Chapter 5　IL(Instruction List)

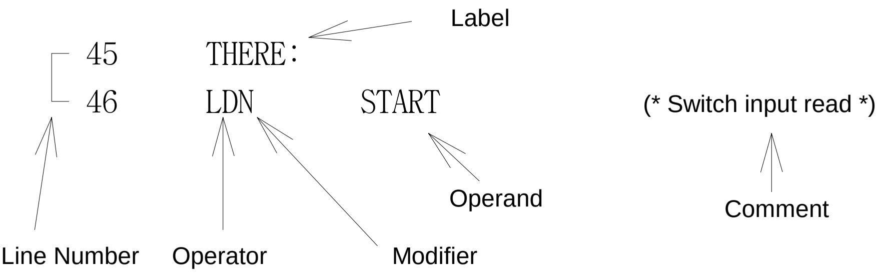
5. IL(Instruction List)

5.1 Overview

☐ IL is the type of assembly language.

☐ IL can be applied to simple PLC system.

☐ Type



5.2. Current Result(CR)

☐ Current Result(CR) is the resistor that has the operation result any time.

☐ Only one CR exists in IL.

☐ CR can be any data type.

☐ LD(Load) is the operator that defines the data type of CR by inputting any value in CR.

Example

Variable %IX0.0.0 value is input for LD %IX0.0.0 in CR. The data type of CR will be BOOL since the data type of variable expressed by X is BOOL. If VAL variable is declared as INT and used as LD VAL, the variable value of VAL will be input to CR and data type of CR will be INT.

☐ ST(Store) is the operator that stores CR to certain variable.

Example

If VAL variable is declared as INT and used as ST VAL, it means that CR is input to the variable of VAL. At this moment, CR shall be the data type of INT. If CR is other data type except INT, the error will occur during compiling.

When

```
LD          %IX0.0.0
ST          VAL (assume that VAL variable is declared as INT)
```

is set, as CR is defined as BOOL in first line and CR is used as INT, the error occurs during compiling.

```
LD          %IX0.0.0
ST    START
LD          20
ST          VAL (assume that START variable is declared as BOOL and VAL variable is as INT)
```

As the data type is same in above example when selected CR is stored, this is normal program.

5.3. Instructions

- □ IL consists of sequential command.
- □ Each command consists of the operator, which contains the modifier, and operand.

5.3.1. Label

- □ Label is marked at the region of operator putting colon(:) behind the label name.
- □ Label is used for the destination of jump command.

5.3.2. Modifiers

- □ Modifier is put behind the operator and executed by modifying original operation function. The type of modifier is N, (and C.
- □ Modifier "N" indicates Boolean Negation of operand.

Example

ANDN   %IX2.0.0 is translated as below.

    CR <= CR AND NOT %IX2.0.0

When N is used for JMP, CAL or RET, it means that the command is executed if CR is BOOL 0.

- □ Modifier "(" is the operation of operator")" is delayed till it meets the operator.

Operates other operation since one CR exists and IL stores CR for a short time and the result and the delay operation which stored CR value can be executed.

Type	Item	Description
(	Modifier	Indicate the delay of operation.
)	Operator	Executes delayed operation.

Example

AND(%IX1.0.0

OR %IX2.0.0

)

CR <=CR AND (%IX1.0.0 OR %IX2.0.0)

Therefore, AND execution is delayed till) appears. The operation %IX1.0.0 OR %IX2.0.0 in parentheses is first executed and the result is operated lately.

□ Modifier " C " means that the selected command is executed only when current operated CR is BOOL 1.

Example

JMPC THERE

If CR is BOOL1, jump to THERE

5.3.3. Operators

□ Basic operator is described as below.

No.	Operator	Modifiers	Operand	Description
1	LD	N	Data	Operand is input to Current Result(CR).
2	ST	N	Data	Current Result(CR) is stored at the operand.
3	S		BOOL	If CR is BOOL1, the operand will be 1.
	R		BOOL	If CR is BOOL1, the operand will be 0.
4	AND	N,(	Data	Logic AND operation
5	OR	N,(	Data	Logic OR operation
6	XOR	N,(	Data	Logic XOR operation
7	ADD	(	Data	Arithmetic + operation
8	SUB	(	Data	Arithmetic - operation
9	MUL	(	Data	Arithmetic * operation
10	DIV	(	Data	Arithmetic / operation
11	GT	(	Data	Comparison operation: >(greater than)
12	GE	(	Data	Comparison operation: >=(greater than or equal)
13	EQ	(	Data	Comparison operation: =(equal)
14	NE	(	Data	Comparison operation: <>(not equal)
15	LE	(	Data	Comparison operation: <=(less than or equal)
16	LT	(	Data	Comparison operation: < (less than)
17	JMP	C,N	Label	Jump to label
18	CAL	C,N	Name	Call function block
19	RET	C,N		Return at the function or function block
20	)			Executes delayed operation with '(' modifier.

