWRT instruction supports only positioning modules as below:

CM1 Series	CM3 Series
PS02A	SP16Mxx
PS08N	SP32Mxx

Assigning *Base/Slot*

	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Slot No.: 2 digits in hexadecimal (H00~H0B)

⚠ In case of PLCS, assign H0000 to *Base/Slot*.

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

Assigning *S* - [Control code and data]

Assign head address of word devices where control data for positioning are stored. According to the value assigned to *S*+1, different control data are needed to be configured as shown below.

- CM1 CPU (PS02A, PS08N)

When the value assigned to *S*+1 is 1, 2, 6, 9, the instruction occupies 4 words starting from *S*.

S	Assign an axis for positioning. (1: Axis X 2: Axis Y).
S+1	Assign a control code. 1: Change current coordinate 2: Change operation speed 6: Change target coordinate 9: Error Clear (Input "1" at *S*+2 to clear the error of the corresponding axis)
S+2	Assign a value according to the control code. (Lower word)
S+3	Assign a value according to the control code. (Upper word)

When the value assigned to *S*+1 is 10, the instruction occupies 12 words starting from *S*.

S	Assign an axis for positioning. (1: Axis X 2: Axis Y).
S+1	Assign a control code. 10: Indirect positioning data
S+2	Positioning data (e.g. Control type, interpolation axes etc. Refer to the table below.)
S+3	Assign M code[24]. (0~65535)
S+4	Assign dwell time. (0~65535ms)
S+5	Reserved
S+6	Operation speed (Lower word)
S+7	Operation speed (Upper word)
S+8	Destination address or movement amount (Lower word)
S+9	Destination address or movement amount (Upper word)
S+10	Circular interpolation address (Lower word)
S+11	Circular interpolation address (Upper word)

⚠ Positioning data no. 600 is for internal use. It means that the value you assigned to the positioning data no. 600 is not reatained.

[24] M Code: M Code is an additional function for sub operations. There are three modes; After mode, with mode and none. When M code is ON, following positioning operation is not executed. M code can be turns OFF by PLC program.

- Positioning data (*S*+2)

Bit	Bit 0~1	Bit 2~3	Bit 4~5	Bit 6~7	Bit 8~15
Item	Control type	Interpolation axes	Acceleration No.	Deceleration No.	Control Instructions

- CM3 CPU (SP16Mxx, SP32Mxx)

When the value assigned to *S*+1 is 1, 2, 3, 6, the instruction occupies 4 words starting from *S*.

S	Assign an axis for positioning. (1: Axis X 2: Axis Y).
S+1	Assign a control code. 1: Change current coordinate 2: Change operation speed 3: Inching operation 6: Change target coordinate
S+2	Assign a value according to the control code. (Lower word)
S+3	Assign a value according to the control code. (Upper word)

When the value assigned to *S*+1 is 7, the instruction occupies 4 words starting from *S*.

S	Assign an axis for positioning. (1: Axis X 2: Axis Y).
S+1	Assign a control code. 7: Change parameters (Word)
S+2	Parameter offset (*) Refer to the table 'Positioning data.
S+3	Assign a value according to the parameter offset. (Word)

When the value assigned to *S*+1 is 8, the instruction occupies 5 words starting from *S*.

S	Assign an axis for positioning. (1: Axis X 2: Axis Y).
S+1	Assign a control code. 8: Change parameters (Double word)
S+2	Parameter offset (*) Refer to the table 'Positioning data.
S+3	Assign a value according to the parameter offset. (Lower word)
S+4	Assign a value according to the parameter offset. (Upper word)

When the value assigned to *S*+1 is 10, the instruction occupies 10 words starting from *S*.

