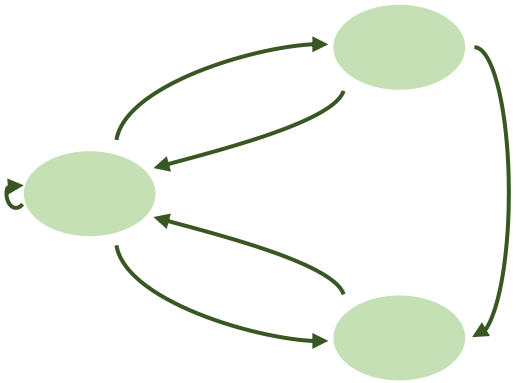
Elementy automatyki



Układy sekwencyjne

Sterowniki PLC

Metoda SFC

Materiały
http://staff.uz.zgora.pl/ipajak

Pojęcia podstawowe

Sekwencyjne układy cyfrowe

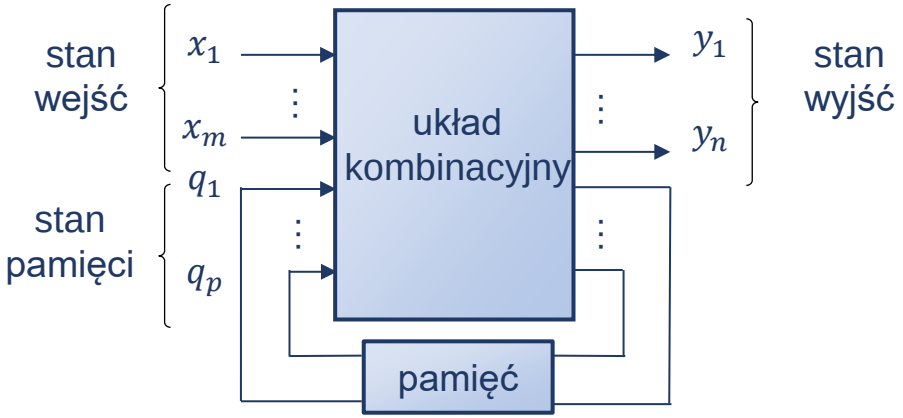
Układ sekwencyjny to układ cyfrowy, w którym zależność między wartościami sygnałów wejściowych (tzw. stan wejść) i wyjściowych (tzw. stan wyjść) nie jest jednoznaczna, tzn.:

- temu samemu stanowi wejść mogą odpowiadać różne stany wyjść
- stan wyjść układu sekwencyjnego zależy nie tylko od aktualnego stanu wejść ale również od poprzednich stanów wejść tzn. od kolejności (sekwencji) zmian stanów wejść

Sekwencyjne układy cyfrowe

Układy sekwencyjne zapamiętują historię oddziaływań sygnałów wejściowych, dlatego nazywane są układami z pamięcią.

Pamięć układu, tzw. blok pamięci, tworzy minimalna liczba wielkości niezbędnych do opisania wszystkich skutków przeszłych oddziaływań. Wielkości te nazywane są stanem pamięci wewnętrznej (stanem pamięci lub stanem wewnętrznym) układu.



Sekwencyjne układy cyfrowe

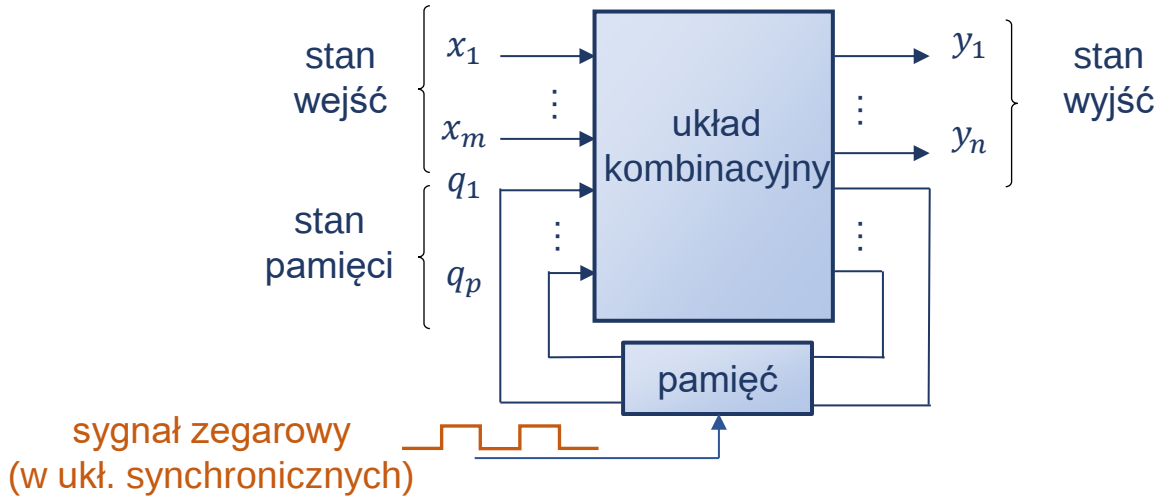
Uwzględniając sposób oddziaływania sygnałów wejściowych na układ (a właściwie na blok pamięci) układy sekwencyjne dzielone są na:

układy asynchroniczne

zmiana stanu pamięci następuje bezpośrednio (w dowolnej chwili czasu) pod wpływem zmiany stanu wejść

układy synchroniczne

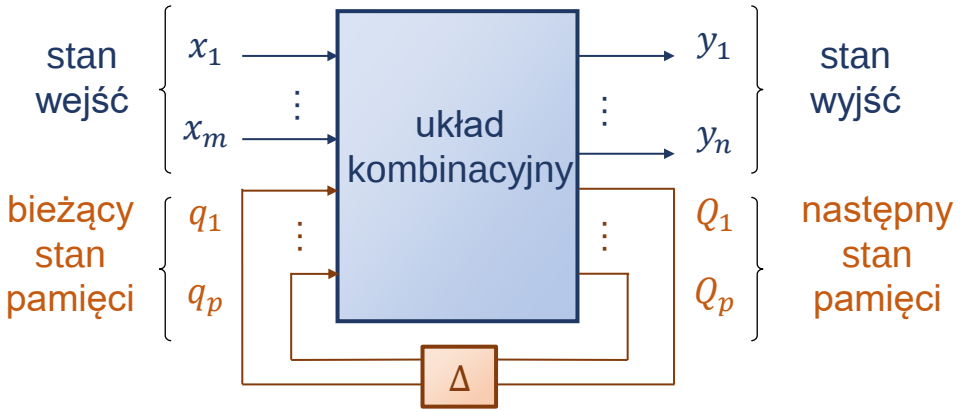
zmiana stanu pamięci następuje tylko w ściśle określonych chwilach czasu wyznaczanych przez dodatkowy sygnał wejściowy układu tzw. sygnał taktujący (zegarowy, synchronizujący)



Asynchroniczne układy sekwencyjne

Model Huffmana

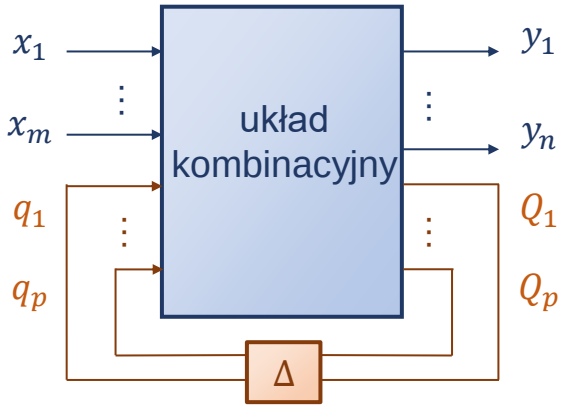
Huffman zaproponował model, w którym układ kombinacyjny został rozbudowany o pętle sprzężeń zwrotnych z dodatkowymi elementami opóźniającymi Δ. Takie podejście pozwoliło na realizację pamięci układu sekwencyjnego.



Asynchroniczne układy sekwencyjne

Działanie

- jeśli dla danego stanu wejść sygnały stanu pamięci $q_i = Q_i$ dla każdego $i = 1 \cdots p$ to układ jest w **stabilnym stanie wewnętrznym** lub krótko w **stanie stabilnym**
- po zmianie stanu wejściowego układ kombinacyjny wygeneruje nowe wartości sygnałów Q_i a część z nich, ze względu na istnienie opóźnień Δ_i będzie różna od wartości aktualnych, tzn. $q_i \neq Q_i$, oznacza, to że układ znajdzie się w **niestabilnym stanie wewnętrznym** lub krótko w **stanie niestabilnym**
- po upływie czasu Δ_i sygnały stanu wewnętrznego bieżącego i następnego mogą mieć już takie same wartości i w konsekwencji układ będzie w stanie stabilnym, układ może również znaleźć się w kolejnym stanie niestabilnym i przez serię stanów niestabilnych może przejść do stanu stabilnego



Asynchroniczne układy sekwencyjne

Ograniczenia

Podstawowym wymaganiem pozwalającym na poprawne funkcjonowanie układu jest dopuszczenie zmian stanu wejściowego tylko w stanach stabilnych.

Tryb podstawowy

Zakłada się, że po zmianie sygnału wejściowego, kolejny sygnał wejściowy może się zmienić dopiero po przejściu układu w stan stabilny, tzn. nie jest możliwa jednoczesna zmiana kilku sygnałów wejściowych.

Sekwencyjne układy cyfrowe

Opis

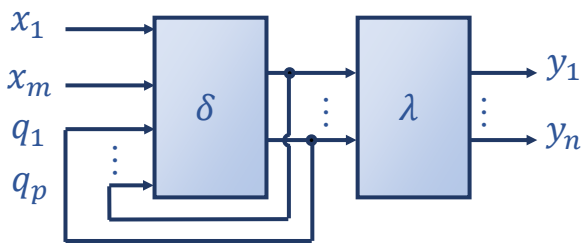
Układy definiowane są formalnie przy pomocy tzw. "piątki":

$$(\mathbb{X}, \mathbb{A}, \mathbb{Y}, \delta, \lambda)$$

- $\mathbb{X}$ to zbiór wszystkich stanów wejściowych (zawiera maksymalnie 2^m stanów, m to liczba sygnałów wejściowych),
- $\mathbb{A}$ to zbiór wszystkich stanów wewnętrznych (zawiera maksymalnie 2^p stanów, p to liczba elementów tworzących pamięć układu,
- $\mathbb{Y}$ to zbiór wszystkich stanów wyjściowych (zawiera maksymalnie 2^n stanów, n to liczba sygnałów wyjściowych),
- δ to **funkcja przejść**, która odpowiada za pamięć układu, $\delta\colon \mathbb{A} \times \mathbb{X} \rightarrow \mathbb{A}$,
- λ to **funkcja wyjść**, która określa stan wyjściowy układu, w układach o architekturze Moore'a $\lambda\colon \mathbb{A} \rightarrow \mathbb{Y}$, a w architekturze Mealy'ego $\lambda\colon \mathbb{A} \times \mathbb{X} \rightarrow \mathbb{Y}$.

Sekwencyjne układy cyfrowe

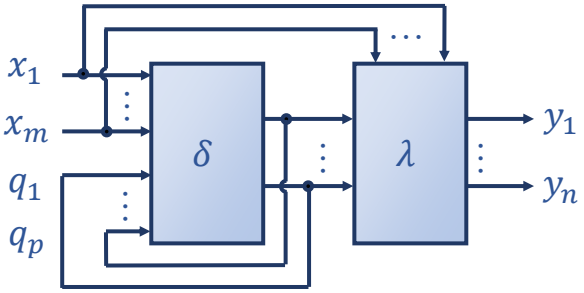
Układ Moore'a



$$\delta\colon \mathbb{A} \times \mathbb{X} \to \mathbb{A}$$

$$\lambda\colon \mathbb{A} \to \mathbb{Y}$$

Układ Mealy'ego



$$\delta\colon \mathbb{A} \times \mathbb{X} \to \mathbb{A}$$

$$\lambda\colon \mathbb{A} \times \mathbb{X} \to \mathbb{Y}$$

Opis układu

$$\begin{bmatrix} Q_1 \\ \vdots \\ Q_p \end{bmatrix} = \delta\left(\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} q_1 \\ \vdots \\ q_p \end{bmatrix}\right)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \lambda\left(\begin{bmatrix} q_1 \\ \vdots \\ q_p \end{bmatrix}\right)$$

$$\begin{bmatrix} Q_1 \\ \vdots \\ Q_p \end{bmatrix} = \delta\left(\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} q_1 \\ \vdots \\ q_p \end{bmatrix}\right)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \lambda\left(\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} q_1 \\ \vdots \\ q_p \end{bmatrix}\right)$$

Metody opisu

Metody opisu układów sekwencyjnych

Metody opisu

- metody zewnętrzne

  metody uwzględniają jedynie zależności pomiędzy sygnałami wejściowymi i wyjściowymi, nie uwzględniając informacji o stanach wewnętrznych układu, nie pozwalają na bezpośrednie zdefiniowanie funkcji przejść i wyjść układu

- metody pełne

  w odróżnieniu od metod zewnętrznych uwzględniają również stany wewnętrzne układu

metody	
zewnętrzne	**pełne**
• opis słowny • wykres czasowy	• graf przejść • tablice przejść i tablice wyjść

Metody opisu układów sekwencyjnych

Opis słowny

- przedstawia funkcjonowanie układu w sposób opisowy,
- z opisu powinny wynikać wszystkie stany wejściowe (a właściwie wszystkie możliwe sekwencje tych stanów),
- z opisu powinny wynikać wszystkie stany wyjściowe (lub ich sekwencje),
- z opisu nie wynika bezpośrednio ilość elementów pamięci niezbędnych do realizacji układu,
- opis komplikuje się w miarę wzrostu złożoności układu.

