UKŁADY PRZEŁĄCZAJĄCE

1. Podstawowe pojęcia

Algebra Boole'a. Analiza oraz opis własności układów przełączających jest przeprowadzany przy użyciu algebry Boole'a. Wartości argumentów oraz funkcji należą do dwuargumentowego zbioru $\{0, 1\}$, na którym wykonywane są trzy podstawowe operacje:

- *suma logiczna (dysjunkcja)* $a+b$ $(a \vee b)$
- *iloczyn logiczny (koniunkcja)* $a \cdot b$ $(a \wedge b)$
- *negacja (inwersja)* $\overline{a}$

Podstawowe prawa algebry Boole'a:

- prawo przemienności: $a+b=b+a$, $a \cdot b = b \cdot a$
- prawo łączności: $(a+b)+c=a+(b+c)$, $(a \cdot b)\cdot c = a \cdot (b \cdot c)$
- prawo rozdzielności: $a+(b \cdot c)=(a+b)\cdot(a+c)$, $a \cdot (b+c) = (a \cdot b)+(a \cdot c)$
- prawa de Morgana: $\overline{a \cdot b \cdot c \cdot \ldots} = \overline{a}+\overline{b}+\overline{c}+\ldots$, $\overline{a+b+c+\ldots} = \overline{a}\cdot\overline{b}\cdot\overline{c}\cdot\ldots$

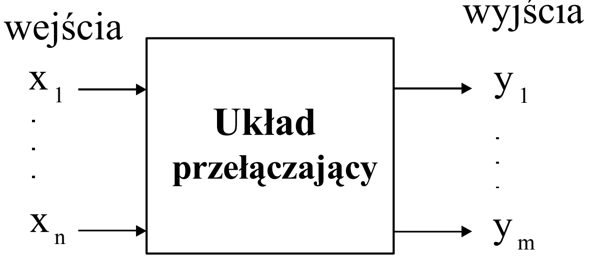
Podstawowe tożsamości:

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

$a+0=a$, $a+1=1$, $a+a=a$, $a+\overline{a}=1$, $a+(a \cdot b)=a$,

$a \cdot 0=0$, $a \cdot 1=a$, $a \cdot a=a$, $a \cdot \overline{a}=0$, $a \cdot (a+b)=a$.

Układem przełączającym (automatem cyfrowym) nazywamy układ służący do przetwarzania sygnałów dwuwartościowych (binarnych). Można go przedstawić w postaci bloku posiadającego wejścia $x_1, x_2, ..., x_n$ i wyjścia $y_1, y_2, ..., y_m$. Każde wejście i wyjście przyjmuje tylko jedną z dwóch wartości: 0 lub 1. Ciąg sygnałów wejściowych $x_1, x_2, ..., x_n$ określa tzw. stan wejść, natomiast ciąg $y_1, y_2, ..., y_m$ stan wyjść automatu.



$X=(x_1, x_2, ..., x_n)$ stan wejść

$Y=(y_1, y_2, ..., y_m)$ stan wyjść

Rys. 1. Schemat blokowy układu przełączającego.

Układy przełączające można podzielić na:

- **układy kombinacyjne** - stan wyjść w dowolnej chwili czasu t zależy wyłącznie od aktualnego stanu wejść układu: $Y^{[t]} = f(X^{[t]})$,
- **układy sekwencyjne** - stan wyjść układu zależy nie tylko od aktualnego stanu wejść, ale także od poprzednich: $Y^{[t]} = f(X^{[t]}, X^{[t-\tau]}, \ldots)$

Funkcje przełączające. Funkcją przełączającą nazywamy taką funkcję f_i, która wartościom binarnym zmiennych x_i przyporządkowuje odpowiednią wartość binarną zmiennych y_i

$$y_i = f_i(x_1, x_2, \ldots, x_n) \qquad \text{gdzie } i = 1,2,\ldots,m .$$

Jest to zatem funkcja określająca stan wyjść (lub jednego z wyjść) na podstawie stanu wejść.