S	Assign an axis for positioning. (1: Axis X 2: Axis Y).
S+1	Assign a control code. 10: Indirect positioning data
S+2	Positioning data (e.g. Control type, interpolation axes etc. Refer to the table 'Positioning data.)
S+3	Reserved
S+4	Assign dwell time. (0~65535ms)
S+5	Reserved
S+6	Operation speed (Lower word)
S+7	Operation speed (Upper word)
S+8	Target coordinate (Lower word)
S+9	Target coordinate (Upper word)

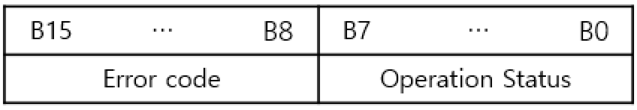
- Positioning data

Offset	Item	Description	
0	Positioning data	Bit 0~1: Control type Bit 2~3: Interpolation axes Bit 4~5: Acceleration No. Bit 6~7: Deceleration No. Bit 8~15: Control Instructions	
1	Reserved	-	
2	Dwell time	0~65535ms	
3	Reserved	-	
4	Operation speed	Lower word	1~100000 pps[25]
5		Upper word	
6	Target coordinate	Lower word	-2147483648~2147483647 pulse
7		Upper word	

Assigning *D* - [Operation status and error code]

Assign an address of word device to store the operation status of positioning. Operation status and error codes are stored in the word device *D*.

Result format is as following:



B15 ··· B8	B7 ··· B0
Error code	Operation Status

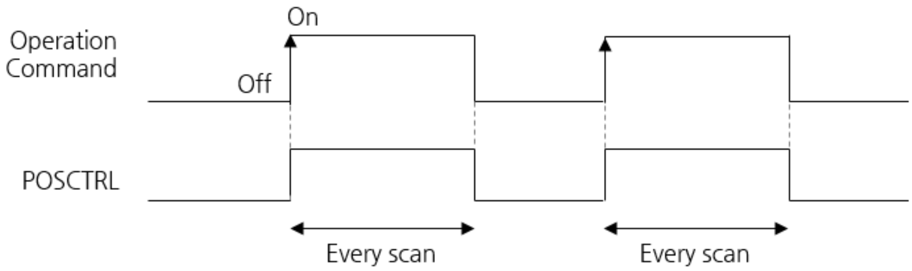
Upper byte: Error code

H00	No error
H01	The command is not identified.
H10	The parameter is mistyped.
HFF	The instruction is not supported.

Lower byte: Operation Status

Bit 0	The flag is set when the instruction is in process.
Bit 1	The flag is set when the process is completed.
Bit 2	The flag is set when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

Execution Condition



⚠ Using pulse contact as execution condition of PINIT instruction is recommended.

[25] pps: Pulse per second

Operation Error

Error flag (F110)

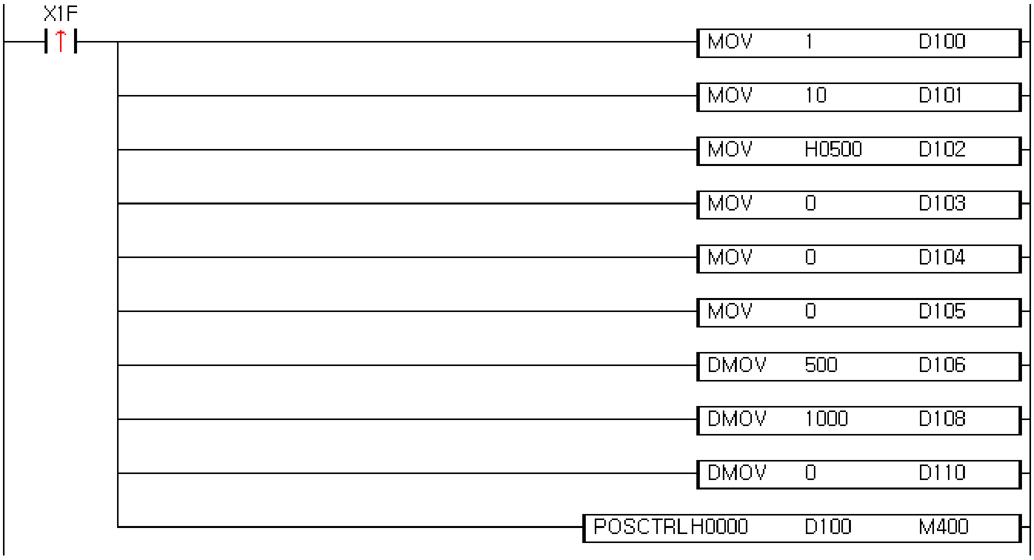
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