Układ 1

Układ umożliwia cykliczne włączanie i wyłączanie urządzenia przy pomocy przycisku Jeśli urządzenie nie pracuje, wciśnięcie przycisku powoduje jego włączenie, jeśli urządzenie pracuje, wciśnięcie przycisku powoduje jego wyłączenie. Zwalnianie przycisku nie powoduje zmian stanu pracy urządzenia.

Układ 2

Układ umożliwia włączanie i wyłączanie urządzenia przy pomocy dwóch przełączników.

Włączenie przełącznika włączającego powoduje włączenie urządzenia. Wyłączenie tego przełącznika nie powoduje jednak zatrzymania pracy urządzenia.

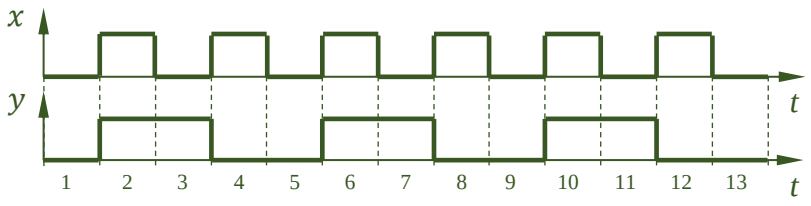
Urządzenie może być wyłączone tylko przy pomocy przełącznika wyłączającego. Przełącznik wyłączający ma wyższy priorytet i w sytuacji gdy obydwa przełączniki są włączone urządzenie nie pracuje.

Metody opisu układów sekwencyjnych

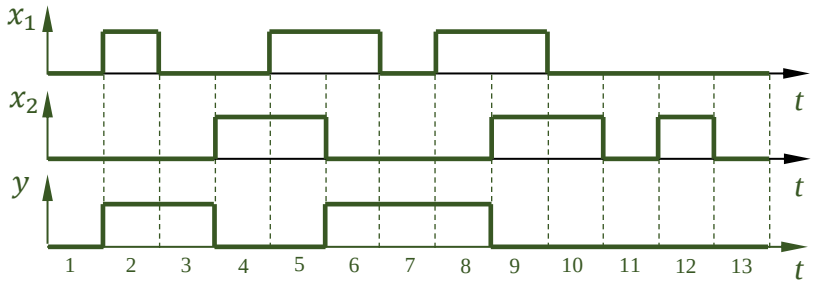
Wykres czasowy

- przedstawia przebieg w czasie zmian sygnałów wejściowych i odpowiadających im sygnałów wyjściowych
- opóźnienia wynikające z czasu reakcji elementów układu nie są uwzględniane
- oś czasu na ogół nie odzwierciedla czasu trwania poszczególnych faz i skalowana jest taktami (takt odpowiada czasowi pomiędzy kolejnymi zmianami sygnałów wejściowych)
- do jednoznacznego opisu działania układu niezbędne jest zaznaczenie na wykresie wszystkich możliwych sekwencji stanów wejść i wyjść

Układ 1



Układ 2

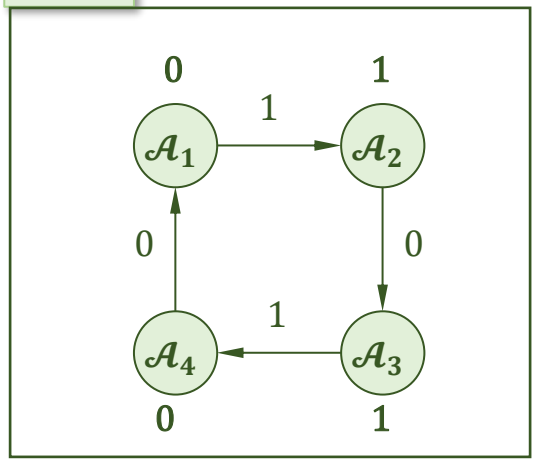


Metody opisu układów sekwencyjnych

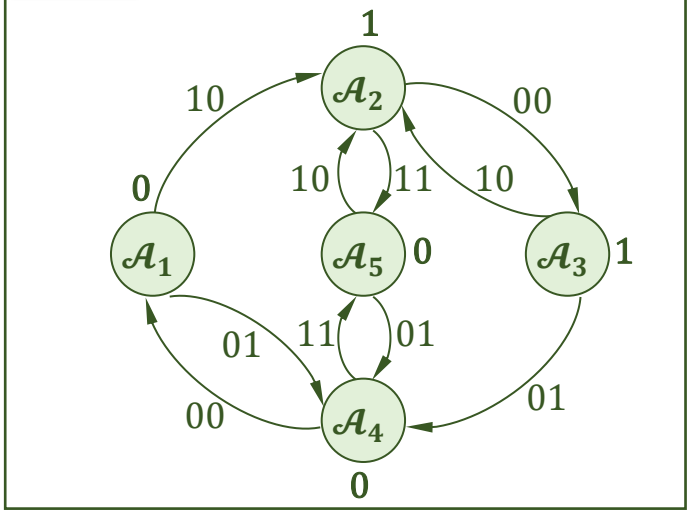
Graf przejść

- graf, którego wierzchołki odpowiadają stanom wewnętrznym układu a krawędzie oznaczają przejścia pomiędzy stanami wymuszane określonymi stanami wejściowymi
- w układach Moore'a stan wyjść zależy wyłącznie od stanu wewnętrznego więc jest przyporządkowywany odpowiedniemu wierzchołkowi grafu
- w układach Mealye'go stan wyjść zależy od stanu wewnętrznego i od stanu wejść jest więc przyporządkowany podobnie jak stan wejść odpowiedniej krawędzi grafu

Układ 1



Układ 2

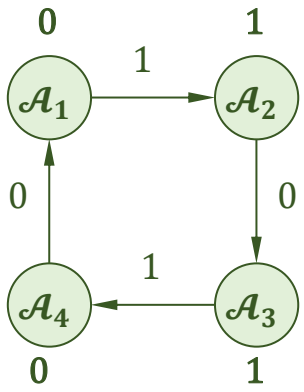


Metody opisu układów sekwencyjnych

Tablice przejść, tablice wyjść, tablice przejść–wyjść

- w formie tabelarycznej przedstawiają informacje zawarte w grafie przejść
- tablica przejść odpowiada funkcji przejść δ układu, wiersze opisywane są stanami wewnętrznymi, kolumny stanami wejściowymi, kratki tablicy zawierają stany wewnętrzne, do których przechodzi układ który był poprzednio w stanie określonym przez wiersz tablicy pod wpływem stanu wejściowego określonego przez kolumnę
- tablica wyjść odpowiada funkcji wyjść λ układu; w układach Moore'a: wiersze opisywane stanami wewnętrznymi, kolumny sygnałami wyjściowymi, kratki tablicy zawierają stany wyjściowe odpowiadające stanom wewnętrznym, tablica wyjść łączona jest z tablicą przejść w tablicę przejść – wyjść

Układ 1



$\mathcal{A}$ \ x	0	1
$\mathcal{A}_1$	$\mathcal{A}_1$	$\mathcal{A}_2$
$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_2$
$\mathcal{A}_3$	$\mathcal{A}_3$	$\mathcal{A}_4$
$\mathcal{A}_4$	$\mathcal{A}_1$	$\mathcal{A}_4$

$\mathcal{A}$	y
$\mathcal{A}_1$	0
$\mathcal{A}_2$	1
$\mathcal{A}_3$	1
$\mathcal{A}_4$	0

$\mathcal{A}$ \ x	0	1	y
$\mathcal{A}_1$	$\mathcal{A}_1$	$\mathcal{A}_2$	0
$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_2$	1
$\mathcal{A}_3$	$\mathcal{A}_3$	$\mathcal{A}_4$	1
$\mathcal{A}_4$	$\mathcal{A}_1$	$\mathcal{A}_4$	0

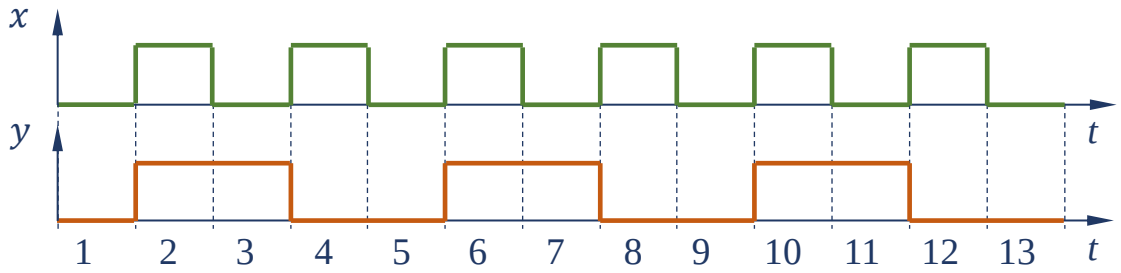
Układ 1

Opis słowny

Układ umożliwia cykliczne włączanie i wyłączanie urządzenia przy pomocy przycisku (przycisk jest łącznikiem monostabilnym pozostającym w stanie włączony tylko gdy jest wciskany przez operatora, po zwolnieniu przechodzi w stan wyłączony).

Jeśli urządzenie nie pracuje, wciśnięcie przycisku powoduje jego włączenie, jeśli urządzenie pracuje, wciśnięcie przycisku powoduje jego wyłączenie. Zwalnianie przycisku nie powoduje zmian stanu pracy urządzenia.

Wykres czasowy



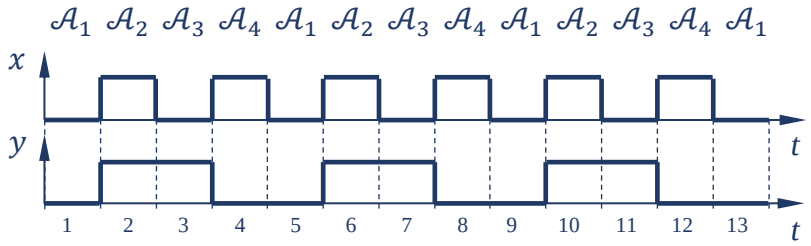
x — **przycisk**
y — **urządzenie**

układ jest sekwencyjny:

- w taktach nieparzystych $x = 0$ a $y = 0$ lub $y = 1$
- w taktach parzystych $x = 1$ a $y = 0$ lub $y = 1$

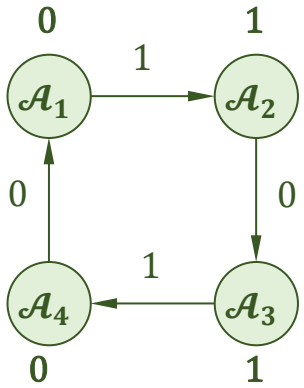
Układ 1

Graf przejść



Z analizy wykresu czasowego wynika, że układ może znajdować się w 4 stanach:

- $\mathcal{A}_1$ – przycisk zwolniony i urządzenie nie pracuje ($x = 0$ i $y = 0$)
- $\mathcal{A}_2$ – przycisk wciśnięty i urządzenie pracuje ($x = 1$ i $y = 1$)
- $\mathcal{A}_3$ – przycisk zwolniony i urządzenie pracuje ($x = 0$ i $y = 1$)
- $\mathcal{A}_4$ – przycisk wciśnięty i urządzenie nie pracuje ($x = 1$ i $y = 0$)



- wierzchołki odpowiadają stanom wewnętrznym układu
- krawędzie odpowiadają sygnałom wejściowym, które wymuszają zmiany stanów
- w układach Moore'a sygnały wyjściowe są przyporządkowywane wierzchołkom grafu

Układ 1

Tablica przejść, tablica wyjść, tablica przejść–wyjść

$\mathcal{A}$ \ x	0	1
$\mathcal{A}_1$	$\mathcal{A}_1$	$\mathcal{A}_2$
$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_2$
$\mathcal{A}_3$	$\mathcal{A}_3$	$\mathcal{A}_4$
$\mathcal{A}_4$	$\mathcal{A}_1$	$\mathcal{A}_4$

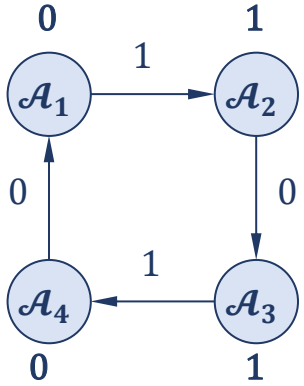
tablica przejść

$\mathcal{A}$	y
$\mathcal{A}_1$	0
$\mathcal{A}_2$	1
$\mathcal{A}_3$	1
$\mathcal{A}_4$	0

tablica wyjść

$\mathcal{A}$ \ x	0	1	y
$\mathcal{A}_1$	$\mathcal{A}_1$	$\mathcal{A}_2$	0
$\mathcal{A}_2$	$\mathcal{A}_3$	$\mathcal{A}_2$	1
$\mathcal{A}_3$	$\mathcal{A}_3$	$\mathcal{A}_4$	1
$\mathcal{A}_4$	$\mathcal{A}_1$	$\mathcal{A}_4$	0