Można ją zapisać:

a) analitycznie (wzorem), np. $y = a + \bar{b} \cdot c$,
b) tablicą zależności (tablicą funkcji),
c) graficznie, np. schemat z bramek.

Jeżeli dla każdej kombinacji argumentów (odpowiedni wiersz tablicy zależności) utworzyć wyrażenia S_i (składniki jedynki – dla nich wartość funkcji wynosi 1), to funkcję przełączającą można jednoznacznie przedstawić:

- w postaci kanonicznej sumy

$$y = S_0 + S_1 + \ldots + S_{n-1} \qquad \text{np.} \qquad y = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + a\bar{b}c + abc$$

$$y = \sum (001,010,100,101,111)_{abc}$$

Postać kanoniczną funkcji można zwykle uprościć, a proces ten nazywamy minimalizacją funkcji. W jej wyniku otrzymujemy równanie typu „suma iloczynów” zwane *postacią normalną sumy*, bądź typu „iloczyn sum” określane jako *postać normalna iloczynu*.

Realizacja funkcji przełączającej w zminimalizowanej postaci stanowi zwykle końcowy etap projektowania kombinacyjnego układu przełączającego i odbywa się przy użyciu podstawowych funktorów logicznych (bramek), które przedstawiono poniżej.

a)	AND	OR	NOT	NAND	NOR
b)	$y = a \cdot b$	$y = a + b$	$y = \bar{a}$	$y = \overline{a \cdot b}$	$y = \overline{a + b}$

c)

AND:

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

OR:

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

NOT:

a	y
0	1
1	0

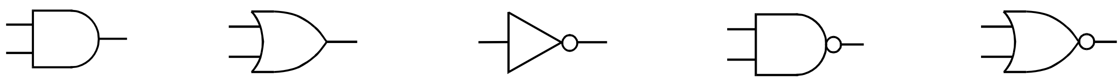
NAND:

a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

NOR:

a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

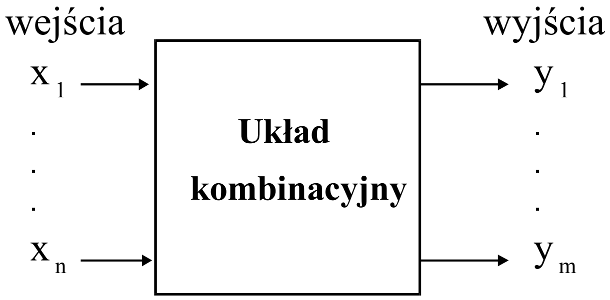
d)



Rys. 2. Podstawowe funktory logiczne: a) nazwa, b) wzór, c) tablica zależności, d) symbol.

2. Synteza układów kombinacyjnych

Układem kombinacyjnym o n wejściach $x_1, x_2, ..., x_n$ oraz m wyjściach $y_1, y_2, ..., y_m$ nazywamy układ, dla którego każda kombinacja wartości sygnałów wejściowych (stan wejść) określa jednoznacznie kombinację wartości sygnałów wyjściowych (stan wyjść).



Strukturę i działanie takiego układu opisuje zespół funkcji przełączających (tzw. funkcje wyjścia)

$$y_i = f_i(x_1, x_2, ..., x_n)$$

gdzie $x_i, y_j \in \{0,1\}$; $i = 1,2,...,n$, $j = 1,2,...,m$

Rys. 3. Schemat układu kombinacyjnego

Funkcje wyjść można otrzymać ze sporządzonej tablicy zależności, tablicy Karnaugh, przebiegów czasowych sygnałów wejściowych i wyjściowych lub wprost ze słownego opisu pracy układu.

W układach kombinacyjnych zmiana stanu wyjść jest bezpośrednim następstwem zmiany stanu wejść (stan wyjść zależy wyłącznie od stanu wejść układu). Z tego też względu układy kombinacyjne nazywane są układami przełączającymi bez pamięci.

