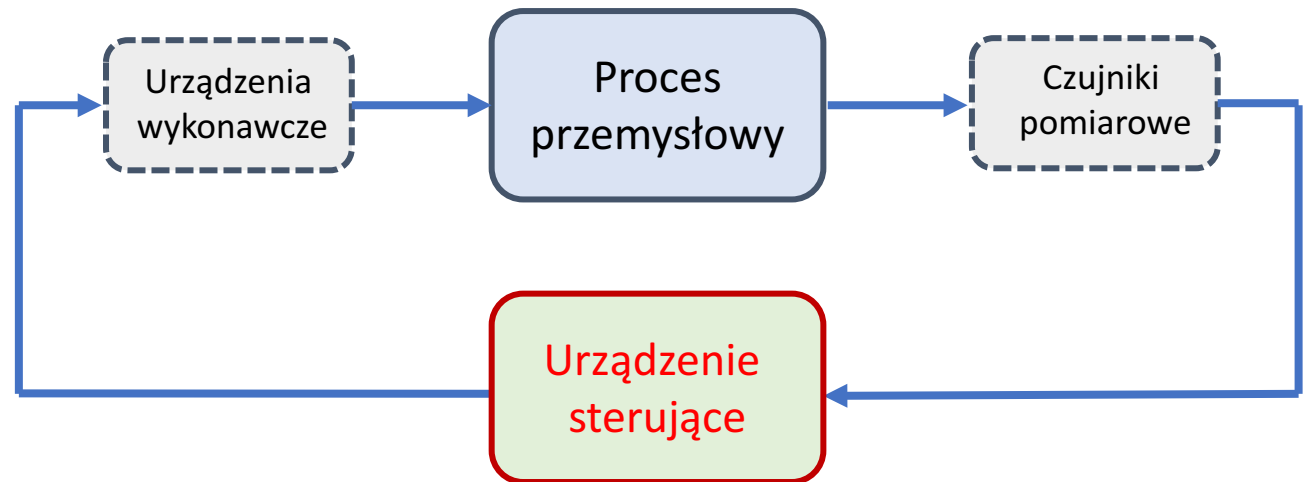
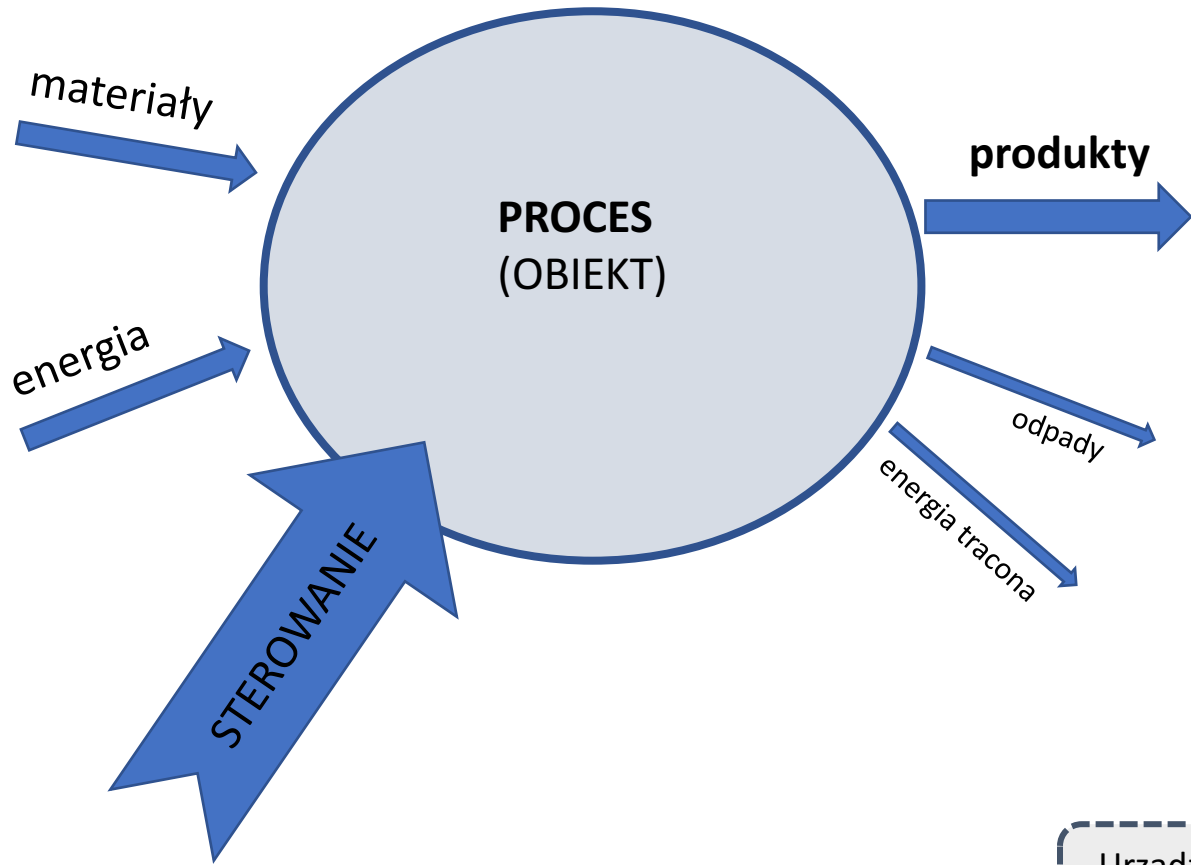


## STEROWANIE MASZYN I URZĄDZEŃ I

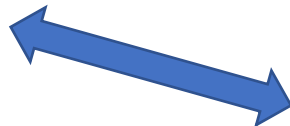
- **Materiały pomocnicze do wykładu**
- **Materiały pomocnicze do laboratorium  
(przykładowe programy sterujące na PLC)**



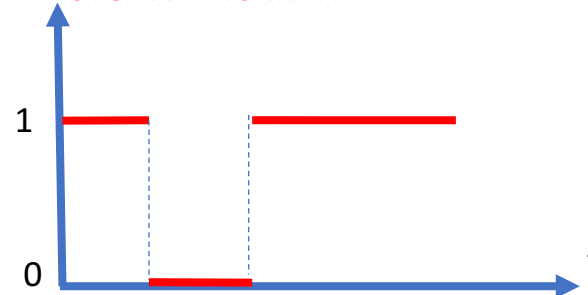
# UKŁAD STEROWANIA

CIĄGŁY

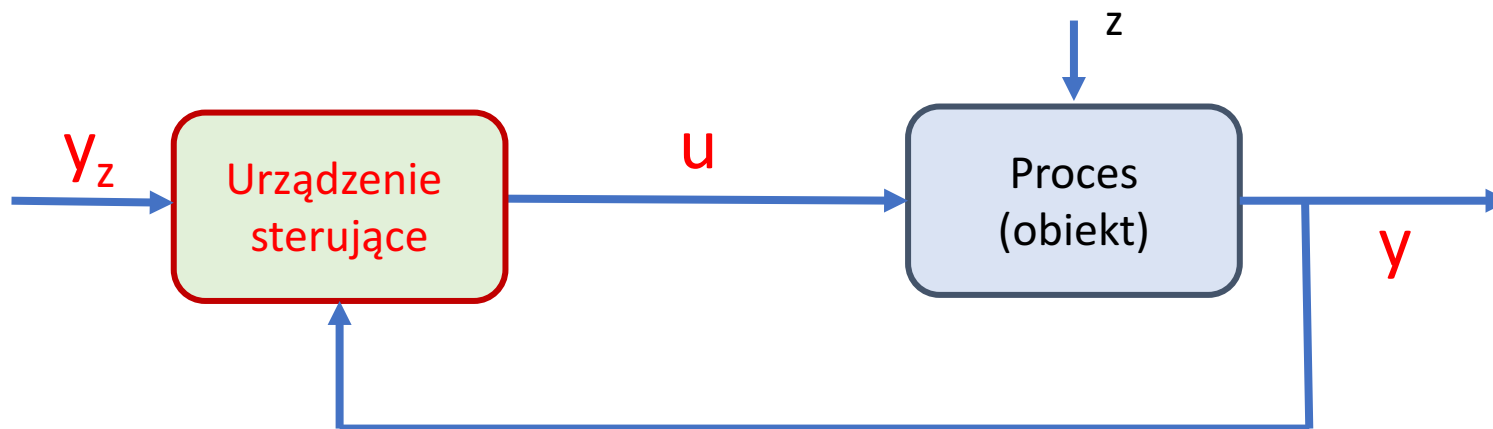
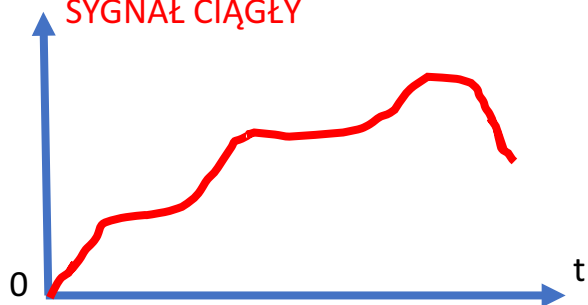
DYSKRETNY



SYGNAŁ DYSKRETNY



SYGNAŁ CIĄGŁY

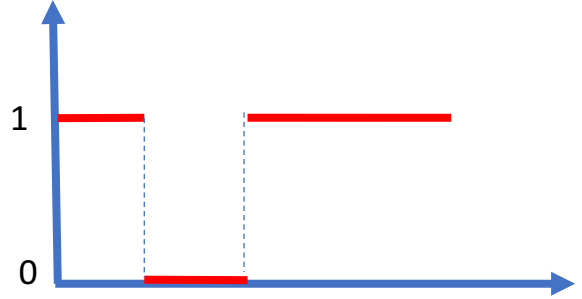


# UKŁAD STEROWANIA

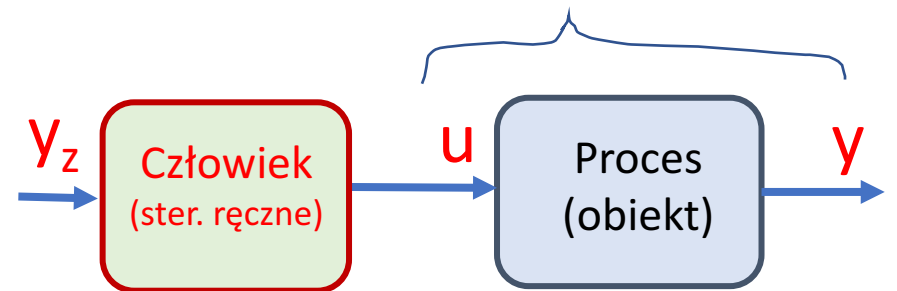
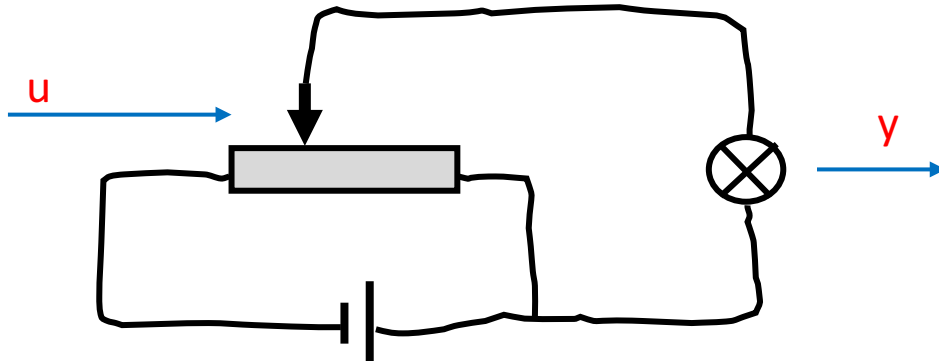
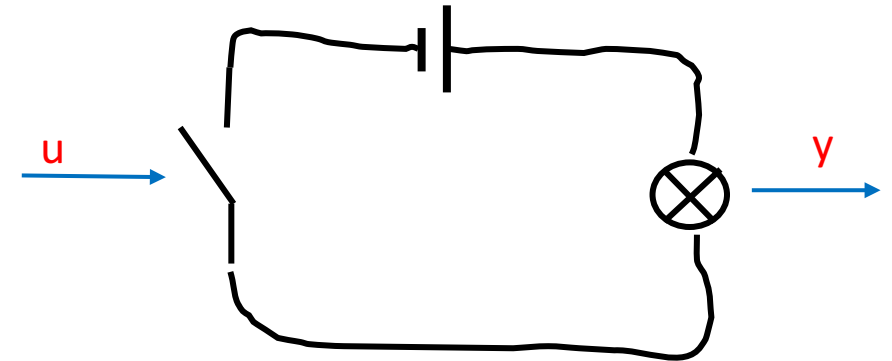
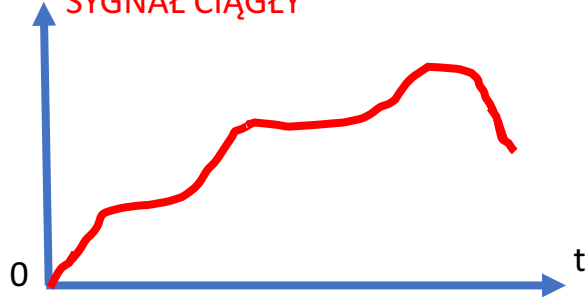
CIĄGŁY

DYSKRETNY

SYGNAŁ DYSKRETNY



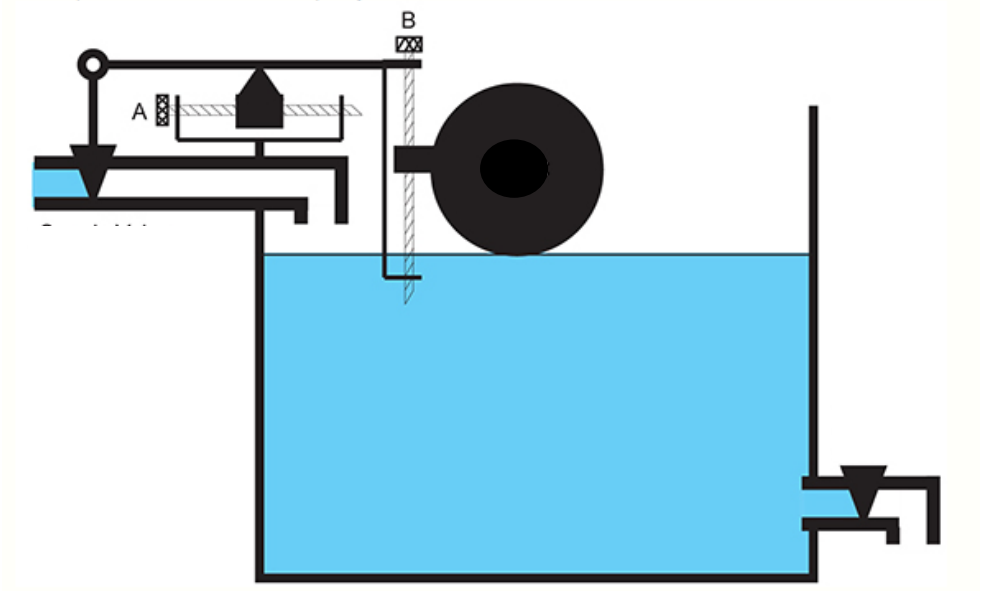
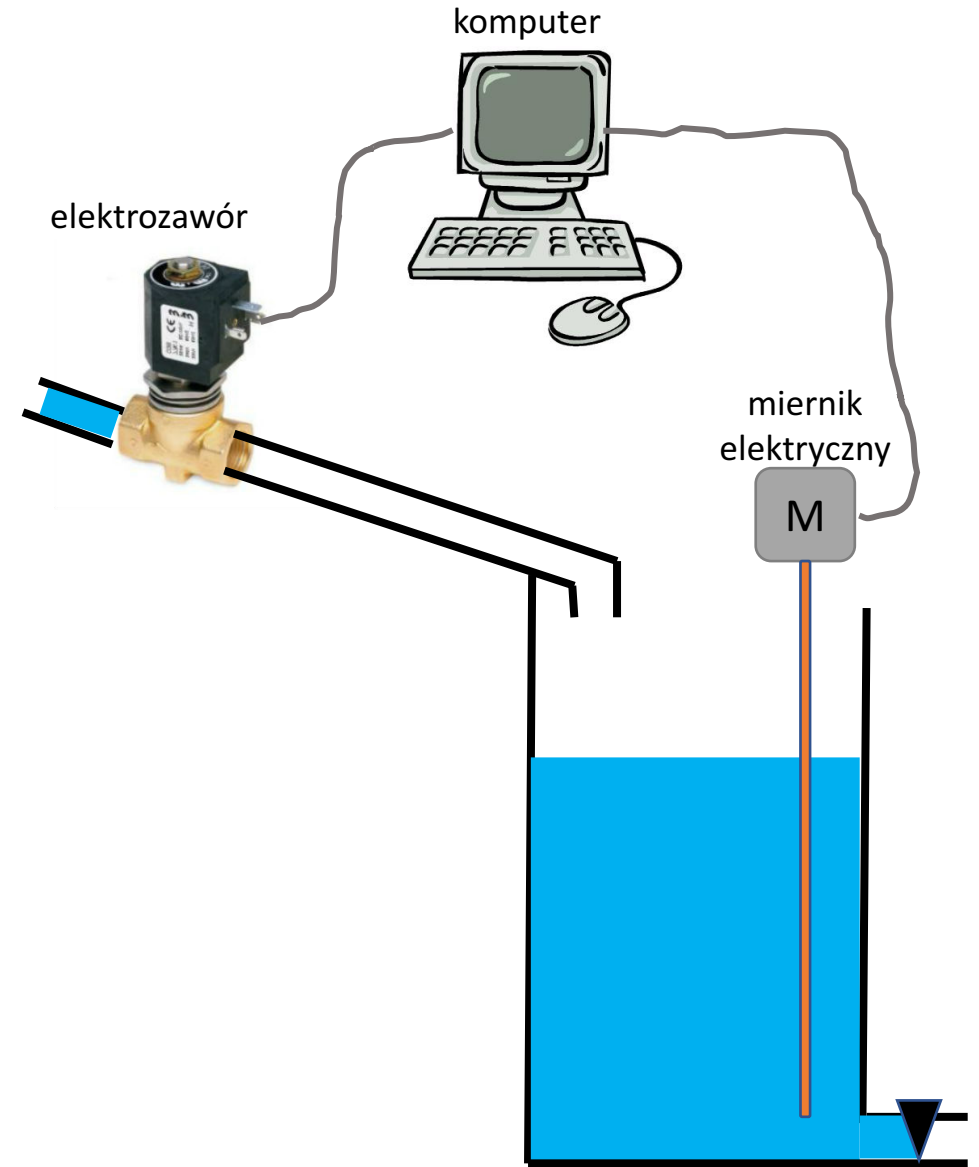
SYGNAŁ CIĄGŁY



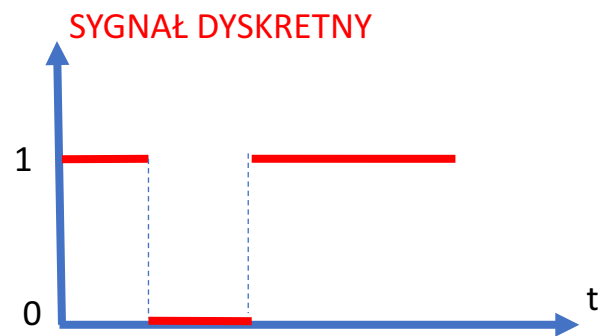
# UKŁAD STEROWANIA

ANALOGOWY

CYFROWY



# LOGIKA MATEMATYCZNA



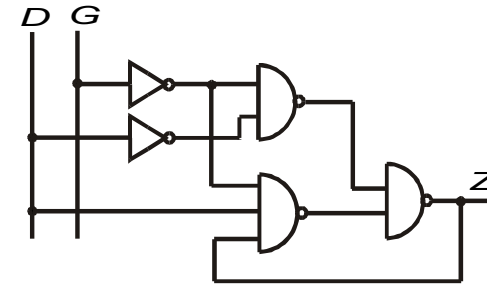
$x_1$	$x_2$	$x_3$	$y$
0	0	0	1
1	0	0	1
0	1	0	0
0	0	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	1	1	1

W tym semestrze !

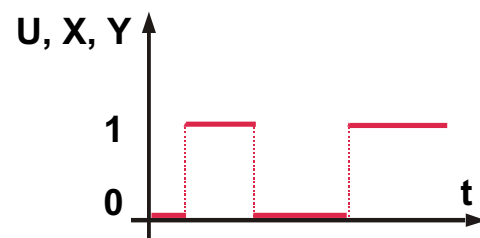
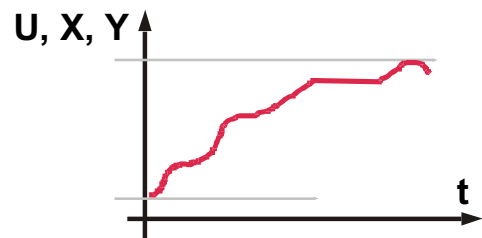
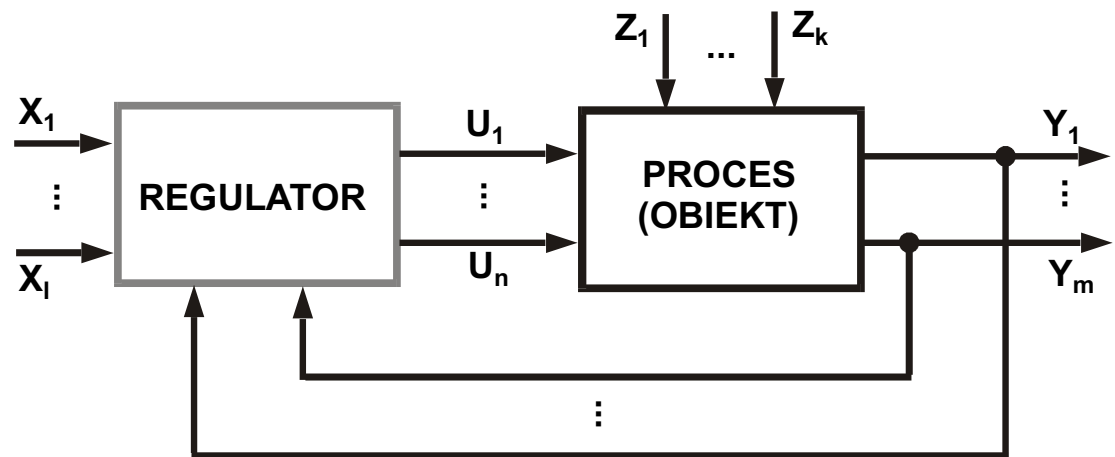
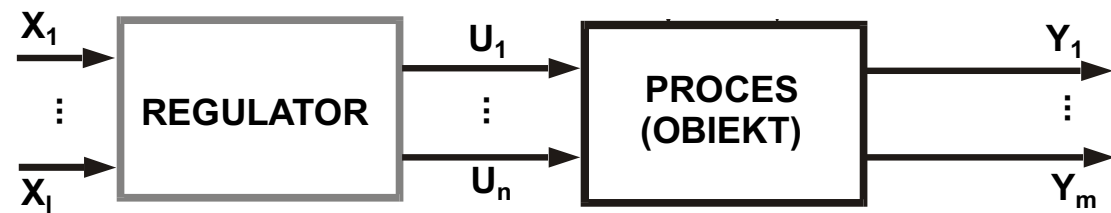
# AUTOMATYZACJA DYSKRETNYCH PROCESÓW PRODUKCYJNYCH

## CEL WYKŁADU

- Przedstawienie podstawowych zagadnień związanych z realizacją przełączających (logicznych) układów regulacji i sterowania
- W szczególności zaprezentowanie sposobu realizacji takich układów z wykorzystaniem sterowników programowalnych PLC



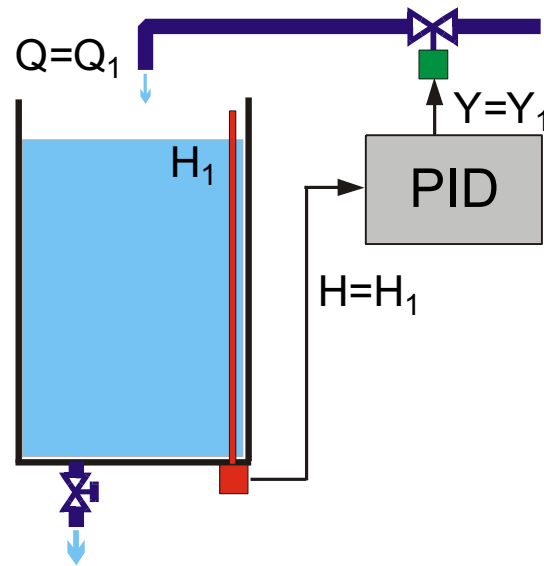
# Procesy ciągłe i dyskretne – cd.





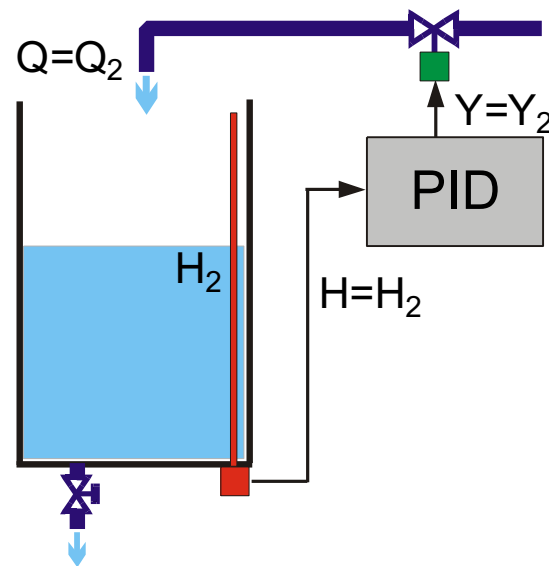
Procesy ciągłe i dyskretne – cd.

sensory i człony wykonawcze



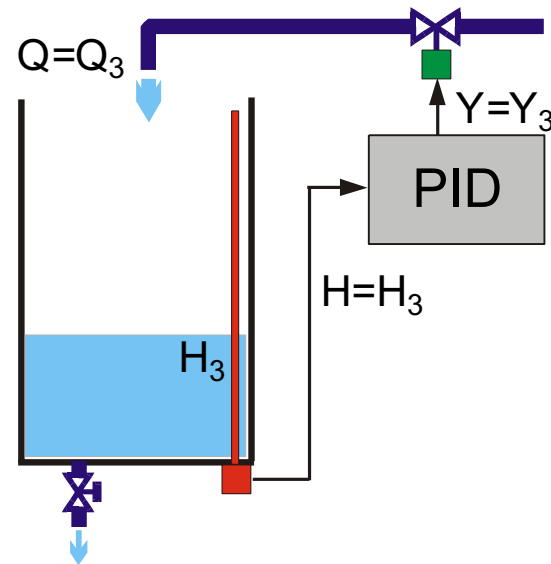
Procesy ciągłe i dyskretne – cd.

sensory i człony wykonawcze



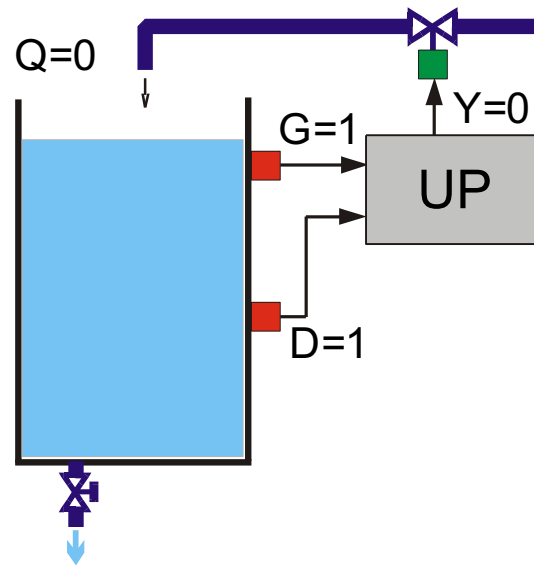
Procesy ciągłe i dyskretne – cd.

sensory i człony wykonawcze



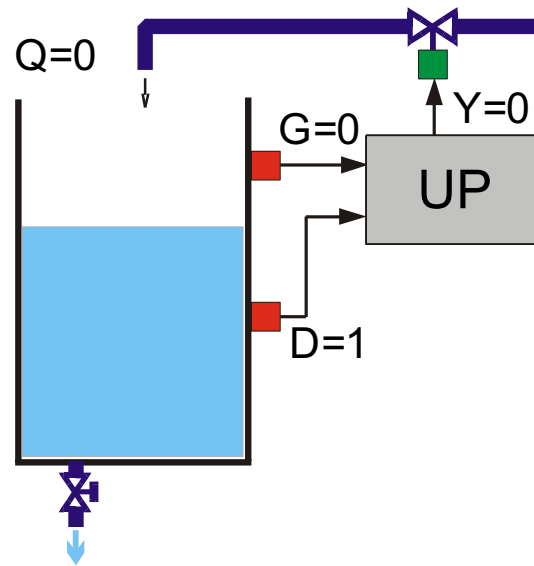
Procesy ciągłe i dyskretne – cd.

sensory i człony wykonawcze



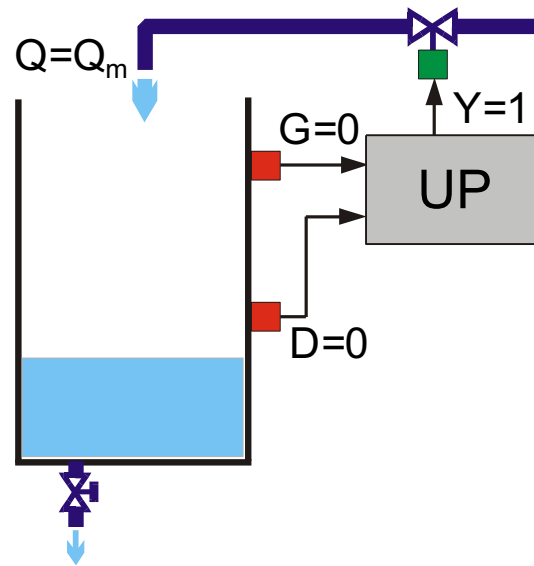
Procesy ciągłe i dyskretne – cd.

sensory i człony wykonawcze



Procesy ciągłe i dyskretne – cd.

sensory i człony wykonawcze



**UKŁADY PRZEŁĄCZAJĄCE (LOGICZNE)**

```
graph TD; A[UKŁADY PRZEŁĄCZAJĄCE (LOGICZNE)] --> B[KOMBINACYJNE]; A --> C[SEKWENCYJNE]
```

**KOMBINACYJNE**

**SEKWENCYJNE**

## **UKŁADY PRZEŁĄCZAJĄCE (LOGICZNE)**

```
graph TD; A[UKŁADY PRZEŁĄCZAJĄCE (LOGICZNE)] --> B[KOMBINACYJNE]; A --> C[SEKWENCYJNE]; B --- D[Realizują zależności funkcyjne];
```

### **KOMBINACYJNE**

**Realizują zależności funkcyjne**

### **SEKWENCYJNE**



## **UKŁADY PRZEŁĄCZAJĄCE (LOGICZNE)**

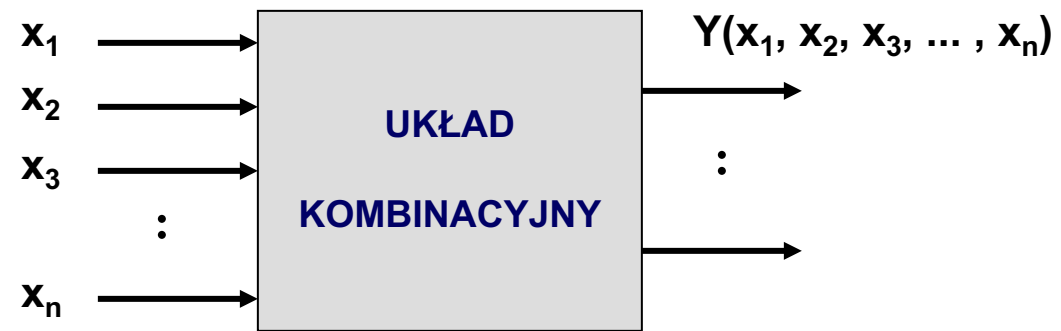
```
graph TD; A[UKŁADY PRZEŁĄCZAJĄCE (LOGICZNE)] --> B[KOMBINACYJNE]; A --> C[SEKWENCYJNE]; B --- B_desc[Realizują zależności funkcyjne]; C --- C_desc[Układy z pamięcią i uzależnieniem czasowym];
```

### **KOMBINACYJNE**

**Realizują zależności funkcyjne**

### **SEKWENCYJNE**

**Układy z pamięcią i uzależnieniem czasowym**



## WYBRANE ELEMENTY LOGIKI MATEMATYCZNEJ

$x_1$	$x_2$	$x_3$	$y$
0	0	0	1
1	0	0	1
0	1	0	0
0	0	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	1	1	1

## FUNKCJE LOGICZNE

$$y = f(x)$$

$$x \in \{0, 1\}$$

$$y \in \{0, 1\}$$

$$y = f(x_1, x_2)$$

$$x_1, x_2 \in \{0, 1\}$$

$$y \in \{0, 1\}$$

Sposób przedstawiania: **wzory, tabele, ~~wykresy~~**

# ELEMENTARNE FUNKCJE LOGICZNE

x	y=x
0	0
1	1

$$y = x$$

x	$y = \bar{x}$
0	1
1	0

$$y = \bar{x} \quad \text{negacja}$$

$x_1$	$x_2$	$y=x_1+x_2$
0	0	0
1	0	1
0	1	1
1	1	1

$$y=x_1+x_2 \quad \text{suma logiczna}$$

$x_1$	$x_2$	$y=x_1 \cdot x_2$
0	0	0
1	0	0
0	1	0
1	1	1

$$y=x_1 \cdot x_2 \quad \text{iloczyn logiczny}$$

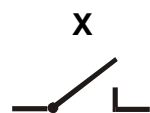
$x_1$	$x_2$	$y = \overline{x_1 + x_2}$
0	0	1
1	0	0
0	1	0
1	1	0

$$y = \overline{x_1 + x_2} \quad \text{negacja sumy log.}$$

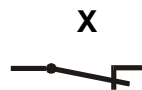
$x_1$	$x_2$	$y = \overline{x_1 \cdot x_2}$
0	0	1
1	0	1
0	1	1
1	1	0

$$y = \overline{x_1 \cdot x_2} \quad \text{negacja iloczynu log.}$$

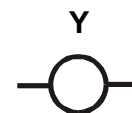
## FIZYCZNA REALIZACJA FUNKCJI LOGICZNYCH



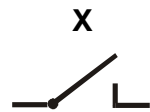
STYK  
NORMALNIE OTWARTY



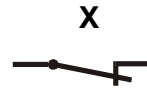
STYK  
NORMALNIE ZAMKNIĘTY



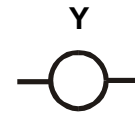
PRZEKAŹNIK



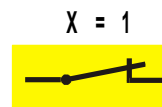
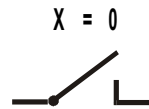
STYK  
NORMALNIE OTWARTY

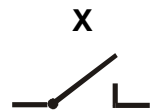


STYK  
NORMALNIE ZAMKNIĘTY

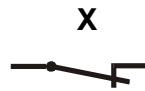


PRZEKAŹNIK

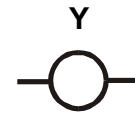




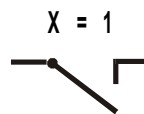
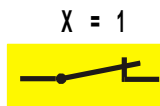
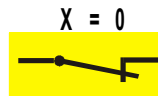
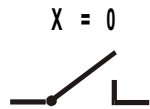
STYK  
NORMALNIE OTWARTY



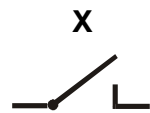
STYK  
NORMALNIE ZAMKNIĘTY



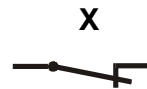
PRZEKAŹNIK



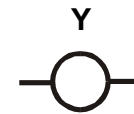




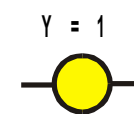
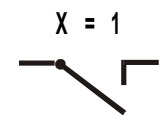
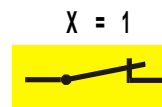
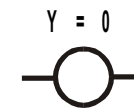
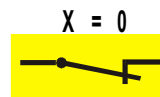
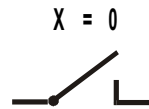
STYK  
NORMALNIE OTWARTY

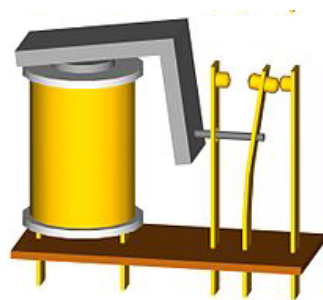
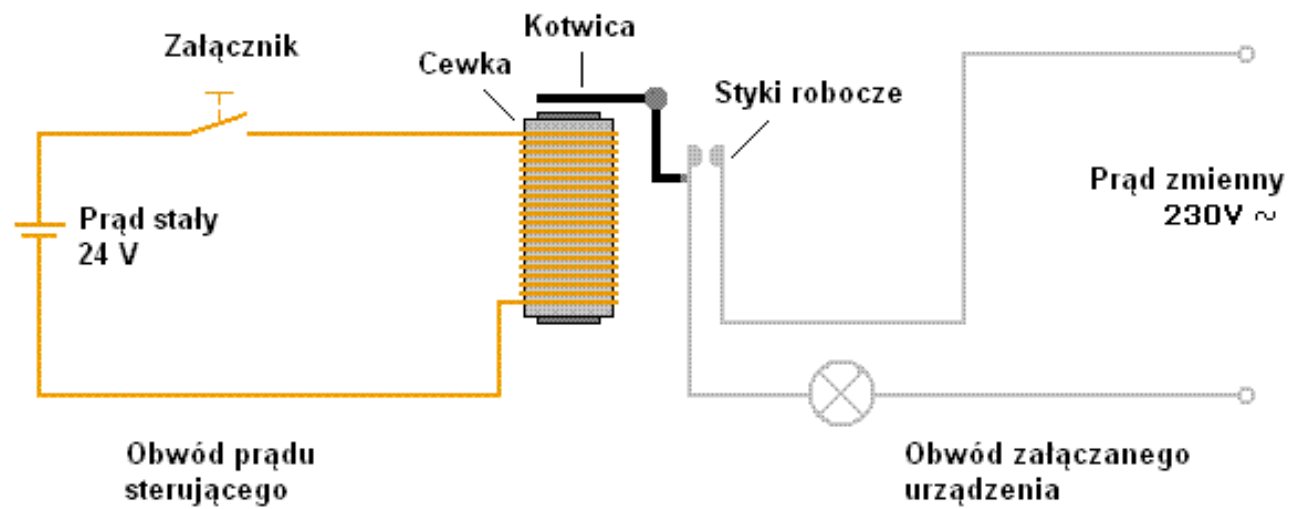


STYK  
NORMALNIE ZAMKNIĘTY

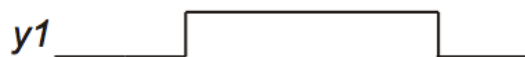
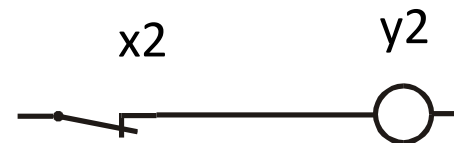


PRZEKAŹNIK





Styk normalnie zamknięty  
umożliwia realizację NEGACJI

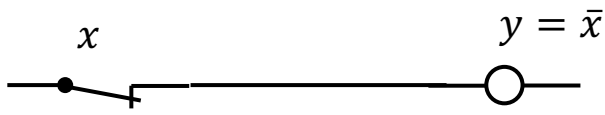


# FIZYCZNA REALIZACJA FUNKCJI LOGICZNYCH

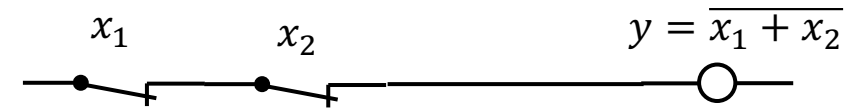
x	y=x
0	0
1	1



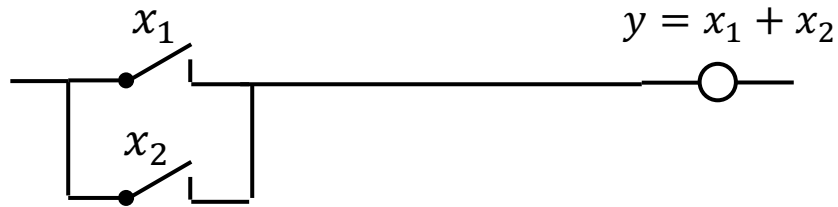
x	y = $\bar{x}$
0	1
1	0



x <sub>1</sub>	x <sub>2</sub>	y = $\overline{x_1 + x_2}$
0	0	1
1	0	0
0	1	0
1	1	0



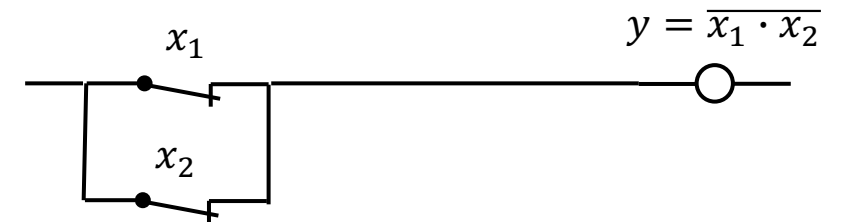
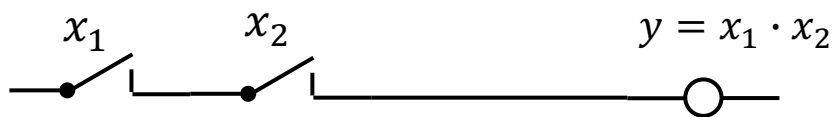
x <sub>1</sub>	x <sub>2</sub>	y=x <sub>1</sub> +x <sub>2</sub>
0	0	0
1	0	1
0	1	1
1	1	1



$y = \bar{x}$

x <sub>1</sub>	x <sub>2</sub>	y = $\overline{x_1 \cdot x_2}$
0	0	1
1	0	1
0	1	1
1	1	0

x <sub>1</sub>	x <sub>2</sub>	y=x <sub>1</sub> ·x <sub>2</sub>
0	0	0
1	0	0
0	1	0
1	1	1



$$\bar{\bar{x}} = x$$

$$x + x = x$$

$$x \cdot x = x$$

$$x + 0 = x$$

$$x + 1 = 1$$

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

$$(A \cdot B) + (A \cdot \bar{B}) = A$$

$$(A + B) \cdot (A + \bar{B}) = A$$

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$x_1$	$x_2$	$x_3$	$y$
0	0	0	1
1	0	0	1
0	1	0	0
0	0	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	1	1	1

$$y = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot 1 +$$

$$x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot 1 +$$

$$\bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot 0 +$$

$$\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot 0 +$$

$$x_1 \cdot x_2 \cdot \bar{x}_3 \cdot 0 +$$

$$x_1 \cdot \bar{x}_2 \cdot x_3 \cdot 1 +$$

$$\bar{x}_1 \cdot x_2 \cdot x_3 \cdot 0 +$$

$$x_1 \cdot x_2 \cdot x_3 \cdot 1$$

$$y = (x_1 + x_2 + x_3 + 1) \cdot$$

$$(\bar{x}_1 + x_2 + x_3 + 1) \cdot$$

$$(x_1 + \bar{x}_2 + x_3 + 0) \cdot$$

$$(x_1 + x_2 + \bar{x}_3 + 0) \cdot$$

$$(\bar{x}_1 + \bar{x}_2 + x_3 + 0) \cdot$$

$$(\bar{x}_1 + x_2 + \bar{x}_3 + 1) \cdot$$

$$(x_1 + \bar{x}_2 + \bar{x}_3 + 1) \cdot$$

$$(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + 1)$$

(wybrane) PRAWA LOGIKI MATEMATYCZNEJ

$x_1$	$x_2$	$x_3$	$y$
0	0	0	1
1	0	0	1
0	1	0	0
0	0	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	1	1	1

$$y = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 +$$

$$x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 +$$

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

$$(x_1 + x_2 + \bar{x}_3) \cdot$$

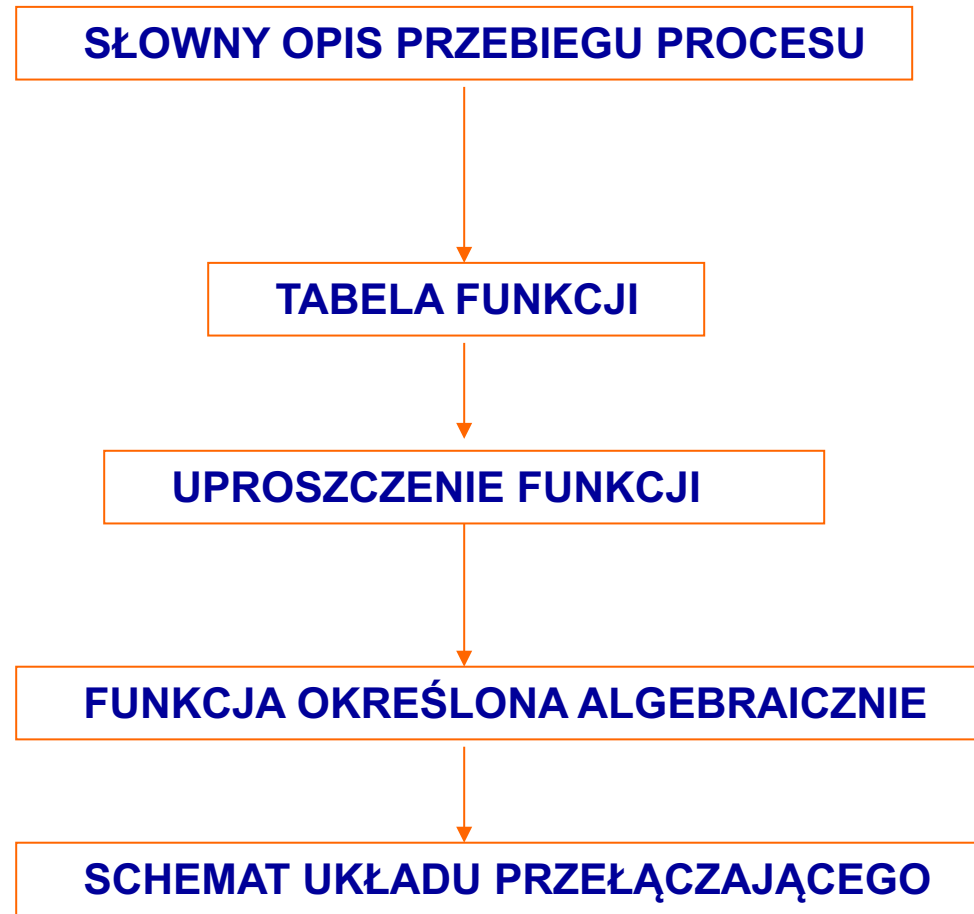
$$(\bar{x}_1 + \bar{x}_2 + x_3) \cdot$$

$$(x_1 + \bar{x}_2 + \bar{x}_3)$$

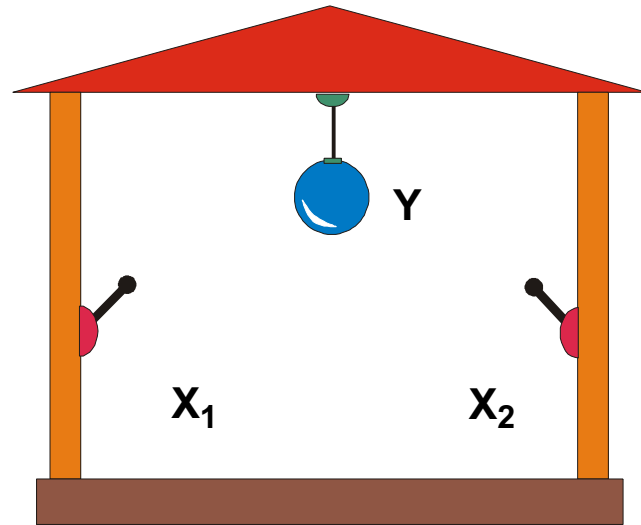
$$x_1 \cdot \bar{x}_2 \cdot x_3 +$$

$$x_1 \cdot x_2 \cdot x_3$$

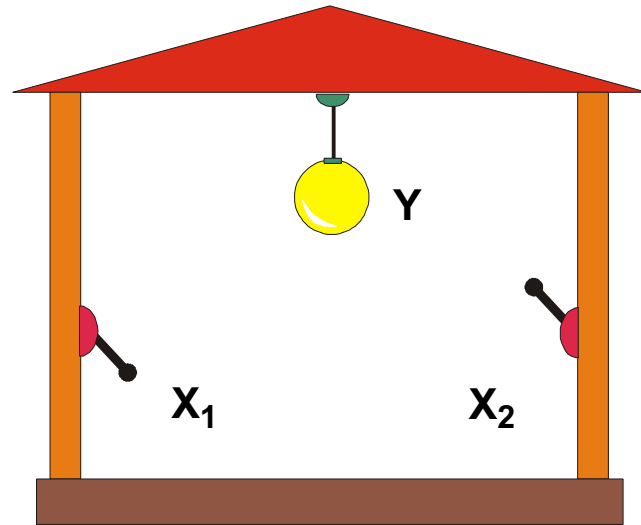
# PROJEKTOWANIE



# PRZYKŁAD 1

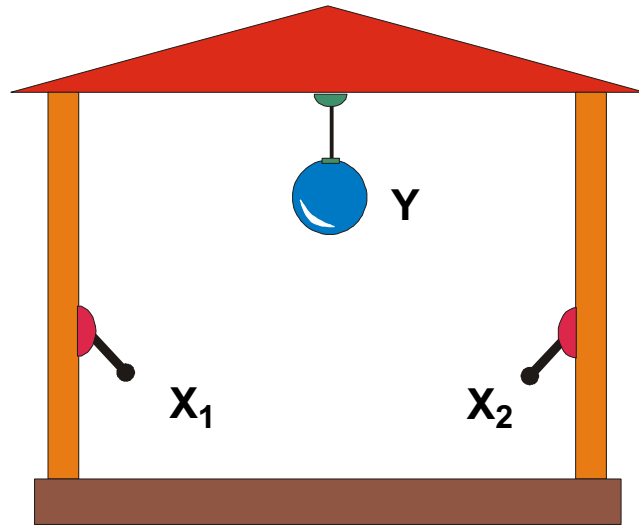


# PRZYKŁAD 1

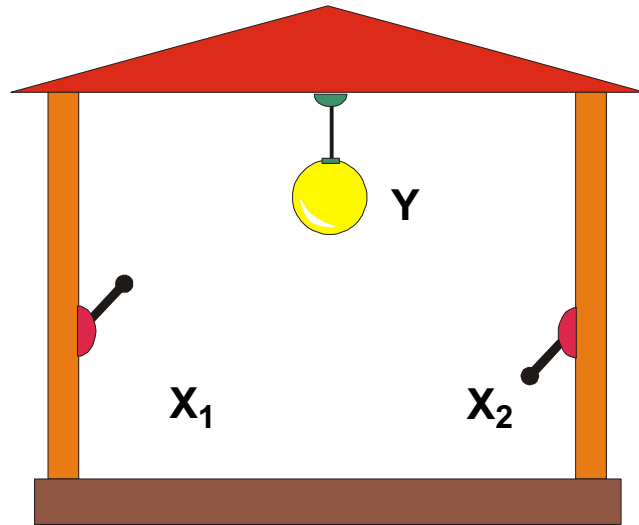




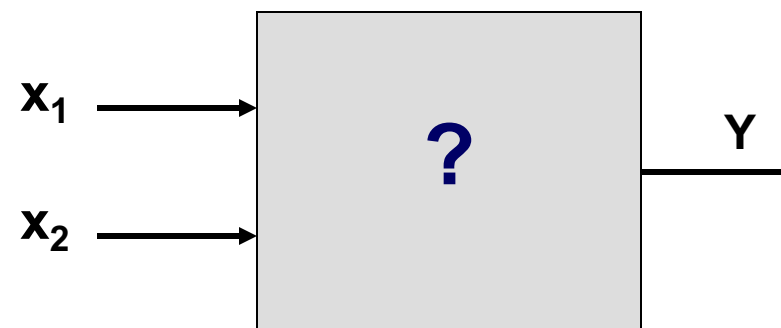
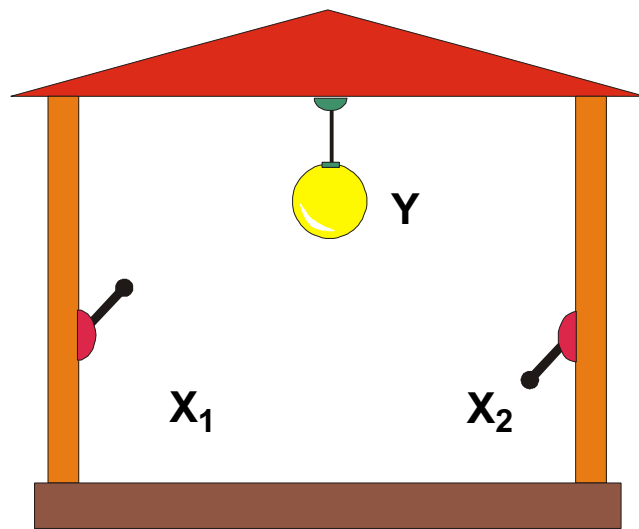
# PRZYKŁAD 1



# PRZYKŁAD 1



## PRZYKŁAD 1



**SŁOWNY OPIS PRZEBIEGU PROCESU:**

**ZMIANA STANU DOWOLNEGO SYGNAŁU WEJŚCIOWEGO  
POWODUJE ZMIANĘ STANU SYGNAŁU WYJŚCIOWEGO.**

## PRZYKŁAD 1

### SŁOWNY OPIS PRZEBIEGU PROCESU:

ZMIANA STANU DOWOLNEGO SYGNAŁU WEJŚCIOWEGO  
POWODUJE ZMIANĘ STANU SYGNAŁU WYJŚCIOWEGO.