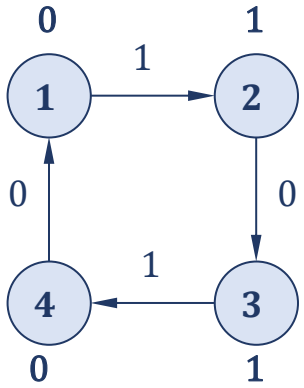
tablica przejść – wyjść



$\mathcal{A}$ \ x	0	1
1	①	2
2	3	②
3	③	4
4	1	④

$\mathcal{A}$	y
1	0
2	1
3	1
4	0

$\mathcal{A}$ \ x	0	1	y
1	①	2	0
2	3	②	1
3	③	4	1
4	1	④	0



w dolnym rzędzie stany zostały opisane numerami porządkowymi a stany stabilne otoczone kółkiem

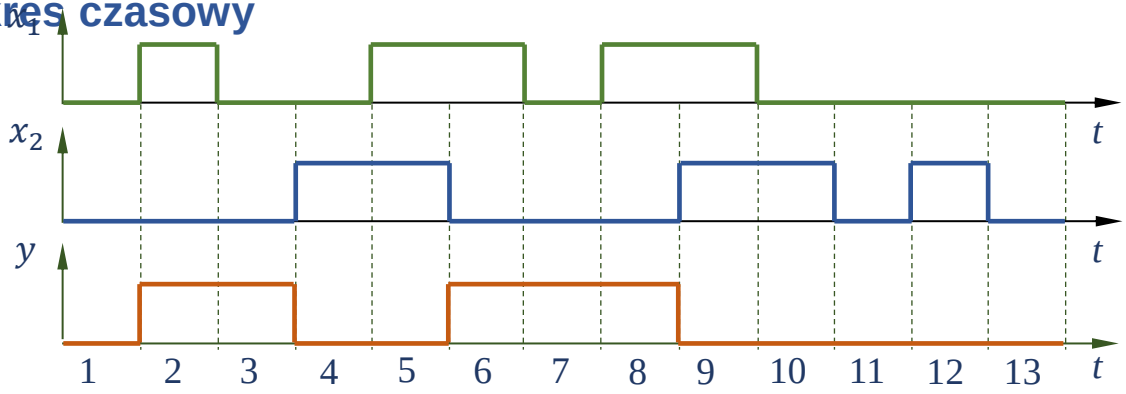
Układ 2

Opis słowny

Układ umożliwia włączanie i wyłączanie urządzenia przy pomocy dwóch przełączników (przełączniki po naciśnięciu zmieniają trwale swój stan: przełączają się ze stanu wyłączony we włączony i ze stanu włączony w wyłączony).

Włączenie przełącznika włączającego powoduje włączenie urządzenia. Wyłączenie tego przełącznika nie powoduje jednak zatrzymania pracy urządzenia. Urządzenie może być wyłączone tylko przy pomocy przełącznika wyłączającego. Przełącznik wyłączający ma wyższy priorytet i w sytuacji gdy obydwa przełączniki są włączone urządzenie nie pracuje. Dodatkowo zakłada się, że układ pracuje w trybie podstawowym.

Wykres czasowy



x_1 — włącznik
x_2 — wyłącznik
y — urządzenie

układ jest sekwencyjny: w taktach 1, 3, 7, 11, 13 $x_1 = 0, x_2 = 0$ a $y = 0$ lub $y = 1$

Układ 2

Graf przejść

Z analizy wykresu czasowego wynika, że układ może znajdować się w 5 stanach:

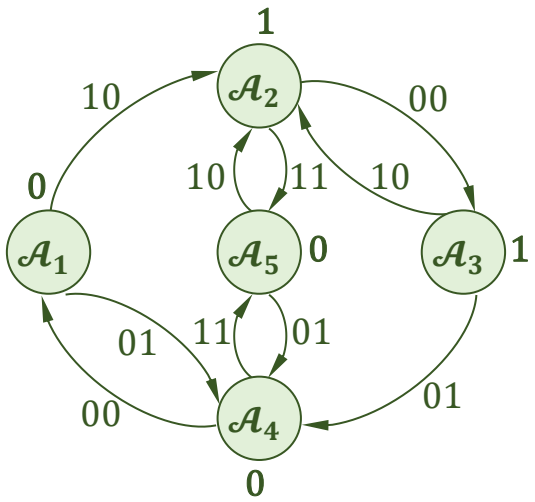
$\mathcal{A}_1$ – przełączniki wyłączone, urządzenie nie pracuje ($x_1 = 0$, $x_2 = 0$, $y = 0$),

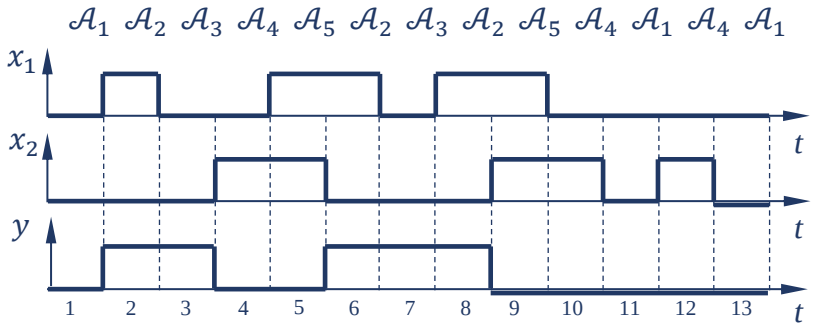
$\mathcal{A}_2$ – włączony włącznik, urządzenie pracuje ($x_1 = 1$, $x_2 = 0$, $y = 1$),

$\mathcal{A}_3$ – przełączniki wyłączone, urządzenie pracuje ($x_1 = 0$, $x_2 = 0$, $y = 1$),

$\mathcal{A}_4$ – włączony wyłącznik, urządzenie nie pracuje ($x_1 = 0$, $x_2 = 1$, $y = 0$),

$\mathcal{A}_5$ – przełączniki włączone, urządzenie nie pracuje ($x_1 = 1$, $x_2 = 1$, $y = 0$).





Układ 2

Tablica przejść–wyjść

$\mathcal{A}$ \ x_1x_2	00	01	11	10
$\mathcal{A}_1$	$\mathcal{A}_1$	$\mathcal{A}_4$	–	$\mathcal{A}_2$
$\mathcal{A}_2$	$\mathcal{A}_3$	–	$\mathcal{A}_5$	$\mathcal{A}_2$
$\mathcal{A}_3$	$\mathcal{A}_3$	$\mathcal{A}_4$	–	$\mathcal{A}_2$
$\mathcal{A}_4$	$\mathcal{A}_1$	$\mathcal{A}_4$	$\mathcal{A}_5$	–
$\mathcal{A}_5$	–	$\mathcal{A}_4$	$\mathcal{A}_5$	$\mathcal{A}_2$

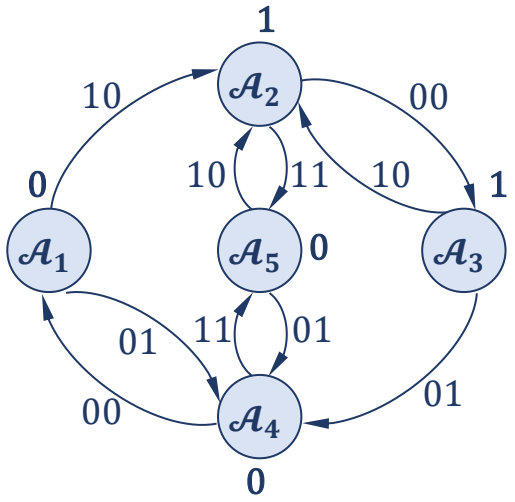
tablica przejść

$\mathcal{A}$	y
$\mathcal{A}_1$	0
$\mathcal{A}_2$	1
$\mathcal{A}_3$	1
$\mathcal{A}_4$	0
$\mathcal{A}_5$	0

tablica wyjść

$\mathcal{A}$ \ x_1x_2	00	01	11	10	y
$\mathcal{A}_1$	$\mathcal{A}_1$	$\mathcal{A}_4$	–	$\mathcal{A}_2$	0
$\mathcal{A}_2$	$\mathcal{A}_3$	–	$\mathcal{A}_5$	$\mathcal{A}_2$	1
$\mathcal{A}_3$	$\mathcal{A}_3$	$\mathcal{A}_4$	–	$\mathcal{A}_2$	1
$\mathcal{A}_4$	$\mathcal{A}_1$	$\mathcal{A}_4$	$\mathcal{A}_5$	–	0
$\mathcal{A}_5$	–	$\mathcal{A}_4$	$\mathcal{A}_5$	$\mathcal{A}_2$	0

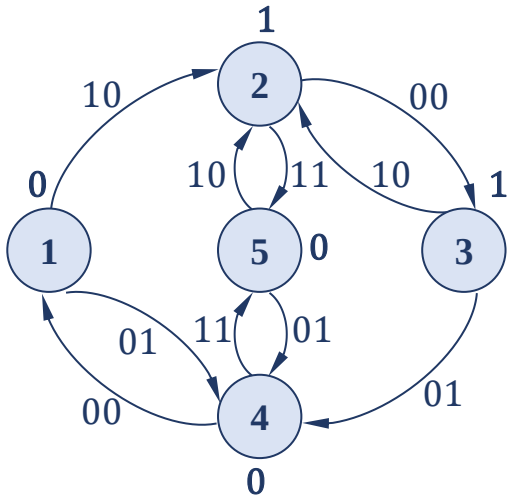
tablica przejść – wyjść



x_1x_2 \ $\mathcal{A}$	00	01	11	10
1	①	4	–	2
2	3	–	5	②
3	③	4	–	2
4	1	④	5	–
5	–	4	⑤	2

$\mathcal{A}$	y
1	0
2	1
3	1
4	0
5	0

x_1x_2 \ $\mathcal{A}$	00	01	11	10	y
1	①	4	–	2	0
2	3	–	5	②	1
3	③	4	–	2	1
4	1	④	5	–	0
5	–	4	⑤	2	0

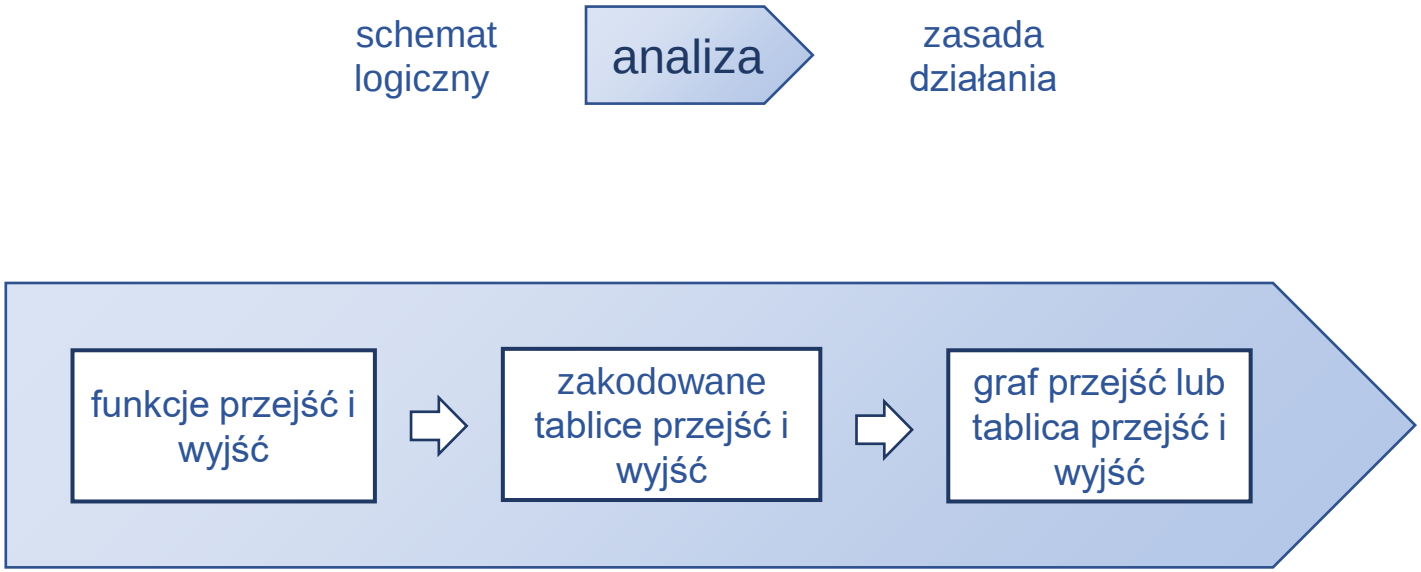


Analiza układów sekwencyjnych

Analiza układów sekwencyjnych

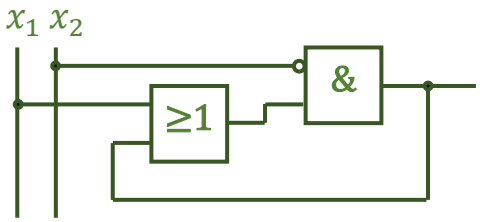
Analiza

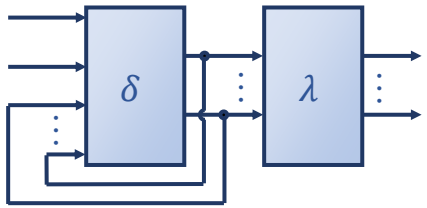
Głównym celem analizy jest zrozumienie zasad działania układu, w przypadku gdy znana jest jego struktura; celem analizy może być również chęć wykrycia źródeł niewłaściwego zachowania układu (hazard, wyścig, niestabilność).