Synteza kombinacyjnego układu przełączającego składa się zwykle z następujących etapów:

- określenia pierwotnej tablicy funkcji (tablicy zależności układu) zawierającej możliwe do wystąpienia w układzie stany (kombinacje) wejść oraz odpowiadające im stany wyjść,
- minimalizacji funkcji logicznej opisującej działanie układu (wynikającej z tablicy funkcji),
- sporządzenia schematu logicznego układu zbudowanego z zestawu wybranych elementów.

Metoda tablic (siatek) Karnaugh

Metoda tablic (siatek) Karnaugh jest jedną z bardziej popularnych metod minimalizacji wyrażeń boolowskich. Podobnie jak i w innych metodach, wykorzystuje się w niej fakt, że dla dowolnego A, B zachodzi:

$$A \cdot x + A \cdot \bar{x} = A, \qquad (B + x) \cdot (B + \bar{x}) = B \qquad \text{(reguły „sklejania”).}$$

Dwa człony iloczynowe wyrażenia, różniące się jedną negacją, można zastąpić jednym członem bez literału różnicującego. Działanie takie nosi nazwę sklejania, a sklejane człony to wyrażenia sąsiednie, np.

$$\left.\begin{matrix} 1001 \\ 1101 \end{matrix}\right\} \rightarrow 1-01 \qquad \text{czyli} \qquad \left.\begin{matrix} x_1\bar{x}_2\bar{x}_3x_4 \\ x_1x_2\bar{x}_3x_4 \end{matrix}\right\} \rightarrow x_1\bar{x}_3x_4$$

Minimalizację przeprowadza się na podstawie tzw. tablic Karnaugh w oparciu o pierwotną tablicę funkcji (stanów wejść/wyjść). Dla n-wejściowego układu tablica Karnaugh jest uporządkowaną strukturą złożoną z 2^n elementarnych prostokątów (kratek), z których każdy reprezentuje pełny iloczyn z kanonicznej formy sumacyjnej tj. odpowiedni wiersz tablicy funkcji. Rozkład kratek jest taki, że dowolne dwie sąsiednie kratki (tj. stykające się bokami) różnią się stanem tylko jednej zmiennej. Dodatkowo, każda kratka styka się swymi bokami z tyloma różnymi kratkami sąsiednimi, ile wynosi liczba zmiennych. Dla otrzymania efektu sąsiedztwa współrzędne pól tablicy odpowiadające wejściom układu kombinacyjnego opisuje się refleksyjnym kodem Gray'a.

Iloczyny pełne w tablicy Karnaugh oznacza się przy użyciu zapisu dwójkowego, w którym zmienne proste określane są cyfrą 1, a zmienne zanegowane cyfrą 0 (np. $\overline{a}b\overline{c} \to 010$).

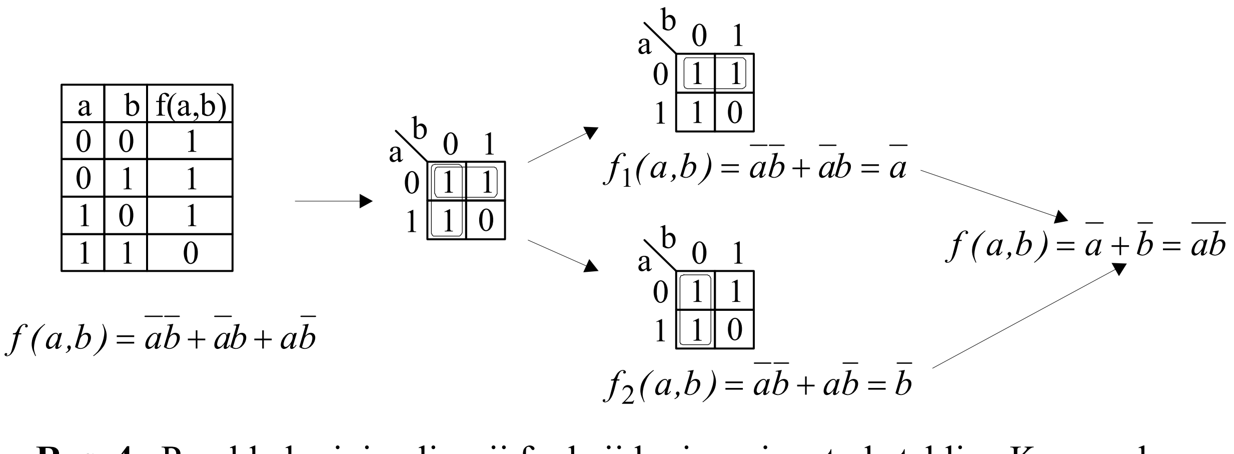
a \ bc	00	01	11	10
0	$\overline{a}\,\overline{b}\,\overline{c}$	$\overline{a}\,\overline{b}\,c$	$\overline{a}\,bc$	$\overline{a}\,b\overline{c}$
1	$a\overline{b}\,\overline{c}$	$a\overline{b}\,c$	abc	$ab\overline{c}$