### POSTAĆ ALGEBRAICZNA FUNKCJI:

$$Y = \bar{x}_1 \cdot \bar{x}_2 \cdot 0 + x_1 \cdot \bar{x}_2 \cdot 1 + \bar{x}_1 \cdot x_2 \cdot 1 + x_1 \cdot x_2 \cdot 0$$

$$Y = (x_1 + x_2 + 0) \cdot (\bar{x}_1 + x_2 + 1) \cdot (x_1 + \bar{x}_2 + 1) \cdot (\bar{x}_1 + \bar{x}_2 + 0)$$

$$Y = x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2$$

$$Y = (x_1 + x_2) \cdot (\bar{x}_1 + \bar{x}_2)$$

### TABELA FUNKCJI:

$x_1$	$x_2$	Y
0	0	0
1	0	1
0	1	1
1	1	0

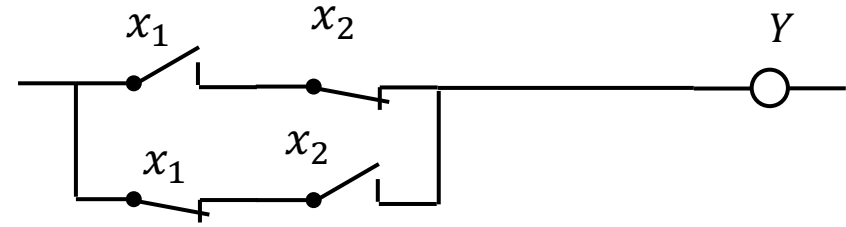
### SCHEMAT UKŁADU:



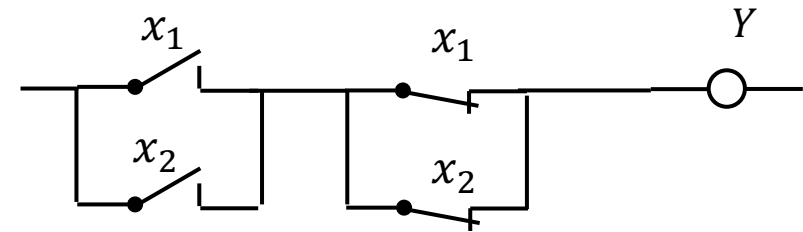
## PRZYKŁAD 1

SCHEMAT(Y) UKŁADU:

$$Y = x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2$$



$$Y = (x_1 + x_2) \cdot (\bar{x}_1 + \bar{x}_2)$$



## SPOSOBY REALIZACJI ZALEŻNOŚCI OKREŚLONYCH PRZEZ FUNKCJE LOGICZNE

STYKI i PRZEKAŹNIKI

BRAMKI LOGICZNE [pneumatyczne, hydrauliczne, elektroniczne (w tym układy scalone)]

KOMPUTERY [realizacja programowa]

# BRAMKI LOGICZNE [pneumatyczne, hydrauliczne, elektroniczne (w tym układy scalone)]

X	Y
0	0
1	1

X	Y
0	1
1	0

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

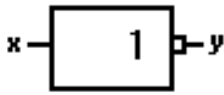
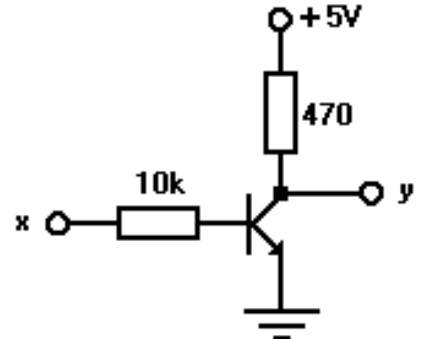
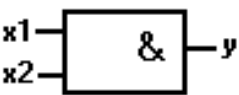
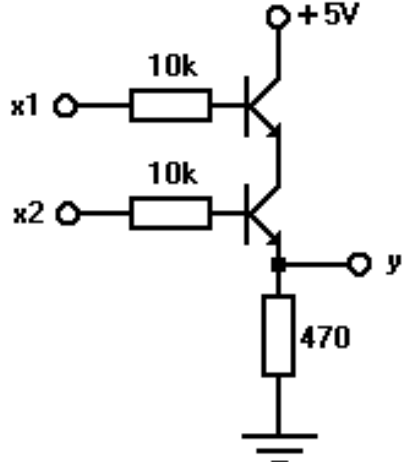
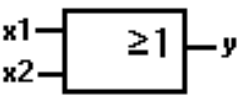
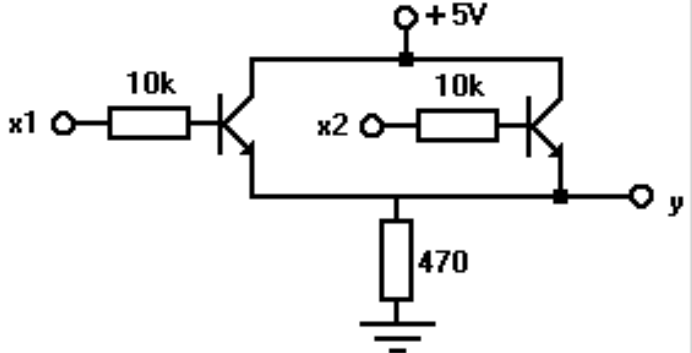
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

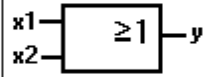
Funkcja	1	2
Powtórzenie		
Negacja		
Koniunkcja		
Alternatywa		
NAND		
NOR		

Oznaczenia bramek logicznych

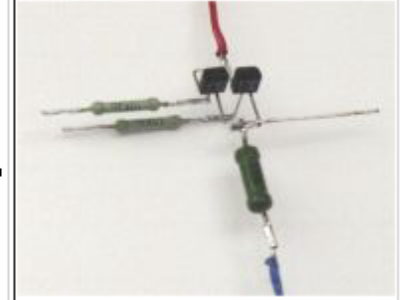
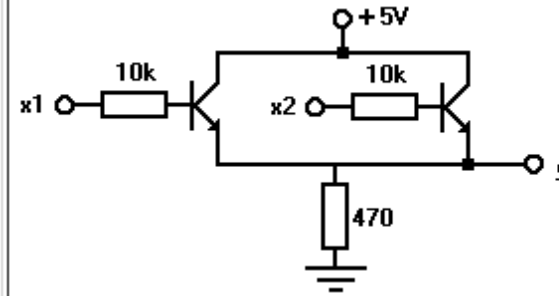
NOT		<table border="1" data-bbox="1057 214 1370 385"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	0	1	1	0										
x	y																	
0	1																	
1	0																	
AND		<table border="1" data-bbox="955 585 1472 871"> <thead> <tr> <th>x1</th> <th>x2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x1	x2	y	0	0	0	0	1	0	1	0	0	1	1	1	
x1	x2	y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		<table border="1" data-bbox="955 1021 1472 1306"> <thead> <tr> <th>x1</th> <th>x2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x1	x2	y	0	0	0	0	1	1	1	0	1	1	1	1	
x1	x2	y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																



OR



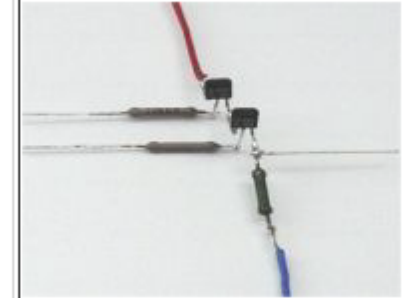
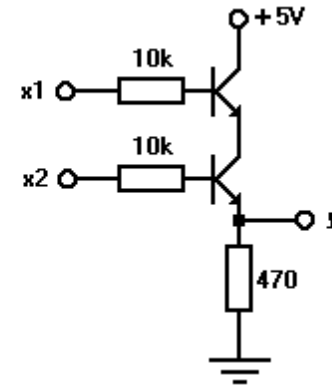
x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	1

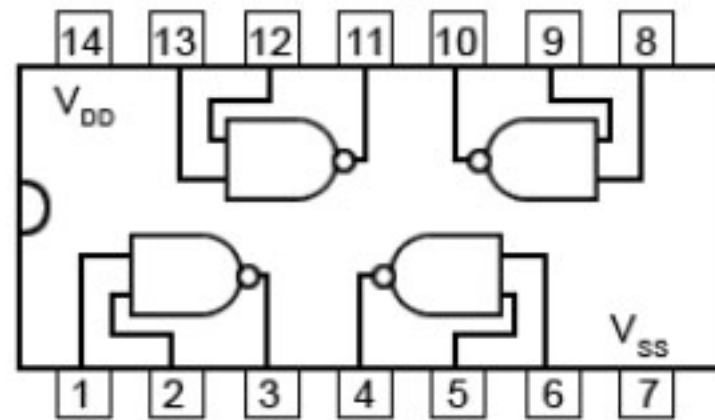
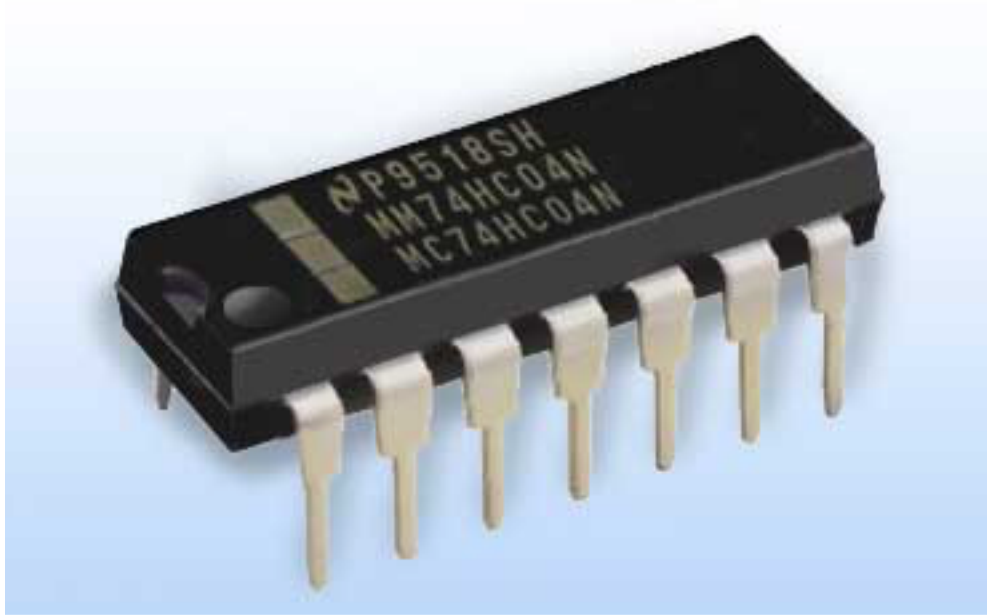


AND



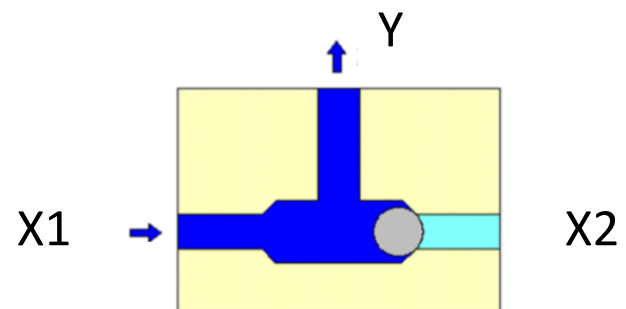
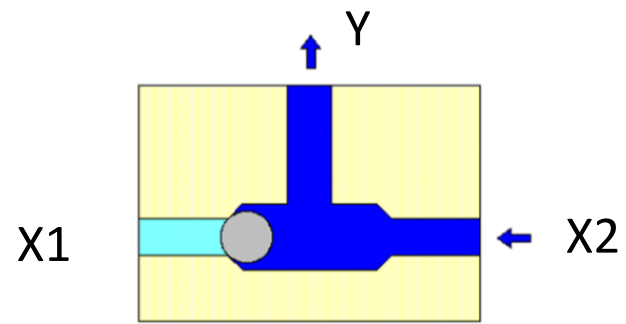
x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1



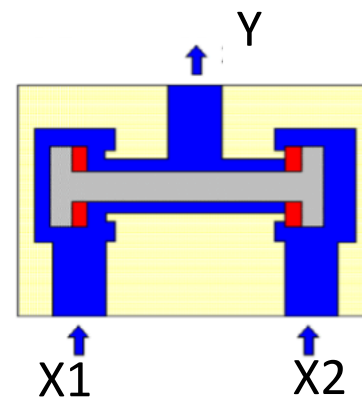
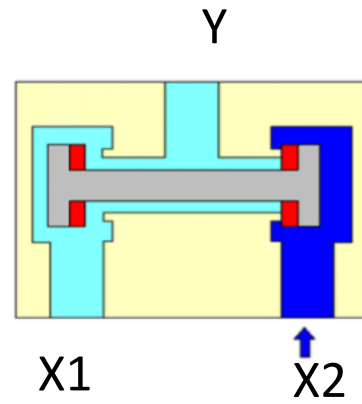


4011 CMOS NAND

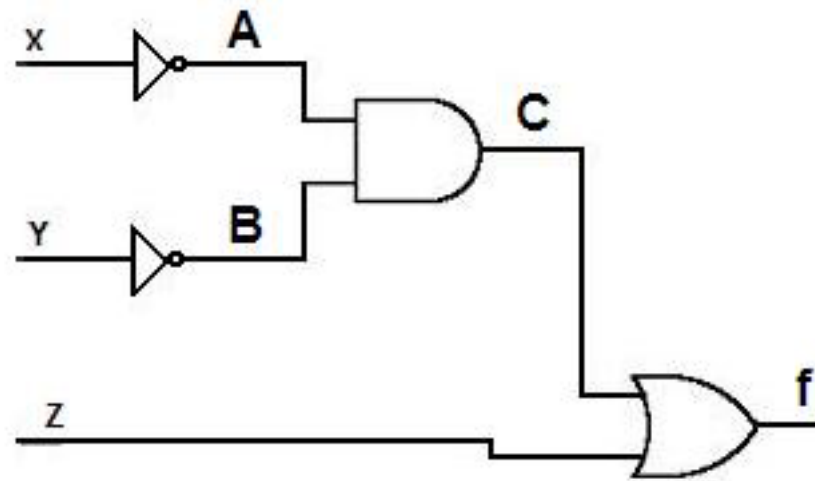
$$Y = X1 + X2$$



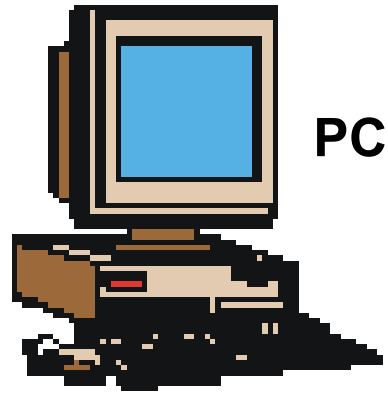
$$Y = X1 \cdot X2$$



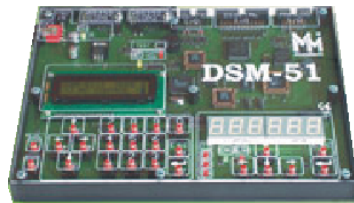
Układ połączonych 4. bramek logicznych



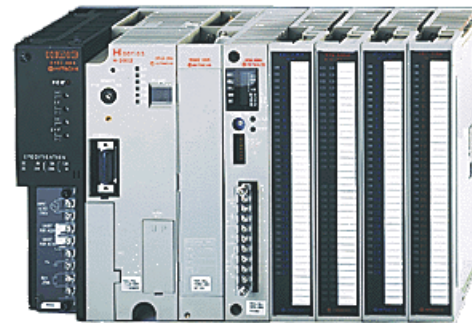
## KOMPUTEROWE SYSTEMY STEROWANIA



PC



Specjalny system  
mikroprocesorowy



PLC - Programmable Logic Controller

STEROWNIK PROGRAMOWALNY

## KOMPUTEROWE SYSTEMY STEROWANIA

**STEROWNIKI SĄ TO KOMPUTERY  
PRZEMYSŁOWE.**

**MAJĄ ARCHITEKTURĘ KLASYCZNEGO  
KOMPUTERA**

**UMOŻLIWIAJĄ STEROWANIE W  
UKŁADZIE OTWARTYM I / LUB  
ZAMKNIĘTYM.**



**PLC - Programmable Logic Controller**

**STEROWNIK PROGRAMOWALNY**

## KOMPUTEROWE SYSTEMY STEROWANIA

**PIERWSZE STEROWNIKI POJAWIŁY SIĘ W  
LATACH 60-TYCH W USA**

**- DO STEROWANIA PRODUKCJĄ  
KALKULATORÓW (TEXAS INSTRUMENTS)**

**- W PRZEMYŚLE SAMOCHODOWYM  
(GENERAL MOTORS)**



**PLC - Programmable Logic Controller**

**STEROWNIK PROGRAMOWALNY**



## PODZIAŁ STEROWNIKÓW

LICZBA WEJŚĆ / WYJŚĆ:

MAŁE (DO 64)

DUŻE (POWYŻEJ 512)

BUDOWA:

KOMPAKTOWE (SZTYWNA ARCHITEKTURA)

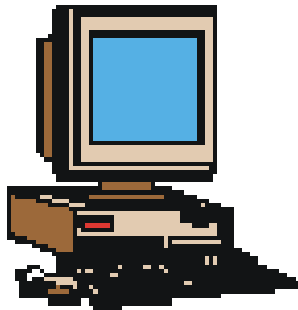
MODUŁOWE (MOŻLIWOŚĆ KOMPONOWANIA STRUKTURY SPRZĘTOWEJ)



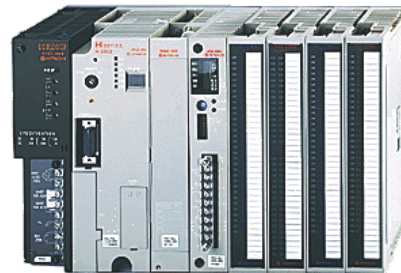
PLC

# PROGRAMOWANIE STEROWNIKA

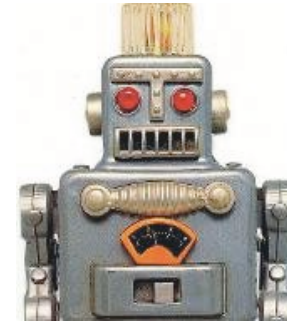
PC



PLC



PROCES  
(OBIEKT)



Komputer programator  
+  
Program narzędziowy  
(do tworzenia programów  
sterujących)

## PRZESYŁANIE PROGRAMU STERUJĄCEGO DO STEROWNIKA

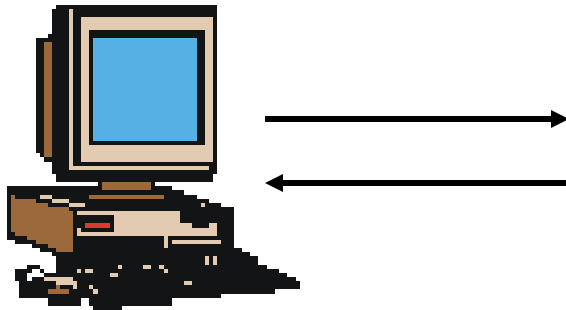
### TESTOWANIE PROGRAMU



Komputer programator  
+  
Program narzędziowy  
(do tworzenia programów  
sterujących)

## WIZUALIZACJA

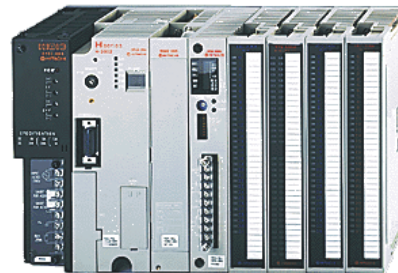
PC



Komputer interfejs operatora  
+  
Program wizualacyjny  
(do monitoringu procesu)

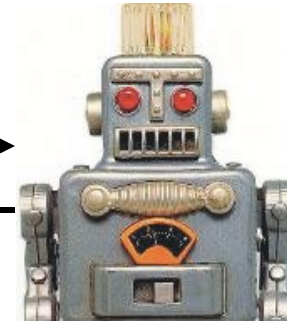
## STEROWANIE PROCESEM

PLC

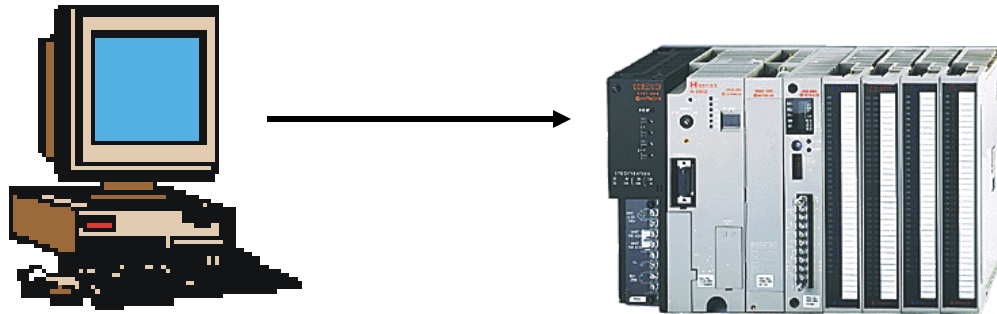


Sterownik  
+  
Program sterujący

PROCES  
(OBIEKT)



## **JEZYKI PROGRAMOWANIA STEROWNIKÓW**



**NORMA EN 61131:**

**LISTA INSTRUKCJI (Instruction List IL)**

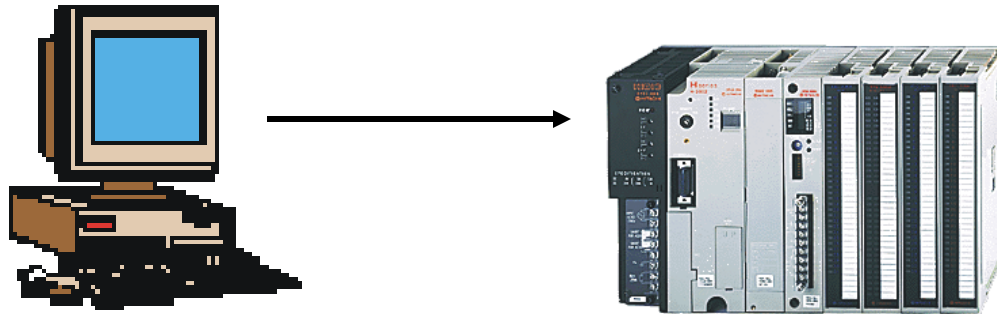
**TEKST STRUKTURALNY (Structured Text ST)**

**SCHEMAT DRABINKOWY (Ladder Diagram LD)**

**SCHEMAT BLOKÓW FUNKCYJNYCH (Function Block Diagram FBD)**

**SEKWENCYJNA SIEĆ DZIAŁAŃ (Sequential Function Chart SFC)**

## JEZYKI PROGRAMOWANIA STEROWNIKÓW



**NORMA EN 61131:**

**LISTA INSTRUKCJI (Instruction List IL)**

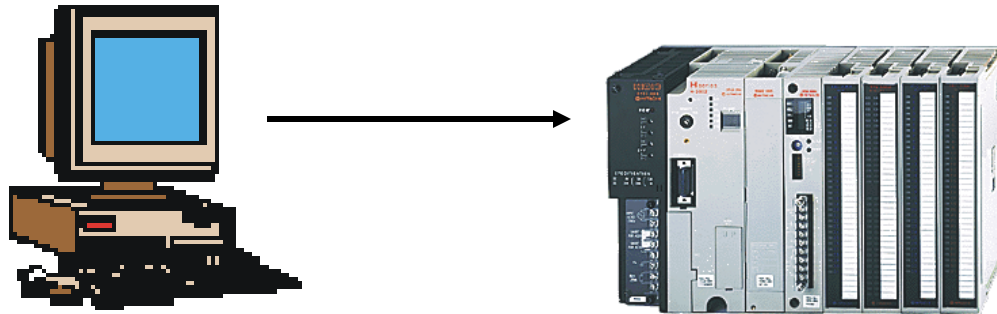
**TEKST STRUKTURALNY (Structured Text ST)**

**SCHEMAT DRABINKOWY (Ladder Diagram LD)**

**SCHEMAT BLOKÓW FUNKCYJNYCH (Function Block Diagram FBD)**

**SEKWENCYJNA SIEĆ DZIAŁAŃ (Sequential Function Chart SFC)**

## JEZYKI PROGRAMOWANIA STEROWNIKÓW



**NORMA EN 61131:**

**LISTA INSTRUKCJI (Instruction List IL)**

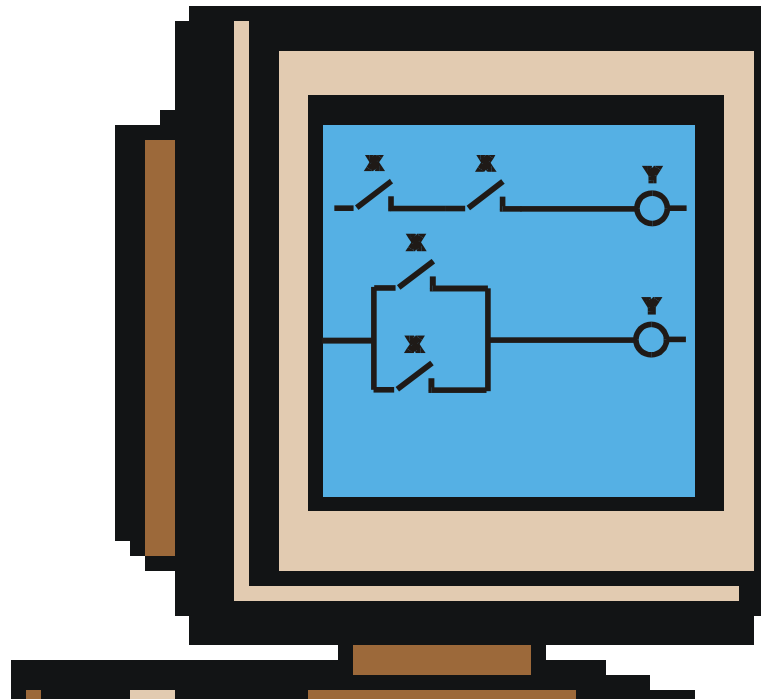
**TEKST STRUKTURALNY (Structured Text ST)**

**SCHEMAT DRABINKOWY = SCHEMAT STYKOWO-PRZEKAŹNIKOWY**

**SCHEMAT BLOKÓW FUNKCYJNYCH (Function Block Diagram FBD)**

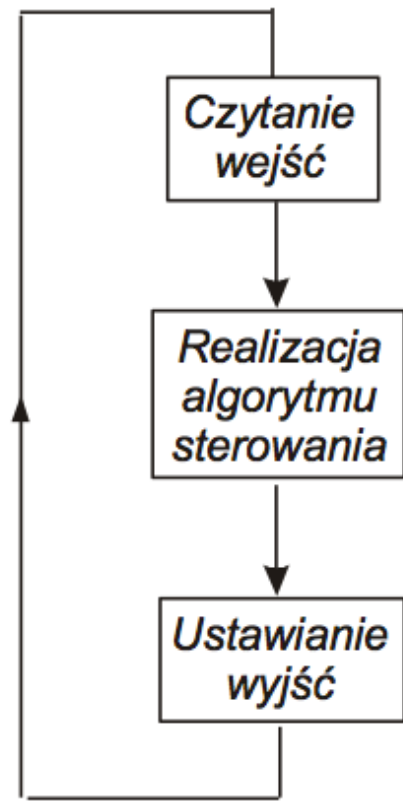
**SEKWENCYJNA SIEĆ DZIAŁAŃ (Sequential Function Chart SFC)**

# JEZYKI PROGRAMOWANIA STEROWNIKÓW

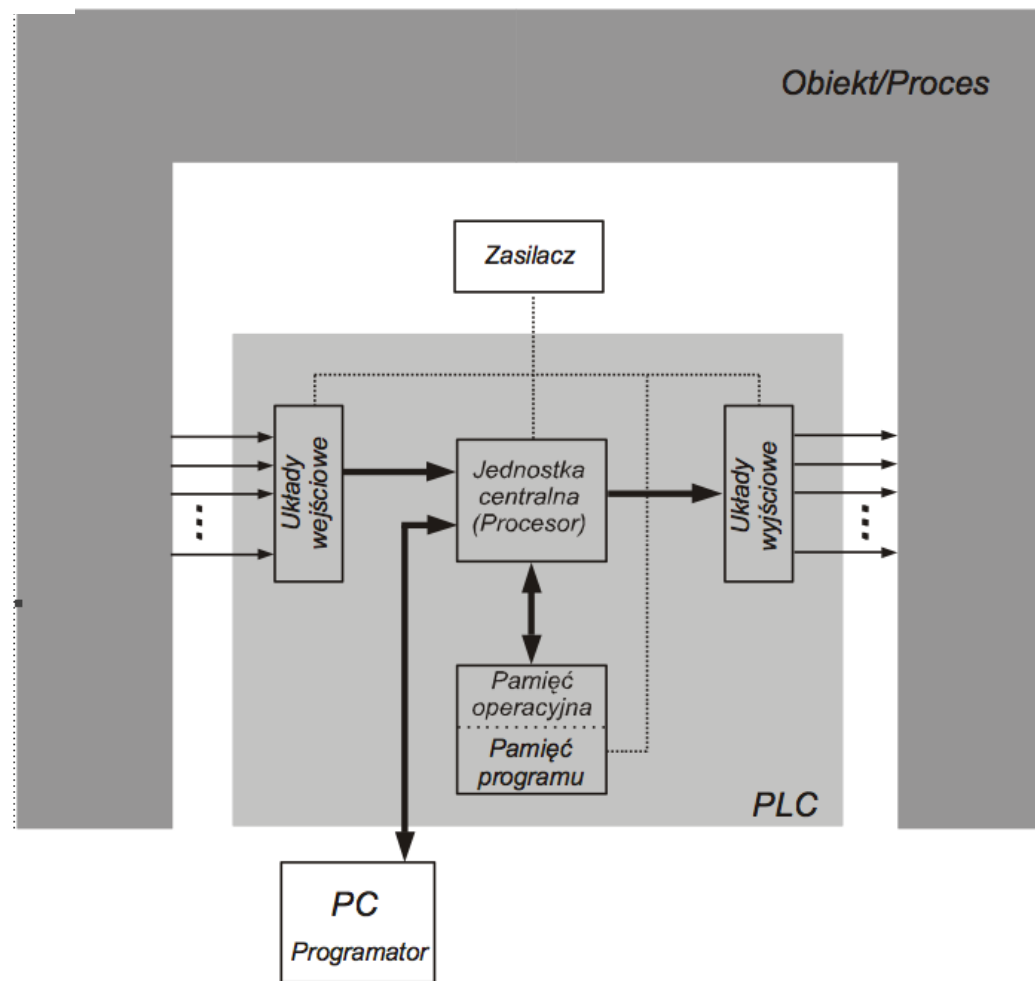


**SCHEMAT DRABINKOWY = SCHEMAT STYKOWO-PRZEKAŹNIKOWY**





Cykl pracy PLC



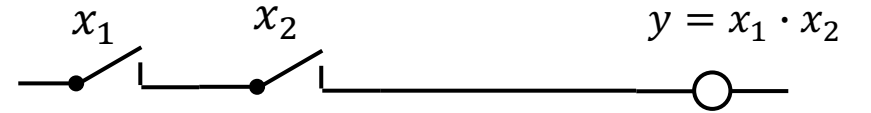
Ogólny schemat funkcjonalny PLC



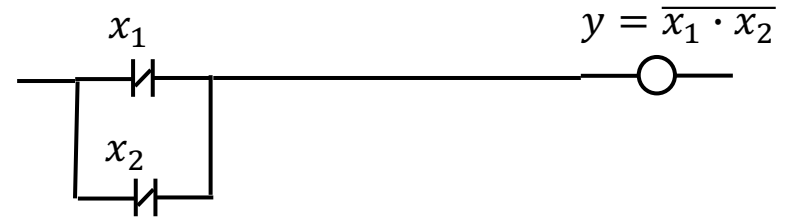
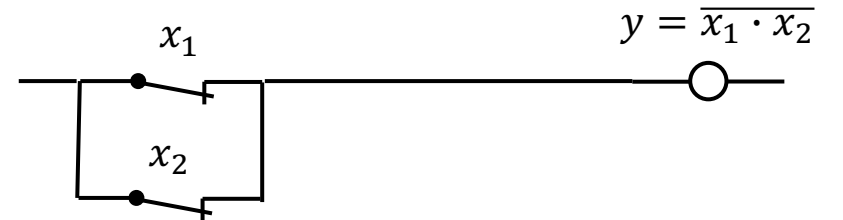
*w PLC*

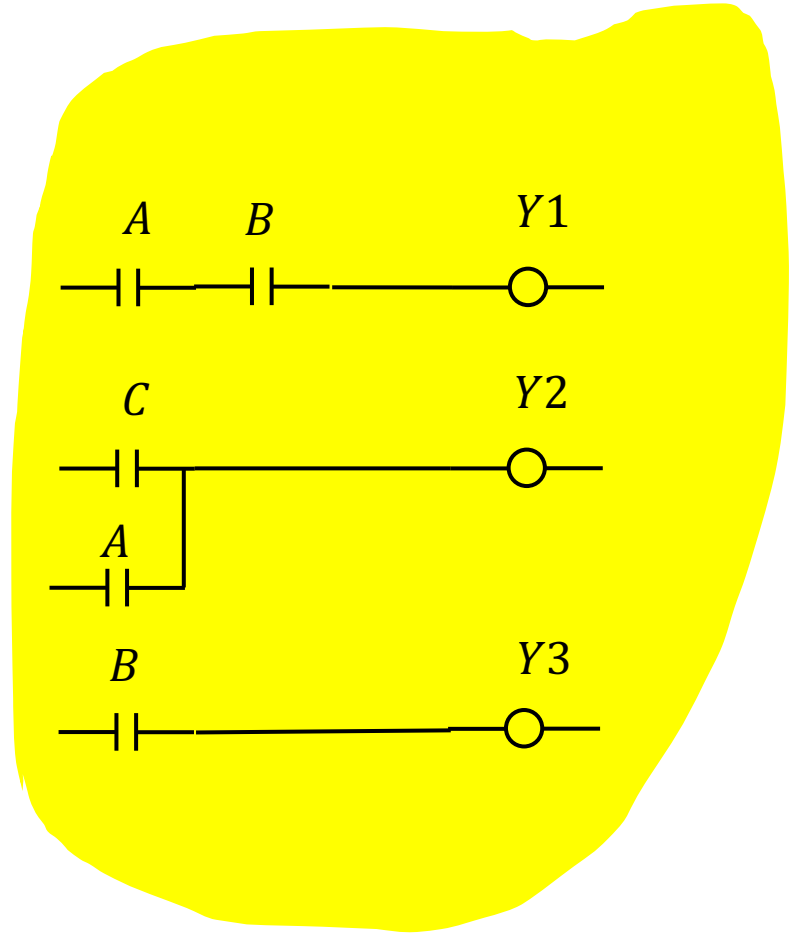
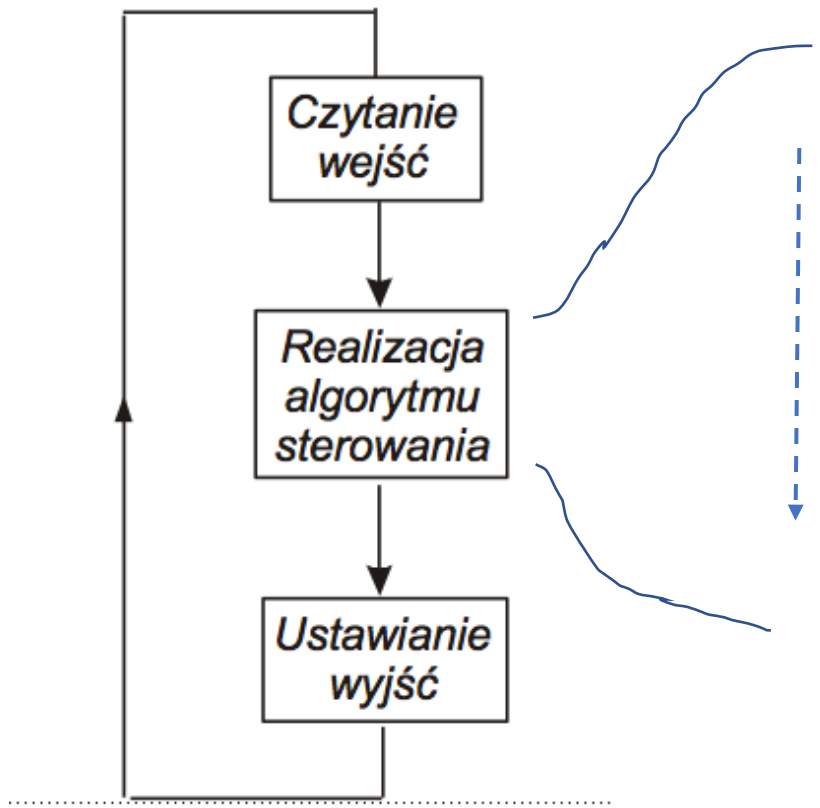
SYMBOLE GRAFICZNE

$x_1$	$x_2$	$y = x_1 \cdot x_2$
0	0	0
1	0	0
0	1	0
1	1	1



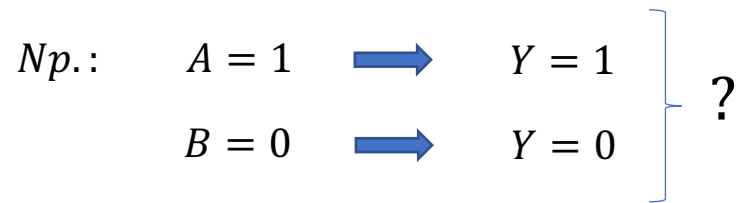
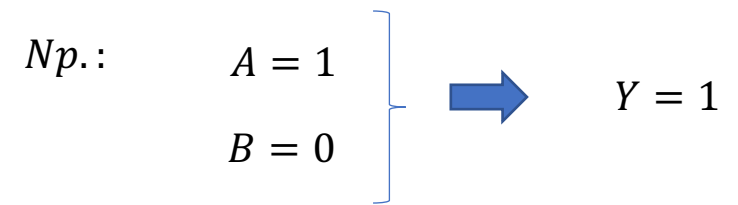
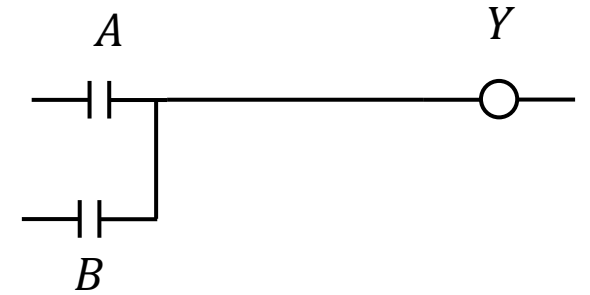
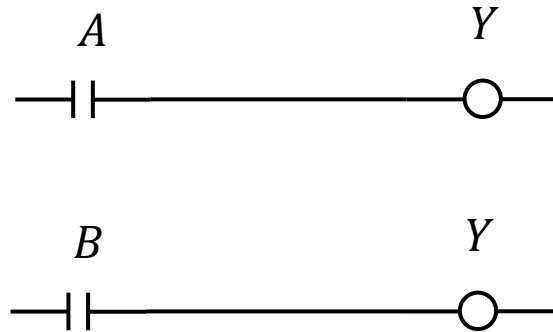
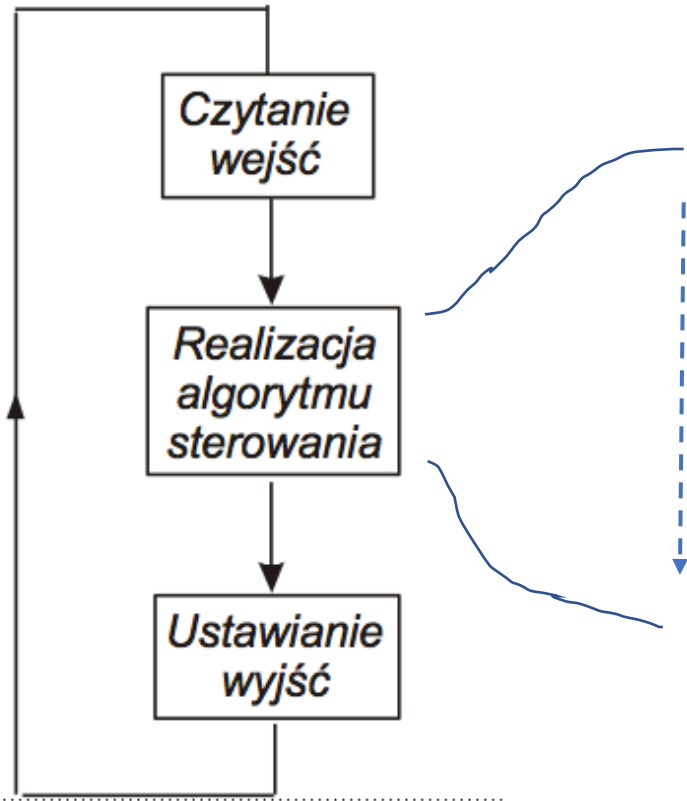
$x_1$	$x_2$	$y = \overline{x_1 \cdot x_2}$
0	0	1
1	0	1
0	1	1
1	1	0

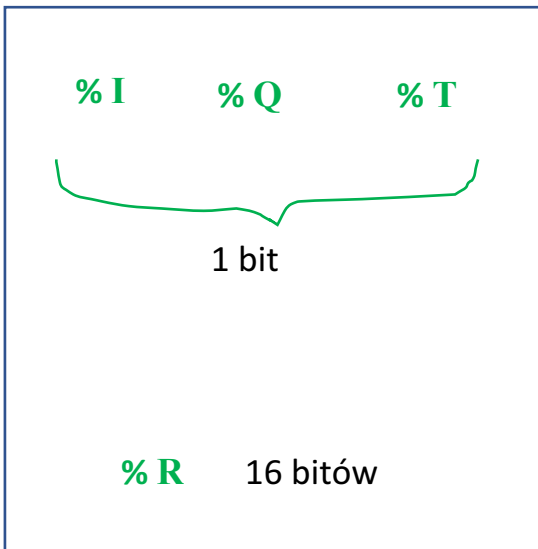
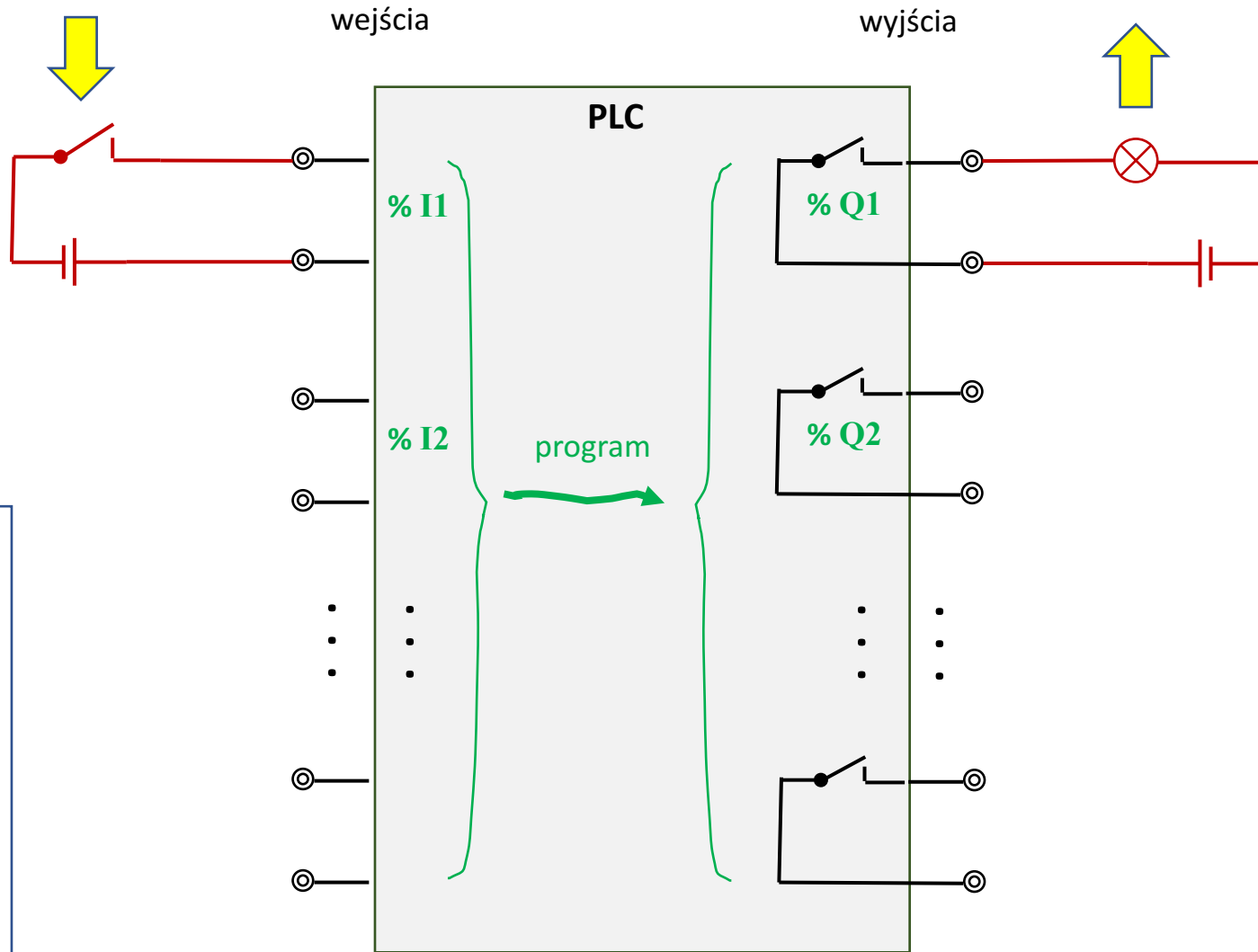




Algorytm sterowania (PROGRAM)

# UWAGA





Podłączanie wejść/wyjść 2. stanowych do PLC

# Kryteria doboru sterownika

- Rodzaj obiektu:
  - pojedyncza maszyna
  - linia technologiczna
- Wielkość obiektu określana liczbą punktów/kanałów:
  - liczba wejść/wyjść cyfrowych, napięcia operacyjne, obciążalność prądowa
  - liczba wejść/wyjść analogowych, wielkość pomiarowa, zakres
  - liczba modułów określona na podstawie liczby punktów/kanałów obsługiwanych przez sterownik
  - liczba sterowników, ich konfiguracja
- Główne zadania stawiane sterownikowi:
  - zbieranie danych o obiekcie, monitorowanie tych danych, wizualizacja
  - sterowanie w układzie otwartym, sterowanie w układzie zamkniętym
  - obliczenia statystyczne, raportowanie
- Warunki eksploatacyjne:
  - temperatura, wilgotność, wibracje, środowiska wybuchowe
- Kryteria ekonomiczne:
  - cena sterownika i osprzętu, możliwości rozbudowy systemu,
  - koszty projektu części elektrycznej i automatyki, koszty montażu i rozruchu
  - czas realizacji

Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

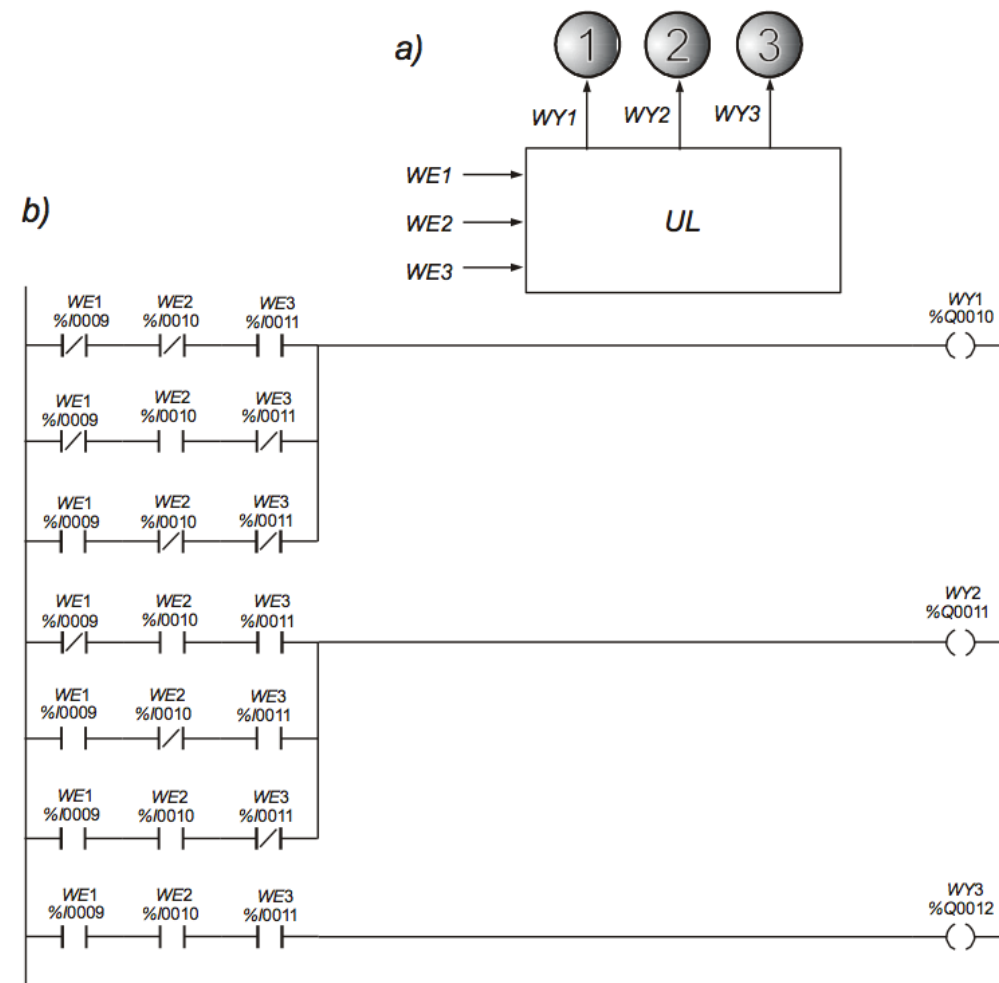
Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$

$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$

$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2

Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

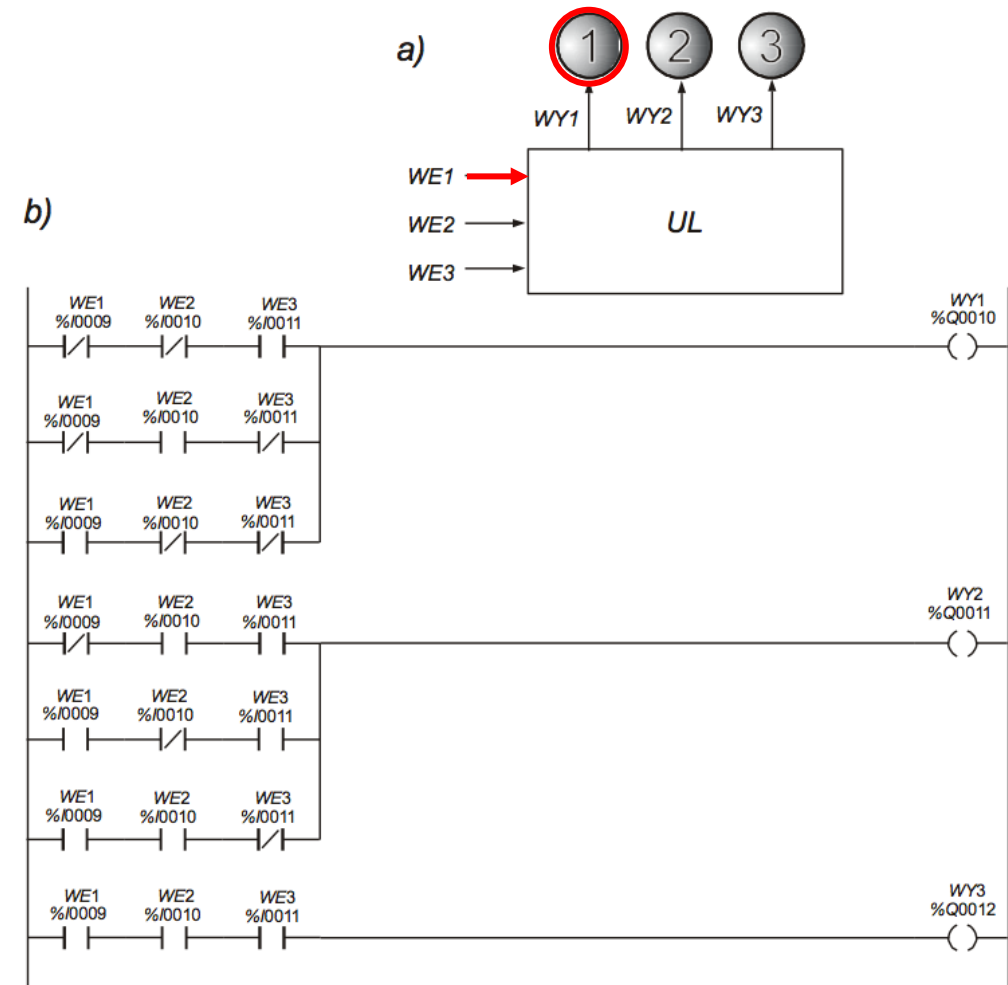
Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$

$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$

$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2



Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

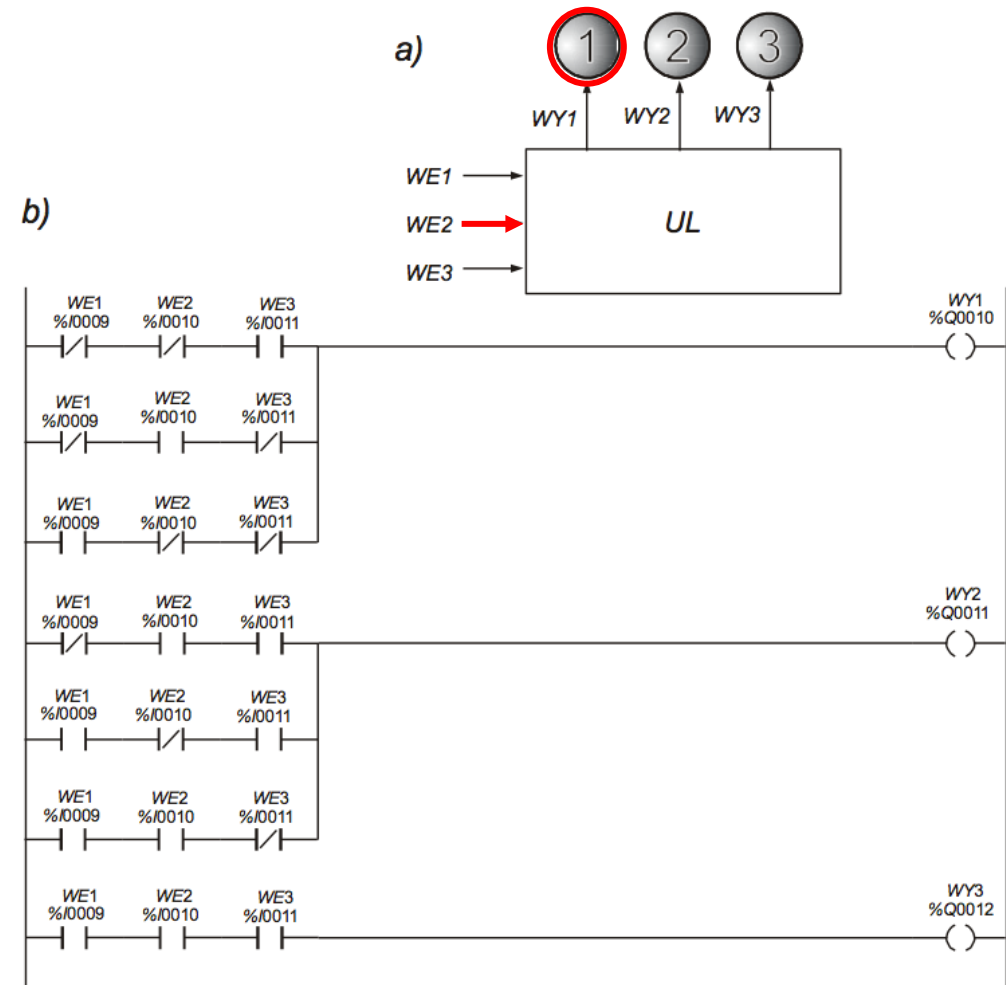
Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$

$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$

$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2

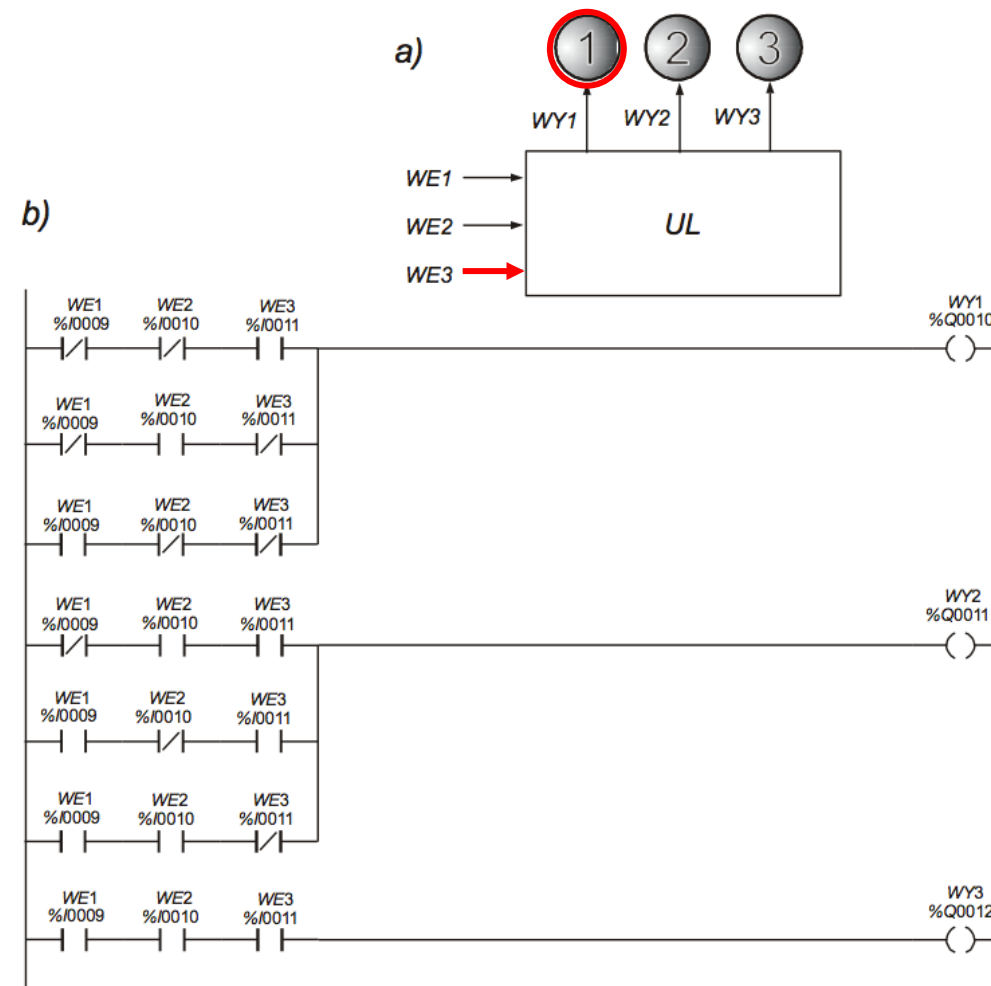
Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$
$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$
$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2

Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

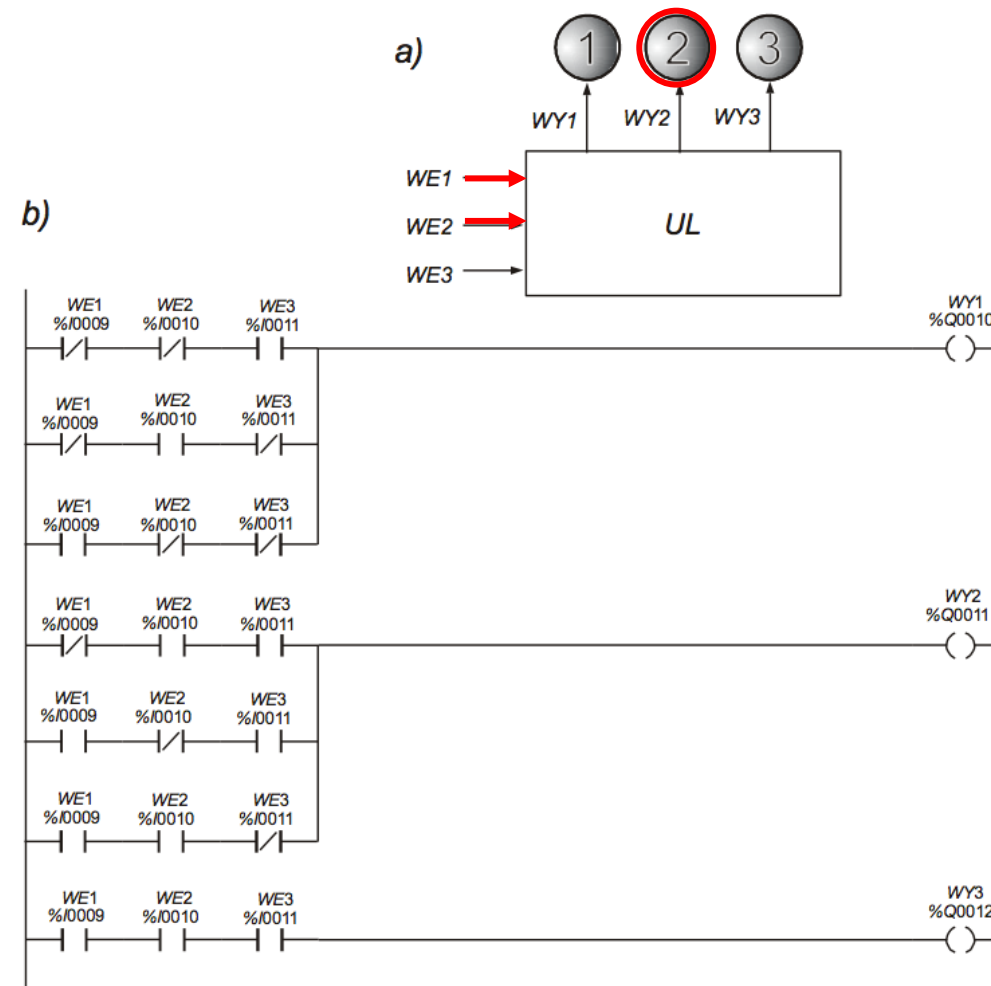
Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$

$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$

$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2

Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

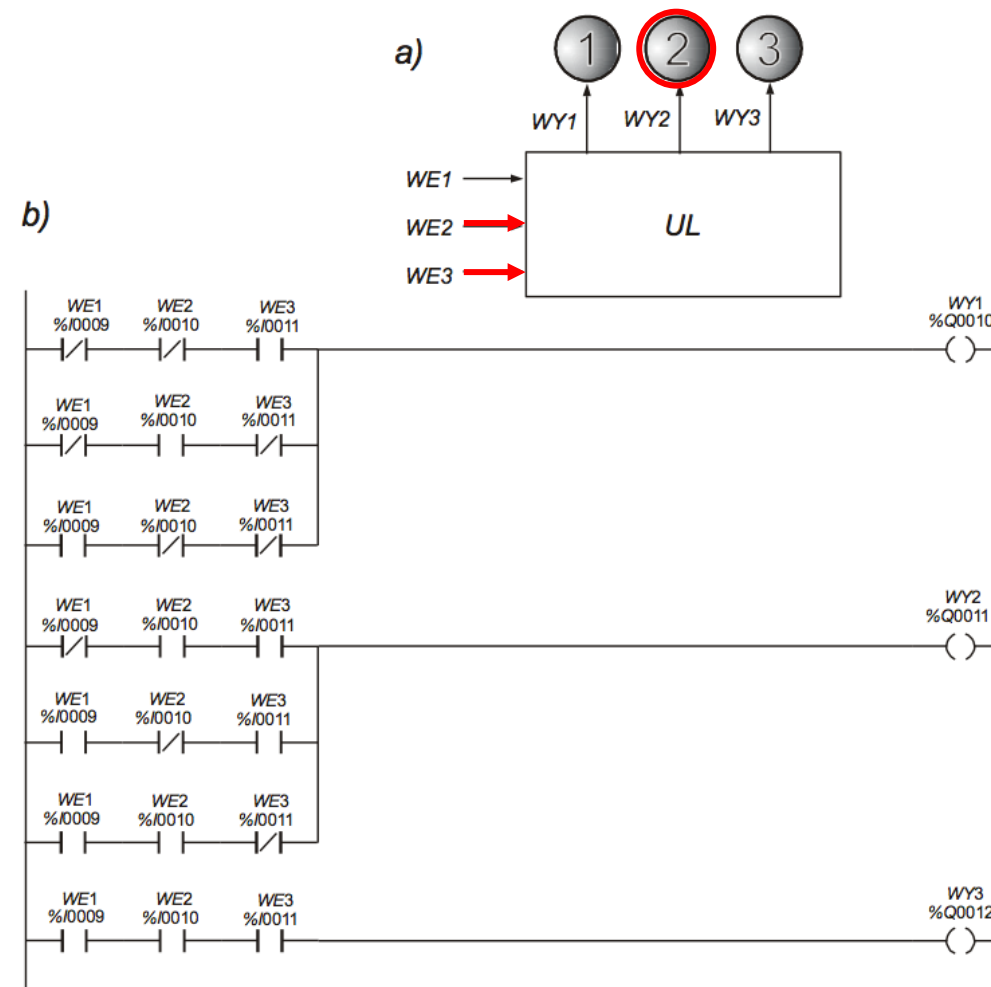
Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$

$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$

$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2

Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

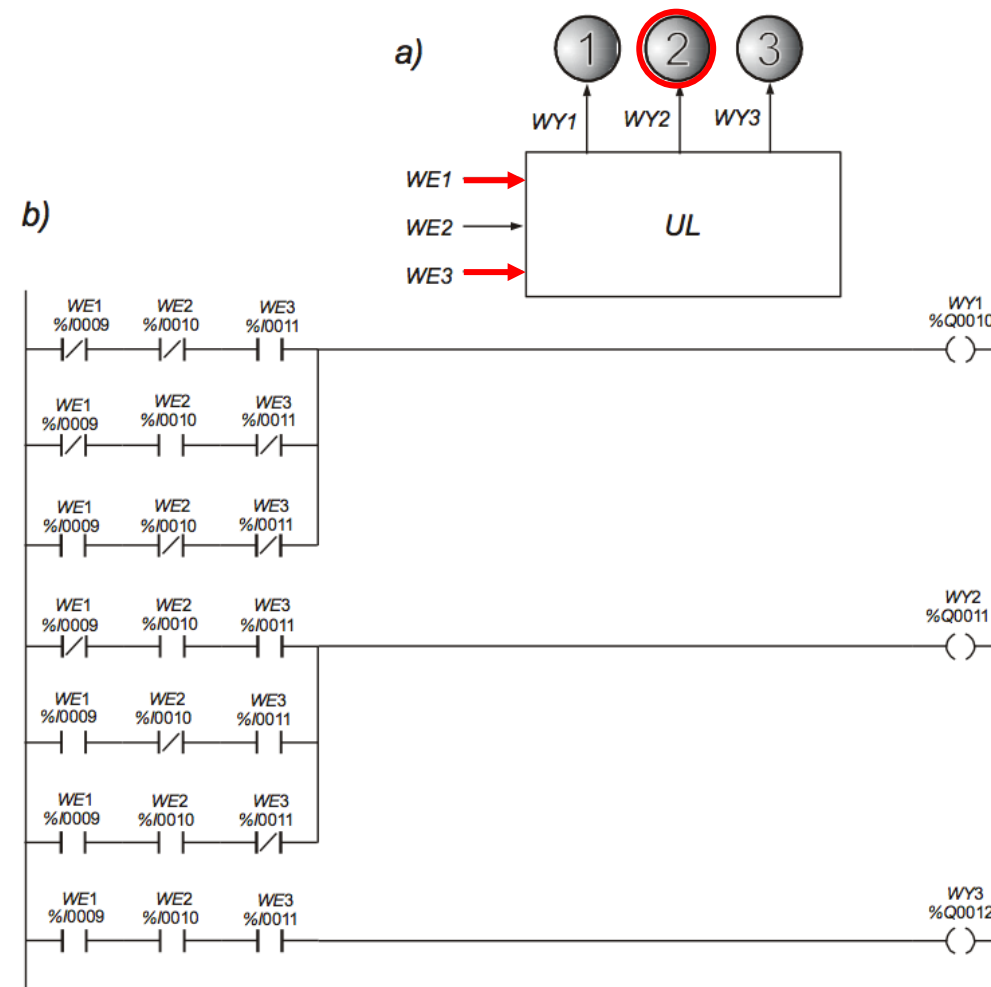
Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$

$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$

$$WY3 = WE1 \cdot WE2 \cdot WE3$$



PRZYKŁAD 2

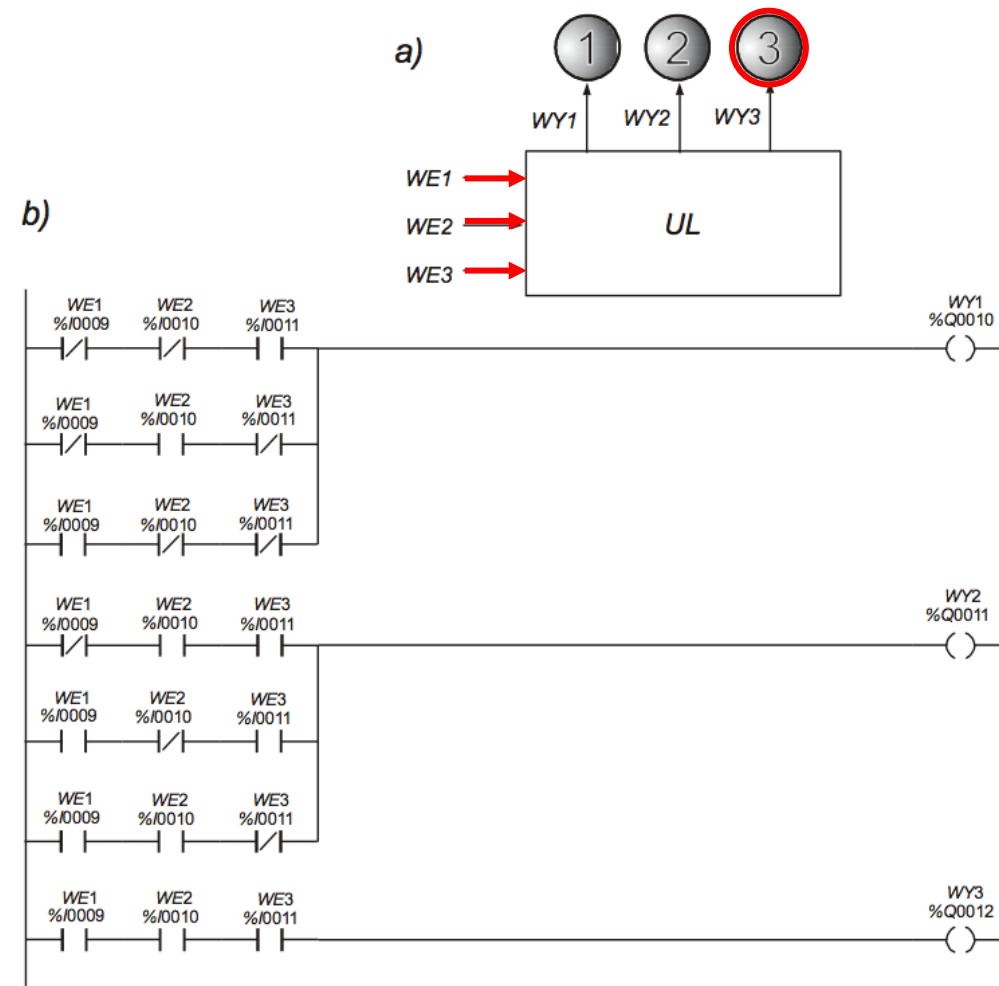
Zrealizować układ logiczny sygnalizujący liczbę „jedynek” wśród sygnałów  $WE1$ ,  $WE2$ ,  $WE3$ .

