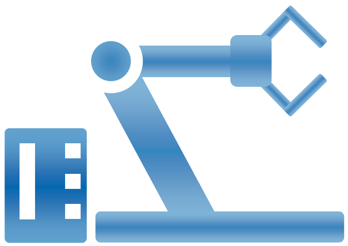
Automatyka i robotyka przemysłowa

Wprowadzenie do automatyki

Pojęcia podstawowe

Układy cyfrowe

Funkcje boolowskie



Literatura

- Kost G., Łebkowski P., Węsierski Ł. – *Automatyzacja i robotyzacja procesów produkcyjnych*, PWE, Warszawa, 2013
- Mikulczyński T. – *Automatyzacja procesów produkcyjnych*, WNT, Warszawa, 2006
- Pająk I., Pająk G. – *Cyfrowe układy automatyki*, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra, 2003 *(wersja elektroniczna na stronie przedmiotu)*
- Kasprzyk J. – *Programowanie sterowników przemysłowych*, WNT, Warszawa, 2006
- Craig J. J. – *Wprowadzenie do robotyki. Mechanika i sterowanie*, WNT, Warszawa, 1993
- *Podstawy robotyki. Teoria i elementy manipulatorów i robotów*, praca zbiorowa pod red. A. Moreckiego i J. Knapczyka, WNT, Warszawa, 1999.
- Universal Robots, *Universal Robots e-Series Podręcznik użytkownika*.
- Universal Robots Academy, E-szkolenia na temat e-Series: https://academy.universal-robots.com/pl/bezplatne-e-szkolenie/e-szkolenia-na-temat-e-series/

Proces produkcyjny (proces produkcji, produkcja) – działania mające na celu przekształcenie surowców i materiałów w produkty gotowe (finalne, końcowe).

Proces technologiczny – zasadnicza część procesu produkcyjnego, zbiór czynności zmieniających własności fizykochemiczne surowców i materiałów lub prowadzących do wykonania połączeń elementów w wyrobie.

Proces pomocniczy – zbiór czynności, które ułatwiają wykonanie procesu technologicznego, np. transport (wewnętrzny i zewnętrzny), kontrola jakości, magazynowanie, itp.

System produkcyjny – celowo zaprojektowany układ materialny, energetyczny i informacyjny eksploatowany przez człowieka i służący wytworzeniu określonych produktów w celu zaspokajania potrzeb konsumentów.

Zautomatyzowany system produkcyjny – system produkcyjny, w którym praca człowieka (fizyczna i umysłowa) została w znaczący sposób zastąpiona przez pracę maszyn.

Systemy produkcyjne jako systemy mechatroniczne

Mechatronika – pojęcie powstałe w 1969 roku w japońskiej firmie Yaskawa Electric Corporation, przez połączenie słów mechanika i elektronika. Pierwotnie oznaczało uzupełnienie komponentów mechanicznych przez elektronikę w mechanice precyzyjnej. Obecnie interdyscyplinarna nauka inżynierska, która stanowi połączenie mechaniki z elektroniką i informatyką.

Mechanika
- elementy mechaniczne
- mechanika precyzyjna
- maszyny i urządzenia

Elektronika
- sensory
- sterowniki
- elementy wykonawcze

Zautomatyzowany system produkcyjny

Informatyka
- software
- sztuczna inteligencja
- teoria sterowania

Sygnał – przebieg czasowy określonej wielkości fizycznej, za pomocą której przekazywane są informacje. Na sygnał składają się:

- treść sygnału – informacja przenoszona przez sygnał,
- nośnik sygnału – wielkość fizyczna, której zmiany umożliwiają przekazanie określonych treści (np. ciśnienie gazu lub cieczy, natężenie lub napięcie prądu, itp.) .

Układ – umownie wyodrębniony fragment rzeczywistości. Oddziaływania zewnętrzne, które wpływają na zachowanie układu nazywane są sygnałami wejściowymi, a miejsca ich oddziaływania wejściami układu. Wielkości opisujące oddziaływanie układu na środowisko nazywane są sygnałami wyjściowymi układu, a miejsca ich oddziaływania wyjściami układu. Układy przetwarzają sygnały wejściowe w sygnały wyjściowe.

Sterowanie – oddziaływanie na określony układ w celu zapewnienia pożądanego zachowania.

Obiekt sterowania – układ, w którym należy wymusić pożądane przebiegi procesów za pomocą oddziaływań sterujących.