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

Funkcję przełączającą (w kanonicznej formie sumacyjnej) oznacza się przez wpisanie cyfry 1 w kratkach odpowiadających wszystkim pełnym iloczynom sumy (dla których wartość funkcji wynosi 1). Pozostałe kratki oznacza się cyfrą 0 (jeśli wartość funkcji wynosi 0) lub wpisuje kreskę (gdy wartość funkcji jest nieokreślona).

Etapy procesu minimalizacji funkcji logicznej.

- Przygotowanie tablicy (lub kilku tablic, dla każdego wyjścia oddzielnie) Karnaugh i przepisanie do jej kratek odpowiednich wartości funkcji z pierwotnej tablicy funkcji (stanów we/wy). W polach odpowiadających kombinacjom zmiennych, dla których wartość funkcji jest nieokreślona, należy wpisać znak nieokreśloności np. $-$ (kreska) lub $\varnothing$.
- Zaznaczenie (ramkami) możliwie największych obszarów obejmujących wyłącznie jedynki (dla postaci sumacyjnej sąsiadujące ze sobą. Rysując ramki (”obwiednie”) należy stosować następujące zasady:
  1. liczba pól elementarnych połączonych ze sobą musi być potęgą dwójki (1, 2, 4, ... ,2^n),
  2. łączone pola muszą być sąsiednimi (tj. stykać się „bokami”),
  3. łączone pola muszą mieć kształt symetryczny względem osi (kwadraty lub prostokąty),
  4. znaki nieokreśloności można łączyć (w zależności od potrzeb) z jedynkami lub zerami.
- Zapisanie tak „zakreślonej” funkcji logicznej w postaci zminimalizowanej.



