Programowanie PLC w językach LD i IL.

Czym jest program?

Program jest serią połączonych ze sobą instrukcji zrozumiałych dla PLC. Aby stworzyć program należy napisać go w jednym z dostępnych języków programowania a następnie skompilować (zamienić źródło programu zrozumiałe dla projektanta na program wykonywalny zrozumiały dla PLC) i wysłać go do pamięci sterownika.

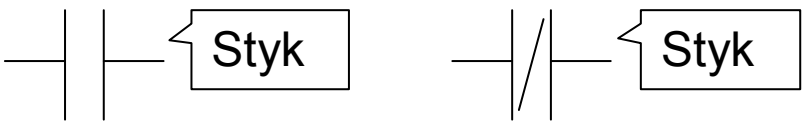
Oznaczenia podstawowych elementów w programowaniu PLC.

Istnieje sześć podstawowych elementów, którymi będziemy posługiwać się w programowaniu:

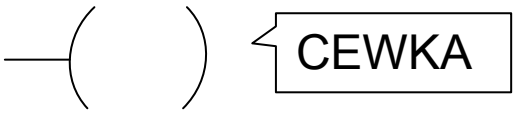
Nazwa elementu	Oznaczenie
Wejścia	I lub X
Wyjścia	Q lub Y
Przekaźniki czasowe – TIMER	T
Liczniki	C
Wewnętrzne znaczniki binarne – Pamięć, flagi	M, S

Podstawy języka LD i IL.

LD – język drabinkowy jest analogiczny do układów zaszytych sterowania (zbudowanych na stycznikach i okablowaniu). Występują w nim dwa rodzaje styków NO (normal open) normalnie otwarte i NC (normal closed) normalnie zamknięte. Na schematach przedstawione są w stanie nieaktywnym. Reprezentują one urządzenia wejściowe do PLC.



Wyjścia z PLC reprezentowane są przez cewki.(w programie ta sama cewka nie może być wysterowana dwa razy. Może to spowodować nieprzewidywalność działania programu.)



Przykład podwójnego wysterowania cewki:

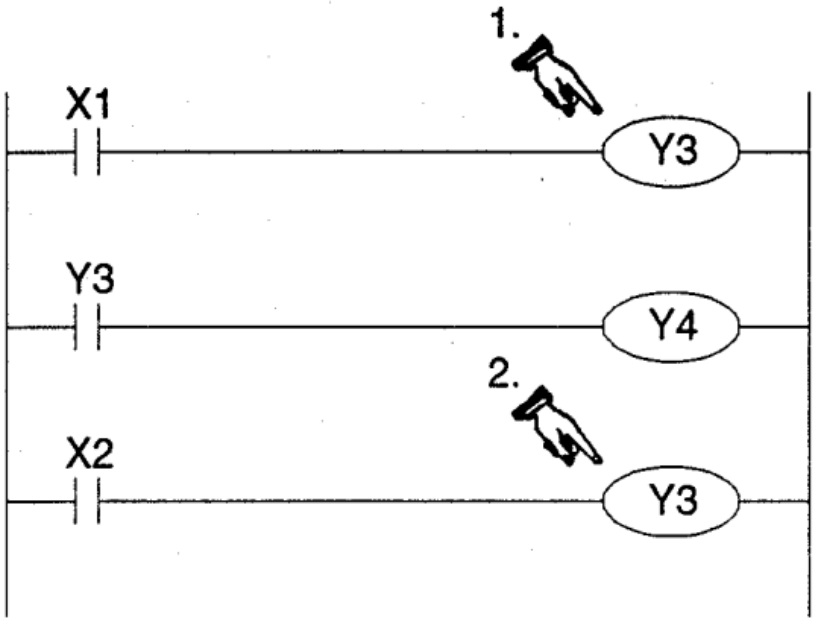
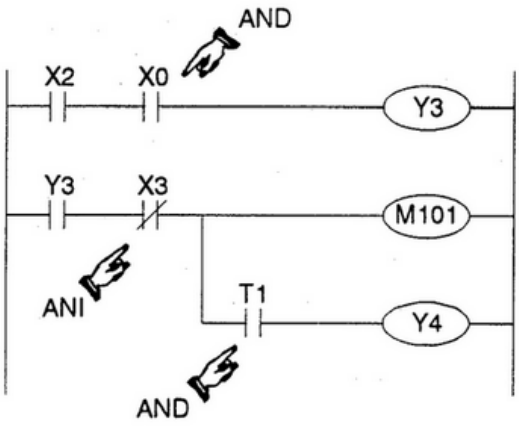


