Opracowanie:
18.X.2005
06.XII.2005
16.X.2006

Katedra Automatyki i Biomechaniki P.Ł.

LABORATORIUM PODSTAW AUTOMATYKI

Ćwiczenie E: ***Układy logiczne kombinacyjne***

15. UKŁADY LOGICZNE AUTOMATYKI

15.1. Wstęp

Od wielu lat układy logiczne znajdują w technice coraz szersze zastosowanie. W oparciu o układy logiczne rozwinęły się takie dziedziny techniki, jak:

- elektroniczna technika obliczeniowa,
- teletechnika,
- miernictwo elektryczne,
- elektronika,
- technika sterowania procesami przemysłowymi i badawczymi oraz szereg pokrewnych dziedzin, wykorzystujących zasadniczą cechę układów logicznych jaką jest istnienie dwóch stanów przesyłanego sygnału.

Założenie dwóch wartości (stan 0 i stan 1) nominalnych sygnału decyduje o projektowaniu, konstrukcji i opisie układu logicznego (zwanego także układem przełączającym).

Układy logiczne mogą być realizowane:

- elektronicznie,
- elektromechanicznie,
- pneumatycznie.

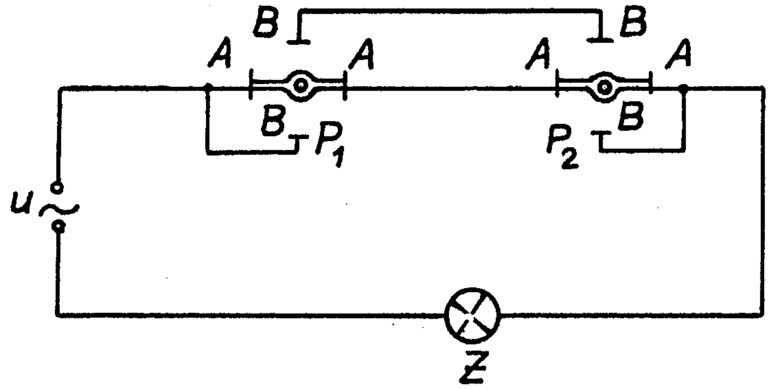
Stan 1 jest określony istnieniem napięcia elektrycznego lub ciśnienia na danym poziomie, np.: +5 V, 2,5 bara.

Stan 0 z kolei jest określony najczęściej poziomem sygnału bliskim zera, np. 0,25 V, 0,05 bara lub ewentualnie równym 0.

15.2. Sposób opisu układów przełączających

Przykład 15.1

Wyłącznik hotelowy (rys. 15.1) jest zbudowany w ten sposób, że dowolnym przełącznikiem (P_1 lub P_2) możemy zgasić lub zapalić światło w długim korytarzu. Oczywiście żarówka będzie się świecić, gdy oba przełączniki będą jednocześnie w położeniu A lub B. Jeżeli przełączniki będą w różnych położeniach, to żarówka nie będzie się świecić.



Rys. 15.1.

Utwórzmy tabelę funkcji, gdzie położeniu A przełączników P_1 i P_2 odpowiada stan 0, a położeniu B - stan 1 oraz stanowi "żarówka nie świeci" odpowiada stan Z = 0, zaś stanowi "żarówka świeci" odpowiada stan Z = 1; otrzymujemy

P_1	P_2	Żarówka
A	A	świeci
A	B	nie świeci
B	A	nie świeci
B	B	świeci

co odpowiada zapisowi:

P_1	P_2	Z
0	0	1
0	1	0
1	0	0
1	1	1

Układ oświetlenia korytarza żarówką realizuje pewną funkcję dwóch zmiennych niezależnych (położenia przełączników) wg tabeli stanów podanej wyżej.

Przykład 15.2

Otworzenie drzwi zamkniętych na 2 zamki jest możliwe tylko przy jednoczesnym ustawieniu w położeniu "otwarte" obu zamków. Jako zmienne niezależne traktujemy zamki (Z_1, Z_2), a ich stany odpowiednio: zamknięty - 0, otwarty - 1. Zmienną zależną jest sytuacja drzwi (D) - ich stan: zamknięte - 0, otwarte - 1. Jest to realizowana funkcja $D = f(Z_1, Z_2)$ opisana tabelą stanów:

Z_1	Z_2	D
0	0	0
1	0	0
0	1	0
1	1	1

Dla drzwi zamykanych na 3 zamki tabela stanów będzie:

