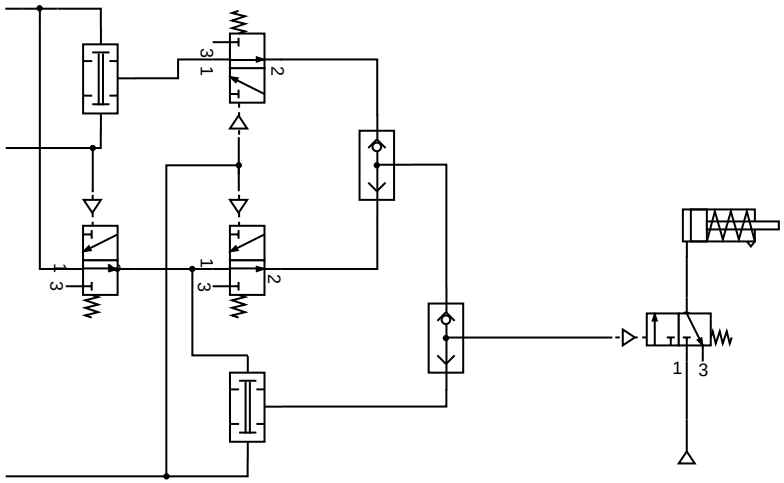
Podstawy automatyki i elektrotechniki

Układy i funkcje logiczne



Sygnały w układach sterowania

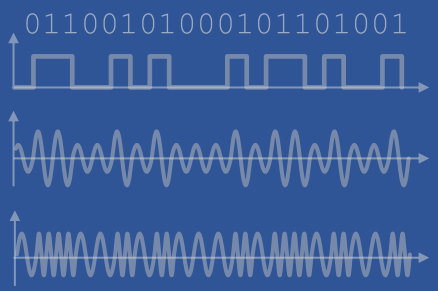
Układy logiczne, algebra Boole'a

Funkcje logiczne

Materiały

http://staff.uz.zgora.pl/gpajak

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



Sygnały

Sygnały

Sygnał

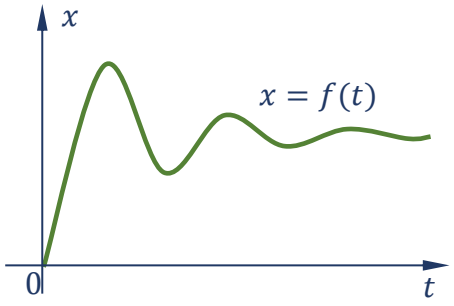
przebieg czasowy wielkości fizycznej, za pomocą której przekazywane są informacje; na sygnał składają się:

treść sygnału

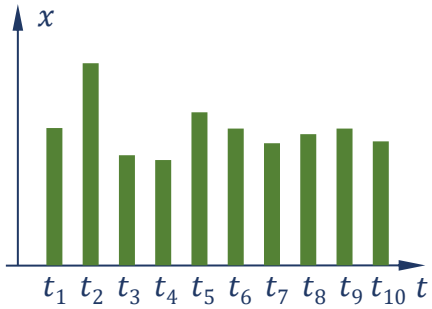
informacja (wiadomość) przenoszona przez sygnał;

nośnik sygnału

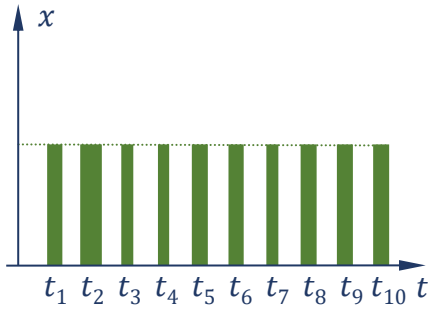
wielkość fizyczna (np. natężenie/napięcie prądu, ciśnienie cieczy/gazu itp.), której zmiany umożliwiają przekazanie określonych treści; do przekazywania informacji wykorzystuje się np. wartość amplitudy, szerokość impulsów, częstotliwość nośnika; w trakcie przesyłania nośnik może być zmieniany (np. przetwornik elektropneumatyczny przetwarza elektryczny sygnał prądowy na sygnał pneumatyczny).



treść sygnału: amplituda



treść sygnału: amplituda



treść sygnału: szerokość impulsów

Klasyfikacja sygnałów

Sygnały analogowe

sygnały ciągłe w czasie o wartościach ciągłych; większość sygnałów pochodzących z otaczającej rzeczywistości (np. temperatura otoczenia, natężenie prądu).

Sygnały kwantowane

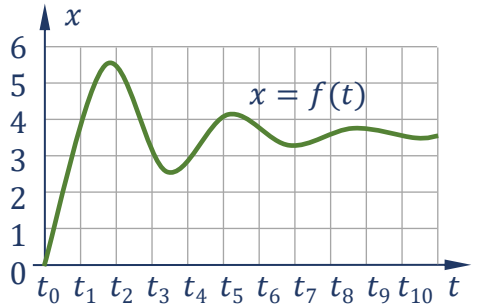
sygnały ciągłe w czasie o wartościach dyskretnych (np. łącznik pozwala na przepływ prądu w obwodzie lub nie).

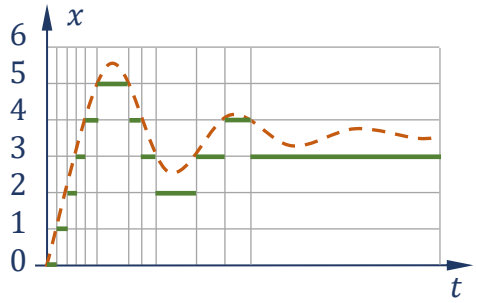
Sygnały próbkowane

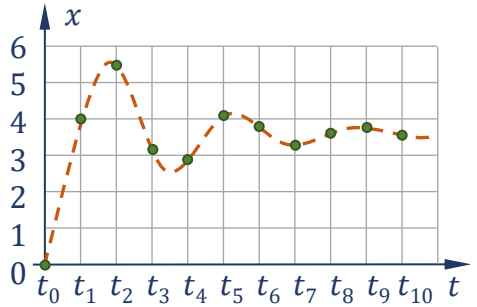
sygnały dyskretne w czasie o wartościach ciągłych (np. ciąg okresowych odczytów miernika analogowego – wskazania miernika są sygnałami analogowymi, odczytywane okresowo stają się sygnałami próbkowanymi).

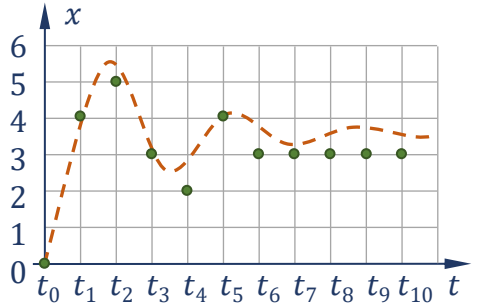
Sygnały cyfrowe

sygnały dyskretne w czasie o wartościach dyskretnych (np. ciąg okresowych odczytów miernika cyfrowego)









Przetwarzanie analogowo-cyfrowe

Przetwornik analogowo-cyfrowy A/C (ang. A/D lub ADC)

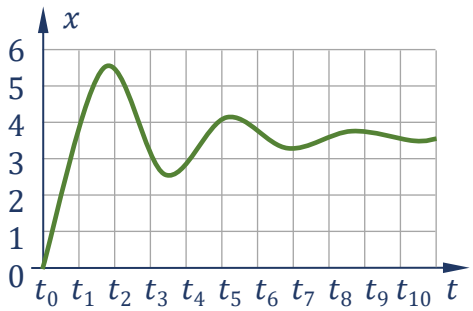
układ służący do zamiany sygnału analogowego na cyfrowy, zamiana przebiega w trzech etapach: próbkowanie, kwantowanie i kodowanie (kodowanie to proces polegający na przyporządkowaniu określonej informacji ustalonego zestawu symboli, w tym przypadku wartości liczbowej odpowiadającej poziomowi kwantowania

Przetwornik cyfrowo-analogowy C/A (ang. D/A lub DAC)

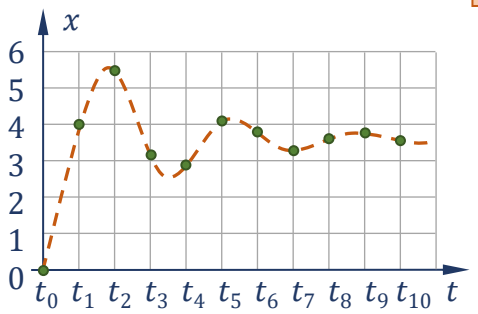
układ przetwarzający sygnał cyfrowy na równoważny mu sygnał analogowy

próbkowanie

kwantowanie

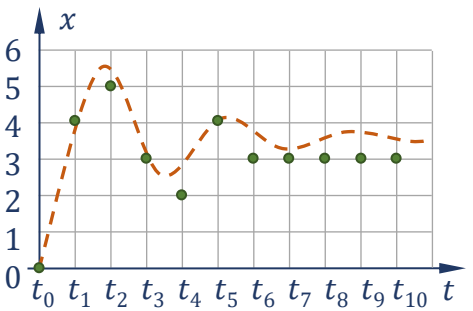


sygnał analogowy $x = f(t)$



sygnał dyskretny w czasie

$x_d = \{0;\ 4{,}1;\ 5{,}5;\ 3{,}2;\ 2{,}9;\ \ldots\}$



sygnał cyfrowy

$x_c = \{0;\ 4;\ 5;\ 3;\ 2;\ \ldots\}$

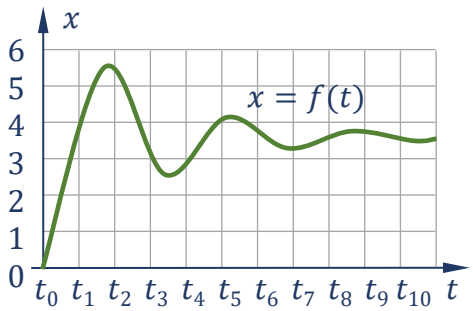
Przetwarzanie analogowo-cyfrowe

Próbkowanie

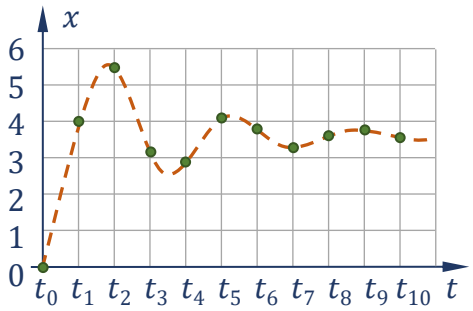
proces polegający na pobieraniu wartości sygnału ciągłego w określonych odstępach czasu, wynikiem próbkowania jest sygnał dyskretny w czasie