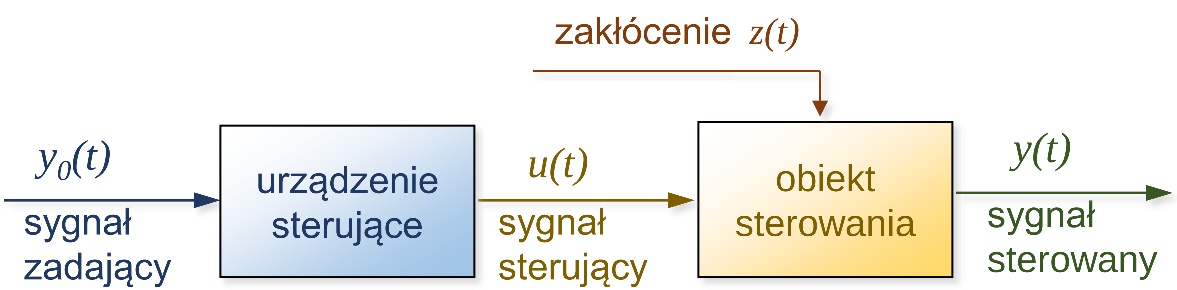
Urządzenie sterujące – układ, który generują sygnały sterujące.

Układ sterowania – układ złożony z obiektu sterowania i urządzenia sterującego.

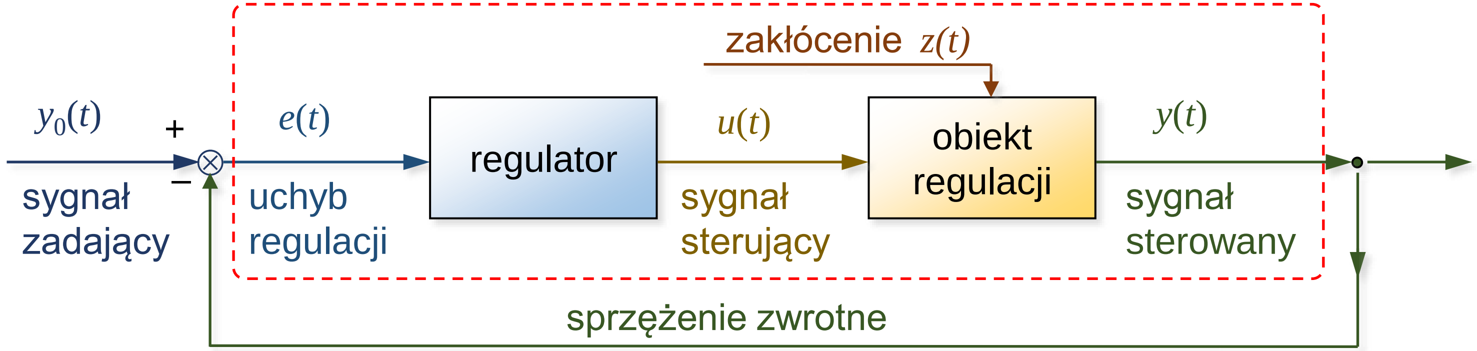
Sygnały występujące w układach sterowania:

- sygnały zadające (sygnały zadane) – określają pożądaną zmianę sygnałów sterowanych (wyjściowych),
- sygnały sterujące (sterowania) – sygnały doprowadzane do wejść obiektu sterowania, zmieniane tak, aby osiągnąć pożądany przebieg sterowanego procesu,
- sygnały zakłócające (zakłócenia) – wszelkie oddziaływania, które utrudniają realizację zadania sterowania, na ogół mają charakter przypadkowy,
- sygnały sterowane – sygnały wyjściowe obiektu sterowania, są wynikiem oddziaływania na obiekt sygnałów sterujących i zakłóceń.

W układach otwartych urządzenie sterujące otrzymuje informacje dotyczące pożądanego celu sterowania (sygnały zadające). Dodatkowo może otrzymywać pewne informacje o zakłóceniach lub pewnych wielkościach pomocniczych charakteryzujących pracę obiektu. Urządzenie sterujące w układach otwartych nie ma informacji o sygnale sterowanym, więc oddziaływanie na obiekt sterowania nie zależy od osiągniętych wyników sterowania. Układ otwarty pracuje w oparciu o pewien model obiektu sterowania i jest wrażliwy na zakłócenia.