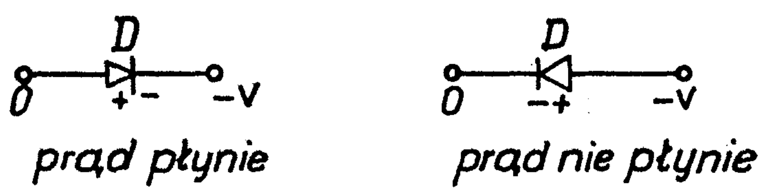
Z_1	Z_2	Z_3	D
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	1

$$D = f(Z_1, Z_2, Z_3)$$

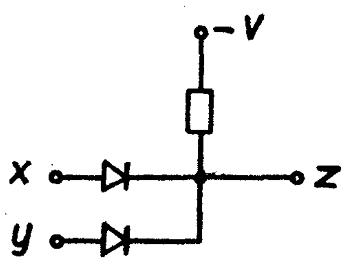
<u>Przykład 15:3</u>

Pojedyncza dioda (rys. 15.2) wstawiona w obwód prądu stałego może go przepuszczać albo stanowić dla tego obwodu przerwę (zaporę), jeżeli zostanie ustawiona zaporowo. Dla przepływu prądu jest wówczas realizowana funkcja

$$P = f(D).$$



Rys. 15.2



Rys. 15.3

D	P
0	0
1	1

gdzie stany diody D są oznaczone: 0 - dioda ustawiona zaporowo, 1 - dioda ustawiona w kierunku przewodzenia; stany obwodu prądu P zaś: 0 - prąd nie płynie, 1 - prąd płynie.

Dla układu (obwodu elektrycznego - rys. 15.3), gdzie: x, y - zmienne niezależne, z - zmienna zależna, zmienne te mogą przyjmować stany: stan 0 odpowiada potencjałowi V (potencjał zasilania), stan 1 odpowiada potencjałowi 0V; realizowana jest funkcja z = f(x, y).

Tabela stanów tej funkcji jest następująca:

x	y	z
0	0	0
1	0	1
0	1	1
1	1	1

Z podanych przykładów wynika postać tabelek dla 1, 2 i 3 zmiennych niezależnych; są one:

dla 1 zmiennej

x
0
1

dla 2 zmiennych

x	y
0	0
0	1
1	0
1	1

dla 3 zmiennych

x	y	z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

W tabelkach tych są zawarte wszystkie możliwości, a więc zbiór zamknięty kombinacji zmiennych niezależnych. Jeżeli jest znany schemat układu przełączającego, to z jego zasady działania można napisać odpowiedź układu, czyli wyznaczyć funkcję jaką tworzą zmienne niezależne, tak jak to było pokazane w przykładach. Powstaje zatem pytanie: ile różnych funkcji można utworzyć w zależności od ilości zmiennych niezależnych? Na przykład dla 1 zmiennej mamy 4 funkcje;

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

posiadają one swoje nazwy:

f_0 - stała 0.

f_1 - tożsamość,

f_2 - negacja,

f_3 - stała 1.

Dla 2 zmiennych można utworzyć 16 funkcji.

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Wprawdzie wszystkie funkcje 2 zmiennych posiadają swoje nazwy i bywają stosowane, jednak zasadnicze znaczenie mają funkcje:

f_8 - negacja sumy (funkcja Peirce'a) - NOR,

f_{14} - negacja iloczynu (funkcja Sheffera) - NAND;

stanowią one bowiem systemy funkcjonalne pełne i jako takie znalazły zastosowanie w praktyce.

Pojęcie "system funkcjonalnie pełny" - oznacza, że na bazie funkcji, która stanowi ten system można stosując rachunek Boole'owski utworzyć dowolną funkcję wielu zmiennych i tak np. przy wykorzystaniu jako funkcji bazowej f_8 (czyli negacji sumy $\overline{x + y}$) można zapisać funkcję f_{13}

$$f_{13} = \overline{x} + y = \overline{\overline{(\overline{(x + 0)} + y)} + 0};$$

aby jednak zrozumieć ten zapis należy poznać zasady algebry Boole'a i jej podstawowe operacje.

15.3. Podstawowe operacje i zasady rachunku Boole'a

Iloczyn logiczny dwóch zmiennych przyjmuje wartości:

$$\begin{aligned} 0 \cdot 0 &= 0 \\ 0 \cdot 1 &= 0 \\ 1 \cdot 0 &= 0 \\ 1 \cdot 1 &= 1 \end{aligned} \tag{15.1}$$

Ogólnie iloczyn logiczny wynosi 1 tylko wtedy, jeżeli wszystkie czynniki są równe 1; stąd wynikają własności:

$$a \cdot 0 = 0, \tag{15.2}$$

$$a \cdot 1 = a, \tag{15.3}$$

gdzie a - zmienna o dowolnej wartości (0 lub 1).