Kwantowanie

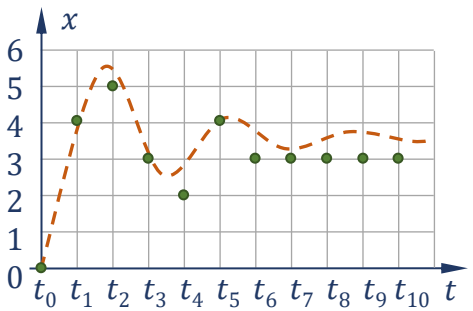
proces polegający na przekształcaniu ciągłego zbioru wartości sygnału w skończony zbiór wartości, zbiór wartości sygnału dzielony jest na tzw. przedziały kwantowania (zwykle stosuje się podział równomierny), następnie z każdego przedziału wybierana jest jedna wartość reprezentująca ten przedział tzw. poziom kwantowania, kwantowanie polega na przyporządkowaniu każdej wartości sygnału odpowiedniego poziomu kwantowania



sygnał analogowy



sygnał po próbkowaniu



sygnał po próbkowaniu i kwantowaniu

Układy logiczne

Algebra Boole'a

Układy logiczne, algebra Boole'a

Układ logiczny

szczególny przypadek układu cyfrowego, w którym sygnały przyjmują dwa poziomy 0 i 1 (fałsz, prawda); działanie oparte na dwuelementowej algebrze Boole'a; najczęściej wykorzystywany typ układów cyfrowych.

Dwuelementowa algebra Boole'a

uogólnienie rachunku zdań; definiowana jako układ:

$$B = \left(\{0,1\},\ +,\ \cdot,\ \bar{},\ 0,\ 1\right)$$

gdzie: $\{0,1\}$ – zbiór elementów algebry; $0, 1$ – stałe algebry; $+\ \cdot\ \bar{}$ – operacje algebry: dwuargumentowa alternatywa (OR) i koniunkcja (AND), jednoargumentowa negacja.(NOT)

alternatywa (OR)		
a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	1

koniunkcja (AND)		
a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

negacja (NOT)	
a	$\bar{a}$
0	1
1	0

Uwaga: symbol koniunkcji zazwyczaj jest pomijany, stąd $ab = a \cdot b$

Aksjomaty algebry Boole'a

Dla dowolnych $a, b, c \in B$ spełnione są następujące aksjomaty, które w pełni charakteryzują własności algebry Boole'a:

- Domknięcie

  $a + b \in B$ $\qquad$ $a \cdot b \in B$

- Przemienność

  $a + b = b + a$ $\qquad$ $a \cdot b = b \cdot a$

- Rozdzielność

  $a \cdot (b + c) = a \cdot b + a \cdot c$ $\qquad$ $a + b \cdot c = (a + b) \cdot (a + c)$

- Istnienie elementu neutralnego (identycznościowego)

  $a + 0 = a$ $\qquad$ $a \cdot 1 = a$

- Istnienie dopełnienia

  $a + \bar{a} = 1$ $\qquad$ $a \cdot \bar{a} = 0$

Prawa algebry Boole'a

- Prawo łączności

$$a + (b + c) = (a + b) + c \qquad a(b\,c) = (a\,b)c$$

- Prawo pochłaniania

$$a + a\,b = a \qquad a(a + b) = a$$

- Prawo sklejania

$$(a + b) \cdot (a + \bar{b}) = a \qquad a \cdot b + a \cdot \bar{b} = a$$

- Prawo idempotentności

$$a + a = a \qquad a\,a = a$$

- Prawo de Morgana (dla dwóch i wielu argumentów)

$$\overline{a + b} = \bar{a}\,\bar{b} \qquad \overline{ab} = \bar{a} + \bar{b}$$

$$\overline{a + b + c + d} = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \qquad \overline{a \cdot b \cdot c \cdot d} = \bar{a} + \bar{b} + \bar{c} + \bar{d}$$

- Prawa elementów neutralnych

$$a + 1 = 1 \qquad a \cdot 0 = 0$$

- Prawo podwójnej negacji $\qquad \bar{\bar{a}} = a$

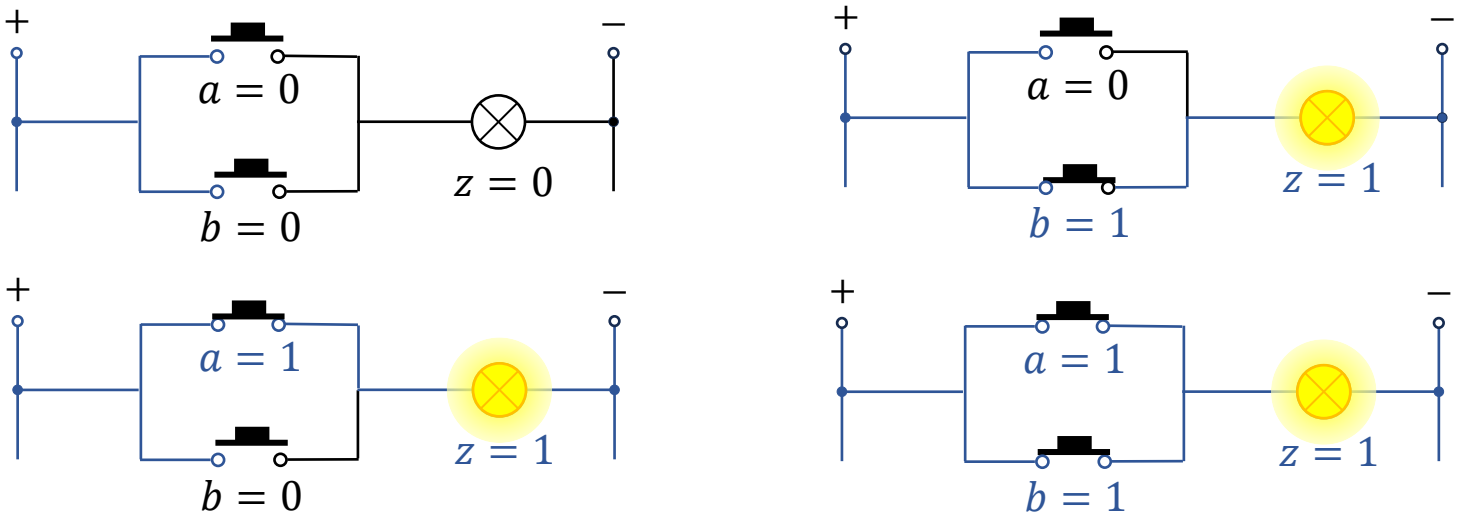
Operacje logiczne – alternatywa (OR)

$$z = a + b$$

Sygnał wyjściowy $z = 1$ zawsze gdy dowolny z sygnałów wejściowych jest równy 1 (tzn. $a = 1$ lub $b = 1$).

alternatywa (OR)		
a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Układ elektryczny
(dwa łączniki monostabilne NO)