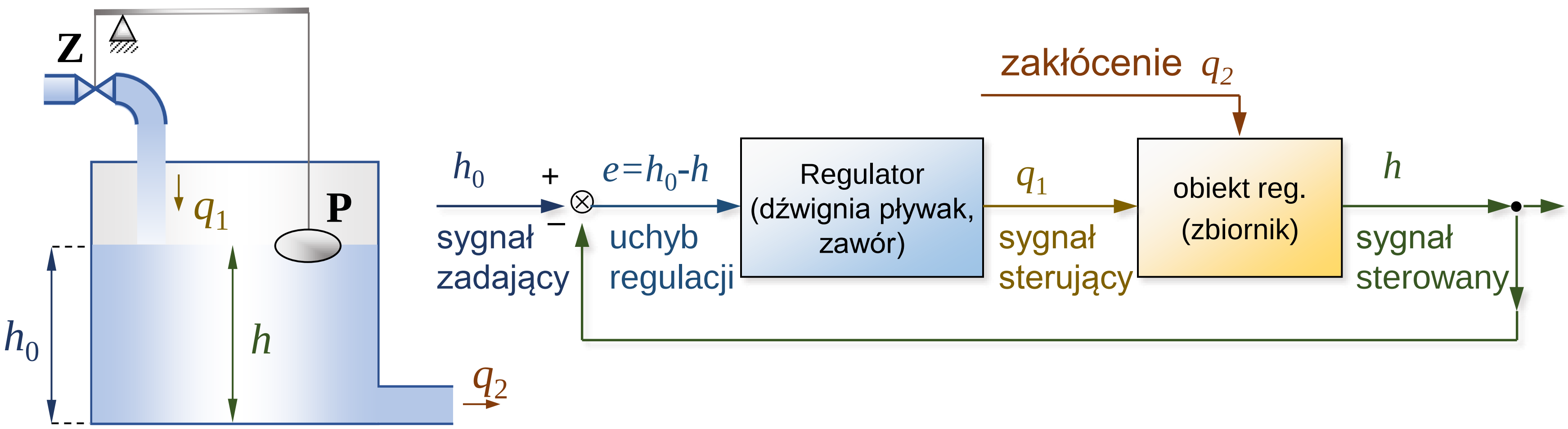
W zamkniętych układach sterowania urządzenie sterujące otrzymuje informacje dotyczące zarówno pożądanego celu sterowania (sygnały zadające) jak również skutków sterowania (sygnały sterowane). W układach zamkniętych oddziaływanie urządzeń sterujących na obiekt sterowania zależy od osiągniętych wyników sterowania tzn. zależy od przebiegu sygnałów sterowanych. Wprowadzenie do urządzenia sterującego sygnałów wyjściowych uniezależnia układ sterowania od zakłóceń, ponieważ kontrola skutków sterowania umożliwia reakcję na zmiany zachodzące w środowisku i bieżące korygowanie sygnału sterującego.



Uchyb regulacji – różnica między wartością zadaną wielkości regulowanej a jej wartością rzeczywistą.

Sprzężenie zwrotne – wprowadzenie na wejście układu informacji o jego wielkości wyjściowej.

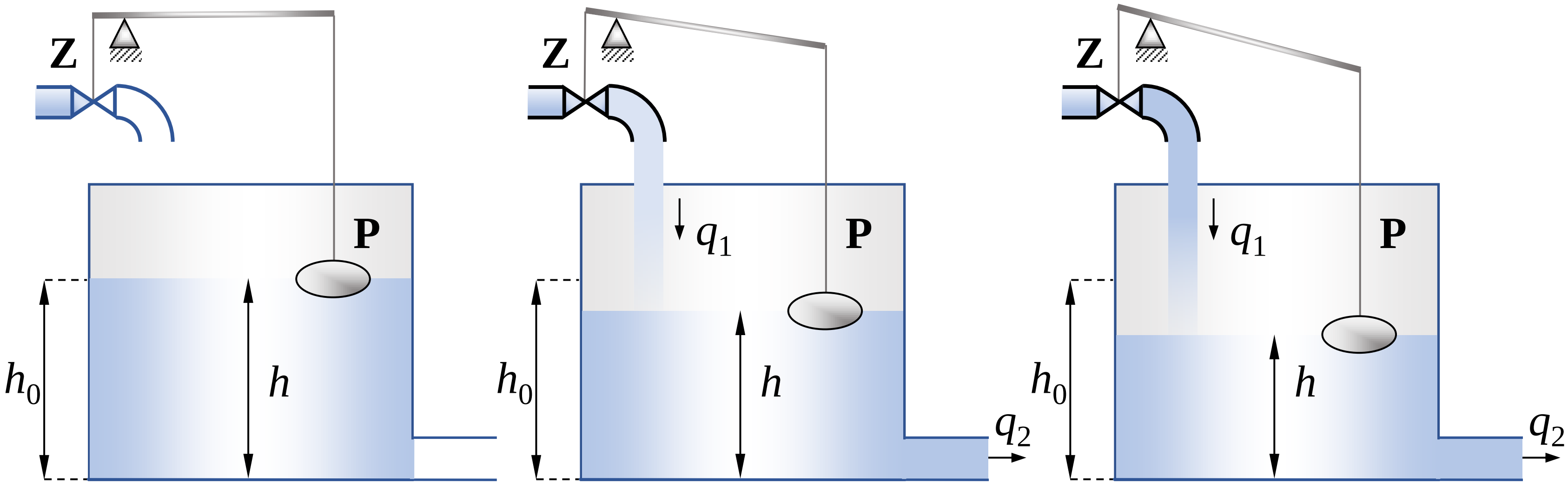
Przykładowy układ sterowania



Układ utrzymuje stały poziomu cieczy (h_0) w zbiorniku, przy zmieniającej się w sposób przypadkowy wartości strumienia cieczy wypływającej (q_2) ze zbiornika regulując wielkość strumienia cieczy dopływającej (q_1).