POSCTRL

When X1F turns ON, MOV and DMOV instructions are executed. The instructions configure the positioning data starting from D100 (D100~D111). Then, POSCTRL instruction is executed. POSCTRL instruction operates indirect positioning with positioning module equipped at slot number 0 of local base (=H0000). The POSCTRL instruction uses data and parameters stored in 12 word devices starting from D100. Then, the operation status is stored in M400.

Ladder Diagram (LD)



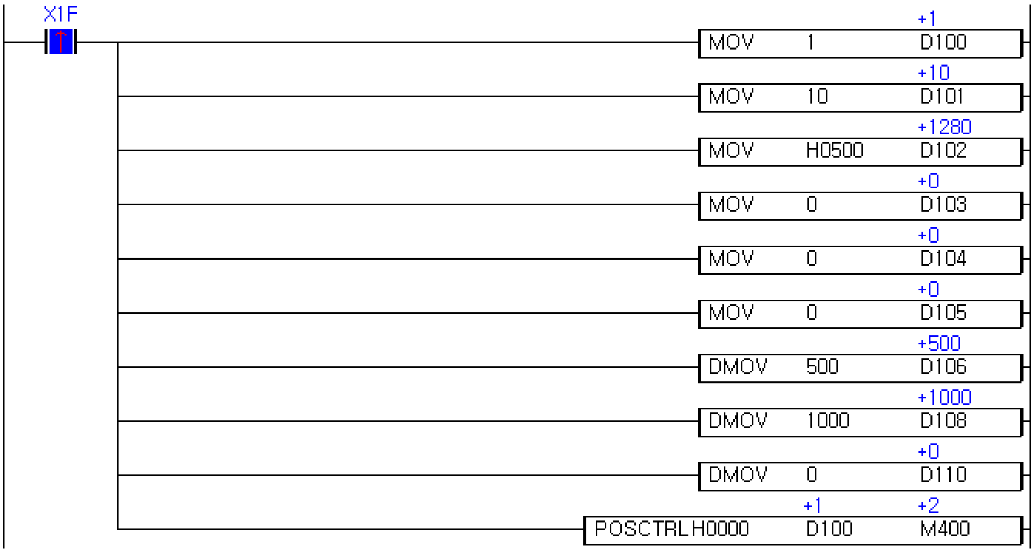
Instruction List (IL)

Instruction	Device		
LDP	X1F		
MOV	1	D100	
MOV	10	D101	
MOV	H0500	D102	
MOV	0	D103	
MOV	0	D104	
MOV	0	D105	
DMOV	500	D106	
DMOV	1000	D108	
DMOV	0	D110	
POSCTRL	H0000	D100	M400

- When X1F turns ON, MOV and DMOV instructions are executed. Values assigned to D100~D111 are positioning data.

- When POSCTRL instruction is executed, the positioning is operated as following:
  - The operation uses axis X. (Value assigned to D100 is “1”)
  - Control code is “10”. It means the indirect positioning. (Value assigned to D101 is “10”)
  - Positioning data is “H0500”. (Value assigned to D102 is “H0500”)
    It means that control type is ‘Individual control’, interpolation axes are not assigned, acceleration/deceleration no.1 is selected and control instruction is ‘INC’.
  - M code and dwell time is “0”. (Values assigned to D102 and D103 are “0”)
  - Operation speed is 500pps. (Value assigned to double word device D106 (D106~D107) is “500”)
  - Target coordinate is 1000. (Value assigned to double word device D108 (D108~D109) is “1000”)
  - Circular interpolation coordinate is 0. (Value assigned to double word device D110 (D110~D111)
    is “0”)