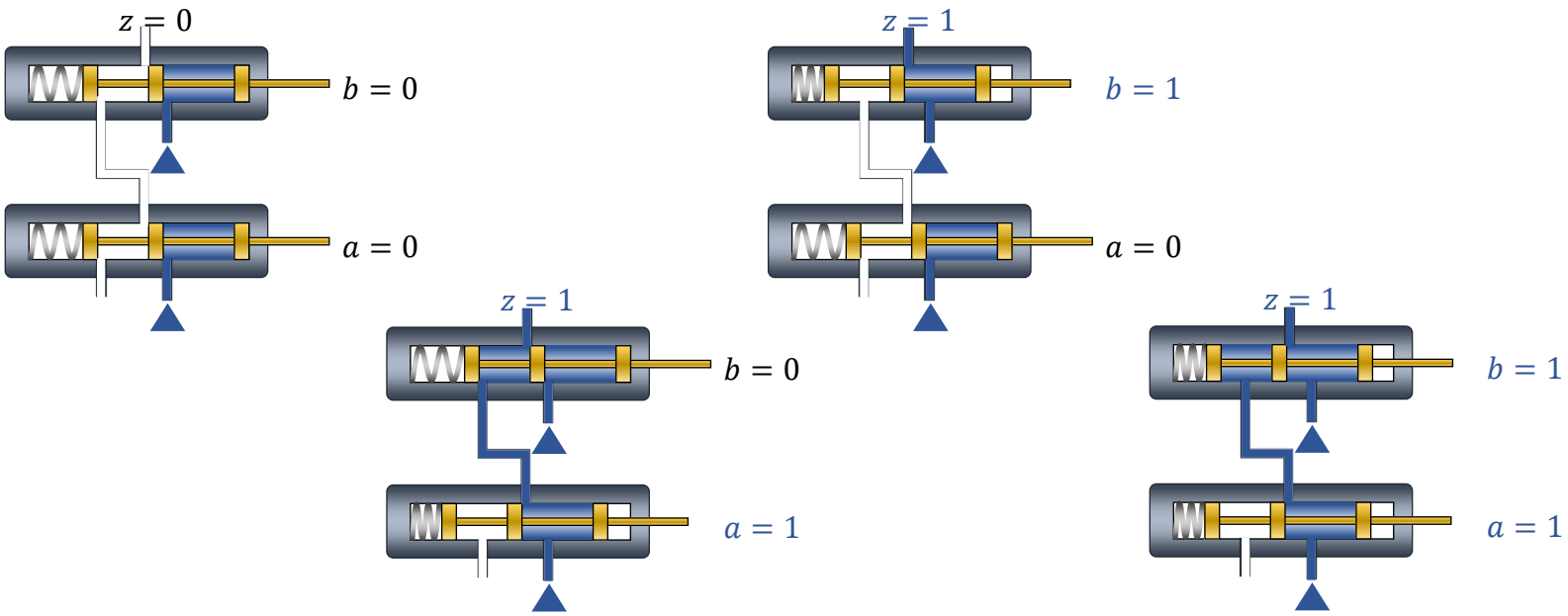
Operacje logiczne – alternatywa (OR)

$$z = a + b$$

Sygnał wyjściowy $z = 1$ zawsze gdy dowolny z sygnałów wejściowych jest równy 1 (tzn. $a = 1$ lub $b = 1$).

alternatywa (OR)		
a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Układ pneumatyczny (dwa zawory 3/2 NC)



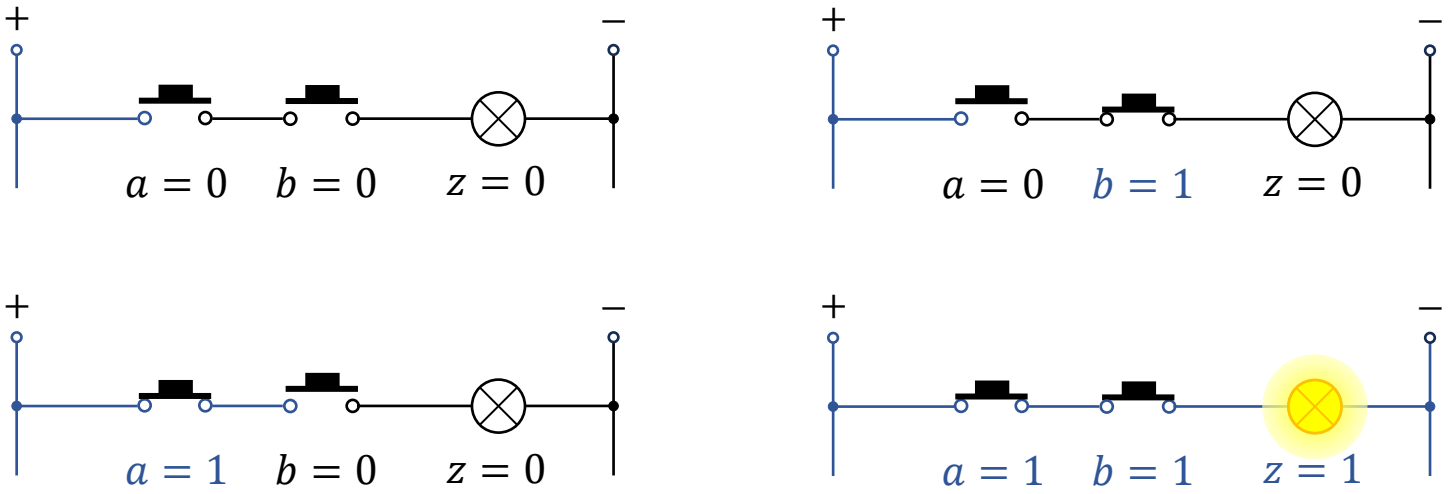
Operacje logiczne – koniunkcja (AND)

$$z = a \cdot b$$

Sygnał wyjściowy $z = 1$ tylko gdy obydwa sygnały wejściowe są równe 1 (tzn. $a = 1$ i $b = 1$).

koniunkcja (AND)		
a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

Układ elektryczny
(dwa łączniki monostabilne NO)



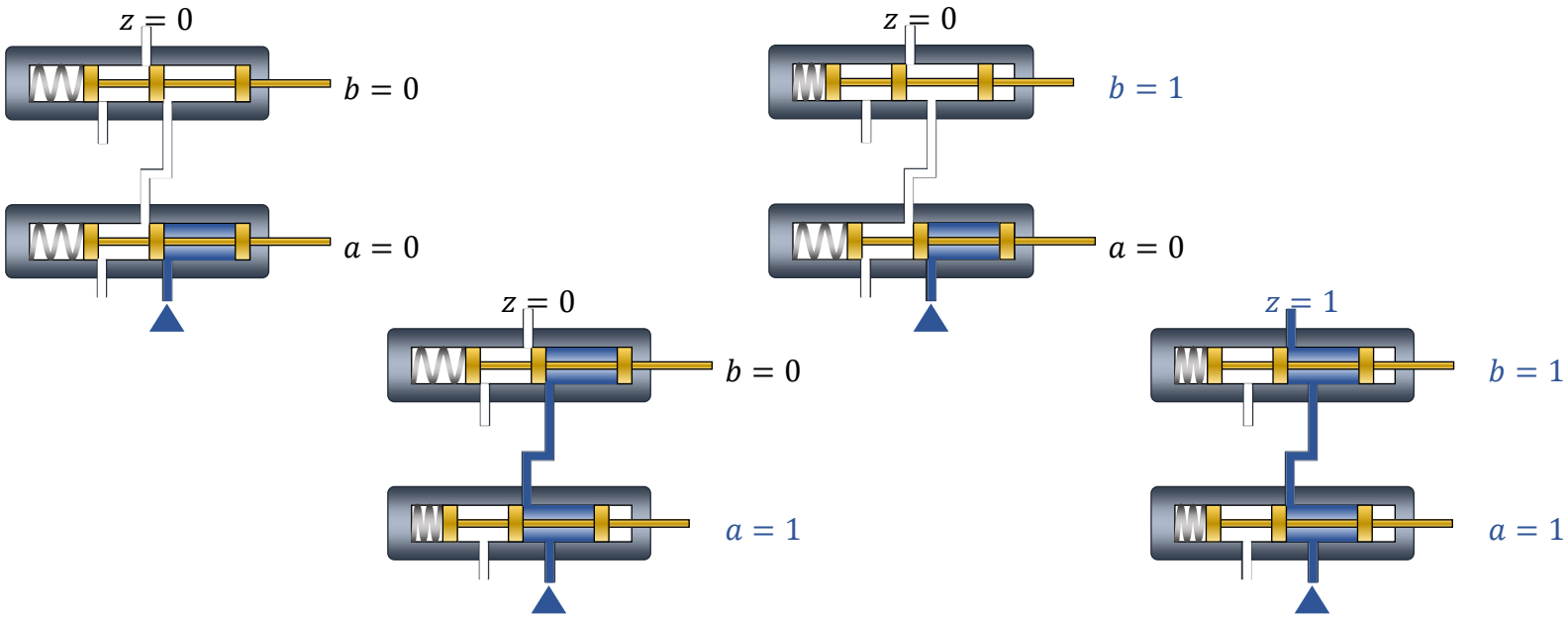
Operacje logiczne – koniunkcja (AND)

$$z = a \cdot b$$

Sygnał wyjściowy $z = 1$ tylko gdy obydwa sygnały wejściowe są równe 1 (tzn. $a = 1$ i $b = 1$).

koniunkcja (AND)		
a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	0

Układ pneumatyczny (dwa zawory 3/2 NC)



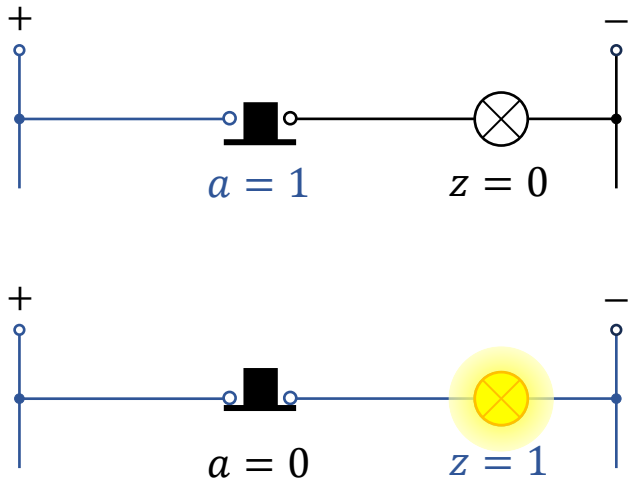
Operacje logiczne – negacja (NOT)

$$z = \bar{a}$$

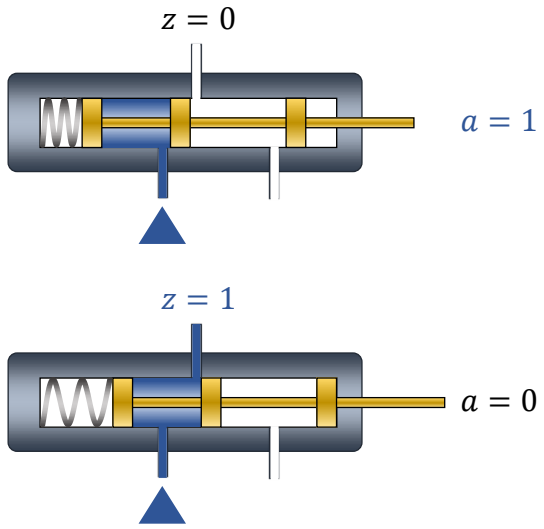
negacja (NOT)	
a	$\bar{a}$
0	1
1	0

Sygnał wyjściowy z jest przeciwieństwem sygnału wejściowego, tzn. $z = 1$ gdy $a = 0$ i $z = 0$ gdy $a = 1$.

