
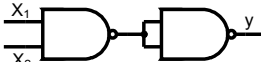
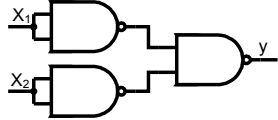

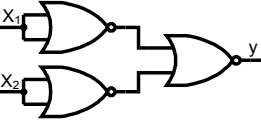
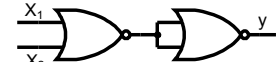


# Realizacja funkcji przełączających

## 1. Wprowadzenie teoretyczne

### 1.1. Podstawowe funkcje logiczne

Funkcja logiczna	NOT	AND	OR
Zapis	$y = \overline{x_1}$	$y = x_1 \cdot x_2$	$y = x_1 + x_2$
NAND			
NOR			

### 1.2. Metoda minimalizacji funkcji metodą tablic Karnaugh

Metoda tablicy Karnaugh należy do grupy najszybszych metod minimalizacji funkcji przełączających małej liczby zmiennych, co wynika z dużej komplikacji samego zapisu następującej wraz ze wzrostem ilości zmiennych.

Upraszczając funkcję przełączającą przy wykorzystaniu tablicy Karnaugh, należy pamiętać o następujących problemach:

- wiersze i kolumny tablicy Karnaugh opisane są w kodzie Greya, tzn. każdy kolejny wiersz i kolumna różnią się od siebie o negację jednej zmiennej,
- zakreślając jedyne (zera), tworzy się grupy liczące 2, 4, 8, 16 ... elementów,
- zawsze zakreśla się grupy z największą możliwą ilością jedynek (zer), przy czym należy pamiętać o możliwości sklejania ze sobą krawędzi równoległych tablicy,
- grupy mogą posiadać części wspólne,
- liczba grup jedynek (zer) odpowiada liczbie składników sumy (iloczynu) poszukiwanej funkcji,
- w przypadku kiedy istnieje możliwość zakreślenia grup na kilka sposobów, arbitralnie wybiera się jeden z nich,
- dana grupa reprezentuje iloczyn (sumę) tych zmiennych, które nie zmieniają swojej wartości,
- w przypadku gdy funkcja przełączająca posiada elementy o wartości nieokreślonej elementy te wpisujemy do tabeli wprowadzając dla nich specjalne oznaczenie np. – a następnie wykorzystujemy lub pomijamy w zależności od potrzeby przy tworzeniu grup (patrz punkt b).

### **1.3. Program LabVIEW**

LabVIEW (**L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench) umożliwia tworzenie programów za pomocą języka graficznego (tzw. język G). Programowanie w LabVIEW polega na budowie schematu blokowego i korespondującego z nim panelu stanowiącego interfejs użytkownika. Budowa tego interfejsu jest możliwa dzięki dostępnym bibliotekom gotowych elementów takich, jak: wyświetlacze cyfrowe, mierniki, potencjometry, termometry, diody LED, tabele, wykresy itp. Elementy te konfiguruje się w zależności od zastosowania. Panel użytkownika umożliwia zbudowanie wirtualnego przyrządu obsługiwanego: z klawiatury, za pomocą myszy lub innego urządzenia wejściowego służącego do komunikacji komputera z użytkownikiem.

Następnie, przy pomocy graficznego języka konstruuje się odpowiedni schemat blokowy, będący równocześnie kodem źródłowym. Budowany schemat blokowy można porównać z grafem przepływu informacji, a jego elementy to funkcje zawarte w bibliotekach, np. algebraiczne, boolowskie, statystyczne, związane z obsługą plików, przetwarzaniem sygnałów lub obsługą urządzeń we/wy itp. Relacje między blokami funkcyjnymi reprezentowane są przez połączenia o różnych kolorach i grubościach. Rodzaj połączenia świadczy o typie przekazywanych danych. Można łączyć ze sobą tylko elementy tego samego typu. Tworzone aplikacje nazywane są virtual instruments (VI), ponieważ ich wygląd i operacje imitują działanie rzeczywistych przyrządów. Program zawiera wszystkie narzędzia niezbędne do akwizycji, analizy i prezentacji danych.

Wszystkie aplikacje używają struktury hierarchicznej i modularnej. Oznacza to, że można ich używać również jako podprogramy. Aplikacje użyte w innej aplikacji nazywane są subVI.

## **2. Przebieg ćwiczenia**

### Przykład 1.

Zaprojektować układ o trzech wejściach  $x_1$ ,  $x_2$  i  $x_3$ , w którym sygnał wyjściowy  $y = 1$  gdy na wejściu pojawi się liczba w naturalnym kodzie binarnym podzielna przez trzy lub nieparzysta. Wyznaczyć postać minimalną funkcji  $y = f(x_1, x_2, x_3)$  oraz przedstawić schemat logiczny tego układu z zastosowaniem bramek NAND i NOR.