Zasada działania:

- Spadek poziomu cieczy h powoduje opadanie pływaka **P**, który ciągnie prawe ramię dźwigni. Lewe ramię dźwigni unosi się i otwiera zawór **Z** zwiększając wartość strumienia wody q_1 co prowadzi do podniesienia poziomu cieczy h.
- Wzrost poziomu cieczy w h powoduje podniesienie pływaka **P**, który za pośrednictwem dźwigni zamyka zawór **Z** zmniejszając wartość strumienia cieczy q_1 Długości ramion regulatora dobrane są tak, że $q_1=0$ gdy $h=h_0$.



$q_2 = 0, h = h_0$
$e = 0, q_1 = 0$

$q_2 > 0, h < h_0$
$e > 0, q_1 > 0$

$q_2 > 0, h < h_0$
$e > 0, q_1 > 0$
h maleje, e rośnie

Gdy $h = h_0$ uchyb regulacji jest zerowy, więc $q_1 = 0$

Gdy $q_2 > 0$ to $h < h_0$ i uchyb regulacji jest niezerowy zerowy, więc $q_1 > 0$

Większy spadek poziomu oznacza większy uchyb regulacji i większy strumień q_1

Sygnał analogowy – sygnał, który może przyjmować dowolną wartość z ciągłego przedziału i może być określony w dowolnej chwili czasu.

Kwantyzacja – nieliniowe odwzorowanie zmniejszające dokładność danych poprzez ograniczenie ich zbioru wartości. W wyniku procesu kwantyzacji każda wartość wyjściowa należąca do określonego przedziału jest odwzorowana na jedną wartość wyjściową (poziom reprezentacji) przypisaną temu przedziałowi.

Dyskretyzacja (próbkowanie, kwantyzacja w czasie) – proces tworzenia sygnału dyskretnego w dziedzinie czasu, reprezentującego sygnał ciągły.

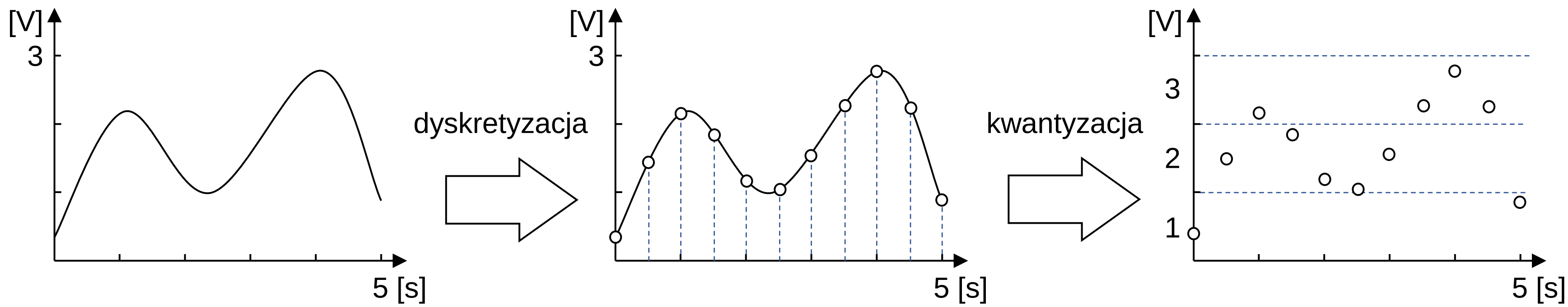
Sygnał dyskretny – sygnał o nieciągłym przebiegu, którego wartości określone są tylko w pewnych chwilach czasowych. s.d. ma postać ciągu liczbowego, pojedynczą wartość ciągu nazywa się próbką.

Sygnał cyfrowy – sygnał, którego dziedzina i zbiór wartości są dyskretne (mogą przyjmować tylko określone wartości ze skończonego zbioru).

Sygnał binarny – szczególny przypadek sygnału o dyskretnych wartościach, który przyjmuje dwie wartości: niską i wysoką, interpretowane zazwyczaj jako 0 i 1.

Układ cyfrowy – układ przetwarzający sygnały dyskretne. u.c. są podstawowymi układami sterowania stosowanymi do automatyzacji procesów produkcyjnych.

Przetwarzanie analogowo-cyfrowe



sygnał ciągły $y = f(t)$ | sygnał dyskretny $y_d = \{0.4, 1.5, 2.2, 1.8, 1.2, 1.0, \ldots\}$ | sygnał cyfrowy $y_c = \{1,2,3,2,2,2,2,3,3,3,1\}$

Przetwornik analogowo-cyfrowy A/C (ang. A/D lub ADC) – układ służący do zamiany sygnału analogowego na cyfrowy.

Przetwornik cyfrowo-analogowy C/A (ang. DAC) – układ przetwarzający dyskretny sygnał cyfrowy na równoważny mu sygnał analogowy.