Układ elektryczny (łącznik monostabilny NC)



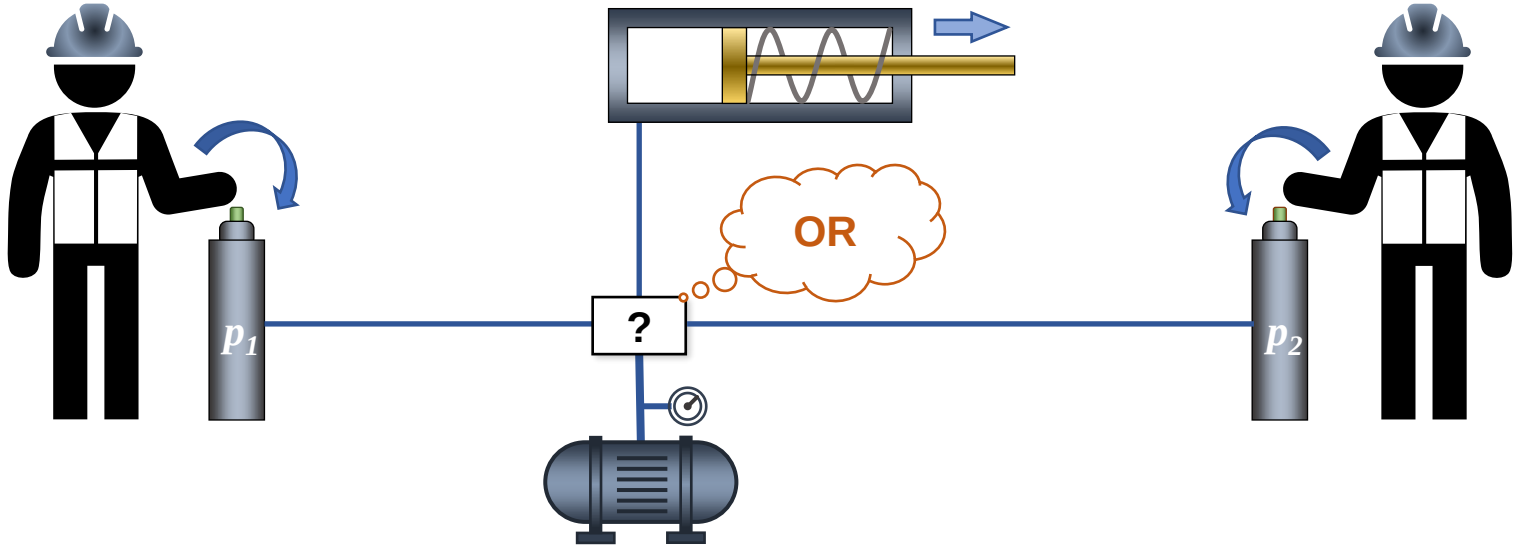
Układ pneumatyczny (zawór 3/2 NO)



Operacje logiczne – zastosowania

Sterowanie siłownikiem z dwóch miejsc

Układ sterowania składa się z dwóch przycisków p_1 i p_2. Siłownik można wysunąć naciskając jeden z nich lub obydwa jednocześnie.



Warunek uruchomienia: wciśnięcie p_1 lub p_2

Warunek zatrzymania: zwolnienie obydwu przycisków

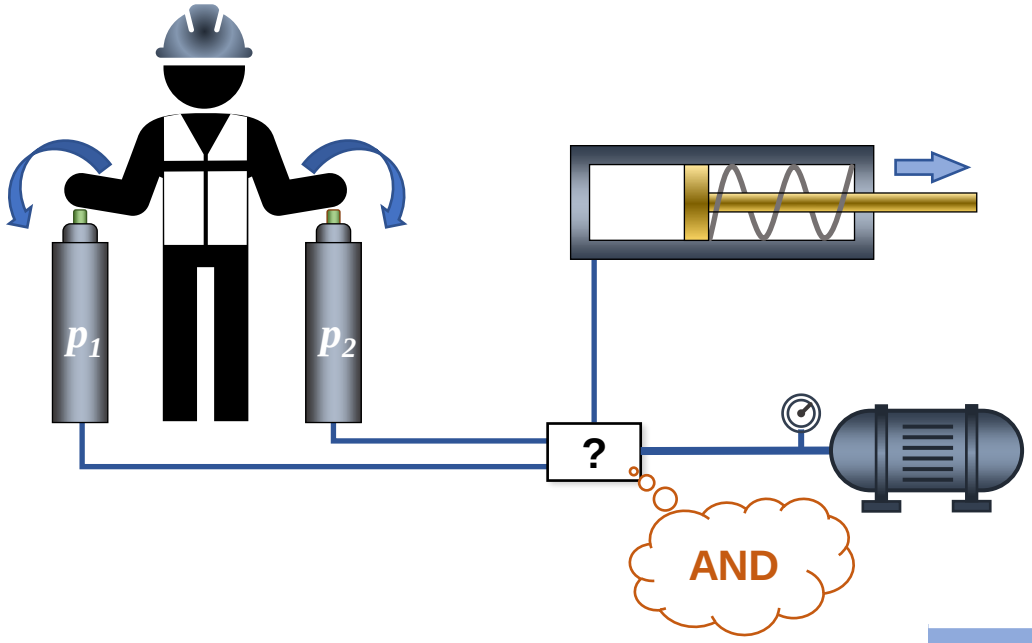
Realizacja: operacja alternatywy (OR)

p_1	p_2	$p_1 + p_2$
0	0	0
0	1	1
1	0	1
1	1	1

Operacje logiczne – zastosowania

Dwuręczne sterowanie siłownikiem

Układ sterowania składa się z dwóch przycisków p_1 i p_2. Siłownik można wysunąć naciskając jednocześnie obydwa przyciski.



Warunek uruchomienia: jednoczesne wciśnięcie p_1 i p_2

Warunek zatrzymania: zwolnienie dowolnego przycisku

Realizacja: operacja koniunkcji (AND)

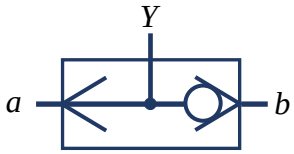
p_1	p_2	$p_1 \cdot p_2$
0	0	0
0	1	0
1	0	0
1	1	1

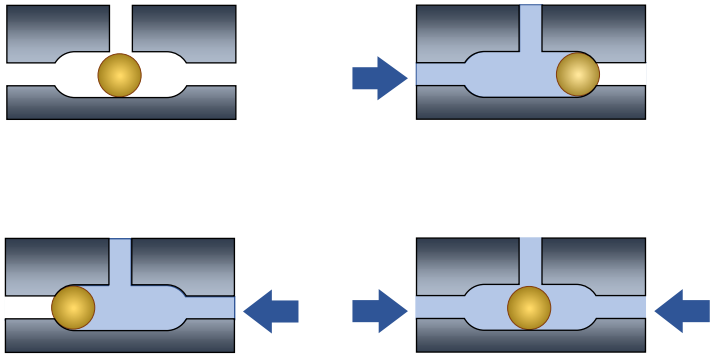
Zawory logiczne

Zawory logiczne sterują ciśnieniem czynnika roboczego (gazu) w układzie, mogą realizować funkcje logiczne (AND, OR) lub blok pamięci.

Zawór logiczny OR

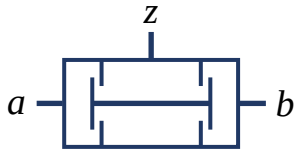
$$z = a + b$$

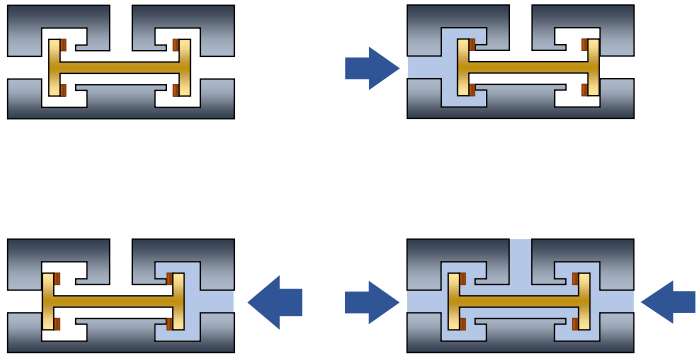




Zawór logiczny AND

$$z = a \cdot b$$





$$z_1 = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$$

$$z_2 = (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$$

Funkcje logiczne

Funkcje boolowskie

Funkcją boolowska (logiczna) n zmiennych $f(x_1, x_2, \cdots, x_n)$

odwzorowanie postaci: $f: \boldsymbol{X} \to \boldsymbol{B}$,

gdzie: $\boldsymbol{X} \subseteq \boldsymbol{B}^n = \boldsymbol{B} \times \boldsymbol{B} \times \cdots \times \boldsymbol{B}$ jest dziedziną funkcji logicznej (zbiór n-elementowych ciągów zerojedynkowych); $\boldsymbol{B} = \{0, 1\}$ jest zbiorem wartości funkcji.