Działanie układu opisuje poniższa tabela stanów:

Liczba wejściowa	Wejścia			Wyjście
	$x_1$	$x_2$	$x_3$	$y$
0	0	0	0	0
1	1	0	0	1
2	0	1	0	0
3	1	1	0	1
4	0	0	1	0
5	1	0	1	1
6	0	1	1	1
7	1	1	1	1

Na podstawie tabeli można napisać równanie funkcji  $y$  w kanonicznej postaci alternatywnej

$$y = x_1\bar{x}_2\bar{x}_3 + x_1x_2\bar{x}_3 + x_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1x_2x_3$$

lub w kanonicznej postaci koniunkcyjnej

$$y = (x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(x_1 + x_2 + \bar{x}_3)$$

Minimalizacji funkcji  $y$  dokonujemy za pomocą tablic Karnaugh

$x_1x_2$	00	01	11	10
$x_3$				
0	0	0	1	1
1	0	1	1	1

Postać alternatywna

$x_1x_2$	00	01	11	10
$x_3$				
0	0	0	1	1
1	0	1	1	1

Postać koniunkcyjna

Właściwą minimalizację przeprowadzamy sklejając jedynki (dla postaci alternatywnej) lub zera (dla postaci koniunkcyjnej) otrzymując

$$y = x_1 + x_2x_3$$

$$y = (x_1 + x_3)(x_1 + x_2)$$

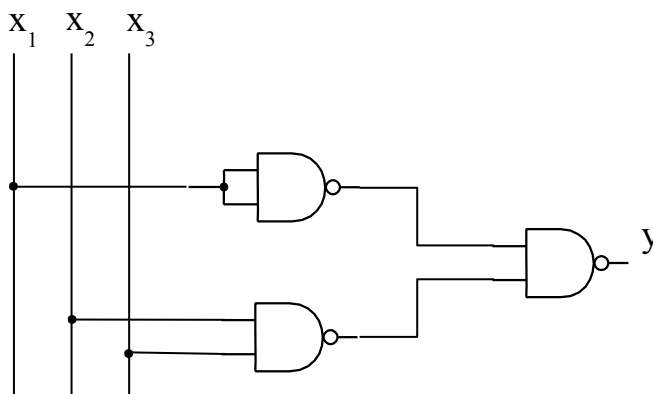
Stosując prawa rozdzielności i pochłaniania, przekształcając  $y = (x_1 + x_3)(x_1 + x_2)$ , otrzymamy

$$y = (x_1 + x_3)(x_1 + x_2) = x_1x_1 + x_1x_2 + x_3x_1 + x_3x_2 = x_1(x_1 + x_2 + x_3) + x_3x_2 = x_1 + x_3x_2$$

Stąd wniosek, że postać koniunkcyjna jest równoważna postaci alternatywnej.

Schemat logiczny układu z zastosowaniem bramek NAND przedstawia rys. 1.

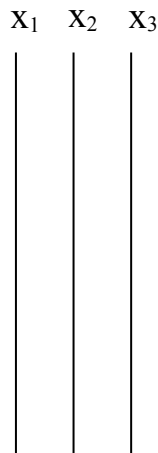
$$y = x_1 + x_2x_3 = \overline{\overline{x_1 + x_2x_3}} = \overline{\overline{x_1} \cdot \overline{x_2x_3}}$$



Rys. 1. Schemat logiczny z bramek NAND

Schemat logiczny układu z zastosowaniem bramek NOR przedstawia rys. 2.

$$y = (x_1 + x_3)(x_1 + x_2) = \overline{\overline{(x_1 + x_3)} \overline{(x_1 + x_2)}} = \overline{\overline{(x_1 + x_3)} + \overline{(x_1 + x_2)}}$$