- The operation status is stored in M400. Since the operation is finished, bit 1 is set. There is no error occurred in this operation.



```
X1F
                              +1
MOV      1                    D100
                              +10
MOV      10                   D101
                              +1280
MOV      H0500                D102
                              +0
MOV      0                    D103
                              +0
MOV      0                    D104
                              +0
MOV      0                    D105
                              +500
DMOV     500                  D106
                              +1000
DMOV     1000                 D108
                              +0
DMOV     0                    D110
                     +1       +2
POSCTRL  H0000       D100     M400
```

2.21.4 PINIT

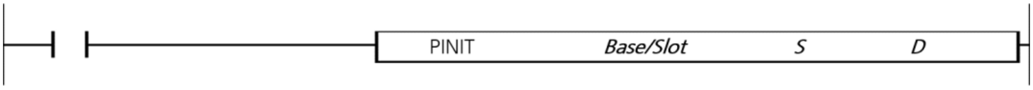
Supported PLC Series

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

Function

PINIT instruction initializes parameters and positioning data stored in flash memory.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
PINIT	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				



Base/Slot	Constant or address of base and slot number where positioning module is equipped.
S	Constant or address of word device where control data type is stored. (*S* =0~2)
D	Address of word device where operation status will be stored.

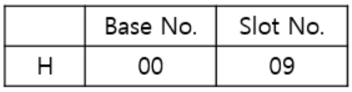
PINIT

- PINIT instruction initializes parameters and/or positioning data stored in flash memory of positioning module equipped at *Base/Slot*. The data to initialize is decided by the value assigned to word device *S*. The operation status will be stored in the word device *D*.
- *Base/Slot* specifies the positioning module with the assigned value.
- According to the value assigned to word device *S,* different data stored in flash memory are initialized. It means that all setting values are restored into the default settings (Factory-set).
- After the execution of instruction, the operation status is stored in word device *D*.
- PINIT instruction supports only positioning modules as below:

CM1 Series	CM3 Series
PS02A	SP16Mxx
	SP32Mxx

- CM1-PS08N is not supported.

Assigning *Base/Slot*



	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Slot No.: 2 digits in hexadecimal (H00~H0B)

⚠ In case of PLCS, assign H0000 to ***Base/Slot***.

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

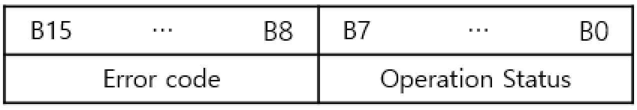
Assigning *S* - [Data to initialize from flash memory]

Assign constant or an address of word device to select which data to initialize from flash memory. The range of value assigned to the *S* is from 0 to 2.
Data to write on the flash memory is as below:

S=0	Initializes both parameters and positioning data stored in flash memory.
S=1	Initializes only parameters stored in flash memory.
S=2	Initializes only positioning data stored in flash memory.

Assigning *D* - [Operation status and error code]

Assign an address of word device to store the operation status of positioning. Operation status and error codes are stored in the word device *D*.
Result format is as following:



B15 … B8	B7 … B0
Error code	Operation Status

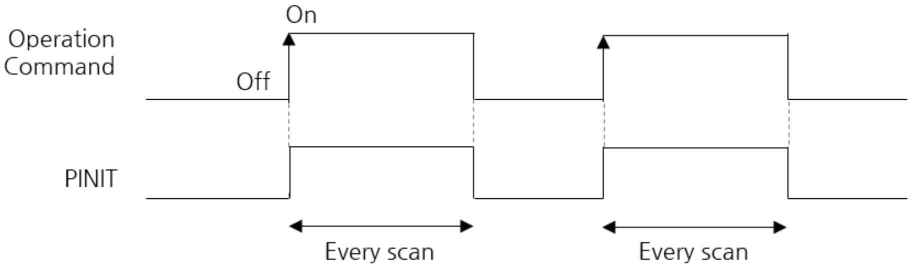
Upper byte: Error code

H00	No error
H10	The parameter is mistyped.
HFF	The instruction is not supported.