Uwaga: Funkcja boolowska przypisuje elementom z dziedziny (każdemu ciągowi zerojedynkowemu) wartość 0 lub 1, za jej pomocą można opisać działanie dowolnego układu logicznego.

Funkcja zupełna

funkcja, której wartości są określone dla wszystkich możliwych ciągów n-elementowych (wszystkich elementów z dziedziny), dla funkcji n-zmiennych 2^n elementów.

Funkcja niezupełna (nie w pełni określona)

funkcja, której wartości są określone tylko dla niektórych ze wszystkich możliwych ciągów n-elementowych.

Uwaga: istnienie funkcji nie w pełni określonych wynika wyłącznie z praktycznych zastosowań funkcji boolowskich do opisu działania układów logicznych; matematycznie można określić wartość funkcji logicznej dla dowolnych argumentów.

Cyfrowe układy automatyki, p. 2.2, 3.4

Reprezentacja funkcji boolowskich

Opis słowny

przedstawia działanie funkcji w formie opisu określającego jakie wartości są przyporządkowywane poszczególnym elementom z dziedziny.

Przykład: funkcja zmiennych a i b przyjmuje wartość 1 wtedy i tylko wtedy obydwa argumenty są różne (tzn. $a \neq b$).

Tabela prawdy

przedstawia działanie funkcji logicznej w postaci tabeli określającej wartość funkcji dla każdego ciągu z dziedziny; w przypadku funkcji n zmiennych pierwsze n kolumn tabeli zawiera wartości zmiennych, a kolumna $n+1$ odpowiadające im wartości funkcji.

Przykład

a	b	$f(a,b)$
0	0	0
0	1	1
1	0	1
1	1	0

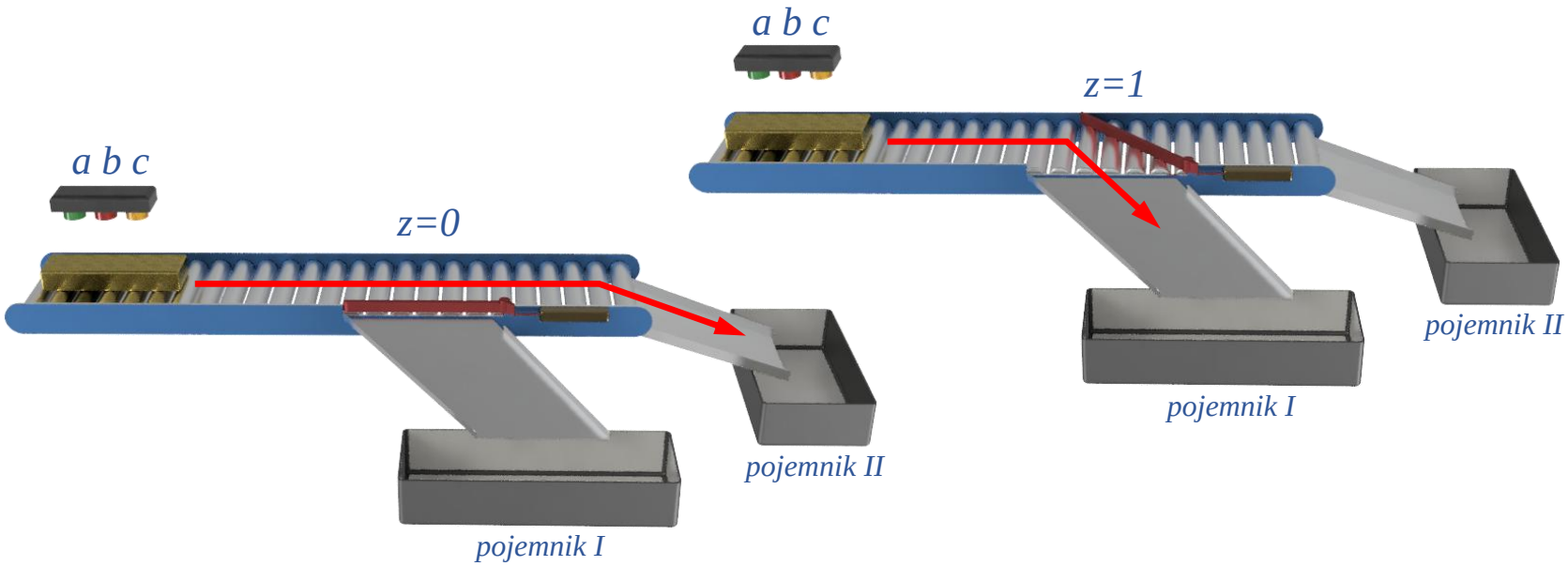
Wyrażenie boolowskie (logiczne)

przedstawia funkcję zapisaną za pomocą operatorów algebry Boole'a.

Przykład: $f(a,b) = \bar{a}b + a\bar{b}$

Przykład 1. – Jednowyjściowy układ logiczny

Przenośnik transportuje detale wytwarzane w pewnym procesie produkcyjnym. Na stanowisku kontroli, przy pomocy odpowiednich czujników, badane są trzy cechy detalu (**a**, **b**, **c**). Każdy z czujników sygnalizuje badaną cechę wartością **1** – prawidłowa lub **0** – nieprawidłowa. Na podstawie wyniku badania zwrotnica kieruje detale do jednego z dwóch pojemników. Jeżeli prawidłowa jest cecha a i co najwyżej jedna z dwóch pozostałych detal trafia do *pojemnika I* w pozostałych przypadkach do *pojemnika II*. Należy zaprojektować urządzenie sterujące pracą siłownika przestawiającego zwrotnicę w taki sposób, żeby kierować detal do odpowiedniego pojemnika. Sygnał $z = 1$ (siłownik wysunięty) kieruje detal do *pojemnika I*, $z = 0$ (siłownik wsunięty) do *pojemnika II*.



Konstrukcja tabeli prawdy

Dwa podejścia do konstrukcji tabeli prawdy na podstawie opisu układu sterującego:

1. Indywidualna analiza każdej możliwej kombinacji sygnałów wejściowych (w układach logicznych n zmienianych istnieje 2^n kombinacji) i identyfikacja sygnału wyjściowego.
2. Identyfikacja wszystkich stanów wejść odpowiadających konkretnej wartości (np. 1) sygnału wyjściowego.

Przykład. Opis pracy sortownika

Podejście pierwsze: 8 kombinacji (trzy czujniki, $2^3 = 8$).

$000 \rightarrow$ trzy nieprawidłowe $\rightarrow$ pojemnik II $\rightarrow z = 0$

$001 \rightarrow$ nieprawidłowe a, b $\rightarrow$ pojemnik II $\rightarrow z = 0$

$100 \rightarrow$ prawidłowa tylko a $\rightarrow$ pojemnik I $\rightarrow z = 1$

$101 \rightarrow$ prawidłowe a *i* c $\rightarrow$ pojemnik I $\rightarrow z = 1$

$111 \rightarrow$ trzy prawidłowe $\rightarrow$ pojemnik II $\rightarrow z = 1$ …

a	b	c	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Podejście drugie: $z = 1$ gdy prawidłowa jest cecha a i co najwyżej jedna z dwóch pozostałych, tzn. 100, 101, 110 w pozostałych przypadkach $z = 0$.

Kanoniczna postać dysjunkcyjna funkcji boolowskiej

Kanoniczna postać dysjunkcyjna

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f(1,1,\dots,1)x_1x_2\dots x_n + f(0,1,\dots,1)\bar{x}_1x_2\dots x_n + \\ &+ f(1,0,\dots,1)x_1\bar{x}_2\dots x_n + f(1,1,\dots,0)x_1x_2\dots\bar{x}_n + \\ &+ \dots + f(0,0,\dots,0)\bar{x}_1\bar{x}_2\dots\bar{x}_n, \end{aligned}$$

gdzie koniunkcje wszystkich zmiennych postaci $x_1x_2\dots x_n$, $\bar{x}_1x_2\dots x_n$ itd. nazywane są iloczynami pełnymi lub minitermami.

Własności

- składniki są koniunkcją wartości funkcji o stałych argumentach oraz iloczynów pełnych (miniterm),
- zmienna o wartości 0 w iloczynie pełnym występuje z negacją, zmienna o wartości 1 bez negacji,
- $a \cdot 0 = 0$, więc składniki, w których $f(x_1, x_2, \dots, x_n) = 0$ mogą być pominięte, ponieważ zero nie zmienia wyniku alternatywy ($a + 0 = a$).