Diagram języka drabinkowego czytamy od lewej do prawej i od góry w dół. Lewą linię diagramu nazywamy lewą linią zasilania, prawą linię nazywamy prawą linią zasilania (w wielu aplikacjach jest ona pomijana).

Podstawowe instrukcje języka LD i IL:

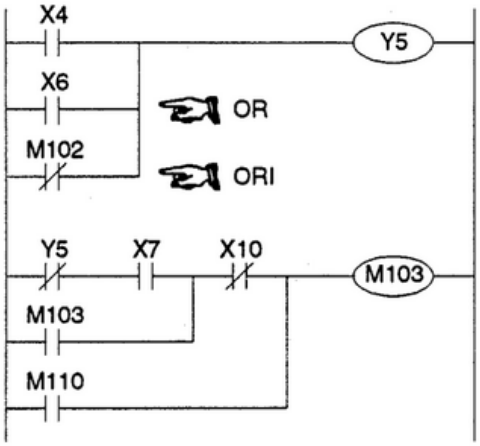
Przykład programu:




```
0  LD   X  2
1  AND  X  0
2  OUT  Y  3
3  LD   Y  3
4  ANI  X  3
5  OUT  M 101
6  AND  T  1
7  OUT  Y  4
```

1. Instrukcja **LD** odnosi się do rozpoczęcia nowej sieci stykiem NO w języku drabinkowym.
2. Instrukcja **LDI** odnosi się do rozpoczęcia nowej sieci stykiem NC w języku drabinkowym.
3. **AND** opisuje użycie funkcji logicznej AND dla styku NO
4. **ANI** opisuje użycie funkcji logicznej AND INVERS dla styku NC
5. **OUT** dotyczy wyjścia na cewkę

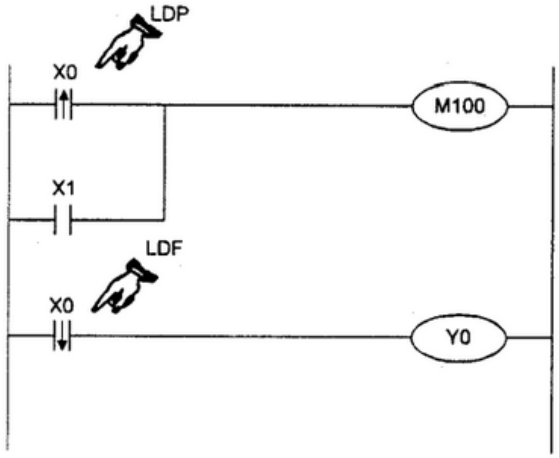
Przykład programu:




```
0   LD    X   4
1   OR    X   6
2   ORI   M   102
3   OUT   Y   5
4   LDI   Y   5
5   AND   X   7
6   OR    M   103
7   ANI   X   10
8   OR    M   110
9   OUT   M   103
```

6. **OR** opisuje użycie funkcji logicznej OR dla styku NO
7. **ORI** opisuje użycie funkcji logicznej OR INVERS dla styku NC

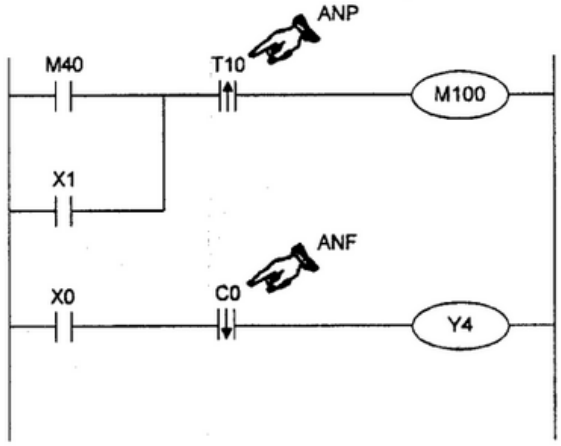
Przykład programu:




```
0   LDP   X   0
2   OR    X   1
3   OUT   M   100
4   LDF   X   0
6   OUT   Y   0
```