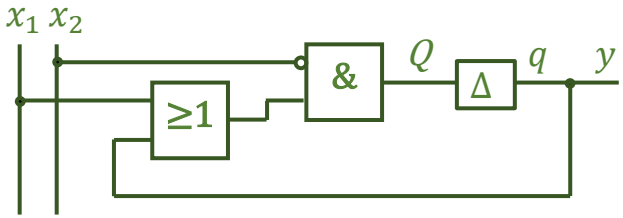
Analiza – układ A1

Funkcje przejść i wyjść





- układ ma wyłącznie blok pamięci (opisywany funkcją przejść δ),
- działanie bloku wyjściowego (opisywanego funkcją wyjść λ) należy więc opisać funkcją tożsamościową (brak dodatkowego przetwarzania sygnału),
- przed zapisem funkcji wygodnie jest wprowadzić element opóźniający wprowadzający stany bieżący q i następny Q.



funkcja przejść

$$Q = \bar{x}_2(x_1 + q)$$

funkcja wyjść

$$y = q$$

Analiza – układ A1

Zakodowana tablica przejść–wyjść

q \ x_1x_2	00	01	11	10	y
0	0	0	0	1	0
1	1	0	0	1	1

Q

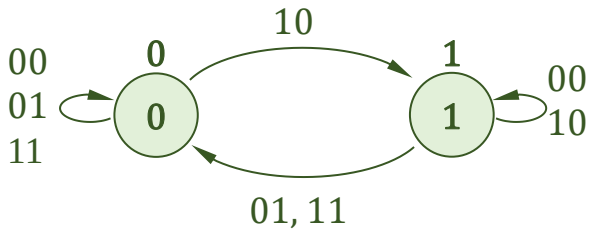
funkcja przejść

$$Q = \bar{x}_2(x_1 + q)$$

funkcja wyjść

$$y = q$$

Graf przejść

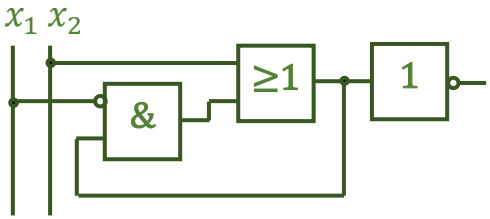


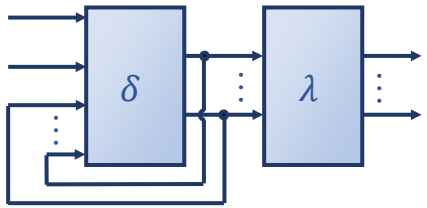
q \ x_1x_2	00	01	11	10	y
0	(0)	(0)	(0)	1	0
1	(1)	0	0	(1)	1

Q

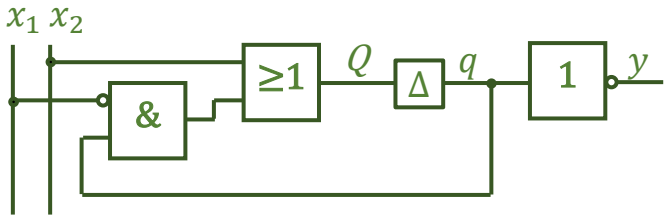
Analiza – układ A2

Funkcje przejść i wyjść





- układ ma blok pamięci i blok wyjściowy,
- blok wyjściowy to inwerter, realizuje funkcję negacji,
- po wprowadzeniu elementu opóźniającego i stanów bieżącego q i następnego Q można zapisać funkcje δ i λ.



funkcja przejść

$$Q = \bar{x}_1 q + x_2$$

funkcja wyjść

$$y = \bar{q}$$

Analiza – układ A2

Zakodowana tablica przejść – wyjść

q \ x_1x_2	00	01	11	10	y
0	0	1	1	0	1
1	1	1	1	0	0

Q

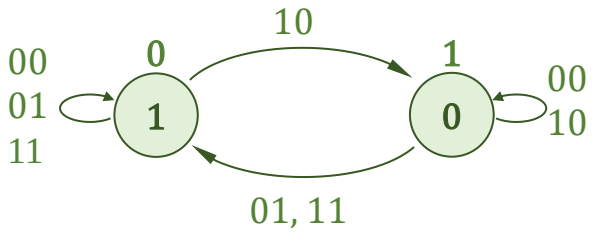
funkcja przejść

$$Q = \bar{x}_1 q + x_2$$

funkcja wyjść

$$y = \bar{q}$$

Graf przejść

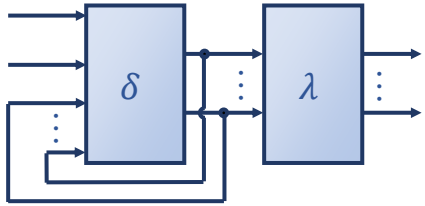


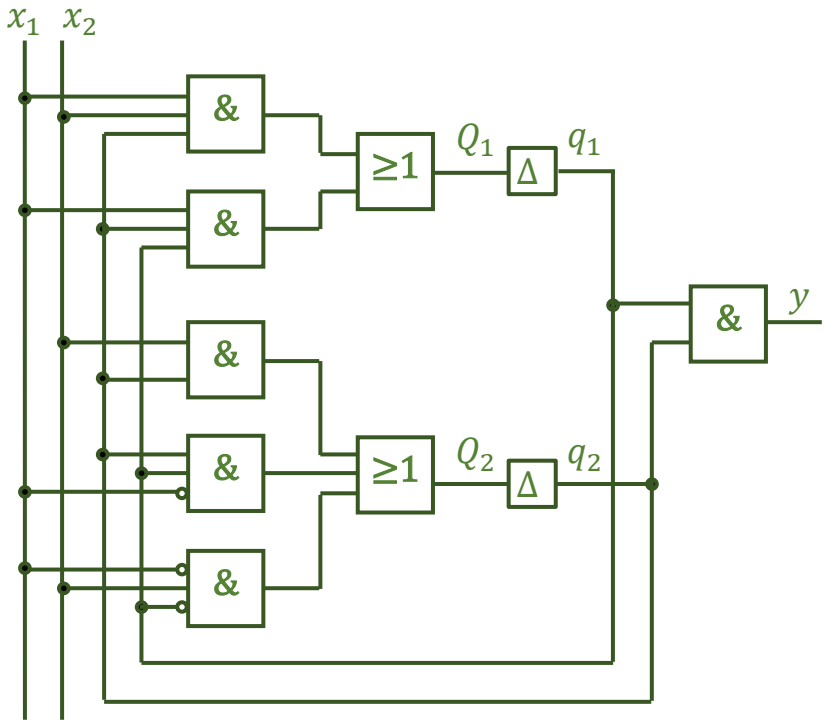
q \ x_1x_2	00	01	11	10	y
0	(0)	1	1	(0)	1
1	(1)	(1)	(1)	0	0

Q

Analiza – układ A3

Funkcje przejść i wyjść





funkcja przejść

$$Q_1 = x_1 x_2 q_2 + x_1 q_1 q_2$$

$$Q_2 = x_2 q_2 + \bar{x}_1 q_1 q_2 + \bar{x}_1 x_2 \bar{q}_1$$

funkcja wyjść

$$y = q_1 q_2$$

Analiza – układ A3

Zakodowana tablica przejść – wyjść

q_1q_2 \ x_1x_2	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	1
10	0	0	0	0

Q_1

q_1q_2 \ x_1x_2	00	01	11	10
00	0	1	0	0
01	0	1	1	0
11	1	1	1	0
10	0	0	0	0

Q_2

q_1q_2	y
00	0
01	0
11	1
10	0

funkcja przejść

$$Q_1 = x_1x_2q_2 + x_1q_1q_2$$

$$Q_2 = x_2q_2 + \bar{x}_1q_1q_2 + \bar{x}_1x_2\bar{q}_1$$

funkcja wyjść

$$y = q_1q_2$$

q_1q_2 \ x_1x_2	00	01	11	10	y
00	00	01	00	00	**0**
01	00	01	11	00	**0**
11	01	01	11	10	**1**
10	00	00	00	00	**0**

Analiza – układ A3

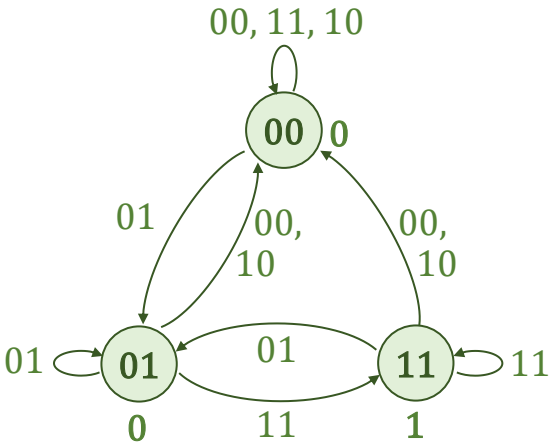
Tablica przejść – wyjść, graf przejść

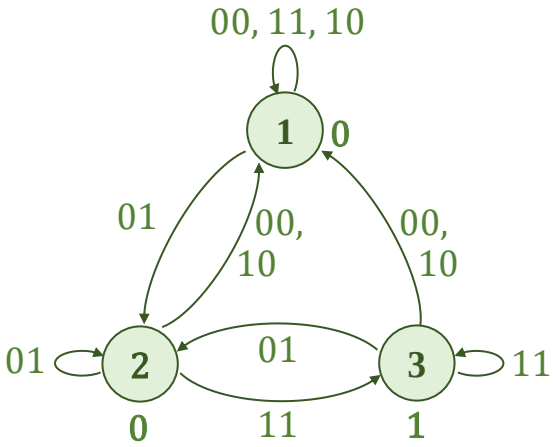
q_1q_2 \ x_1x_2	00	01	11	10	y
00	(00)	01	(00)	(00)	0
01	00	(01)	11	00	0
11	01	01	(11)	10	1
10	00	00	00	00	0

kody

00 – 1
01 – 2
11 – 3
10 – 4

q_1q_2 \ x_1x_2	00	01	11	10	y
1	(1)	2	(1)	(1)	0
2	1	(2)	3	1	0
3	2	2	(3)	4	1
4	1	1	1	1	0





wyjście układu:

- jest ustawiane jeżeli na wejściu wystąpi sekwencja: 01, 11,
- jest zerowane w pozostałych przypadkach.

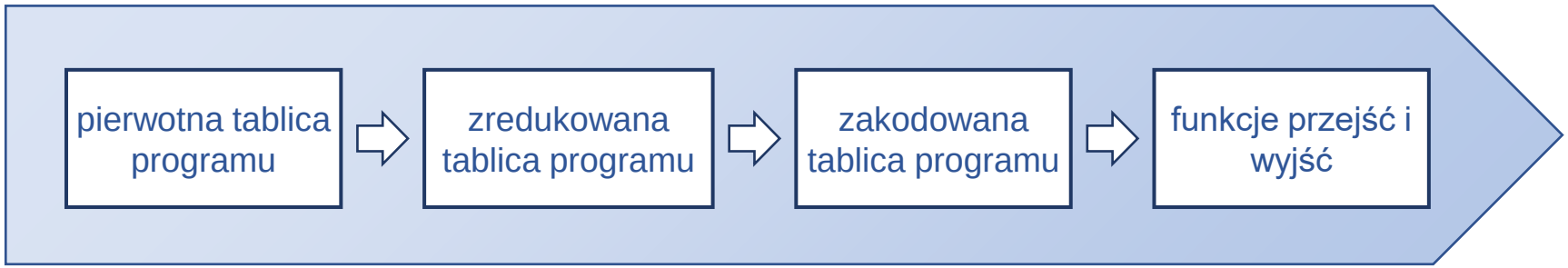
Synteza układów sekwencyjnych

Synteza układów sekwencyjnych

Synteza

Synteza to proces odwrotny do analizy, prowadzi od założeń definiujących sposób działania układu do jego projektu.

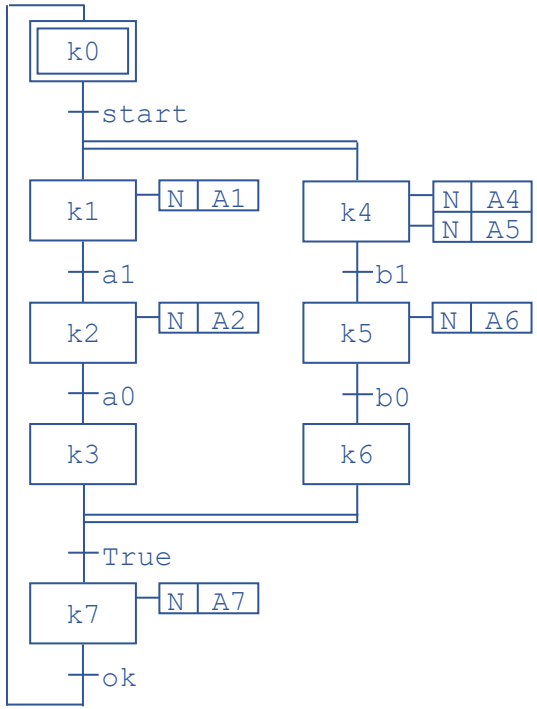
Metoda Huffmana



PLC - Metoda SFC

***SFC** – metoda sekwencyjnego schemat funkcjonalnego* (ang. Sequential Function Chart)

forma organizacji programu oparta na teorii sieci Petriego typu P/T (*pozycja*/*tranzycja*), opracowana jako alternatywny sposób opisu złożonych układów automatyki (ograniczenia metod klasycznych, np. metody Huffmana, przyczyniły się do poszukiwania nowych metod opisu układów sekwencyjnych).