Lower byte: Operation Status

Bit 0	The flag is set when the instruction is in process.
Bit 1	The flag is set when the process is completed.
Bit 2	The flag is set when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

Execution Condition



⚠ Using pulse contact as execution condition of PINIT instruction is recommended.

Operation Error

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

PINIT

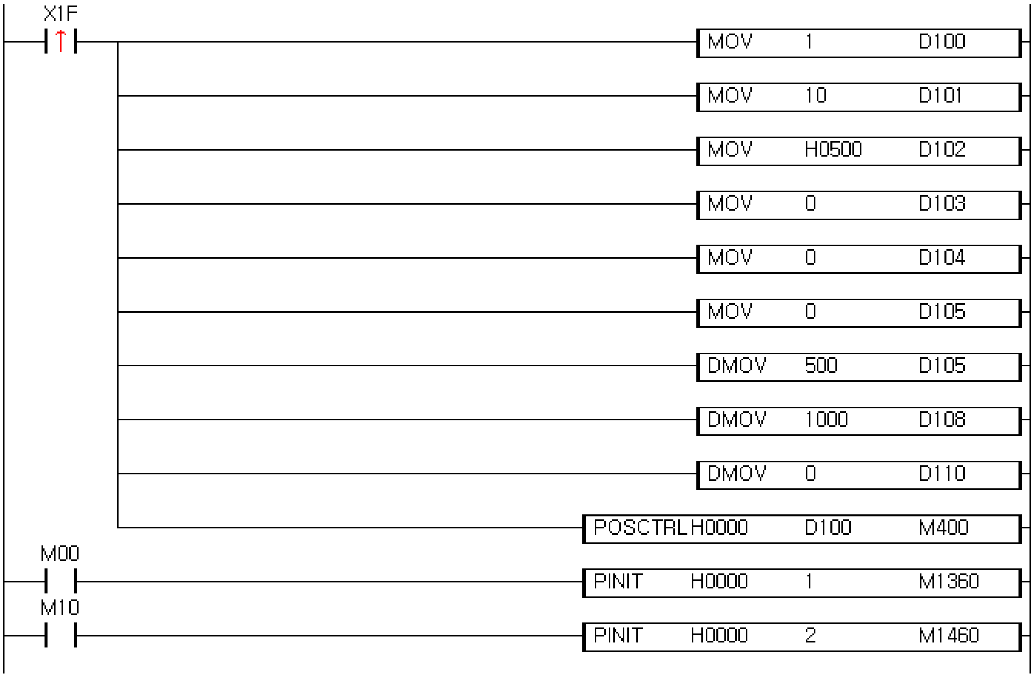
When X1F turns ON, MOV, DMOV and POSCTRL instructions are executed. Control data are stored in word devices starting from D100 (D100~D111). The stored data will be used for the positioning.

When M00 turns ON, PINIT instruction is executed. This PINIT instruction initializes the parameters stored in flash memory of positioning module which is located on slot number 0 of local base (=H0000). Then, the status of operation is stored in M1360.

When M10 turns ON, PINIT instruction is executed. This PINIT instruction initializes the positioning data stored in flash memory of positioning module which is located on slot number 0 of local base (=H0000). Then, the status of operation is stored in M1460.

(*) For the execution of POSCTRL instruction, refer to the Program Example of 2.21.3 POSCTRL.

Ladder Diagram (LD)



Instruction List (IL)