Cechy układów cyfrowych

Zalety układów cyfrowych

- duża odporność na zakłócenia,
- większa niezawodność układów,
- łatwość zapamiętywania i przechowywania informacji cyfrowych,
- duża dokładność przetwarzania (zależy od dokładności informacji wejściowych),
- możliwość realizacji złożonych algorytmów przetwarzania sygnałów,
- niski koszt w stosunku do realizowanych funkcji.

Klasyfikacja ze względy na sposób przetwarzania sygnałów

- układy kombinacyjne – każda kombinacja wartości sygnałów wejściowych (stan wejść) określa jednoznacznie kombinację wartości sygnałów wyjściowych (stan wyjść), układy te nazywane są również układami bez pamięci,
- układy sekwencyjne – istnieje przynajmniej jeden stan wejść któremu odpowiada kilka stanów wyjść, układy nazywane są również układami z pamięcią.

Układ logiczny – szczególny przypadek układu cyfrowego, w którym sygnały przyjmują dwa poziomy określane wartościami liczbowymi 0 i 1. Działanie układu tego typu jest oparte na dwuelementowej algebrze Boole'a. u.l. należą do najczęściej wykorzystywanych układów cyfrowych.

Dwuelementowa algebra Boole'a jest sformalizowanym uogólnieniem rachunku zdań. Jest ona definiowana jako układ:

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

gdzie: {0,1} – zbiór elementów algebry; 0,1 – stałe algebry; +,· – dwuargumentowe operacje alternatywy i koniunkcji; ¯ – jednoargumentowa operacja negacji.

a	b	$a+b$	$a \cdot b$	$\bar{a}$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

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

Wybrane prawa algebry Boole'a

Prawa istnienia elementu identycznościowego

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

Prawa istnienia dopełnienia

$a + \overline{a} = 1$ $\qquad a\,\overline{a} = 0$

Prawa przemienności

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

Prawa łączności

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

Prawa rozdzielności

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

Prawa de Morgana

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

Prawa idempotentności

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

Prawa pochłaniania

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

Prawa sklejania

$$(a + b)(a + \overline{b}) = a$$

$$a\,b + a\,\overline{b} = a$$

Prawo podwójnej negacji

$$\overline{\overline{a}} = a$$

Inne własności

$$a\;0 = 0$$

$$a + 1 = 1$$

Prawa de Morgana dla wielu argumentów

$$\overline{a + b + c + d} = \overline{a}\,\overline{b}\,\overline{c}\,\overline{d}$$

$$\overline{a\,b\,c\,d} = \overline{a} + \overline{b} + \overline{c} + \overline{d}$$

Funkcją boolowską (logiczną) nazywane jest odwzorowanie postaci:

$$f\colon \boldsymbol{X} \to \boldsymbol{Y},$$

gdzie: $\boldsymbol{X} \subseteq \boldsymbol{B}^n = \boldsymbol{B} \times \boldsymbol{B} \times \ldots \times \boldsymbol{B}$; $\boldsymbol{Y} \subseteq \boldsymbol{B}^m$; $\boldsymbol{B} = \{0, 1\}$

Elementami $\boldsymbol{X}$ i $\boldsymbol{Y}$ są odpowiednio *n*-elementowe oraz *m*-elementowe ciągi będące kombinacjami zerojedynkowymi.

Jeżeli odwzorowanie f jest określone dla każdego elementu $\boldsymbol{B}^n$ (czyli $\boldsymbol{X}=\boldsymbol{B}^n$) to funkcja jest nazywana *funkcją zupełną*, jeżeli $\boldsymbol{X}$ jest podzbiorem właściwym $\boldsymbol{B}^n$ *funkcją niezupełną* lub *funkcją nie w pełni określoną*.

Za pomocą funkcji boolowska można opisać działanie układu logicznego.

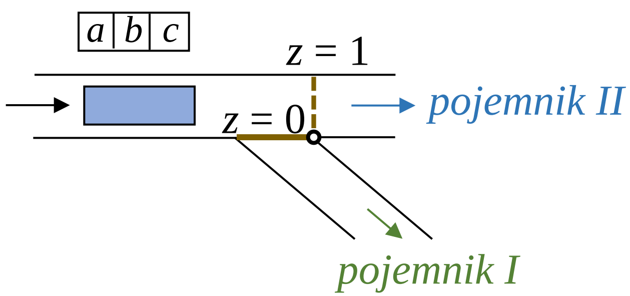
Reprezentacja funkcji boolowskich

- opis słowny (np. funkcja przyjmuje wartość 1 wtedy i tylko wtedy gdy jej obydwa argumenty są różne),
- wyrażenia boolowskie (np. $\bar{a}b + a\bar{b}$),
- tablice prawdy,
- tablice Karnaugha (zmodyfikowane tablice prawdy).

a	*b*	*f*
0	0	0
0	1	1
1	0	1
1	1	0

Przykład 1. Jednowyjściowy układ kombinacyjny

Przenośnik taśmowy umieszcza detale w jednym z dwóch pojemników. Na stanowisku kontroli, przy pomocy odpowiednich czujników, badane są trzy cechy detalu (a, b, c). Każdy z czujników sygnalizuje zbadaną cechę wartością "1" lub "0" (1 – wartość prawidłowa, 0 – nieprawidłowa). Detal trafia do pojemnika I jeżeli co najmniej dwie cechy mają prawidłowe wartości i jedną z nich jest cecha a w przeciwnym wypadku trafia do pojemnika II. Należy zaprojektować urządzenie sterujące pracą zwrotnicy. Zwrotnica w stanie załączonym (z = 1) kieruje produkt do pojemnika I w stanie nie załączonym (z = 0) do pojemnika II.