□ The operator from 4 to 16 executes below function.

CR **<=** ***CR*** **operand**

CR and operand's values are operated and, then, the result is stored to CR again.

Example

AND %IX1.0.0 is translated as below.

CR <= CR AND %IX1.0.0

□ The comparison operator operates left CR and right operand and stores the BOOL result in CR.

Example

For GT %MW10, CR will be BOOL 1 if CR is larger than memory word 10's value and will be 0 if not.

□ Most of operation command does not change CR data type after operation. But, the data type of CR is changed for the comparison command.

Example

```
LD      VAL       (a)
EQ      GROSS     (b)
AND     %IX0.0.0  (c)
ST      START     (d)
```

(Assuïe that START variable is declared as BOOL and VAL and CR variables are declared as INT.)

Input VAL of INT value to CR in (a) line. Input BOOL 1 value in CR if CR and INT values called GROSS are same in (b) line and input BOOL 0 in CR if not. The data type of CR will be changed from INT to BOOL. Therefore, the compile error does not occur normally in case of using command in (c) and (d) lines.

5.3.3.1. Details of operator

(1) LD

Description	Input operand to CR. At this moment, CR's data type is changed to the operand's one.	
Modifier	N: The operand value is reversed and input to CR if the operand is BOOL.	
Operand	All data type is available. The constant is available.	
Example	LD TRUE	Input BOOL 1 value in CR. (BOOL data type)
	LD INT_VALUE	Input INT_VALUE as INT variable in CR. (INT data type)
	LD T#1S	Input T#1S as duration constant in CR. (TIME data type)
	LDN B_VALUE	B_VALUE as BOOL variable is reversed and be put in CR. (BOOL data type)

(2) ST

Description	Input CR to operand At this moment, CR's data type shall be same to the operand's one. CR value is not changed.	
Modifier	N: In case the CR data type is BOOL, input CR to certain variable value in the operand after reversing. CR value is not changed.	
Operand	All data type is available. The constant shall not input. Shall be same to CR's data type.	
Example	LD FALSE	Input BOOL 0 to CR. CR's data type is BOOL.
	ST B_VALUE1	Input CR value 0 to B_VALUE1 variable of BOOL data type.
	STN B_VALUE2	Convert CR value(1) and input it to B_VALUE2 variable of BOOL data type.
	LD INT_VALUE	Input INT_VALUE of INT variable to CR. CR's data type is INT.
	ST I_VALUE1	Input CR value to I_VALUE1 variable of INT data type.
	LD D#1995-12-25	Input D#1995-12-25 of date constant to CR. CR's data type is DATE.
	ST D_VALUE1	Input CR value to D-VALUE1 variable of DATE data type.

(3) S(Set)

Description	If CR value is BOOL1, operand value of BOOL data type will be 1. If CR value is BOOL0, no operation is executed. CR value is not changed.	
Modifier	None	
Operand	BOOL data type is available only. The constant shall not input.	
Example	LD FALSE	Input BOOL 0 to CR. CR's data type is BOOL.
	S B_VALUE1	No operation is executed since CR is 0. B_VALUE1 variable will not be changed.
	LD TRUE	Input BOOL1 value to CR. CR's data type is BOOL.
	S B_VALUE2	B_VALUE2 variable of BOOL data type will be 1 since CR is 1.

(4) R(Reset)

Description	If CR value is BOOL1, operand value of BOOL data type will be 0. If CR value is BOOL0, no operation is executed. CR value is not changed.	
Modifier	None	
Operand	BOOL data type is available only. The constant shall not input.	
Example	LD FALSE	Input BOOL 0 to CR. CR's data type is BOOL.
	R B_VALUE1	No operation is executed since CR is 0. B_VALUE1 variable will not be changed.
	LD TRUE	Input BOOL1 value to CR. CR's data type is BOOL.
	R B_VALUE2	Because CR value is 1, B_VALUE 2 variable of BOOL data type will be 0.
	ST B_VALUE3	Input CR 1 to B_VALUE3 variable of BOOL data type.

