

Język programowania IL (Instruction List)

Norma IEC61131 - *Programmable Logic Controllers* definiuje następujące języki programowania:

- a) języki graficzne:
 - język drabinkowy LD (Ladder Diagram),
 - język bloków funkcyjnych FBD (Function Block Diagram),
- b) języki tekstowe:
 - lista instrukcji IL (Instruction List),
 - tekst strukturalny ST (Structured Text),
 - język grafów strukturalnych SFC (Sequential Function Chart).

Niektórzy producenci sterowników PLC korzystają z jeszcze innych języków programowania, ale nie są one zdefiniowane w normie. Dla przykładu Siemens w niektórych swoich sterownikach używa zmodyfikowanej wersji języka IL, którą nazwał STL (STatement List). O różnicach między IL a STL będzie w dalszej części tego opracowania, bo taki język też pojawia się w pytaniach egzaminacyjnych.

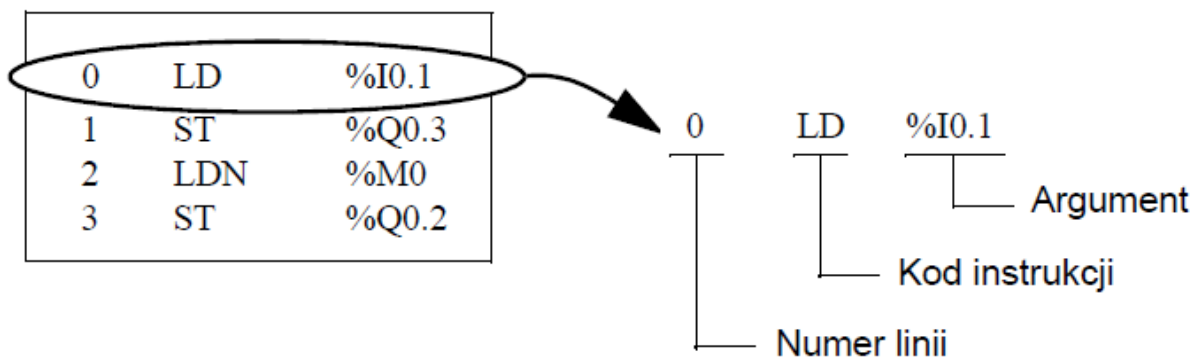
Języki graficzne FBD i LD (nazywany przez Siemens'a językiem LAD) oraz język tekstowy SFC już poznaliście. Przyszedł czas na kolejne języki tekstowe.

Język IL (Instruction List = lista rozkazów) stanowi odpowiednik języka niskiego poziomu, zbliżonego do języka maszynowego, języka typu assembler. Program napisany w tym języku jest bardzo prosty i czytelny z punktu widzenia sterownika, dlatego jego przetwarzanie i wykonywanie przez sterownik jest bardzo szybkie. Dla człowieka program w języku IL jest dość trudny i złożony w pisaniu, wymaga dużej wiedzy programistycznej i znajomości kodów poszczególnych instrukcji. Na szczęście pytania na egzaminie pisemnym dotyczą jedynie tych najprostszych programów w języku IL.

Program napisany w języku IL składa się z szeregu instrukcji wykonywanych kolejno przez sterownik. Każda instrukcja reprezentowana jest przez pojedynczą linię programu złożoną z trzech elementów:

- a) numeru linii,
- b) kodu instrukcji,
- c) argumentu.

Przykładowy program:



Numer linii wstawiany jest automatycznie podczas pisania instrukcji.

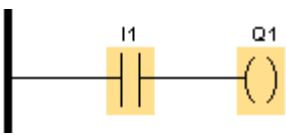
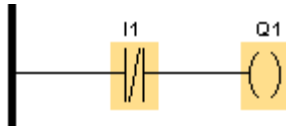
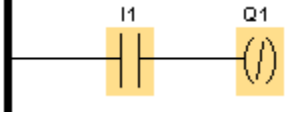
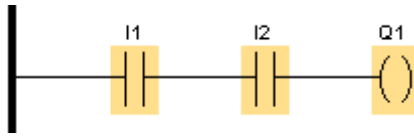
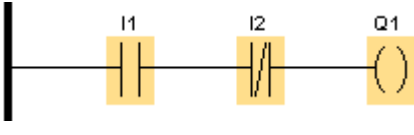
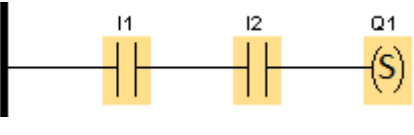
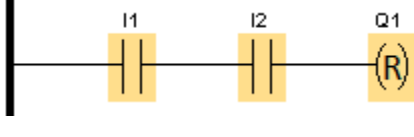
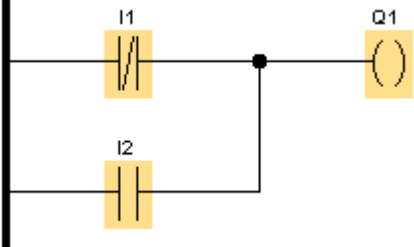
Kod instrukcji to symboliczny zapis operacji, która ma być wykonana na występującym po niej argumente.