Tablica prawdy

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

Opis słowny: zwrotnica jest w stanie załączonym (z = 1) wtedy i tylko wtedy gdy prawidłową wartość sygnalizują czujniki a i b lub a i c w przeciwnym wypadku z = 0.

Funkcja logiczna: $z = a\bar{b}c + ab\bar{c} + abc$

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

1. Indywidualna analiza każdej możliwej kombinacji sygnałów wejściowych (w układach logicznych 2^n kombinacji), identyfikacja stanu układu i określenie wartości sygnału wyjściowego.
2. Identyfikacja wszystkich stanów wejść odpowiadających danej wartości sygnału wyjściowego.

Przykład

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

000 $\to$ trzy nieprawidłowe cechy $\to$ pojemnik II $\to z = 0$

001 $\to$ dwie nieprawidłowe cechy $\to$ pojemnik II $\to z = 0$

011 $\to$ nieprawidłowa cecha a $\to$ pojemnik II $\to z = 0$

100 $\to$ dwie nieprawidłowe cechy $\to$ pojemnik II $\to z = 0$

101 $\to$ dwie prawidłowe cechy w tym a $\to$ pojemnik I $\to z = 1$ …

Podejście drugie: $z = 1$ gdy co najmniej dwa czujniki zwracają 1 i jednym z nich jest a, tzn. 101, 110, 111 w pozostałych wierszach należy zapisać 0.

Kanoniczna postać dysjunkcyjna funkcji boolowskiej

Twierdzenie 1.

Każdą funkcję boolowską n zmiennych $f(x_1, x_2, \dots, x_n)$ można przedstawić jako rozwinięcie ze względu na zmienną x_1:

$$f(x_1, x_2, \dots, x_n) = f(1, x_2, \dots, x_n)x_1 + f(0, x_2, \dots, x_n)\overline{x}_1$$

Korzystając z *Tw.1.* $f(1, x_2, \dots, x_n)$ i $f(0, x_2, \dots, x_n)$ można rozwinąć względu na x_2:

$$f(1, x_2, \dots, x_n) = f(1,1, \dots, x_n)x_2 + f(1,0, \dots, x_n)\bar{x}_2$$

$$f(0, x_2, \dots, x_n) = f(0,1, \dots, x_n)x_2 + f(0,0, \dots, x_n)\bar{x}_2$$

Powtarzając operację rozwijania dla wszystkich n zmiennych otrzymuje się 2^n sum iloczynów pełnych i wartości funkcji o stałych argumentach. Taka postać nazywana jest **kanoniczną postacią dysjunkcyjną (alternatywną) funkcji boolowskiej**:

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

Uwaga: ponieważ $0 \cdot a = 0$ przy tworzeniu kanonicznej postaci dysjunkcyjnej można usunąć wszystkie iloczyny w których funkcja ma wartość 0.

Kanoniczna postać koniunkcyjna funkcji boolowskiej

Twierdzenie 2.

Każdą funkcję boolowską n zmiennych $f\,(x_1,\ x_2,\ \ldots,\ x_n)$ można przedstawić jako rozwinięcie ze względu na zmienną x_1:

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

Korzystając z *Tw.2.* $f\,(1,\ x_2,\ \ldots,\ x_n)$ i $f\,(0,\ x_2,\ \ldots,\ x_n)$ można rozwinąć względu na x_2:

$$f(1, x_2, \dots, x_n) = (f(1{,}0, \dots, x_n) + x_2) + (f(1{,}1, \dots, x_n) + \bar{x}_2)$$

$$f(0, x_2, \dots, x_n) = (f(0{,}0, \dots, x_n) + x_2) + (f(0{,}1, \dots, x_n) + \bar{x}_2)$$

Powtarzając operację rozwijania dla wszystkich n zmiennych otrzymuje się 2^n iloczynów sum pełnych i wartości funkcji o stałych argumentach. Taka postać nazywana jest **kanoniczną postacią koniunkcyjną funkcji boolowskiej**:

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

Uwaga: ponieważ $1{+}a = 1$ przy tworzeniu kanonicznej postaci dysjunkcyjnej można usunąć wszystkie iloczyny w których funkcja ma wartość 1.

Przykład 1. – kanoniczna postać funkcji

Kanoniczna postać funkcji logicznej opisującej pracę sortownika z Przykładu 1.