Jeśli jeden z trzech sygnałów wejściowych jest równy 1 powinna zostać podświetlona lampka z cyfrą 1 sterowana sygnałem  $WY1$ .

Jeśli na wejściu dwa sygnały przyjmują wartości równe 1 zapala się lampka z cyfrą 2.

Przy trzech „jedynekach” na wejściu zapala się lampka sterowana sygnałem  $WY3$ .

$$WY1 = \overline{WE1} \cdot \overline{WE2} \cdot WE3 + \overline{WE1} \cdot WE2 \cdot \overline{WE3} + WE1 \cdot \overline{WE2} \cdot \overline{WE3}$$
$$WY2 = \overline{WE1} \cdot WE2 \cdot WE3 + WE1 \cdot \overline{WE2} \cdot WE3 + WE1 \cdot WE2 \cdot \overline{WE3}$$
$$WY3 = WE1 \cdot WE2 \cdot WE3$$

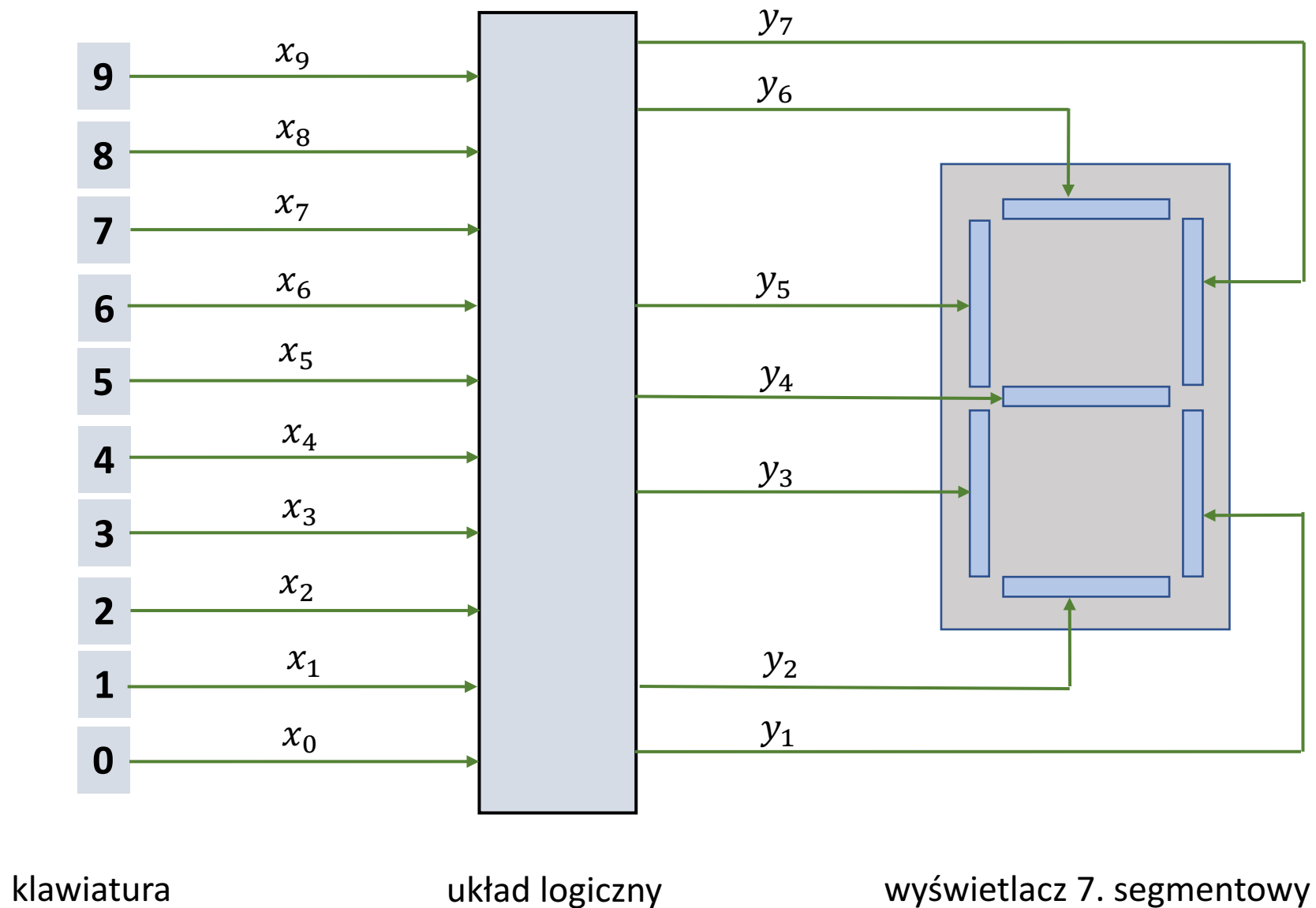


PRZYKŁAD 2

PRZYKŁAD 3

$$y_i = f(x_0, x_1, \dots, x_9)$$

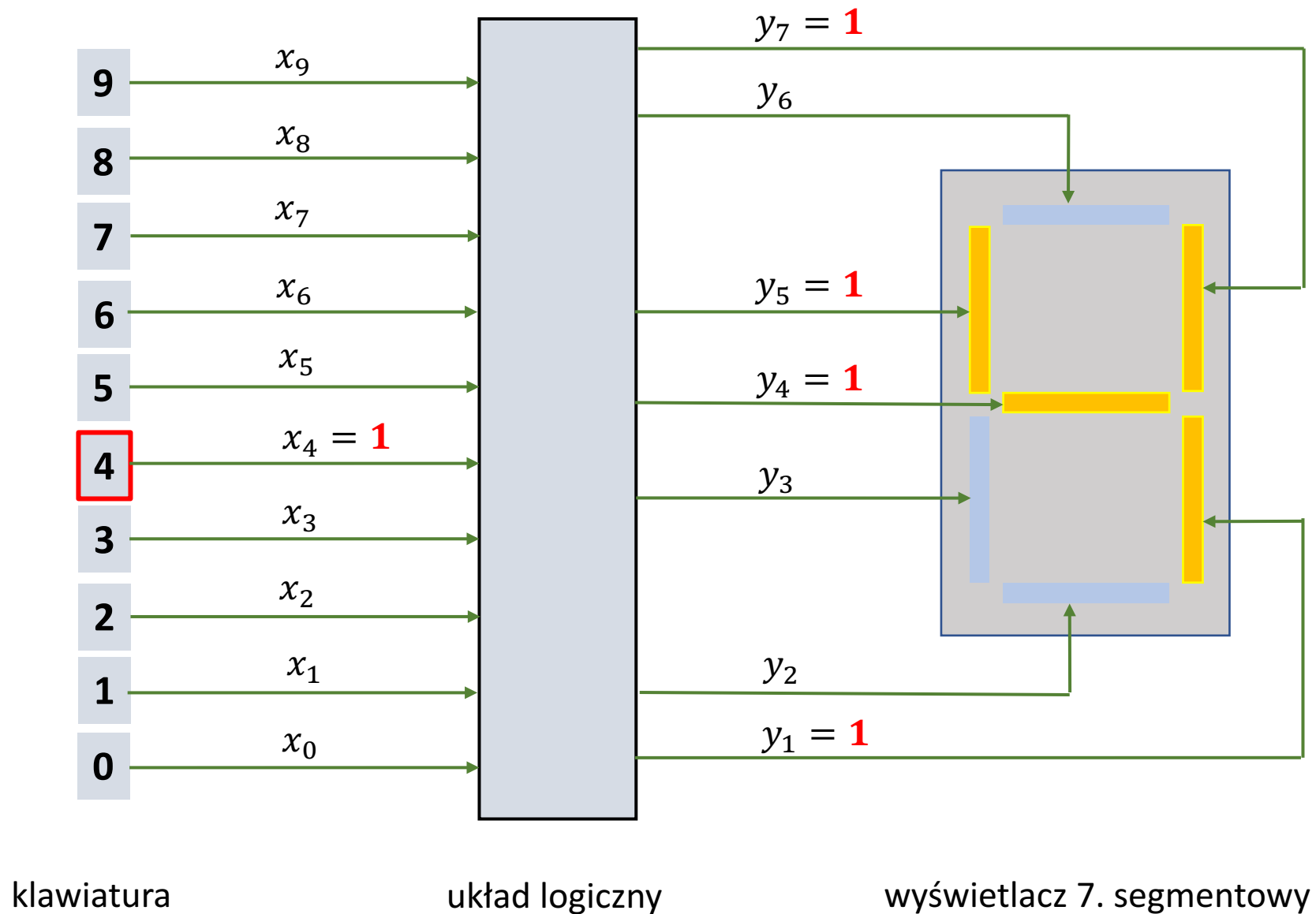
$$i = 1, 2, \dots, 7$$



PRZYKŁAD 3

$$y_i = f(x_0, x_1, \dots, x_9)$$

$$i = 1, 2, \dots, 7$$

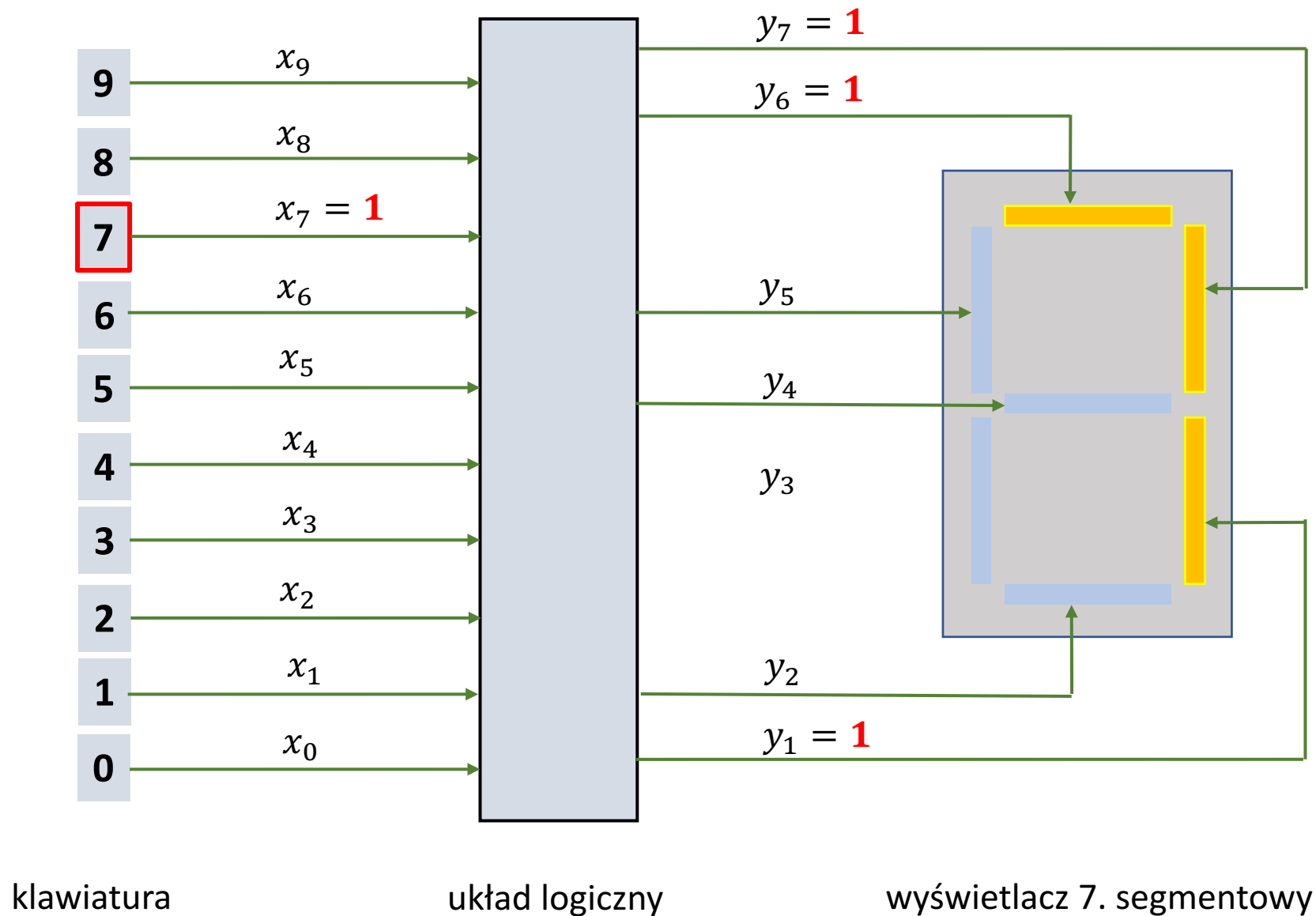




PRZYKŁAD 3

$$y_i = f(x_0, x_1, \dots, x_9)$$

$$i = 1, 2, \dots, 7$$

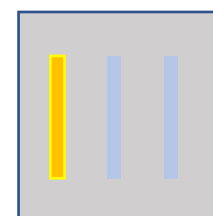
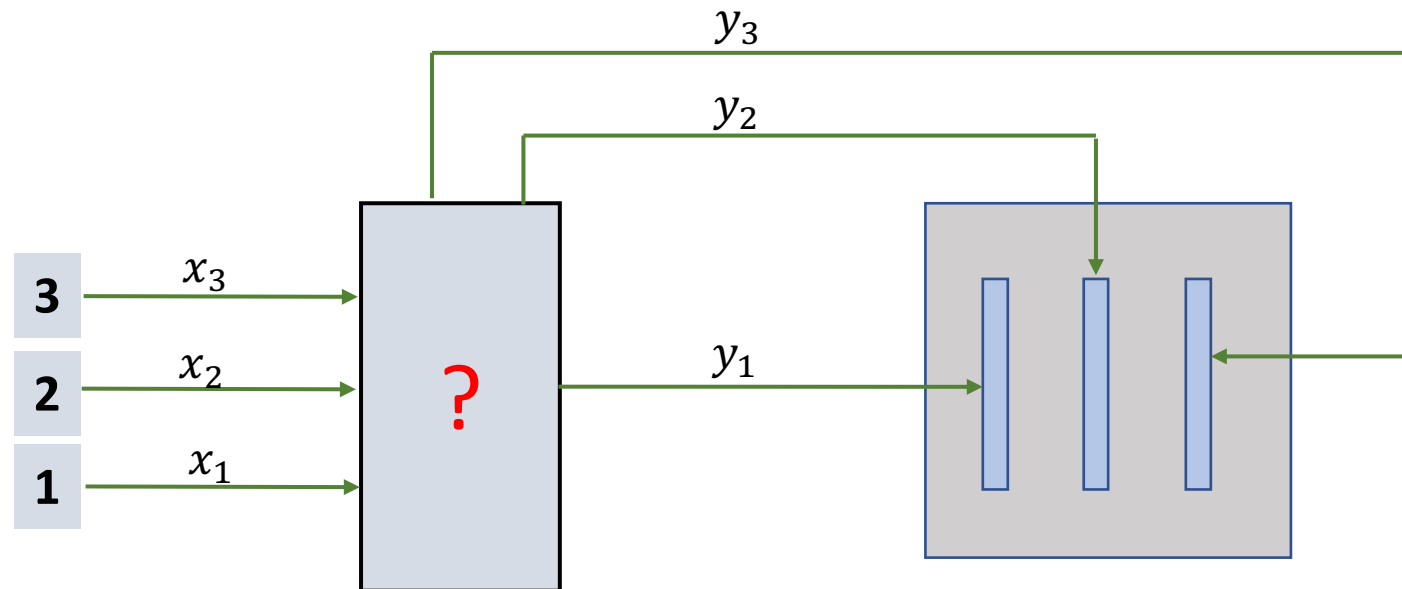


PRZYKŁAD 3

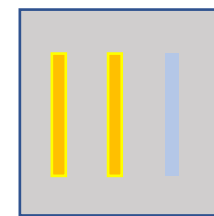
?

$$y_i = f(x_1, x_2, x_3)$$

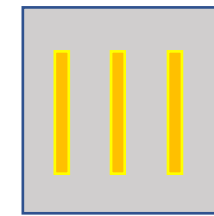
$$i = 1, 2, 3$$



1

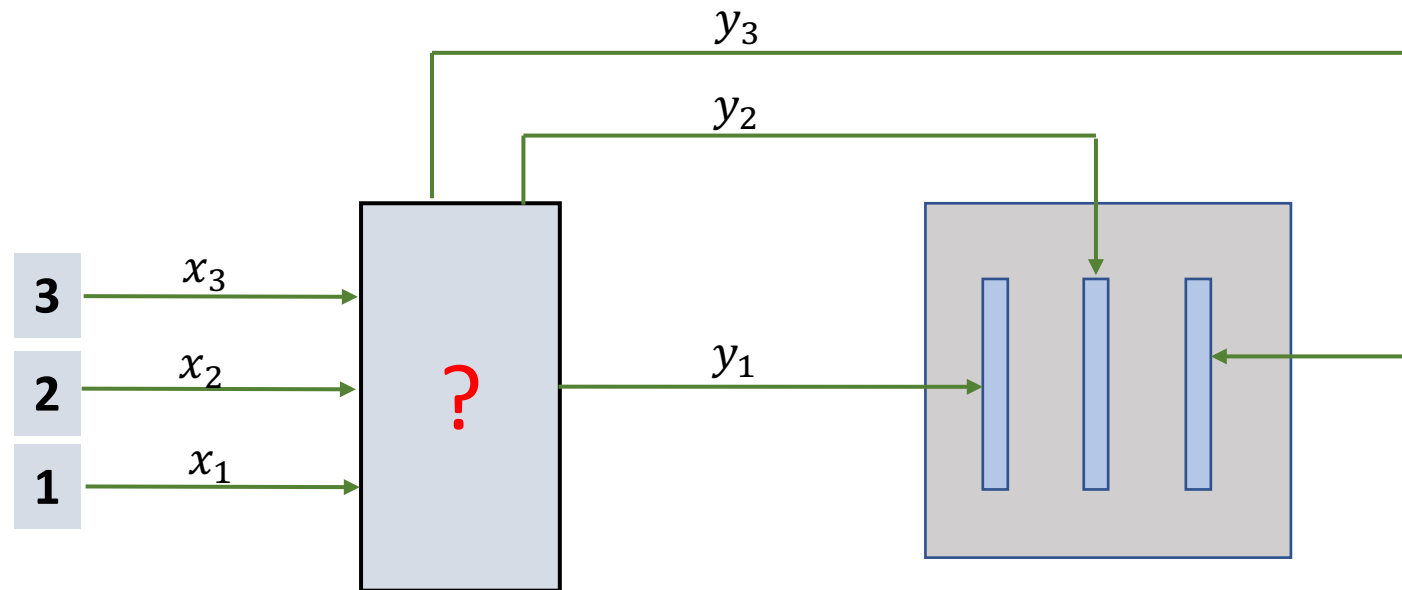


2



3

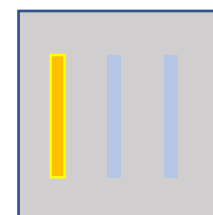
$x_1$	$x_2$	$x_3$	$y_1$	$y_2$	$y_3$
0	0	0	0	0	0
1	0	0	1	0	0
0	1	0	1	1	0
0	0	1	1	1	1
1	1	0	0	0	0
1	0	1	0	0	0
0	1	1	0	0	0
1	1	1	0	0	0



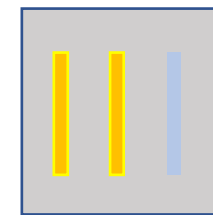
$$y_1 = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

$$y_2 = x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

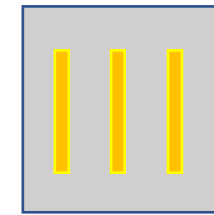
$$y_3 = x_1 \cdot x_2 \cdot x_3$$



1



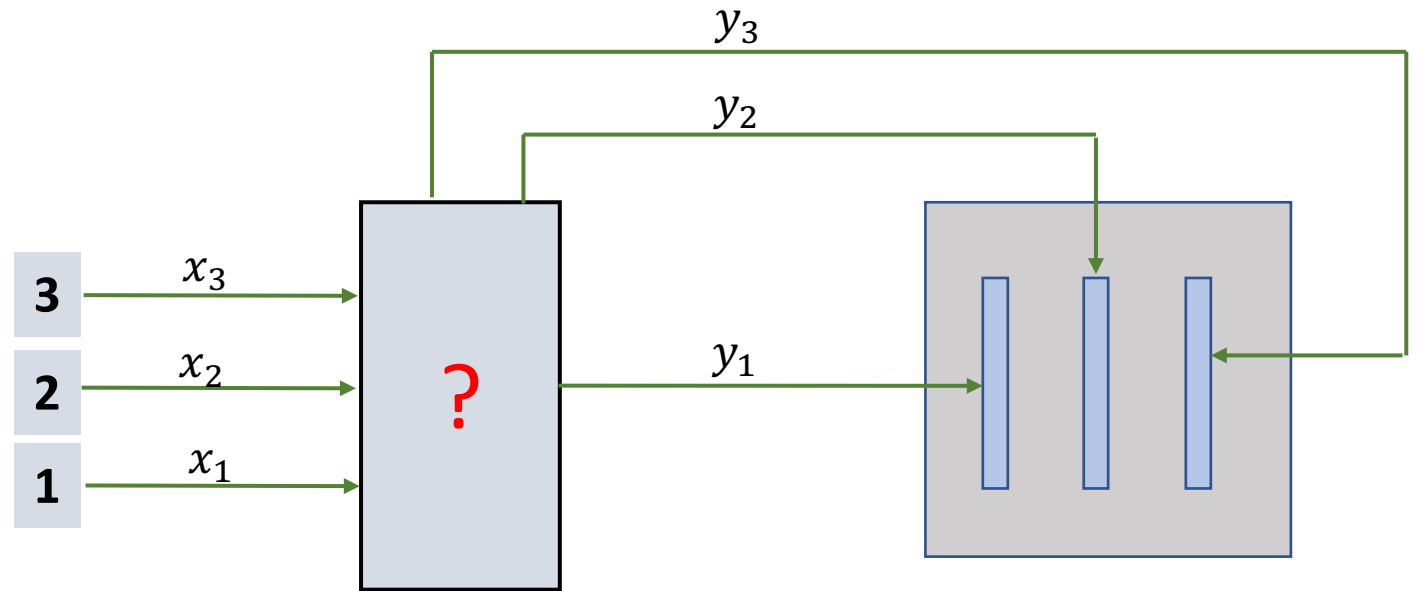
2



3

PRZYKŁAD 3

$x_1$	$x_2$	$x_3$	$y_1$	$y_2$	$y_3$
0	0	0	0	0	0
1	0	0	1	0	0
0	1	0	1	1	0
0	0	1	1	1	1
1	1	0	0	0	0
1	0	1	0	0	0
0	1	1	0	0	0
1	1	1	0	0	0



$$y_1 = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

$$y_2 = x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

$$y_3 = x_1 \cdot x_2 \cdot x_3$$

$$y_1 = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + y_2 + y_3$$

$$y_2 = x_1 \cdot x_2 \cdot \bar{x}_3 + y_3$$

$$y_3 = x_1 \cdot x_2 \cdot x_3$$

PRZYKŁAD 3

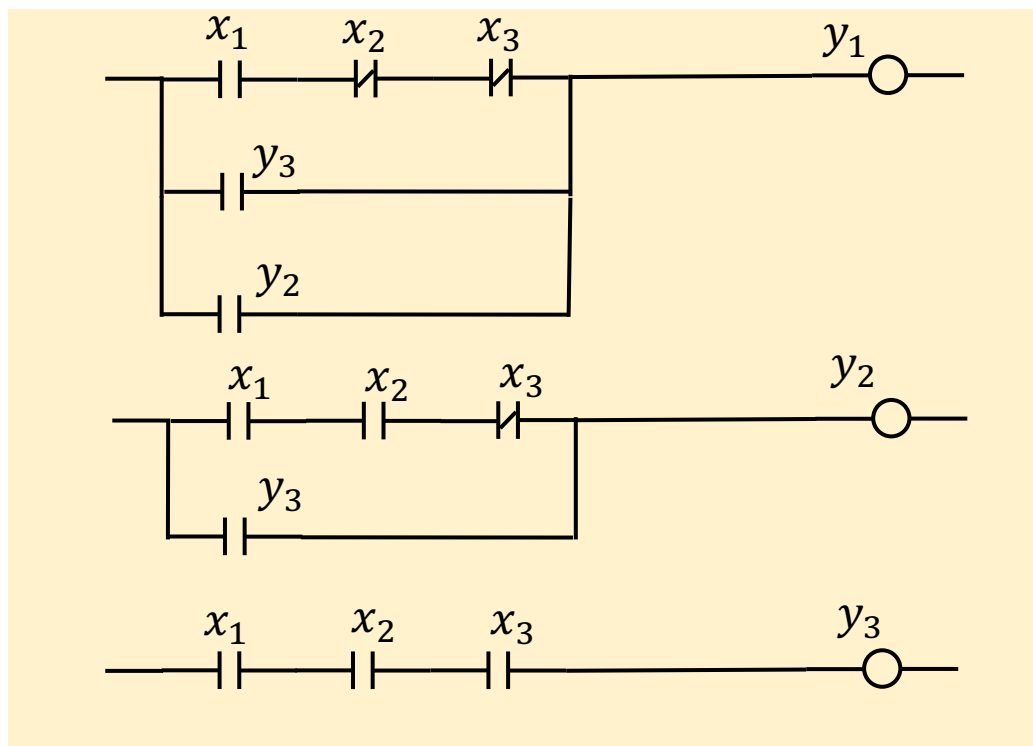
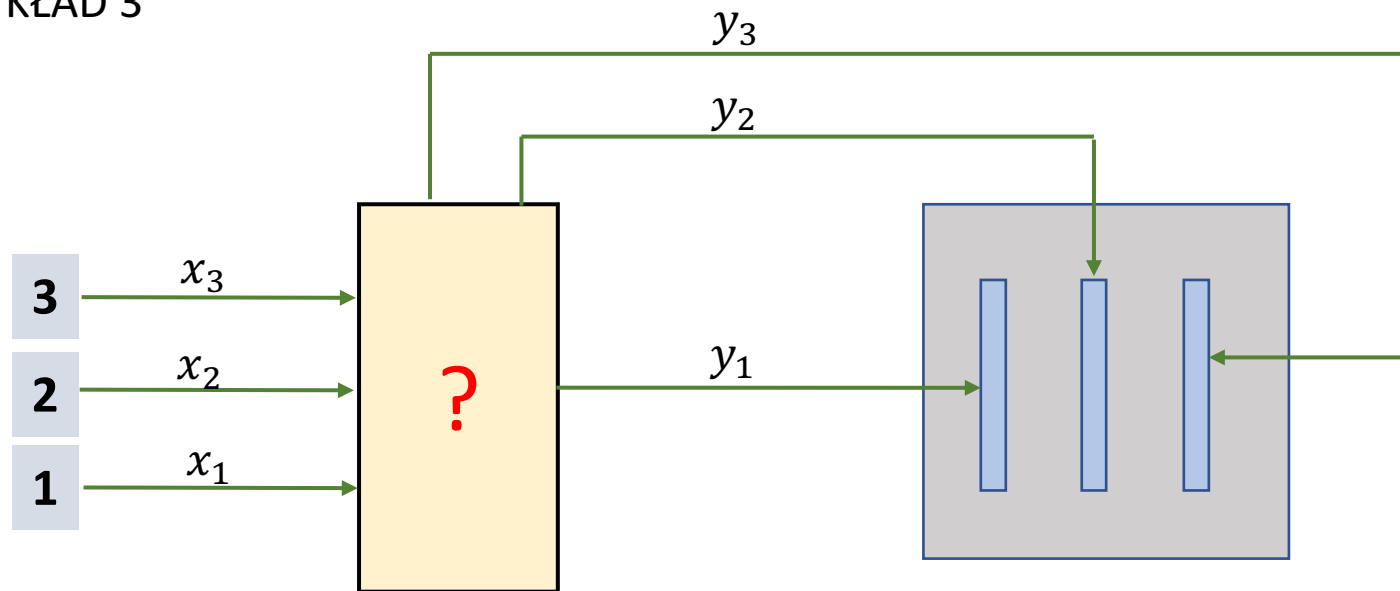
### PRZYKŁAD 3

$x_1$	$x_2$	$x_3$	$y_1$	$y_2$	$y_3$
0	0	0	0	0	0
1	0	0	1	0	0
0	1	0	1	1	0
0	0	1	1	1	1
1	1	0	0	0	0
1	0	1	0	0	0
0	1	1	0	0	0
1	1	1	0	0	0

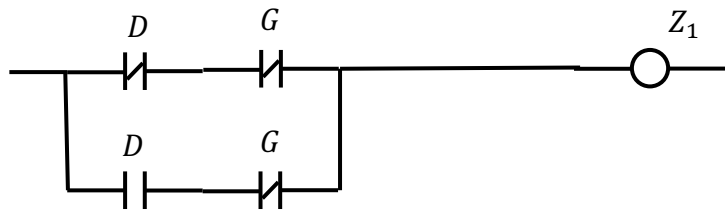
$$y_1 = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + y_2 + y_3$$

$$y_2 = x_1 \cdot x_2 \cdot \bar{x}_3 + y_3$$

$$y_3 = x_1 \cdot x_2 \cdot x_3$$



$$Z_1 = \bar{D} \cdot \bar{G} + D \cdot \bar{G}$$



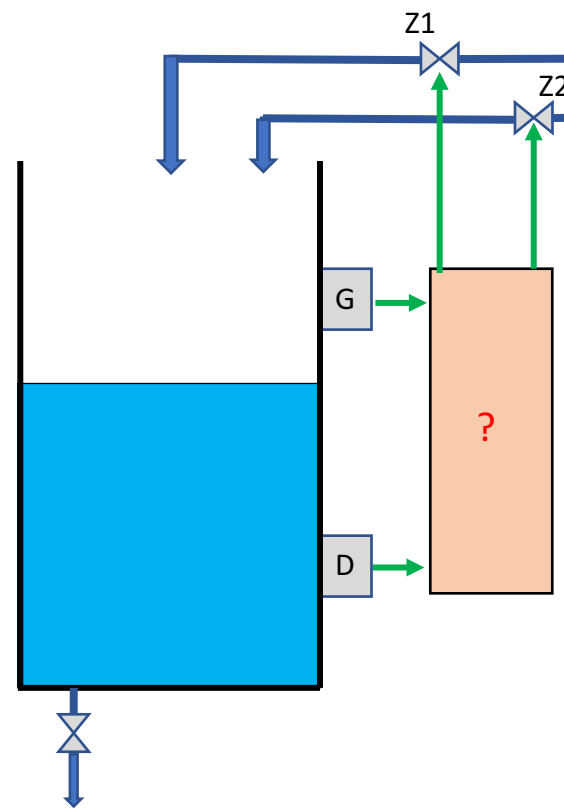
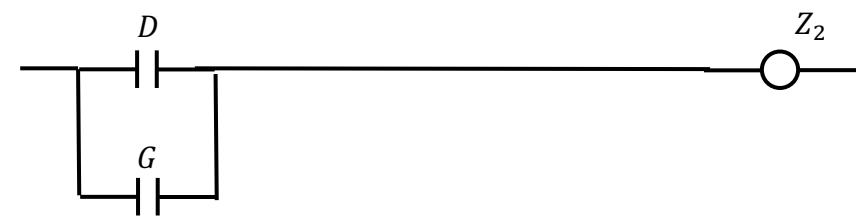
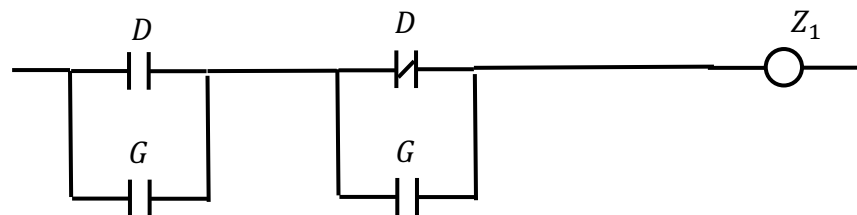
$$Z_2 = \bar{D} \cdot \bar{G}$$

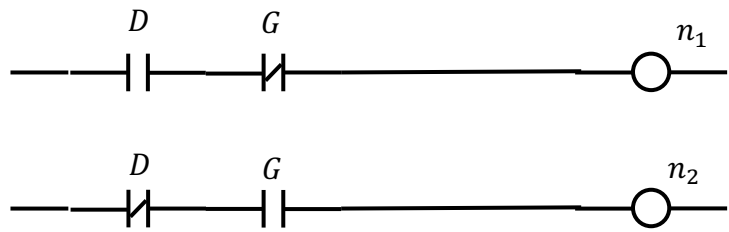


D	G	Z <sub>1</sub>	Z <sub>2</sub>
0	0	1	1
1	0	1	0
1	1	0	0
0	1	nie występuje	

$$Z_1 = (D + G) \cdot (\bar{D} + G)$$

$$Z_2 = (D + G)$$



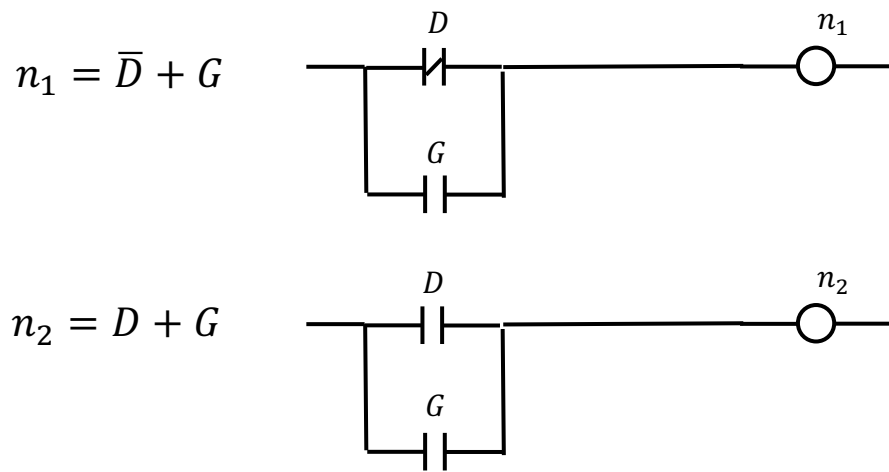


$$n_1 = D \cdot \bar{G}$$

$$n_2 = \bar{D} \cdot G$$

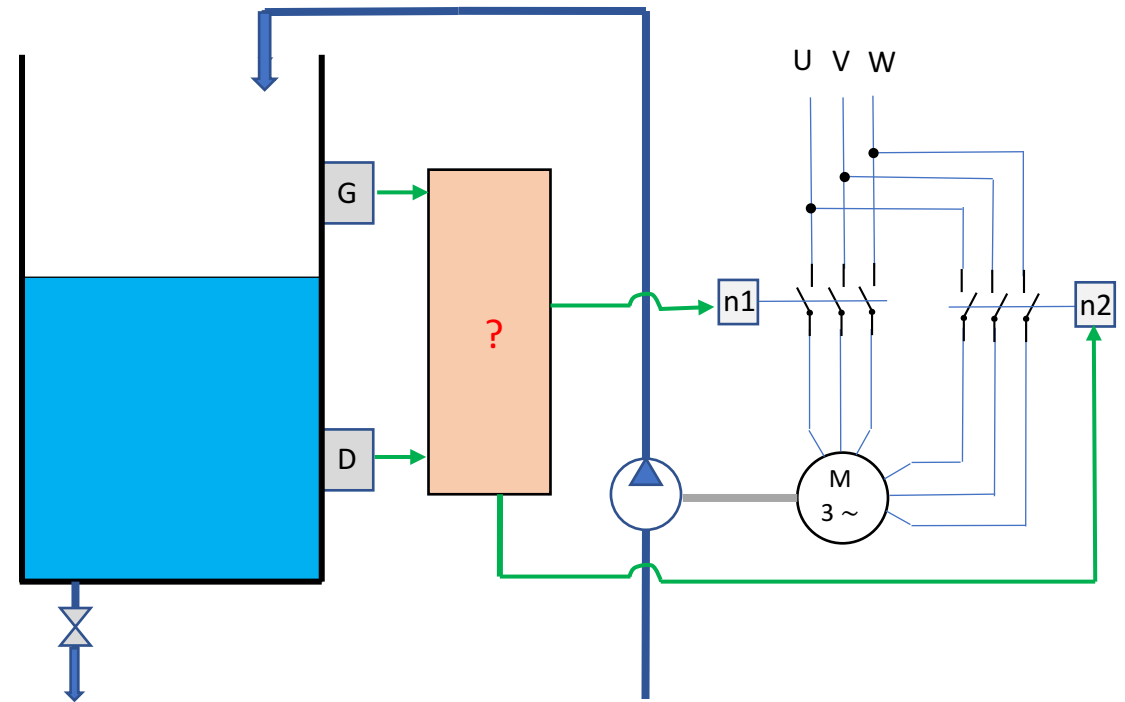
$$n_2 = \bar{D}$$

D	G	n <sub>1</sub>	n <sub>2</sub>
0	0	0	1
1	0	1	0
1	1	0	0
0	1	nie występuje	



$$n_1 = \bar{D} + G$$

$$n_2 = D + G$$

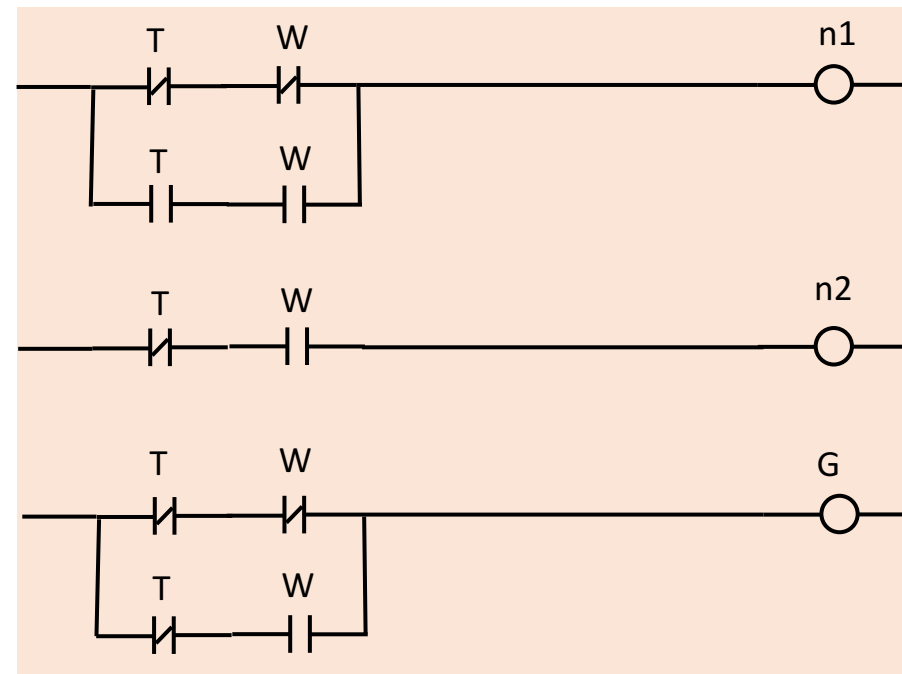
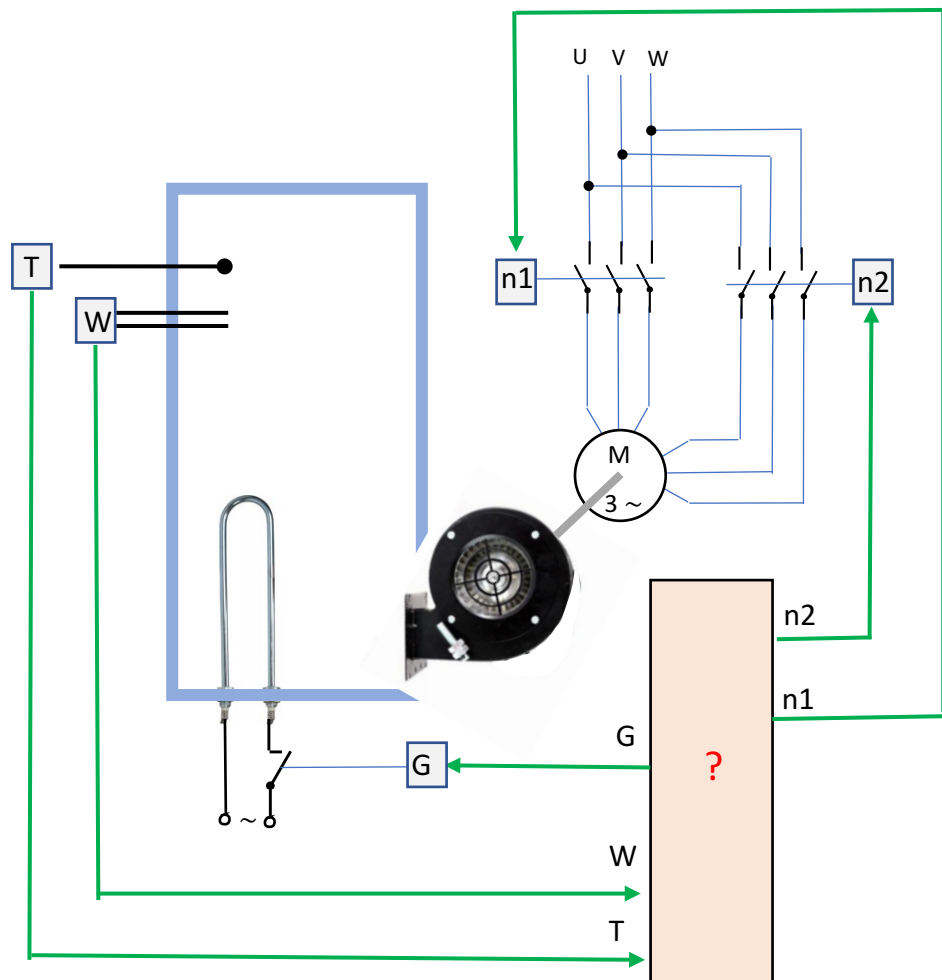


T	W	n1	n2	G
0	0	1	0	1
1	0	0	0	0
0	1	0	1	1
1	1	1	0	0

$$n1 = \bar{T} \cdot \bar{W} + T \cdot W$$

$$n2 = \bar{T} \cdot W$$

$$G = \bar{T} \cdot \bar{W} + \bar{T} \cdot W$$



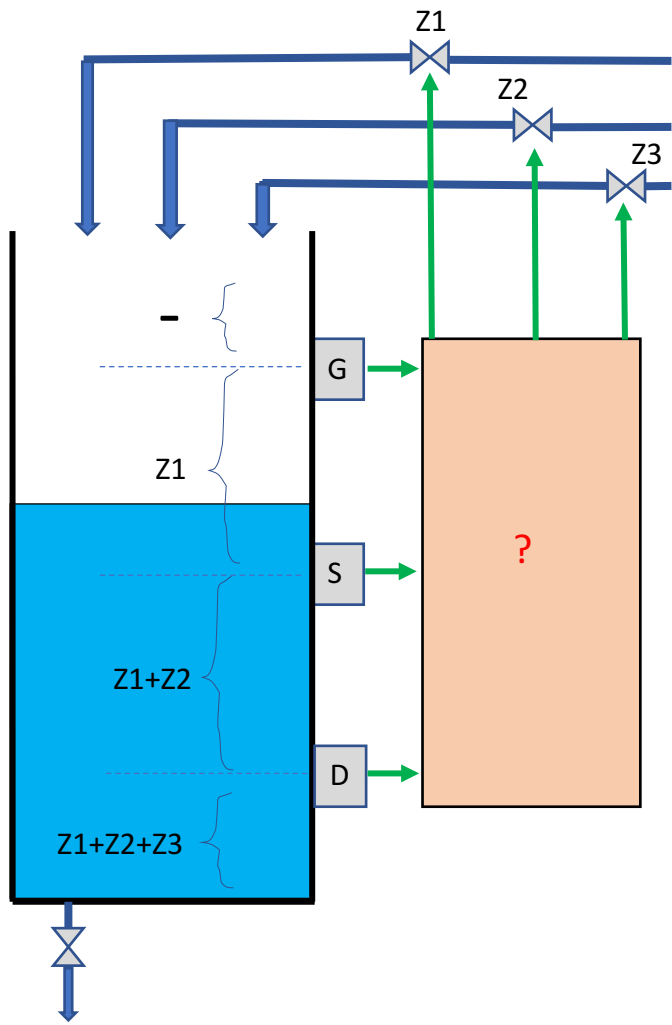
	$T_n (T=0)$	$T_w (T=1)$
$W_m (W=0)$	n1, G	
$W_d (W=1)$	n2, G	n1

PRZYKŁAD 6



wilgoć powoduje zmiany pojemności w czujniku sondy pomiarowej.





D	S	G	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>
0	0	0	1	1	1
1	0	0	1	1	0
0	1	0	0	0	0
0	0	1	0	0	0
1	1	0	1	0	0
1	0	1	0	0	0
0	1	1	0	0	0
1	1	1	0	0	0

$$Z1 = \bar{D} \cdot \bar{S} \cdot \bar{G} + D \cdot \bar{S} \cdot \bar{G} + D \cdot S \cdot \bar{G}$$

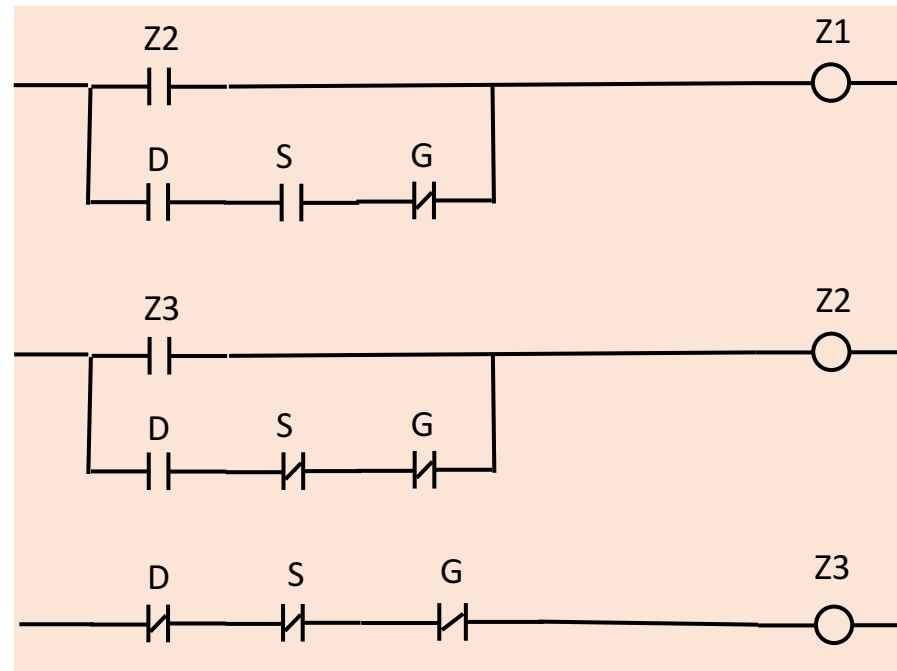
$$Z2 = \bar{D} \cdot \bar{S} \cdot \bar{G} + D \cdot \bar{S} \cdot \bar{G}$$

$$Z3 = \bar{D} \cdot \bar{S} \cdot \bar{G}$$

$$Z1 = Z2 + D \cdot S \cdot \bar{G}$$

$$Z2 = Z3 + D \cdot \bar{S} \cdot \bar{G}$$

$$Z3 = \bar{D} \cdot \bar{S} \cdot \bar{G}$$



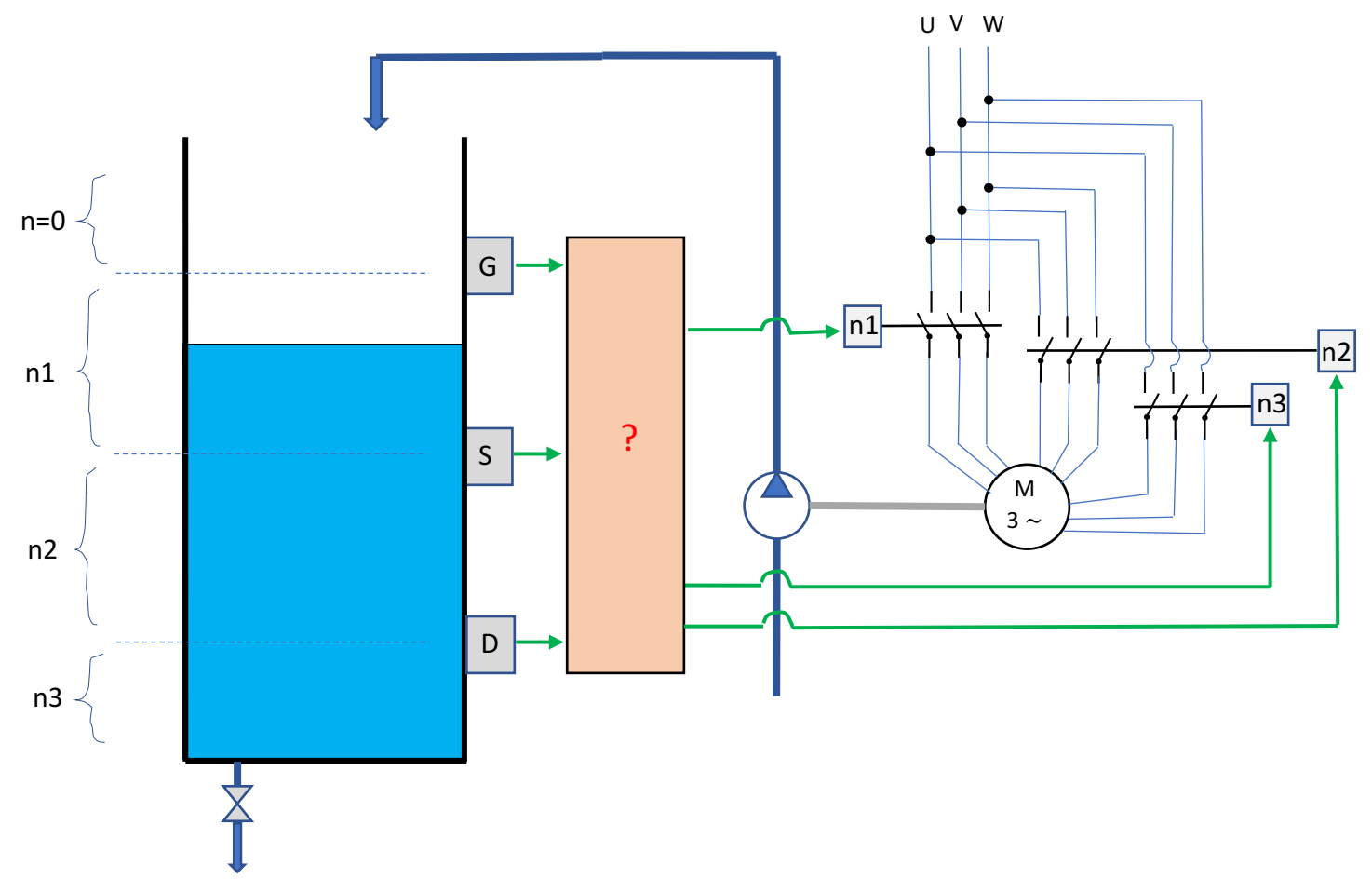
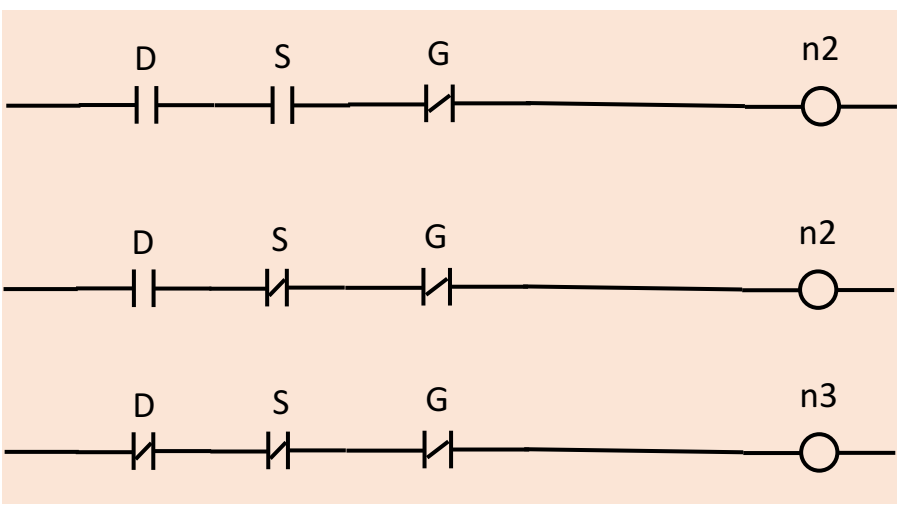
PRZYKŁAD 7

D	S	G	n <sub>1</sub>	n <sub>2</sub>	n <sub>3</sub>
0	0	0	0	0	1
1	0	0	0	1	0
0	1	0	0	0	0
0	0	1	0	0	0
1	1	0	1	0	0
1	0	1	0	0	0
0	1	1	0	0	0
1	1	1	0	0	0

$$n3 = \bar{D} \cdot \bar{S} \cdot \bar{G}$$

$$n2 = D \cdot \bar{S} \cdot \bar{G}$$

$$n1 = D \cdot S \cdot \bar{G}$$

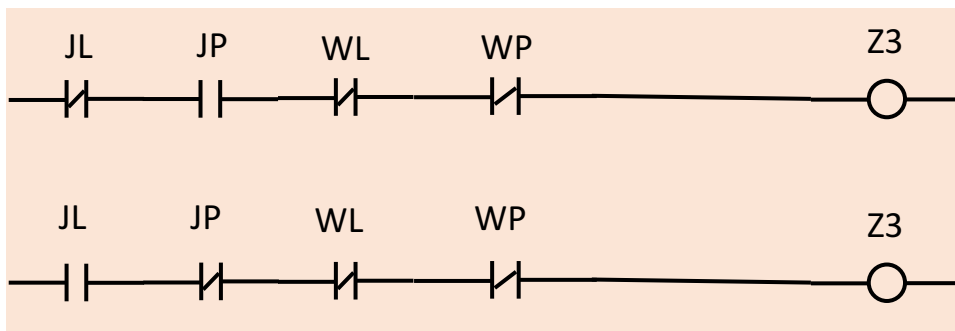
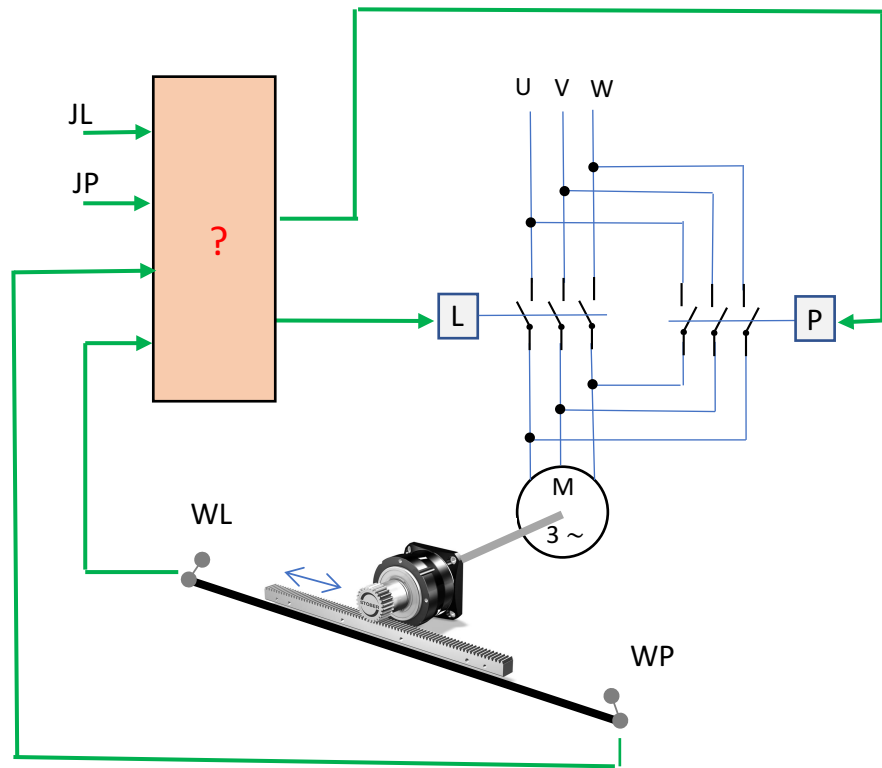


PRZYKŁAD 8

JL	JP	W <sub>L</sub>	W <sub>P</sub>	L	P
0	0	0	0	0	0
1	0	0	0	1	0
0	1	0	0	0	1
0	0	1	0	0	0
0	0	0	1	0	0
1	1	0	0	0	0
0	1	1	0	0	0
0	0	1	1	0	0
1	0	1	0	0	0
1	0	0	1	0	0
0	1	0	1	0	0
1	1	1	0	0	0
1	1	0	1	0	0
1	0	1	1	0	0
0	1	1	1	0	0
1	1	1	1	0	0

$$L = JL \cdot \overline{JP} \cdot \overline{WL} \cdot \overline{WP}$$

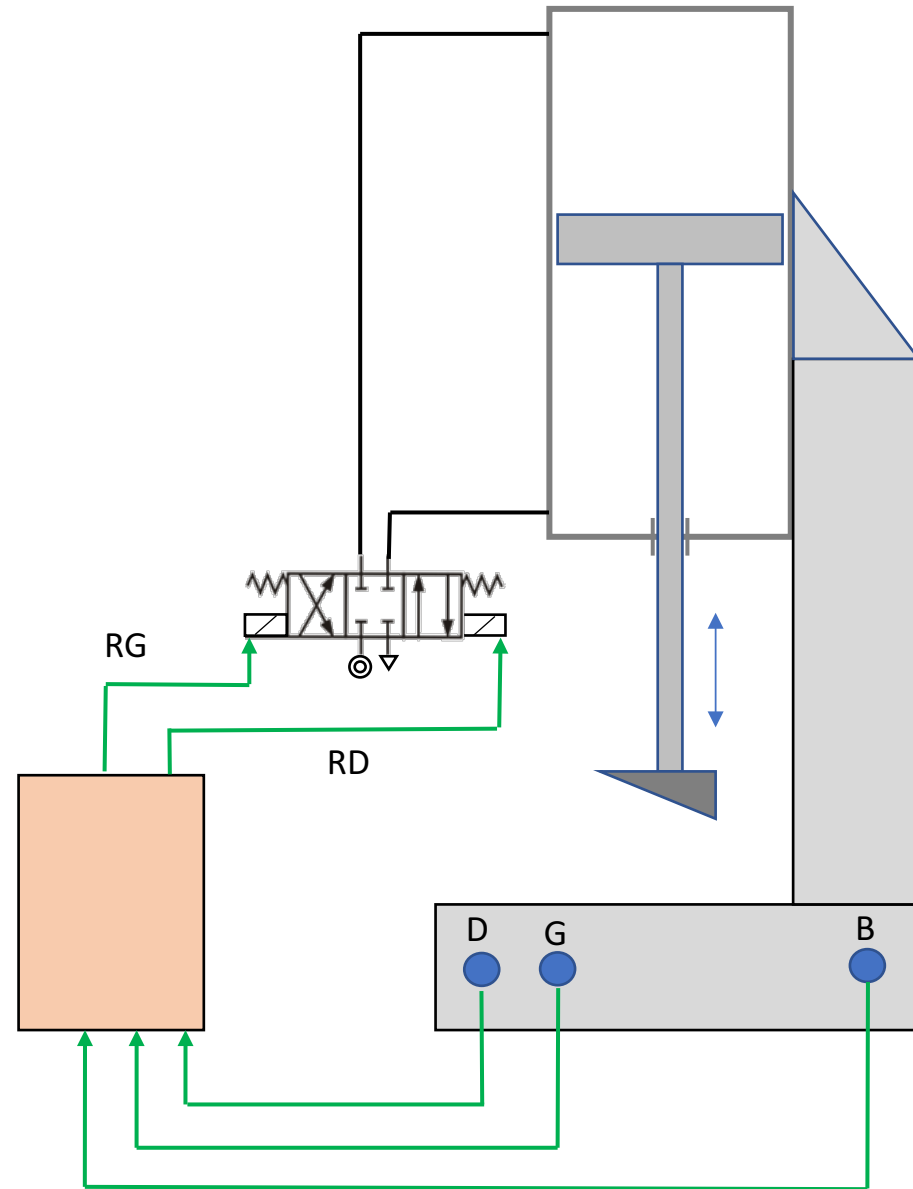
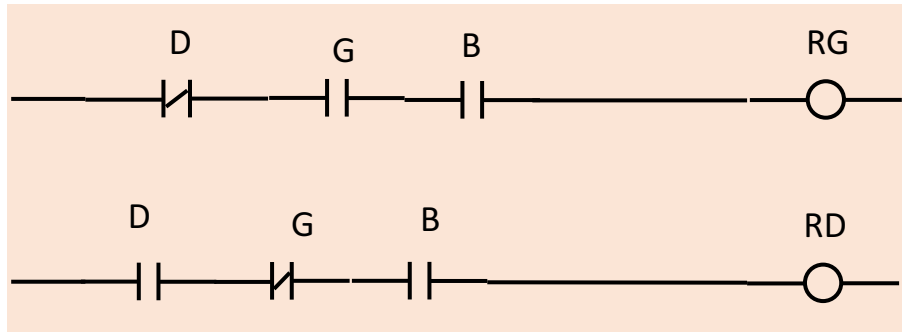
$$P = \overline{JL} \cdot JP \cdot \overline{WL} \cdot \overline{WP}$$



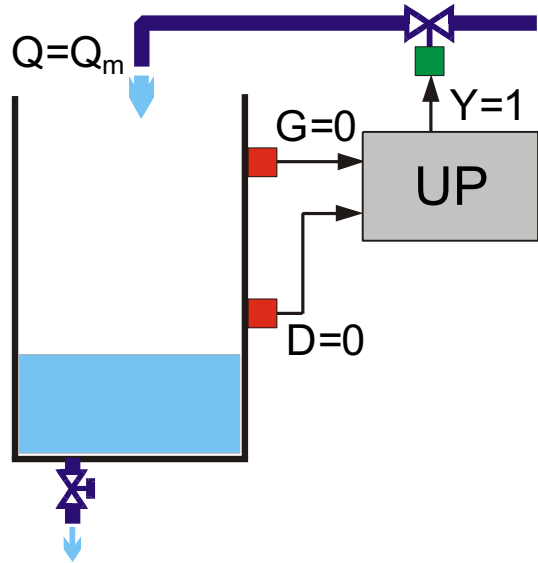
PRZYKŁAD 9

# PRZYKŁAD 10

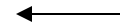
D	G	B	RG	RD
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
1	1	0	0	0
1	0	1	0	1
0	1	1	1	0
1	1	1	0	0