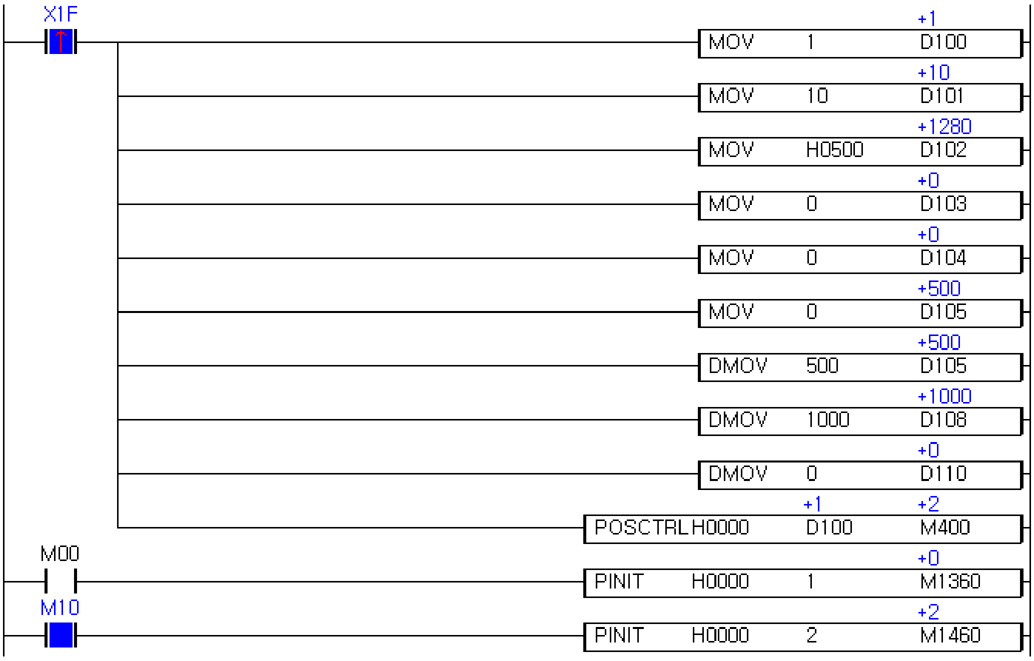
Instruction	Device		
LDP	X1F		
MOV	1	D100	
MOV	10	D101	
MOV	H0500	D102	
MOV	0	D103	
MOV	0	D104	
MOV	0	D105	
DMOV	500	D106	
DMOV	1000	D108	
DMOV	0	D110	
POSCTRL	H0000	D100	M400
LD	M00		
PINIT	H0000	1	M1360
LD	M10		
PINIT	H0000	2	M1460

- When X0F turns ON, control data assigned to word and double word devices starting from D100 (D100~D111) will be used for the positioning.
(*) Refer to the Program Example of 2.21.3 POSCTRL.

- When M00 turns ON, parameters of positioning module equipped at slot 0 of local base will be initialized.

- When M10 turns ON, positioning data of positioning module equipped at slot 0 of local base will be initialized.

- Since bit 1 of M1460 is turned ON, initialization of positioning data is completed.



```
X1F                                        +1
─┤↑├──┬────────────────────── MOV     1       D100
      │                                    +10
      ├────────────────────── MOV     10      D101
      │                                    +1280
      ├────────────────────── MOV     H0500   D102
      │                                    +0
      ├────────────────────── MOV     0       D103
      │                                    +0
      ├────────────────────── MOV     0       D104
      │                                    +500
      ├────────────────────── MOV     0       D105
      │                                    +500
      ├────────────────────── DMOV    500     D105
      │                                    +1000
      ├────────────────────── DMOV    1000    D108
      │                                    +0
      ├────────────────────── DMOV    0       D110
      │                             +1     +2
      └──────────── POSCTRL H0000   D100    M400
M00                                        +0
─┤ ├─────────────── PINIT   H0000   1       M1360
M10                                        +2
─┤ ├─────────────── PINIT   H0000   2       M1460
```