- graficzna metoda organizacji programu,
- zadania sterowania przedstawiane są w postaci *grafu sekwencji* składającego się z *kroków* (*etapów*) i *tranzycji (przejść)* między tymi krokami,
- graf sekwencji to graf skierowany o wierzchołkach typu *krok* lub *tranzycja*,
- *kroki* i *tranzycie* oprogramowywane są w wybranym języku normy.

Metoda SFC – geneza

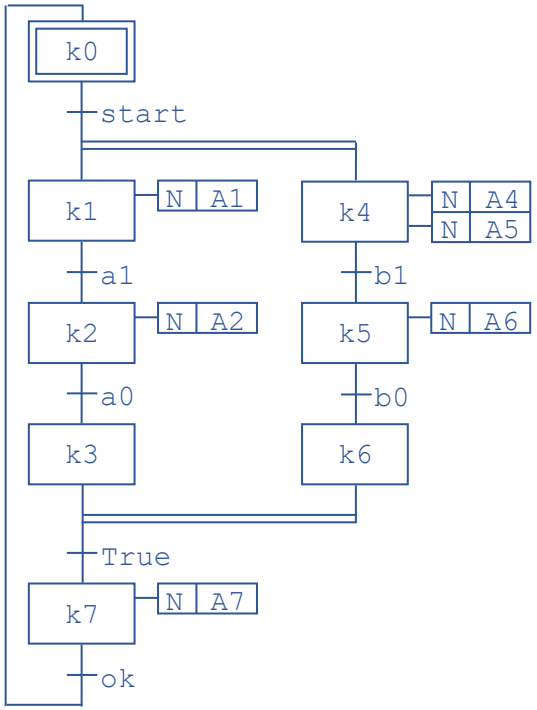
Grafcet

metoda opisana w 1977 przez firmę Telemecanique,

GRAPH 5, ***GRAPH 7***, ***Grafpol***, ***SFC***

metody będące modyfikacjami *Grafcet*, metoda *SFC* została opisana w normie IEC 61131-3.

Metody *Grafcet*, *GRAPH 5*, *GRAPH 7*, *Grafpol*, *SFC* stanowią podstawę języków programowania sterowników PLC, są *językami sekwencyjnych schematów funkcjonalnych* lub *językami sterowania sekwencyjnego*, norma IEC 61131 nazywa *SFC* metodą umożliwiającą organizację programu (a nie językiem).



Sieć SFC – kroki

Kroki reprezentują etapy sterowanego procesu, są zapisywane jako prostokąty z unikalnymi w ramach danej sieci nazwami umieszczonymi we wnętrzu.

a) 

b) 

a) symbol kroku (* * nazwa kroku)*

b) symbol kroku początkowego

Własności

- sterownik rozpoczyna wykonanie programu od kroku początkowego,
- sieć SFC zawiera tylko jeden krok początkowy,
- krok jest **aktywowany** w chwili rozpoczęcia jego wykonania,
- krok jest **dezaktywowany** w chwili rozpoczęcia wykonania kroku następnego,
- podczas aktywności kroku realizowana jest skojarzona z nim **akcja**.

Zmienne związane z krokiem

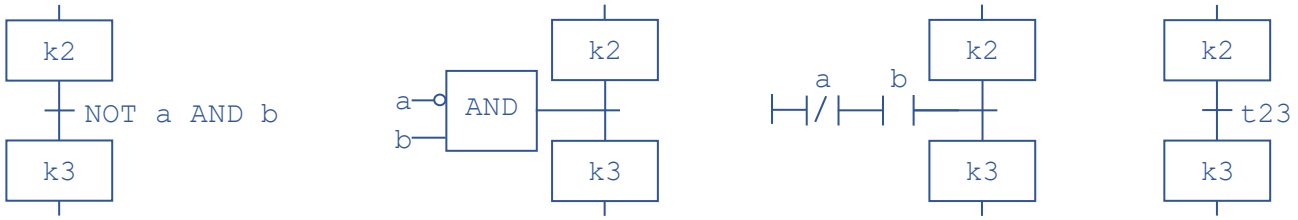
- `***.x` – zmienna typu `BOOL`, określa aktywność kroku (przyjmuje wartość 1 dla kroku aktywnego i 0 dla kroku nieaktywnego),
- `***.T` – zmienna typu `TIME` (czas) określa czas aktywności kroku.

Sieć SFC – tranzycje

Tranzycje (*przejścia*) opisują warunki, których spełnienie kończy wykonanie kończy wykonywanie kroku lub kroków bezpośrednio poprzedzających *tranzycję* i rozpoczyna wykonywanie kroku lub kroków bezpośrednio następujących po *tranzycji*.

Bezpośrednio obok symbolu może być zapisywana jej nazwa lub w*arunki przejścia*.

Warunek przejścia to wynik wyrażenia logicznego, które może być zapisane w językach IL, ST, FBD i LD.



Wybrane sposoby reprezentacji tranzycji i warunku przejścia $\bar{a}\,b$

ST

```
TRANSITION t23
  := NOT a AND b;
END_TRANSITION
```

FBD

```
TRANSITION t23:
```


```
END_TRANSITION
```

Sieć SFC – akcje

Akcje określają operacje realizowane podczas aktywności kroku, każdym krokiem może być skojarzony dowolny zbiór akcji, dopuszczalne są kroki nie połączone z żadnymi akcjami (K*roki* tego typu oczekują na spełnienie *warunku przejścia* do kolejnego *kroku*).

kwalifikator określa warunki wykonania akcji (kiedy i jak długo akcja jest wykonywana),

nazwa określa nazwę akcji.

Blok akcji, postać uproszczona

Akcje dopuszczane przez normę

- zmienna boolowska,
- ciąg instrukcji w języku tekstowym (IL, ST)
- zbiór obwodów w języku graficznym (LD, FBD),
- sieć SFC.

Sieć SFC – akcje

Kwalifikatory

brak N	akcja nieprzechowywana (ang. Non stored) instrukcje akcji wykonywane są przez cały czas aktywności kroku w każdym cyklu programowym sterownika
S	akcja zapamiętywana (ang. Set) akcja jest uruchamiana gdy krok zyskuje aktywność, utrata aktywności nie przerywa akcji – akcja jest wykonywania do momentu skasowania w innym kroku
R	akcja nadrzędnie kasowana (ang. overriding Reset) kasuje akcję uruchomioną w innym kroku
P	akcja impulsowa (ang. Pulse) instrukcje akcji wykonywane są tylko raz gdy krok zyskuje aktywność w niektórych sterownikach *akcje impulsowe* są wykonywane gdy krok zyskuje i traci aktywność

Sieć SFC – akcje

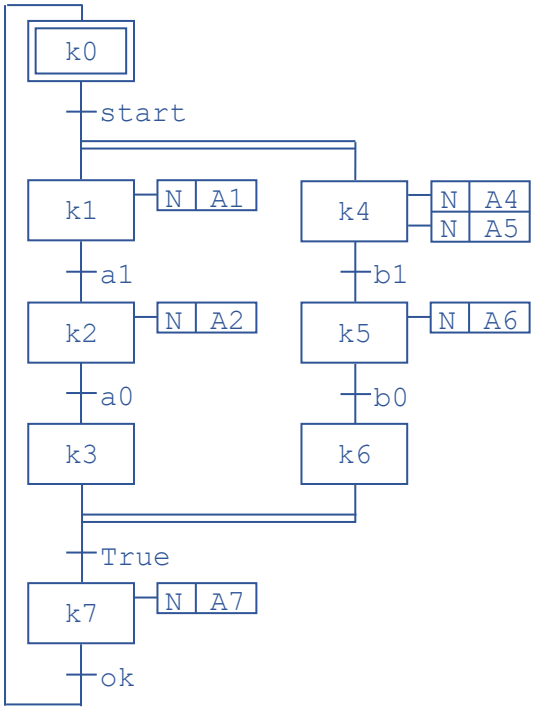
Kwalifikatory

`L`	akcja ograniczona w czasie (ang. time Limited) wykonywanie akcji kończy się albo po określonym czasie albo po utracie aktywności przez krok, użycie np.: `Lt#1s`
`D`	akcja opóźniona w czasie (ang. time Delayed) wykonywanie akcji rozpoczyna się po określonym czasie ale jeśli krok utraci aktywność przed uruchomieniem akcji to nie zostanie ona uruchomiona, użycie np.: `Dt#1s`
`SD`	akcja zapamiętywana i opóźniona (ang. Stored and time Delayed) akcja jest uruchamiana po określonym czasie niezależnie od tego czy krok utracił aktywność, akcja jest wykonywania do momentu skasowania w innym kroku np.: `SDt#1s`
`DS`	akcja opóźniona i zapamiętywana (ang. Delayed and Stored) akcja jest uruchamiana po określonym czasie pod warunkiem, że krok po upływie tego czasu jest jeszcze aktywny, akcja jest wykonywania do momentu skasowania w innym kroku, użycie np.: `DSt#1s`
`SL`	akcja zapamiętywana i ograniczona w czasie (ang. Stored and time Limited) wykonywanie akcji kończy się po określonym czasie niezależnie od aktywności skojarzonego kroku albo wcześniej jeśli zostanie skasowana, użycie np.: `SLt#1s`

Sieć SFC

Struktura grafu

- graf jest grafem skierowanym, wierzchołki grafu tzn. *kroki* i *tranzycje* połączone są krawędziami skierowanymi wskazującymi kierunek aktywowania kolejnych kroków,
- pomiędzy *dwoma kolejnymi krokami* w grafie musi znajdować się *dokładnie jedna tranzycja*,
- pomiędzy *dwoma kolejnymi tranzycjami* musi znajdować się *dokładnie jeden krok*,
- graf musi zawierać *krok początkowy*,
- *domyślnym kierunkiem* zmiany aktywności kroków jest kierunek *z góry do dołu*, na gałęziach grafu prowadzących z kroków położonych niżej do kroków położonych wyżej można umieszczać groty wskazujące kierunek połączeń.



Język ST

Język tekstu strukturalnego (ang. Structured Text)

jest odpowiednikiem języka wysokiego poziomu,

zawiera podobny zestaw instrukcji jak Pascal czy C,

podstawowymi elementami języka są *wyrażenia* i *instrukcje*.

Wyrażenie

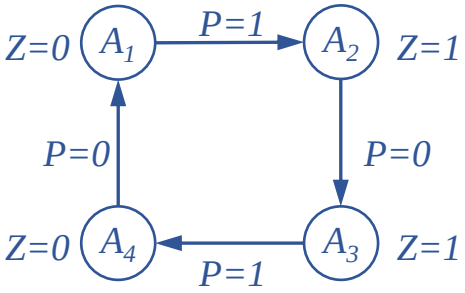
zwraca wartość wyznaczaną na podstawie występujących w wyrażeniu wartości, zmiennych, operatorów oraz funkcji.