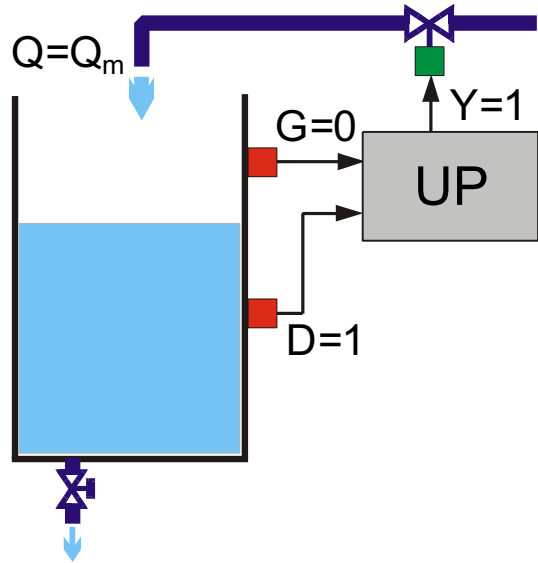
# UKŁADY SEKWENCYJNE



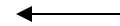
D	G	Y
0	0	1



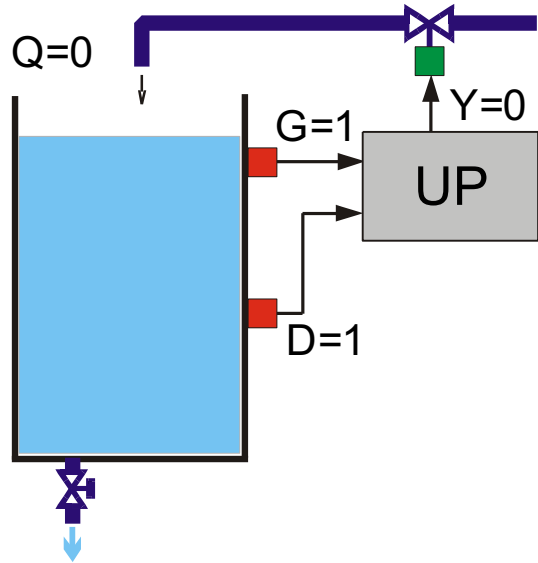
# UKŁADY SEKWENCYJNE



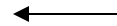
D	G	Y
0	0	1
1	0	1



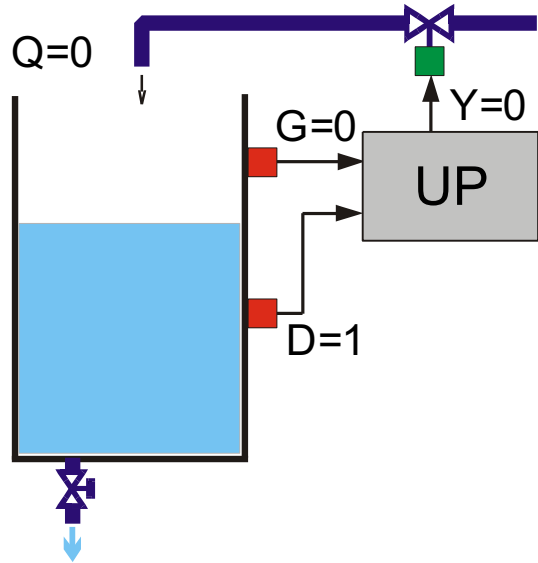
# UKŁADY SEKWENCYJNE



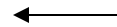
D	G	Y
0	0	1
1	0	1
1	1	0



# UKŁADY SEKWENCYJNE

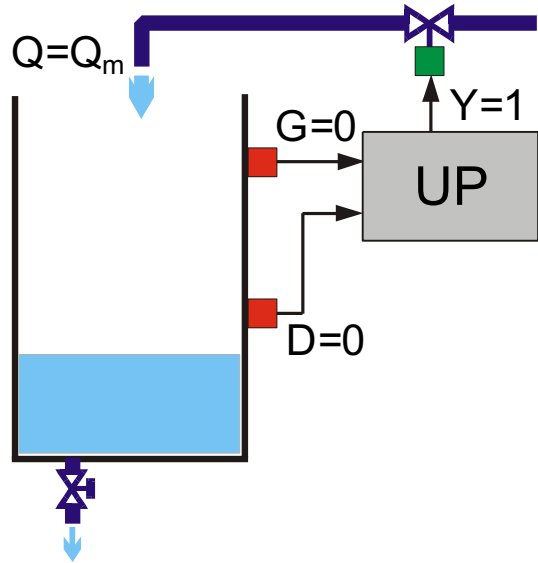


D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

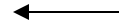




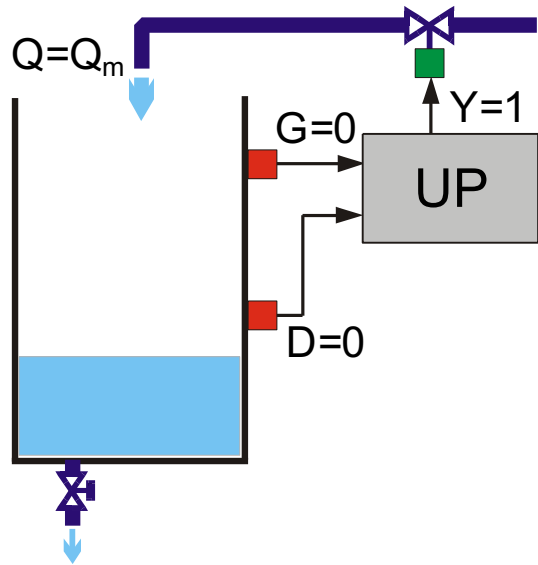
# UKŁADY SEKWENCYJNE



D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0



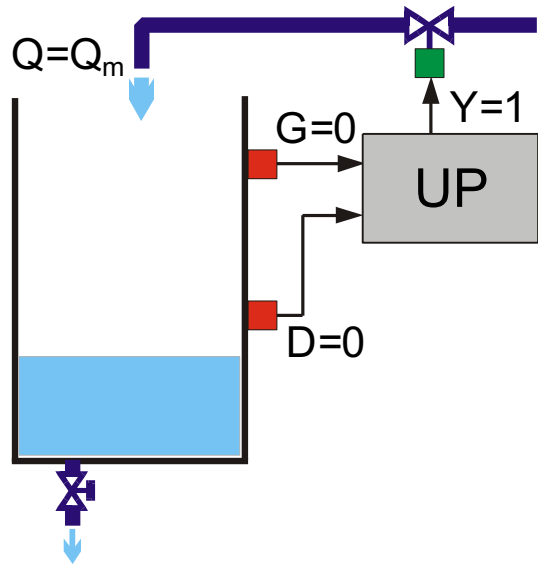
# UKŁADY SEKWENCYJNE



D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

Stan niejednoznaczny

# UKŁADY SEKWENCYJNE

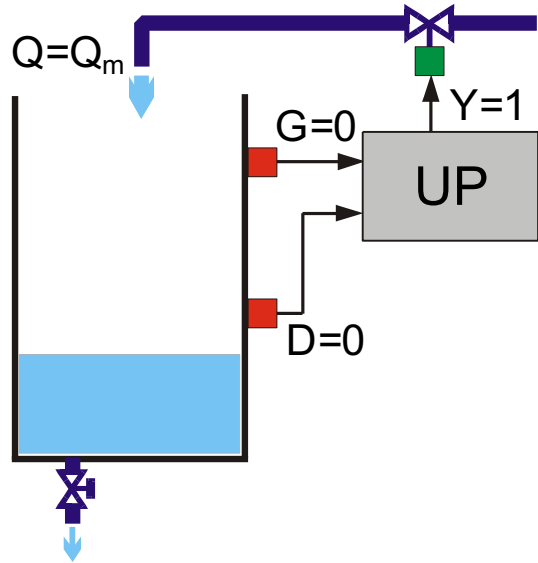


D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

Stan niejednoznaczny

Pamięć stanu poprzedniego

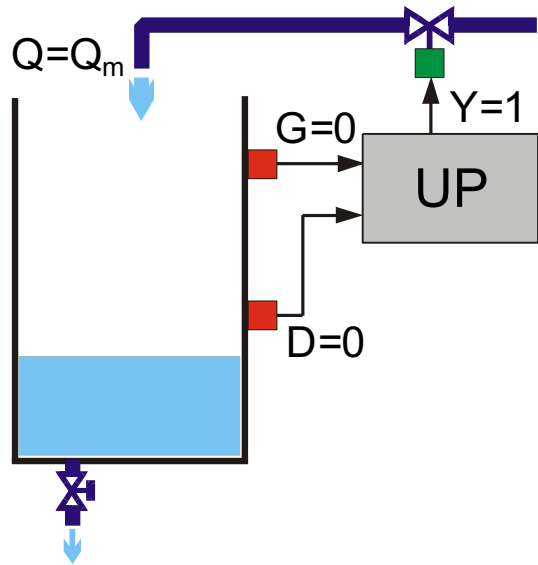
# UKŁADY SEKWENCYJNE



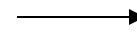
D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

D	G	Y
0	0	
1	0	
0	1	
1	1	

# UKŁADY SEKWENCYJNE



D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

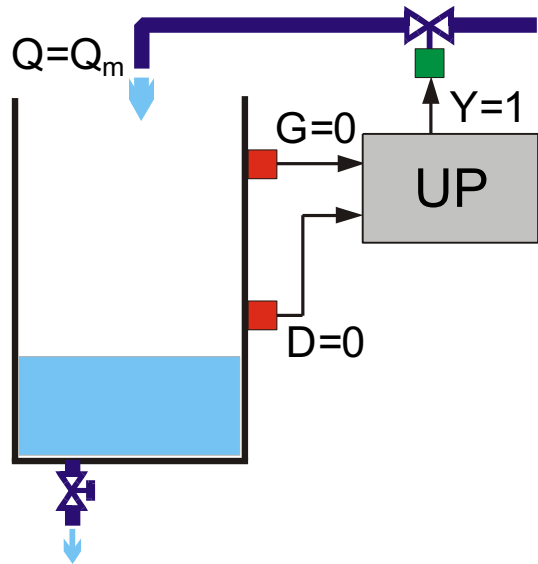


D	G	Y
0	0	1
1	0	
0	1	
1	1	



**Stan jednoznaczny**

# UKŁADY SEKWENCYJNE



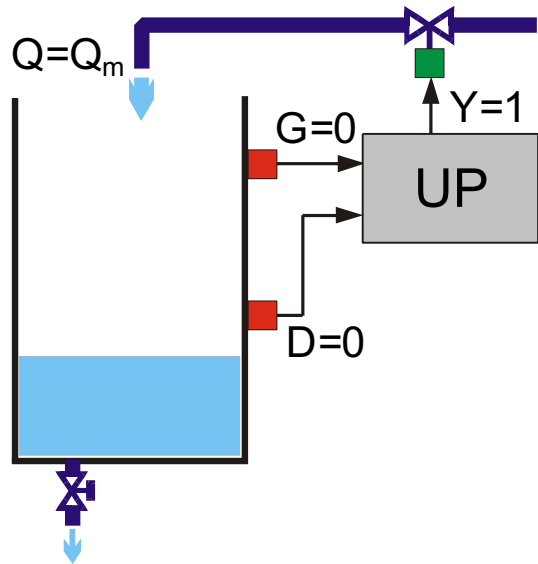
D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

D	G	Y
0	0	1
1	0	Y*
0	1	
1	1	



Pamięć stanu poprzedniego

# UKŁADY SEKWENCYJNE



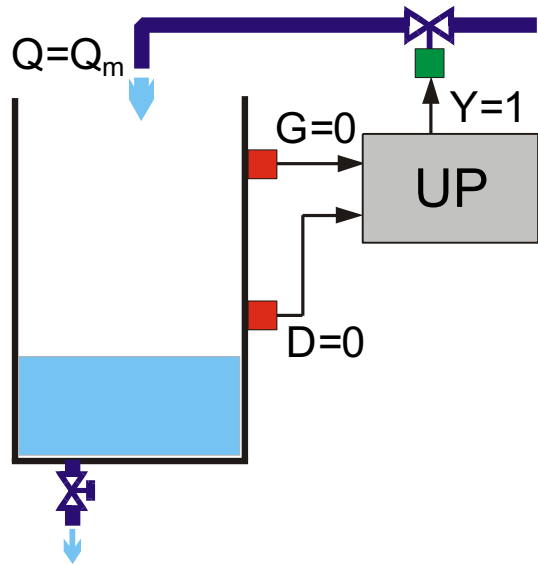
D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

D	G	Y
0	0	1
1	0	Y*
0	1	0
1	1	



**Stan nieistniejący**

# UKŁADY SEKWENCYJNE



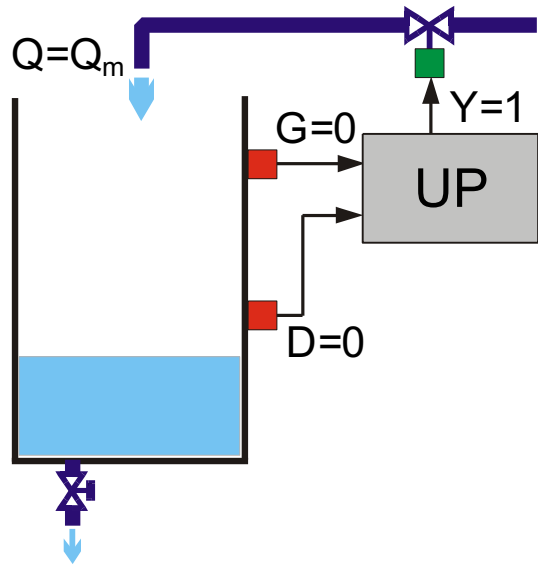
D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

D	G	Y
0	0	1
1	0	Y*
0	1	0
1	1	0

Stan jednoznaczny



## UKŁADY SEKWENCYJNE



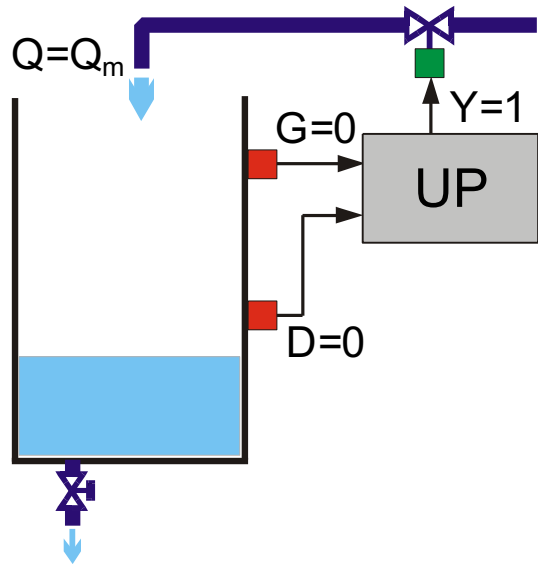
D	G	Y
0	0	1
1	0	1
1	1	0
1	0	0

D	G	Y
0	0	1
1	0	Y*
0	1	0
1	1	0

$$Y = \bar{D} \cdot \bar{G} \cdot 1 + D \cdot \bar{G} \cdot Y^* + \bar{D} \cdot G \cdot 0 + D \cdot G \cdot 0$$

$$Y = \bar{D} \cdot \bar{G} + D \cdot \bar{G} \cdot Y^*$$

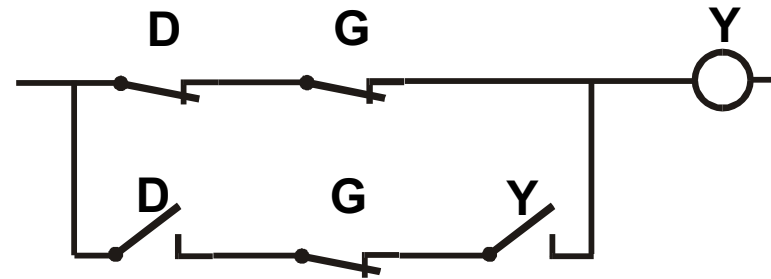
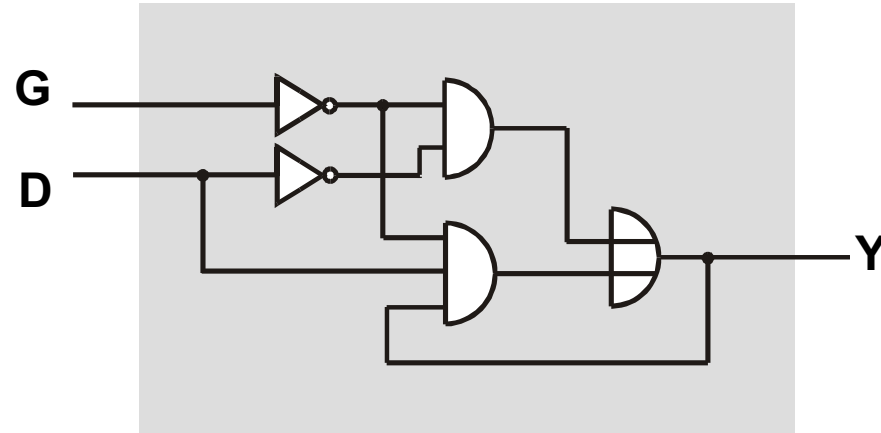
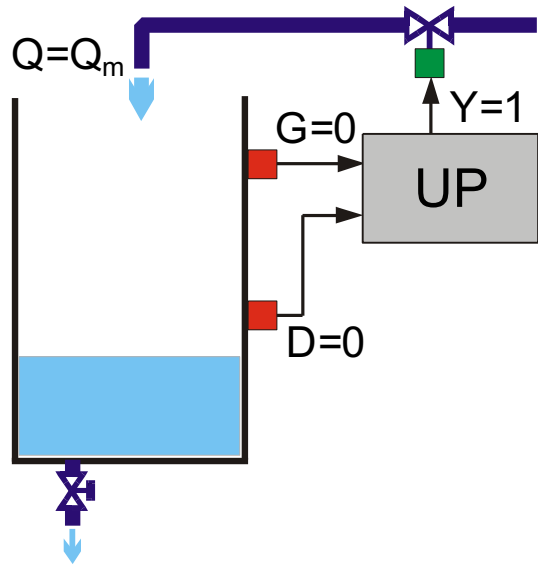
# UKŁADY SEKWENCYJNE



stan poprzedni (pamiętany) sygnału Y

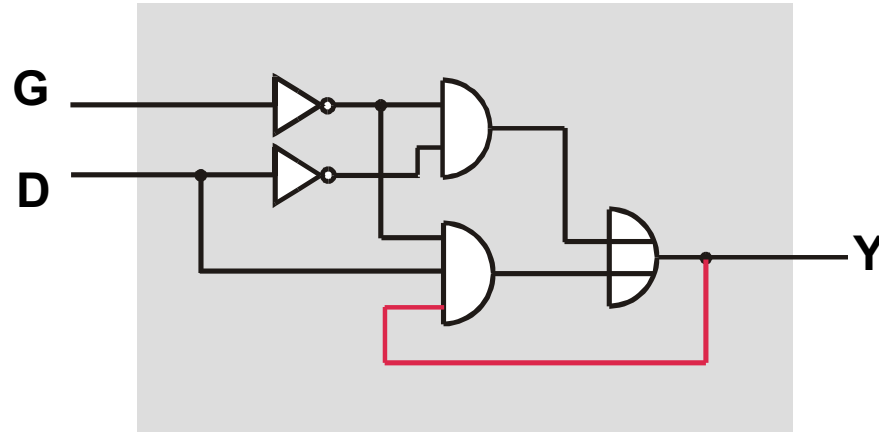
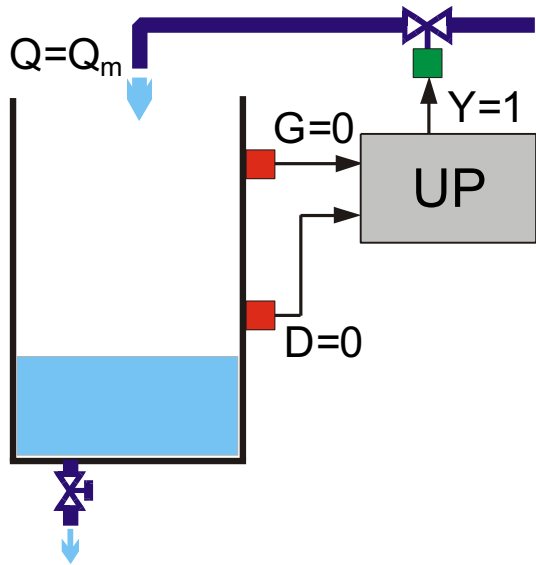
$$Y = \bar{D} \cdot \bar{G} + D \cdot \bar{G} \cdot Y^*$$

# UKŁADY SEKWENCYJNE

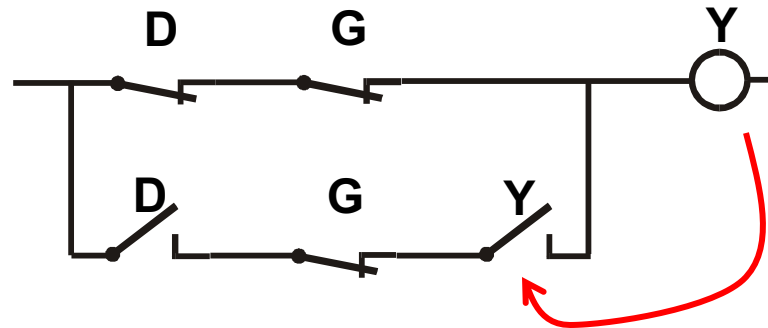


$$Y = \bar{D} \cdot \bar{G} + D \cdot \bar{G} \cdot Y^*$$

# UKŁADY SEKWENCYJNE



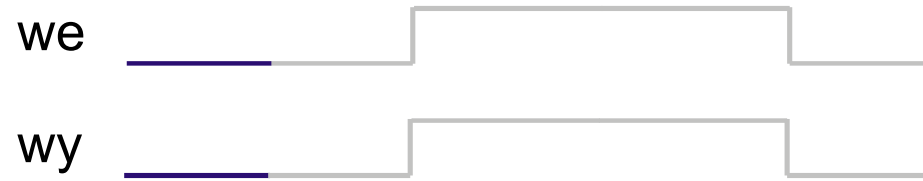
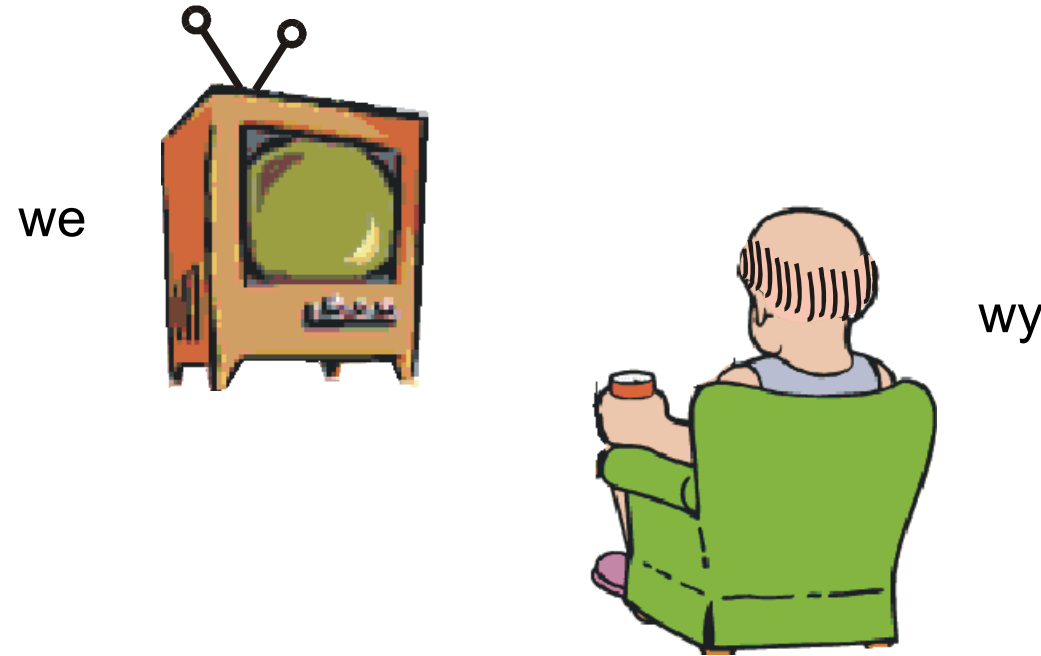
$$Y = \bar{D} \cdot \bar{G} + D \cdot \bar{G} \cdot Y^*$$



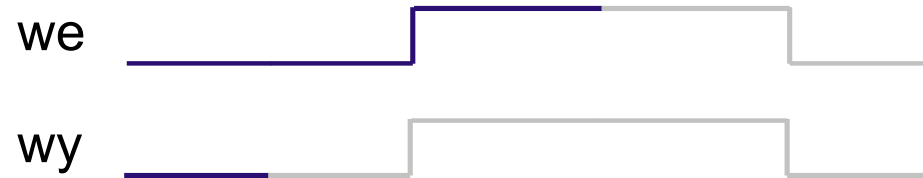
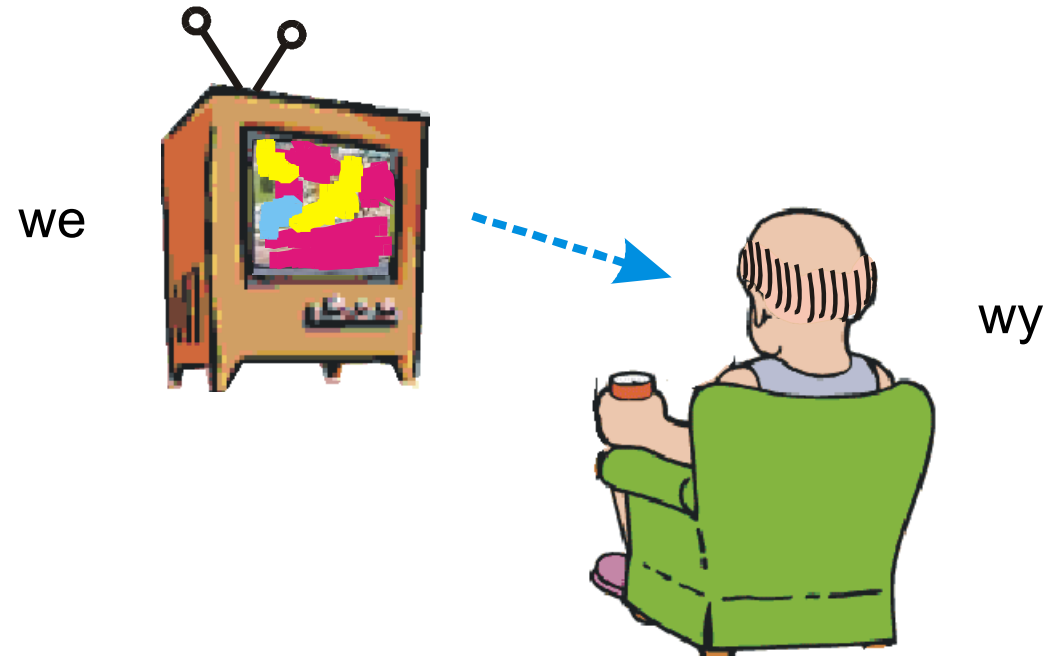
PAMIĘTANIE REALIZUJE SIĘ PRZEZ SPRZĘŻENIE ZWROTNE (PODTRZYMANIE)

UKŁADY  
SEKWENCYJNE

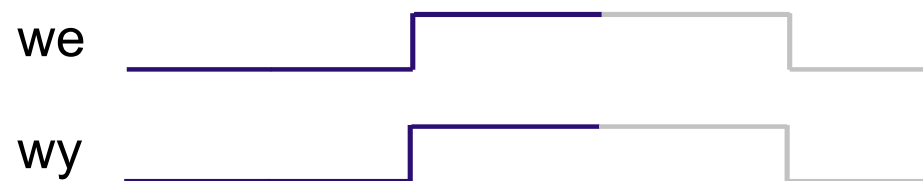
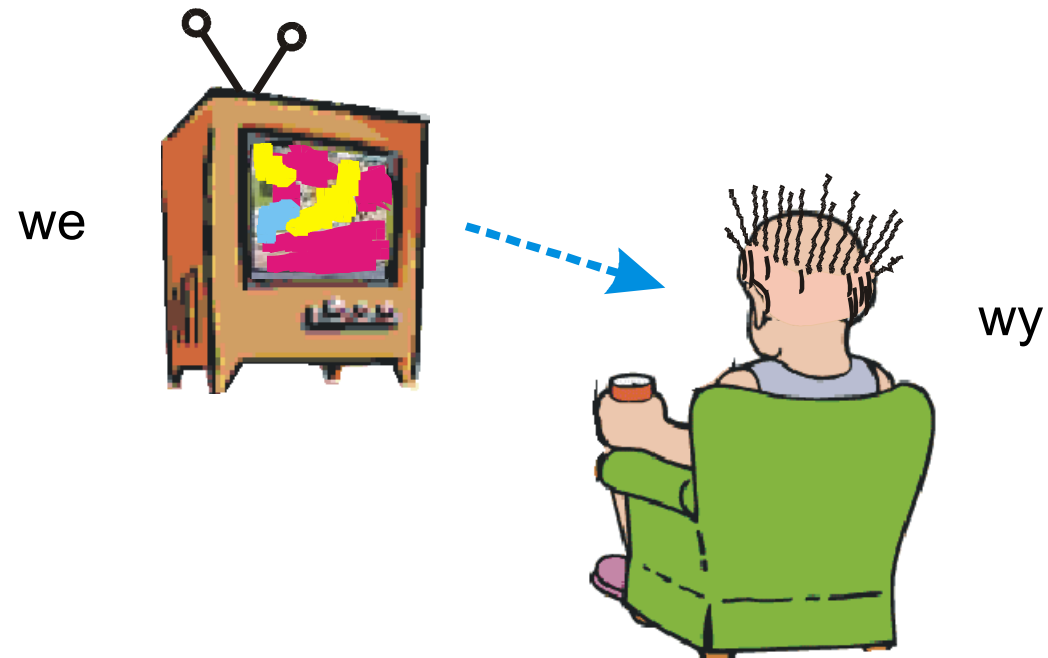
REALIZACJA PAMIĘCI



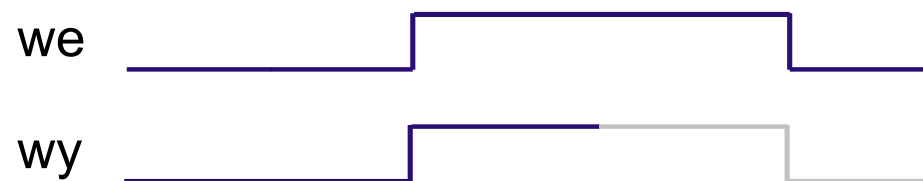
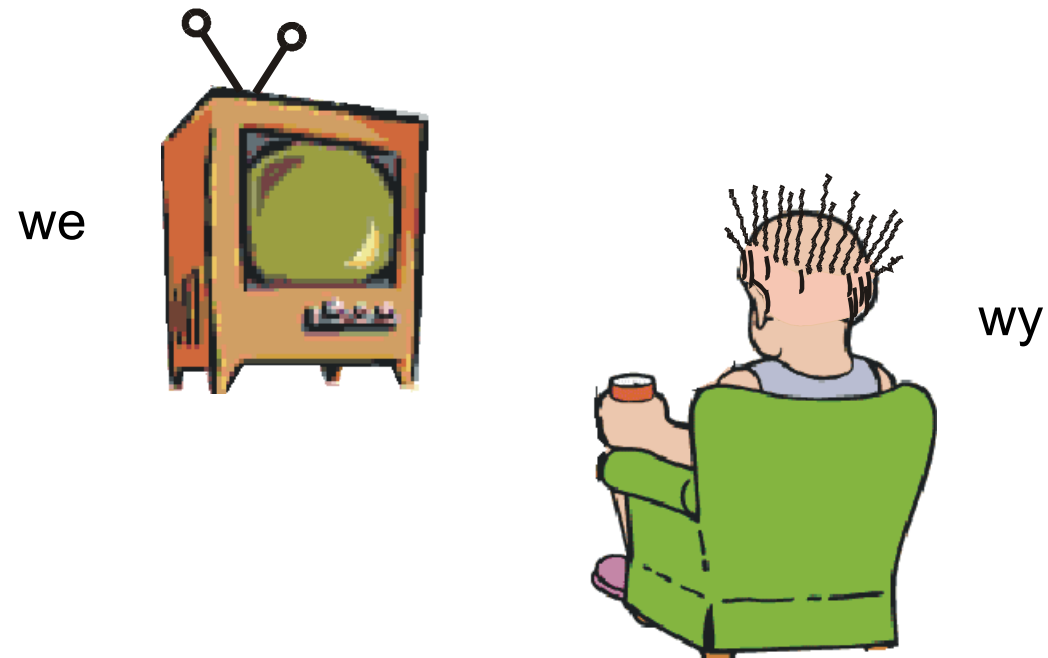
UKŁAD BEZ PAMIĘCI



**UKŁAD BEZ PAMIĘCI**

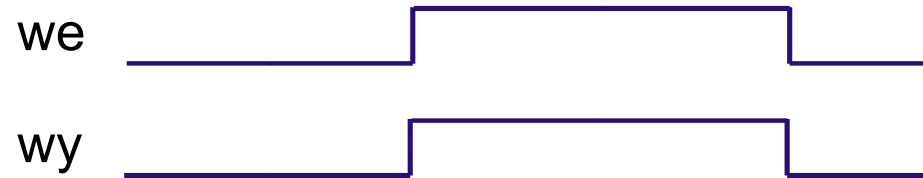
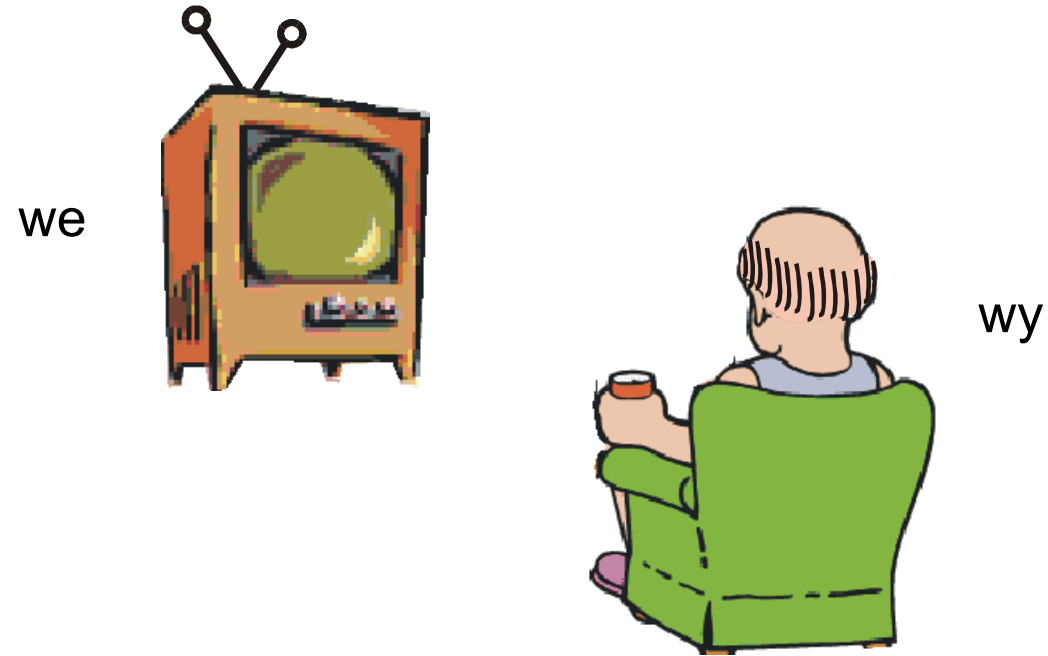


**UKŁAD BEZ PAMIĘCI**

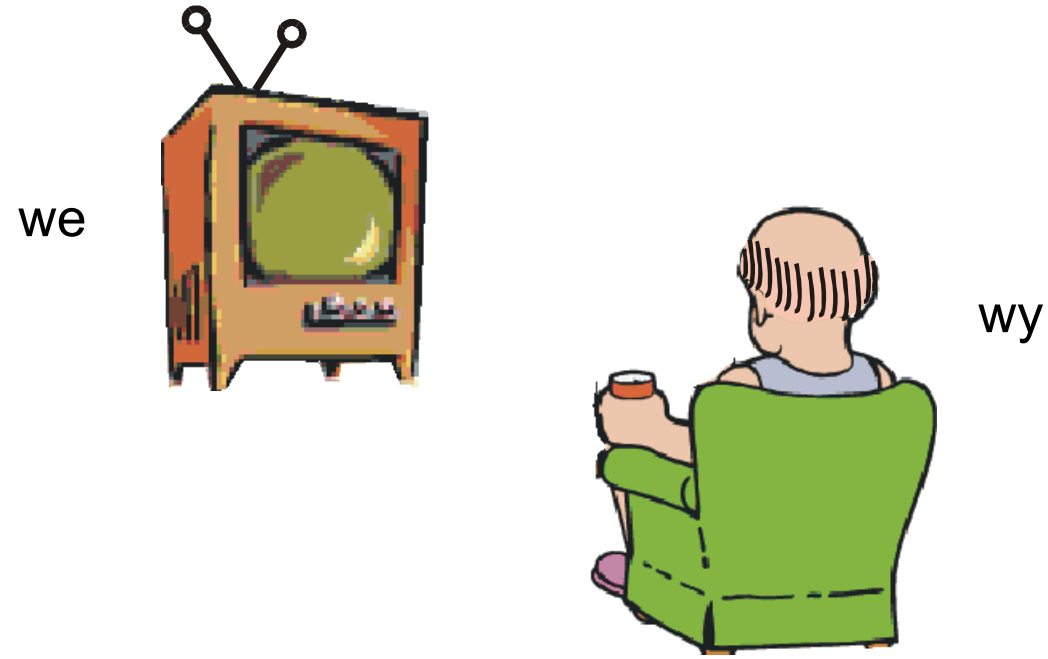


**UKŁAD BEZ PAMIĘCI**





**UKŁAD BEZ PAMIĘCI**

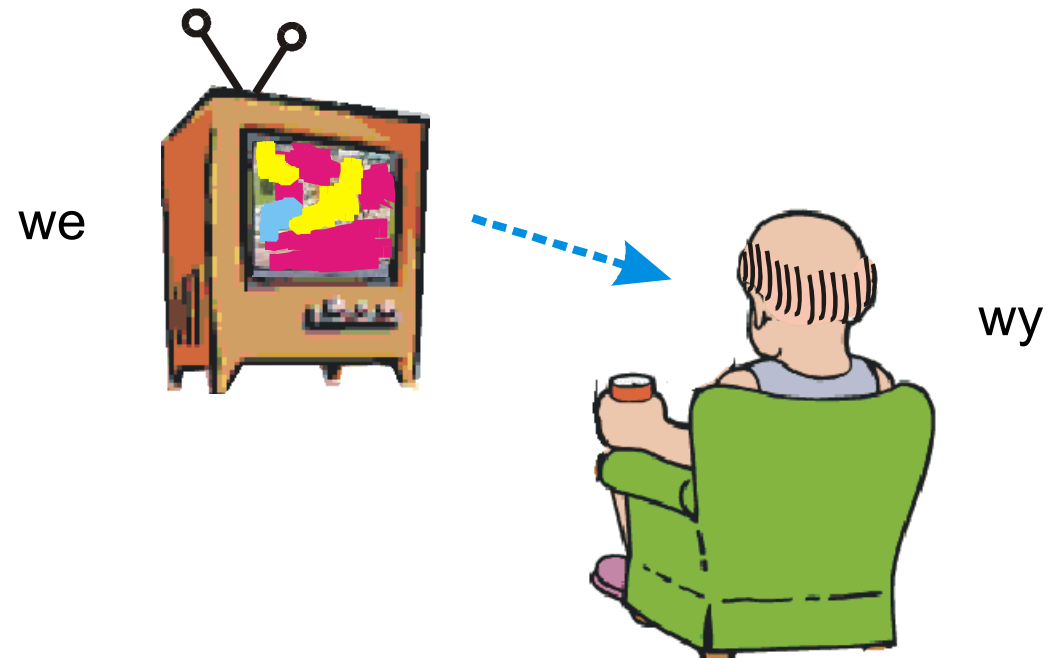


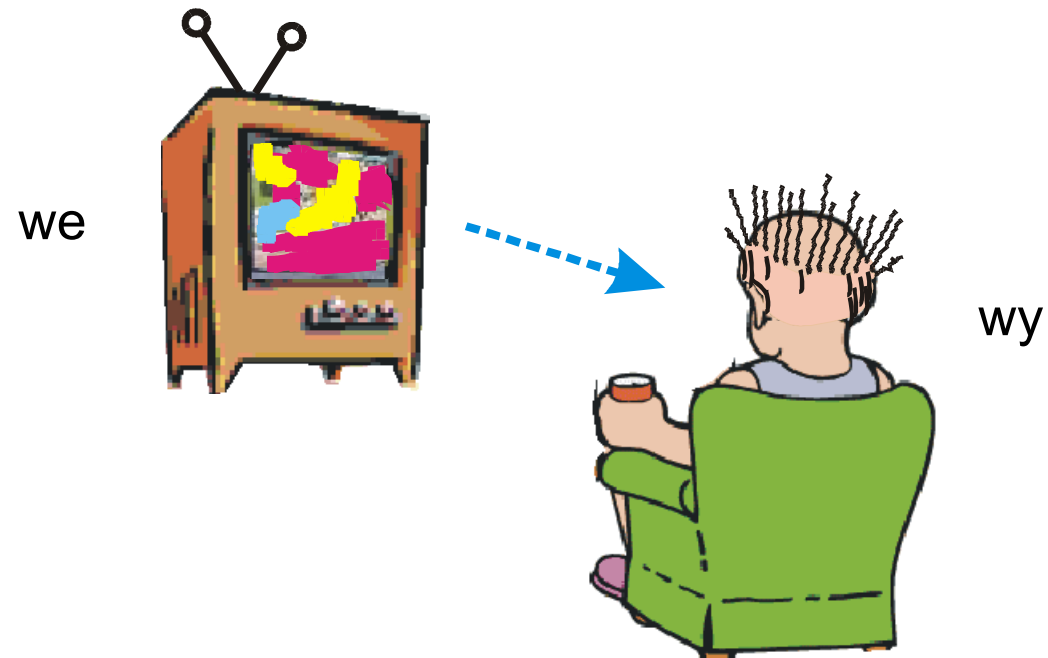
we

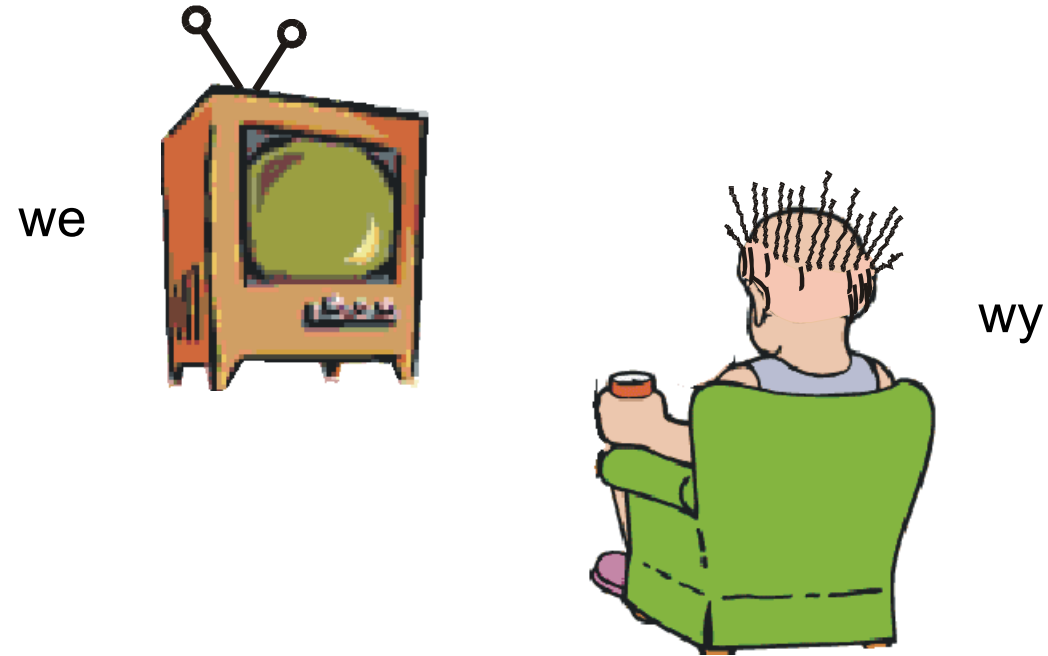
wy



**UKŁAD Z PAMIĘCIĄ**







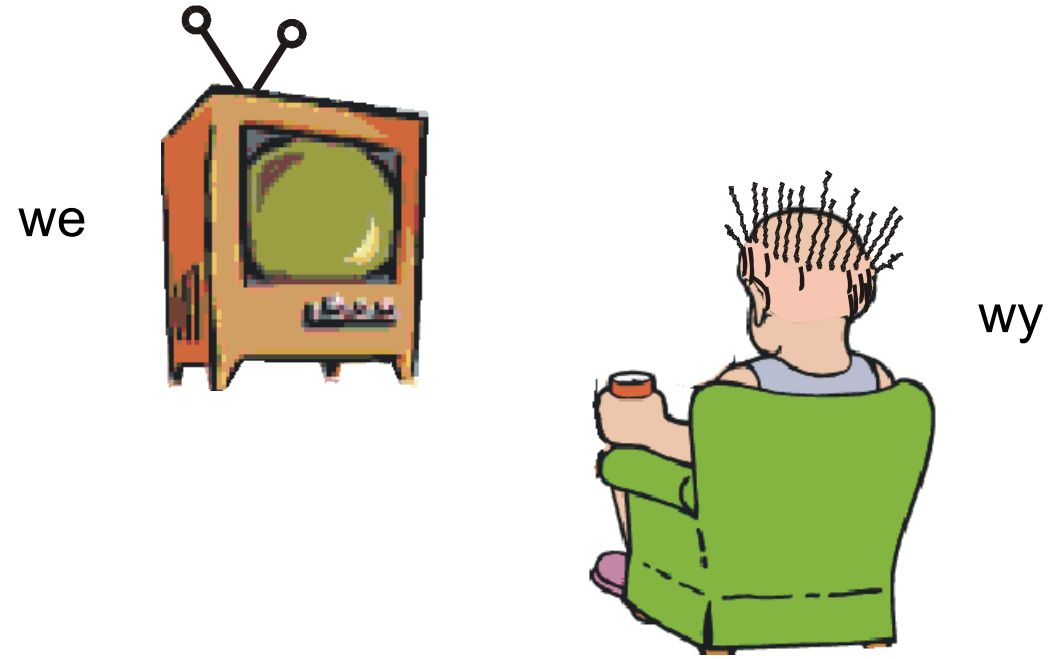
we



wy



**UKŁAD Z PAMIĘCIĄ**



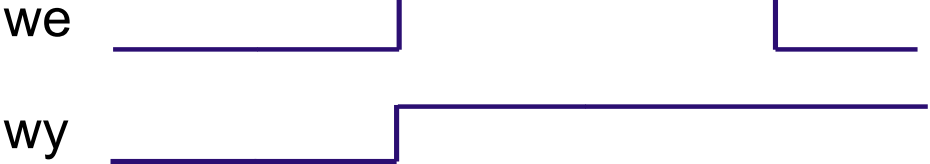
we

wy

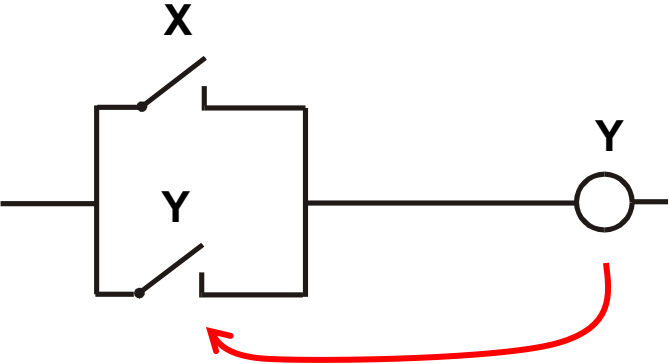


**UKŁAD Z PAMIĘCIĄ**

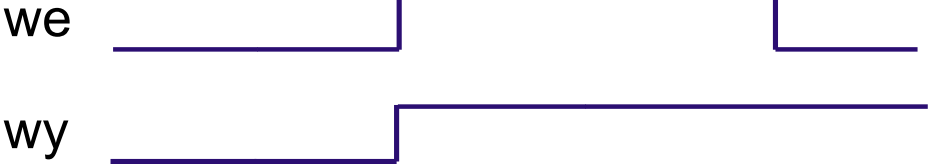
UKŁAD PAMIĘCI



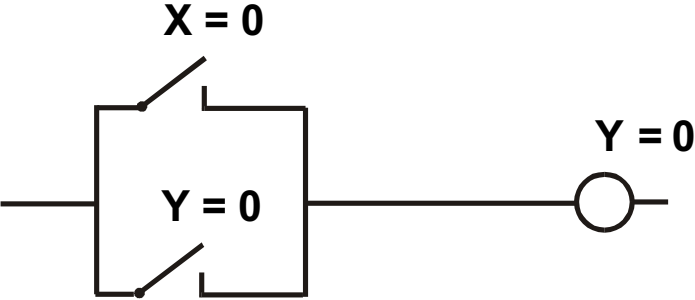
UKŁAD Z PAMIĘCIĄ



UKŁAD PAMIĘCI

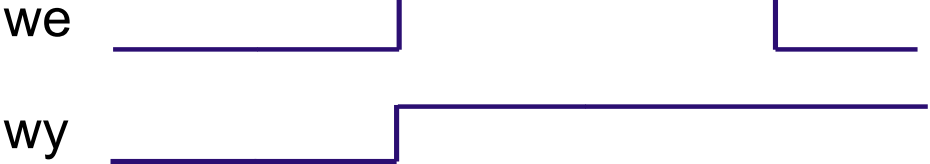


UKŁAD Z PAMIĘCIĄ

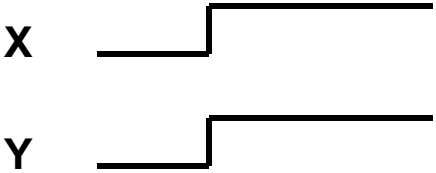
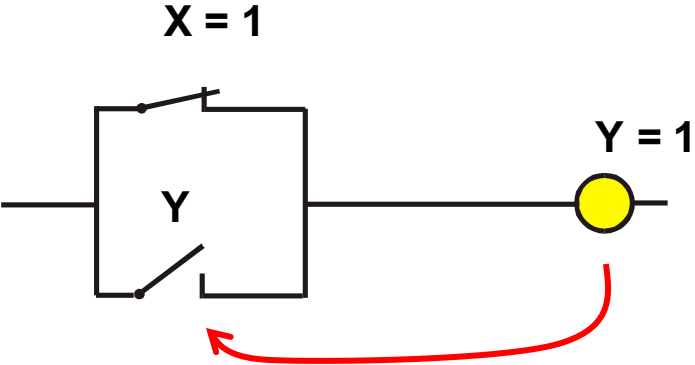




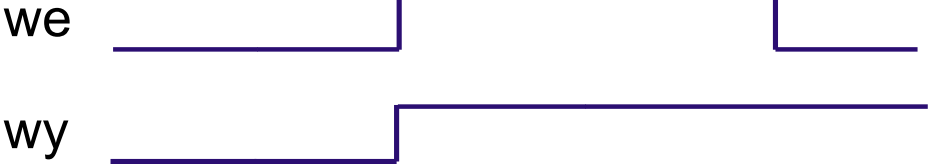
UKŁAD PAMIĘCI



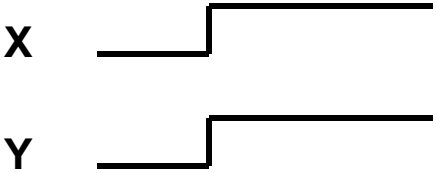
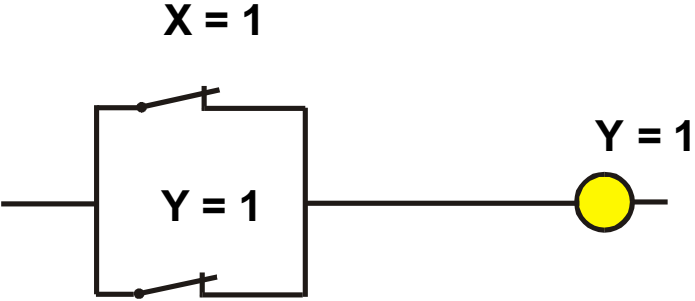
UKŁAD Z PAMIĘCIĄ



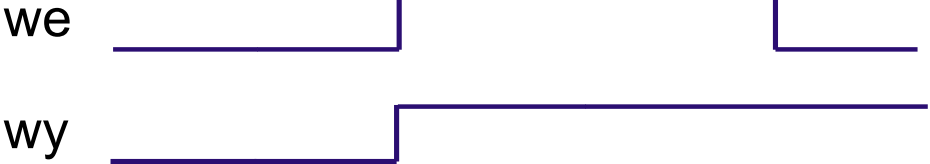
UKŁAD PAMIĘCI



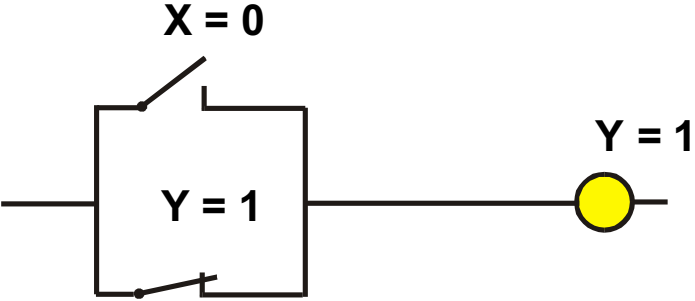
UKŁAD Z PAMIĘCIĄ



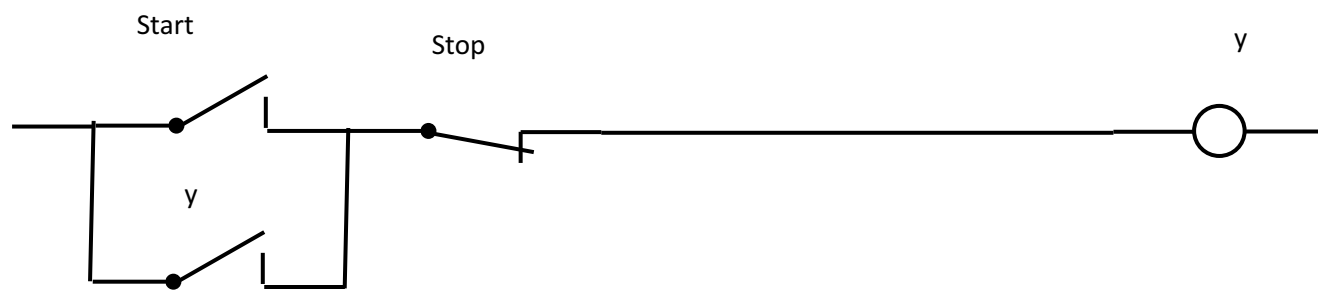
UKŁAD PAMIĘCI



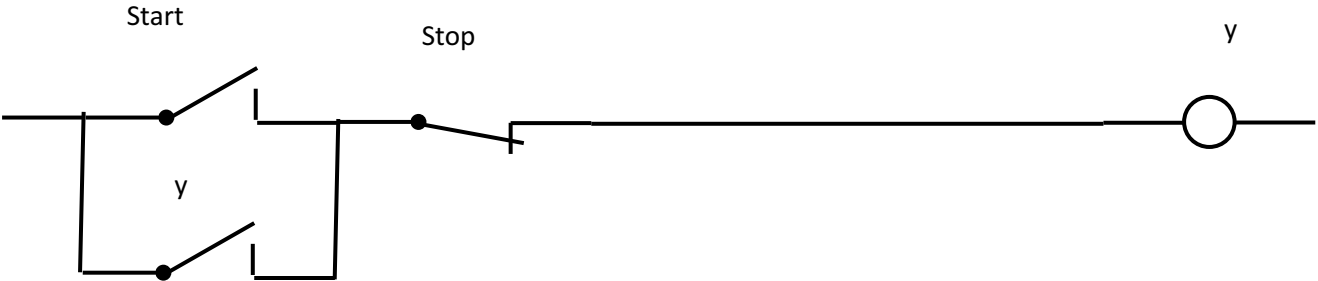
UKŁAD Z PAMIĘCIĄ



# UKŁAD PAMIĘCI



# UKŁAD PAMIĘCI

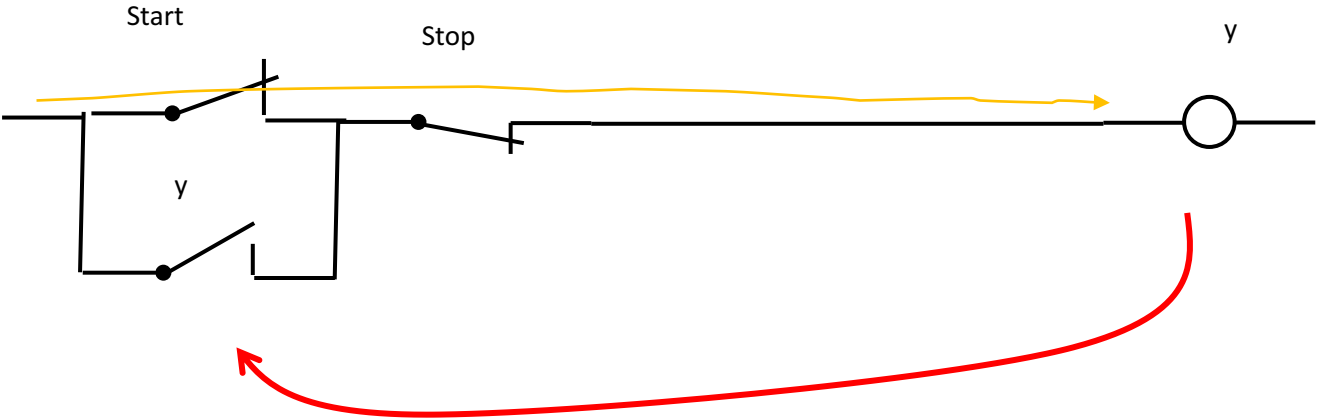


**Start** ———

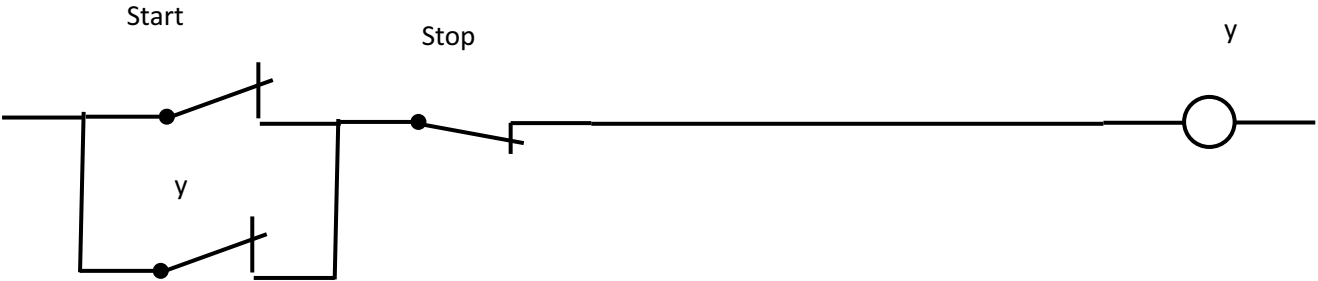
**Stop** ———

**Y** ———

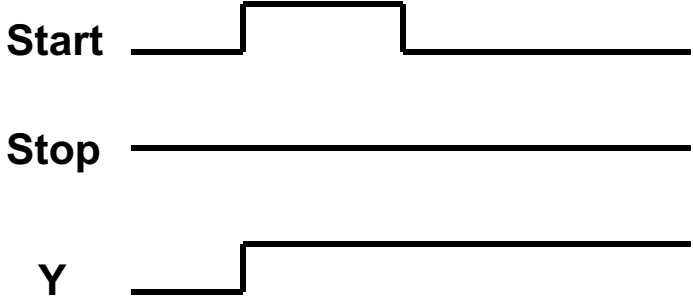
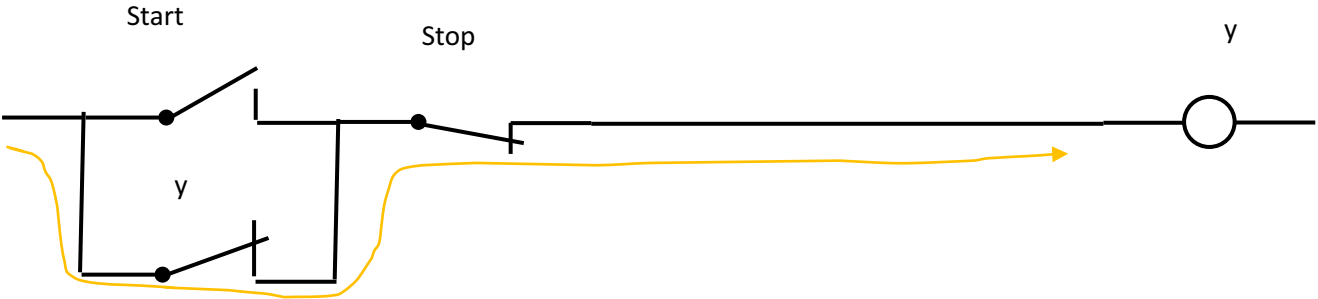
# UKŁAD PAMIĘCI



# UKŁAD PAMIĘCI

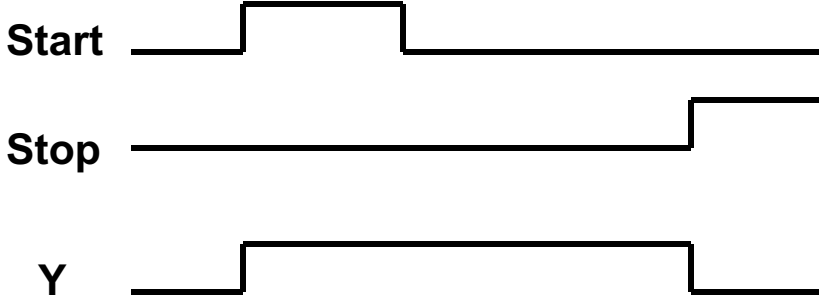
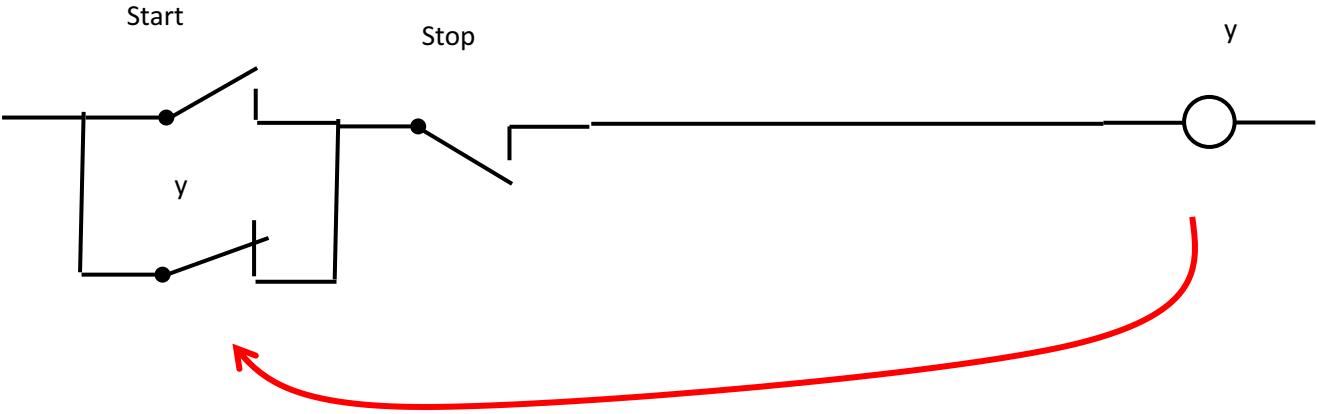