8. **LDP** instrukcja operacji logicznej przy narastającym zboczu impulsu
9. **LDF** instrukcja operacji logicznej przy opadającym zboczu impulsu

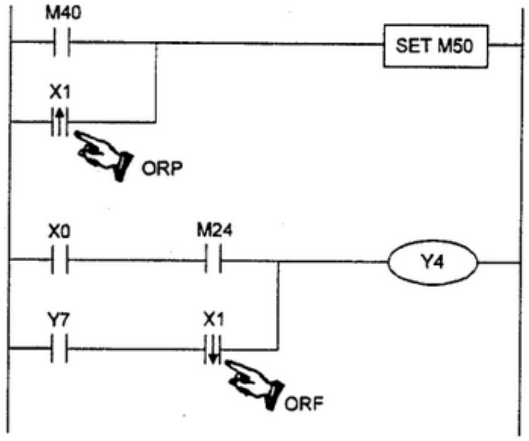
Przykład programu:



0	LD	M	40
1	OR	X	1
2	ANP	T	10
4	OUT	M	100
5	LDF	X	0
6	ANF	C	0
8	OUT	Y	4

10. **ANP, ANF** funkcja AND przy narastającym lub opadającym zboczu sygnału

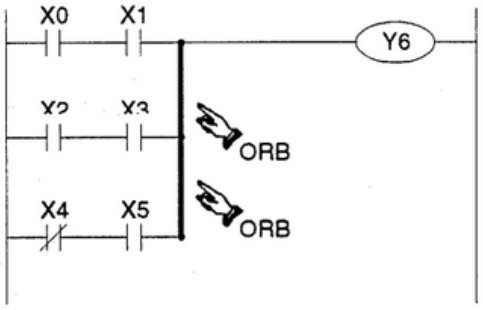
Przykład programu:



0	LD	M	40
1	ORP	X	1
3	SET	M	50
4	LD	X	0
5	AND	M	24
6	LD	Y	7
7	ORF	X	1
9	ORB		
10	OUT	Y	4

11. **ORP, ORF** funkcja OR przy narastającym lub opadającym zboczu sygnału

Przykład programu:



Zalecana metoda programowania sekwencyjnego

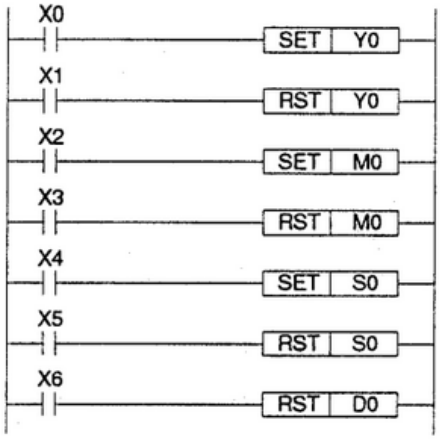
0	LD	X	0
1	AND	X	1
2	LD	X	2
3	AND	X	3
4	ORB		
5	LDI	X	4
6	AND	X	5
7	ORB		
8	OUT	Y	6

Niezalecana metoda programowania wsadowego

0	LD	X	0
1	AND	X	1
2	LD	X	2
3	AND	X	3
4	LDI	X	4
5	AND	X	5
6	ORB		
7	ORB		
8	OUT	Y	6

12. **ORB** funkcja łączenia szczebli w diagramie drabinkowym.

Przykład programu:




```
0   LD   X 0
1   SET  Y 0
2   LD   X 1
3   RST  Y 0
4   LD   X 2
5   SET  M 0
6   LD   X 3
7   RST  M 0
8   LD   X 4
9   SET  S 0
10  LD   X 5
11  RST  S 0
12  LD   X 6
13  RST  D 0
```

13. **SET** ustawia stan 1 na elemencie programu
14. **RST** ustawia stan 0 na elemencie programu