(5) AND

Description	Executes logic AND operation of CR value and operand value and input the result to CR. The CR's data type shall be same to the operand's one. The operand value is not changed.	
Modifier	N: If the operand is BOOL, reverse the operand value and calculate with CR value. (: If operand's data type is BOOL. Current CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	BOOL, BYTE, WORD, DWORD and LWORD data types are available. The constant is available also.	
Example	LD B_VALUE1	Input B_VALUE1 of BOOL data type to CR.
	AND B_VALUE2	Execute AND operation of CR and B_VALUE2 of BOOL data type and input the result to CR.
	ANDN B_VALUE3	Convert CR and B_VALUE3 of BOOL data type and execute AND operation and input the result to CR.
	ST B_VALUE4	Input CR to B_VALUE4 of BOOL data type. B_VALUE4 <= B_VALUE1 AND B_VALUE2 AND NOT(B_VALUE3)
	LD W_VALUE1	Input W_VALUE1 of WORD variable to CR. CR's data type is WORD.
	AND W_VALUE2	Execute AND operation of CR and W_VALUE2 of WORD data type and input the result to CR.
	ST W_VALUE3	Input CR to W_VALUE3 of WORD data type. W_VALUE3 <== W_VALUE1 AND W_VALUE2
	LD B_VALUE1	Input B_VALUE1 of BOOL data type to CR. CR's data type is BOOL.
	AND(B_VALUE2	Store CR value to other position and input B_VALUE2 of BOOL data type to CR.
	OR B_VALUE3	Execute OR operation of CR value and B_VALUE3 value of BOOL data type and input the result to CR.
	)	Execute AND operation of current CR and other CR values and input the result to CR.
	ST B_VALUE4	Input CR to B_VALUE4 of BOOL data type. B_VALUE4 <== B_VALUE1 AND (B_VALUE2 OR B_VALUE3)

(6) OR

Description	Executes logic OR operation of CR value and operand value and input the result to CR. The CR's data type shall be same to the operand's one. The operand value is not changed.	
Modifier	N: Reverse the operand value and calculate with CR value if the operand is BOOL data type. (: If operand's data type is BOOL. Current CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	BOOL, BYTE, WORD, DWORD and LWORD data types are available. The constant is available also.	
Example	LD B_VALUE1	Input B_VALUE1 of BOOL data type to CR. CR's data type is BOOL.
	OR B_VALUE2	Execute OR operation of CR and B_VALUE2 of BOOL data type and input the result to CR.
	ORN B_VALUE3	Convert CR and B_VALUE3 of BOOL data type and execute OR operation and input the result to CR.
	ST B_VALUE4	Input CR to B_VALUE4 of BOOL data type. B_VALUE4 <= B_VALUE1 OR B_VALUE2 OR NOT(B_VALUE3)
	LD W_VALUE1	Input W_VALUE1 of WORD variable to CR. CR's data type is WORD.
	OR W_VALUE2	Execute OR operation of CR and W_VALUE2 of WORD data type and input the result to CR.
	ST W_VALUE3	Input CR to W_VALUE3 of WORD data type. W_VALUE3 <== W_VALUE1 AND W_VALUE2
	LD B_VALUE1	Input B_VALUE1 of BOOL data type to CR. CR's data type is BOOL.
	OR(B_VALUE2	Store CR value to other position and input B_VALUE2 of BOOL data type to CR.
	AND B_VALUE3	Execute AND operation of CR value and B_VALUE3 value of BOOL data type and input the result to CR.
	)	Execute OR operation of current CR and other CR values and input the result to CR.
	ST B_VALUE4	Input CR to B_VALUE4 of BOOL data type. B_VALUE4 <== B_VALUE1 OR (B_VALUE2 AND B_VALUE3)

(7) XOR

Description	Executes logic XOR operation of CR value and operand value and input the result to CR. The CR's data type shall be same to the operand's one. The operand value is not changed.	
Modifier	N: In case the operand is BOOL data type, reverse the operand value and calculate with CR value. (: If operand's data type is BOOL. Current CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	BOOL, BYTE, WORD, DWORD and LWORD data types are available. The constant is available also.	
Example	LD B_VALUE1	Input B_VALUE1 of BOOL data type to CR. CR's data type is BOOL.
	XOR B_VALUE2	Execute XOR operation of CR and B_VALUE2 of BOOL data type and input the result to CR.
	XORN B_VALUE3	Convert CR and B_VALUE3 of BOOL data type and execute XOR operation and input the result to CR.
	ST B_VALUE4	Input CR to B_VALUE4 of BOOL data type. B_VALUE4 <= B_VALUE1 XOR B_VALUE2 OR NOT(B_VALUE3)
	LD W_VALUE1	Input W_VALUE1 of WORD variable to CR. CR's data type is WORD.
	XOR W_VALUE2	Execute XOR operation of CR and W_VALUE2 of WORD data type and input the result to CR.
	ST W_VALUE3	Input CR to W_VALUE3 of WORD data type. W_VALUE3 <== W_VALUE1 AND W_VALUE2
	LD B_VALUE1	Input B_VALUE1 of BOOL data type to CR. CR's data type is BOOL.
	XOR(B_VALUE2	Store CR value to other position and input B_VALUE2 of BOOL data type to CR.
	AND B_VALUE3	Execute AND operation of CR value and B_VALUE3 value of BOOL data type and input the result to CR.
	)	Execute XOR operation of current CR and other CR values and input the result to CR.
	ST B_VALUE4	Input CR to B_VALUE4 of BOOL data type. B_VALUE4 <== B_VALUE1 XOR (B_VALUE2 AND B_VALUE3)