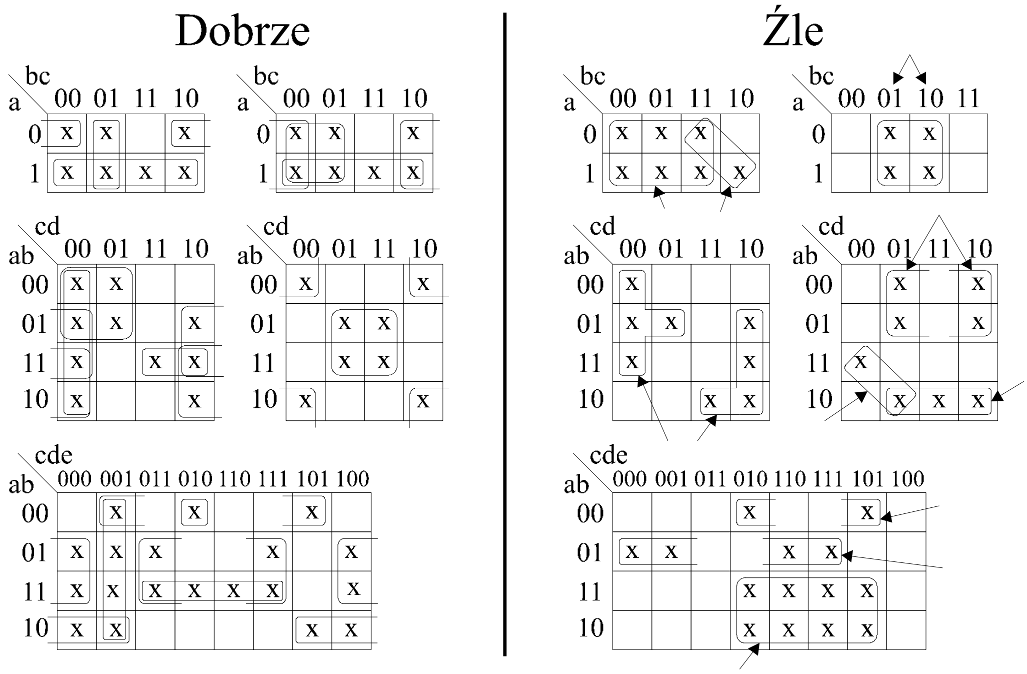
Rys. 4. Przykład minimalizacji funkcji logicznej metodą tablicy Karnaugh.

W wyniku minimalizacji metodą tablic Karnaugh otrzymuje się funkcję w postaci sumy iloczynów (postać skrócona sumy) lub w postaci iloczynu sum (postać skrócona iloczynu). Pierwsza z nich otrzymywana jest w przypadku zakreślania „jedynek” (rys. 5a), druga zaś gdy zakreślamy zera (rys. 3.5b). Zapisując funkcję w postaci skróconej iloczynu należy pamiętać o zanegowaniu argumentów opisujących odpowiednie kratki tablicy Karnaugh oraz odpowiedniej zamianie operatorów (wzór Shannona).

$y = \bar{a}\bar{b} + \bar{b}c + ab$ $\quad\quad$ $y = \left(a + \bar{b} + \bar{c}\right)\left(\bar{a} + b\right)$

Rys. 5. Minimalizacja: a) postać skrócona sumy, b) postać skrócona iloczynu.

Na rys. 6 przedstawiono kilka przykładów dobrego i złego zakreślania (sklejania) sąsiednich kratek tablicy. W zależności od wyboru końcowej postaci funkcji, symbol *x* może oznaczać zarówno „0” (dla iloczynu sum) jak i „1” (dla sumy iloczynów).



Rys. 6. Przykłady dobrego i złego zakreślania pól sąsiednich.

pamiętaj: - *współrzędne pól tablicy Karnaugh opisuj kodem Gray'a (refleksyjnym)*
- *skrajne pola zaznaczonych obszarów też muszą być dla siebie sąsiednimi*

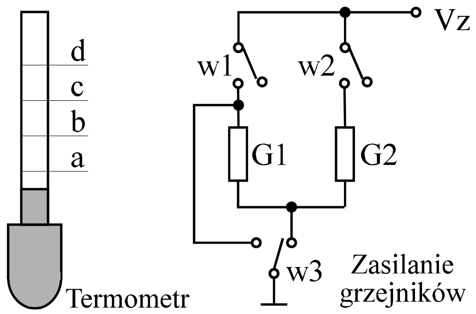
Przykład 1

Termometr kontaktowy (rys. 7) generuje sygnały *a, b, c, d* gdy temperatura przekroczy odpowiednie wartości. Zaprojektować układ sterujący wyłącznikami *w1, w2, w3*, tak aby spełnić wymagane warunki włączeń grzejników w zależności od temperatury *t*:

- $t < ta$ - oba grzejniki włączone równolegle ;
- $ta \le t < tb$ - włączony tylko grzejnik G1 ;
- $tb \le t < tc$ - włączony tylko grzejnik G2 ;
- $tc \le t < td$ - oba grzejniki włączone szeregowo ;
- $td < t$ - oba grzejniki wyłączone.