# UKŁAD PAMIĘCI

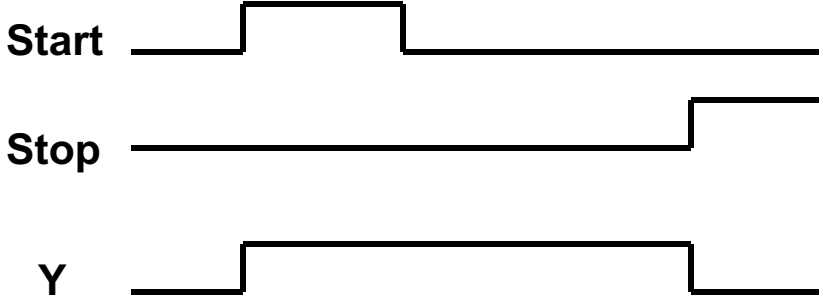
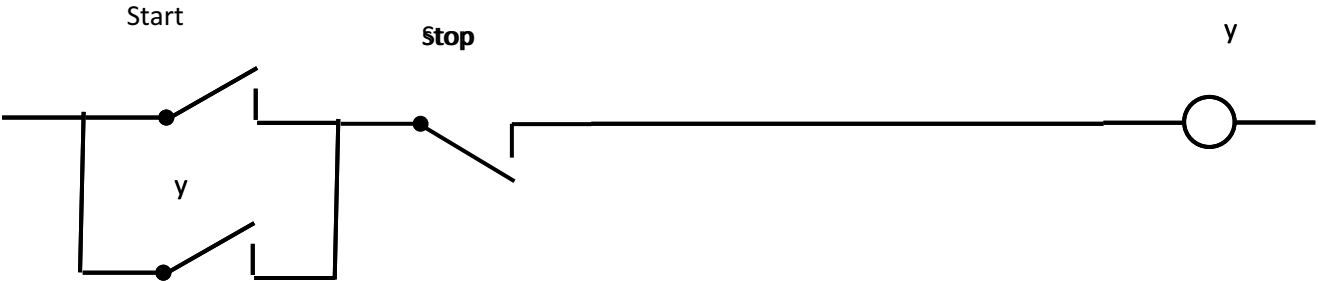




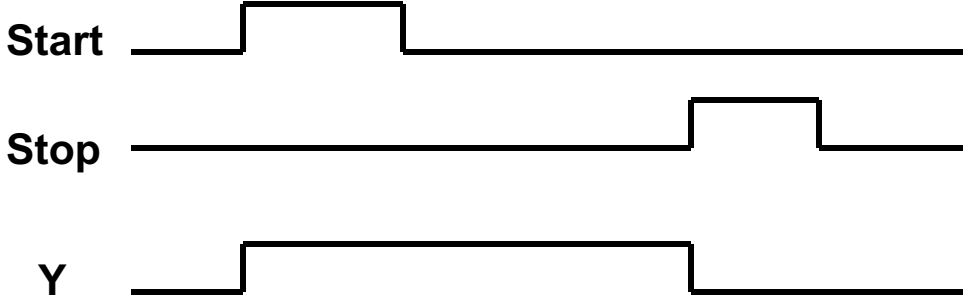
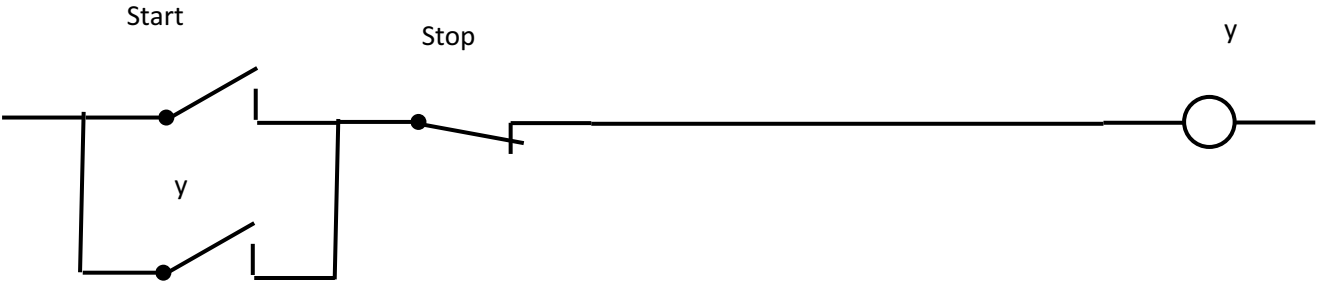
# UKŁAD PAMIĘCI



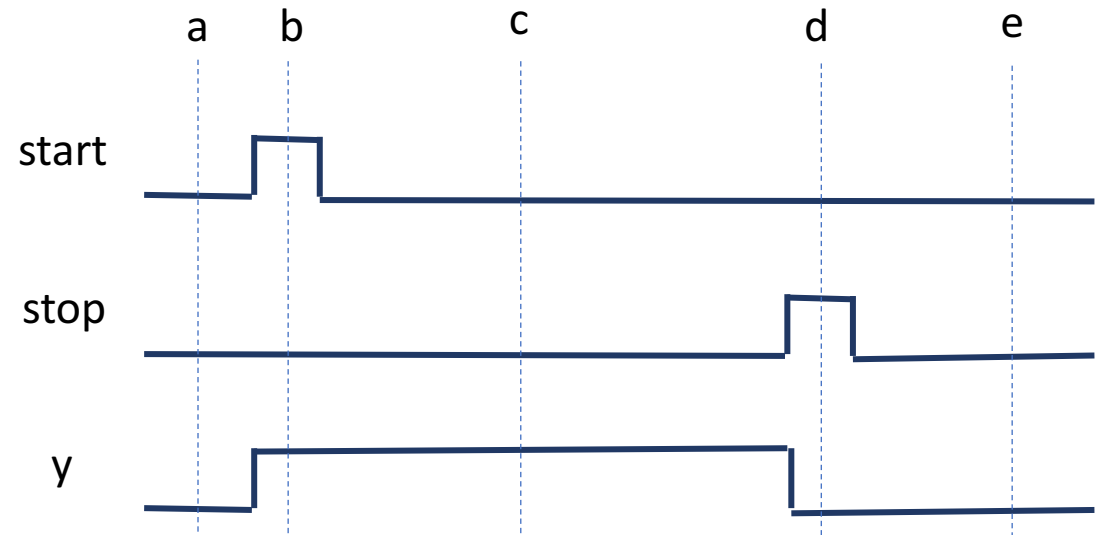
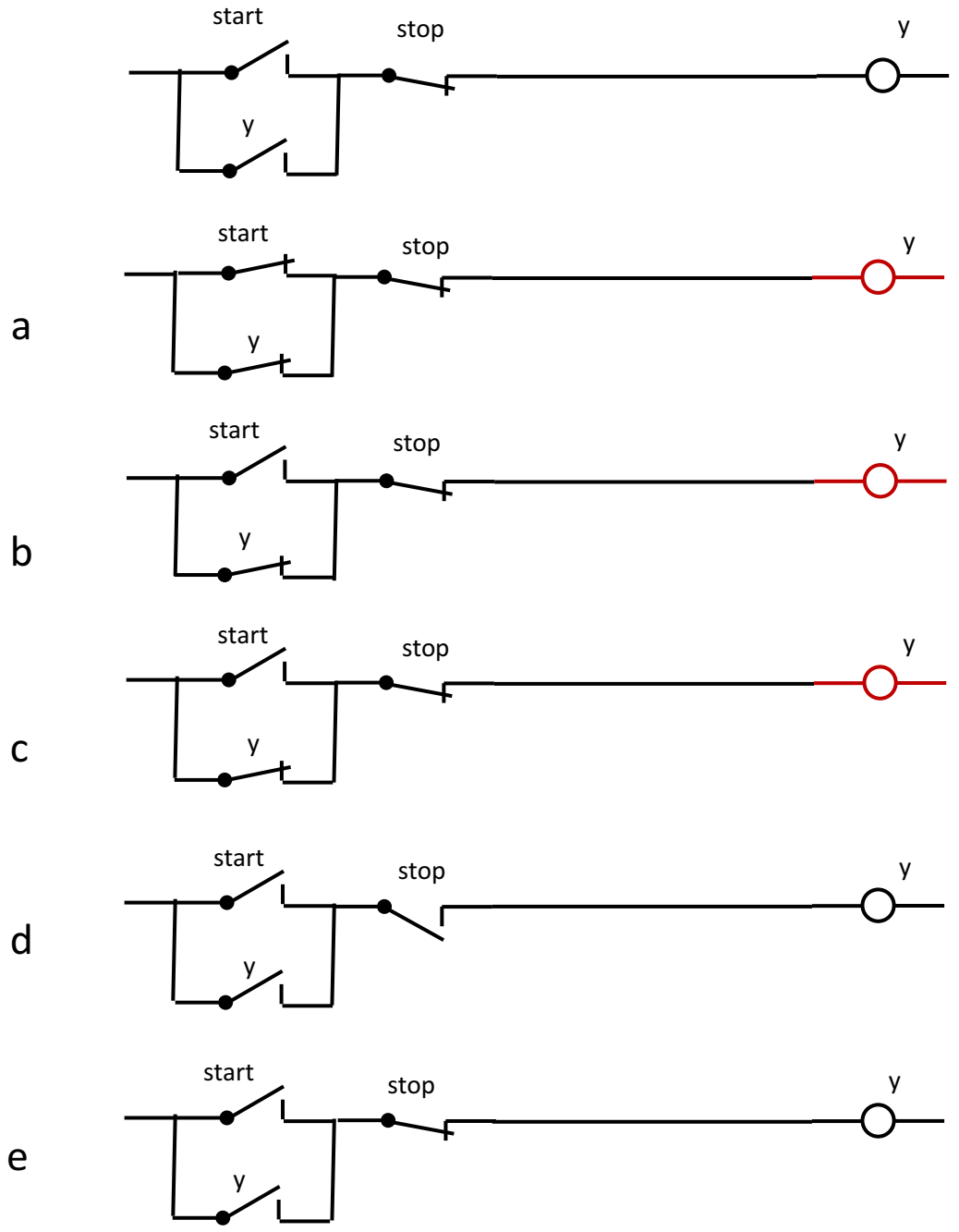
# UKŁAD PAMIĘCI



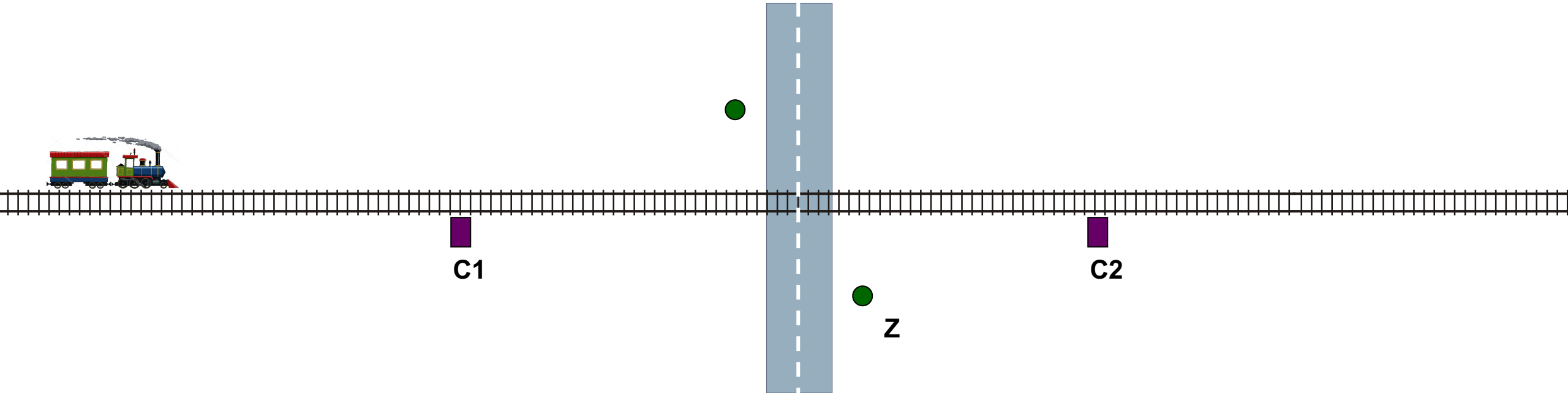
# UKŁAD PAMIĘCI



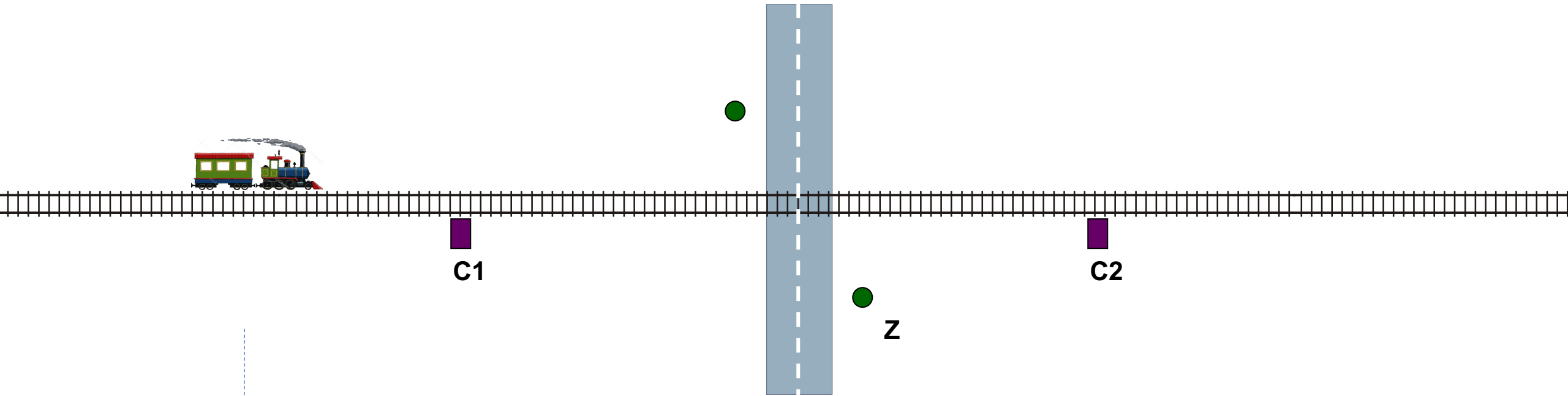
UKŁAD START-STOP z podtrzymaniem



# PRZYKŁAD 11



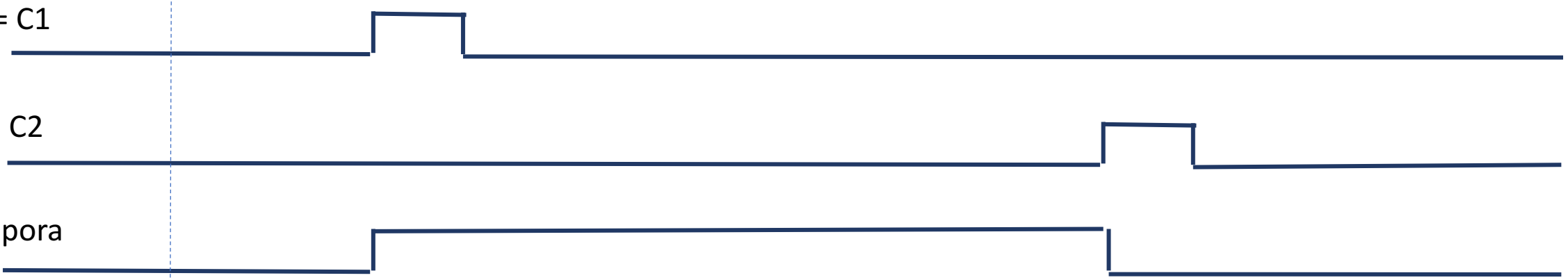
# PRZYKŁAD 11



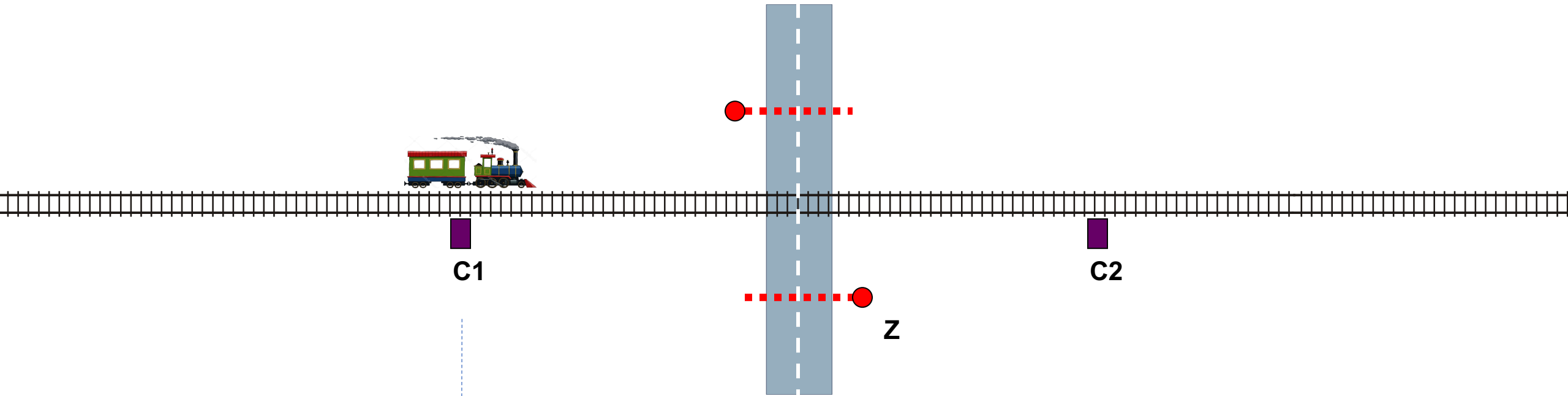
Start = C1

Stop = C2

Y = Zapora



# PRZYKŁAD 11



Start = C1



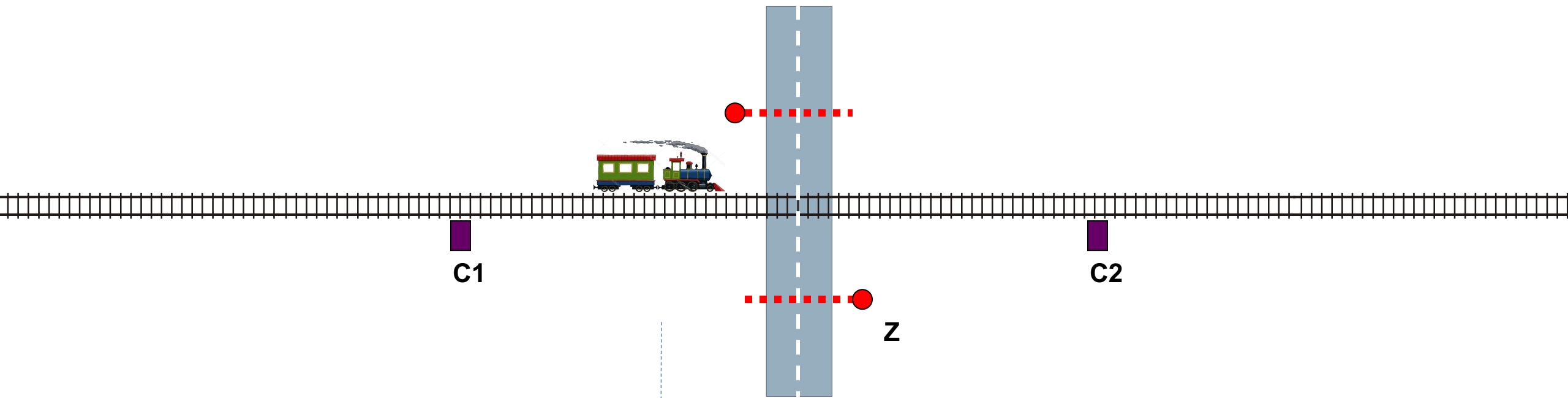
Stop = C2



Y = Zapora



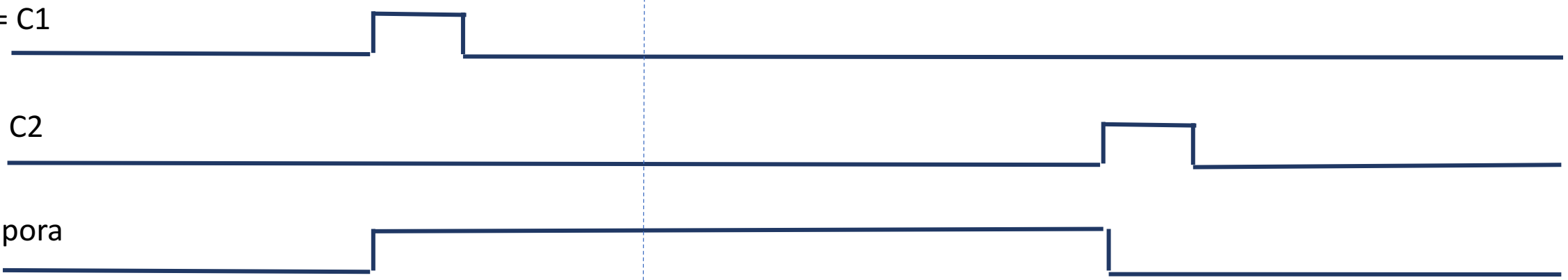
# PRZYKŁAD 11



Start = C1

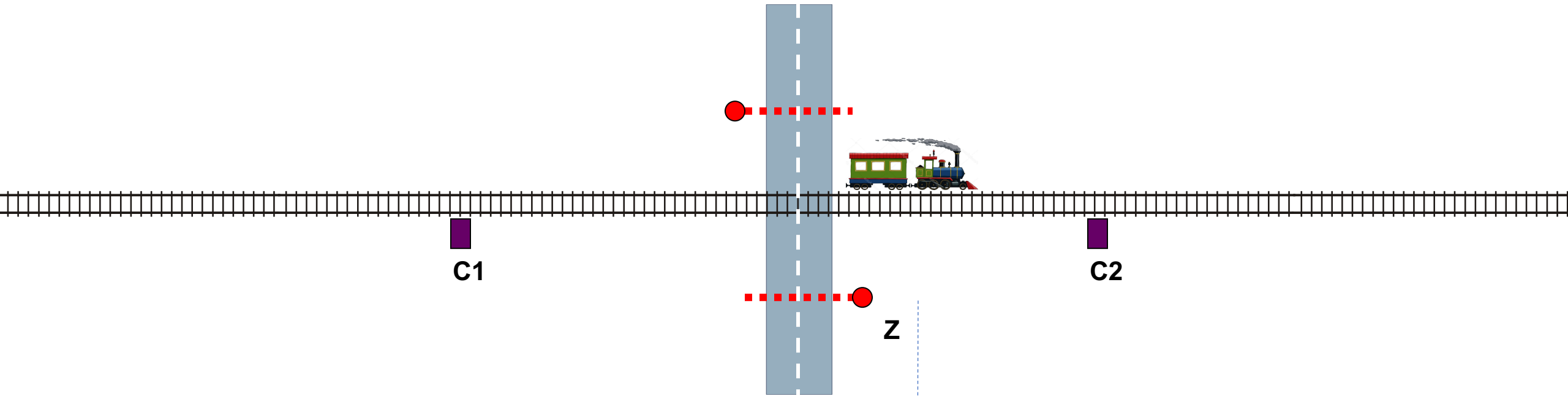
Stop = C2

Y = Zapora





# PRZYKŁAD 11



Start = C1



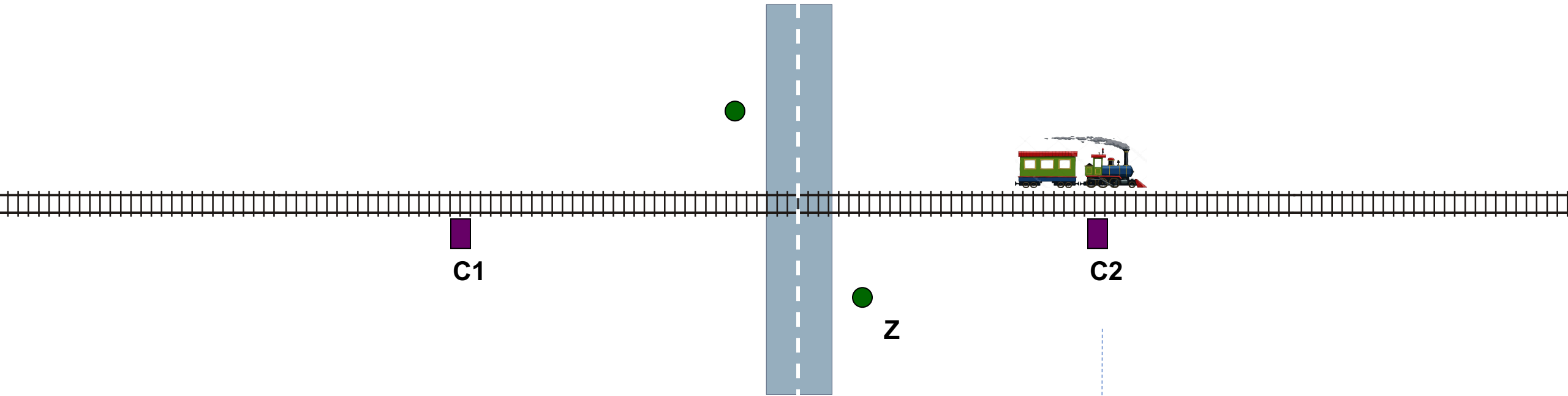
Stop = C2



Y = Zapora



# PRZYKŁAD 11



Start = C1



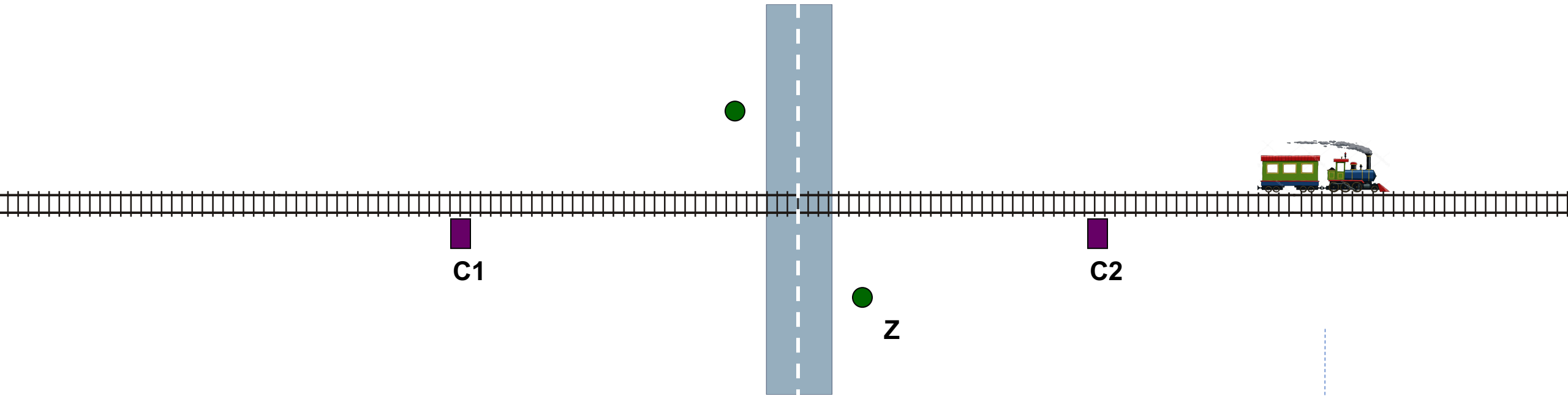
Stop = C2



Y = Zapora



# PRZYKŁAD 11



Start = C1



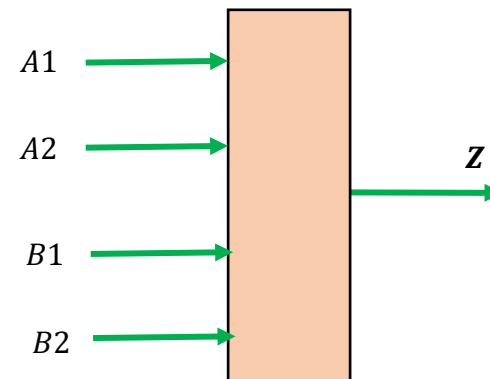
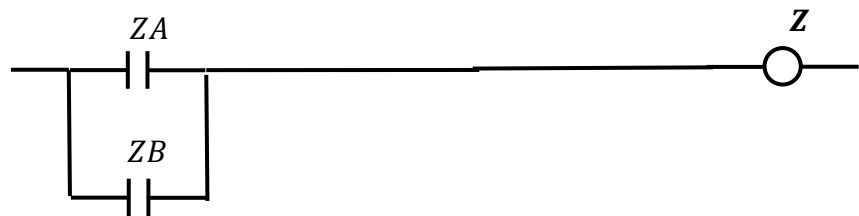
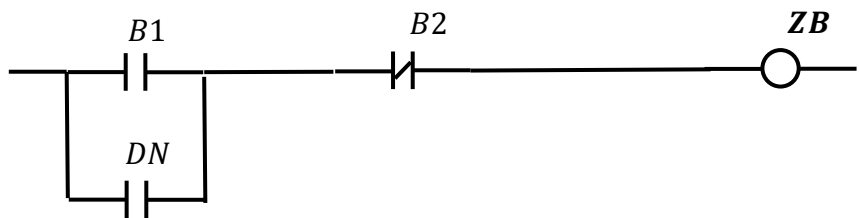
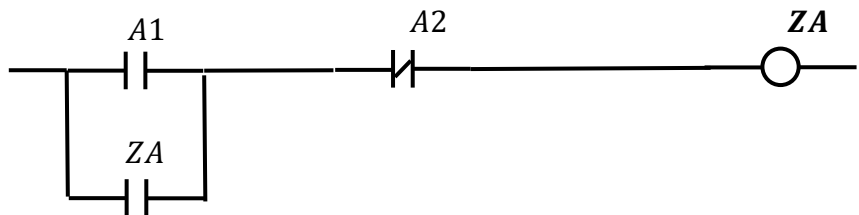
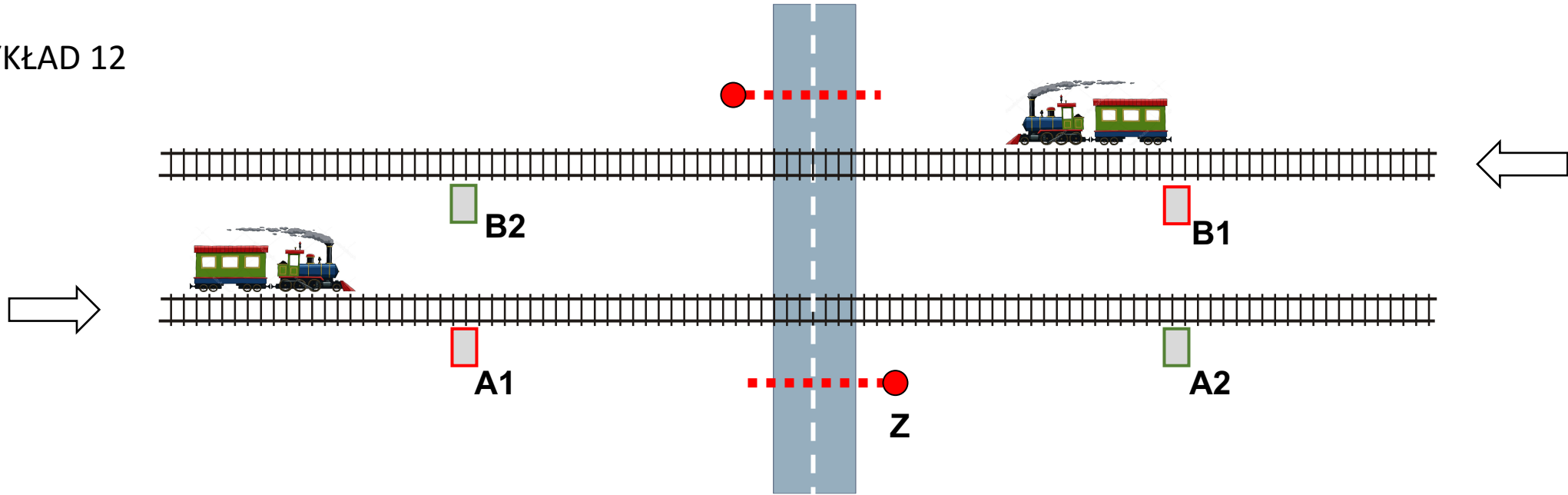
Stop = C2

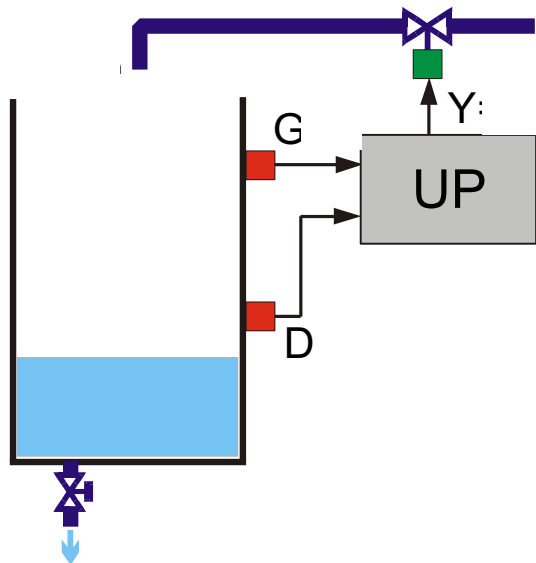


Y = Zapora



# PRZYKŁAD 12





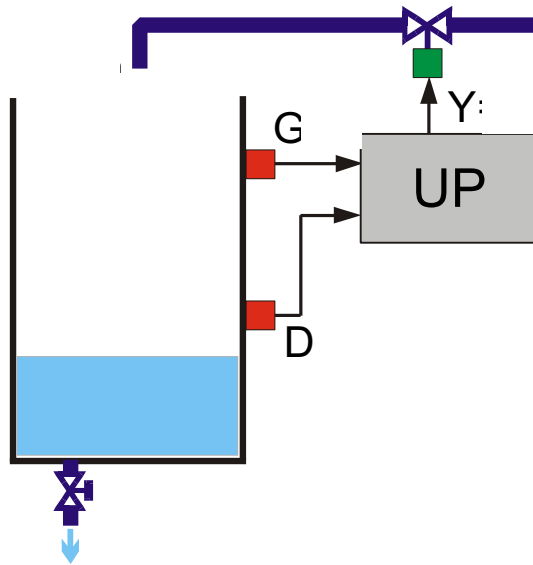
## REGULACJA 2. POŁOŻENIOWA

TEMPERATURY

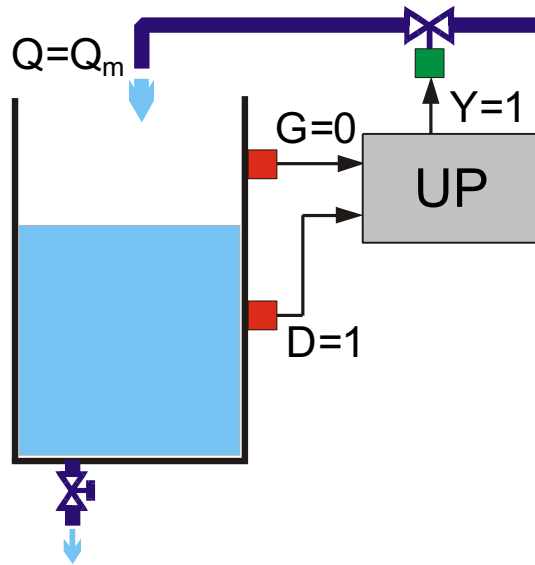
POZIOMU CIECZY

CIŚNIENIA

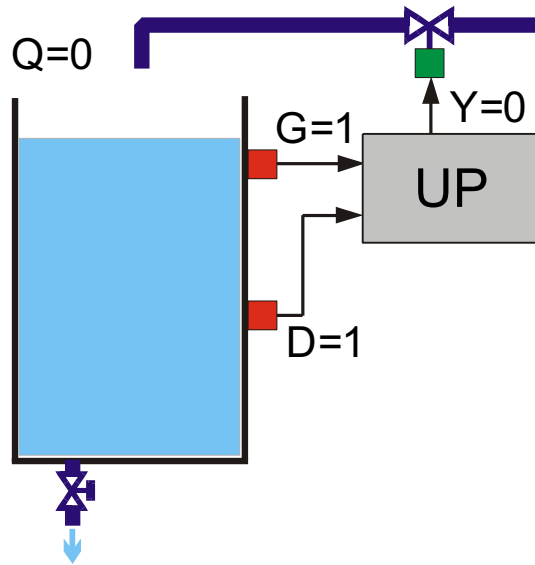
## REGULACJA 2. POŁOŻENIOWA



## REGULACJA 2. POŁOŻENIOWA

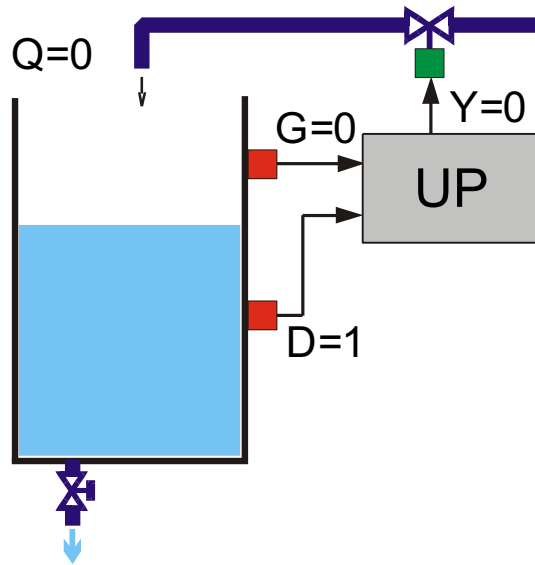


## REGULACJA 2. POŁOŻENIOWA

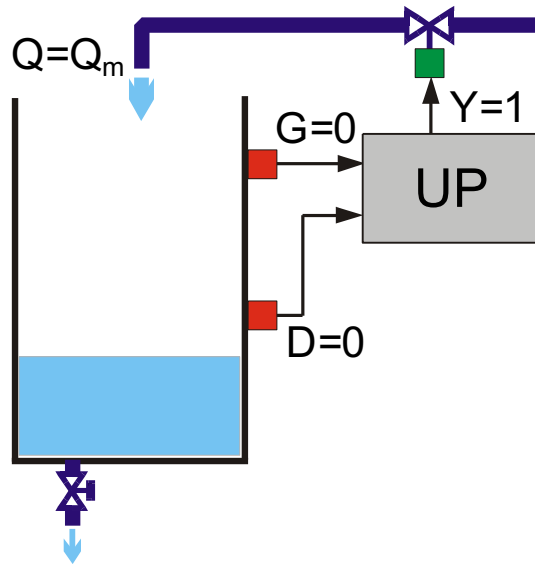




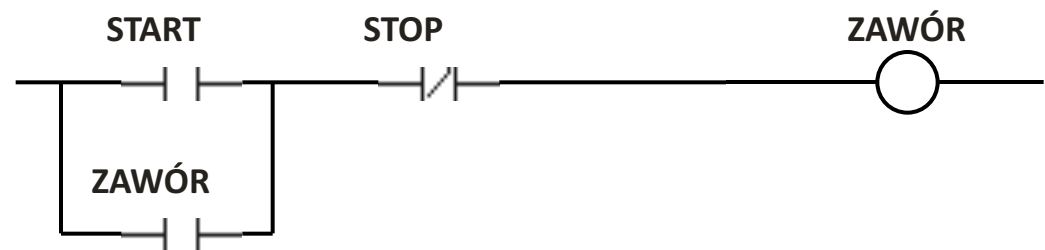
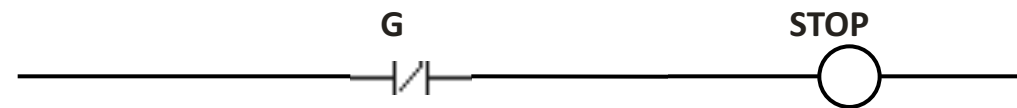
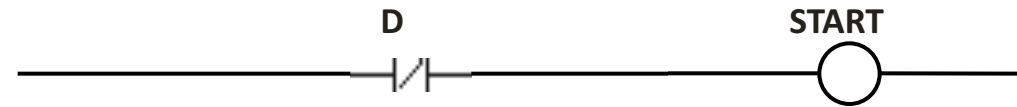
## REGULACJA 2. POŁOŻENIOWA



# PRZYKŁAD 13

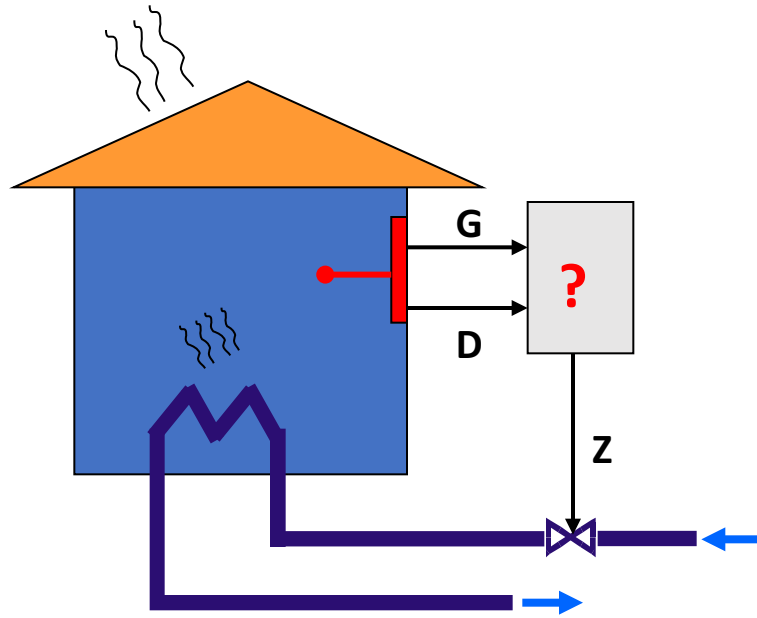


## REGULACJA 2. POŁOŻENIOWA

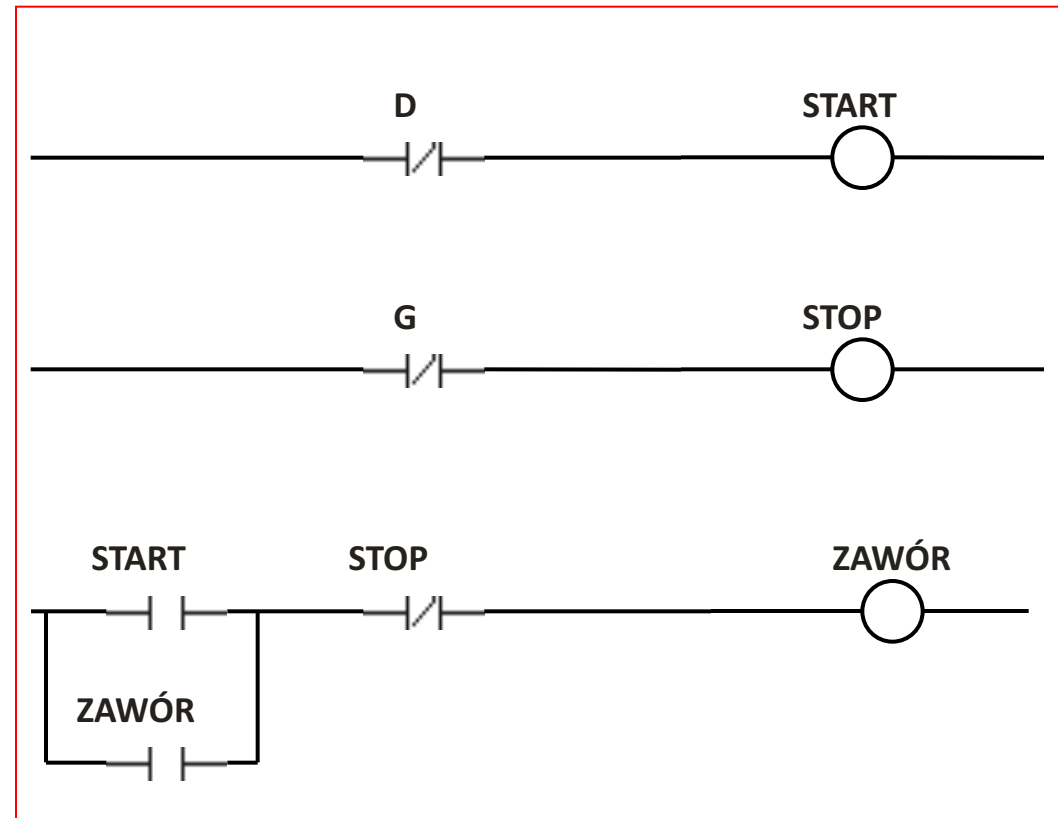


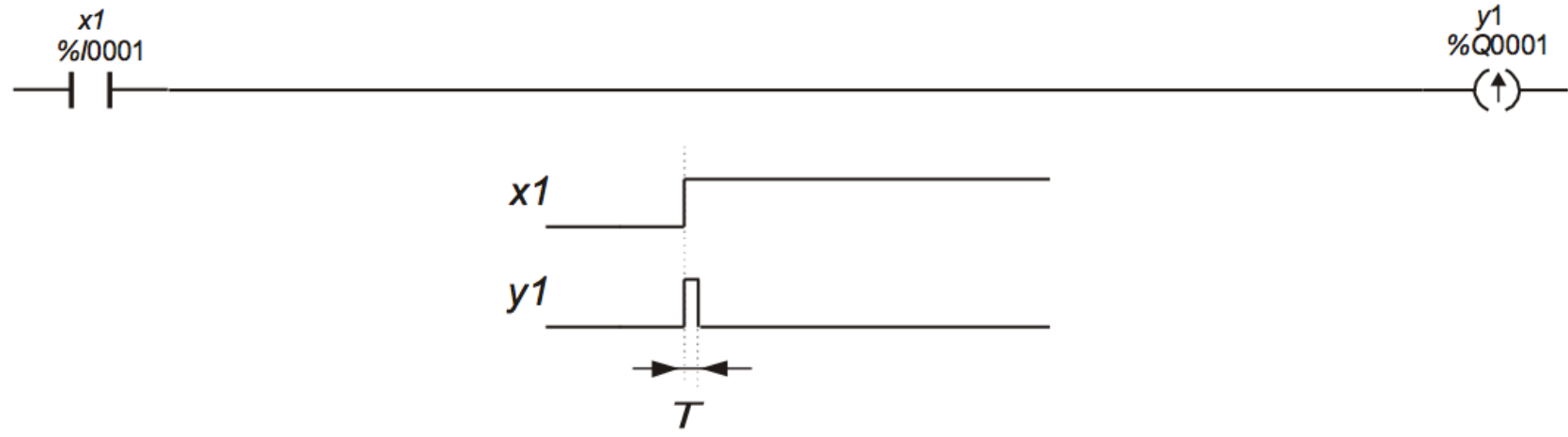
(ZAWÓR = Y)

# PRZYKŁAD 14

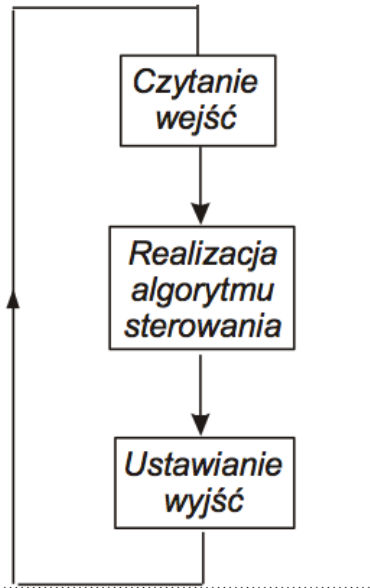


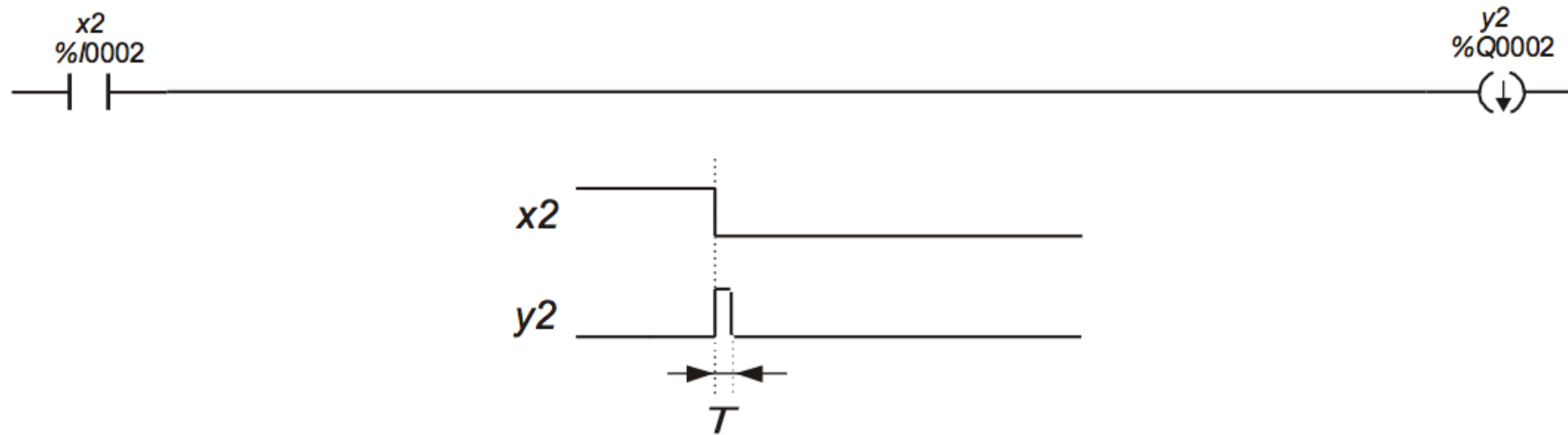
## REGULACJA 2. POŁOŻENIOWA TEMPERATURY





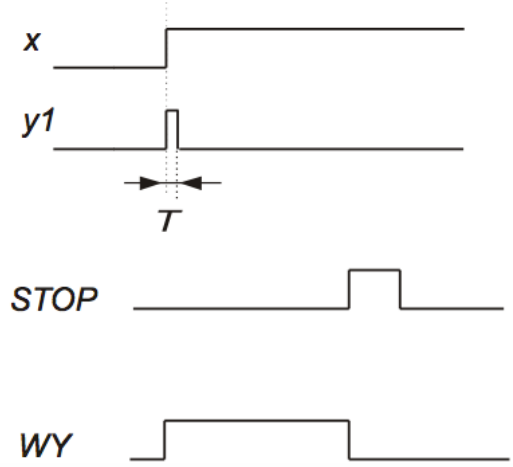
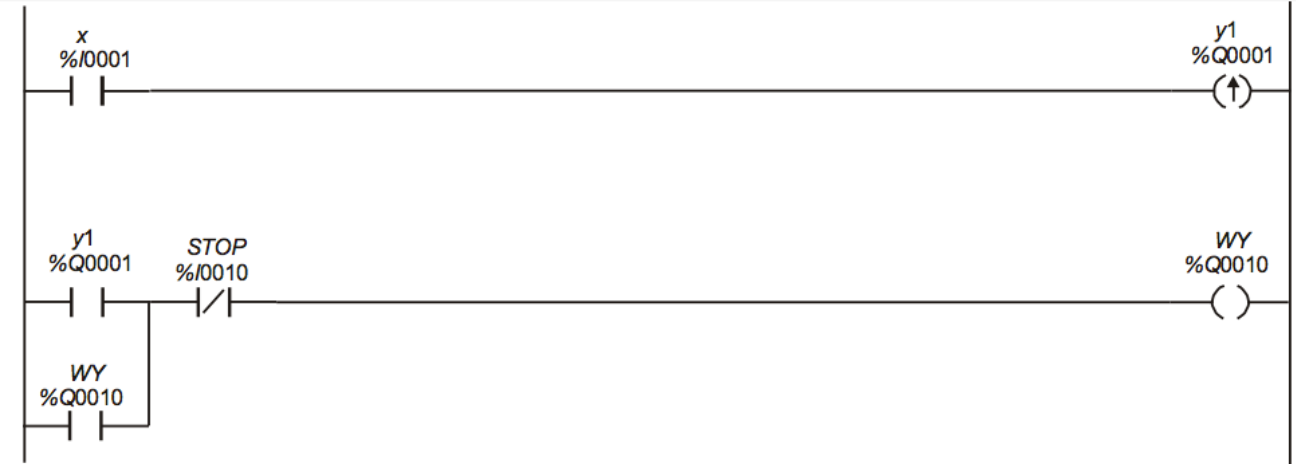
Działanie przekaźnika uruchamianego zboczem narastającym



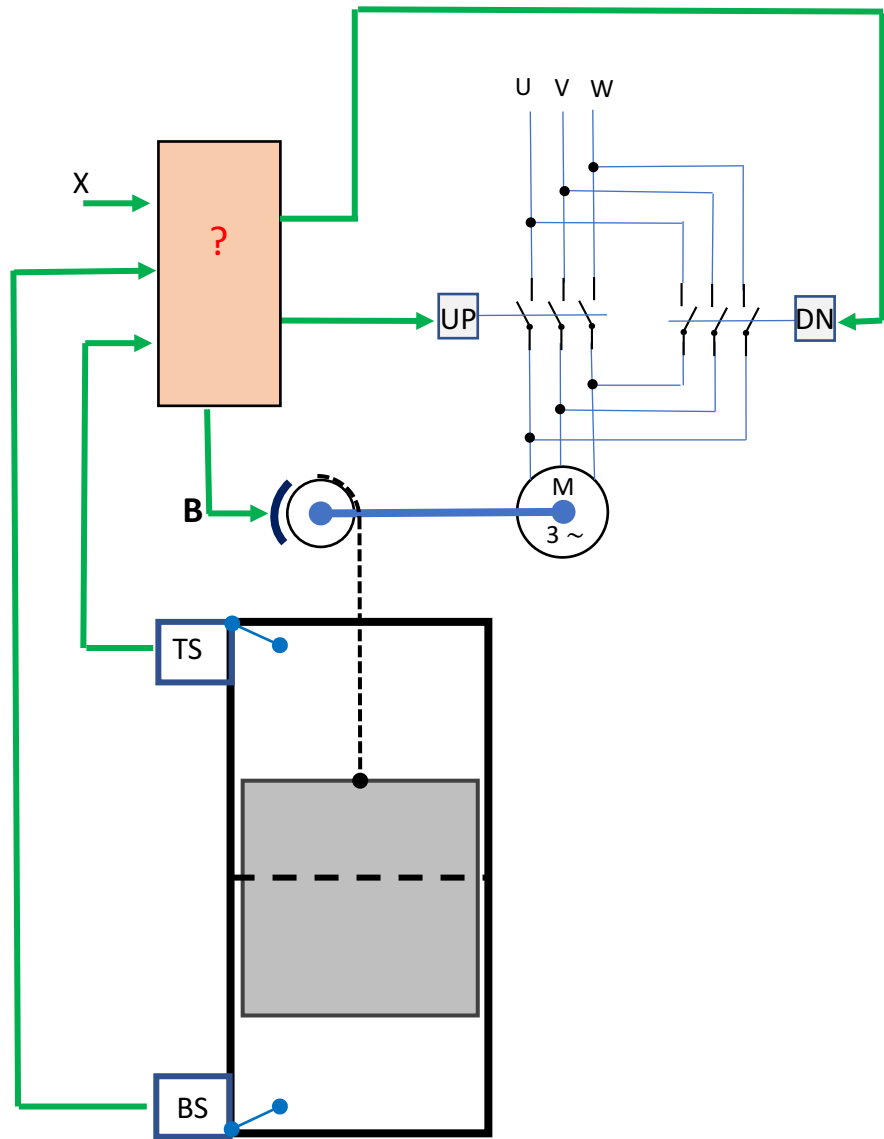
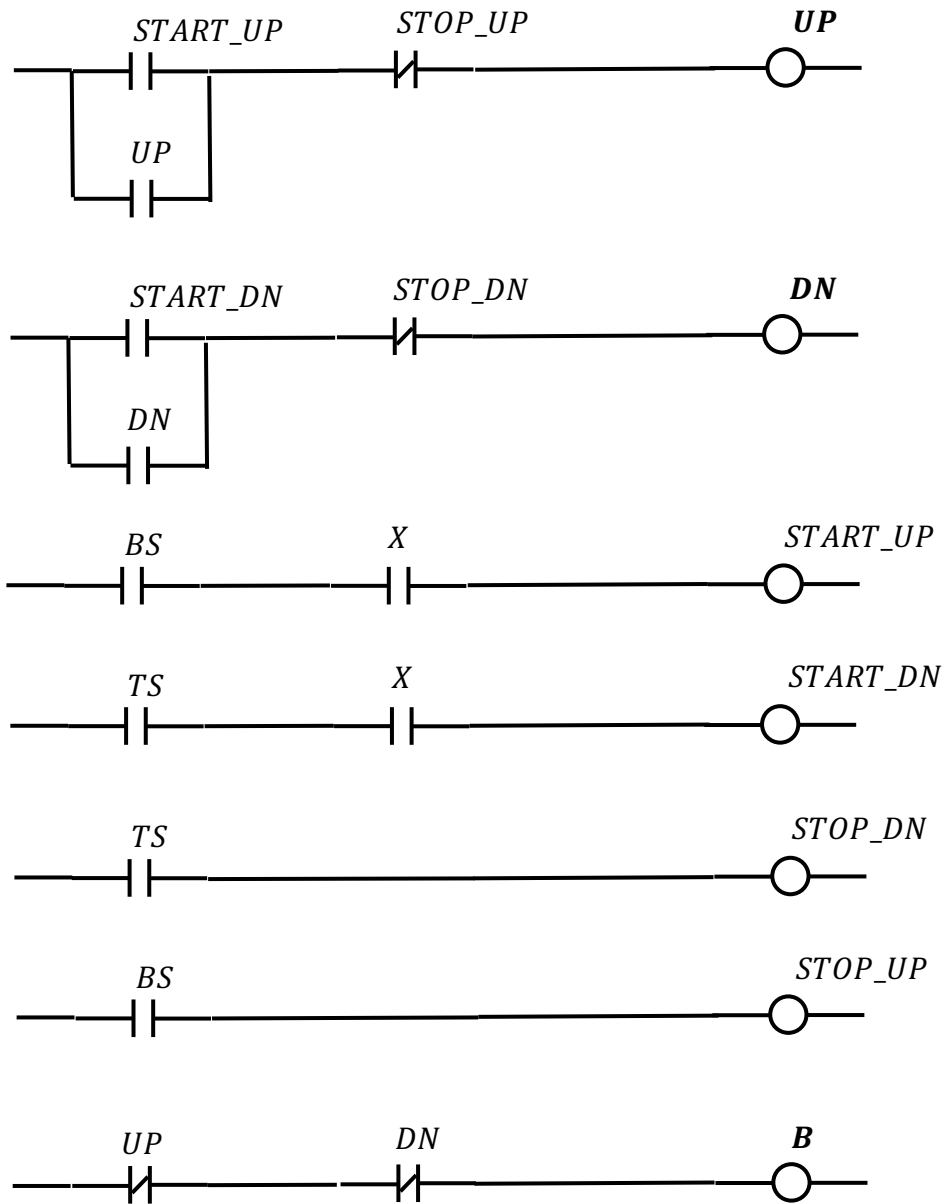


Działanie przekaźnika uruchamianego zboczem opadającym

PRZYKŁAD 15

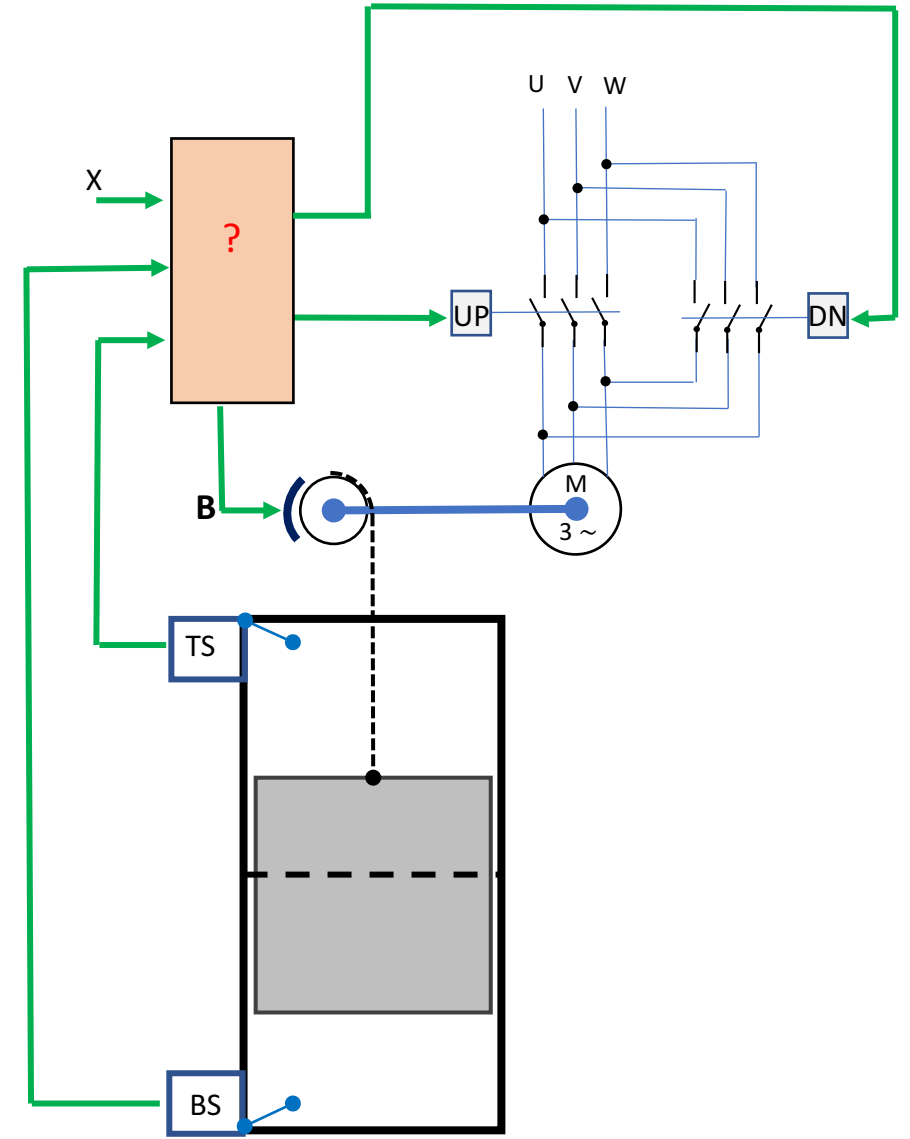
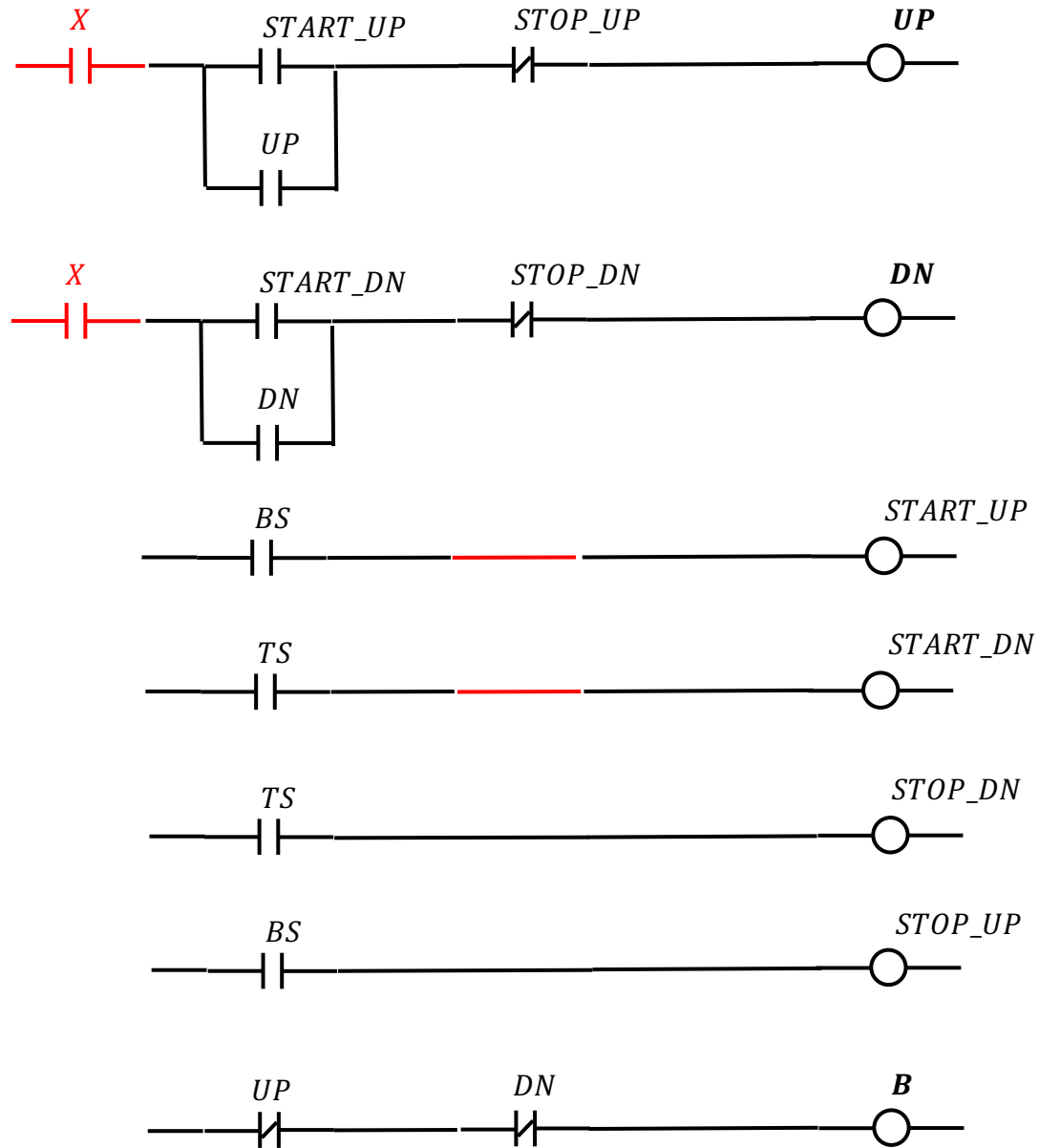