(8) ADD

<table>
<tr><td>Description</td><td colspan="3">Executes arithmetic ADD operation of CR value and operand value and input the result to CR.
The CR's data type shall be same to the operand's one.
The operand value is not changed.</td></tr>
<tr><td>Modifier</td><td colspan="3">(:CR is stored at other position temporarily and operand's value will be input to CR.
(delay operation)</td></tr>
<tr><td>Operand</td><td colspan="3">SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL and LREAL data types are available.
The constant is available also.</td></tr>
<tr><td rowspan="8">Example</td><td>LD</td><td>I_VALUE1</td><td>Input I_VALUE1 of INT data type to CR.
CR's data type is INT.</td></tr>
<tr><td>ADD</td><td>I_VALUE2</td><td>Execute + operation of CR and I_VALUE2 of INT data type and input the result to CR.</td></tr>
<tr><td>ST</td><td>I_VALUE3</td><td>Input CR to I_VALUE3 of INT data type.
I_VALUE3 <= I_VALUE1 + I_VALUE2</td></tr>
<tr><td>LD</td><td>D_VALUE1</td><td>Input D_VALUE1 of DINT data type to CR.
CR's data type is DINT.</td></tr>
<tr><td>ADD(</td><td>D_VALUE2</td><td>Store CR value to other position and input D_VALUE2 of DINT data type to CR.</td></tr>
<tr><td>DIV</td><td>D_VALUE3</td><td>Execute arithmetic / operation of CR and D_VALUE3 of DINT data type and input the result to CR.</td></tr>
<tr><td>)</td><td></td><td>Execute + operation of current CR and other CR values and input the result to CR.</td></tr>
<tr><td>ST</td><td>D_VALUE4</td><td>Input CR to D_VALUE4 of DINT data type.
D_VALUE4 <== D_VALUE1 + (D_VALUE2 / D_VALUE3)</td></tr>
</table>

(9) SUB

Description	Executes arithmetic - operation of CR value and operand value and input the result to CR. The CR's data type shall be same to the operand's one. The operand value is not changed.	
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL and LREAL data types are available. The constant is available also.	
Example	LD I_VALUE1	Input I_VALUE1 of INT data type to CR. CR's data type is INT.
	SUB I_VALUE2	Execute - operation of CR and I_VALUE2 of INT data type and input the result to CR.
	ST I_VALUE3	Input CR to I_VALUE3 of INT data type. I_VALUE3 <= I_VALUE1 - I_VALUE2
	LD D_VALUE1	Input D_VALUE1 of DINT data type to CR. CR's data type is DINT.
	SUB(D_VALUE2	Store CR value to other position and input D_VALUE2 of DINT data type to CR.
	MUL D_VALUE3	Execute arithmetic * operation of CR and D_VALUE3 of DINT data type and input the result to CR.
	)	Execute - operation of current CR and other CR values and input the result to CR.
	ST D_VALUE4	Input CR to D_VALUE4 of DINT data type. D_VALUE4 <== D_VALUE1 - (D_VALUE2 * D_VALUE3)

(10) MUL

Description	Executes arithmetic * operation of CR value and operand value and input the result to CR. The CR's data type shall be same to the operand's one. The operand value is not changed.	
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL and LREAL data types are available. The constant is available also.	
Example	LD I_VALUE1	Input I_VALUE1 of INT data type to CR. CR's data type is INT.
	MUL I_VALUE2	Execute * operation of CR and I_VALUE2 of INT data type and input the result to CR.
	ST I_VALUE3	Input CR to I_VALUE3 of INT data type. I_VALUE3 <= I_VALUE1 * I_VALUE2
	LD D_VALUE1	Input D_VALUE1 of DINT data type to CR. CR's data type is DINT.
	MUL(D_VALUE2	Store CR value to other position and input D_VALUE2 of DINT data type to CR.
	SUB D_VALUE3	Execute arithmetic - operation of CR and D_VALUE3 of DINT data type and input the result to CR.
	)	Execute MUL operation of current CR and other CR values and input the result to CR.
	ST D_VALUE4	Input CR to D_VALUE4 of DINT data type. D_VALUE4 <== D_VALUE1 * (D_VALUE2 - D_VALUE3)