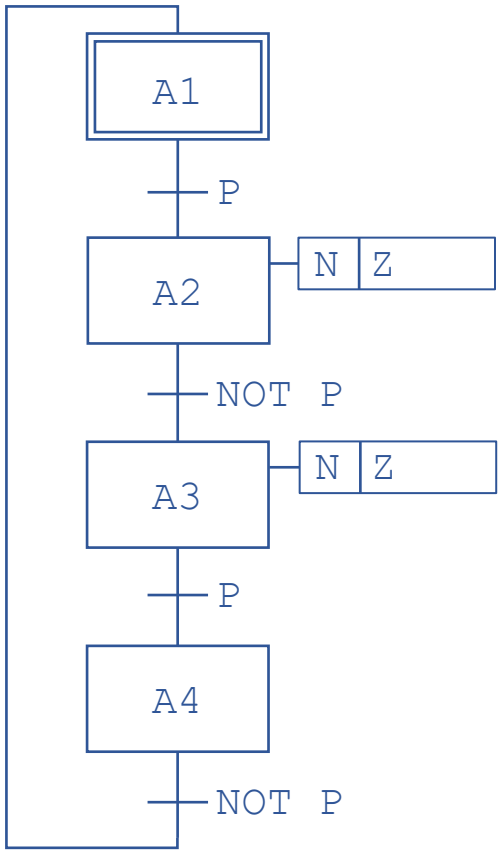
Wybrane operatory języka ST

Operator	Opis
`AND`	iloczyn logiczny `A AND B` – zwraca wartość prawda jeżeli zmienne `A` i `B` są prawdziwe
`OR`	suma logiczna `A OR B` – zwraca wartość prawda jeżeli jedna ze zmiennych `A` lub `B` jest prawdziwa
`NOT`	negacja logiczna `NOT A` – zwraca wartość prawda jeżeli zmienna `A` jest fałszywa

Przykład 1 – wersja I

Przycisk P zapala i gasi żarówkę Z. Jeżeli żarówka jest zgaszona naciśnięcie przycisku zapala ją, jeżeli żarówka jest zapalona naciśnięcie przycisku ją gasi. Zwolnienie przycisku nie zmienia stanu układu.





krok A1: krok startowy, brak skojarzonej akcji
żadna zmienna nie jest ustawiana **(Z=0)**

tranzycja: aktywność **kroku A1** kończy się gdy **P=1 (TRUE)**

krok A2: akcja z kwalifikatorem **N**
zmienna **Z** jest ustawiana na **1** w czasie aktywności kroku

tranzycja: aktywność **kroku A2** kończy się gdy **P=0 (FALSE)**

krok A3: akcja z kwalifikatorem **N**
zmienna **Z** jest ustawiana na **1** w czasie aktywności kroku

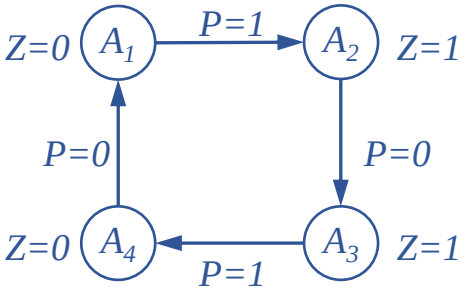
tranzycja: aktywność **kroku A3** kończy się gdy **P=1 (TRUE)**

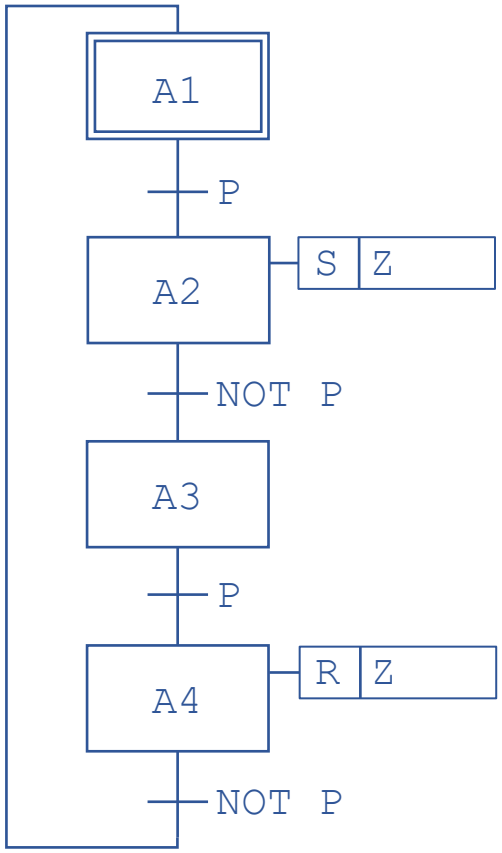
krok A4: brak skojarzonej akcji
żadna zmienna nie jest ustawiana **(Z=0)**

tranzycja: aktywność **kroku A4** kończy się gdy **P=0 (FALSE)**

Przykład 1 – wersja II

Przycisk P zapala i gasi żarówkę Z. Jeżeli żarówka jest zgaszona naciśnięcie przycisku zapala ją, jeżeli żarówka jest zapalona naciśnięcie przycisku ją gasi. Zwolnienie przycisku nie zmienia stanu układu.





krok A1: krok startowy, brak skojarzonej akcji
żadna zmienna nie jest ustawiana (**Z=0**)

tranzycja: aktywność **kroku A1** kończy się gdy **P=1** (**TRUE**)

krok A2: akcja z kwalifikatorem **S**
Z zostaje ustawiona na **1** do chwili skasowania

tranzycja: aktywność **kroku A2** kończy się gdy **P=0** (**FALSE**)

krok A3: brak skojarzonej akcji
Z pozostaje ustawiona (**Z=1**)

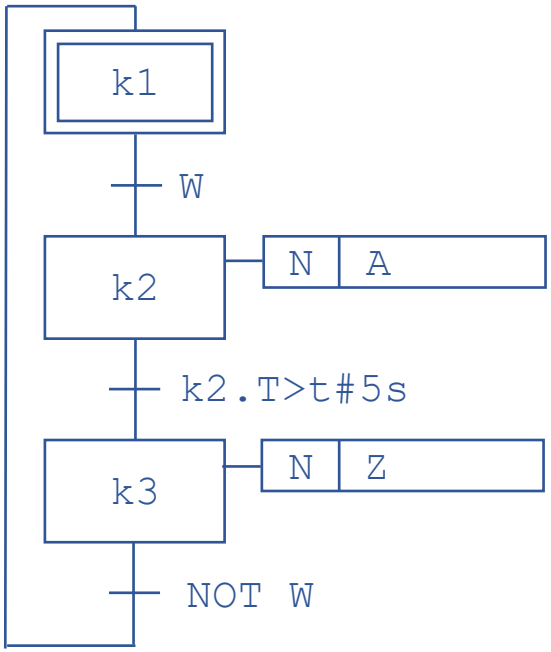
tranzycja: aktywność **kroku A3** kończy się gdy **P=1** (**TRUE**)

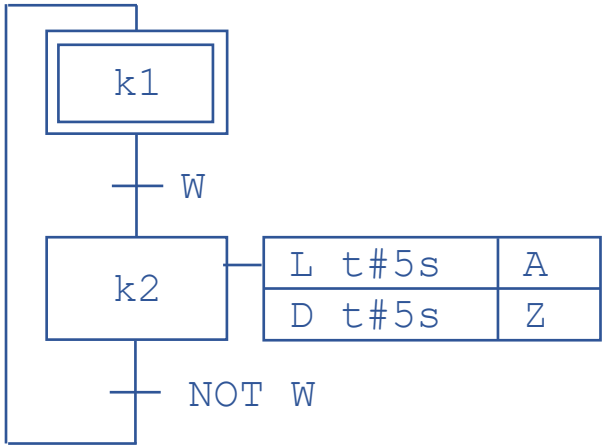
krok A4: akcja z kwalifikatorem **R**
ustawienie **Z** zostaje skasowane (**Z=0**)

tranzycja: aktywność **kroku A4** kończy się gdy **P=0** (**FALSE**)

Sieć SFC – Przykład 2

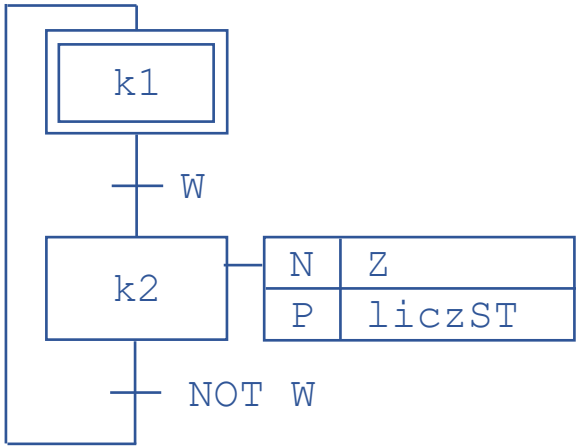
Przycisk W włącza i wyłączą urządzenie Z. Przed włączeniem urządzenia na 5 sekund włącza się alarm A, który jest wyłączany z chwilą włączenia urządzenia.



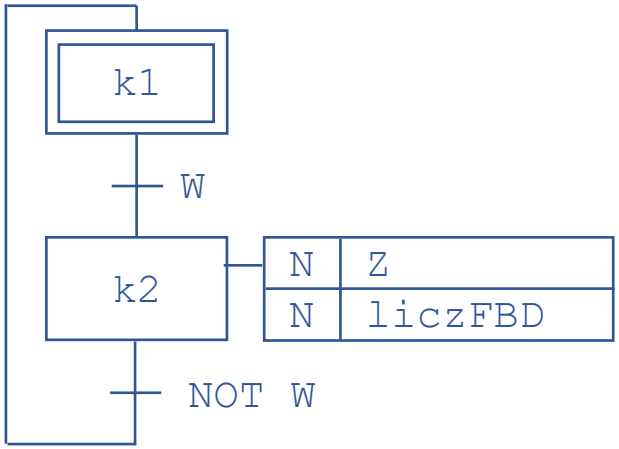


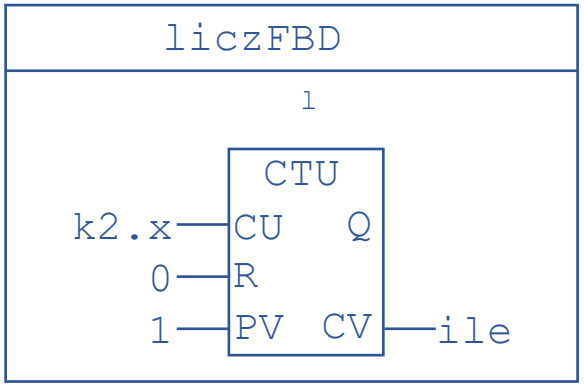
Sieć SFC – Przykład 3

Przycisk `W` włącza i wyłączą urządzenie `Z`. Włączenia sią zliczane w zmiennej `ile`.



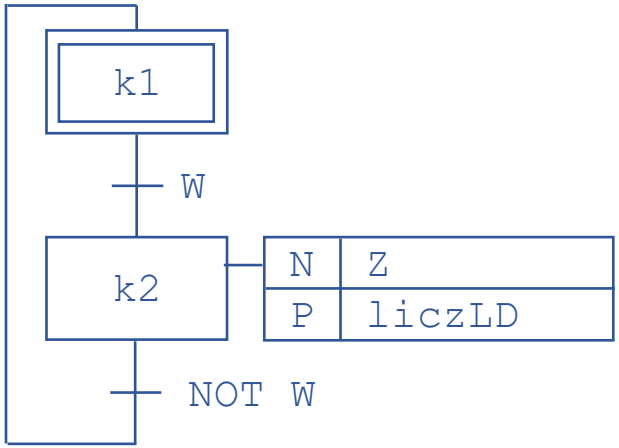

```
ACTION liczST
  ile := ile+1;
END_ACTION
```

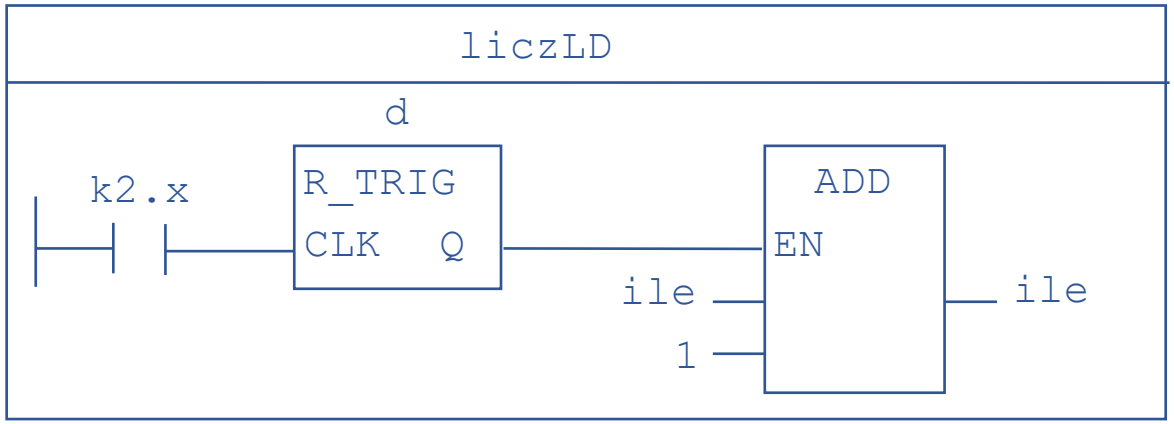




Sieć SFC – Przykład 3

Przycisk `W` włącza i wyłączą urządzenie `Z`. Włączenia sią zliczane w zmiennej `ile`.