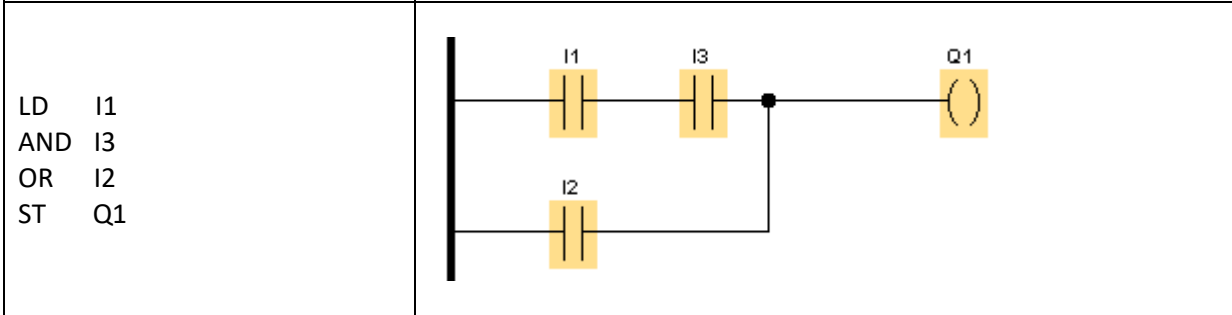
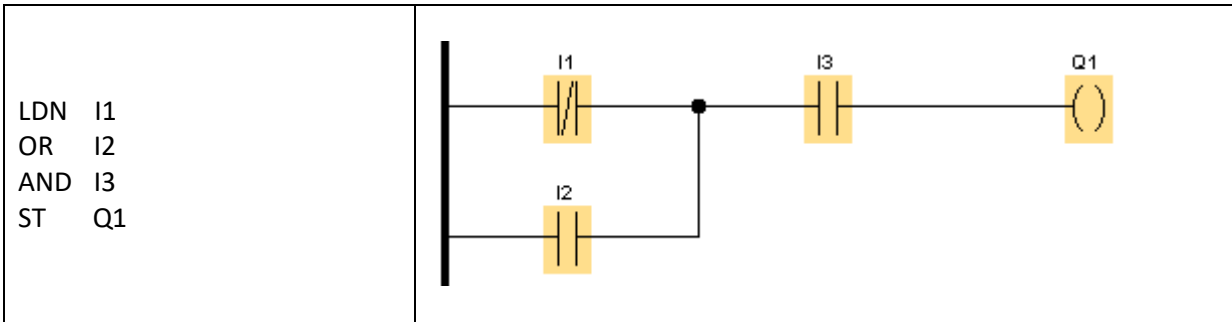
Argumentem może być stan wejścia o danym numerze (np. %I0.1 lub I1), stan wyjścia o danym numerze (np. %Q0.3 lub Q3), wewnętrzne zmienne sterownika, tzw. flagi (np. %M0 lub M0) itp.

Podstawowe kody instrukcji języka IL i ich znaczenie:

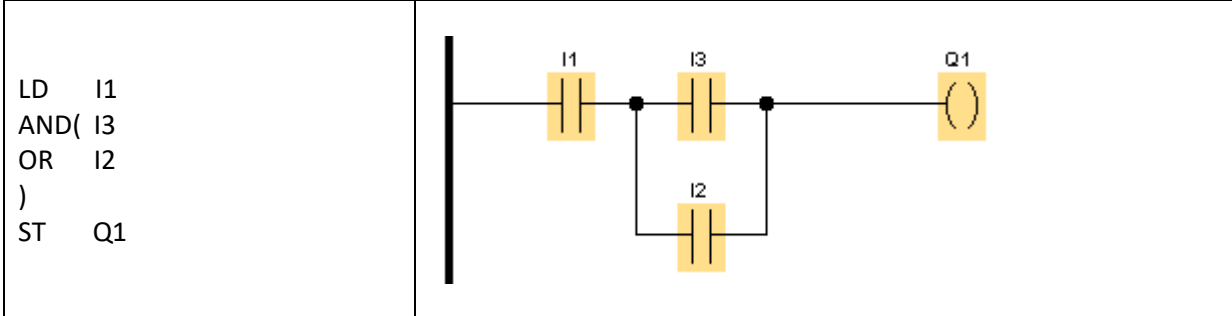
Kod instrukcji języka IL	Znaczenie kodu
LD (skrót od LOAD)	ładuje do rejestru pamięci sterownika (tzw. akumulatora) wartość odpowiadającą argumentowi skojarzonemu z tym kodem
N (skrót od NOT)	neguje wartość odpowiadającą argumentowi skojarzonemu z tym kodem
LDN (skrót od LOAD + NOT)	ładuje do rejestru pamięci sterownika (tzw. akumulatora) wartość przeciwną (zanegowaną) aktualnemu stanowi argumentu skojarzonego z tym kodem
AND	wykonuje operację logiczną AND na aktualnej wartości akumulatora i argumencie skojarzonym z tym kodem; wynik tej operacji wpisywany jest do akumulatora (zastępuje dotychczasową wartość akumulatora)
ANDN	wykonuje operację logiczną AND na aktualnej wartości akumulatora i zanegowanej wartości argumentu skojarzonego z tym kodem; wynik tej operacji wpisywany jest do akumulatora (zastępuje dotychczasową wartość akumulatora)
OR	wykonuje operację logiczną OR na aktualnej wartości akumulatora i argumencie skojarzonym z tym kodem; wynik tej operacji wpisywany jest do akumulatora (zastępuje dotychczasową wartość akumulatora)
ORN	wykonuje operację logiczną OR na aktualnej wartości akumulatora i zanegowanej wartości argumentu skojarzonego z tym kodem; wynik tej operacji wpisywany jest do akumulatora (zastępuje dotychczasową wartość akumulatora)
XOR	wykonuje operację logiczną XOR na aktualnej wartości akumulatora i argumencie skojarzonym z tym kodem; wynik tej operacji wpisywany jest do akumulatora (zastępuje dotychczasową wartość akumulatora)
XORN	wykonuje operację logiczną XOR na aktualnej wartości akumulatora i zanegowanej wartości argumentu skojarzonego z tym kodem; wynik tej operacji wpisywany jest do akumulatora (zastępuje dotychczasową wartość akumulatora)
ST (skrót od STORE)	nadaje argumentowi skojarzonemu z tym kodem aktualną wartość akumulatora
STN (skrót od STORE + NOT)	nadaje argumentowi skojarzonemu z tym kodem wartość przeciwną do aktualnej wartości akumulatora
S (skrót od SET)	nadaje argumentowi skojarzonemu z tym kodem wartość 1 pod warunkiem, że aktualna wartość akumulatora jest równa 1; stan argumentu pozostaje równy 1 aż do wystąpienia kodu R
R (skrót od RESET)	nadaje argumentowi skojarzonemu z tym kodem wartość 0 pod warunkiem, że aktualna wartość akumulatora jest równa 1; stan argumentu pozostaje równy 0 aż do wystąpienia kodu S

To tyle teorii. Teraz kilka przykładów. Dla rozjaśnienia tematu przyrównam w nich najprostsze programy w języku IL do ich odpowiedników w języku LD.

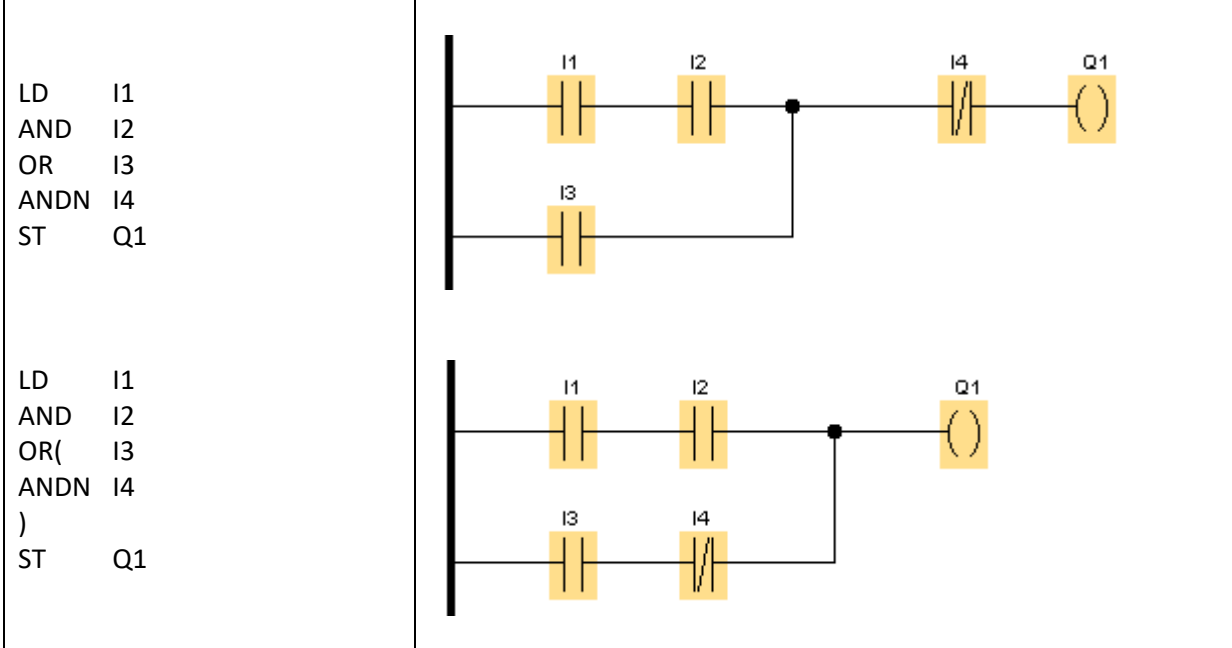
Język IL	Język LD
LD I1 ST Q1	
LDN I1 ST Q1	
LD I1 STN Q1	
LD I1 AND I2 ST Q1	
LD I1 ANDN I2 ST Q1	
LD I1 AND I2 S Q1	
LD I1 AND I2 R Q1	
LDN I1 OR I2 ST Q1	