# UKŁADY SEKWENCYJNE PRZYKŁAD 16



**TWO-LEVEL LIFT**

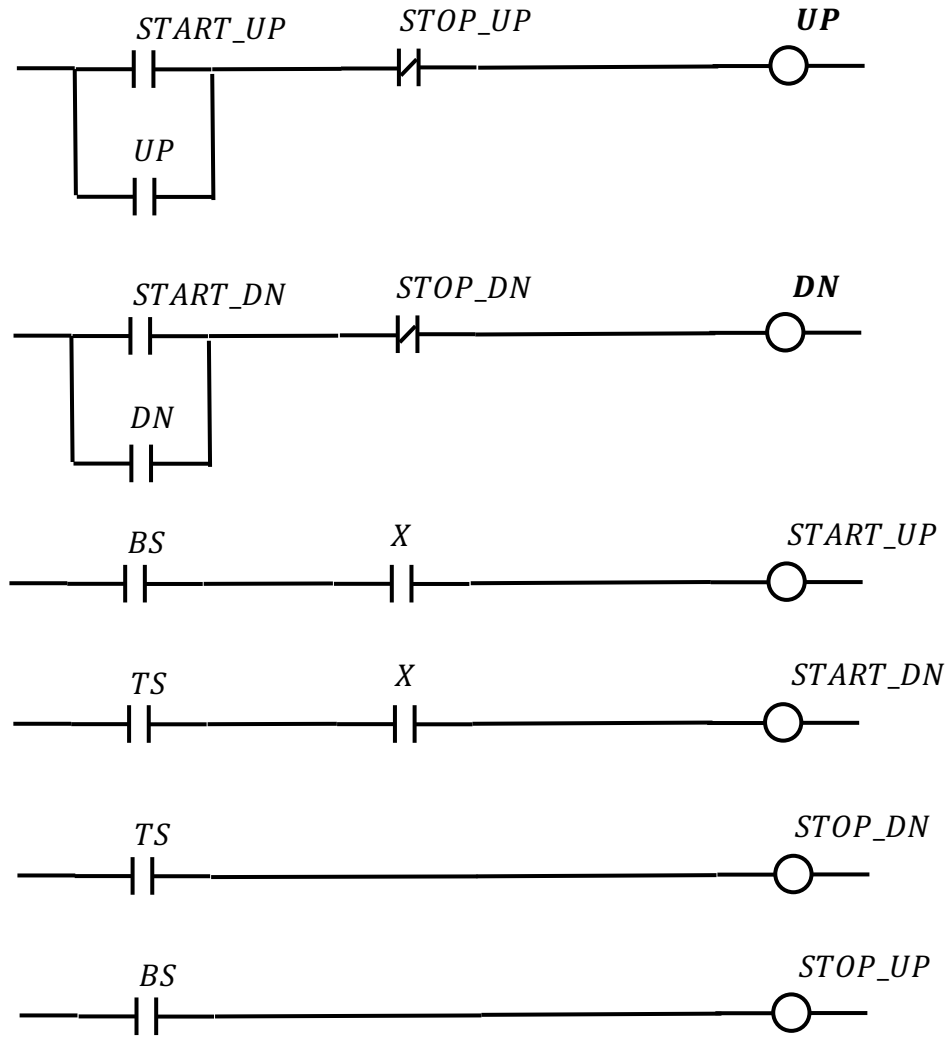
PRZYKŁAD 17



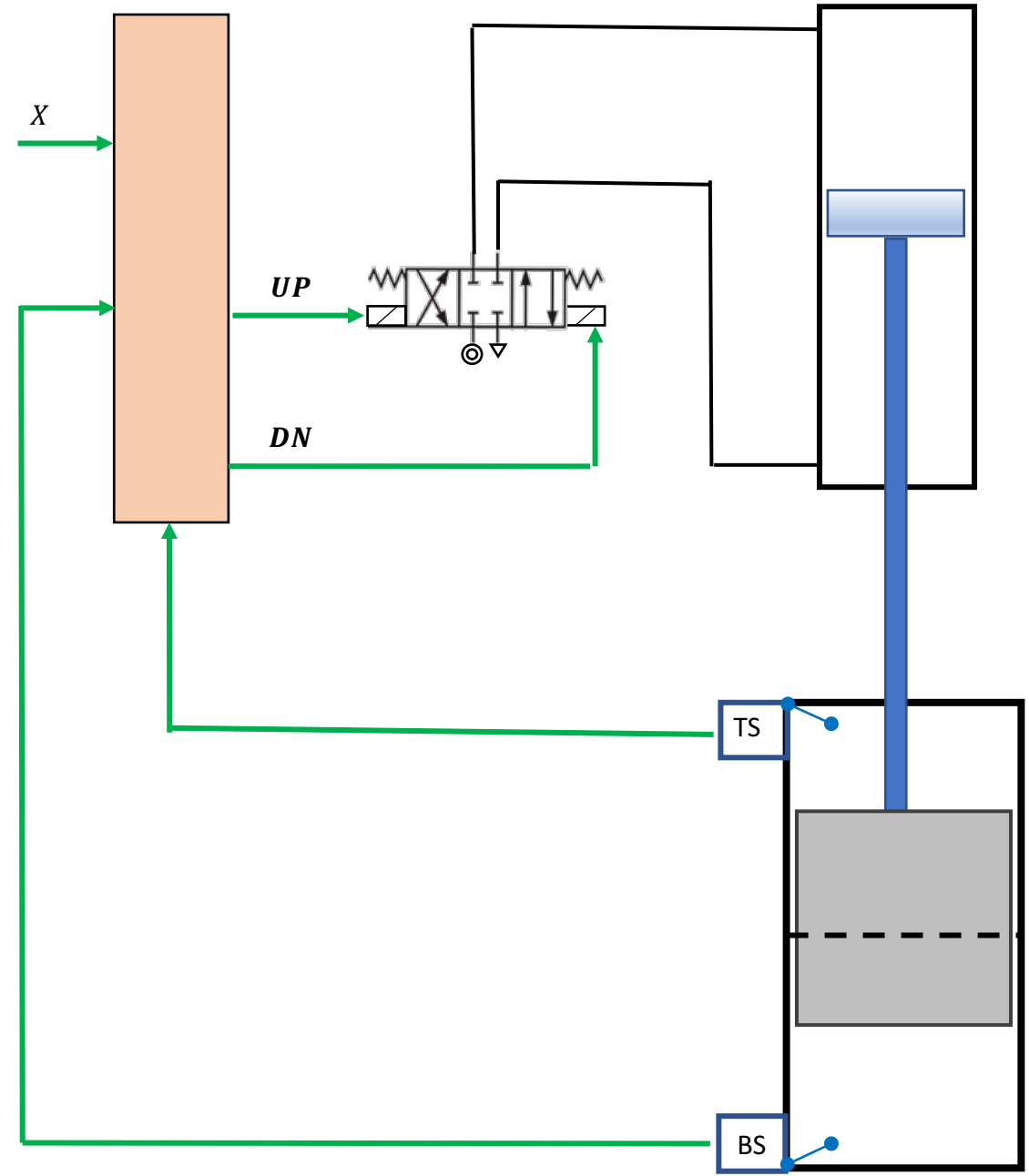
CRAZY LIFT

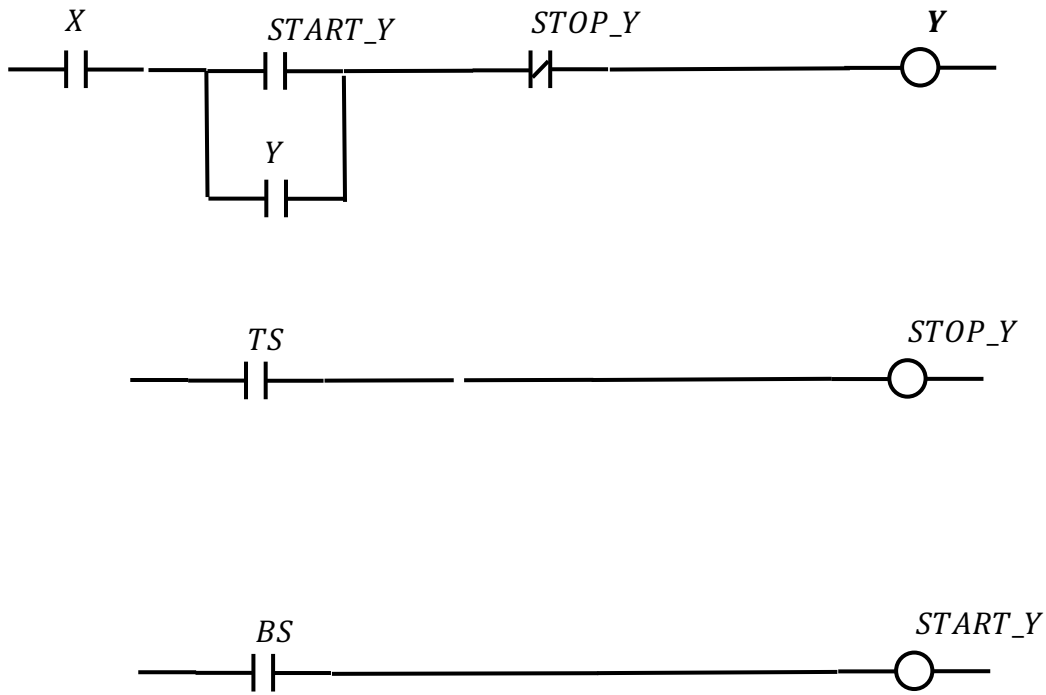


# PRZYKŁAD 18



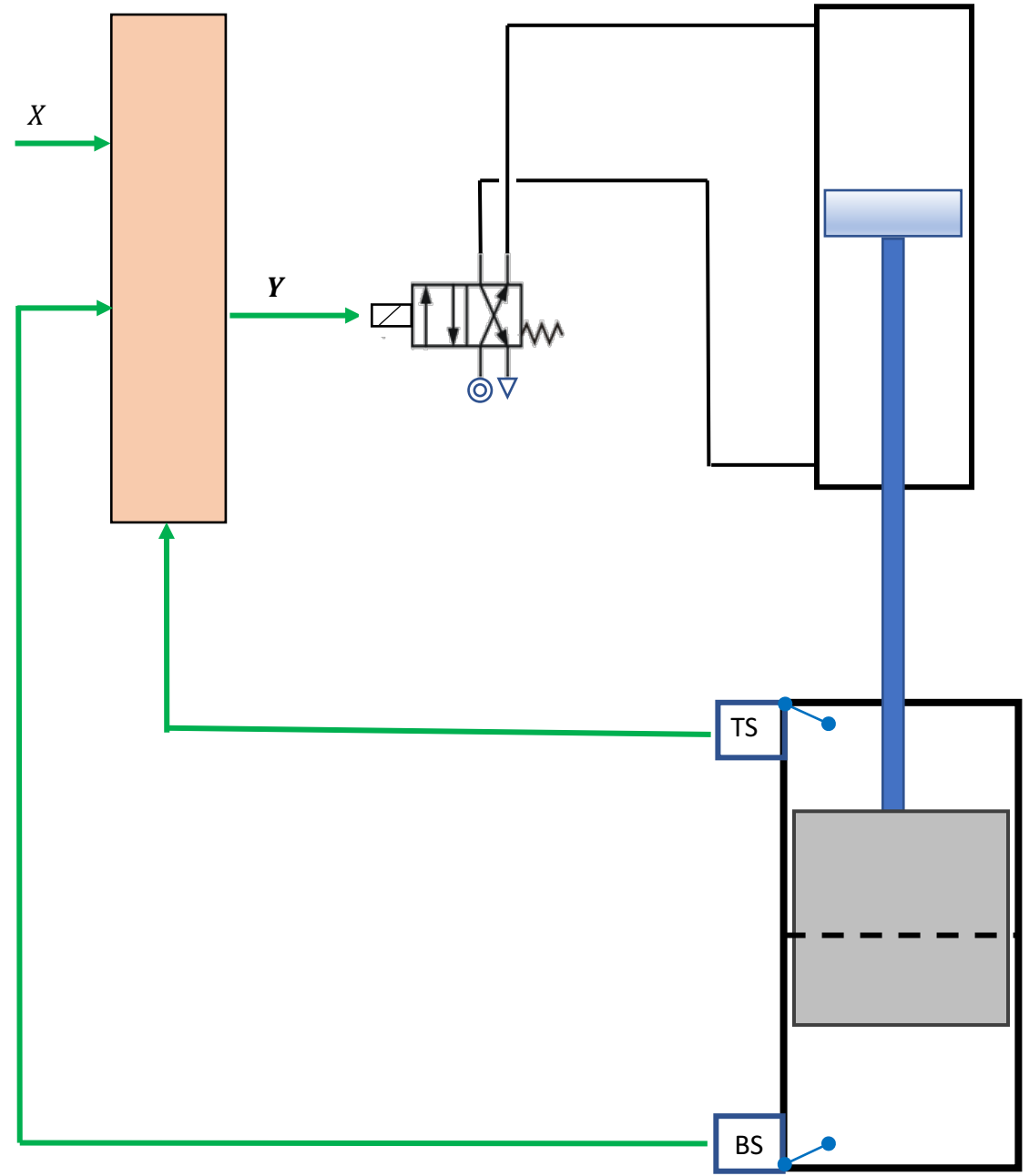
WINDA Z NAPĘDZEM HYDRAULICZNYM

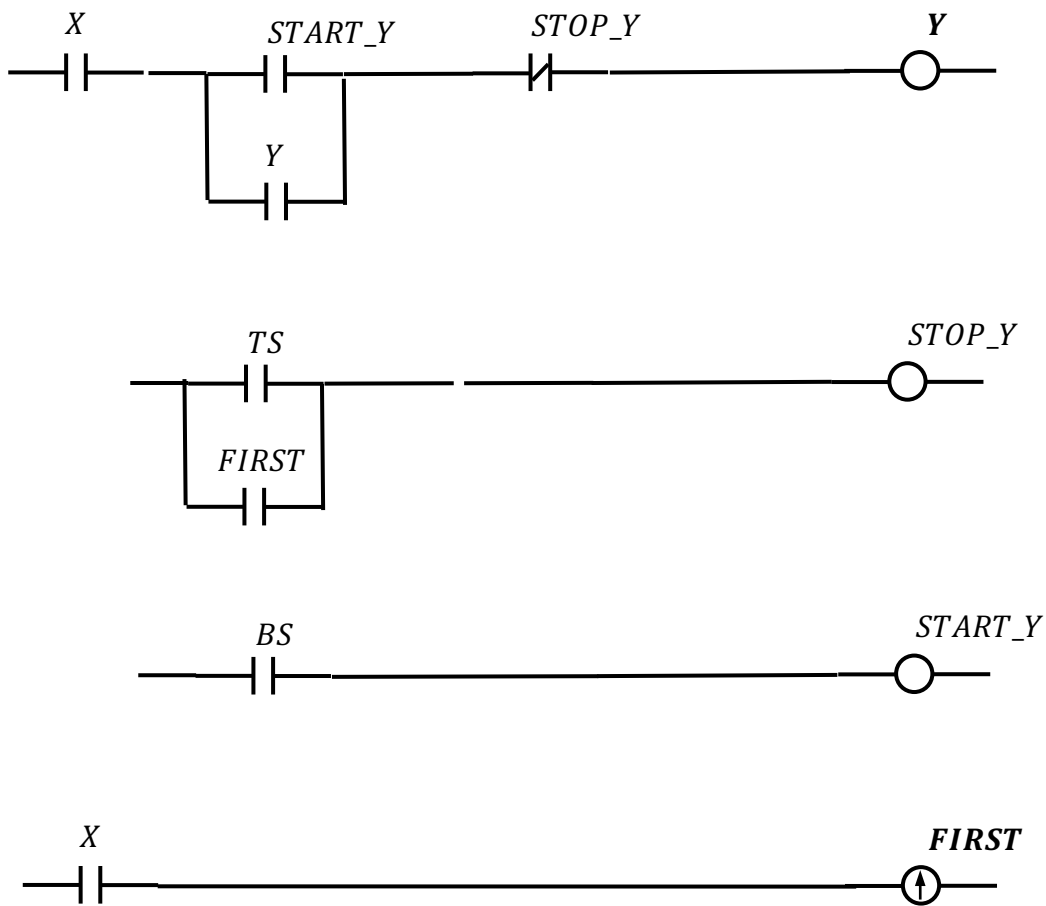




PRZYKŁAD 19

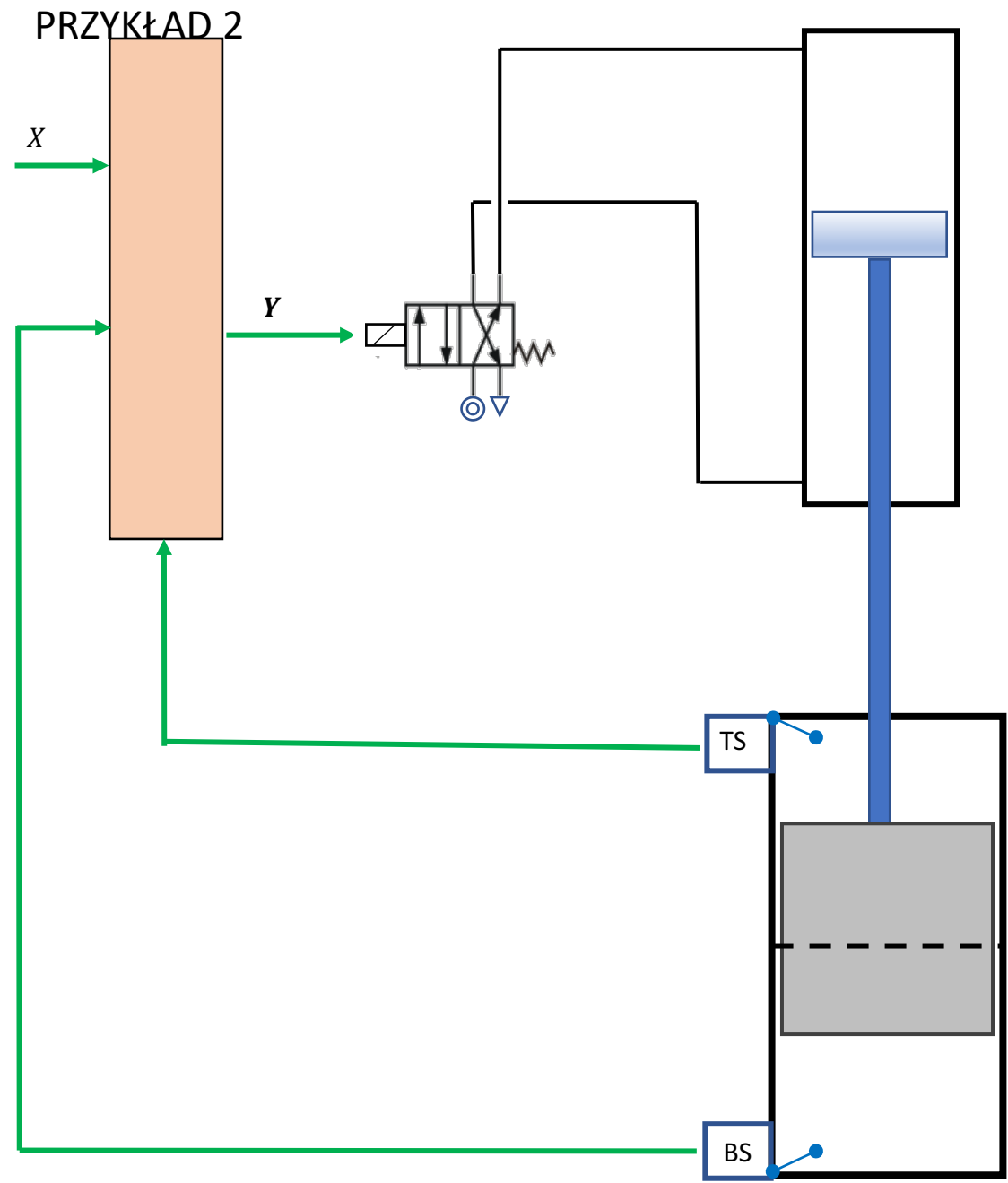
CRAZY LIFT 2

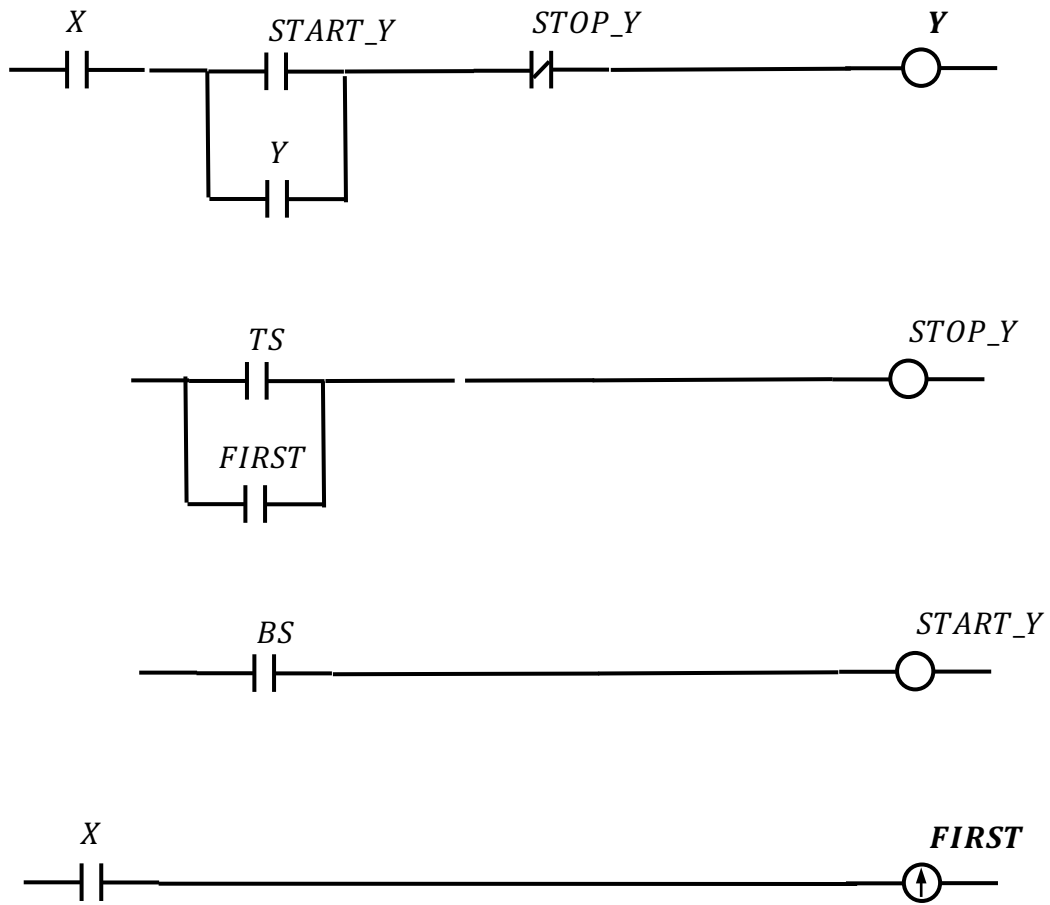




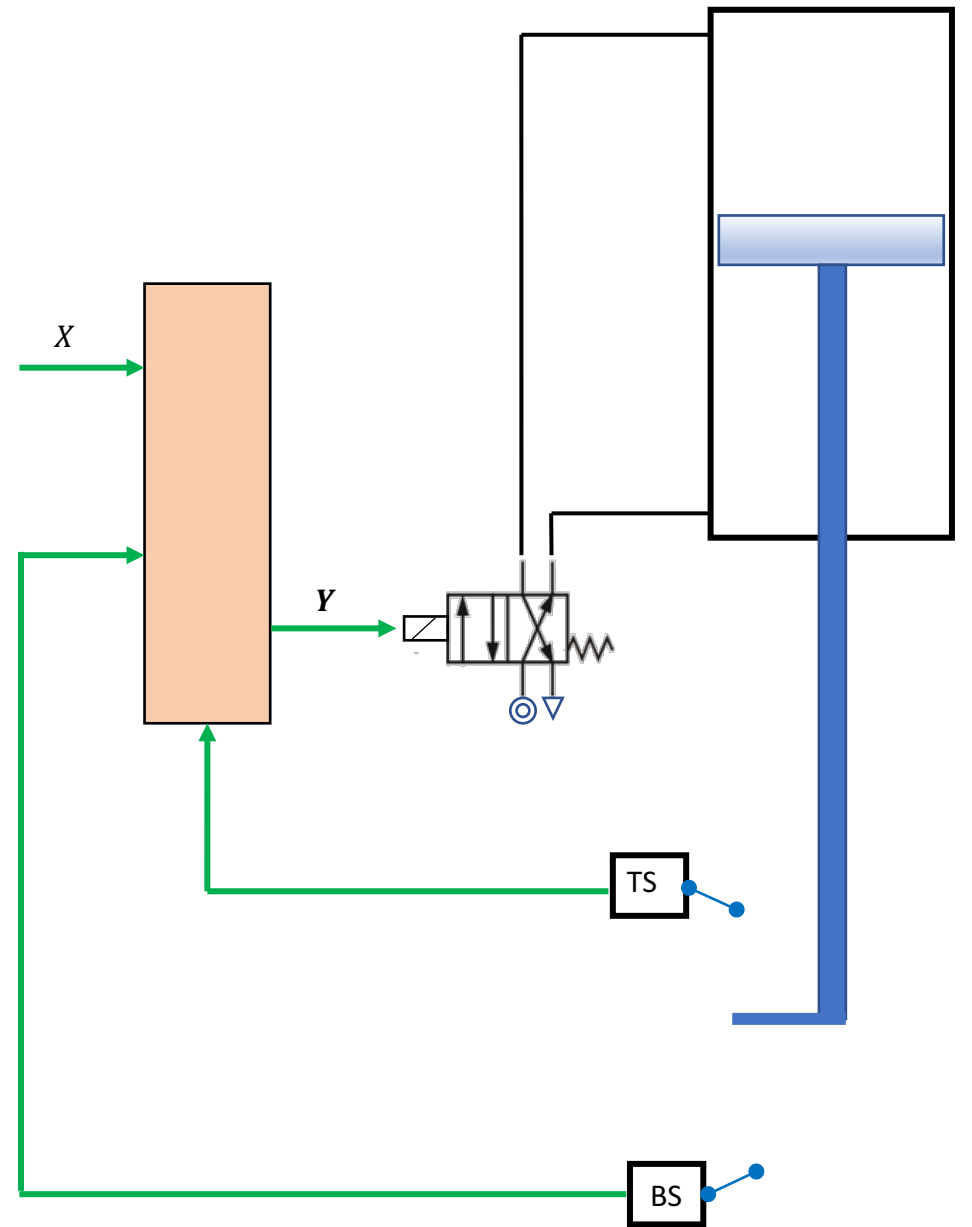
PRZYKŁAD 20

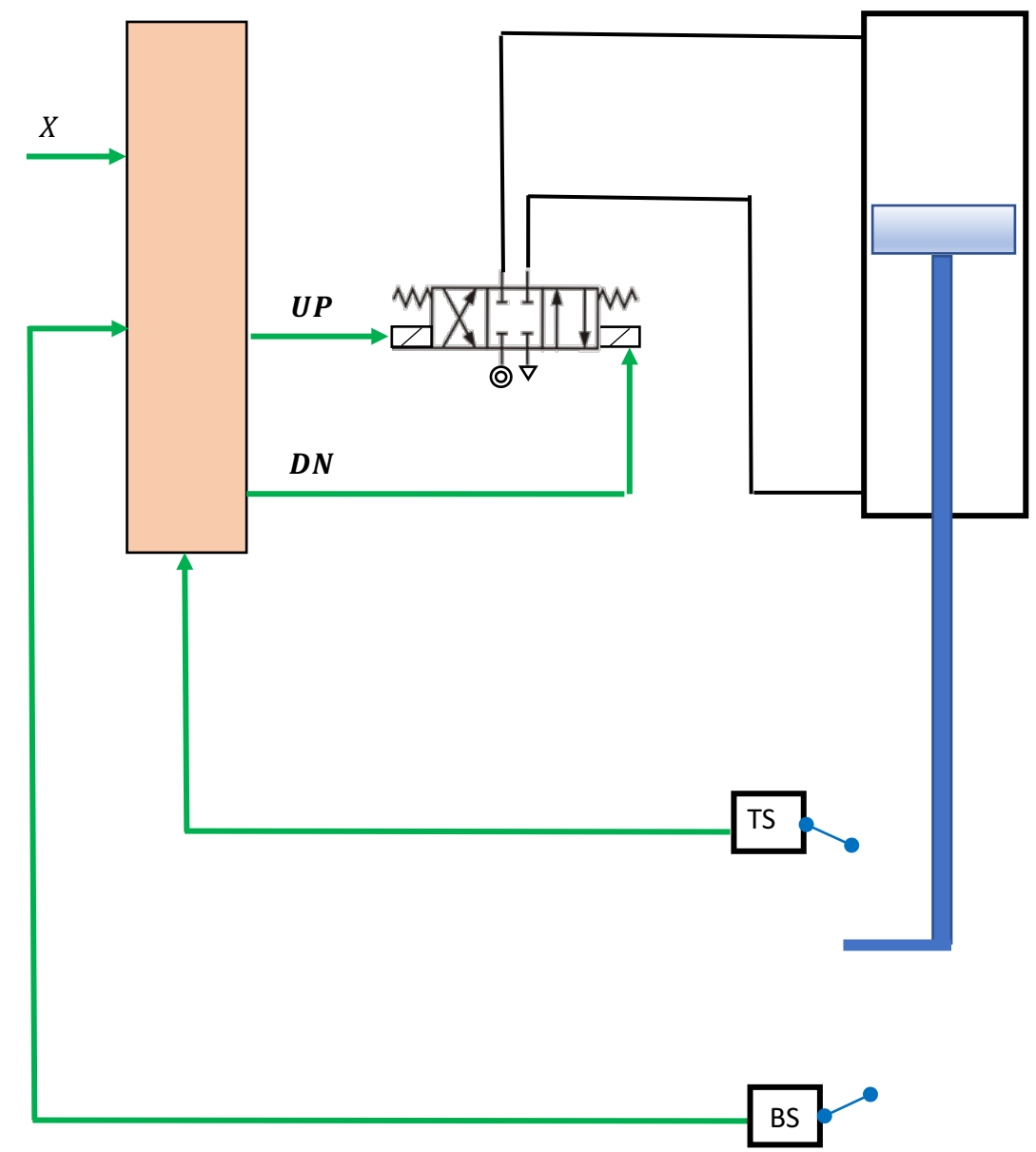
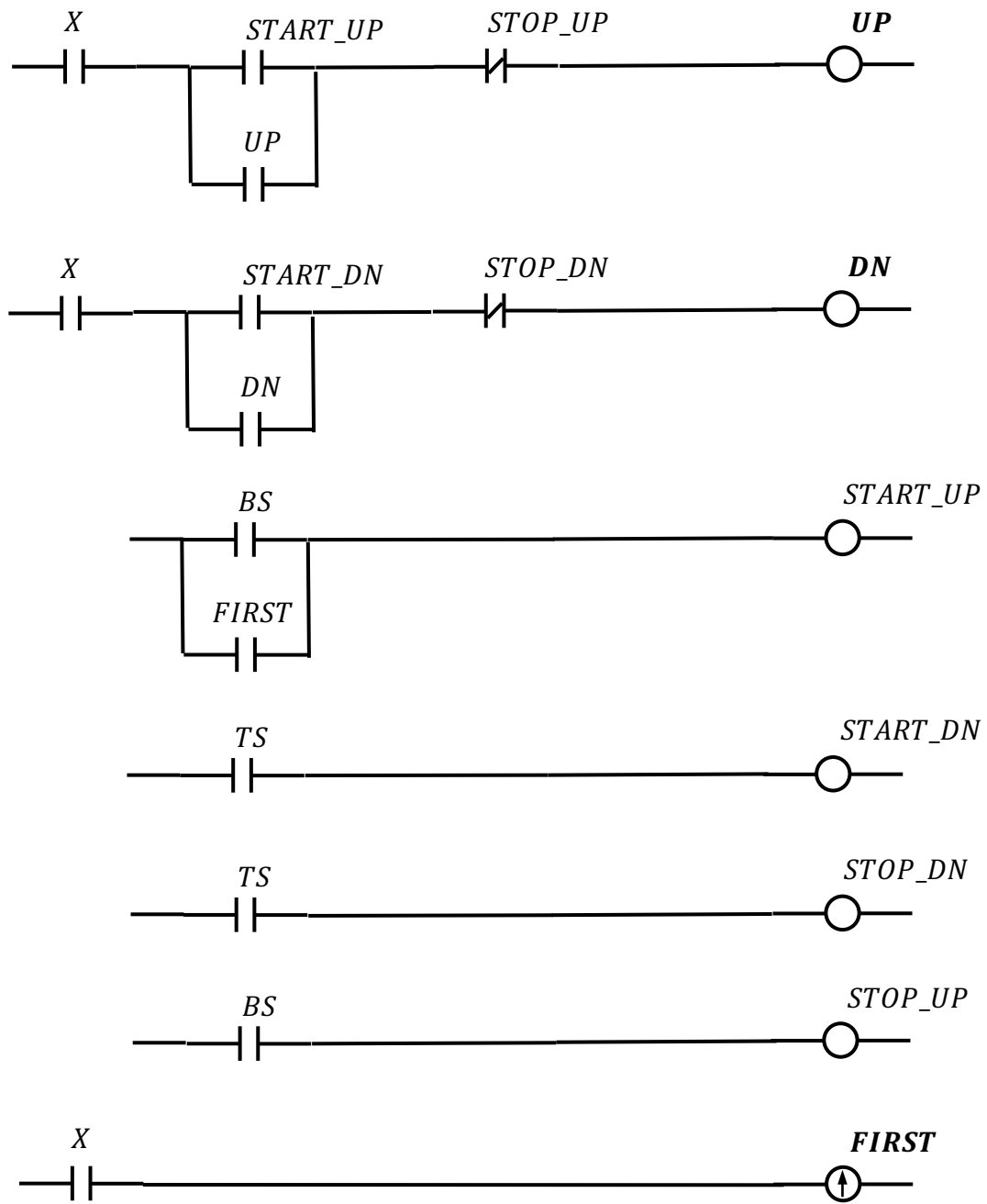
CRAZY LIFT 3 - start z dowolnej pozycji



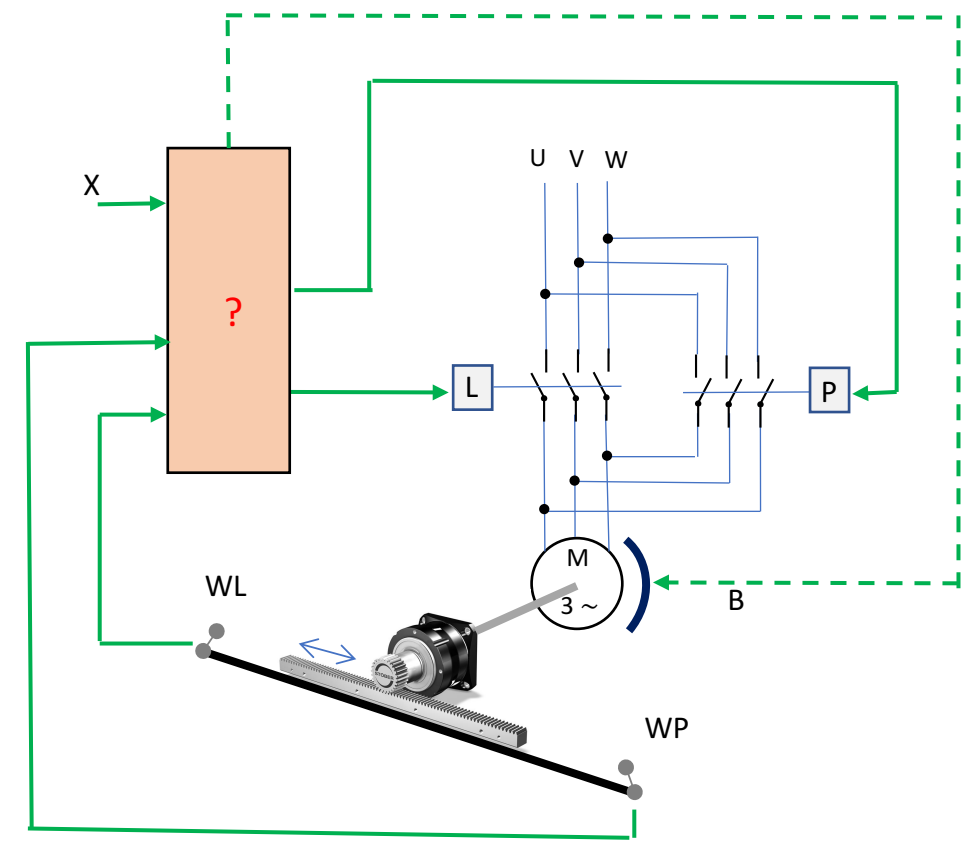
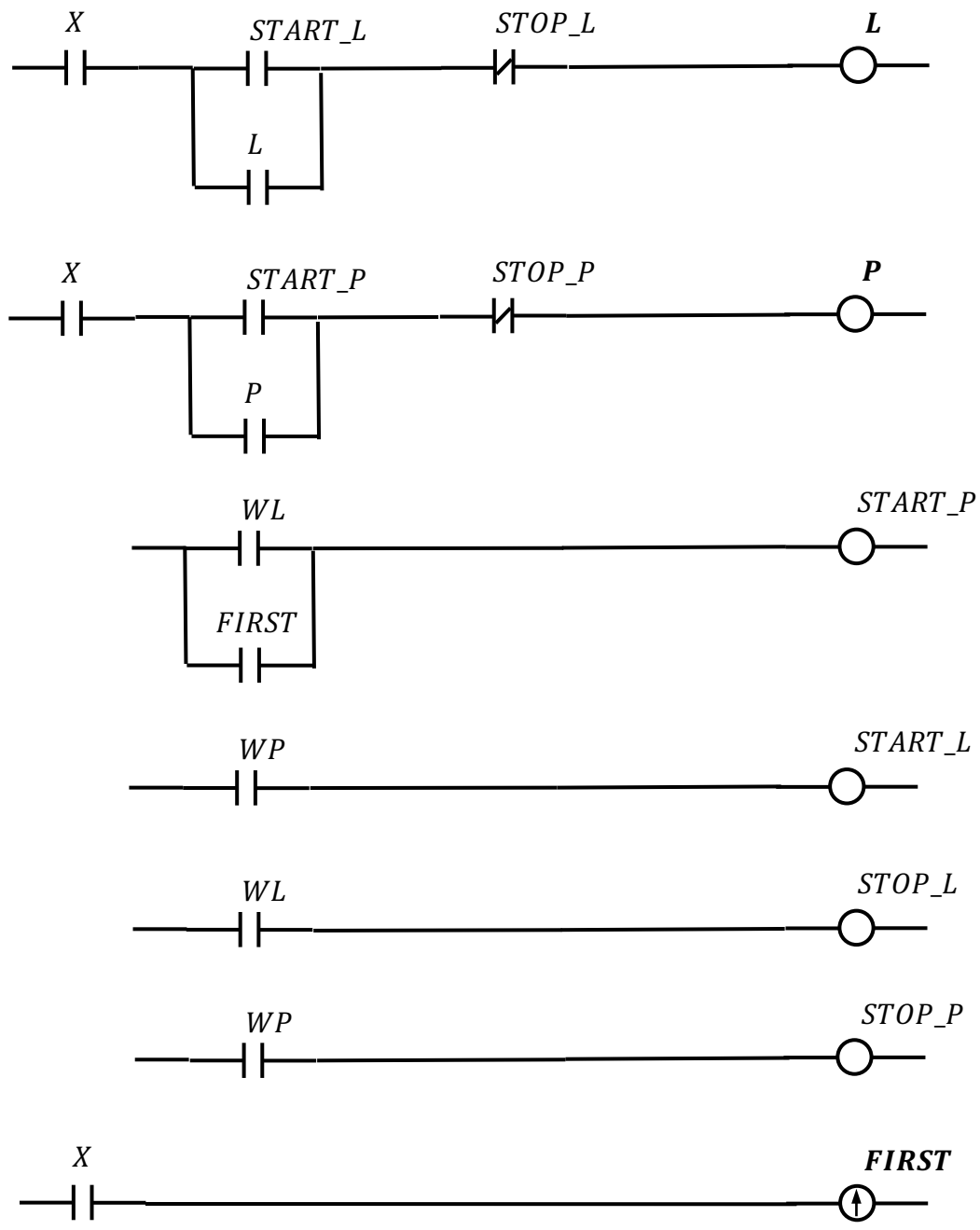


PRZYKŁAD 21

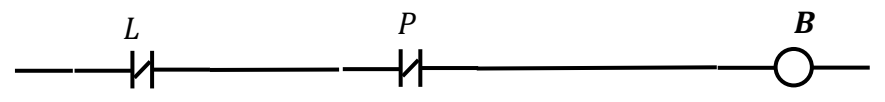




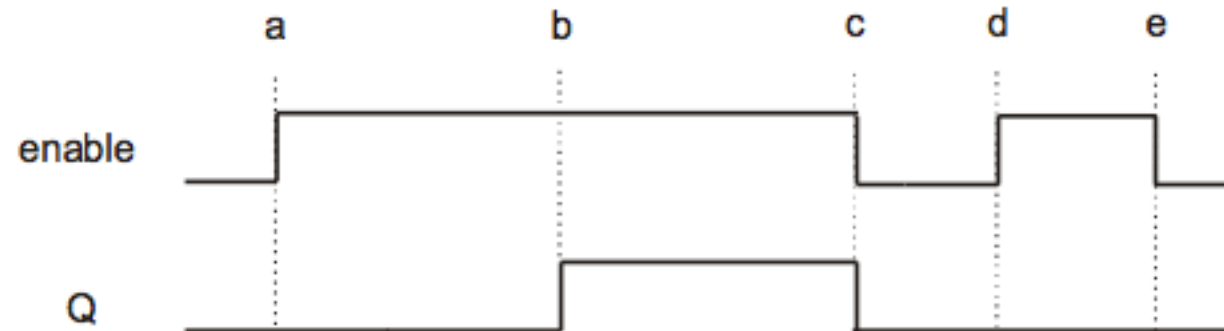
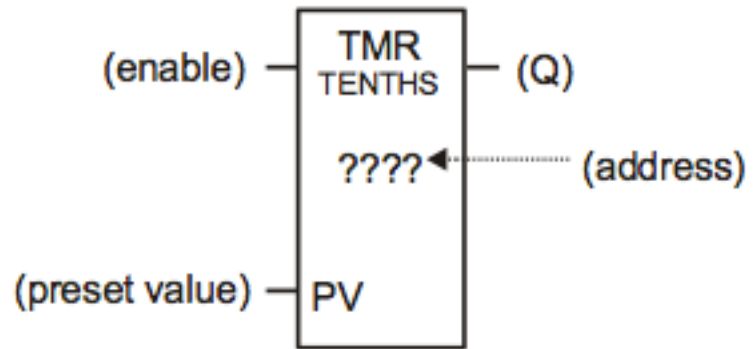
PRZYKŁAD 22

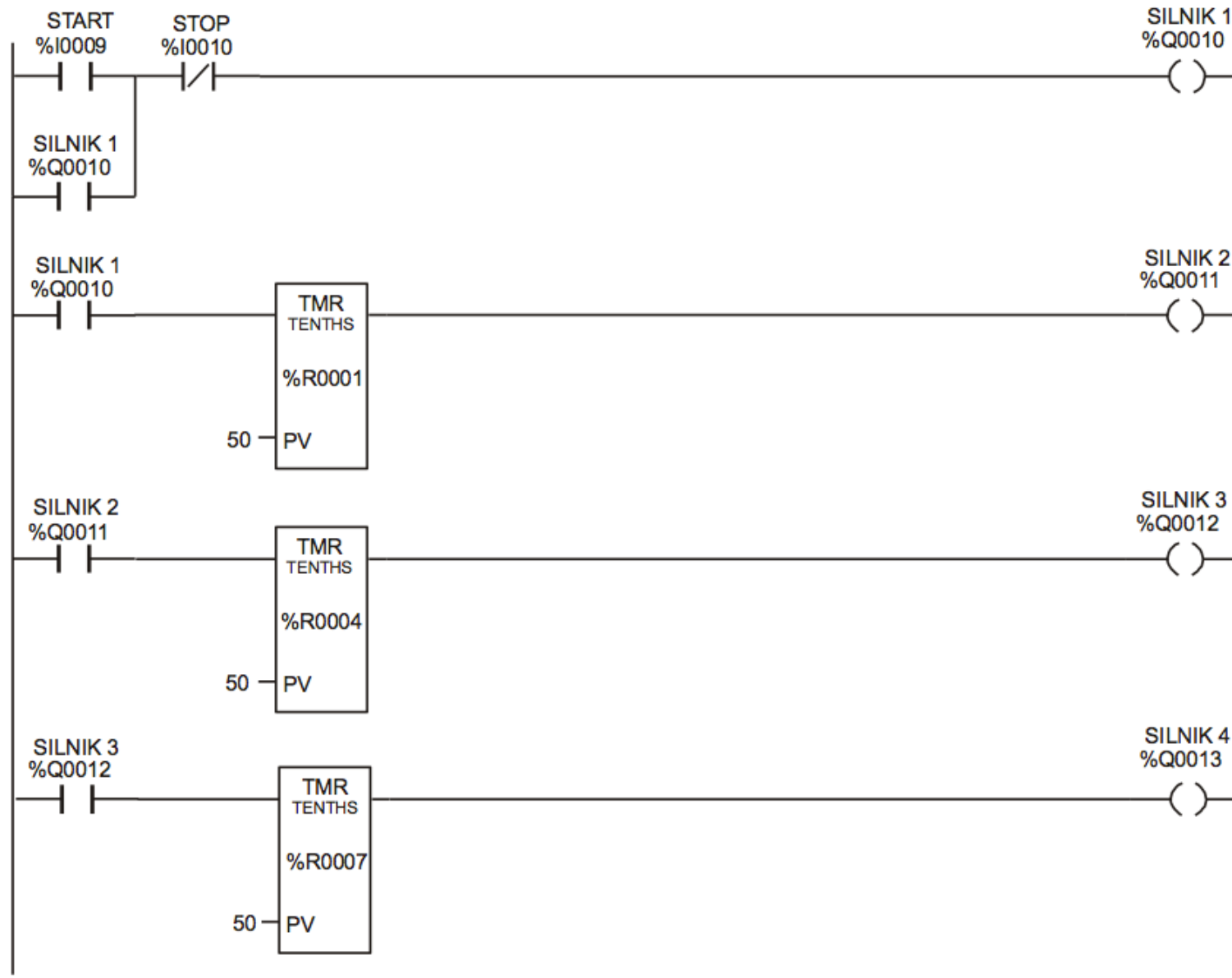


PRZYKŁAD 23  
**RUCH CYKLICZNY**



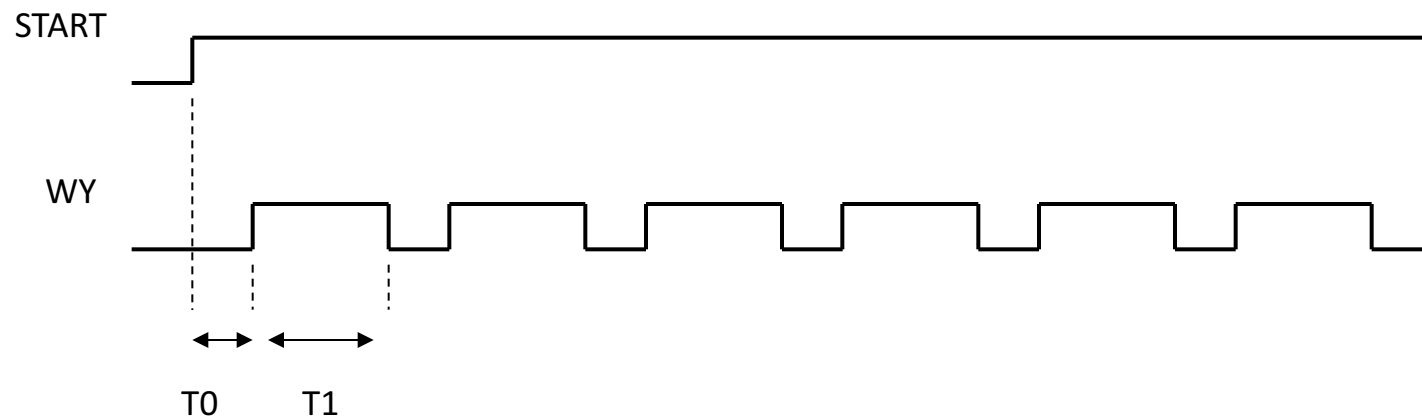
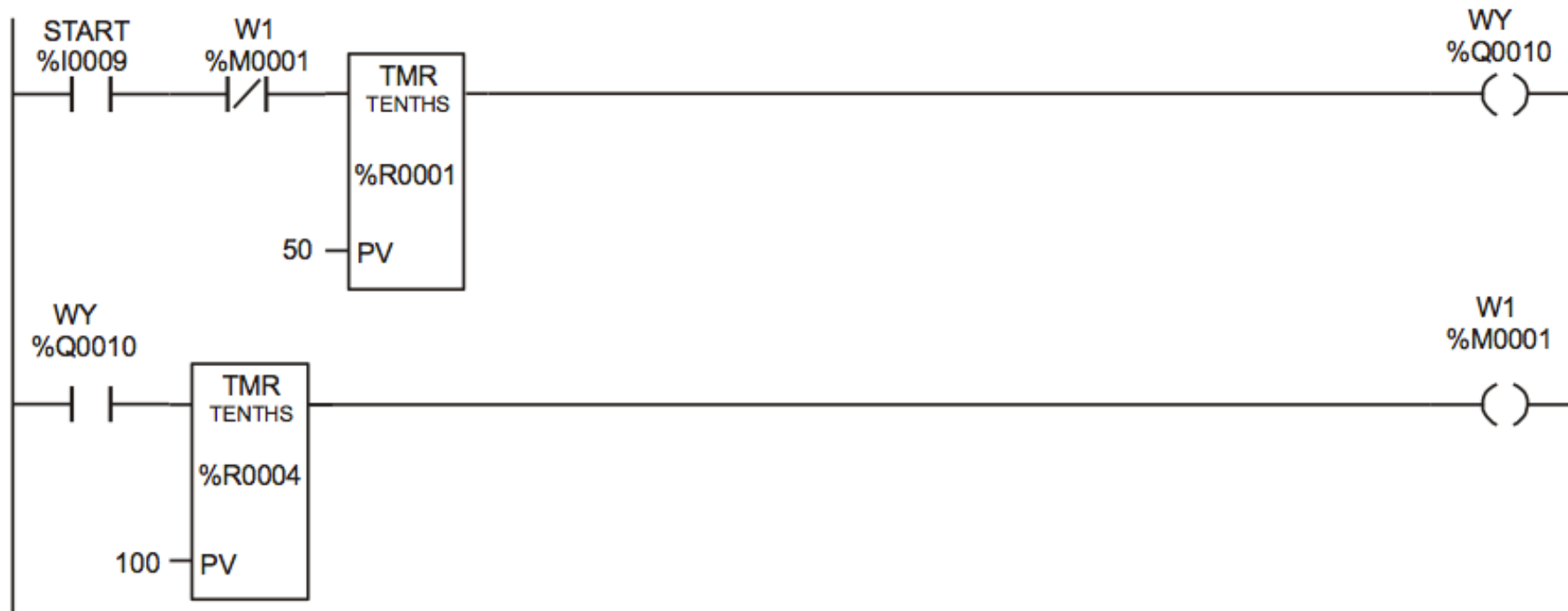
# PRZEKAŹNIK CZASOWY



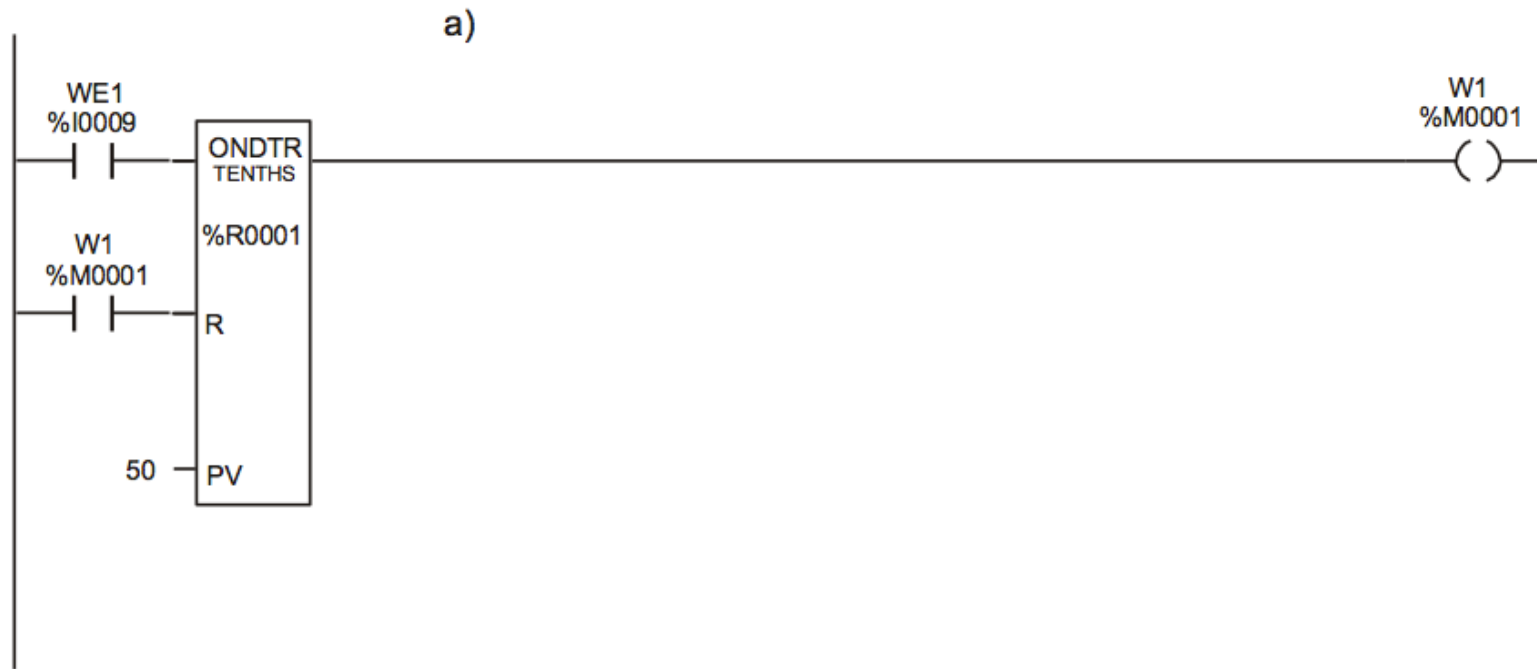


TMR PRZYKŁAD 24  
 Sekwencyjne załączanie 4 silników

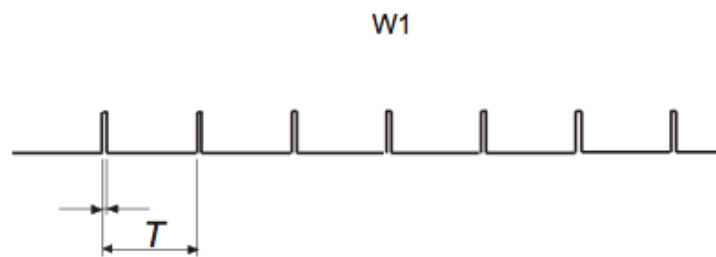


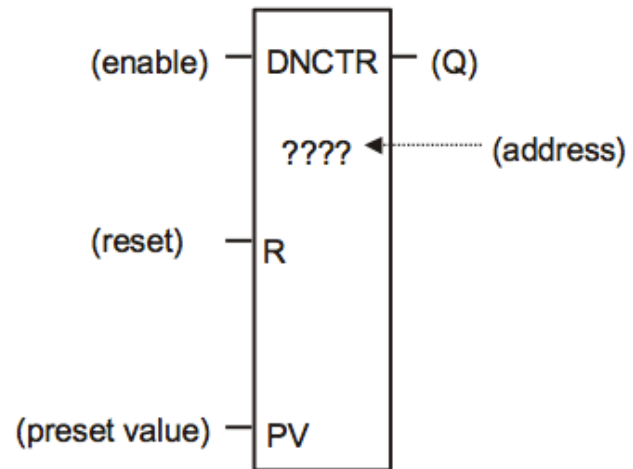
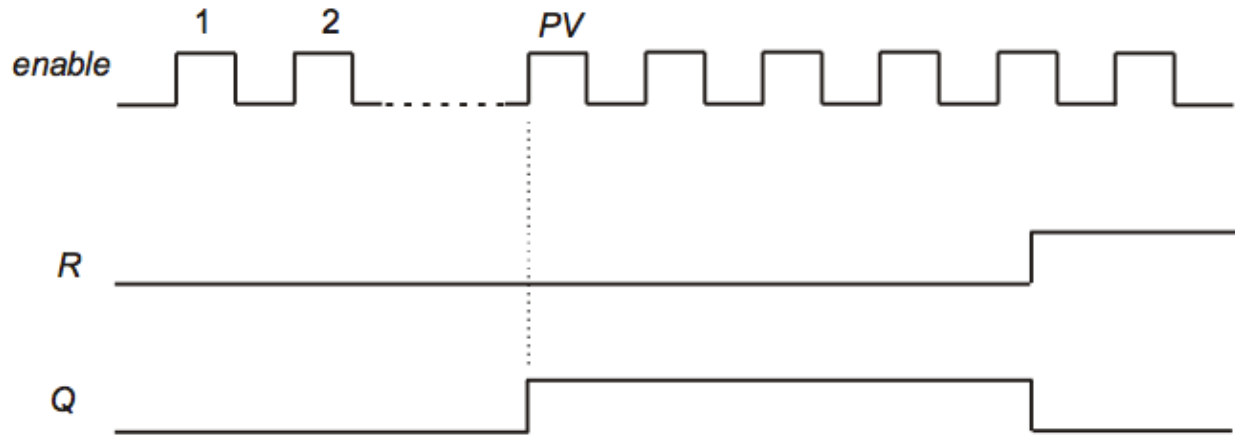
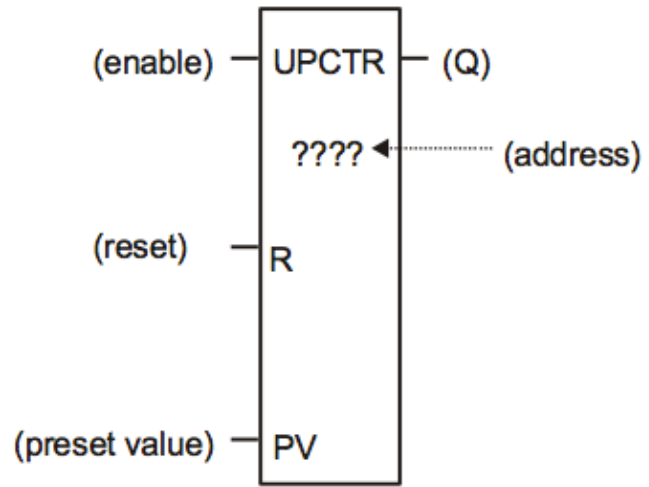