Kanoniczna postać koniunkcyjna funkcji boolowskiej

Kanoniczna postać koniunkcyjna

$$f(x_1, x_2, \dots, x_n) = (f(0,0,\dots,0) + x_1 + x_2 + \cdots + x_n) \cdot (f(1,0,\dots,0) + \bar{x}_1 + x_2 + \cdots + x_n) \cdot \cdot (f(0,1,\dots,0) + x_1 + \bar{x}_2 + \cdots + x_n) \cdot (f(0,0,\dots,1) + x_1 + x_2 + \cdots + \bar{x}_n) \cdot \cdot \cdots \cdot (f(1,1,\dots,1) + \bar{x}_1 + \bar{x}_2 + \cdots + \bar{x}_n),$$

gdzie alternatywy wszystkich zmiennych postaci $x_1 + x_2 + \cdots + x_n$, $\bar{x}_1 + x_2 + \cdots + x_n$ itd. nazywane są sumami pełnymi lub maxtermami.

Własności

- składniki są alternatywą wartości funkcji o stałych argumentach oraz sum pełnych (maxterm),
- zmienna o wartości 1 w sumie pełnej występuje z negacją, zmienna o wartości 0 bez negacji,
- $a + 1 = 1$, więc składniki, w których $f(x_1, x_2, \dots, x_n) = 1$ mogą być pominięte, ponieważ jedynka nie zmienia wyniku koniunkcji ($a \cdot 1 = a$).

Zapis funkcji w postaci kanonicznej

Postać dysjunkcyjna

- wybierz wiersze tabeli prawdy, w których funkcja ma wartość 1,
- utwórz iloczyny pełne negując zmienne o wartości 0,
- zapisz alternatywę iloczynów pełnych.

Postaci koniunkcyjna

- wybierz wiersze tabeli prawdy, w których funkcja ma wartość 0,
- utwórz sumy pełne negując zmienne o wartości 1,
- zapisz koniunkcję sum pełnych.

Uwaga: obydwie formy funkcji są równoważne (dają ten sam wynik).

a	b	c	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Kanoniczna postać koniunkcyjna

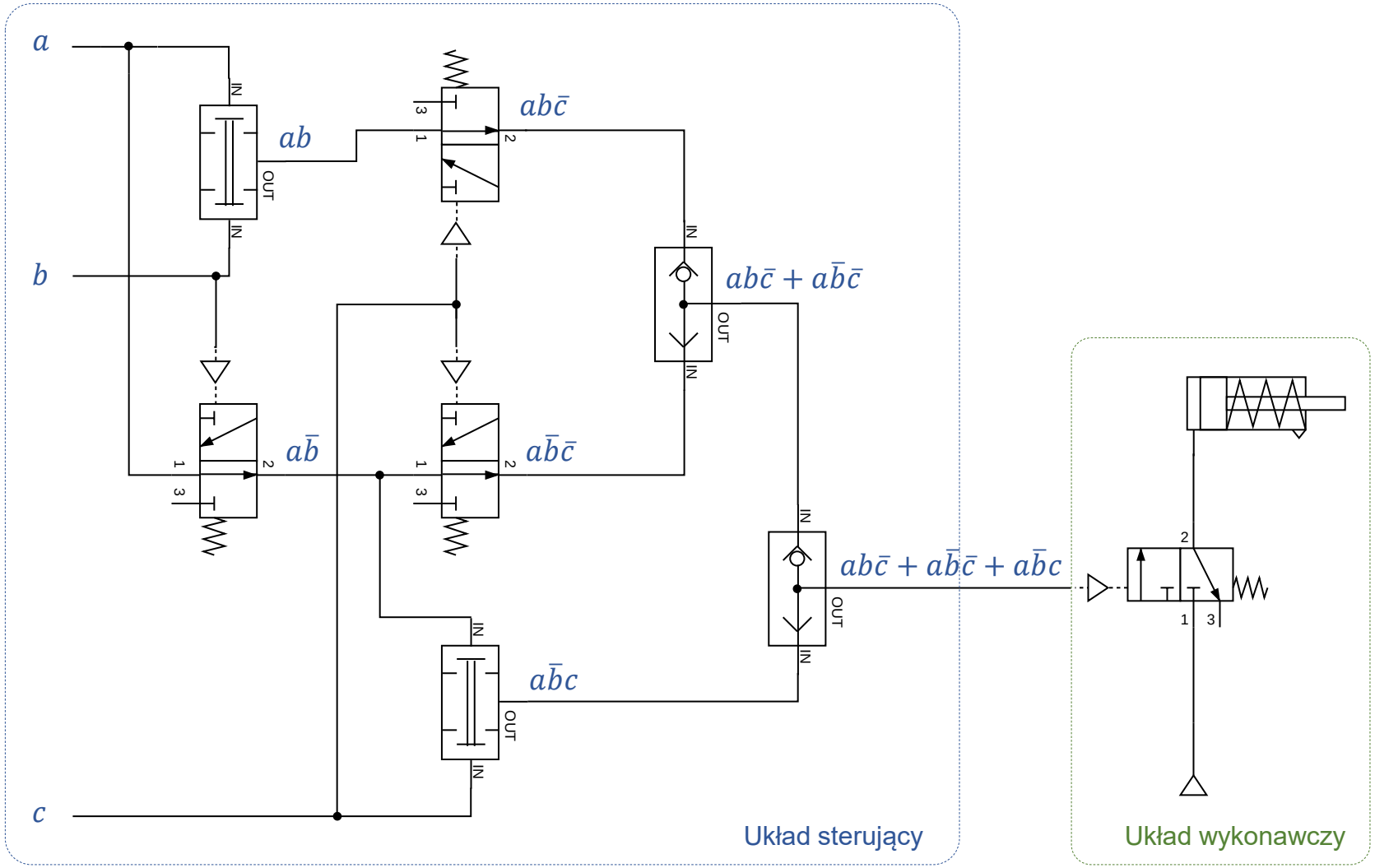
$$z = f(a,b,c) = (a+b+c)\cdot(a+b+\overline{c})\cdot(a+\overline{b}+c)\cdot(a+\overline{b}+\overline{c})\cdot(\overline{a}+\overline{b}+\overline{c})$$

Kanoniczna postać dysjunkcyjna

$$z = f(a,b,c) = a\overline{b}\overline{c} + a\overline{b}c + ab\overline{c}$$

Układ sterowania zwrotnicą

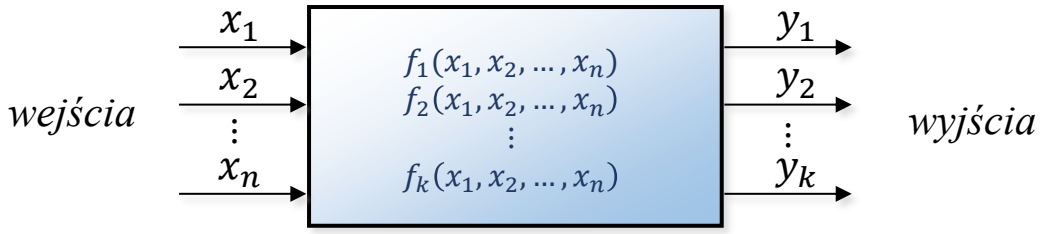
$$z = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$$



Układy wielowyjściowe

Układ wielowyjściowy

układ o n wejściach i k wyjściach, którego działanie opisuje układ k funkcji logicznych o n argumentach; każda z funkcji opisuje jedno z wyjść układu.



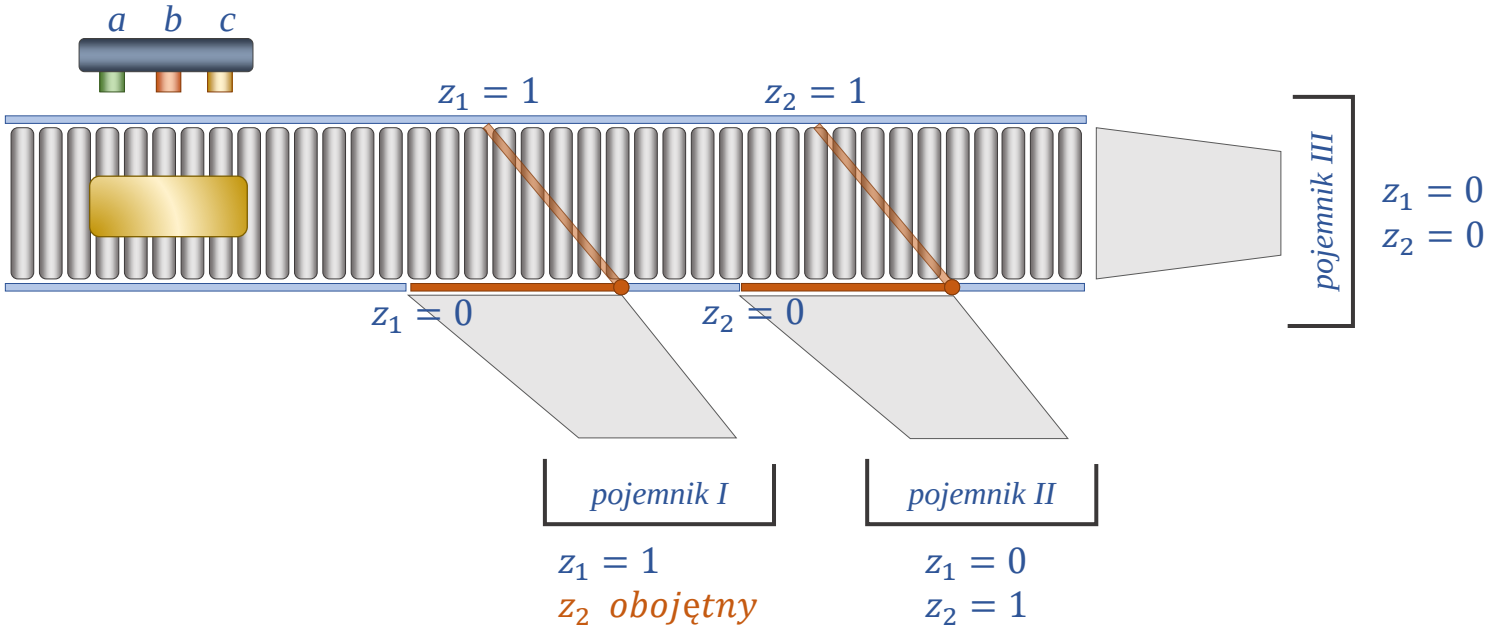
- Układ wielowyjściowy przypisuje każdej wartości z dziedziny (n-elementowemu ciągowi zerojedynkowemu) odpowiadającą mu wartość będącą ciągiem k-elementowym.
- Układy wielowyjściowe są używane w przypadku układów, które sterują pracą kilku urządzeń, których praca zależy od zbioru tych samych sygnałów wyjściowych.

