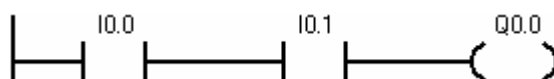


## LABORATORIUM – ELEKTRONIKA

### Budowa układów kombinacyjnych na sterowniku PLC w języku drabinkowym

#### 1. Wstęp

Na sterowniku PLC (Programmable Logic Controller) można realizować m.in. układy kombinacyjne. Podstawowym językiem dla układów PLC jest język drabinkowy, wykorzystujący logikę stykową. Podstawowe działania logiczne realizuje się na stykach w sposób następujący:



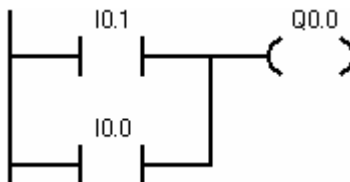
Rysunek 1. AND (I – iloczyn logiczny) na stykach:  $Q0.0 = I0.0 \cdot I0.1$

Rysunek 1 najprościej wytłumaczyć tak:

- o szyna po lewej to zasilanie, z niej „bierzemy” prąd,
- o skoro tak, to aby wyjście Q0.0 „dostało” prąd, muszą się zewrzeć styki I0.0 i I0.1.

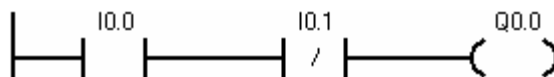
Jest to więc realizacja logicznej funkcji I, czyli AND.

Analogicznie można na stykach zrealizować funkcję LUB, czyli OR. Jeżeli styki zostały ułożone jak na Rysunku 2, to aby wyjście Q1 „dostało” prąd, wystarczy że zewrze się styk I0.0 lub I0.1:



Rysunek 2. OR (LUB – suma logiczna) na stykach:  $Q0.0 = I0.0 + I0.1$

A co z negacją? Styki pokazane wyżej to tzw. „styki zwierne”. Zwierają się one, gdy reprezentowana przez nie zmienna jest równa 1, gdy jest ona równa zero, rozwierają się. Istnieje więc również „styk rozwierny”, działający odwrotnie, więc reprezentujący negację zmiennej którą symbolizuje. Spójrzmy więc na rysunek poniżej.



Rysunek 3. Styk zwierny, reprezentujący wejście wprost: I0.0, oraz styk rozwierny czyli wejście zanegowane:  $\sim I0.1$

W układzie z rysunku powyżej realizowana jest funkcja  $Q0.0 = I0.0 \cdot \sim I0.1$ . Wyjście Q0.0 zadziała wtedy kiedy wejście I0.0 będzie zwarte, natomiast I0.1 rozwarte.

Dysponując powyższą wiedzą można zrealizować każdą funkcję logiczną, bądź to opisaną słownie (jak pokazano wyżej), bądź zadaną w postaci tabeli (niżej).

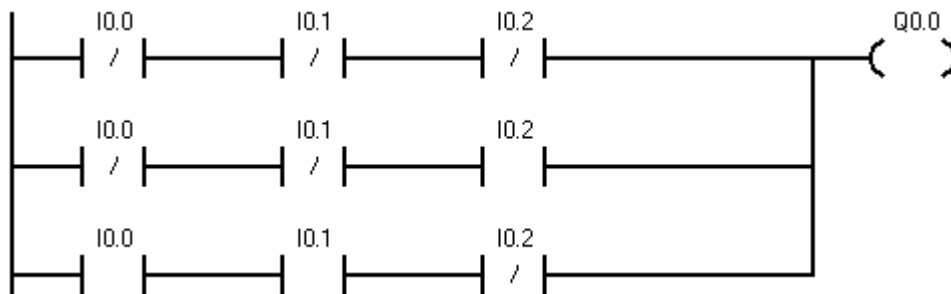
Popatrzmy więc na układ opisany następującą tabelą stanów (wejścia: A, B, C, wyjście OUT):

l.p.	A	B	C	OUT
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	x
4	1	0	0	x
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Na podstawie tej tabeli stanów uzyskujemy opis kanoniczny w zapisie Boola:

$$OUT = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} ,$$

z którego możemy bezpośrednio uzyskać realizację zadanej tą tabelą funkcji logicznej na sterowniku PLC:

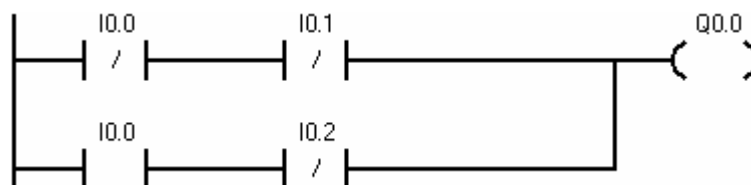


gdzie oczywiście: A=I0.0, B=I0.1, C=I0.2 oraz OUT=Q0.0.