2.21.5 TEACHn (n: 1~2)

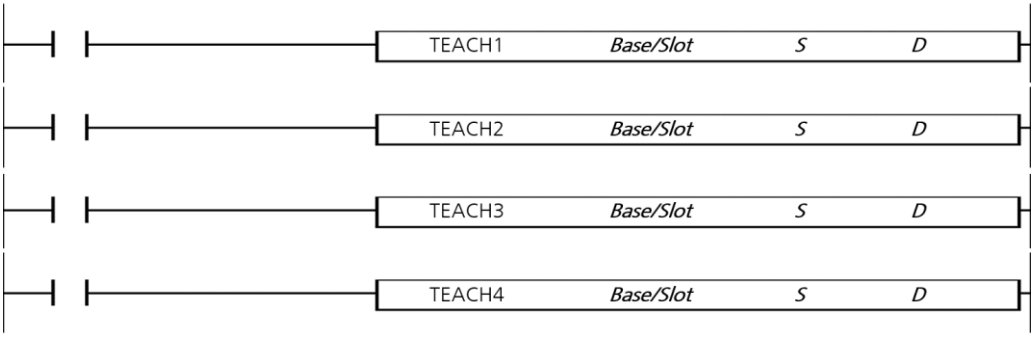
Supported PLC Series

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

Function

TEACHn instructions teach a certain positioning data which is already programmed.

Instruction		Valid Device Type															Steps	Flag		
		M	X	Y	K	L	F	T	C	S	Z	R	Q	D	@D	Constant		Error	Zero	Carry
TEACHn (n: 1~2)	*Base/Slot*	O	O	O	O	O	O	O	O	-	O	-	O	O	O	O	4	O	-	-
	S	O	O	O	O	O	-	-	-	-	O	-	O	O	O	O				
	D	O	-	O	O	O	-	-	-	-	O	-	O	O	O	-				


TEACH1 *Base/Slot* *S* *D*
TEACH2 *Base/Slot* *S* *D*
TEACH3 *Base/Slot* *S* *D*
TEACH4 *Base/Slot* *S* *D*

Base/Slot	Constant or address of base and slot number where positioning module is equipped.
S	Head address of word devices where control data is stored.
D	Address of word device where operation status will be stored.

TEACH1, TEACH2

- TEACHn (n: 1~2) instructions teach a certain positioning data assigned to axis 'n'. The positioning module to operate the instruction is equipped at *Base/Slot*. According to the values assigned to word devices starting from *S*, the positioning data is specified and taught to the module. Then, the operation status is stored in word device *D*.
- *Base/Slot* specifies the positioning module with the assigned value.
- Teaching of positioning data is operated with the value assigned to word devices starting from *S*.
- After the execution of instruction, the operation status is stored in word device *D*.
- Instructions operates with axes as following:
  TEACH1: Axis 1
  TEACH2: Axis 2
- TEACHn instruction supports only positioning modules as below:

CM1 Series	CM3 Series
PS02A	SP16Mxx
	SP32Mxx

- CM1-PS08N is not supported.

Assigning *Base/Slot*

	Base No.	Slot No.
H	00	09

H + [Base No.] + [Slot No.]
H: Stands for hexadecimal
Base No.: 2 digits in hexadecimal (Local: H00, Expansion: H01~H10)
Slot No.: 2 digits in hexadecimal (H00~H0B)
⚠ In case of PLCS, assign H0000 to ***Base/Slot.***

Base Number	Slot	*Base/Slot*
Local base	Slot No. 5	H0005
1st expansion	Slot No. 3	H0103
10th expansion	Slot No. 7	H0A07
14th expansion	Slot No. 11	H0E0B
16th expansion	Slot No. 10	H100A

Assigning *S* - [Positioning data]
Assign constant or an address of word device where positioning data is stored.
Positioning data are as below:

S	Select data type to teach (0~2) and positioning data number (1~600). (*) Refer to the table below for the details.
S+1	Assign positioning data to teach. Target coordinate: -2147483648 ~ 2147483647 Operation speed: CM1: 1 ~ 1000000pps[26] CM3: 1 ~ 100000pps Circular interpolation address: -2147483648 ~ 2147483647 Dwell time: 0 ~ 65535ms
S+2	

Device *S* is configured as below:

Bit 15 … Bit 12	Bit 11 … Bit 8	Bit 7 … Bit 4	Bit 3 … Bit 0
Data type to teach: 0: Target coordinate 1: Operation speed 2: Circular interpolation address 3: Dwell time	Positioning data number CM1 : 1 ~ 600 (H001~H258) CM3 : 1 ~ 30 (H001~H01E)		

[26] pps: Pulse per second

Assigning *D* - [Operation status and error code]

Assign an address of word device to store the operation status of positioning. Operation status and error codes are stored in the word device *D*.