Przykład 2 – wielowyjściowy układ nie w pełni określony

Na stanowisku kontroli, przy pomocy odpowiednich czujników, badane są trzy cechy detalu (a, b, c). Zależnie od wyników pomiaru detal powinien trafić do:

- *pojemnika I* gdy prawidłowa jest cecha a i co najwyżej jedna z dwóch pozostałych,
- *pojemnika II* gdy cecha a jest nieprawidłowa, b i c nie są jednocześnie prawidłowe,
- *pojemnika III* gdy prawidłowe są cechy b i c lub wszystkie.

Należy zaprojektować urządzenie sterujące pracą dwóch siłowników przestawiających zwrotnice z_1 i z_2 w taki sposób, żeby kierować detal do odpowiedniego pojemnika.



Przykład 2 – kanoniczna postać funkcji

Pojemniki

Pojemnik I : prawidłowa a i co najwyżej jedna z pozostałych, wejście: 100, 110, 101.

Pojemnik II : a nieprawidłowe, b i c nie są jednocześnie prawidłowe, wejście: 000, 001, 010.

Pojemnik III : prawidłowe są cechy b i c lub wszystkie, wejście: 011, 111.

Tabela prawdy

a	b	c	z_1	z_2
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	–
1	0	1	1	–
1	1	0	1	–
1	1	1	0	0

Kanoniczna postać koniunkcyjna

$$z_1 = = (a + b + c) \cdot (a + b + \bar{c}) \cdot (a + \bar{b} + c) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$$

$$z_2 = (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + \bar{c})$$

Kanoniczna postać dysjunkcyjna

$$z_1 = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$$

$$z_2 = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c}$$

Uwaga: tworząc kanoniczną postać funkcji nie w pełni określonej stany nieokreślone należy pominąć.

Minimalizacja funkcji logicznej

Minimalizacja funkcji logicznych

Celem minimalizacji funkcji logicznej jest doprowadzenie do postaci o możliwie najmniejszej liczbie operacji logicznych.

Metody minimalizacji:

- wykorzystanie praw algebry Boole'a (s.9 i 10),
- tablice Karnaugha (inaczej: kraty Veitcha lub cykliczne siatki zależności) metoda stosowana do minimalizacji funkcji o liczbie argumentów ≤ 6,
- metoda QMC (Quine'a – McCluskeya) stosowana do minimalizacji funkcji logicznych o dowolnej liczbie argumentów.

Przykład. Minimalizacja z wykorzystaniem praw algebry Boole'a

Kanoniczna postać dysjunkcyjna funkcji z przykładu 1.

$$z = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$$

z prawa idempodentności: $a\bar{b}\bar{c} = a\bar{b}\bar{c} + a\bar{b}\bar{c}$, stąd:

$$z = \boxed{a\bar{b}\bar{c} + a\bar{b}c} + \boxed{a\bar{b}\bar{c} + ab\bar{c}}$$

z prawa sklejania: $a\bar{b}\bar{c} + a\bar{b}c = a\bar{b}$, $a\bar{b}\bar{c} + ab\bar{c} = a\bar{c}$, stosując prawo rozdzielności:

$$z = a\bar{b} + a\bar{c} = a(\bar{b} + \bar{c})$$

Kody liczbowe

Kod

dowolnie uporządkowany układ symboli przedstawiających słowa, liczby, informacje. Symbolami wykorzystywanymi przez kody liczbowe są cyfry.

Kod Graya

kod z grupy kodów refleksyjnych. *n*–pozycyjny kod Graya powstaje z kodu *n*–1 pozycyjnego zgodnie z algorytmem:

- zapisz układ symboli kodu *n*–1 pozycyjnego w kolumnie,
- do symboli kodu *n*–1 pozycyjnego dopisz te same symbole w odwrotnej kolejności,
- do symboli kodu n-1 pozycyjnego dopisz „0”, do pozostałych „1”.

kod 1-poz.	*po odbiciu*	*kod 2-poz.*
0	0	00
1	1	01
	1	11
	0	10

Uwaga: dwie kolejne wartości kodu refleksyjnego różnią się tylko jedną pozycją.

Tablice Karnaugha

Tablica Karnaugha

siatka zbudowana z 2^n kratek, gdzie n to liczba sygnałów wejściowych:

- kwadrat $2^{0.5n} \times 2^{0.5n}$ dla n parzystych,
- prostokąt $2^{0.5(n-1)} \times 2^{0.5(n+1)}$ dla n nieparzystych,
- wiersze i kolumny opisane kodem Graya,
- każda kratka zawiera jedną wartość funkcji odpowiadającą wartości zmiennych.

Przykład. Tablice Karnaugha dla funkcji o n=2, n=3 i n=4 zmiennych

a \ *b*	0	1
0		
1		

a \ *bc*	00	01	11	10
0				
1				

ab \ *cd*	00	01	11	10
00				
01				
11				
10				

Uwaga: tablica, której wiersze i kolumny nie są opisane kodem Graya nie jest tablicą Karnaugha, a jej wykorzystanie do minimalizacji wyrażeń logicznych prowadzi do błędnych wyników.

Tablice Karnaugha – własności

- Każda kratka tablicy Karnaugha odpowiada jednej kombinacji zmiennych wejściowych: jednemu pełnemu iloczynowi kanoniczej postaci dysjunkcyjnej lub jednej pełnej sumie kanonicznej postaci koniunkcyjnej.
- Iloczyn pełny odpowiadający wybranej kratce budowany jest w postaci iloczynu argumentów funkcji (argument jest negowany jeżeli odpowiada sygnałowi 0),
- Suma pełna odpowiadająca wybranej kratce budowana jest w postaci sumy argumentów funkcji (argument jest negowany jeżeli odpowiada sygnałowi 1).
- Sąsiednie kratki (przylegające w pionie lub poziomie) odpowiadają wyrażeniom sąsiednim logicznie, np.:

$$\bar{a}b\bar{c} \quad i \quad ab\bar{c}, \qquad a + \bar{b} + c \quad i \quad \bar{a} + \bar{b} + c$$

- W przypadku funkcji 3 zmiennych sąsiednimi są również kratki z lewego i prawego brzegu tablicy, np.:

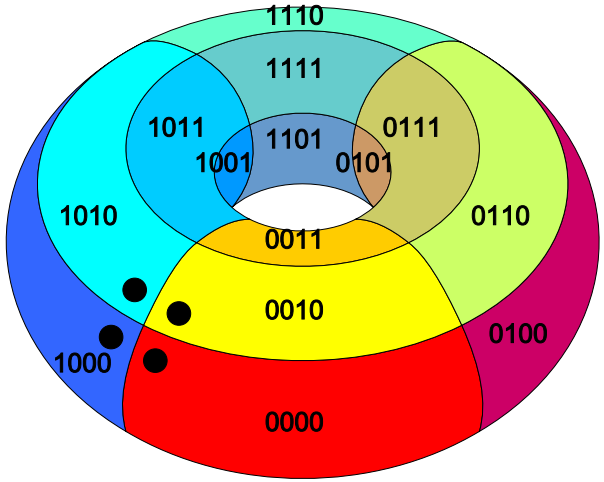
$$\bar{a}\bar{b}\bar{c} \quad i \quad \bar{a}b\bar{c}, \qquad a + b + c \quad i \quad a + \bar{b} + c$$

- W przypadku funkcji 4 zmiennych sąsiednimi są kratki z lewego i prawego oraz dolnego i górnego brzegu tablicy.

Tablice Karnaugha – własności

Wiersze i kolumny tabel Karnaugcha są opisane kodem Graya, stąd sąsiednie są elementy położone w skrajnych wierszach i skrajnych kolumnach (mogą tworzyć grupy kratek). Tabelę Karnaugha dla czterech zmiennych można przedstawić jako torus.

0000	0100	1100	1000
0001	0101	1101	1001
0011	0111	1111	1011
0010	0110	1110	1010



źródło: Wikipedia

Tablice Karnaugha – minimalizacja funkcji boolowskich

Własności tablic Karnaugha:

- automatyzują stosowanie praw sklejania,
- sąsiadujące zera lub jedynki (kratki o wspólnych bokach) odpowiadają sumom lub iloczynom pełnym, które podlegają prawom sklejania,
- wyodrębniając największe możliwe, prostokątne grupy $2^k, k = 0,1,2,3, \ldots$ sąsiadujących kratek, otrzymuje się funkcję zawierającą minimalną liczbę operatorów logicznych.