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

Kanoniczna postać koniunkcyjna

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

Kanoniczna postać dysjunkcyjna

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

Uwaga: obydwie formy funkcji są równoważne

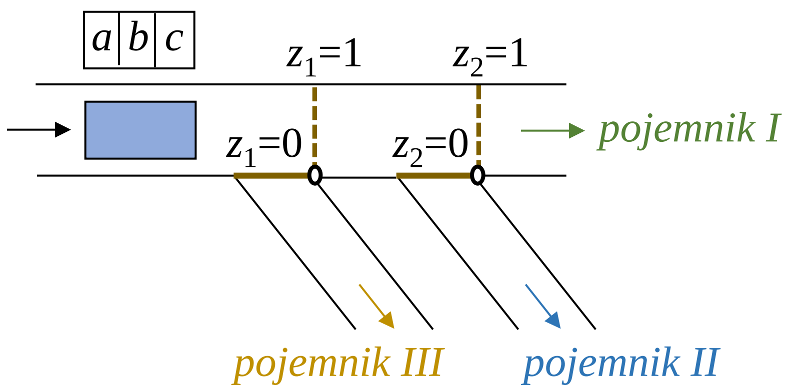
Układy wielowyjściowe, funkcje nie w pełni określone

Układ wielowyjściowy o m wejściach i k wyjściach można zapisać jako układ k funkcji o m argumentach (każda z funkcji opisuje jedno z wyjść układu).

Funkcje które nie są określone dla wszystkich możliwych kombinacji sygnałów wejściowych są **funkcjami nie w pełni określonymi**.

Przykład 2.

Przenośnik taśmowy umieszcza detale w jednym z trzech pojemników. Na stanowisku kontroli, przy pomocy odpowiednich czujników, badane są trzy cechy detalu (a, b, c). Detale o co najmniej dwóch nieprawidłowych cechach powinny trafiać do pojemnika I, detale o wszystkich prawidłowych cechach do pojemnika II, a pozostałe do pojemnika III. Należy zaprojektować układ sterujący dwiema zwrotnicami urządzenia sortującego.



pojemnik I : $z_1=0,\ z_2=0$

pojemnik II : $z_1=0,\ z_2=1$

pojemnik III : $z_1=1$, z_2 *obojętne*

Przykład 2. – kanoniczna postać funkcji

Tabela prawdy

Pojemnik I : 000, 001, 010, 100

Pojemnik II : 111

Pojemnik III : 011, 101, 110

a	b	c	p	z_1	z_2
0	0	0	I	0	0
0	0	1	I	0	0
0	1	0	I	0	0
0	1	1	III	1	-
1	0	0	I	0	0
1	0	1	III	1	-
1	1	0	III	1	-
1	1	1	II	0	1

Kanoniczna postać koniunkcyjna

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

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

Kanoniczna postać dysjunkcyjna

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

$$z_2 = abc$$

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

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

Metody minimalizacji:

- wykorzystanie praw algebry Boole'a (s.15),
- 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 przykładu1.

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

z prawa idempodentności: $abc = abc + abc$, stąd:

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

z prawa sklejania: $abc + ab\overline{c} = ab$, $abc + a\overline{b}c = ac$, stąd ostatecznie:

$$z = ab + ac = a(b + c)$$

Kody liczbowe

Kod to dowolnie uporządkowany układ symboli przedstawiających słowa, liczby, informacje. Symbolami wykorzystywanymi przez **kody liczbowy** są cyfry.

Naturalny kod dwójkowy (binarny) jest kodem wagowym (pozycyjnym) wykorzystującym do kodowania cyfry 0 i 1, każda pozycja (bit) kodu ma określoną wagę: waga i-tej pozycji w n pozycyjnym kodzie wynosi 2^i gdzie $i = 0, 1, 2, \ldots, n–1$.

$$101110_2 = \mathbf{1}\cdot 2^5 + \mathbf{0}\cdot 2^4 + \mathbf{1}\cdot 2^3 + \mathbf{1}\cdot 2^2 + \mathbf{1}\cdot 2^1 + \mathbf{0}\cdot 2^0 = 46_{10}$$

Kod Graya należy do grupy kodów refleksyjnych. n–pozycyjny kod Graya powstaje z kodu n–1 pozycyjnego zgodnie z następującym algorytmem:

- zapisać układ symboli kodu n–1 pozycyjnego w kolumnie,
- do układu symboli kodu n–1 pozycyjnego dopisać jeszcze raz te same symbole ale w odwrotnej kolejności (odbicie lustrzane),
- pierwotne symbole poprzedzić cyfrą „0”, symbole dopisane „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 jest kwadratową lub prostokątną siatką zbudowaną z 2^n kratek. Dla parzystej liczby sygnałów wejściowych tablica ma kształt kwadratu o wymiarach $2^{0.5n}\times2^{0.5n}$, dla nieparzystej – prostokąta o wymiarach $2^{0.5(n-1)}\times2^{0.5(n+1)}$.