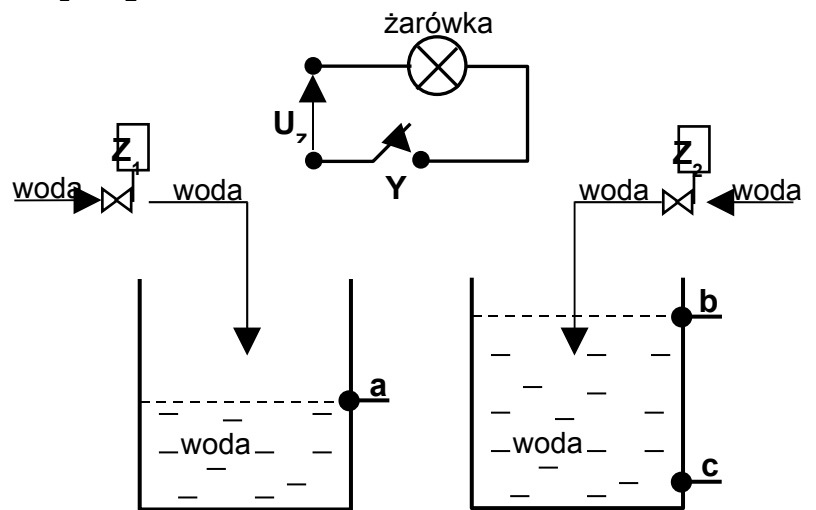
Rys. 2. Schemat logiczny z bramek NOR

### Przykład 2.

Zaprojektować układ sterowania dopływem wody do dwóch zbiorników (rysunek 3). Poziom wody w zbiornikach kontrolowany jest czujnikami **a**, **b**, **c** ( $a = 0$  gdy poziom wody jest poniżej czujnika **a**, natomiast  $a = 1$  gdy poziom wody jest powyżej czujnika **a**, itp. dla pozostałych czujników). Dopływem wody sterują zawory elektromagnetyczne **Z<sub>1</sub>** i **Z<sub>2</sub>**.

Program pracy układu:

- zawór **Z<sub>1</sub>** powinien być otwarty ( $Z_1 = 1$ ) stale, gdy zbiornik jest niepełny ( $a = 0$ ),
- zawór **Z<sub>2</sub>** powinien być otwarty, gdy poziom wody w zbiorniku nie osiągnął poziomu czujnika **c**. Po jego przekroczeniu zawór zamyka się i otwiera się dopiero wtedy, gdy poziom wody w drugim zbiorniku osiągnął poziom czujnika **a**,
- zawór **Z<sub>2</sub>** powinien być zamknięty, gdy zbiornik napełni się ( $b = 1$ ),



Rys. 3. Układ zbiorników z wodą

Napełnienie zbiorników powinno być sygnalizowane mignięciem żarówki. Żarówka jest włączana w obwód „przełącznikiem impulsowym” **Y**. (chwilowe zamknięcie obwodu następuje, gdy  $Y = 1$ ).

Przedstawić **schemat logiczny** tego układu z zastosowaniem bramek logicznych NOR.

### 3. Projekty do wykonania

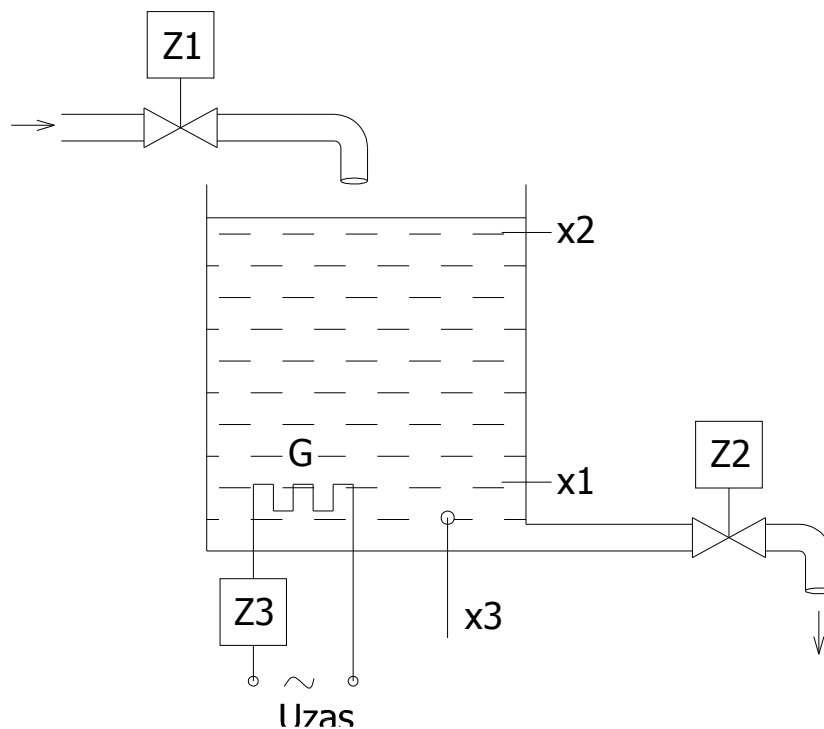
#### Projekt 1.

