# Instruction List (version 1996)

## 3. Textual languages

The textual languages defined in this standard are IL (Instruction List) and ST (Structured Text). The sequential function chart (SFC) elements defined in 2.6 can be used in conjunction with either of these languages.

### 3.1 *Common elements*

See ‘Common Elements’.

### 3.2 *Language IL (Instruction List)*

This subclause defines the semantics of the IL (Instruction List) language whose formal syntax is given in B.2.

#### 3.2.1 *Instructions*

As illustrated in table 51, an *instruction list* is composed of a sequence of *instructions*. Each instruction shall begin on a new line and shall contain an operator with optional *modifiers*, and, if necessary for the particular operation, one or more *operands* separated by commas. Operands can be any of the data representations defined in 2.2 for literals and 2.4 for variables.
The instruction can be preceded by an identifying *label* followed by a colon (:). A *comment*, as defined in 2.1.5, if present, shall be the last element on a line. Empty lines can be inserted between instructions.

**Table 51 - Examples of instruction fields**

| Label | Operator | Operand | Comment |
|---|---|---|---|
| START: | LD | %IX1 | (* PUSH BUTTON *) |
| | ANDN | %MX5 | (* NOT INHIBITED *) |
| | ST | %QX2 | (* FAN ON *) |

#### *3.2.2 Operators, modifiers and operands*

Standard operators with their allowed modifiers and operands shall be as listed in table 52. The typing of operators shall conform to the conventions of 2.5.1.4.
Unless otherwise defined in table 52, the semantics of the operators shall be

result := result OP operand

That is, the value of the expression being evaluated is replaced by its current value operated upon by the operator with respect to the operand. For instance, the instruction AND %IX1 is interpreted as

result := result AND %IX1

The comparison operators shall be interpreted with the current result to the left of the comparison and the operand to the right, with a Boolean result. For instance, the instruction "GT %IW10" will have the Boolean result 1 if the current result is greater than the value of Input Word 10, and the Boolean result 0 otherwise.
The modifier "N" indicates Boolean negation of the operand. For instance, the instruction ANDN %IX2 is interpreted as

result := result AND NOT %IX2

The left parenthesis modifier "(" indicates that evaluation of the operator shall be deferred until a right parenthesis operator ")" is encountered, e.g., the sequence of instructions

```
AND(      %IX1
OR %IX2
)
```

shall be interpreted as

result := result AND (%IX1 OR %IX2)

The modifier "C" indicates that the associated instruction shall be performed only if the value of the currently evaluated result is Boolean 1 (or Boolean 0 if the operator is combined with the "N" modifier).

**Table 52 - Instruction List (IL) operators**

| No. | Operator | Modifiers | Operand | Semantics |
|---|---|---|---|---|
| 1 | LD | N | Note 2 | Set current result equal to operand |
| 2 | ST | N | Note 2 | Store current result to operand location |
| 3 | S | Note 3 | BOOL | Set Boolean operand to 1 |
|  | R | Note 3 | BOOL | Reset Boolean operand to 0 |
| 4 | AND | N, ( | BOOL | Boolean AND |
| 5 | & | N, ( | BOOL | Boolean AND |
| 6 | OR | N, ( | BOOL | Boolean OR |
| 7 | XOR | N, ( | BOOL | Boolean Exclusive OR |
| 8 | ADD | ( | Note 2 | Addition |
| 9 | SUB | ( | Note 2 | Subtraction |
| 10 | MUL | ( | Note 2 | Multiplication |
| 11 | DIV | ( | Note 2 | Division |
| 12 | GT | ( | Note 2 | Comparison: > |
| 13 | GE | ( | Note 2 | Comparison: >= |
| 14 | EQ | ( | Note 2 | Comparison: = |
| 15 | NE | ( | Note 2 | Comparison: <> |
| 16 | LE | ( | Note 2 | Comparison: <= |
| 17 | LT | ( | Note 2 | Comparison: < |
| 18 | JMP | C, N | LABEL | Jump to label |
| 19 | CAL | C, N | NAME | Call function block (note 4) |
| 20 | RET | C, N |  | Return from called function or function block |
| 21 | ) |  |  | Evaluate deferred operation |

NOTES

1 - See 3.2.2 for explanation of modifiers and evaluation of expressions.

2 - These operators shall be either overloaded or typed as defined in 2.5.1.4. The current result and the operand shall be of the same type.

3 - These operations are performed if and only if the value of the current result is Boolean 1.

4 - The function block name is followed by a parenthesized argument list as defined in 3.2.3.

5 - When a JMP instruction is contained in an ACTION... END_ACTION construct, the operand shall be a label within the same construct.

### 3.2.3 *Functions and function blocks*

Functions as defined in 2.5.1 shall be invoked by placing the function name in the operator field. The current result shall be used as the first argument of the function. Additional arguments, if required, shall be given in the operand field. The value returned by a function upon the successful execution of a RET instruction or upon reaching the physical end of the function shall become the "current result" described in 3.2.2.

Function blocks as defined in 2.5.2 can be invoked conditionally and unconditionally via the CAL (Call) operator listed in table 52. As shown in table 53, this invocation can take one of three forms. The input operators shown in table 54 can be used in conjunction with feature 3 of table 53.

**Table 53 - Function block invocation features for IL language**

| No. | Description/Example |
|---|---|
| 1 | CAL with input list:<br>`CAL  C10(CU:=%IX10, PV:=15)` |
| 2 | CAL with load/store of inputs:<br>`LD   15`<br>`ST   C10.PV`<br>`LD   %IX10`<br>`ST   C10.CU`<br>`CAL  C10` |
| 3 | Use of input operators:<br>`LD   15`<br>`PV   C10`<br>`LD   %IX10`<br>`CU   C10` |
| NOTE - A declaration such as VAR C10: CTU ; END_VAR is assumed in the above examples. | |

**Table 54 - Standard function block input operators for IL language**

| No. | Operators | FB Type | Reference |
|---|---|---|---|
| 4 | S1,R | SR | 2.5.2.3.1 |
| 5 | S,R1 | RS | 2.5.2.3.1 |
| 6 | CLK | R_TRIG | 2.5.2.3.2 |
| 7 | CLK | F_TRIG | 2.5.2.3.2 |
| 8 | CU,R,PV | CTU | 2.5.2.3.3 |
| 9 | CD,LD,PV | CTD | 2.5.2.3.3 |
| 10 | CU,CD,R,LD,PV | CTUD | 2.5.2.3.3 |
| 11 | IN,PT | TP | 2.5.2.3.4 |
| 12 | IN,PT | TON | 2.5.2.3.4 |
| 13 | IN,PT | TOF | 2.5.2.3.4 |