(11) DIV

Description	Executes arithmetic / operation of CR value and operand value and input the result to CR. The CR's data type shall be same to the operand's one. The operand value is not changed.		
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)		
Operand	SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL and LREAL data types are available. The constant is available also.		
Example	LD	I_VALUE1	Input I_VALUE1 of INT data type to CR. CR's data type is INT.
	DIV	I_VALUE2	Execute arithmetic / operation of CR and I_VALUE2 of INT data type and input the result to CR.
	ST	I_VALUE3	Input CR to I_VALUE3 of INT data type. I_VALUE3 <= I_VALUE1 / I_VALUE2
	LD	D_VALUE1	Input D_VALUE1 of DINT data type to CR. CR's data type is DINT.
	DIV(	D_VALUE2	Store CR value to other position and input D_VALUE2 of DINT data type to CR.
	ADD	D_VALUE3	Execute arithmetic + operation of CR and D_VALUE3 of DINT data type and input the result to CR.
	)		Execute arithmetic / operation of current CR and other CR values and input the result to CR.
	ST	D_VALUE4	Input CR to D_VALUE4 of DINT data type. D_VALUE4 <== D_VALUE1 / (D_VALUE2 + D_VALUE3)

(12) GT

Description	CR and operand values are compared and BOOL result is input to CR. If CR is greater than the operand, CR will be 1. Otherwise CR will be 0. The CR's data type shall be same to the operand's one. The operand value is not changed. After operation, CR's data type will be BOOL regardless of data type of operand.	
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	All data types excluding ARRAY are available. The constant is available also.	
Example		Ex) If I_VAL1 = 50, I_VAL2 = 100 IVAL_3 = 70
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	GT I_VAL2	Compare CR and I_VAL2 of INT data type and input the result to CR.(CR is 0 since I_VAL1 < I_VAL2)
	ST B_VAL1	Input CR to B_VAL1 variable of BOOL data type. B_VAL1 <== FALSE
	LD I_VAL2	Input I_VAL2 of INT data type to CR.
	GT I_VAL1	Compare CR and I_VAL1 of INT data type and input the result to CR.(CR is 1 since I_VAL1 < I_VAL2)
	ST B_VAL2	Input CR to B_VAL2 variable of BOOL data type. B_VAL2 <== TRUE
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	GT(I_VAL2	Store CR to other position and input I_VAL2 of INT data type to CR.
	SUB I_VAL3	Execute arithmetic / operation of CR and I_VAL3 of INT data type and input the result to CR.
	)	Compare CR stored in other position and current CR values and input the result to CR.(CR is 1 since stored CR > Current CR)
	ST B_VAL3	Input CR to B_VAL3 variable of BOOL data type. B_VAL3 <== TRUE

(13) GE

Description	CR and operand values are compared and BOOL result is input to CR. If CR is greater than or equal the operand, CR will be 1. Otherwise CR will be 0. The CR's data type shall be same to the operand's one. The operand value is not changed. After operation, CR's data type will be BOOL regardless of data type of operand.	
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	All data types excluding ARRAY are available. The constant is available also.	
Example		Ex) If I_VAL1 = 50, I_VAL2 = 100 IVAL_3 = 70
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	GE I_VAL2	Compare CR and I_VAL2 of INT data type and input the result to CR.(CR is 0 since I_VAL1 < I_VAL2)
	ST B_VAL1	Input CR to B_VAL1 variable of BOOL data type. B_VAL1 <== FALSE
	LD I_VAL2	Input I_VAL2 of INT data type to CR.
	GE I_VAL1	Compare CR and I_VAL1 of INT data type and input the result to CR.(CR is 1 since I_VAL1 < I_VAL2)
	ST B_VAL2	Input CR to B_VAL2 variable of BOOL data type. B_VAL2 <== TRUE
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	GE(I_VAL2	Store CR to other position and input I_VAL2 of INT data type to CR.
	SUB I_VAL3	Execute arithmetic - operation of CR and I_VAL3 of INT data type and input the result to CR.
	)	Compare CR stored in other position and current CR values and input the result to CR.(CR is 1 since stored CR > Current CR)
	ST B_VAL3	Input CR to B_VAL3 variable of BOOL data type. B_VAL3 <== TRUE