Zaprojektować układ sterowania pracą podgrzewacza wody. Poziom wody kontrolowany jest czujnikami  $X_1$  oraz  $X_2$  ( $X_i=0$ , gdy poziom wody jest poniżej  $X_i$ , natomiast  $X_i=1$  gdy poziom wody jest powyżej  $X_i$ ,  $i=1,2$ ) a temperatura wody w podgrzewaczu czujnikiem  $X_3$  ( $X_3=0$  gdy  $T_w < T_G$  natomiast  $T_w > T_G$ ,  $T_w$  – temp. wody,  $T_G$  – temp. grzałki). Dopływ i odpływ wody uzależnione są od stanu zaworów elektromagnetycznych  $Z_1$ ,  $Z_2$ . Zbiornik ogrzewany jest grzałką  $G$  włączaną do sieci za pomocą stycznika  $Z_3$ .

Program pracy podgrzewacza jest następujący:

- 1) zawór  $Z_1$  powinien być otwarty ( $Z_1=1$ ) stale, jeżeli zbiornik jest niepełny ( $X_2=0$ )
- 2) zawór  $Z_2$  powinien być otwarty gdy temperatura wody w podgrzewaczu  $T_w > T_G$  i poziom wody przekracza  $X_1$
- 3) grzałka  $G$  powinna być załączona, gdy temperatura wody  $T_w < T_G$  i poziom wody przekracza  $X_1$

Przedstawić schemat logiczny tego układu z zastosowaniem dwuwejściowych bramek NOR.

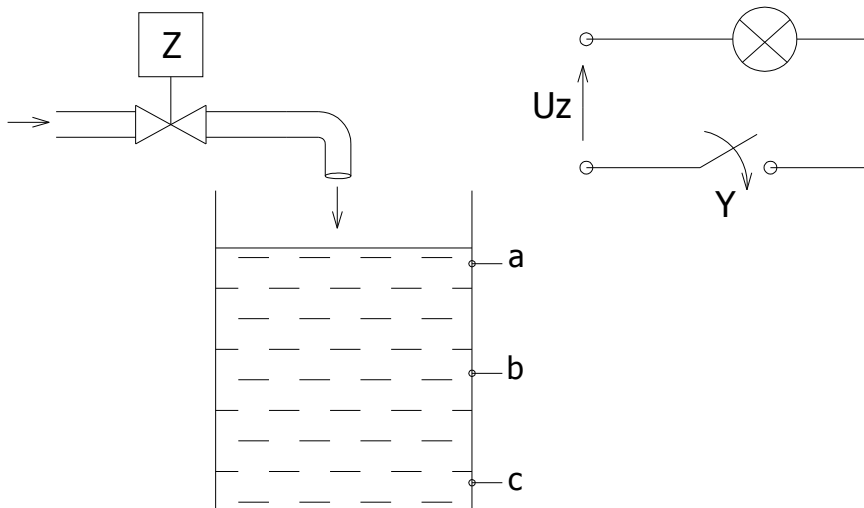


## Projekt 2.

Zaprojektować układ sterowania dopływem wody do zbiornika. Poziom wody kontrolowany jest czujnikami **a**, **b**, **c** ( $a=0$  gdy poziom wody jest poniżej **a**, natomiast  $a=1$  gdy poziom wody jest powyżej **a**, itp. dla pozostałych czujników). Dopływem wody steruje zawór elektromagnetyczny **Z**. Określony poziom wody w zbiorniku jest sygnalizowany mignięciem żarówki.

Program pracy układu:

- 1) zawór **Z** powinien być otwarty ( $Z=1$ ) stale, gdy zbiornik jest niepełny ( $a=0$ )
- 2) osiągnięcie kolejnych poziomów **c**, **b**, **a** powinno być sygnalizowane mignięciem żarówki. Żarówkę włącza w obwód „przełącznik impulsowy” **Y** (chwilowe zamknięcie obwodu następuje, gdy  $Y=1$ )



---

Zaprojektować układ sterowania dopływem wody do 3 jednakowych zbiorników. Trzy czujniki (oznaczone odpowiednio **a**, **b**, **c**) podają informacje o poziomach cieczy w trzech zbiornikach. Dopływem wody w zbiorniku steruje jeden zawór **Z**.

Program pracy układu jest następujący:

- 1) zawór **Z** powinien być otwarty stale jeżeli zbiorniki są niepełne
- 2) poszczególne zawory powinny być otwarte stale jeżeli akurat poziom wody w danym zbiorniku ( $Z=1$ ) nie został osiągnięty ( $a, b, c=0$ )
- 3) osiągnięcie określonych poziomów we wszystkich zbiornikach powinno być zasygnalizowane dźwiękiem generowanym przez dzwonek **D**