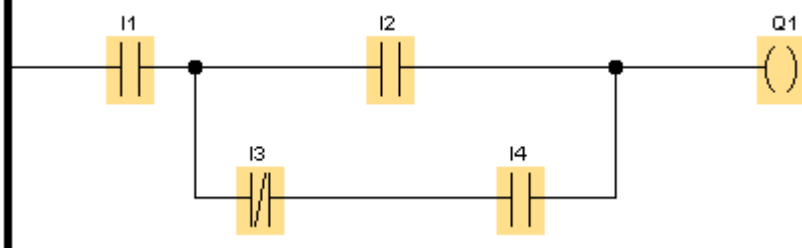
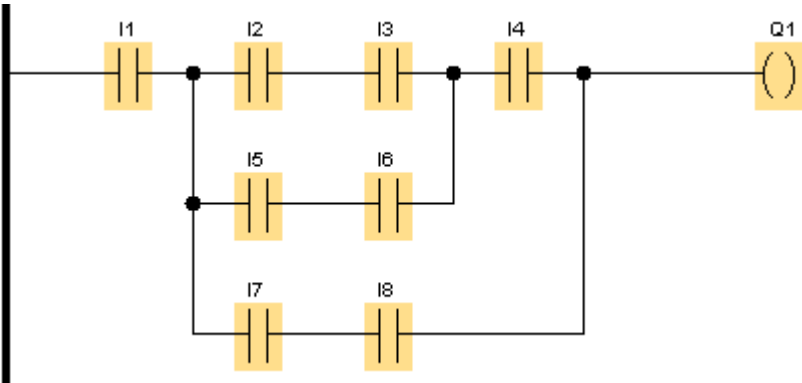
Sterownik wykonuje instrukcje w takiej kolejności, w jakiej są one zapisane w programie. Kolejność wykonywania działań logicznych w programie można zmienić za pomocą nawiasów. Wstawienie nawiasu do poprzedniego programu IL daje skutek widoczny poniżej:



Inny przykład ukazujący efekt zastosowania nawiasu w początkowo tym samym programie:



Zabawa zaczyna się, gdy nawiasów jest więcej. Mówi się wtedy o zagnieżdżaniu nawiasów:

<pre>LD I1 AND(I2 OR(N I3 AND I4)) ST Q1</pre>	
<pre>LD I1 AND(I2 AND I3 OR(I5 AND I6) AND I4 OR(I7 AND I8)) ST Q1</pre>	

Na koniec jeszcze o wspomnianych na początku różnicach między językiem IL a STL. W pytaniach egzaminacyjnych często nazwy te występują zamiennie, ale niektóre kody instrukcji w języku STL są jednak nieco inne (krótsze) niż w języku IL. Poniżej zestawienie tych różnic.

Kod instrukcji języka IL	Kod instrukcji języka STL
AND	A
ANDN	AN
OR	O
ORN	ON
ST	=
Ponadto kod instrukcji LD może być zastępowany w programach języka STL kodem instrukcji, która występuje w następnej linii programu. Kod negacji (jeżeli występował) pozostaje bez zmian:	
<pre>LD I1 AND I2 ST Q1</pre>	<pre>A I1 A I2 = Q1</pre>
<pre>LDN I1 OR I2 ST Q1</pre>	<pre>ON I1 O I2 = Q1</pre>