(14) EQ

<table>
<tr><td>Description</td><td colspan="3">CR and operand values are compared and BOOL result is input to CR.
If CR is equal the operand, CR will be 1.
Otherwise CR will be 0.
The CR's data type shall be same to the operand's one.
The operand value is not changed.
After operation, CR's data type will be BOOL regardless of data type of operand.</td></tr>
<tr><td>Modifier</td><td colspan="3">(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)</td></tr>
<tr><td>Operand</td><td colspan="3">All data types excluding ARRAY are available.
The constant is available also.</td></tr>
<tr><td>Example</td><td></td><td></td><td>Ex) If I_VAL1 = 50, I_VAL2 = 100 IVAL_3 = 50</td></tr>
<tr><td></td><td>LD</td><td>I_VAL1</td><td>Input I_VAL1 of INT data type to CR.</td></tr>
<tr><td></td><td>EQ</td><td>I_VAL2</td><td>Compare CR and I_VAL2 of INT data type and input the result to CR.(CR is 0 since I_VAL1 < I_VAL2)</td></tr>
<tr><td></td><td>ST</td><td>B_VAL1</td><td>Input CR to B_VAL1 variable of BOOL data type.
B_VAL1 <== FALSE</td></tr>
<tr><td></td><td>LD</td><td>I_VAL1</td><td>Input I_VAL2 of INT data type to CR.</td></tr>
<tr><td></td><td>EQ</td><td>I_VAL3</td><td>Compare CR and I_VAL1 of INT data type and input the result to CR.(CR is 1 since I_VAL1 < I_VAL2)</td></tr>
<tr><td></td><td>ST</td><td>B_VAL2</td><td>Input CR to B_VAL2 variable of BOOL data type.
B_VAL2 <== TRUE</td></tr>
<tr><td></td><td>LD</td><td>I_VAL1</td><td>Input I_VAL1 of INT data type to CR.</td></tr>
<tr><td></td><td>EQ(</td><td>I_VAL2</td><td>Store CR to other position and input I_VAL2 of INT data type to CR.</td></tr>
<tr><td></td><td>SUB</td><td>I_VAL3</td><td>Execute arithmetic - operation of CR and I_VAL3 of INT data type and input the result to CR.</td></tr>
<tr><td></td><td>)</td><td></td><td>Compare CR stored in other position and current CR values and input the result to CR.(CR is 1 since stored CR = Current CR)</td></tr>
<tr><td></td><td>ST</td><td>B_VAL3</td><td>Input CR to B_VAL3 variable of BOOL data type.
B_VAL3 <== TRUE</td></tr>
</table>

(15) NE

Description	CR and operand values are compared and BOOL result is input to CR. If CR is not equal the operand, CR will be 1. Otherwise CR will be 0. The CR's data type shall be same to the operand's one. The operand value is not changed. After operation, CR's data type will be BOOL regardless of data type of operand.	
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	All data types excluding ARRAY are available. The constant is available also.	
Example		Ex) If I_VAL1 = 50, I_VAL2 = 100 IVAL_3 = 50
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	NE I_VAL3	Compare CR and I_VAL3 of INT data type and input the result to CR.(CR is 0 since I_VAL1 = I_VAL3)
	ST B_VAL1	Input CR to B_VAL1 variable of BOOL data type. B_VAL1 <== FALSE
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	NE I_VAL2	Compare CR and I_VAL2 of INT data type and input the result to CR.(CR is 1 since I_VAL1 <> I_VAL2)
	ST B_VAL2	Input CR to B_VAL2 variable of BOOL data type. B_VAL2 <== TRUE
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	NE(I_VAL2	Store CR to other position and input I_VAL2 of INT data type to CR.
	SUB I_VAL3	Execute arithmetic - operation of CR and I_VAL3 of INT data type and input the result to CR.
	)	Compare CR stored in other position and current CR values and input the result to CR.(CR is 0 since stored CR = Current CR)
	ST B_VA3	Input CR to B_VAL3 variable of BOOL data type. B_VAL2 <== FALSE

(16) LE

<table>
<tr><td>Description</td><td colspan="2">CR and operand values are compared and BOOL result is input to CR.
The operand is less than or equal the CR, CR will be 1.
Otherwise CR will be 0.
The CR's data type shall be same to the operand's one.
The operand value is not changed.
After operation, CR's data type will be BOOL regardless of data type of operand.</td></tr>
<tr><td>Modifier</td><td colspan="2">(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)</td></tr>
<tr><td>Operand</td><td colspan="2">All data types excluding ARRAY are available.
The constant is available also.</td></tr>
<tr><td rowspan="12">Example</td><td></td><td>Ex) I_VAL1 = 50, I_VAL2 = 100 IVAL_3 = 70</td></tr>
<tr><td>LD I_VAL2</td><td>Input I_VAL2 of INT data type to CR.</td></tr>
<tr><td>LE I_VAL1</td><td>Compare CR and I_VAL1 of INT data type and input the result to CR.(CR is 0 since I_VAL1 < I_VAL2)</td></tr>
<tr><td>ST B_VAL1</td><td>Input CR to B_VAL1 variable of BOOL data type.
B_VAL1 <== FALSE</td></tr>
<tr><td>LD I_VAL1</td><td>Input I_VAL1 of INT data type to CR.</td></tr>
<tr><td>LE I_VAL2</td><td>Compare CR and I_VAL2 of INT data type and input the result to CR.(CR is 1 since I_VAL1 < I_VAL2)</td></tr>
<tr><td>ST B_VAL2</td><td>Input CR to B_VAL2 variable of BOOL data type.
B_VAL2 <== TRUE</td></tr>
<tr><td>LD I_VAL1</td><td>Input I_VAL1 of INT data type to CR.</td></tr>
<tr><td>LE(I_VAL2</td><td>Store CR to other position and input I_VAL2 of INT data type to CR.</td></tr>
<tr><td>SUB I_VAL3</td><td>Execute arithmetic - operation of CR and I_VAL3 of INT data type and input the result to CR.</td></tr>
<tr><td>)</td><td>Compare CR stored in other position and current CR values and input the result to CR.(CR is 0 since stored CR > Current CR)</td></tr>
<tr><td>ST B_VA3</td><td>Input CR to B_VAL3 variable of BOOL data type.
B_VAL2 <== FALSE</td></tr>
</table>