Metoda SFC – Sekwencja pojedyncza

Sekwencja pojedyncza

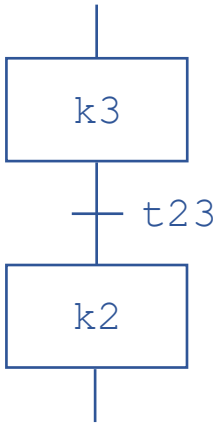
Sekwencja pojedyncza to najprostsza sekwencja (połączenie kroku i tranzycji), w której po spełnieniu warunku przejścia w tranzycji za krokiem aktywowany jest krok następny.

jeżeli
- aktywny jest krok `k2`

i
- spełniony jest warunek przejścia `t23`

to

uaktywniany jest krok `k3`



Metoda SFC – Sekwencja wyboru

Sekwencja wyboru - rozbieżność

Jeżeli po wykonaniu określonego kroku zachodzi potrzeba realizacji jednej z kilku możliwych sekwencji kroków to po symbolu kroku należy na grafie narysować pojedynczą linię poziomą a pod nią zestaw tranzycji odpowiadających możliwym wyborom.

jeżeli
- aktywny jest krok `k2`

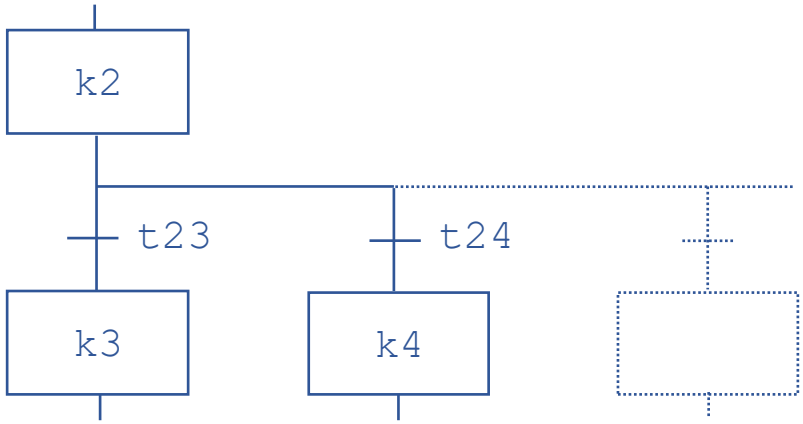
i
- spełniony jest warunek przejścia `t23`

to
uaktywniany jest krok `k3`

jeżeli
- aktywny jest krok `k2`

i
- spełniony jest warunek przejścia `t24`

to
uaktywniany jest krok `k4`



Metoda SFC – Sekwencja wyboru

Sekwencja wyboru - zbieżność

Alternatywne gałęzie grafu można połączyć w jeden ciąg kroków wykorzystując symbol zbieżności rysowany, podobnie jak symbol rozbieżności, w postaci pojedynczej linii poziomej.

jeżeli
- aktywny jest krok `k3`

i
- spełniony jest warunek przejścia `t36`

to

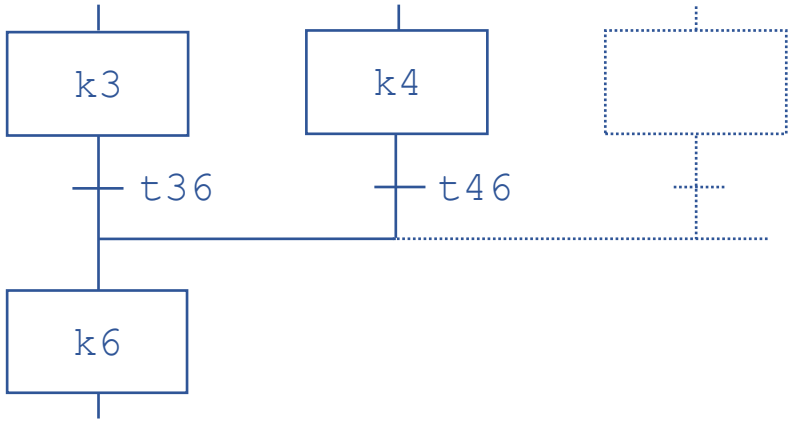
uaktywniany jest krok `k6`

jeżeli
- aktywny jest krok `k4`

i
- spełniony jest warunek przejścia `t46`

to

uaktywniany jest krok `k6`



Metoda SFC – Sekwencja współbieżności

Sekwencja współbieżności – rozbieżność

Jeżeli po wykonaniu określonego kroku zachodzi potrzeba jednoczesnej realizacji kilku sekwencji kroków to po symbolu tranzycji związanej z tym krokiem należy na grafie narysować podwójną linię poziomą a pod nią zestaw odpowiednich kroków.

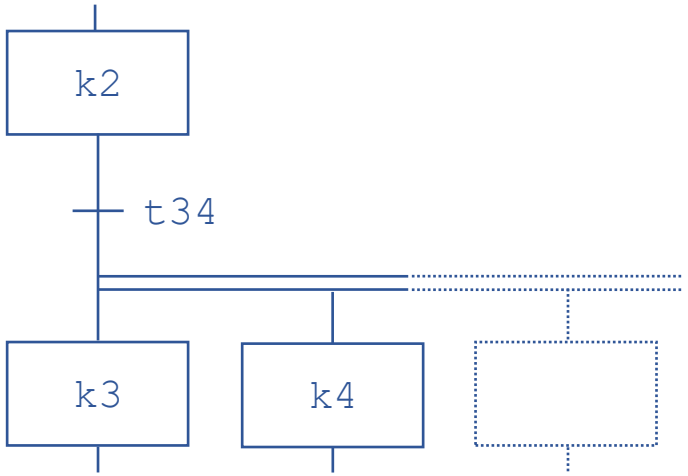
jeżeli
- aktywny jest krok `k2`

i
- spełniony jest warunek przejścia `t34`

to

uaktywniane są kroki `k3`, `k4`, …

przetwarzanie dalszych kroków w gałęziach równoległych odbywa się niezależnie



Metoda SFC – Sekwencja współbieżności

Sekwencja współbieżności – zbieżność

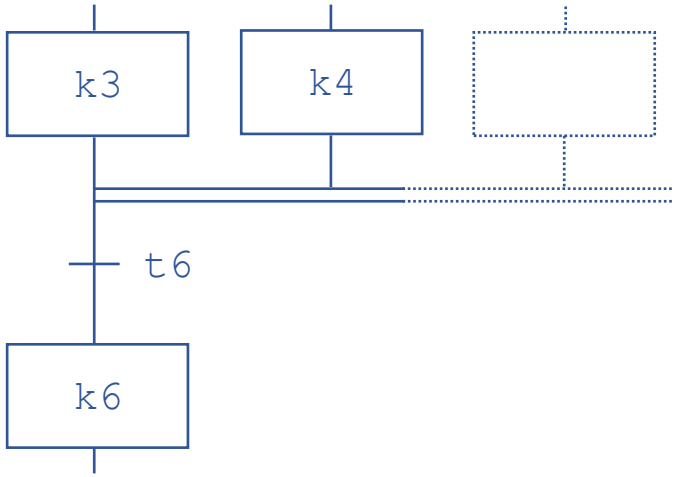
Równoległe gałęzie grafu można połączyć w jeden ciąg kroków używając symbolu zbieżności rysowanego w postaci podwójnej linii poziomej. Połączenie sprowadza się do synchronizacji operacji wykonywanych w krokach współbieżnych: kroki w gałęziach równoległych oczekują na spełnienie warunku tranzycji znajdującej pod symbolem zbieżności a po jego spełnieniu tracą aktywność na rzecz kroku znajdującego się za tranzycją.

jeżeli
- aktywne są kroki `k3`, `k4`, …

i
- spełniony jest warunek przejścia `t6`

to

uaktywniany jest krok `k6`



Metoda SFC – Pomijanie sekwencji kroków

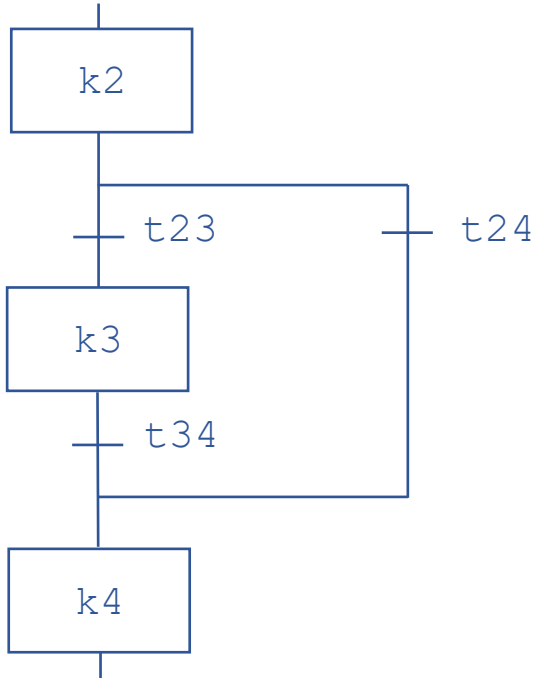
Pomijanie sekwencji kroków

Jeżeli po wykonaniu określonego kroku, przy spełnieniu określonych warunków, zachodzi potrzeba pominięcia sekwencji kroków to kroki te można pominąć wykorzystując pustą gałąź alternatywną.

jeżeli
- aktywny jest kroki `k2`

i
- spełniony jest warunek przejścia `t24`

to
uaktywniany jest krok `k4`
(krok `k3` jest pomijany)



Metoda SFC – Pętle

Pętle

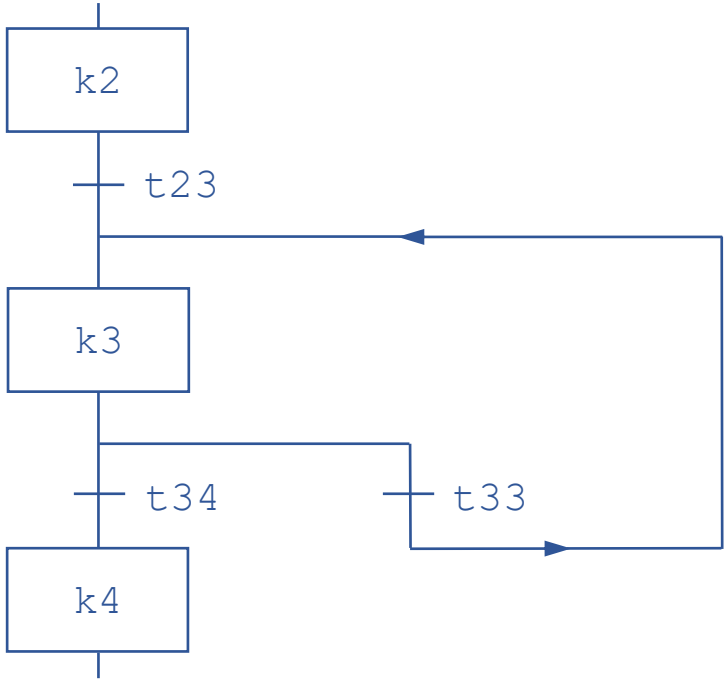
Pętlami nazywane są sekwencje wyboru, w których są gałęzie prowadzące do kroków położonych wyżej. W celu zwiększenia przejrzystości grafu, dopuszcza się w takim przypadku umieszczanie na połączeniach grotów wskazujących kierunek połączeń.

jeżeli
- aktywny jest krok `k3`

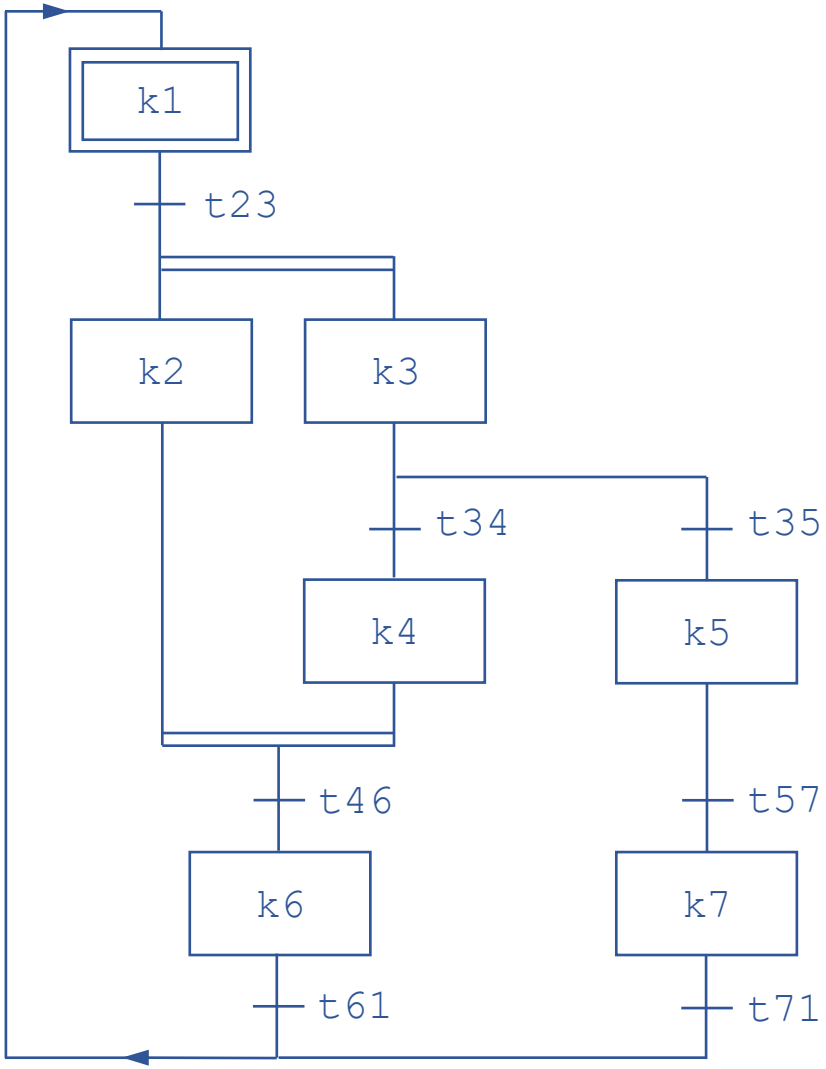
i
- spełniony jest warunek przejścia `t33`

to

ponownie uaktywniany jest krok `k3`



Metoda SFC – Błędy



po uaktywnieniu kroków:

k2 i k3

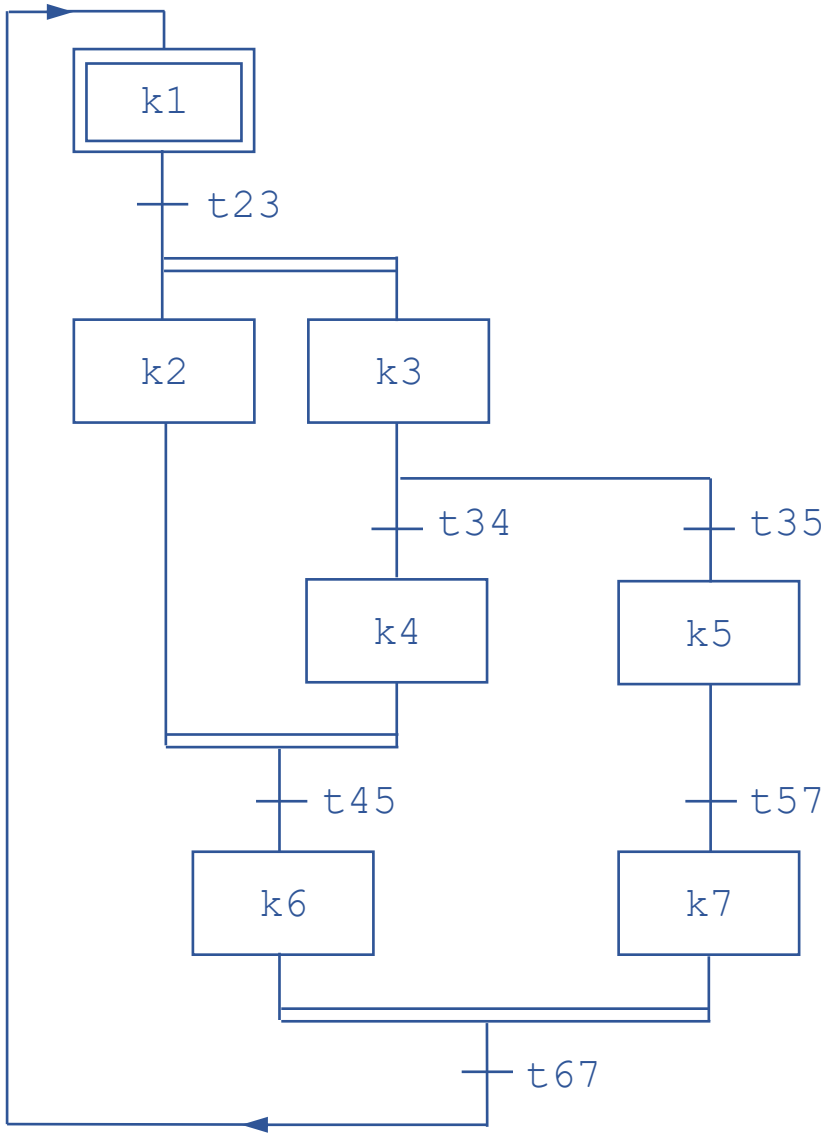
jeśli spełniony będzie warunek t35 to kolejno uaktywnione zostaną kroki:

k5, k7, k1

i jeśli spełniony będzie warunek t23 to krok k2 zostanie ponownie uaktywniony

(krok k2 oczekuje ciągle na spełnienie warunku t46, krok k2 może być uaktywniany wielokrotnie)

Metoda SFC – Błędy



po uaktywnieniu kroków:

k2 i k3

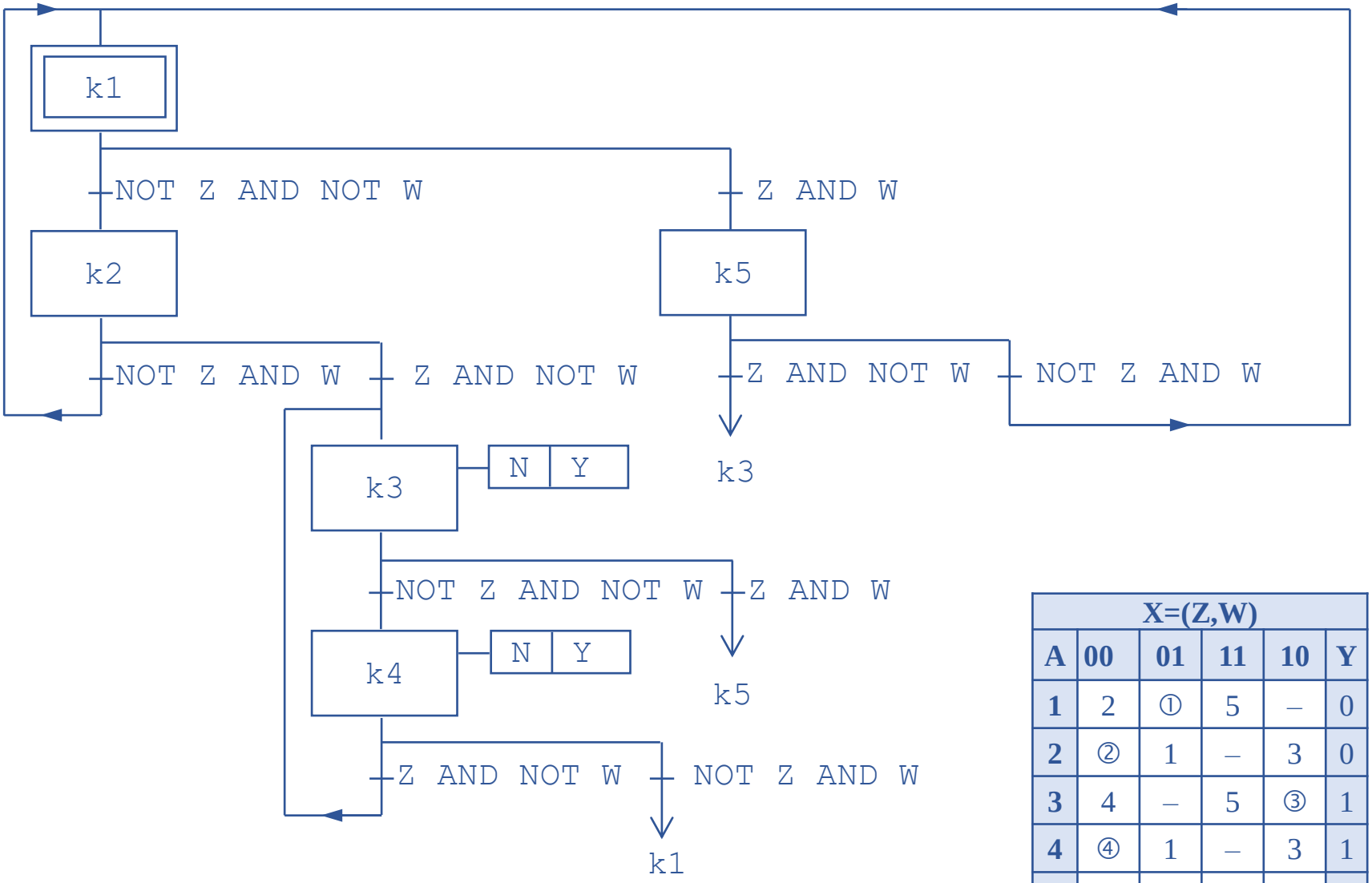
jeśli spełniony będzie warunek t35 to kolejno uaktywnione zostaną kroki:

k5, k7,

warunek przejścia tranzycji t67 nigdy nie będzie sprawdzony ponieważ nie jest możliwe aktywowanie kroku k6

(krok k7 zyska aktywność i nie odda jej żadnemu krokowi programu)

Metoda SFC – przykład 4



X=(Z,W)					
A	**00**	**01**	**11**	**10**	**Y**
1	2	①	5	–	0
2	②	1	–	3	0
3	4	–	5	③	1
4	④	1	–	3	1
5	–	1	⑤	3	0

Przycisk `Z` załącza a przycisk `W` wyłączą urządzenie `Y`.

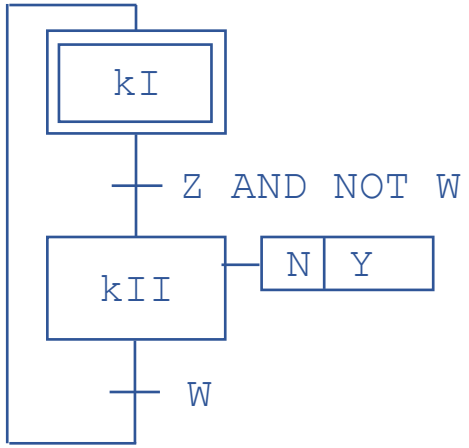
Metoda SFC – przykład 4

Otrzymana w wyniku łączenia odpowiadających sobie wierszy *pierwotna tablicy programu z metody Huffmana* tzn. *zredukowana tablica programu* pozwala na napisanie prostszego programu.

X=(Z,W)					
A	**00**	**01**	**11**	**10**	**Y**
1	2	①	5	–	0
2	②	1	–	3	0
3	4	–	5	③	1
4	④	1	–	3	1
5	–	1	⑤	3	0

X=(Z,W)					
A	**00**	**01**	**11**	**10**	**Y**
1,2,5	②	①	⑤	3	0
3,4	④	1	5	③	1

X=(Z,W)					
A	**00**	**01**	**11**	**10**	**Y**
I	(I)	(I)	(I)	II	0
II	(II)	I	I	(II)	1



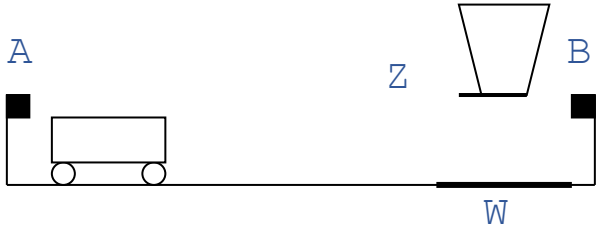
Metoda SFC – przykład 5

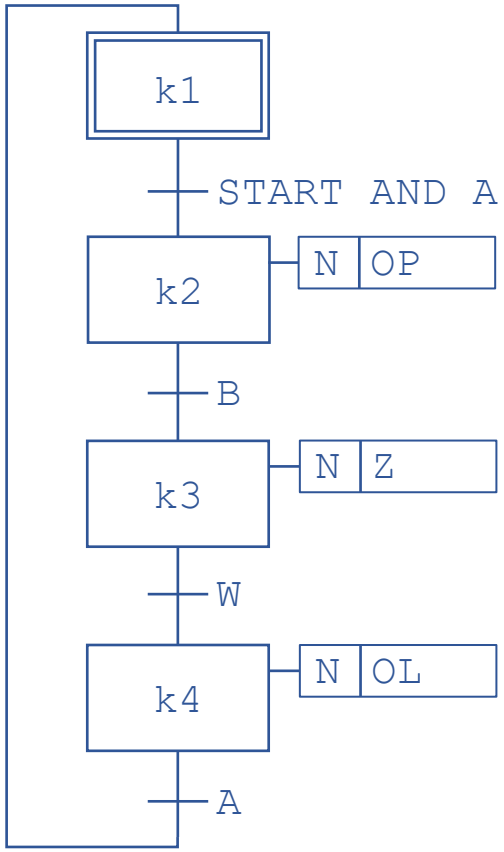
Program steruje procesem załadunku kontrolując:

- obroty silnika wózka:
  `OP=1` załącza obroty w prawo,
  `OL=1` załącza obroty w lewo,
- położenie klapy zamykającej silos:
  `Z=1` otwiera klapę, `Z=0` zamyka klapę.

Przebieg procesu:

- wózek po naciśnięciu przycisku `START (START=1)` podjeżdża do stanowiska załadunku (proces może rozpocząć się tylko gdy wózek znajduje się w pozycji początkowej zgłaszanej przez czujnik `A (A=1)`),
- załadunek rozpoczyna się automatycznie po osiągnięciu przez wózek stanowiska załadunku (zgłaszane przez czujnik `B (B=1)`),
- po załadowaniu właściwej ilości materiału (kontrolowane czujnikiem `W (W=1)`) załadunek jest przerywany a wózek powraca do pozycji początkowej i tam oczekuje na rozładunek.





Metoda SFC – przykład 6

Program steruje pracą 4 zaworów `Z1, …, Z4` umożliwiając cykliczne napełnianie i opróżnianie zbiorników. Założenia:

- `Zi=1`, `Zi=0` – i-ty zawór otwarty i zamknięty,
- zawory `Z1` i `Z3` do napełniania, zawory `Z2` i `Z4` do opróżniania zbiornika1 i zbiornika2,
- stan początkowy: zbiorniki puste a zawory zamknięte.

Przebieg procesu:

- napełnianie zbiorników rozpoczyna się równocześnie po naciśnięciu przycisku `START (START=1)`.
- napełnianie zbiornika jest przerywane po jego całkowitym wypełnieniu (wypełnienie zbiorników sygnalizują czujniki `A1` i `A2 (A1=A2=1)`),
- wypełniony zbiornik jest automatycznie opróżniany,
- po całkowitym opróżnieniu zbiorników (sygnalizowane czujnikami `B1` i `B2 (B1=B2=0)`) zawory odpływu są zamykane a ponowne napełnianie może być rozpoczęte jeżeli przycisk `START` jest włączony.