Result format is as following:

B15 ··· B8	B7 ··· B0
Error code	Operation Status

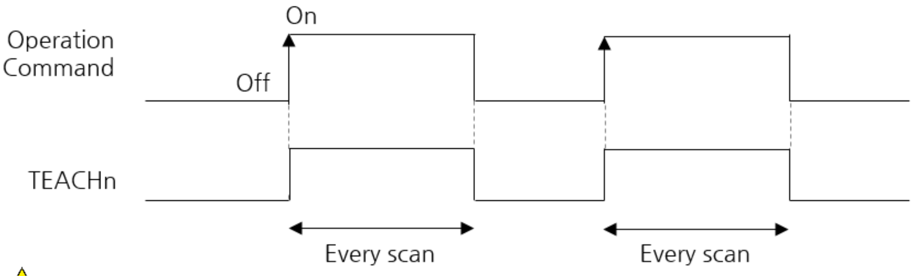
Upper byte: Error code

H00	No error
H10	The parameter is mistyped.
HFF	The instruction is not supported.

Lower byte: Operation Status

Bit 0	The flag is set when the instruction is in process.
Bit 1	The flag is set when the process is completed.
Bit 2	The flag is set when an error occurs. This flag sets simultaneously with bit 1.
Bit 3~7	Reserved

Execution Condition



⚠ Using pulse contact as execution condition of TEACHn instruction is recommended.

Operation Error

Error flag (F110)

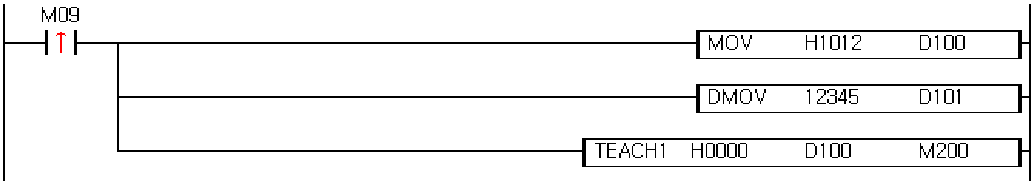
F110 turns ON for 1 scan when the address of device assigned by @D exceeds the range of device D. (Range of device D depends on CPU type)

Program Example

TEACH1

When M09 turns ON, TEACH1 instruction is executed. The positioning module is located on slot number 0 of local base (=H0000). According to the values assigned to D100, D101 and D102, the instruction teaches operation speed of positioning data number 18 of axis X. The operation status is stored in M200.

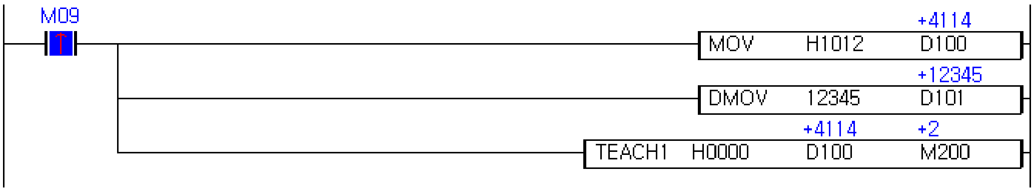
Ladder Diagram (LD)



Instruction List (IL)

Instruction	Device		
LDP	M09		
MOV	H1012	D100	
DMOV	12345	D101	
TEACH1	H0000	D100	M200

- When M09 turns ON, MOV and DMOV instructions are executed. “H1012” is moved to word device D100. “12345” is moved to double word device D101 (D101~D102).
- The value assigned to D100, “H1012” means that operation speed of positioning data number 18 will be changed.
- The value assigned to D101 and D102, “12345” means that opertion speed of positioning data number 18 will be changed into 12345 pps.
- The operation status is stored in M200. Since the operation is finished, bit 1 is set. There is no error occurred in this operation.


M09
+4114
MOV H1012 D100
+12345
DMOV 12345 D101
+4114 +2
TEACH1 H0000 D100 M200