(17) LT

Description	CR and operand values are compared and BOOL result is input to CR. If CR is less than the operand, CR will be 1. Otherwise CR will be 0. The CR's data type shall be same to the operand's one. The operand value is not changed. After operation, CR's data type will be BOOL regardless of data type of operand.	
Modifier	(:CR is stored at other position temporarily and operand's value will be input to CR. (delay operation)	
Operand	All data types excluding ARRAY are available. The constant is available also.	
Example		Ex) If I_VAL1 = 50, I_VAL2 = 100 IVAL_3 = 70
	LD I_VAL2	Input I_VAL2 of INT data type to CR.
	LT I_VAL1	Compare CR and I_VAL1 of INT data type and input the result to CR.(CR is 0 since I_VAL1 < I_VAL2)
	ST B_VAL1	Input CR to B_VAL1 variable of BOOL data type. B_VAL1 <== FALSE
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	LT I_VAL2	Compare CR and I_VAL2 of INT data type and input the result to CR.(CR is 1 since I_VAL1 < I_VAL2)
	ST B_VAL2	Input CR to B_VAL2 variable of BOOL data type. B_VAL2 <== TRUE
	LD I_VAL1	Input I_VAL1 of INT data type to CR.
	LT(I_VAL2	Store CR to other position and input I_VAL2 of INT data type to CR.
	SUB I_VAL3	Execute arithmetic - operation of CR and I_VAL3 of INT data type and input the result to CR.
	)	Compare CR stored in other position and current CR values and input the result to CR.(CR is 0 since stored CR > Current CR)
	ST B_VA3	Input CR to B_VAL3 variable of BOOL data type. B_VAL2 <== FALSE

(18) JMP

Description	Move the execution flow with the label described in the operand.	
Modifier	C : Move to label if CR of BOOL data type is TRUE(1). Next command is executed without moving if CR of BOOL data type is FALSE(0). N : Move to label if CR of BOOL data type is FALSE(0). Next command is executed without moving if CR of BOOL data type is TRUE(1). Move to the label regardless of CR value if there is no modifier.	
Operand	Label name	
Example		Program that input I_VAL1 or I_VAL2 to I_VAL3 according to B_VAL1 of BOOL data type.
	LD B_VAL1	Input B_VAL1 of BOOL data type to CR.
	JMPC THERE1	If CR is 1, move to THERE1 label and if CR is 0, execute next sentence.
	LD I_VAL1	CR <== I_VAL1
	JMP THERE2	Move to THERE2 label.
	THERE1:	THERE1 label
	LD I_VAL2	CR <== I_VAL2
	THERE2:	THERE2 label
	ST I_VAL3	I_VAL3 <== CR
		Program that executes SEL function if B_VAL2 of BOOL data type is 1.
	LD B_VAL2	CR <== B_VAL2
	JMPN THERE3	If CR is 0(FALSE), move to THERE3 label.
	LD B_VALUE	CR <== B_VALUE
	SEL	Call SEL function.
	G:= CURRENT RESULT	
	IN1:= I_VAL1	
	IN2:= I_VAL2	
	ST I_VAL3	I_VAL3 <== CR
	THERE3:	THERE3 label