Układy wielowyjściowe

- każda z funkcji jest minimalizowana niezależnie,
- otrzymany układ można optymalizować poprzez wykorzystanie członów wspólnych.

Funkcje nie w pełni określone

- stany nieokreślone (oznaczone "–") można traktować jako 1 lub 0 powiększając grupy kratek sąsiednich,
- dla danej funkcji konkretny stan nieokreślony musi być traktowany w ten sam sposób,
- nie jest wymagane wykorzystanie wszystkich istniejących stanów nieokreślonych

Tablice Karnaugha – minimalizacja funkcji boolowskich

Wyznaczanie minimalnej postaci koniunkcyjnej (dysjunkcyjnej):

- wyszukaj i zaznacz wśród niezaznaczonych jeszcze kratek tablicy samodzielne prostokątne grupy zer (jedynek) obejmujące 2^k kratek ($k = \cdots, 3, 2, 1, 0$),
- jeżeli po wyodrębnieniu wszystkich samodzielnych grup pozostają niezaznaczone zera (jedynki) połącz je z kratkami zaznaczonymi tak aby uzyskać największą grupę obejmującą 2^k kratek (grupy mają część wspólną),
- dla każdej grupy znajdź zmienne wejściowe, których wartości są takie same dla wszystkich elementów w grupie, pozostałe odrzuć,
- dla każdej grupy zmiennych wejściowych, które pozostały zapisz alternatywę (koniunkcję) negując zmienne, dla których funkcja ma wartość jeden (zero),
- koniunkcja (alternatywa) wyrażeń zapisanych dla wszystkich grup jest minimalną postacią funkcji.

Uwaga 1: Warunkiem osiągnięcia najmniejszej postaci funkcji jest optymalne rozmieszczenie grup (najmniejsza liczba grup z możliwie największą liczbą składników).

Uwaga 2: Zazwyczaj jedna z form funkcji (dysjunkcja, koniunkcja) prowadzi do krótszego wyrażenia, minimalizacja to również wybór właściwej formy.

Przykład 1 – konstrukcja tablicy Karnaugha

a	*b*	*c*	*z*	
0	0	0	0	1
0	0	1	0	2
0	1	0	0	3
0	1	1	0	4
1	0	0	1	5
1	0	1	1	6
1	1	0	1	7
1	1	1	0	8

a \ *bc*	00	01	11	10
0	0 (1)	0 (2)	0 (4)	0 (3)
1	1 (5)	1 (6)	0 (8)	1 (7)

b \ *ac*	00	01	11	10
0	0 (1)	0 (2)	1 (6)	1 (5)
1	0 (3)	0 (4)	0 (8)	1 (7)

c \ *ab*	00	01	11	10
0	0 (1)	0 (3)	1 (7)	1 (5)
1	0 (2)	0 (4)	0 (8)	1 (6)

Przykład 1 – minimalizacja

Minimalna postać koniunkcyjna

a \ bc	00	01	11	10
0	0	0	0	0
1	1	1	0	1

a	b	c
0	0	0
0	0	1
0	1	1
0	1	0

a

a	b	c
0	1	1
1	1	1

$\bar{b} + \bar{c}$

$$z = a\,(\bar{b} + \bar{c})$$

Minimalna postać dysjunkcyjna

a \ bc	00	01	11	10
0	0	0	0	0
1	1	1	0	1

a	b	c
1	0	0
1	0	1

$a\bar{b}$

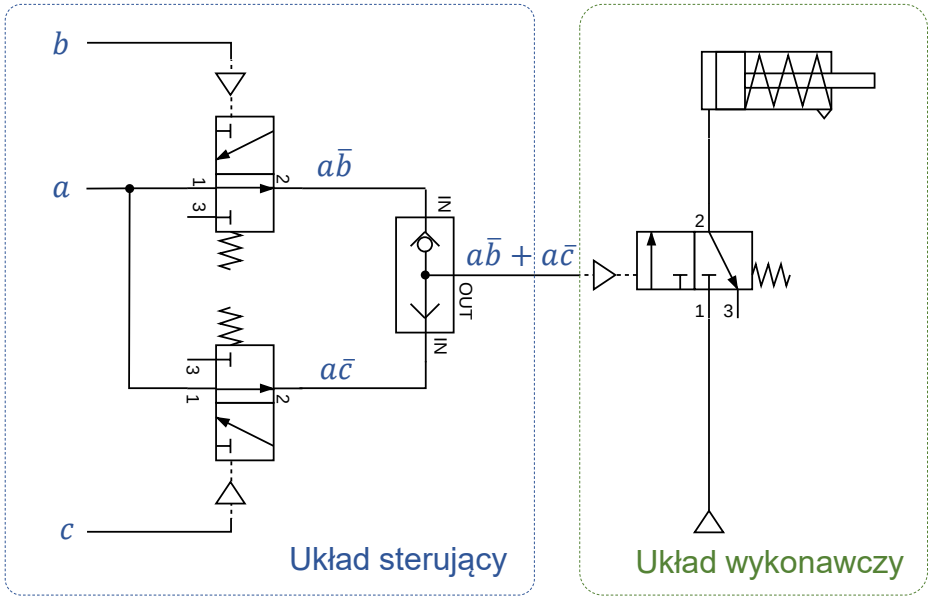
a	b	c
1	0	0
1	1	0

$a\bar{c}$

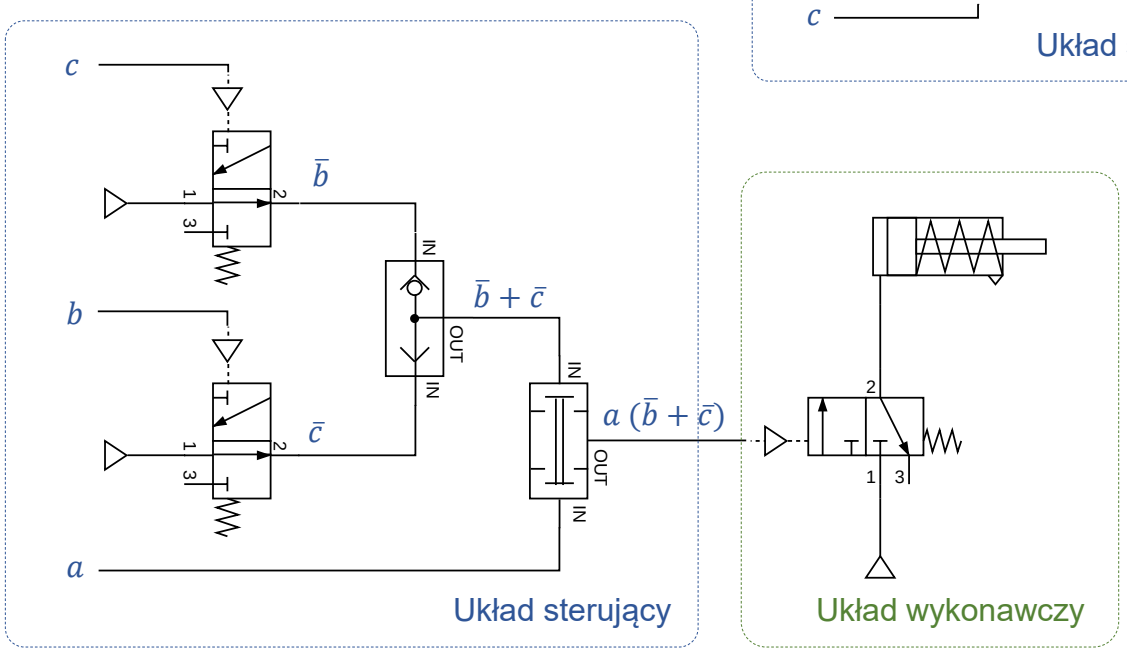
$$z = a\bar{b} + a\bar{c}$$

Układ sterowania zwrotnicą

$$z = a\bar{b} + a\bar{c}$$



$$z = a\,(\bar{b} + \bar{c})$$



Przykład 2 – konstrukcja tablic Karnaugha

a	b	c	z_1	z_2
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	–
1	0	1	1	–
1	1	0	1	–
1	1	1	0	0

Funkcja z_1

a \ bc	00	01	11	10
0	0	0	0	0
1	1	1	0	1

Funkcja z_2

a \ bc	00	01	11	10
0	1	1	0	1
1	–	–	0	–

Przykład 2 – minimalizacja

Minimalna postać funkcji z_1 (jak w przykładzie 1)

a \ bc	00	01	11	10
0	0	0	0	0
1	1	1	0	1

koniunkcyjna: $z_1 = a\,(\bar{b} + \bar{c})$

dysjunkcyjna: $z_1 = a\bar{b} + a\bar{c}$

Minimalna postać funkcji z_2

a \ bc	00	01	11	10
0	1	1	0	1
1	–	–	0	–

a	b	c
0	1	1
1	1	1

koniunkcyjna:
$z_2 = \bar{b}\,\bar{c}$

a \ bc	00	01	11	10
0	1	1	0	1
1	–	–	0	–

a	b	c
0	0	0
0	0	1
1	0	0
1	0	1

a	b	c
0	0	0
0	1	0
1	0	0
1	1	0

dysjunkcyjna:
$z_2 = \bar{b} + \bar{c}$