Styki wyłączników narysowane są w pozycji 0.

Przekroczenie każdego progu temperatury sygnalizowane jest stanem „1” odpowiednich czujników.



Rys. 7. Przykład 1.

<u>*ROZWIĄZANIE:*</u>

1. Rysujemy pierwotną (wynikającą bezpośrednio z warunków zadania) tablicę stanów wejść i odpowiadających im stanów wyjść. Każdej (możliwej do wystąpienia) kombinacji stanów czujników *a, b, c, d* przypisujemy odpowiedni stan wyłączników *w1, w2, w3*. Np.: pierwszy wiersz - żaden z czujników nie jest aktywny, więc włącz *w1* oraz *w2* a wyłącz *w3*.

wejścia				wyjścia		
a	*b*	*c*	*d*	*w1*	*w2*	*w3*
0	0	0	0	1	1	0
1	0	0	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	1	1
1	1	1	1	0	0	-

$t < ta$,	oba włączone równolegle
$ta \le t < tb$,	włączony tylko G1
$tb \le t < tc$,	włączony tylko G2
$tc \le t \le td$,	oba włączone szeregowo
$td \le t$,	oba wyłączone
pozostałe przypadki pomijamy ...	

Rys. 8. Tablica stanów we/wy dla *Przykładu 1.*

2. Osobno, dla kolejnych wyjść *w1, w2, w3*, rysujemy trzy tablice Karnaugh i przepisujemy wartości z tablicy pierwotnej. Każdej kratce odpowiada dokładnie jeden stan wejść (czujników *a, b, c, d*) wraz z przypisanym mu stanem odpowiedniego wyjścia (danego wyłącznika). Minimalizując poszczególne funkcje logiczne otrzymujemy funkcje wyjść wyłączników.

ab \ cd	00	01	11	10
00	1	-	-	-
01	-	-	-	-
11	0	-	0	0
10	1	-	-	-

$$w1 = \overline{b}$$

ab \ cd	00	01	11	10
00	1	-	-	-
01	-	-	-	-
11	1	-	0	1
10	0	-	-	-

$$w2 = \overline{a} + b\overline{d}$$

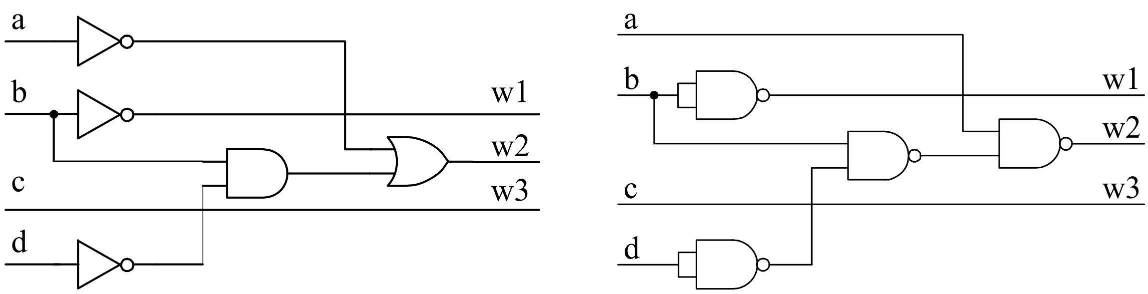
ab \ cd	00	01	11	10
00	0	-	-	-
01	-	-	-	-
11	0	-	-	1
10	0	-	-	-

$$w3 = c$$

Rys. 9. Minimalizacja funkcji logicznych wyłączników *w1,w2,w3*.

3. Po zakończeniu minimalizacji rysujemy odpowiednie schematy logiczne. Na rys. 10. przedstawiono dwa warianty schematów, wykorzystujące różne funktory logiczne.

a) b)



Rys. 10. Schematy układu sterowania: a) postać bezpośrednia, b) z bramek NAND.