(19) CAL

Description	Call the function block named in the operand.	
Modifier	C : The function block is called if CR of BOOL data type is TRUE(1). The function block is not called if CR of BOOL data type is FALSE(0). N : The function block is called if CR of BOOL data type is FALSE(0). The function block is not called if CR of BOOL data type is TRUE(1). Call the function block regardless of CR value if there is no modifier.	
Operand	Function block name	
Example		Program that calls TON, On Delay Timer, if B_VAL1 of BOOL data type is 1(TRUE).
	LD B_VAL1	Input B_VAL1 of BOOL data type to CR.
	CALC TON TIMER1	If CR is 1, the instance calls On delay timer TON of TIMER1.
	IN:= T_INPUT	
	PT:= PRE_TIME	
		Program that calls CTU of up-counter if B_VAL2 of BOOL data type is 0(FALSE).
	LD B_VAL2	Input B_VAL2 of BOOL data type to CR.
	CALN CTU COUNT1	If CR is 0, the instance calls Up-counter CTU of COUNTER1.
	CU:= B_UP	
	R:= B_RESET	
	PV:= 100	
		Program that calls Down-counter CTD regardless CR value.
	CAL CTD COUNT2	The instance calls Down-counter CTD of COUNTER2.
	CD:= B_DOWN	
	LD:= B_LDV	
	PV:= 300	

(20) RET

Description	Returned from the function or function block.	
Modifier	C : Return if CR of BOOL data type is TRUE(1). Not returned if CR of BOOL data type is FALSE(0). N : Return if CR of BOOL data type is FALSE(0). Not returned if CR of BOOL data type is TRUE(1). Returned regardless of CR if there is no modifier.	
Operand	None	
Example		Function that executes MUL operation of I_VAL1 and I_VAL2 of INT data types and inputs the result to I_VAL3. If arithmetic * operation error occurs, input 0 to I_VAL3 and return.
	LD I_VAL1	
	MUL I_VAL2	
	ST I_VAL3	
	LD _ERR	CR <== System error flag
	RETN	The instance is returned if CR is 0.
	LD 0	
	ST I_VAL3	I_VAL3 <== 0
	RET	Returned.

(21))

Description	Execute delayed operation using (.		
Modifier	None		
Operand	None		
Example	LD	I_VAL1	I_VAL4 <== (I_VAL1 + IVAL2) * I_VAL3
	ADD	I_VAL2	
	MUL	I_VAL3	
	ST	I_VAL4	
	LD	I_VAL1	I_VAL4 <== I_VAL1 + (IVAL2 * I_VAL3)
	ADD(	I_VAL2	
	MUL	I_VAL3	
	)		
	ST	I_VAL4	
	LD	L_VAL1	L_VAL7 <== (L_VAL1 + (L_VAL2 * (L_VAL3 - L_VAL4) + L_VAL5)) / L_VAL6
	ADD(	L_VAL2	
	MUL(	L_VAL3	
	SUB	L_VAL4	
	)		
	ADD	L_VAL5	
	)		
	DIV	L_VAL6	
	ST	L_VAL7	

5.4. Calling functions and function blocks

- □ Call the function using the function name as operator.
- □ During calling the function, CR is input first to the function.
- □ If the function input is more than one, the other values is selected and the function is called.
- □ The output of function is input to CR.
- □ CR's data type will be the function output's one.

Example

```
LD          VAL
SIN
ST          RESULT            (Consider that VAL and RESULT is REAL data type)
```

If VAL variable is input to CR in first line and SIN function is called in second line, the CR will be input first input to SIN function. More than one input is impossible for SIN function and the output will be inserted to CR after SIN function execution. CR is stored to RESULT variable in third line.

```
LD          %IX0.0.0
SEL   G:=   CURRENT RESULT
      IN0:= VAL1
      IN1:= VAL2
ST    VAL3
```

This is the example of several input. In first line, CR is set and input as first input of SEL function. For other input, define each value and input the result to CR when calls SEL function and CR is stored to VAL3 variable.

□ Use JMP(JMPN, JMPC) command to call the function optionally.

Example

```
LD          %IX0.0.0
JMPN        THERE
LD          I_VAL1
ADD   IN1:= CURRENT RESULT
      IN2:= I_VAL2
      IN3:= I_VAL3
ST          I_VAL4
THERE:
```

Input %IX0.0.0 of BOOL data type to CR in first line and check the value is 1 or 0 in second line and move to THERE if %IX0.0.0 is 0 with label. If %IX0.0.0 is 1, JMP command is not executed and the function in next line is executed.

□ CAL operator is used for the function block call and instance name of function block declared previously is used for the operand.

□
```
CAL    INSTANCE  /* Call the function block as default. */
CALN   INSTANCE        /* Call the function block if CR is BOOL 0. */
CALC   INSTANCE        /* Call the function block if CR is BOOL 1. */
```

At this time, INSTANCE shall be declared as the instance for function block in advance.

□ CR is not input for the function block. Thus, all input required for the function block shall be set. The output value can not be output with CR.

Example

On-Delay Timer function block

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

(Assume that TIMER0 is declared as the instance of TON)

The input of On-Delay Timer is two and each input is set and each function block is called. The result is stored in TIMER0.Q and TIMER0.ET and, if the value is required, use it like fifth line.

MEMO