Układ ten oczywiście można zminimalizować, metodą tablic Karnaugh (jak niżej), lub inną:

A	0	1
BC		
00	1	X
01	1	0
11	X	0
10	0	1

uzyskując opis kanoniczny:  $OUT = \bar{A} \cdot \bar{B} + A \cdot \bar{C}$  i jego realizację:



gdzie podobnie jak poprzednio: A=I0.0, B=I0.1, C=I0.2 oraz OUT=Q0.0.

## 2. Jak zaprogramować sterownik S7-200 używając programu STEP 7-Micro/WIN?

2.1. Uruchomić program **STEP 7-Micro/WIN**.

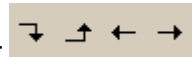
2.2. W oknie *Simatic LAD* zbudować projekt.

2.2.1. Po lewej stronie okna programu dostępne są elementy możliwe do wykorzystania, np. styki zwierny i rozwierny są w katalogu *Bit Logic*.

2.2.2. Elementy wstawia się poprzez ich przeciągnięcie do sieci (*Network*).

2.2.3. Styki należy nazywać zgodnie z opisem na sterowniku (widoczne na obudowie), więc np. jeżeli wejściem ma być styk pierwszy od lewej należy domyślnie wstawiony nad stykiem opis „???” zastąpić opisem „I0.0”. Podobnie jest z wyjściami.

2.2.4. Połączenia drabinki tworzą się w wierszach styków automatycznie, natomiast połączenia między wierszami najłatwiej tworzyć korzystając z przycisków nad oknem drabinki:



2.3. Gdy projekt jest gotowy należy go skompilować: *PLC->Compile* lub przycisk

2.3.1. Jeśli kompilacja była bezbłędna można projekt załadować do sterownika. W tym celu:

2.3.2. Zasilany i podłączony do komputera kablem PPI sterownik ustawić w **tryb STOP** (przełącznik jest pod klapką widoczną z przodu sterownika).

2.3.3. Przesłać program do sterownika: *File->Download* lub przycisk

2.4. Gdy program zostanie przesłany do sterownika (zależnie od wielkości programu zajmie to kilka lub więcej sekund) można przełączyć sterownik w **tryb RUN** i... powinien on już działać, oby zgodnie z projektem.

## 3. Przebieg ćwiczenia

3.1. Należy zbudować w języku drabinkowym i sprawdzić działanie układu zadanego przez prowadzącego\*.

\*Ćwiczenie jest zaliczone bez sprawozdania – na podstawie działającego układu.

## 4. Literatura

[1] Bryan L.A., Bryan E.A., „Programmable Controllers Theory And Implementation”, ITC, Atlanta 1997

[2] Broel-Plater B., "Układy wykorzystujące sterowniki PLC : projektowanie algorytmów sterowania", PWN, Warszawa 2009

[3] Chochowski A., Cieślak H., Laskowski P., Mirski T., „Laboratorium automatyki”, Wydawnictwo SGGW, Warszawa 1999

[4] Flaga S., „Programowanie sterowników PLC w języku drabinkowym”, Wydawnictwo btc, Legionowo 2010

[5] Król A., Moczko-Król J., "S5/S7 Windows : programowanie i symulacja sterowników PLC firmy Siemens", Nakom, Poznań 2003

[6] Kwaśniewski J., „Sterowniki PLC w praktyce inżynierskiej”, Wydawnictwo btc, Legionowo 2008

[7] Mikulczyński T., "Automatyzacja procesów produkcyjnych: metody modelowania procesów dyskretnych i programowania sterowników PLC", WN-T, Warszawa 2006

[8] Nowakowski W., „LOGO! w praktyce”, Wydawnictwo btc, Warszawa 2006

[9] Sałat R., Korzysz K., Obstawski P., "Wstęp do programowania sterowników PLC", WKiŁ, Warszawa 2010