TMR PRZYKŁAD 25  
Generator fali prostokątnej

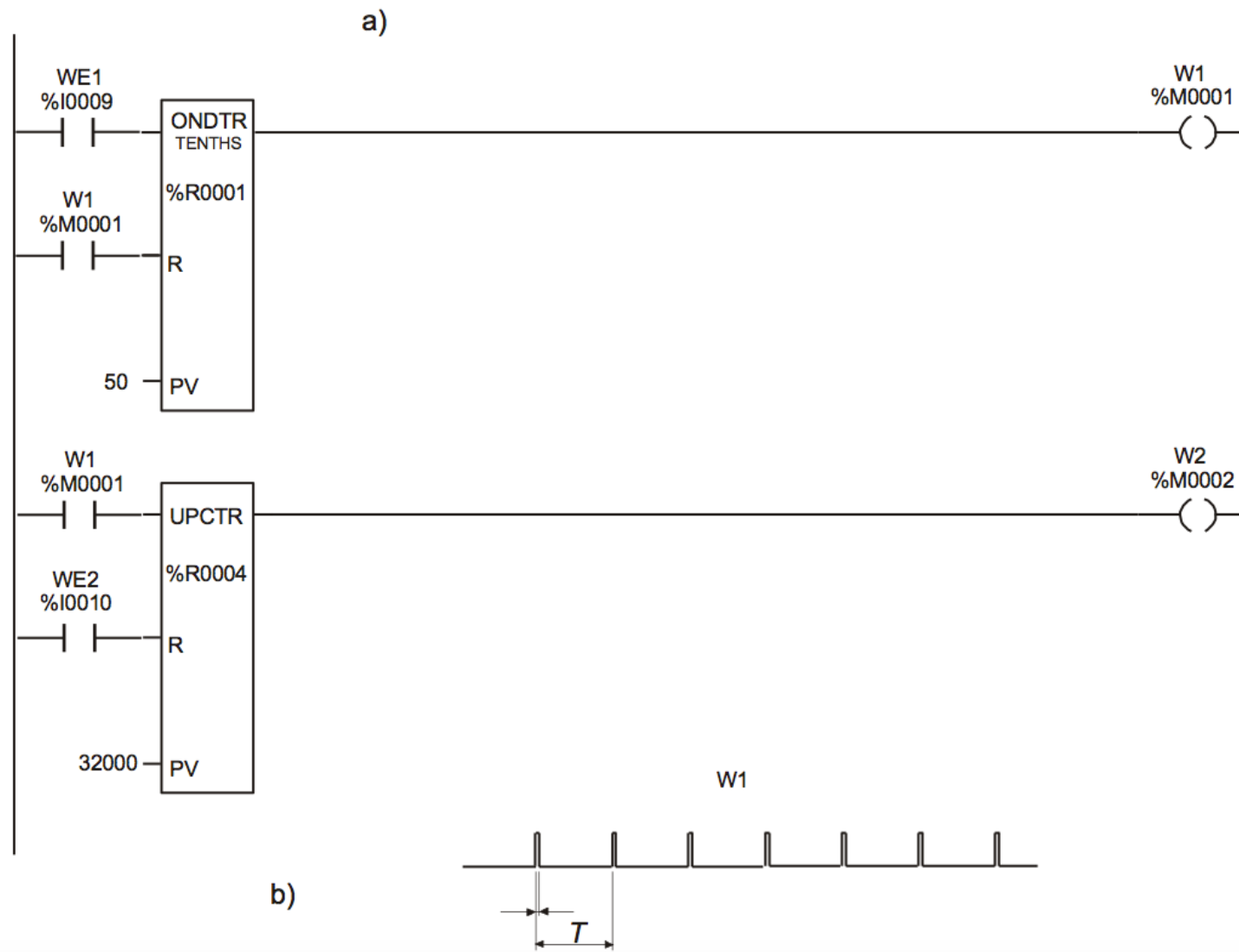


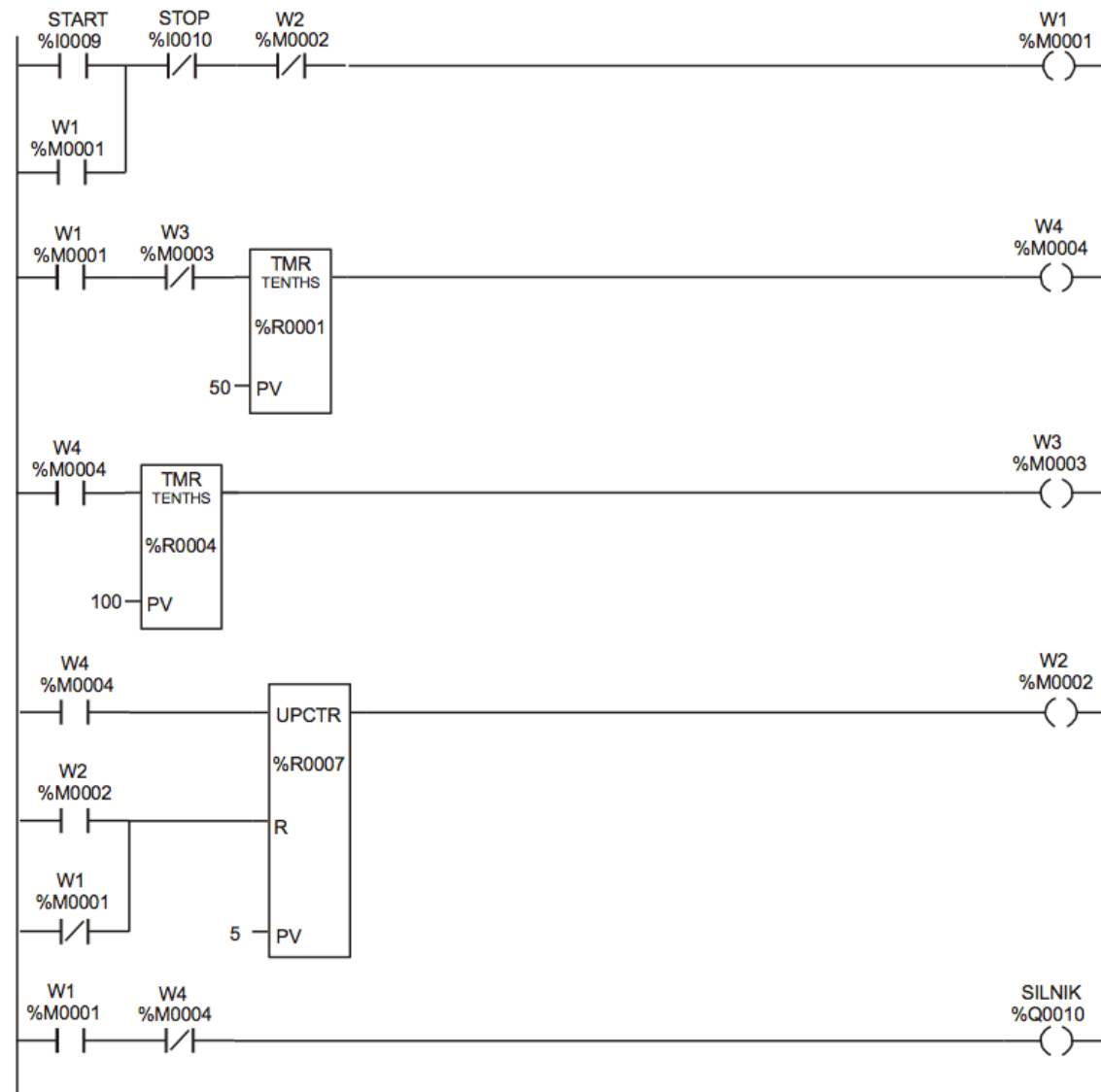
b)



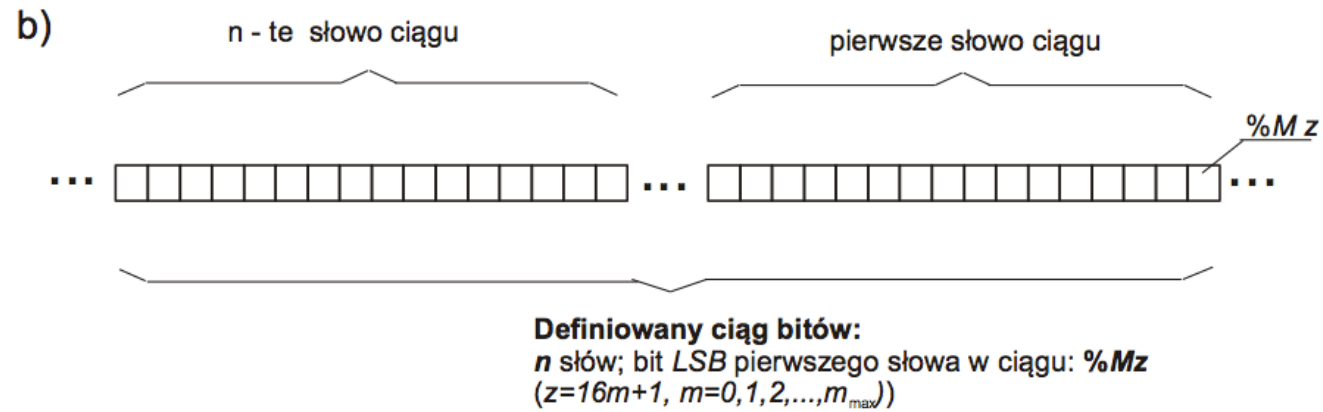
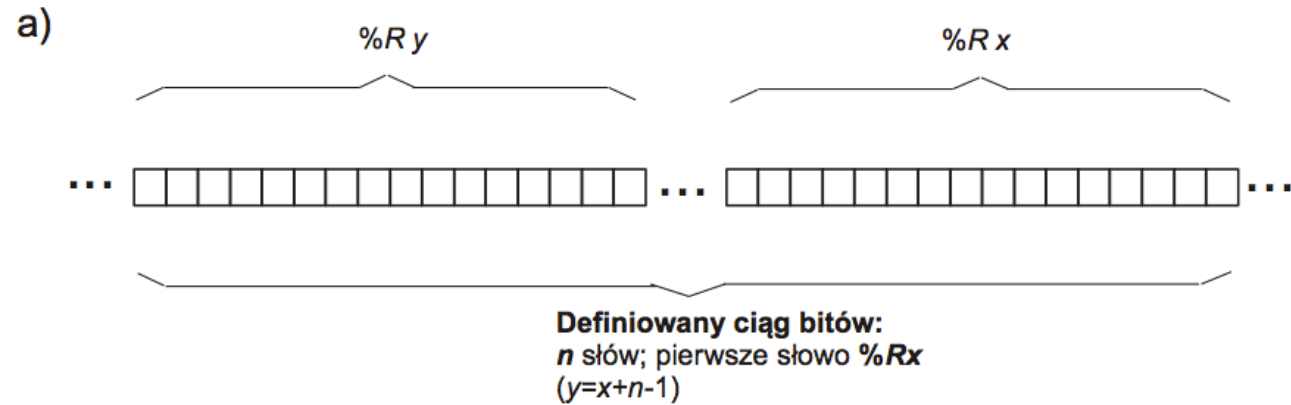


LICZNIKI





# DZIAŁANIA NA CIĄGACH BITÓW



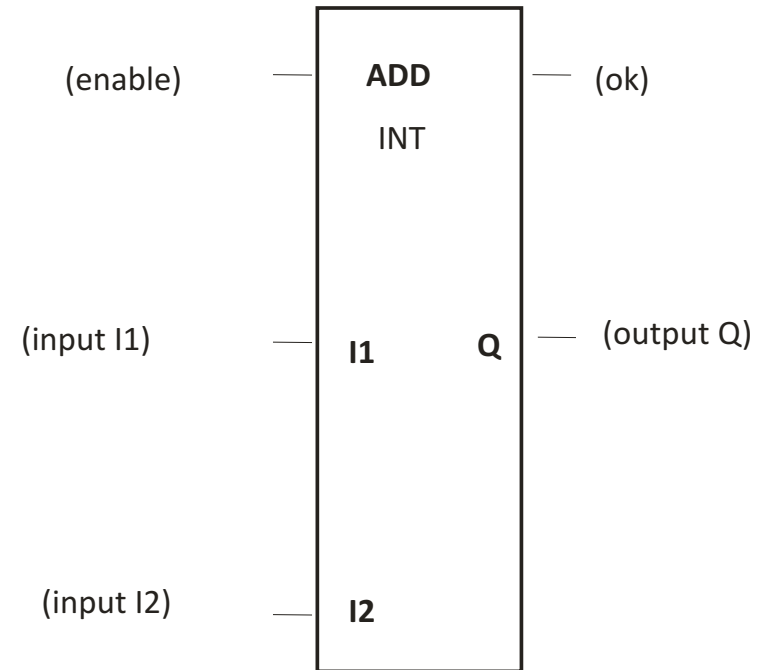
# Działania matematyczne

*ADD* – dodawanie dwóch liczb

*SUB* – odejmowanie dwóch liczb

*MUL* – mnożenie dwóch liczb

*DIV* – dzielenie dwóch liczb



## Relacje matematyczne

*EQ* – równe

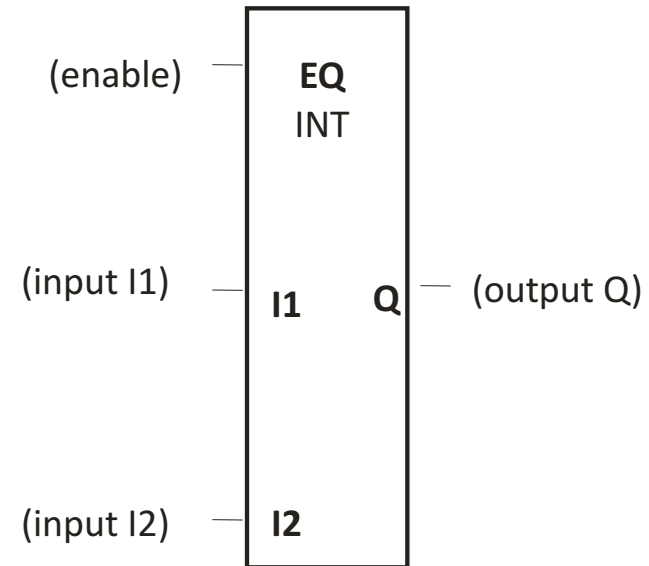
*NE* – nierówne

*GT* – większe

*GE* – większe lub równe

*LT* – mniejsze

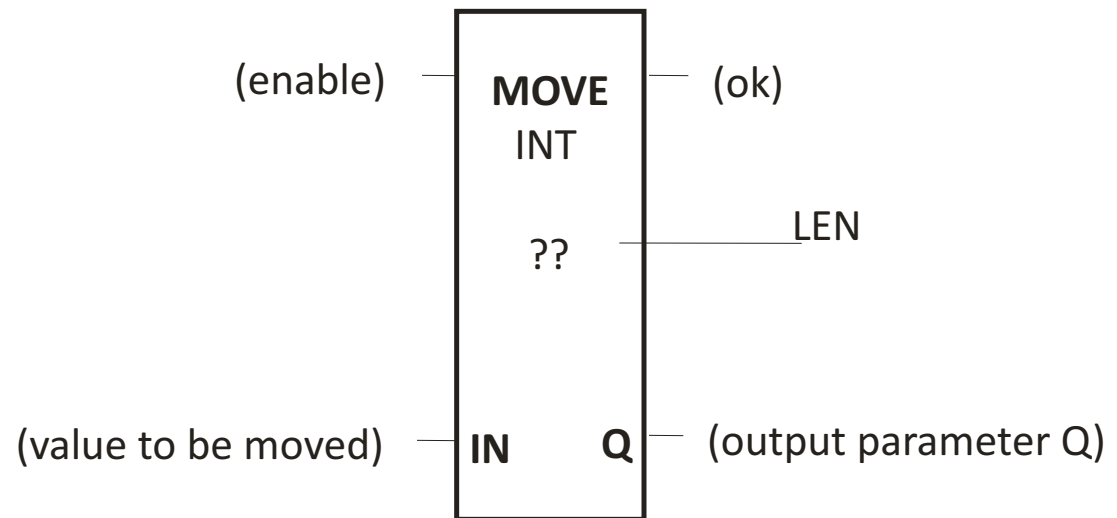
*LE* – mniejsze lub równe

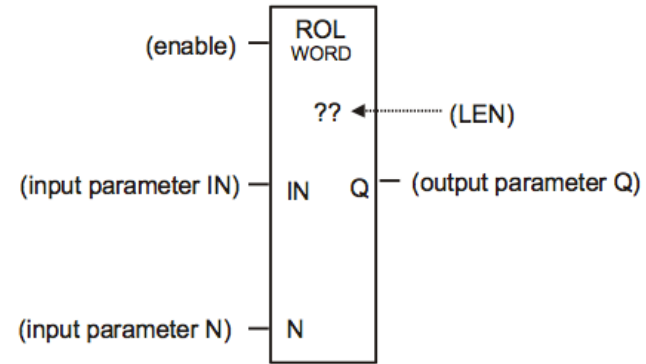
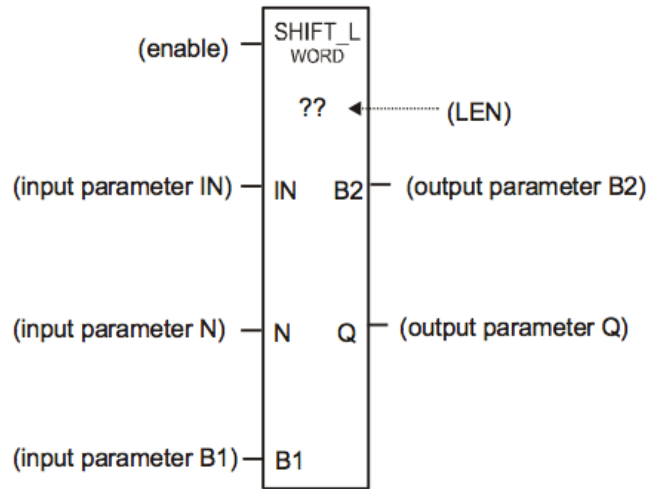
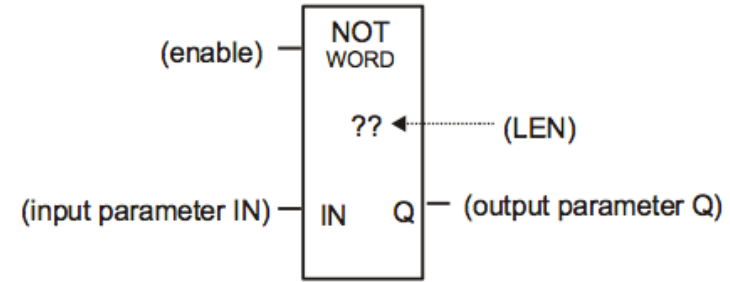
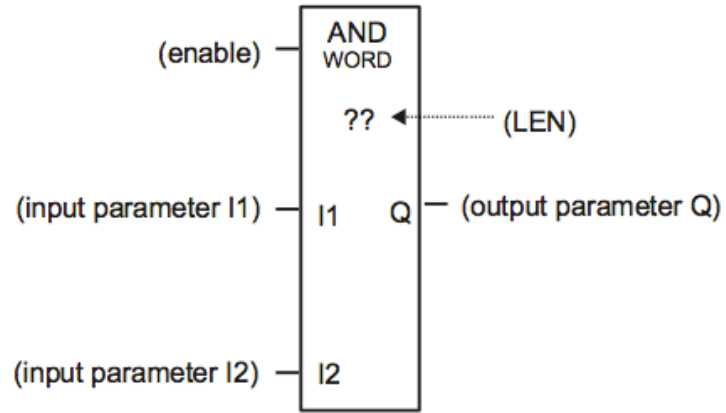


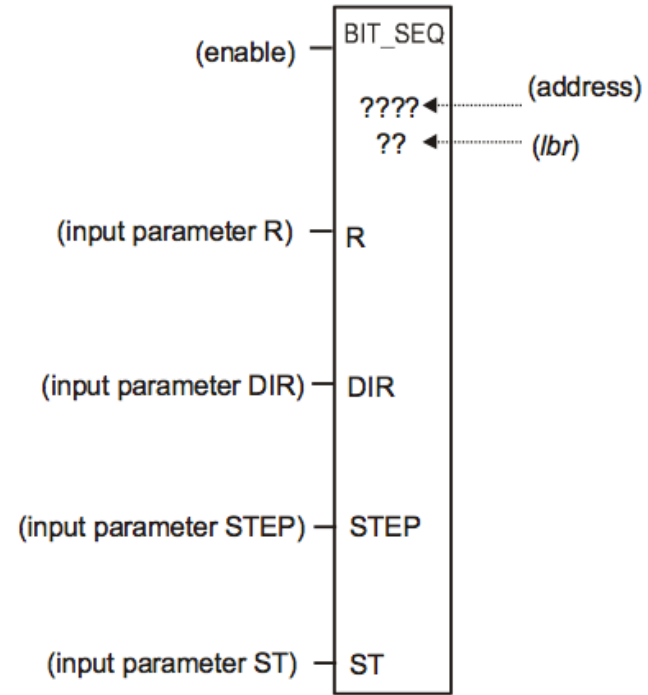
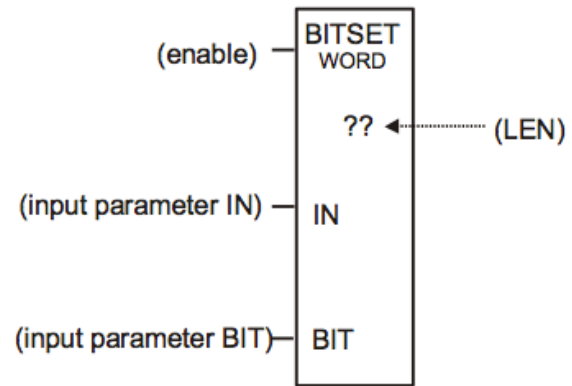


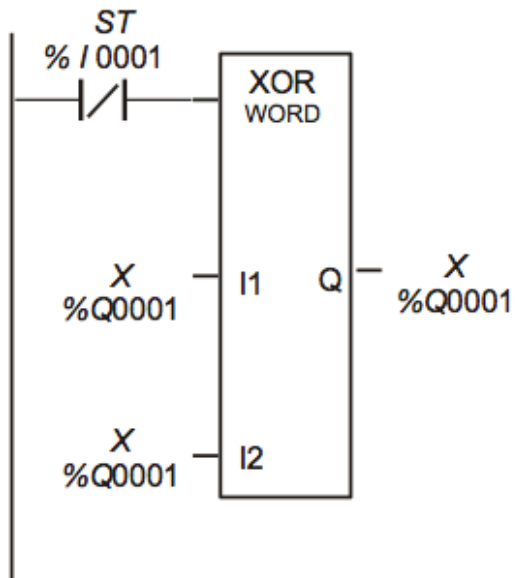
## Przesyłanie danych

Blok *MOVE*



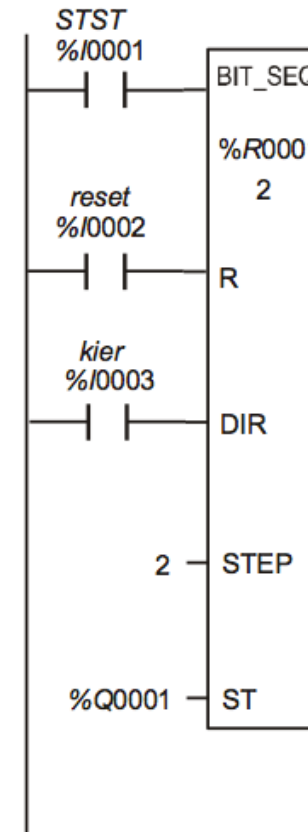






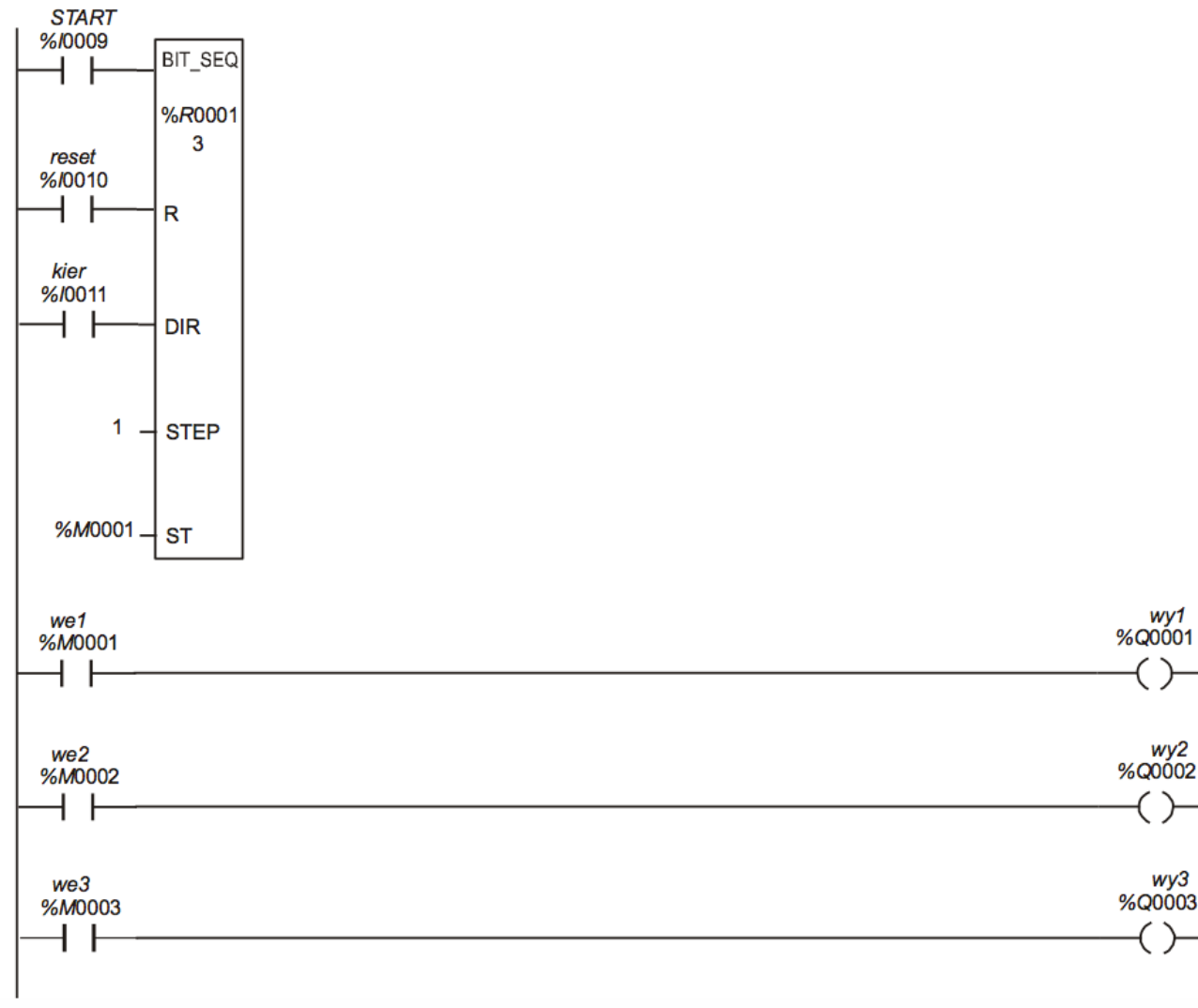
Zerowanie rejestru

DZIAŁANIA NA CIĄGACH BITÓW - PRZYKŁAD 29



Start-Stop -> jednym przyciskiem

DZIAŁANIA NA CIĄGACH BITÓW - PRZYKŁAD 30

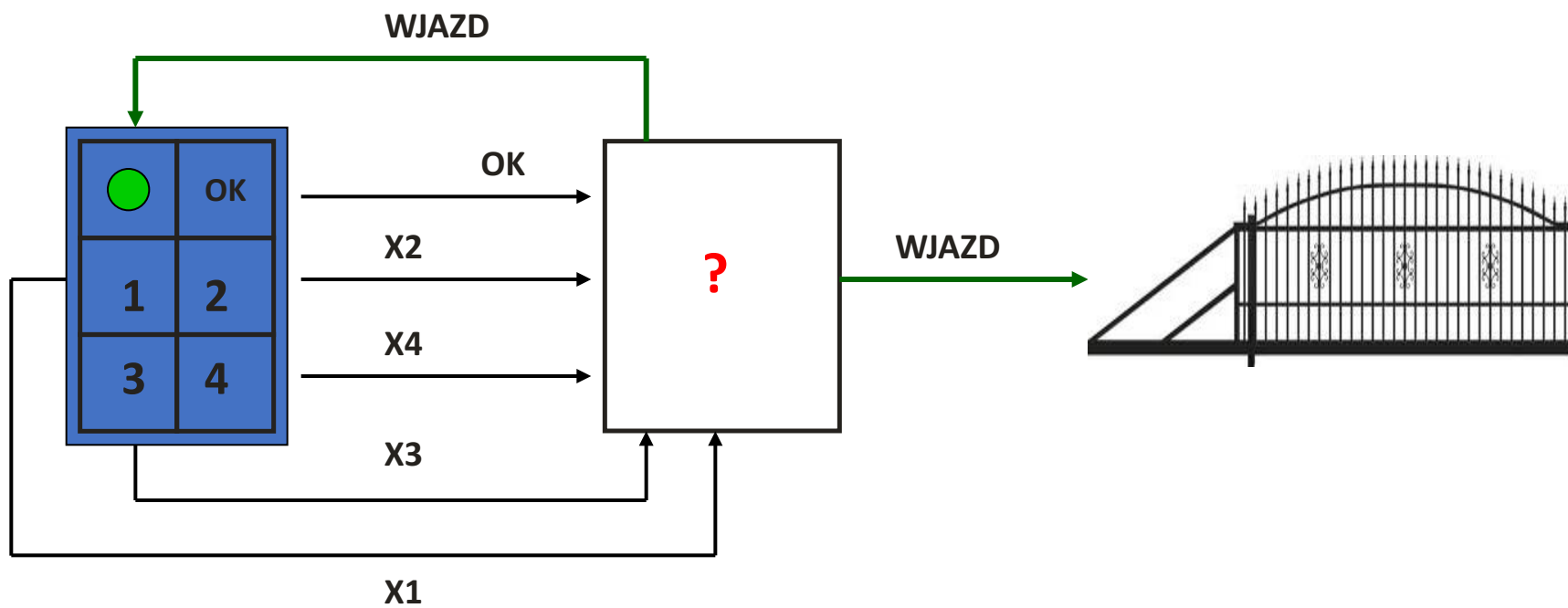


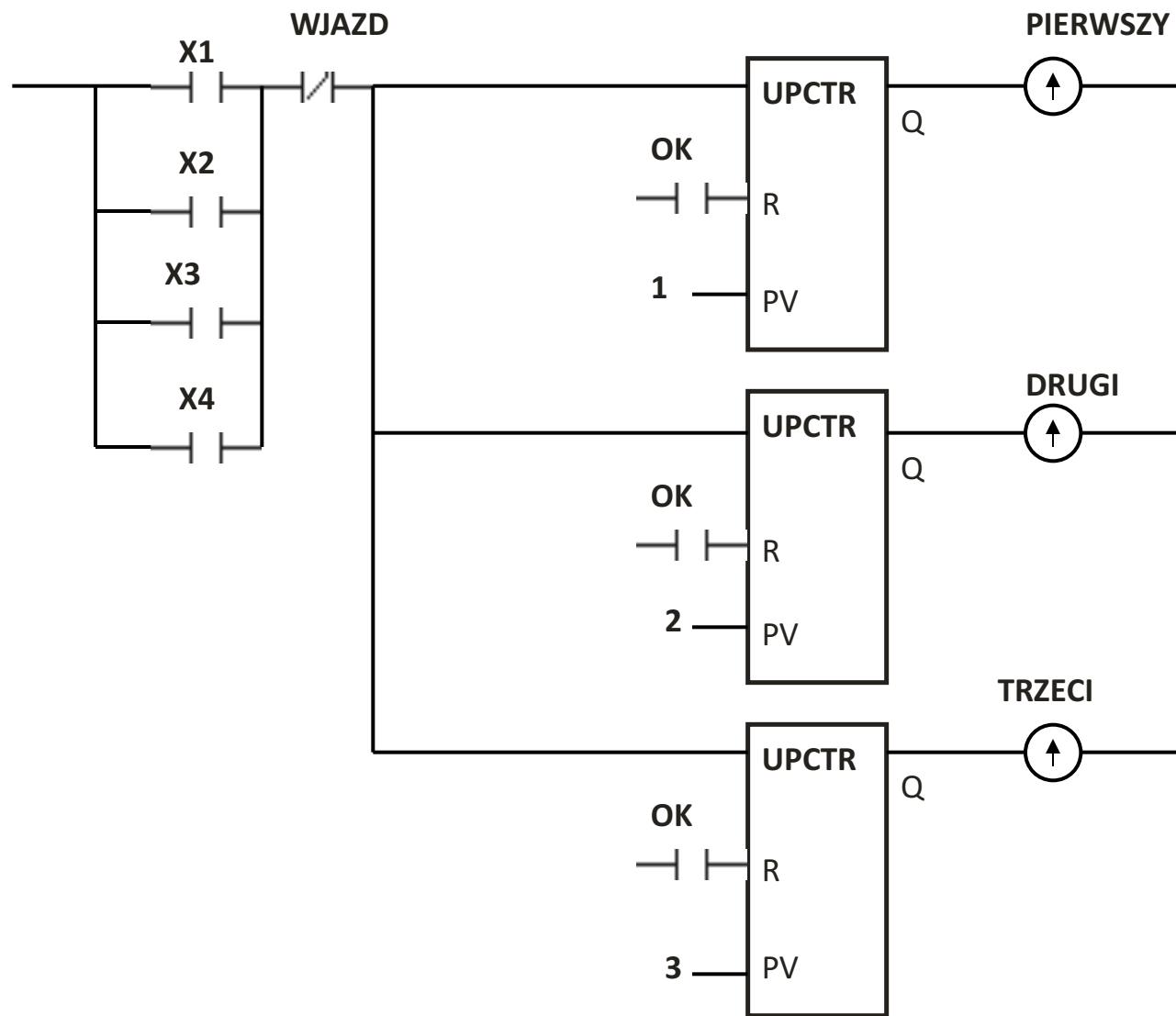
### PRZYKŁAD 31

Dzielnik częstotliwości

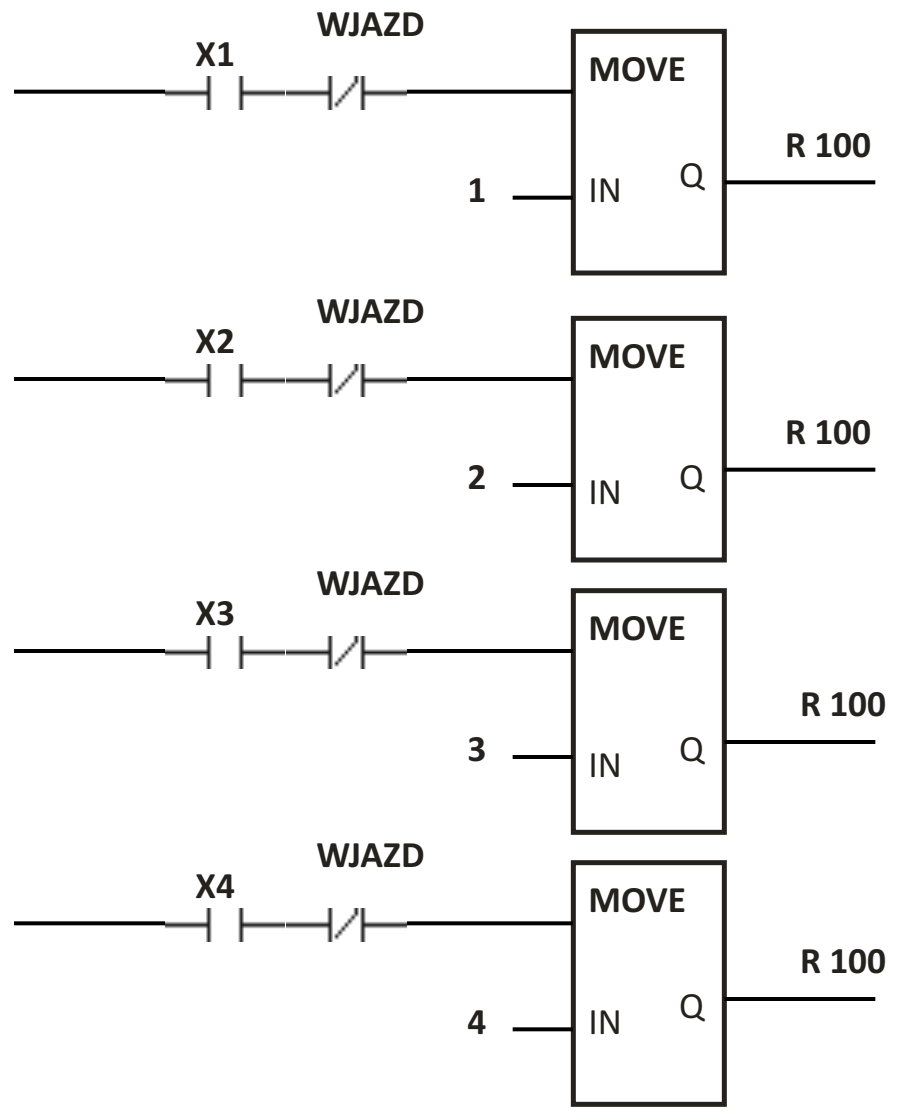
PRZYKŁAD 32: **KOD DOSTĘPU**

(program sterujący na kolejnych 6 rysunkach)



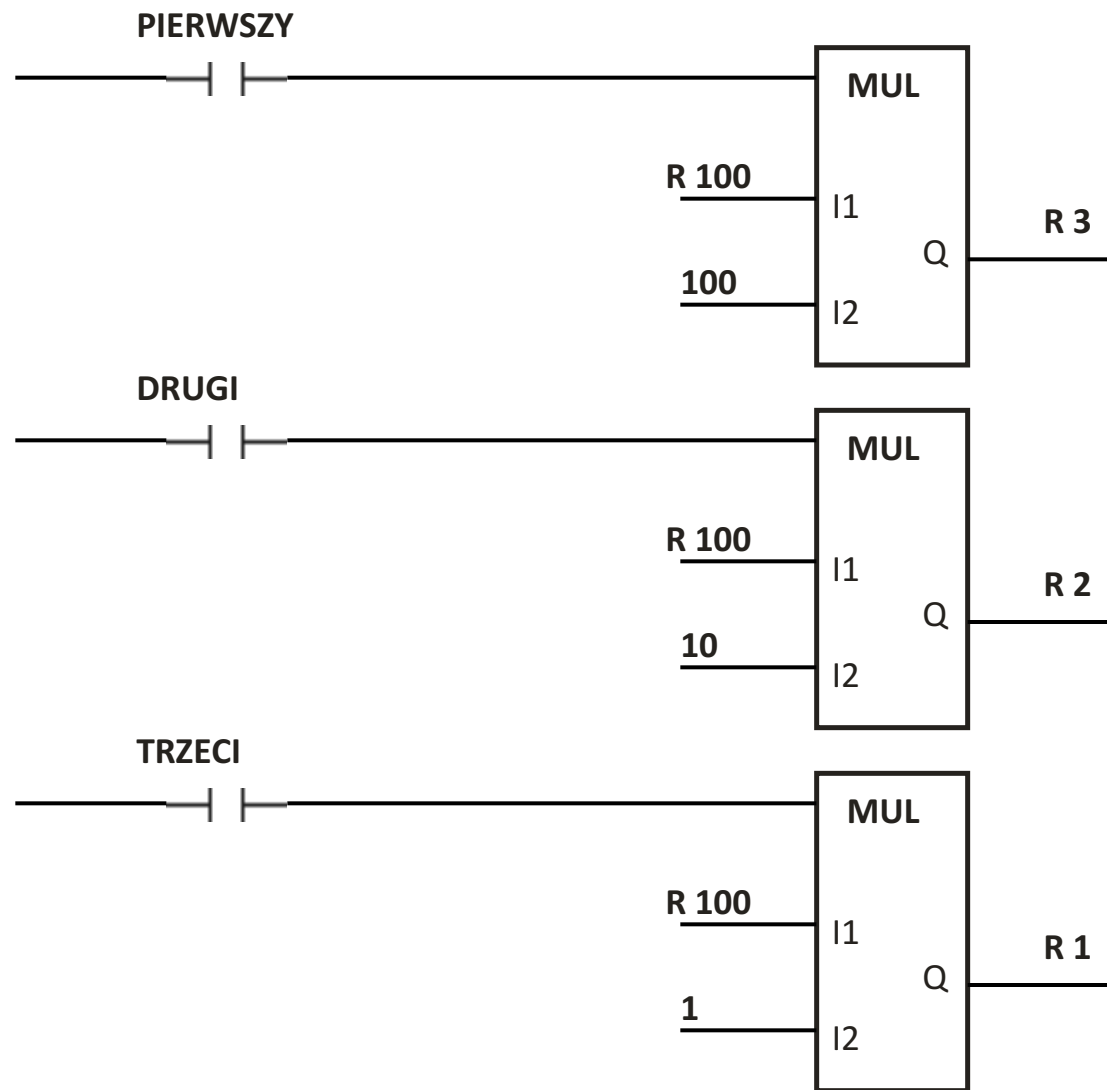


ODCZYTYWANIE KOLEJNOŚCI (1)

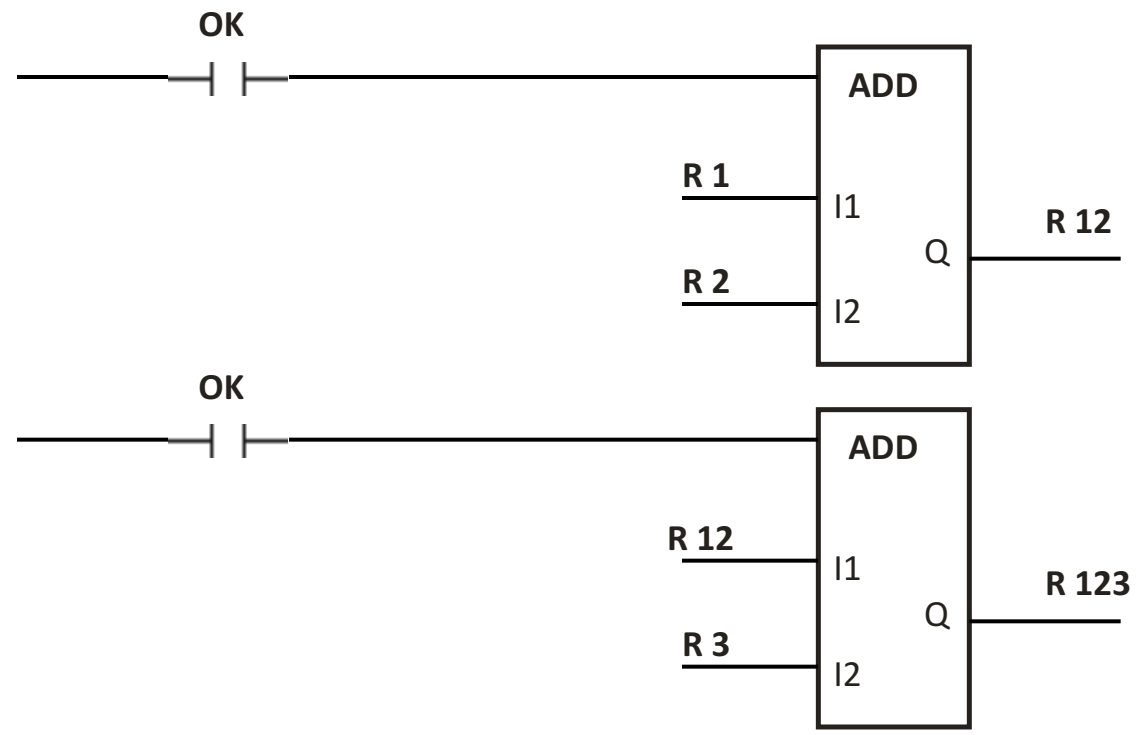


ODCZYTYWANIE KOLEJNOŚCI (2)

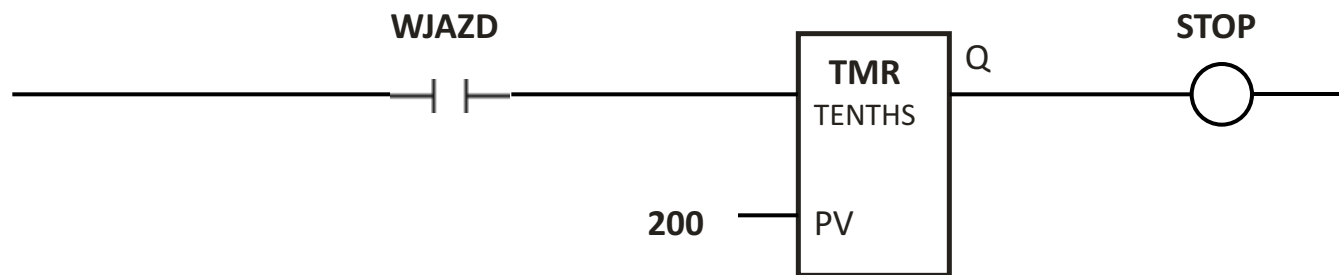
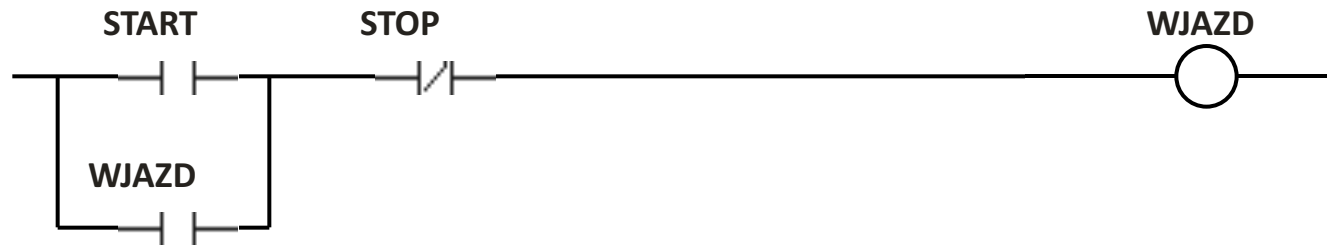
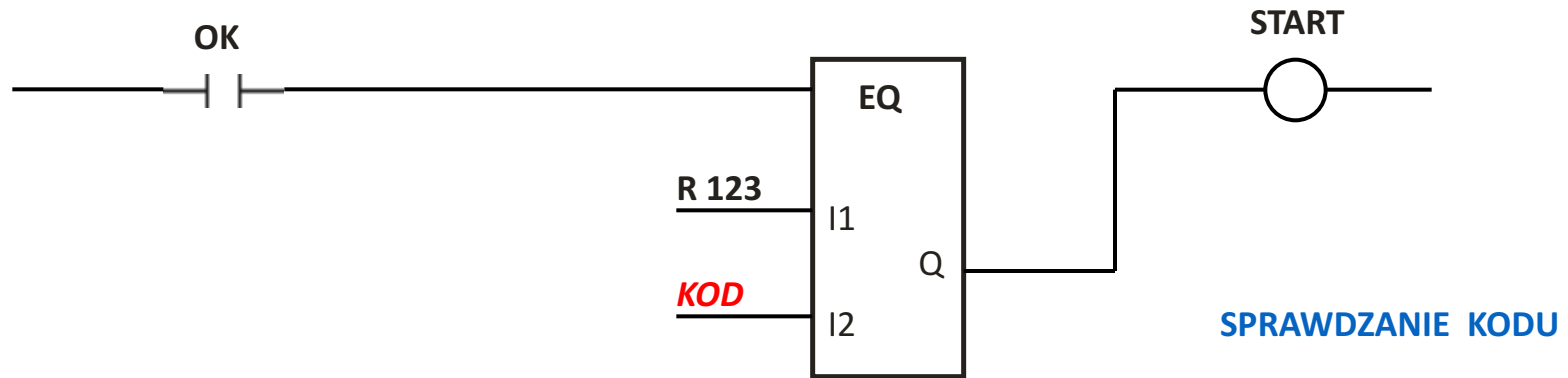




**OBLICZANIE KODU (1)**

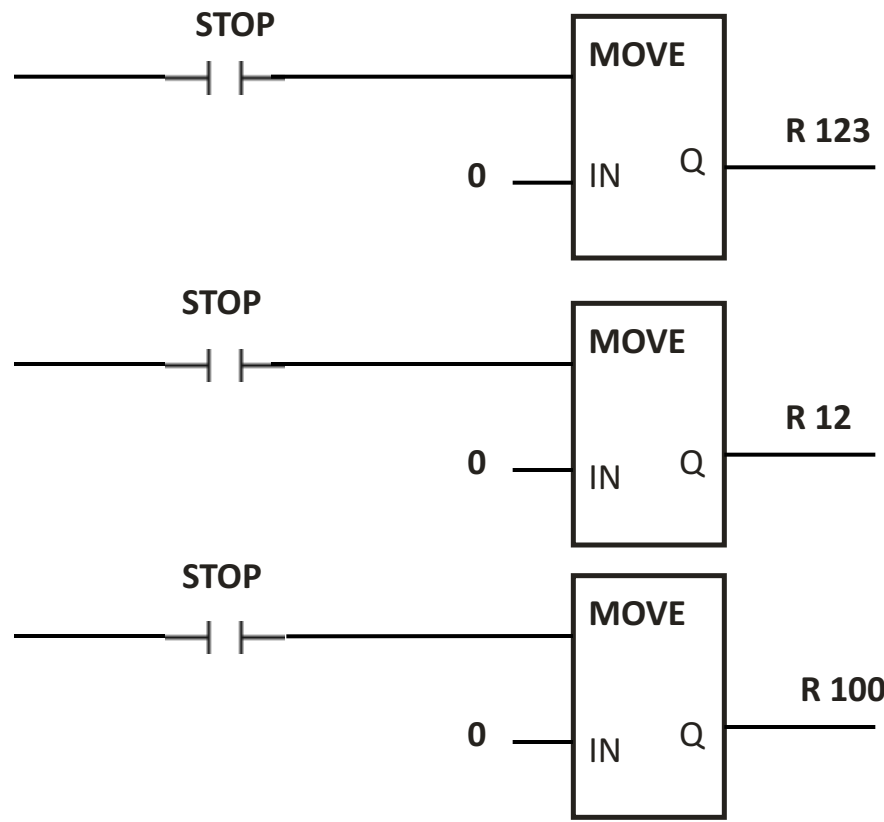


**OBLICZANIE KODU (2)**



OTWIERANIE BRAMY

**stop** zapala lampkę gotowości automatu do kolejnego wprowadzenia polecenia



RESETOWANIE PROGRAMU

# KODOWANIE BINARNE

(sterowanie komputerowe procesami ciągłymi)

## Jednostki binarne

W latach czterdziestych XX wieku opracowywano teoretyczne podstawy działania maszyn cyfrowych i zwrócono uwagę na system binarny.

System ten posiada dwie cyfry 0 i 1, które w prosty sposób można reprezentować w komputerze za pomocą odpowiednich napięć czy prądów elektrycznych.

Układy realizujące operacje na cyfrach binarnych są nieporównywalnie prostsze od analogicznych układów operujących na cyfrach dziesiętnych.

Dwóm symbolom **0** i **1** możemy nadać **dowolne** (pożądane) **znaczenie**.

**Przykład:**

W pomieszczeniu zainstalowano czujnik pożarowy. Jeśli w pomieszczeniu jest normalna temperatura, to czujnik nie przesyła sygnału. Możemy założyć (np.) stan 0 - brak pożaru. Jeśli jednak wykryty zostanie ogień, to czujnik prześle sygnał elektryczny elektryczny. Zinterpretujemy to jako stan 1 - pożar. Czujnik i urządzenie alarmujące komunikują się za pomocą informacji jednobitowej - ich język składa się tylko z dwóch symboli – 0 i 1.

0 - brak pożaru

1 - pożar

Słowo **bit** po raz pierwszy pojawiło się w literaturze informatycznej w roku 1948 w pracach teoretyka informatyki **Claude'a Shannona**



- > **bit - binary digit**
- > bit przyjmuje dwie wartości, które oznaczamy odpowiednio cyfrą 0 i 1.
- > znaczenie fizyczne tych dwóch wartości może być dowolne



0 / 1

brak pożaru / pożar

pożar / brak pożaru

zawór zamknięty / zawór otwarty

niskie napięcie / wysokie napięcie

silnik wyłączony / silnik włączony

przycisk zwolniony / przycisk wciśnięty

Trzy bity pozwalają utworzyć 8 różnych symboli (8-mio elementowych ciągów), a 4 bity 16 symboli itd.

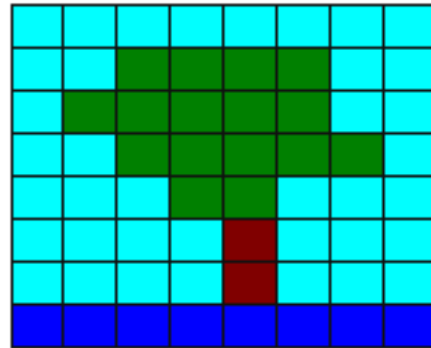
Lp.	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>
1.	0	0	0	0
2.	0	0	0	1
3.	0	0	1	0
4.	0	0	1	1
5.	0	1	0	0
6.	0	1	0	1
7.	0	1	1	0
8.	0	1	1	1
9.	1	0	0	0
10.	1	0	0	1
11.	1	0	1	0
12.	1	0	1	1
13.	1	1	0	0
14.	1	1	0	1
15.	1	1	1	0
16.	1	1	1	1

Liczba bitów	Liczba symboli	Potęga liczby 2
1	2	= 2 <sup>1</sup>
2	4	= 2 <sup>2</sup>
3	8	= 2 <sup>3</sup>
4	16	= 2 <sup>4</sup>
5	32	= 2 <sup>5</sup>
6	64	= 2 <sup>6</sup>
7	128	= 2 <sup>7</sup>
8	256	= 2 <sup>8</sup>
...	...	...
16	65536	= 2 <sup>16</sup>
...	...	...
32	4294967296	= 2 <sup>32</sup>
...	...	...
n	2 <sup>n</sup>	= 2 <sup>n</sup>





$n$  bitów tworzy  $2^n$  różnych symboli binarnych (różnych ciągów złożonych z zer i jedynek).

Dla dowolnej ilości informacji zawsze można dobrać słowo binarne o takiej liczbie bitów, aby utworzyć z nich pożądaną liczbę symboli. W ten sposób powstaje kod binarny.

zakodowanie binarne obrazka



Tablica kodowa kolorów, w której każdemu kolorowi punktu przyporządkujemy jeden symbol dwubitowy:

	- 00
	- 01
	- 10
	- 11



Kod binarny przeznaczony do kodowania małych liter alfabetu łacińskiego

5 bitów tworzy 32 symbole. Zatem 6 symboli nie zostanie wykorzystanych. Ważne jest, aby symboli nie było mniej niż liczba wiadomości do zakodowania.

Binarny kod znakowy							
Znak	Kod	Znak	Kod	Znak	Kod	Znak	Kod
<b>a</b>	00000	<b>h</b>	00111	<b>o</b>	01110	<b>v</b>	10101
<b>b</b>	00001	<b>i</b>	01000	<b>p</b>	01111	<b>w</b>	10110
<b>c</b>	00010	<b>j</b>	01001	<b>q</b>	10000	<b>x</b>	10111
<b>d</b>	00011	<b>k</b>	01010	<b>r</b>	10001	<b>y</b>	11000
<b>e</b>	00100	<b>l</b>	01011	<b>s</b>	10010	<b>z</b>	11001
<b>f</b>	00101	<b>m</b>	01100	<b>t</b>	10011		
<b>g</b>	00110	<b>n</b>	01101	<b>u</b>	10100		

**w**      **a**      **g**      **o**      **n**  
10110    00000    00110    01110    01101

1011000000001100111001101

## Odczytywanie

1001101110010101100001110

po rozdzieleniu na grupy pięciobitowe

10011 01110 01010 11000 01110

w tabeli kodu odczytujemy litery

10011 01110 01010 11000 01110  
**t o k y o**

Binarny kod znakowy							
Znak	Kod	Znak	Kod	Znak	Kod	Znak	Kod
<b>a</b>	00000	<b>h</b>	00111	<b>o</b>	01110	<b>v</b>	10101
<b>b</b>	00001	<b>i</b>	01000	<b>p</b>	01111	<b>w</b>	10110
<b>c</b>	00010	<b>j</b>	01001	<b>q</b>	10000	<b>x</b>	10111
<b>d</b>	00011	<b>k</b>	01010	<b>r</b>	10001	<b>y</b>	11000
<b>e</b>	00100	<b>l</b>	01011	<b>s</b>	10010	<b>z</b>	11001
<b>f</b>	00101	<b>m</b>	01100	<b>t</b>	10011		
<b>g</b>	00110	<b>n</b>	01101	<b>u</b>	10100		

Przyjmuje się pewne ustalone jednostki informacji.

Standardowe grupy bitów można w prosty sposób przechowywać w pamięciach komputerów, na nośnikach danych oraz przesyłać za pomocą sieci teleinformatycznych.

**Bajt** (ang. byte) jest taką właśnie standaryzacją.

Bajt jest grupą 8 bitów.

Oznaczamy go dużą literką **B** w odróżnieniu od bitu oznaczanego **b**.

1B pozwala rozróżnić 256 różnych stanów.