Każdy wiersz i kolumna tablicy opisane są za pomocą kodu Graya. Adres kratki (kod złożony z numeru wiersza i kolumny) wyznacza jednoznacznie wartości argumentów, wartość funkcji (tzn. 0 lub 1) odpowiadającą tym argumentom jest wpisywana we wnętrzu kratki.

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				

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.:

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

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

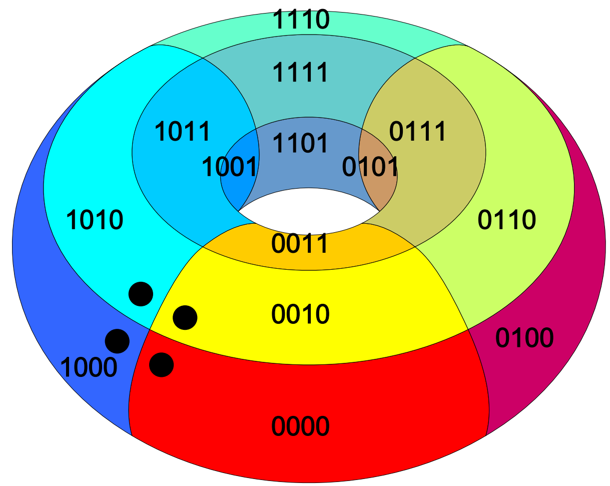
$$\bar{a}\bar{b}\bar{c} \;\; i \;\; \bar{a}b\bar{c}, \quad a+b+c \;\; i \;\; a+\bar{b}+c$$

- 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). Tabelą 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

Tablice Karnaugha automatyzują stosowanie praw sklejania. Korzystając z własności tablicy można wyodrębnić grupy sąsiadujących zer (jedynek) odpowiadające sumom (iloczynom) pełnym, które podlegają prawom sklejania. Wyodrębniając jak najmniejszą liczbę możliwie największych grup otrzymuje się koniunkcyjną (dysjunkcyjną) postać funkcji zawierającą minimalną liczbę operatorów logicznych.

Układy wielowyjściowe

Minimalizację układu wielowyjściowego można przeprowadzić minimalizując każdą z opisujących go funkcji oddzielnie. Otrzymane w ten sposób funkcje są minimalne ale cały układ czasami może być optymalizowany poprzez wykorzystanie członów wspólnych powtarzających się w funkcjach opisujących poszczególne wyjścia.

Funkcje nie w pełni określone

W przypadku funkcji nie w pełni określonej stany nieokreślone (oznaczone "–") można traktować jako 1 lub 0 tak aby uzyskiwane grupy były jak największe. 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 grupy zer (jedynek) obejmujące 2^k kratek (k = …, 3, 2, 1, 0),
- jeżeli po wyodrębnieniu wszystkich samodzielnych grup pozostają jeszcze 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) daje mniejsze wyrażenie, minimalizacja to również wybór właściwej formy.

Przykład 1. – konstrukcja tablicy Karnaugha

Tabele Karnaugha dla sortownika z Przykładu 1.

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

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

b \ *ac*	**00**	**01**	**11**	**10**
1	0	0	**1**	**1**
0	0	0	**1**	0

c \ *ab*	**00**	**01**	**11**	**10**
0	0	0	**1**	0
1	0	0	**1**	**1**

Przykład 1. – minimalizacja

Minimalna postać koniunkcyjna funkcji opisującej pracę sterownika z Przykładu 1.

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

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

a

a	b	c
0	0	0
1	0	0

$b+c$

$$z = a(b+c)$$

Minimalna postać dysjunkcyjna funkcji opisującej pracę sterownika z Przykładu 1.

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

a	b	c
1	0	1
1	1	1

ac

a	b	c
1	1	1
1	1	0

ab

$$z = ac + ab$$

Przykład 2. – konstrukcja tablic Karnaugha

Tabele Karnaugha układu wielowyjściowego z przykładu 2.
(funkcja z_2 nie w pełni określona)

a	*b*	*c*	z_1	z_2
0	0	0	0	0
0	0	**1**	0	0
0	**1**	0	0	0
0	**1**	**1**	**1**	-
1	0	0	0	0
1	0	**1**	**1**	-
1	**1**	0	**1**	-
1	**1**	**1**	0	**1**

funkcja z_1

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

funkcja z_2

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

funkcja z_1

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

a	b	c
0	0	0
0	0	1

a	b	c
0	0	0
1	0	0

a	b	c
0	0	0
0	1	0

a	b	c
1	1	1

a	b	c
1	0	1

a	b	c
0	1	1

a	b	c
1	1	0

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

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

funkcja z_2

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

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

a	b	c
0	1	1
1	1	1

$$forma\ koniunkcyjna: z_2 = bc$$

$$forma\ dysjunkcyjna: z_2 = bc$$