W fizyce i technice stosowane są wielokrotności jednostek podstawowych.  
Podstawą tych wielokrotności jest liczba 10:

**kilo** = 1000 =  $10^3$   
**mega** = 1000000 =  $10^6$  = kilo × 1000  
**giga** = 1000000000 =  $10^9$  = mega × 1000  
**tera** = 1000000000000 =  $10^{12}$  = giga × 1000

W systemie binarnym, ze względu na podobieństwo, zastosowano również podobne mnożniki, jednakże podstawą ich jest liczba 2, nie 10:

**Kilo** = 1024 =  $2^{10}$   
**Mega** = 1048576 =  $2^{20}$  = Kilo × 1024  
**Giga** = 1073741824 =  $2^{30}$  = Mega × 1024  
**Tera** = 1099511627776 =  $2^{40}$  = Giga × 1024

Jednostki binarne			
bitowe		bajtowe	
<b>b</b>	bit	<b>B</b>	bajt
<b>Kb</b>	kilobit	<b>KB</b>	kilobajt
<b>Mb</b>	megabit	<b>MB</b>	megabajt
<b>Gb</b>	gigabit	<b>GB</b>	gigabajt
<b>Tb</b>	terabit	<b>TB</b>	terabajt



## KODOWANIE WARTOŚCI LCZBOWYCH

Naturalny system dwójkowy (ang. NBS – Natural Binary System) jest najprostszym systemem pozycyjnym, w którym podstawa  $p = 2$ .

$$101011_{(2)} = 2^5 + 2^3 + 2^1 + 2^0 = 32 + 8 + 2 + 1 = 43_{(10)}$$

Do wykonywania dodawania niezbędna jest znajomość tabliczki dodawania, czyli wyników sumowania każdej cyfry z każdą inną. W systemie binarnym mamy tylko dwie cyfry 0 i 1, zatem tabliczka dodawania jest niezwykle prosta i składa się tylko z 4 pozycji:

$$\begin{array}{rcl} 0 + 0 & = & 0 \\ 0 + 1 & = & 1 \\ 1 + 0 & = & 1 \\ 1 + 1 & = & \mathbf{1}0 \end{array}$$

$$\begin{array}{r} 1111001 \\ + 10010 \\ \hline \end{array}$$

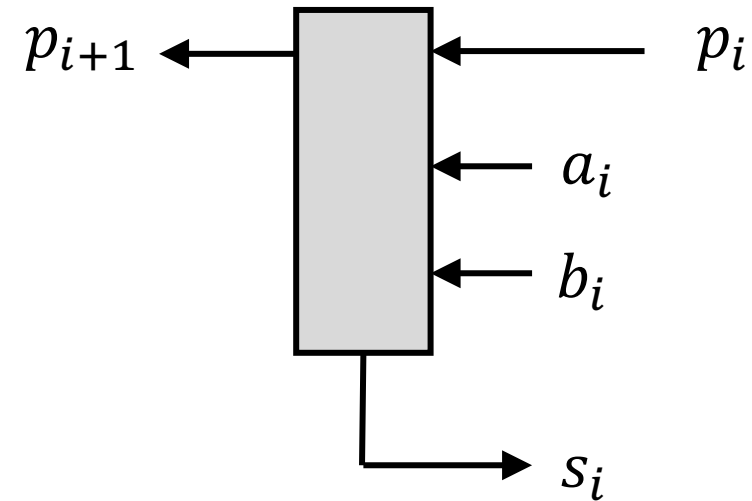
$$\begin{array}{r} \mathbf{111} \\ 01111001 \\ + 00010010 \\ \hline 10001011 \end{array}$$

$$1111001_{(2)} + 10010_{(2)} = 10001011_{(2)} \quad (121 + 18 = 139)$$

$p_{i+1}$	$a_i$	$b_i$	$p_i$	$s_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
1	1	1	0	0
0	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

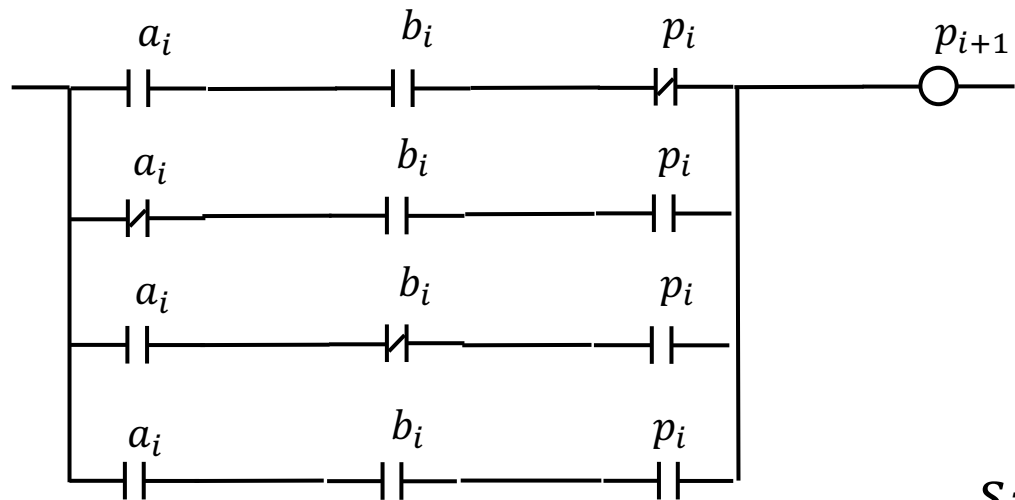
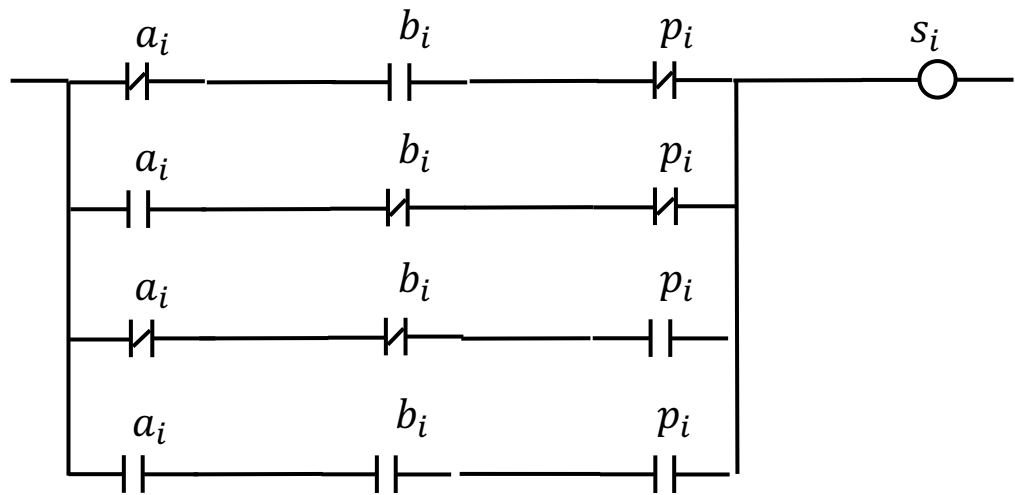
PRZYKŁAD 33:

SUMATOR 1. BITOWY



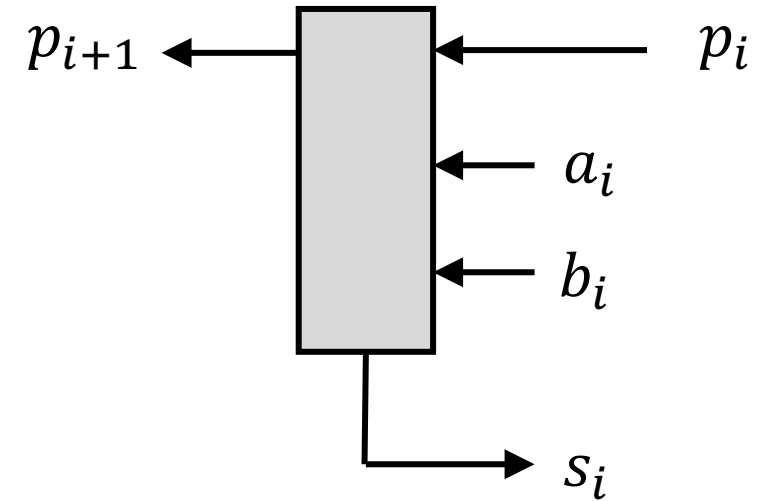
$$s_i = \bar{a}_i \cdot b_i \cdot \bar{p}_i + a_i \cdot \bar{b}_i \cdot \bar{p}_i + \bar{a}_i \cdot \bar{b}_i \cdot p_i + a_i \cdot b_i \cdot p_i$$

$$p_{i+1} = a_i \cdot b_i \cdot \bar{p}_i + \bar{a}_i \cdot b_i \cdot p_i + a_i \cdot \bar{b}_i \cdot p_i + a_i \cdot b_i \cdot p_i$$



PRZYKŁAD 33:

**SUMATOR 1. BITOWY**

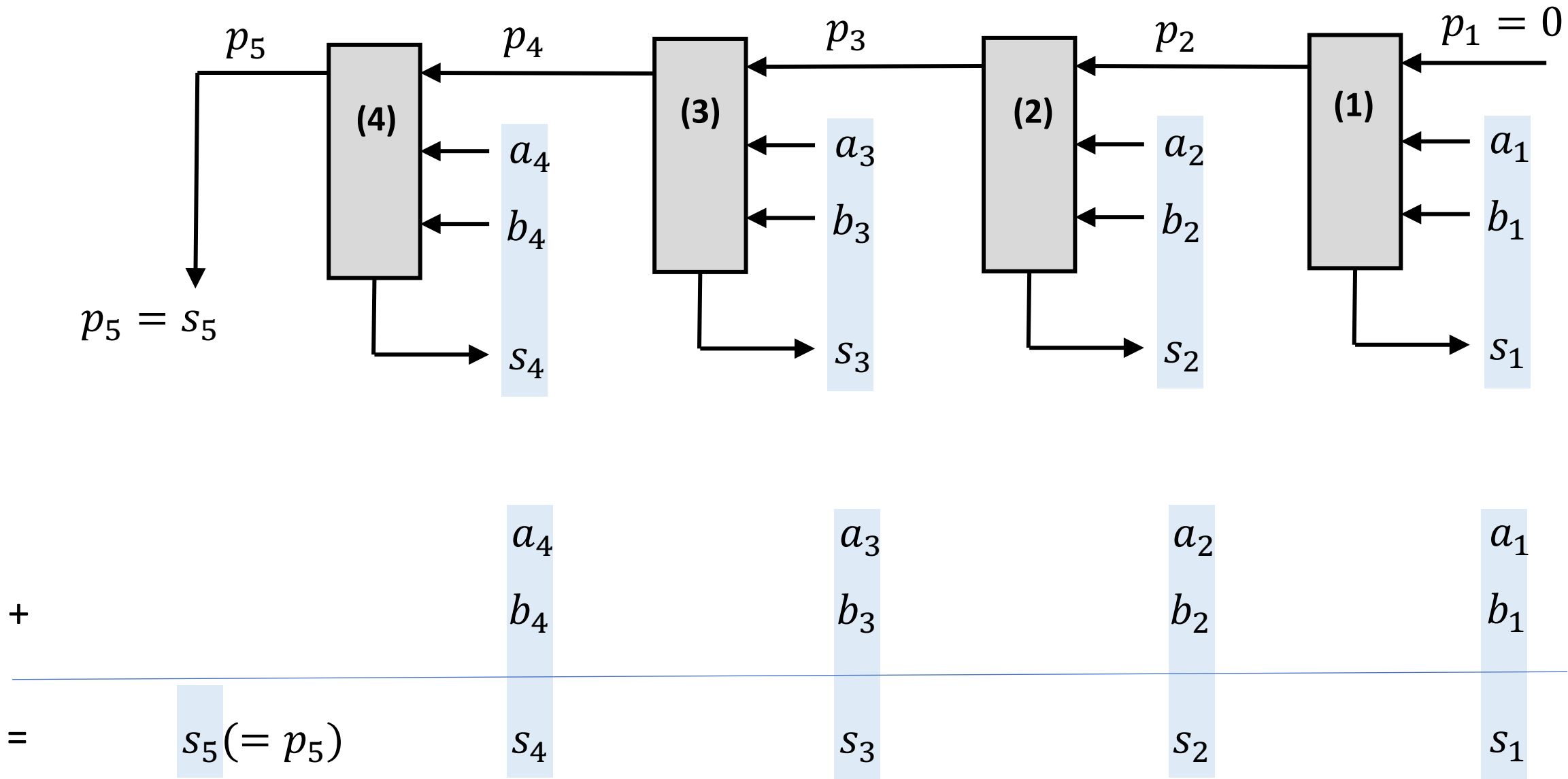


$$s_i = \bar{a}_i \cdot b_i \cdot \bar{p}_i + a_i \cdot \bar{b}_i \cdot \bar{p}_i + \bar{a}_i \cdot \bar{b}_i \cdot p_i + a_i \cdot b_i \cdot p_i$$

$$p_{i+1} = a_i \cdot b_i \cdot \bar{p}_i + \bar{a}_i \cdot b_i \cdot p_i + a_i \cdot \bar{b}_i \cdot p_i + a_i \cdot b_i \cdot p_i$$

PRZYKŁAD 34:

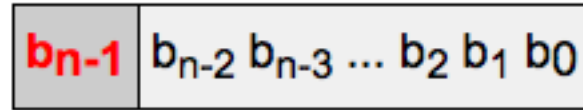
SUMATOR 4. BITOWY



W analogiczny sposób można sporządzić algorytmy odejmowania, mnożenia i dzielenia binarnego. Dzielenie binarne jest najbardziej skomplikowaną operacją arytmetyczną

## Zapisywanie liczb ze znakiem

System Z M (znak- moduł)



$b_{n-1}$  - bit znaku liczby

$b_{n-2} \dots b_0$  - bity modułu liczby

Dla liczb dodatnich i zera bit znaku ma wartość 0, dla liczb ujemnych i zera ma wartość 1.

Moduł liczby **ZM** jest zapisany w naturalnym kodzie dwójkowym NBC

$$b_{n-1}b_{n-2}\dots b_2b_1b_0 = (-1)^{b_{n-1}} \times (b_{n-2}2^{n-2} + \dots + b_22^2 + b_12^1 + b_02^0)$$

gdzie

b - bit, cyfra dwójkowa 0 lub 1

n - liczba bitów w zapisie liczby

System **Z M** (znak- moduł) jest koncepcyjnie prosty ale stwarza poważne problemy przy wykonywaniu operacji arytmetycznych.

Ponadto bit znakowy posiada zupełnie inne znaczenie niż znaczenie pozostałych bitów.

Wygodnym podejściem do rozwiązania problemu liczb ze znakiem jest system uzupełnień do 2 zwany systemem **U2**



Wartości wag pozycji w zapisie U2							
waga	$-2^{n-1}$	$2^{n-2}$	$2^{n-3}$	...	$2^2$	$2^1$	$2^0$
cyfra	$b_{n-1}$	$b_{n-2}$	$b_{n-3}$	...	$b_2$	$b_1$	$b_0$

$$b_{n-1}b_{n-2}b_{n-3}\dots b_2b_1b_0_{(U2)} = b_{n-1}(-2^{n-1}) + b_{n-2}2^{n-2} + b_{n-3}2^{n-3} + \dots + b_22^2 + b_12^1 + b_02^0$$

procedura wyznaczania liczby przeciwnej w kodzie U2

- dokonać negacji wszystkich bitów zapisu liczby U2.
- do wyniku dodać 1

postać ujemna liczby **U2** nie jest tak czytelna jak w przypadku kodów **ZM**

### Przykład:

Wyznaczyć liczbę przeciwną w kodzie U2 do danej liczby  $01101110_{(U2)}$ .

1. Dokonujemy negacji (zmianę na wartości przeciwne) wszystkich bitów liczby U2:

$$\begin{array}{r} \text{NOT } 01101110 \\ \hline 10010001 \end{array}$$

2. Do wyniku negacji dodajemy 1:

$$\begin{array}{r} 10010001 \\ + 00000001 \\ \hline 10010010 \end{array}$$

Liczbą przeciwną do  $01101110_{(U2)}$  jest  $10010010_{(U2)}$ .

Aby się o tym przekonać, obliczmy wartości dziesiętne obu liczb:

$$01101110_{(U2)} = 64 + 32 + 8 + 4 + 2 = 110_{(10)}.$$

$$10010010_{(U2)} = (-128) + 16 + 2 = (-128) + 18 = (-110)_{(10)}.$$

$$b_{n-1}(-2^{n-1}) + b_{n-2}2^{n-2} + b_{n-3}2^{n-3} + \dots + b_22^2 + b_12^1 + b_02^0$$

## 4-bitowe liczby w zapisie U2

4 bitowy system U2		
Kod U2	Przeliczenie	Wartość
0000	0	0
0001	$2^0$	1
0010	$2^1$	2
0011	$2^1 + 2^0$	3
0100	$2^2$	4
0101	$2^2 + 2^0$	5
0110	$2^2 + 2^1$	6
0111	$2^2 + 2^1 + 2^0$	7
1000	$(-2^3)$	(-8)
1001	$(-2^3) + 2^0$	(-7)
1010	$(-2^3) + 2^1$	(-6)
1011	$(-2^3) + 2^1 + 2^0$	(-5)
1100	$(-2^3) + 2^2$	(-4)
1101	$(-2^3) + 2^2 + 2^0$	(-3)
1110	$(-2^3) + 2^2 + 2^1$	(-2)
1111	$(-2^3) + 2^2 + 2^1 + 2^0$	(-1)

W kodzie U2 liczb ujemnych jest o 1 więcej niż dodatnich

np. dla 16 bitów mamy liczby ze znakiem  
od - 32768 do + 32767

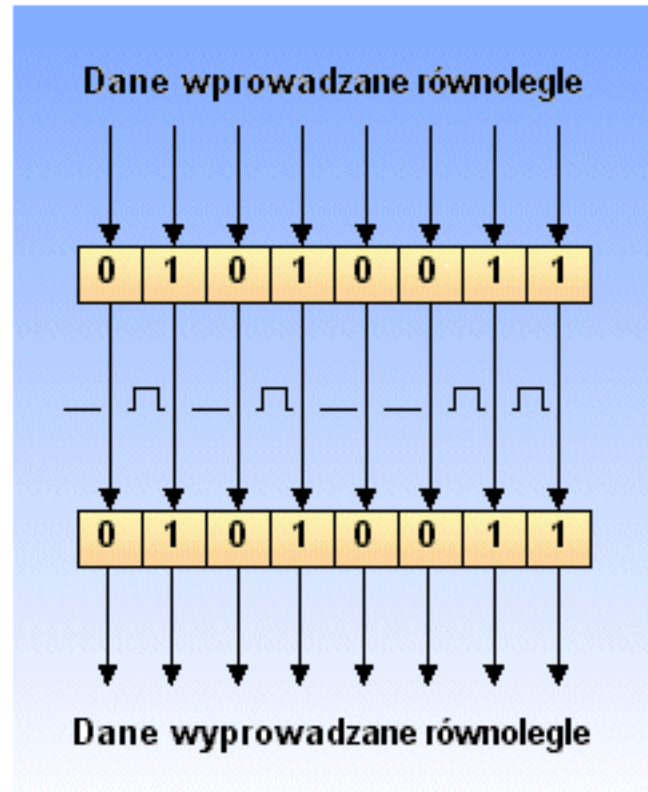
Wartość słowa kodowego w różnych systemach		
NBC	$b_{n-1} \dots b_1 b_0$	$b_{n-1}2^{n-1} + \dots + b_12^1 + b_02^0$
ZM	$b_z b_{n-2} \dots b_1 b_0$	$(-1)^{b_z} \times (\text{kod NBC pozostałych bitów})$
U2	$b_{n-1} b_{n-2} \dots b_1 b_0$	$b_{n-1}(-2^{n-1}) + \text{kod NBC pozostałych bitów}$

Arytmetyka w poznanych kodach dwójkowych	
ZM	Reguły arytmetyki NBC muszą być modyfikowane w skomplikowany sposób.
U2	Reguły arytmetyki NBC zachowane bez zmian

postać ujemna liczby **U2** nie jest tak czytelna jak w przypadku kodów **ZM**

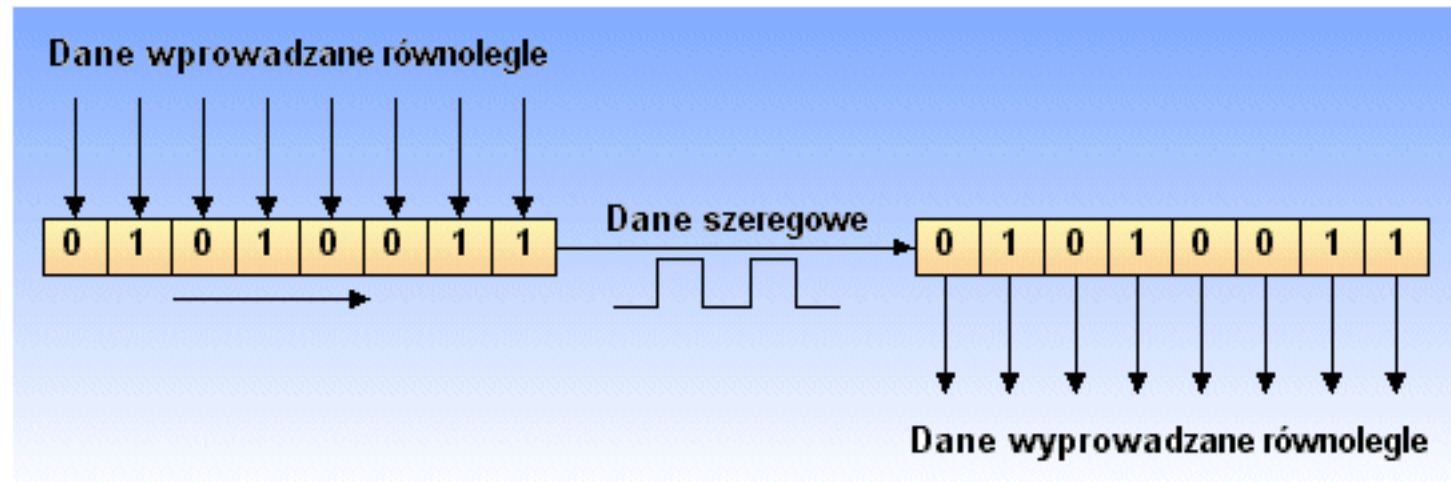
## Transmisja sygnału.

**Transmisja równoległa** polega na przesyłaniu wszystkich pól słowa danych jednocześnie



Dzięki zrównolegleniu przesyłu danych, uzyskuje się znaczną prędkości przesyłu. Wiąże się to jednak z koniecznością stosowania łącza wieloprzewodowego.

Transmisja szeregowa polega na sekwencyjnym przesyłaniu danych bit po bicie.



Zaletą tego rodzaju transmisji jest możliwość przesyłu danych na dużo większą odległość. Przy tym konieczne są tylko dwa przewody do transmisji (jednostronnej) sygnału. Wadą tej metody jest mniejsza prędkość przesyłu.

Złącze RS-232 służy do szeregowej transmisji danych.

**Kontrola parzystości** – metoda wykrywania przekłamań w transmitowanych wiadomościach. Polega na dodawaniu do wysyłanej wiadomości bitu kontrolnego (bitu parzystości).

Bitem parzystości nazywa się bit kontrolny, który przyjmuje wartość 1, gdy liczba jedynek w przesyłanej wiadomości jest nieparzysta, lub 0, gdy parzysta.

•Wiadomość **10111101**<sub>2</sub> ma parzystą liczbę jedynek, więc bit parzystości wynosi 0.

Wiadomość z dołączonym bitem parzystości to

• **101111010**<sub>2</sub>.

•Wiadomość **01110011**<sub>2</sub> ma nieparzystą liczbę jedynek, więc bit parzystości wynosi 1.

Wiadomość z dołączonym bitem parzystości to

• **011100111**<sub>2</sub>.

Gdy bit parzystości nie odpowiada parzystości bajtu:  
**wykryto błąd.**

Jeśli jednak dwa bity (lub parzysta liczba bitów) zmieniłyby się jednocześnie:  
**nie wykryto błędu.**

System kontroli parzystości może wykrywać wyłącznie nieparzystą liczbę błędów, może wykryć jedynie 50% błędów.

Ten mechanizm wykrywania błędów ma jeszcze jedną wadę: nie jest w stanie skorygować znalezionych błędów.



$$f(t) = \frac{dy(t)}{dt} \approx \frac{\Delta y(t)}{\Delta t}$$

$$f_k = \frac{y_k - y_{k-1}}{T}$$

$$y_k = T f_k + y_{k-1}$$

$$y(t) = \int_0^t f(t) dt$$

$$y_1 = T f_1 + y_0$$

$$y_2 = T f_2 + y_1$$

$$y_3 = T f_3 + y_2$$

$$y_4 = T f_4 + y_3$$

$$y_5 = T f_5 + y_4$$

$$\vdots$$

$$y_k = T f_k + y_{k-1}$$

$$y_2 = T f_2 + T f_1 + y_0$$

$$y_3 = T f_3 + T f_2 + T f_1 + y_0$$

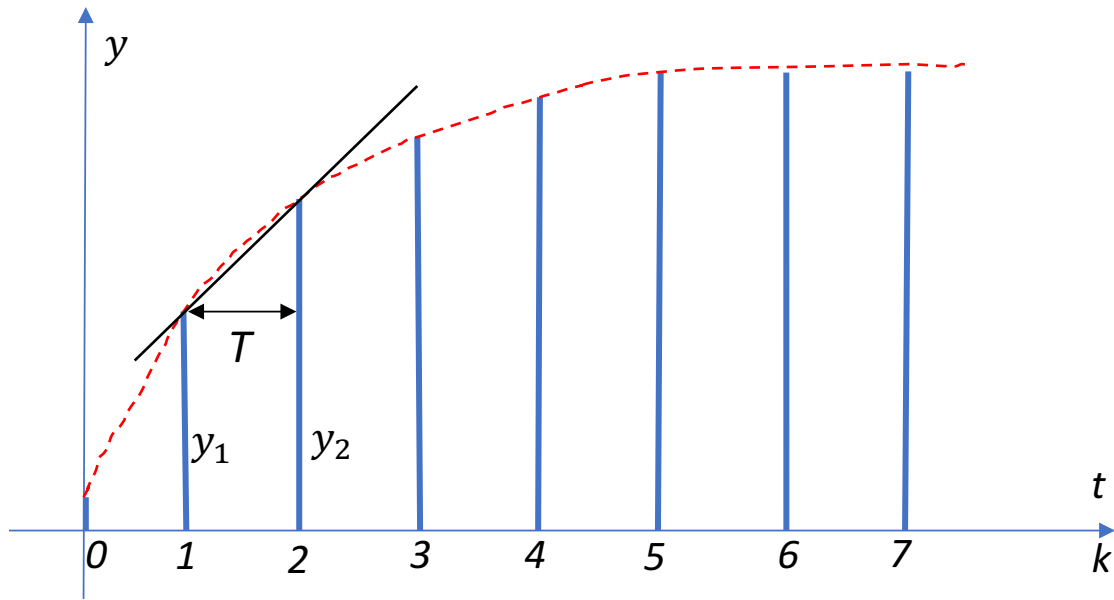
$$y_4 = T f_4 + T f_3 + T f_2 + T f_1 + y_0$$

$$y_5 = T f_5 + T f_4 + T f_3 + T f_2 + T f_1 + y_0$$

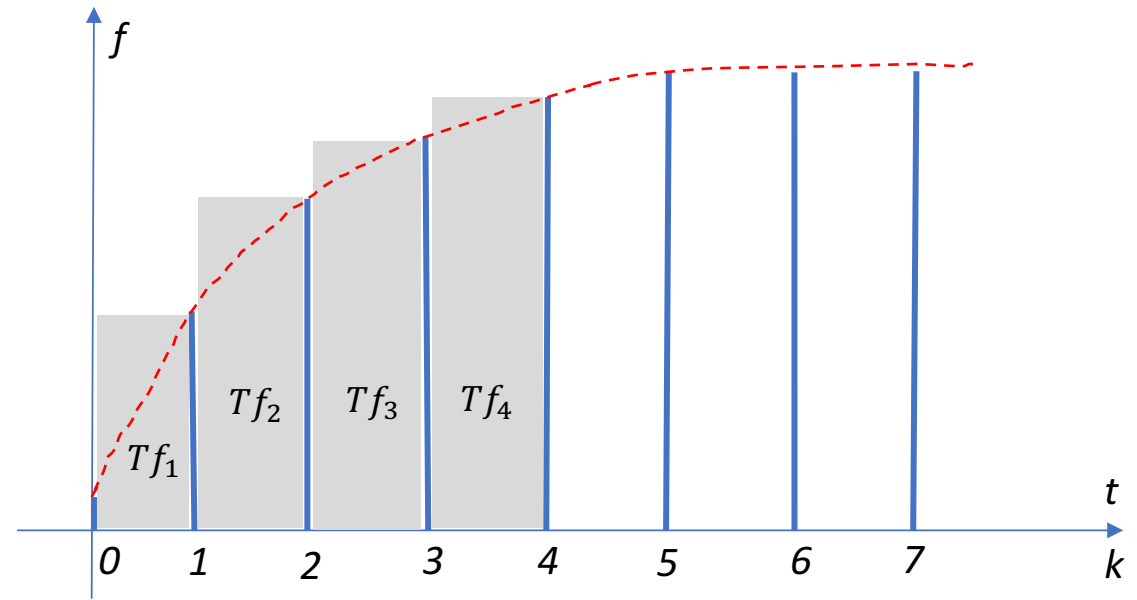
$$\vdots$$

$$y_k = \sum_{i=1}^k T f_i + y_0$$

$$y(t) = \int_0^t f(t) dt$$



$$f_k = \frac{y_k - y_{k-1}}{T}$$



$$y_k = \sum_{i=1}^k T f_i + y_0$$

$$y_k = T f_k + y_{k-1}$$

$$f(t) = \frac{dy(t)}{dt} \approx \frac{\Delta y(t)}{\Delta t}$$

$$f_k = \frac{y_{k+1} - y_k}{T}$$

$$y_{k+1} = T f_k + y_k$$

$$y_k = T f_{k-1} + y_{k-1}$$

$$y(t) = \int_0^t f(t) dt$$

$$y_1 = T f_0 + y_0$$

$$y_2 = T f_1 + y_1$$

$$y_3 = T f_2 + y_2$$

$$y_4 = T f_3 + y_3$$

$$y_5 = T f_4 + y_4$$

⋮

$$y_k = T f_k + y_{k-1}$$

$$y_2 = T f_1 + T f_0 + y_0$$

$$y_3 = T f_2 + T f_1 + T f_0 + y_0$$

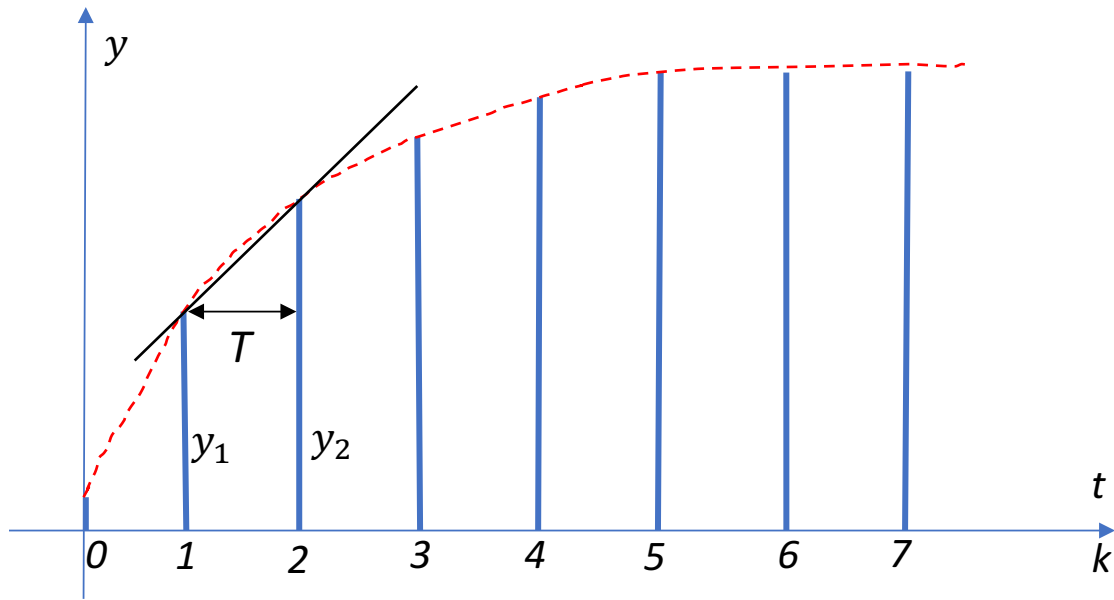
$$y_4 = T f_3 + T f_2 + T f_1 + T f_0 + y_0$$

$$y_5 = T f_4 + T f_3 + T f_2 + T f_1 + T f_0 + y_0$$

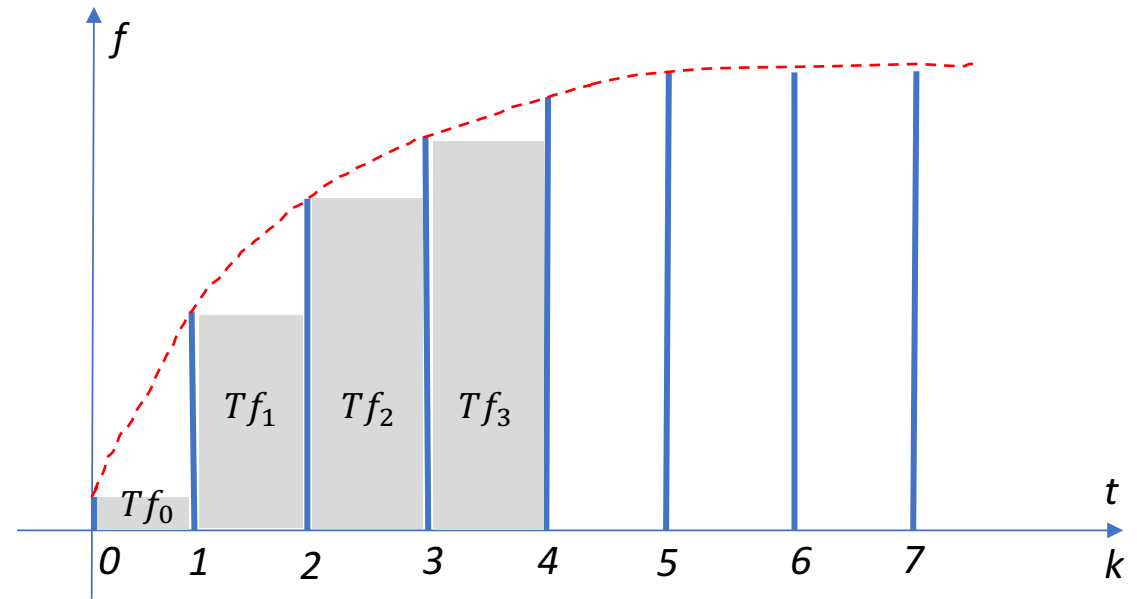
⋮

$$y_k = \sum_{i=0}^{k-1} T f_i + y_0$$

$$y(t) = \int_0^t f(t) dt$$

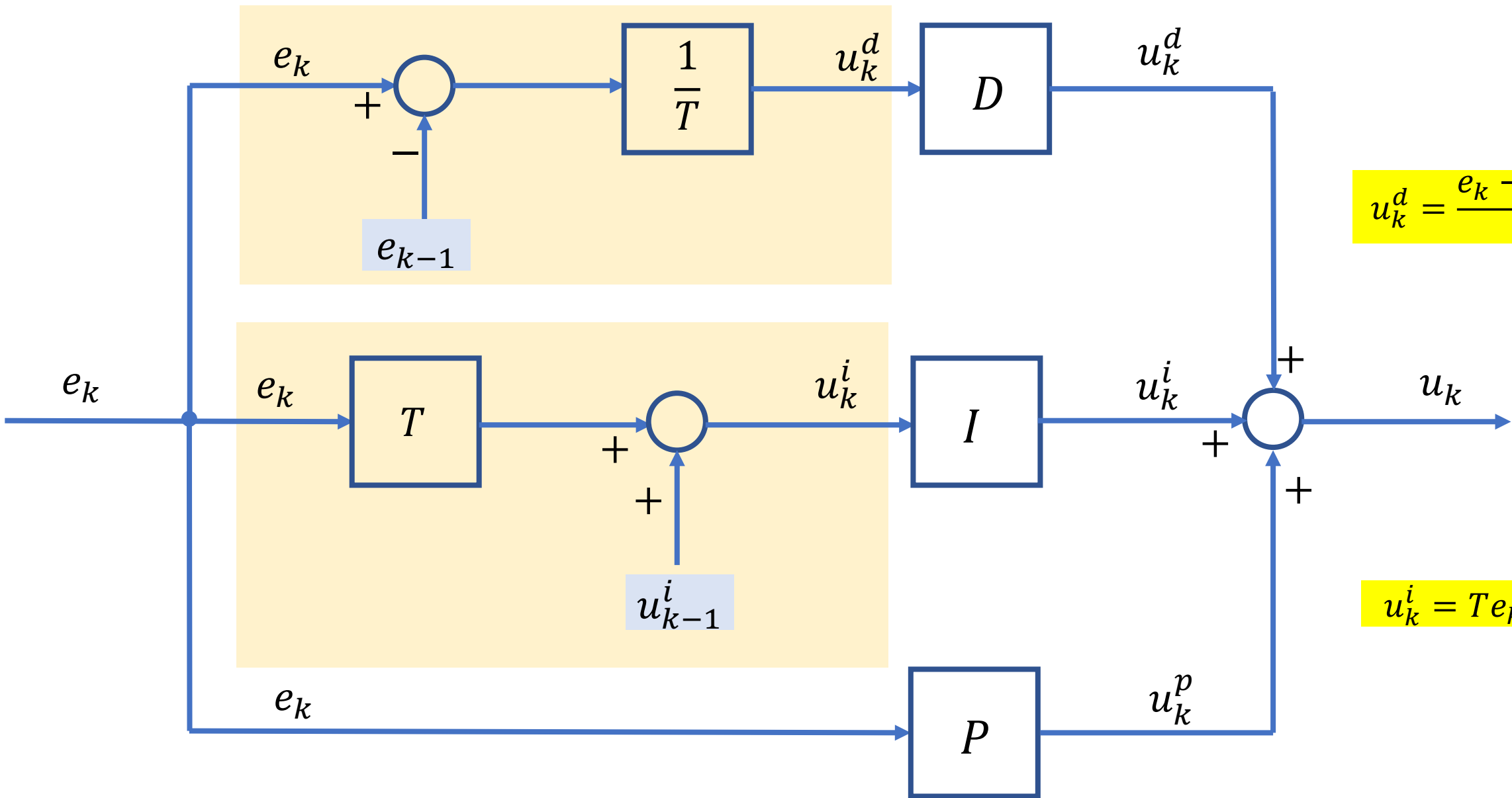


$$f_k = \frac{y_{k+1} - y_k}{T}$$



$$y_k = \sum_{i=0}^{k-1} Tf_i + y_0$$

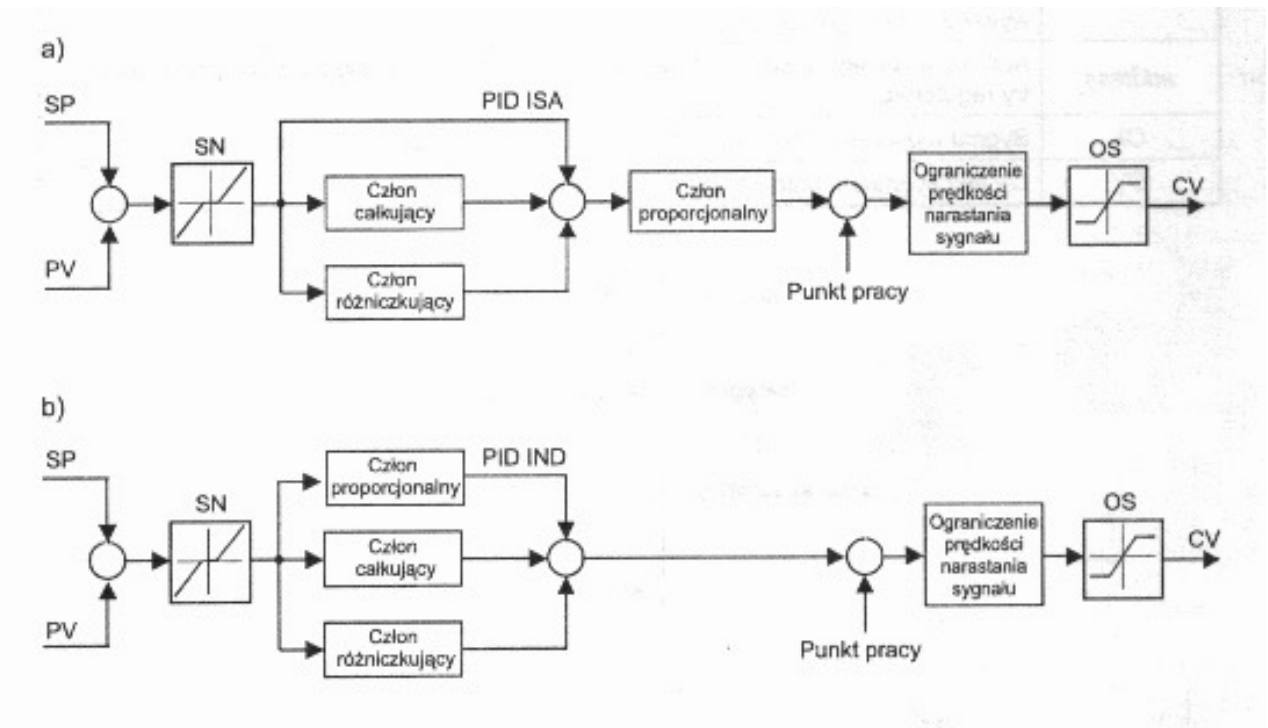
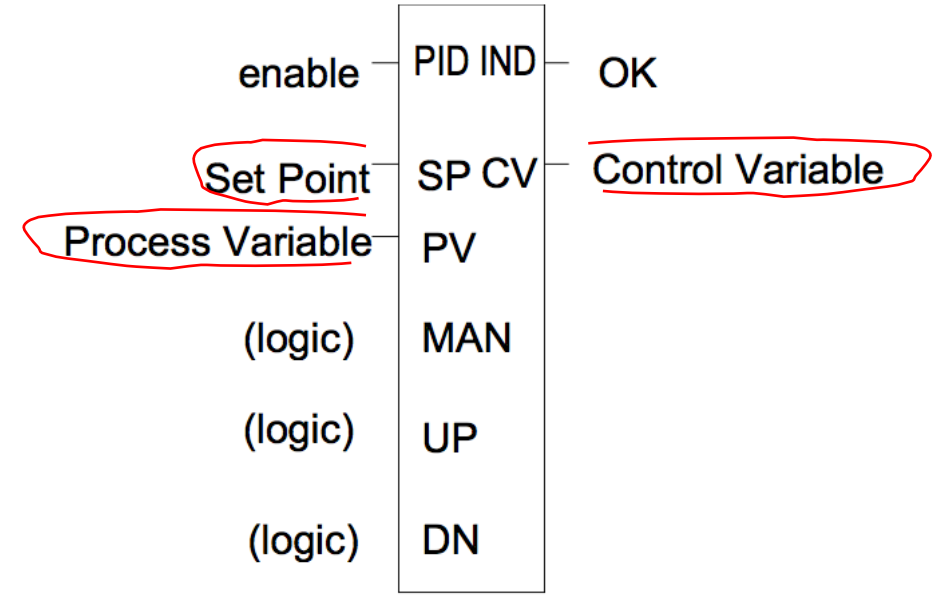
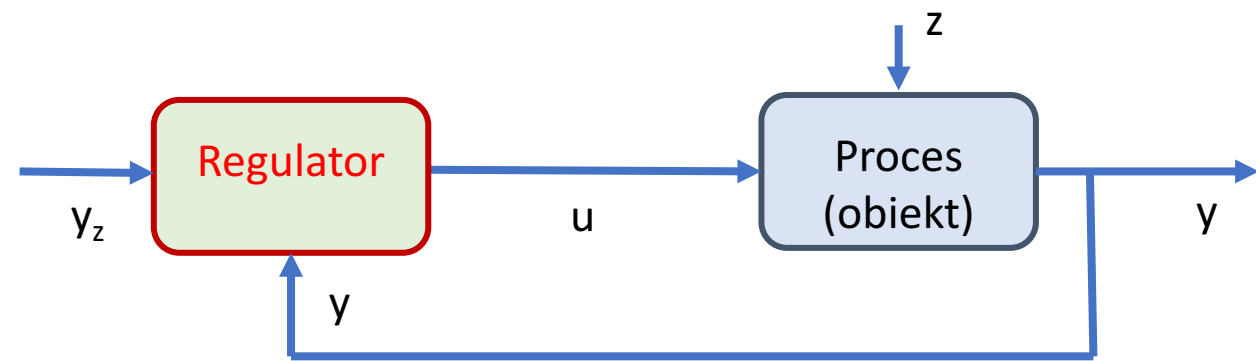
$$y_k = Tf_{k-1} + y_{k-1}$$



$$u_k^d = \frac{e_k - e_{k-1}}{T}$$

$$u_k^i = T e_k + u_{k-1}^i$$

**Blok PID w PLC**



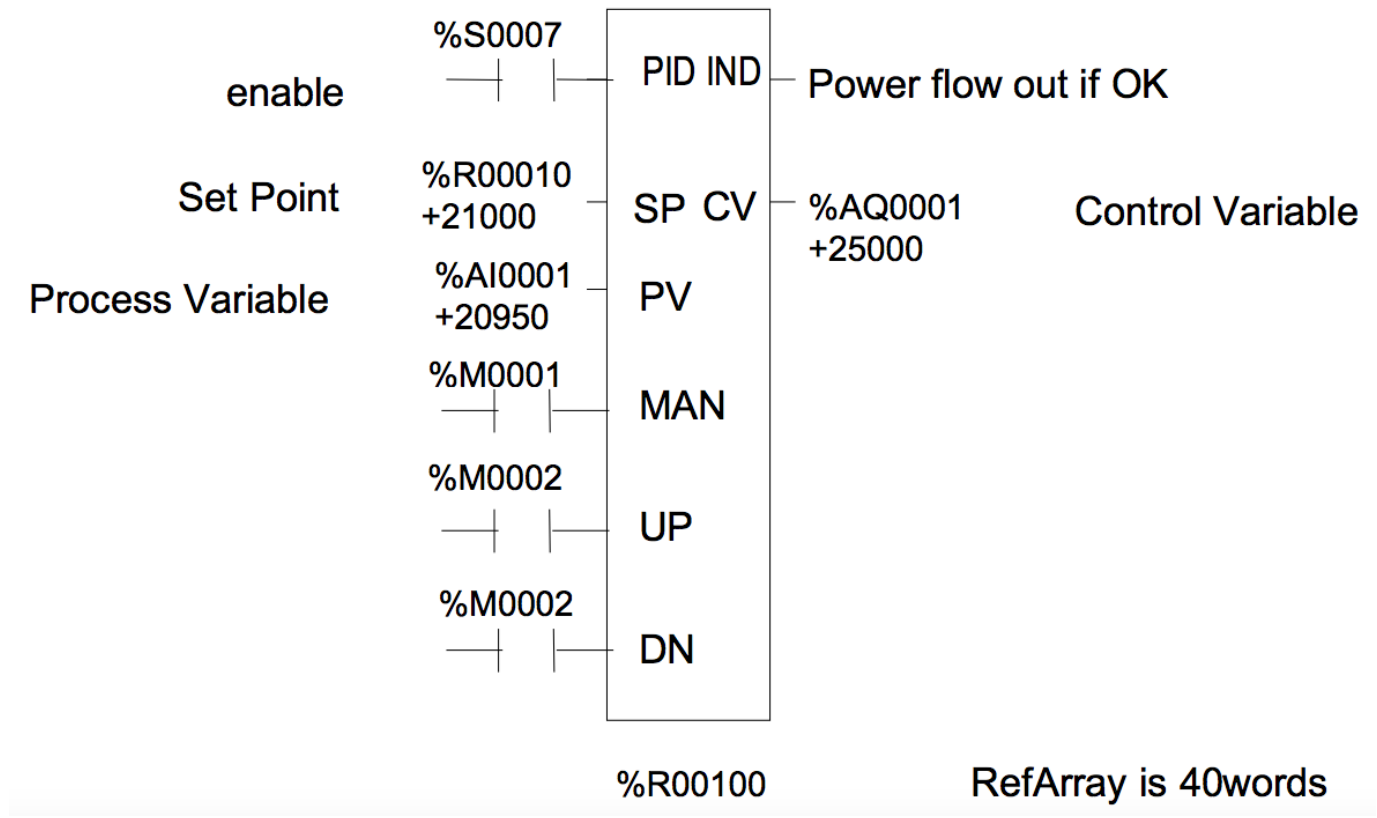
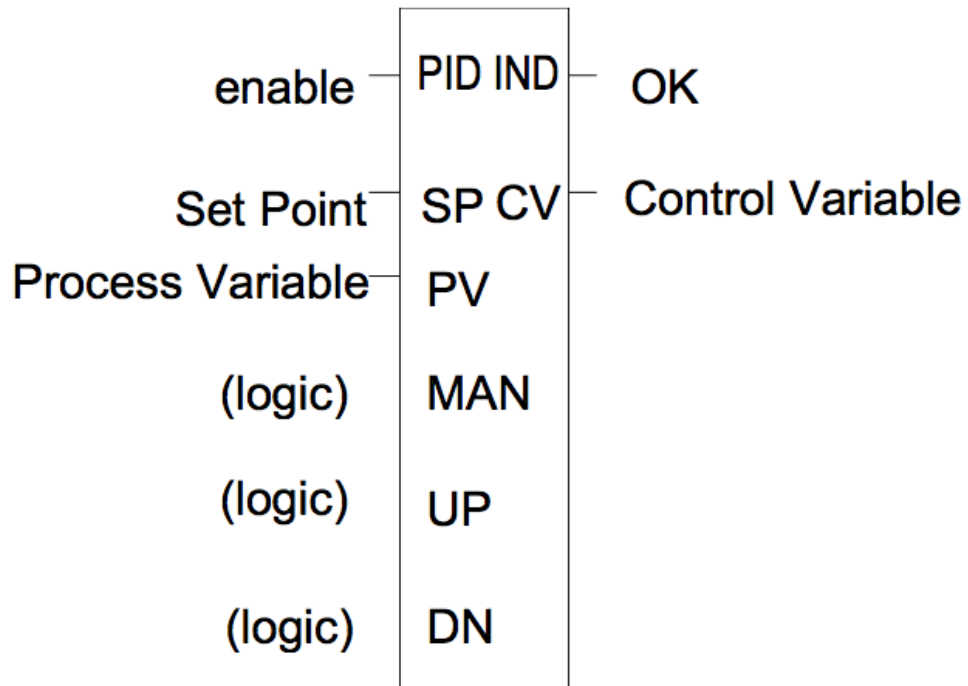
Schemat blokowy przemysłowego regulatora PID

SP = **y<sub>z</sub>** – wartość zadana sygnału y

CV = **u** - sygnał nastawiający

PV = **y** - sygnał regulowany

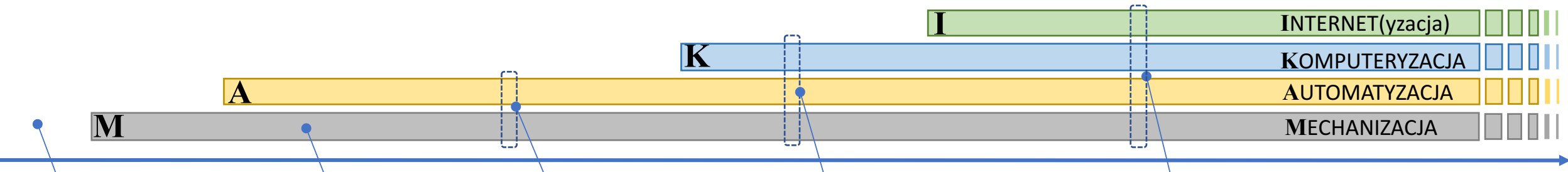
**Blok PID w PLC**

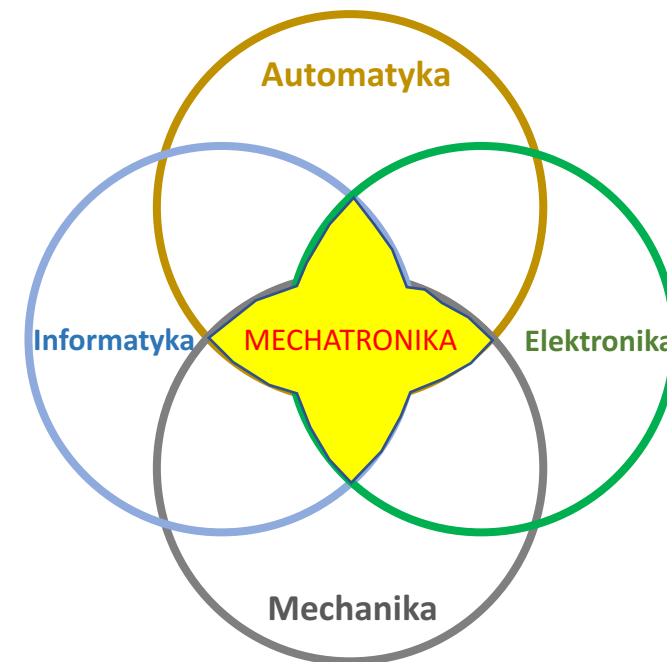
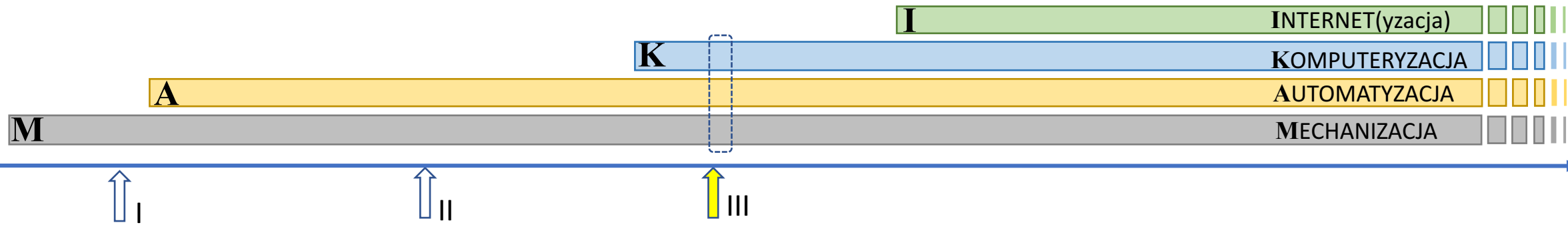


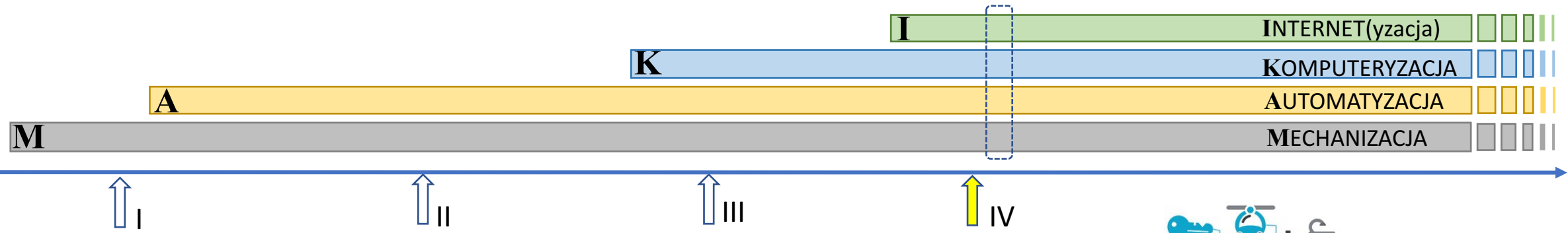
PRZYKŁAD 35

MECHATRONIKA  
SYSTEMY HMI / SCADA









## Rozwiązania Przemysłu 4.0:

**Internet Rzeczy - połączenie urządzeń z internetem**

**Systemy Cyber-Fizyczne - integracja warstwy obliczeń i procesów fizycznych**

**Systemy Cyber-Biologiczne - spięcie człowieka i maszyny**

**Maszyny współpracujące**

**Systemy społecznościowe maszyn**



**M**

**A**

**K**

**I**

INTERNET(yzacja)



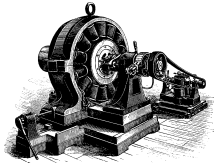
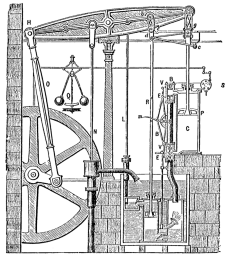
KOMPUTERYZACJA



AUTOMATYZACJA



MECHANIZACJA



MECHANIKA  
TERMODYNAMIKA  
ELEKTROTECHNIKA

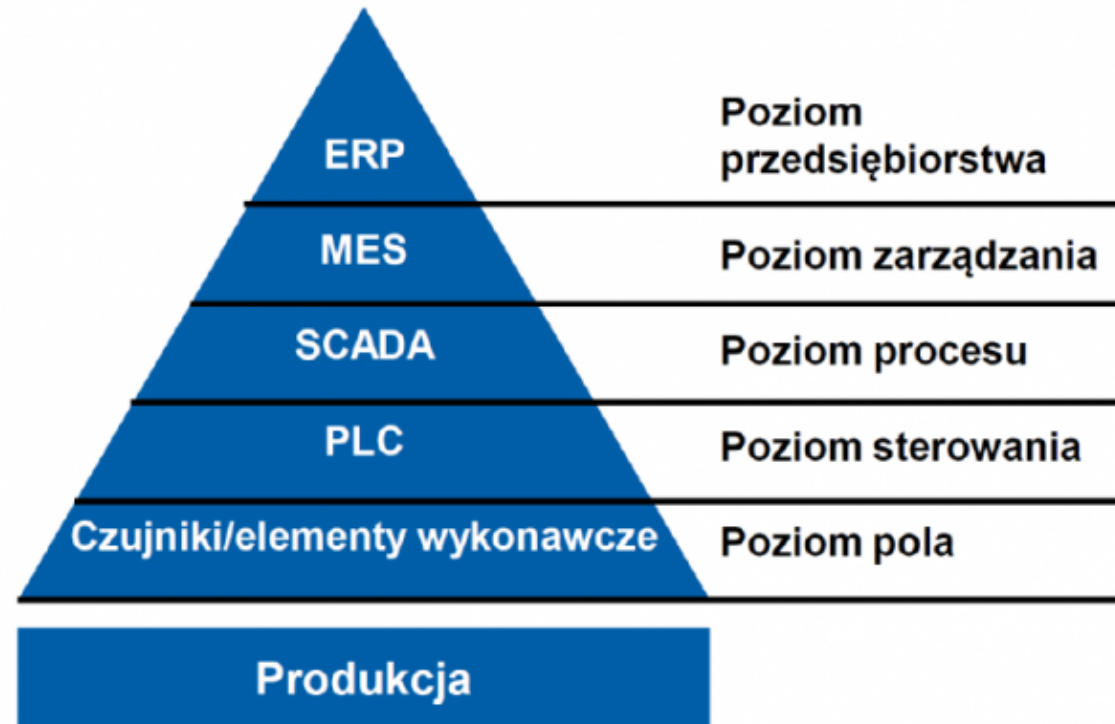
TEORIA INFORMACJI  
INFORMATYKA  
FIZYKA KWANTOWA

ELEKTRONIKA  
ROBOTYKA  
MECHATRONIKA

TEORIA STEROWANIA  
CYBERNETYKA  
TEORIA SYSTEMÓW

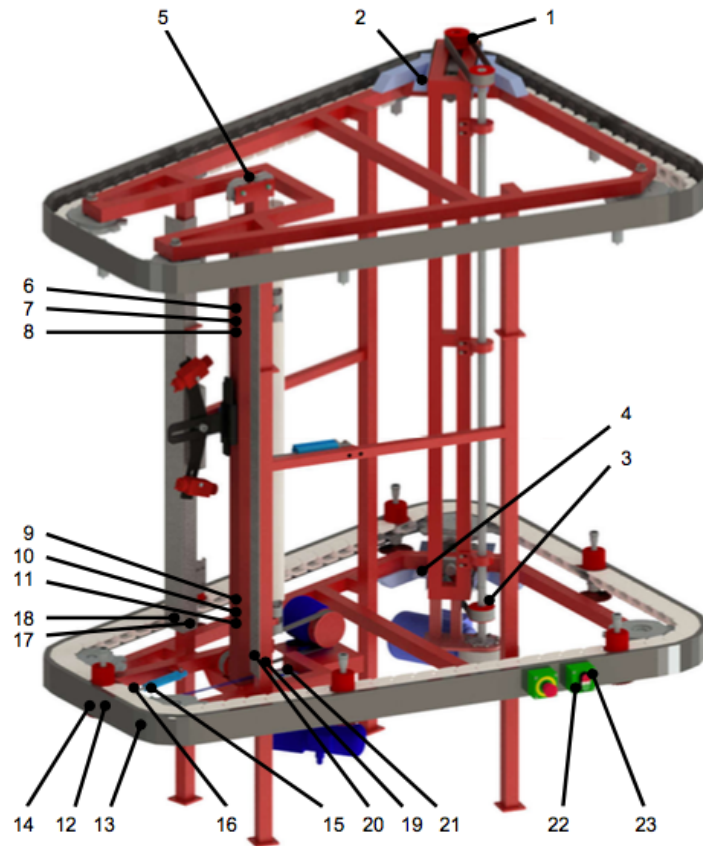
MATEMATYKA

„PIRAMIDA” ZARZĄDZANIA PRODUKCJĄ:

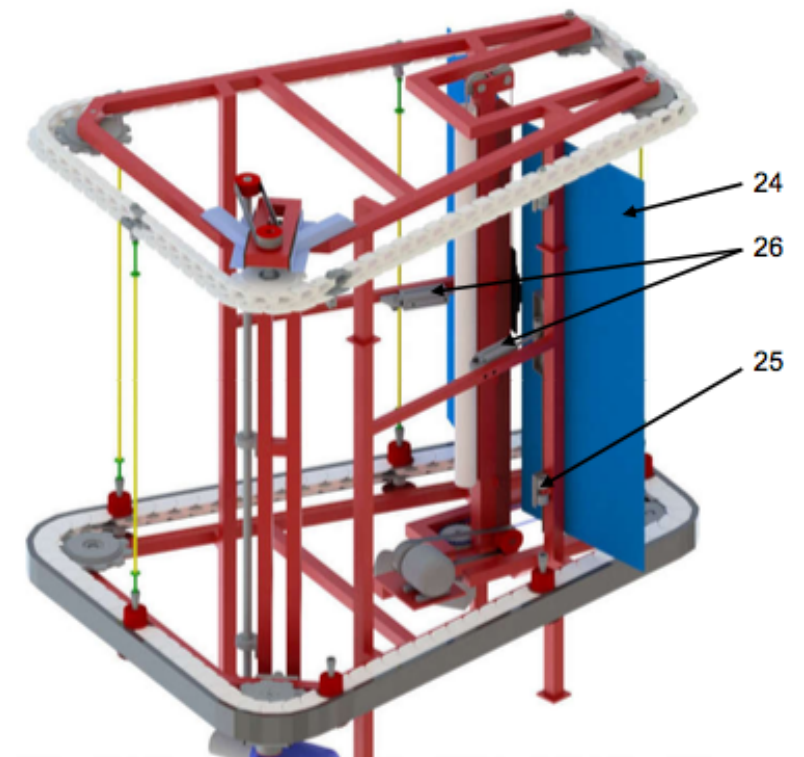


# APLIKACYJNA PRACA DYPLOMOWA z dziedziny MECHATRONIKI

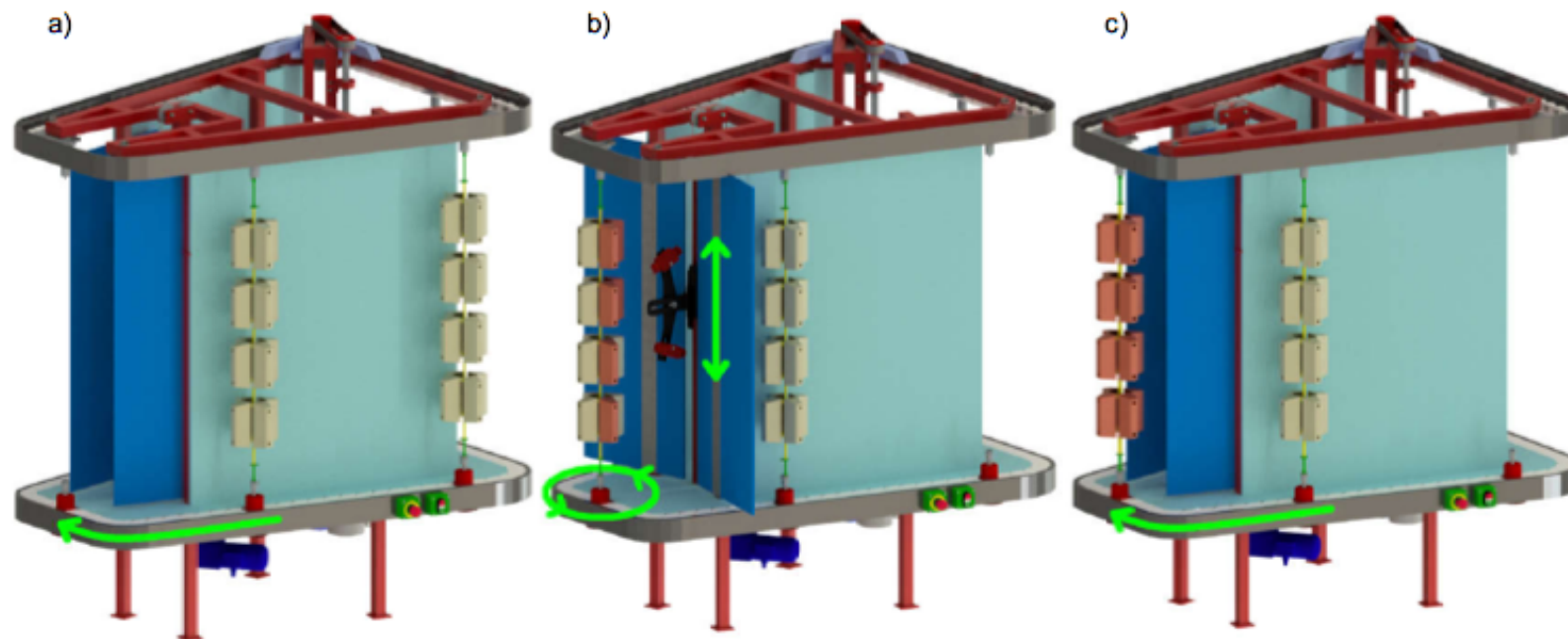
(automat lakierniczy sterowany przy pomocy PLC)



Rys. Z3.2.6. Model automatu lakierniczego z lokalizacją sensorów i przycisków sterowniczych



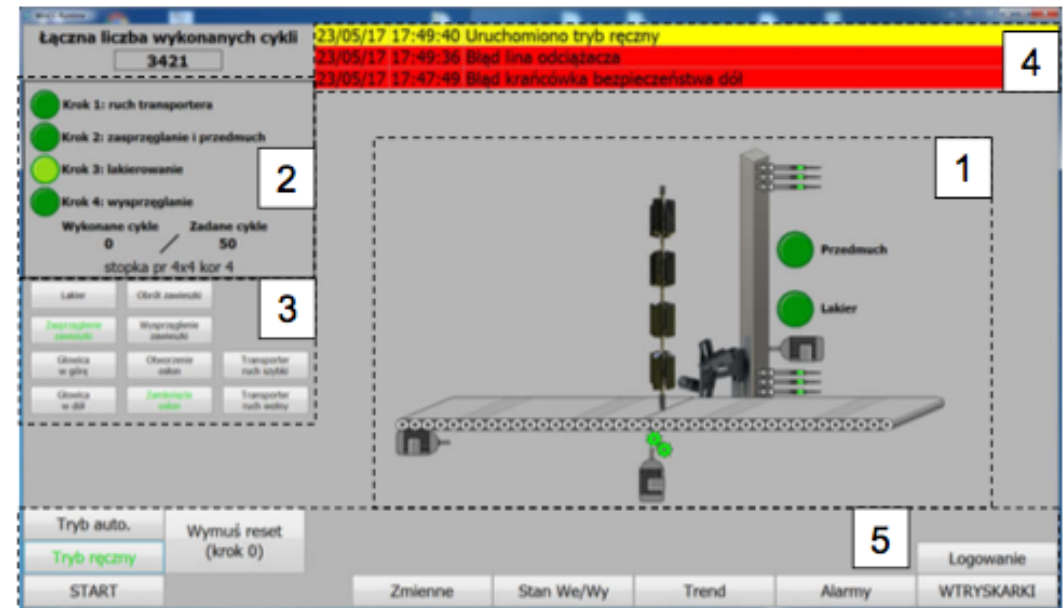
Rys. 1.5.9. Model ruchomych osłon przestrzeni roboczej



Rys. 1.5.10. Wizualizacja etapów pracy automatu lakierniczego: a) transport zawieszki do przestrzeni roboczej; b) proces przedmuchu i lakierowania; c) transport zawieszki z polakierowanymi elementami poza przestrzeń roboczą



Rys. 1.8.1. Panel HMI – ekran główny



Rys. Z2.1.1. SCADA – ekran główny automatu lakierniczego