Suma logiczna dwóch zmiennych wynosi:

$$\begin{aligned} 0 + 0 &= 0, \\ 0 + 1 &= 1, \\ 1 + 0 &= 1, \\ 1 + 1 &= 1. \end{aligned} \tag{15.4}$$

Ogólnie suma logiczna jest równa 0 wtedy i tylko wtedy, jeżeli wszystkie składniki sumy są równe 0. Wynikają stąd własności:

$$a + 0 = a, \tag{15.5}$$

$$a + 1 = 1, \tag{15.6}$$

gdzie zmienna a może przyjmować dowolną wartość.

Negacja

$$\bar{0} = 1 \text{ (czytaj: nie 0 równe 1)},$$
$$\bar{1} = 0 \text{ (czytaj: nie 1 równe 0)}; \qquad (15.7)$$

ogólnie

$$\bar{a} \neq a, \qquad (15.8)$$

natomiast podwójna negacja

$$\bar{\bar{a}} = a. \qquad (15.9)$$

Łatwo wykazać jest związki:

$$\bar{a} \cdot a = 0, \qquad (15.10)$$

$$\bar{a} + a = 1. \qquad (15.11)$$

Dla rachunku zerojedynkowego obowiązują następujące reguły:

1) prawo łączności iloczynu

$$a \cdot b \cdot c = (a \cdot b) \cdot c = a \cdot (b \cdot c), \qquad (15.12)$$

2) prawo łączności sumy

$$a + b + c = (a + b) + c = a + (b + c), \qquad (15.13)$$

3) prawo przemienności iloczynu

$$a \cdot b = b \cdot a, \qquad (15.14)$$

4) prawo przemienności sumy

$$a + b = b + a, \qquad (15.15)$$

5) prawo powtórzenia iloczynu

$$a \cdot a = a, \qquad (15.16)$$

6) prawo powtórzenia sumy

$$a + a = a, \qquad (15.17)$$

7) prawo rozdzielności iloczynu względem sumy

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c), \qquad (15.18)$$

8) prawo rozdzielności sumy względem iloczynu

$$a + (b \cdot c) = (a + b)\cdot(a + c); \text{*)} \qquad (15.19)$$

korzystając z praw 7) i 8) można udowodnić związki:

$$ab + a\bar{b} = a(b + \bar{b}) = a\cdot 1 = a, \qquad (15.20)$$

$$(a + b)(a + \bar{b}) = a + (b \cdot \bar{b}) = a + 0 = a; \qquad (15.21)$$

są to tzw. "reguły sklejania".

Zasadnicze znaczenie w rachunku Boole owskim mają prawa de Morgana dla dwóch zmiennych:

$$\overline{a \cdot b} = \bar{a} + \bar{b}, \qquad (15.22)$$

$$\overline{a + b} = \bar{a} \cdot \bar{b}. \qquad (15.23)$$

Prawa de Morgana są ważne również dla wielu zmiennych:

$$\overline{a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_n} = \overline{a_1} + \overline{a_2} + \overline{a_3} + \ldots + \overline{a_n}, \qquad (15.24)$$

$$\overline{a_1 + a_2 + a_3 + \ldots + a_n} = \overline{a_1}\cdot\overline{a_2}\cdot\overline{a_3}\cdot \ldots \cdot\overline{a_n}. \qquad (15.25)$$

Stosując w (15.24) i (15.25) negację obu stron równania otrzymamy:

$$\overline{\overline{a_1} + \overline{a_2} + \overline{a_3} + \ldots + \overline{a_n}} = a_1\cdot a_2\cdot a_3\cdot \ldots \cdot a_n, \qquad (15.26)$$

$$\overline{\overline{a_1}\cdot\overline{a_2}\cdot\overline{a_3}\cdot \ldots \cdot\overline{a_n}} = a_1 + a_2 + a_3 + \ldots + a_n; \qquad (15.27)$$

stąd widać, że dowolny wzór można przedstawić za pomocą sumy logicznej i negacji albo iloczynu logicznego i negacji. Powyższy wniosek może służyć jako uzasadnienie wcześniej stwierdzonego faktu, że funkcja f_8 (NOR - z ang. "Not or", czyli "nie lub", czyli zaprzeczenie alternatywy, tzn. $\overline{x + y}$) stanowi system funkcjonalnie pełny.

Funkcja f_{14} (NAND - z ang. "Not and", czyli "nie i", czyli zaprzeczenie koniunkcji, tzn. $\overline{xy}$) również spełnia kryteria systemu funkcjonalnie pełnego.

Przykład 15.4

Przedstawić w postaci iloczynowej funkcję $f = a + b + a \cdot c$

---

*) Korzystając z reguły 8 można przykładowo wykazać, że:
$a + b\bar{a} = (a+b)(a+\bar{a}) = (a+b)\cdot 1 = a+b$

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

$$= (\overline{\overline{a + b}})(\overline{\overline{a + b + c}}) = (\overline{\bar{a} \cdot \bar{b}})(\overline{\bar{a} \cdot \bar{b} \cdot \bar{c}}). \qquad (15.28)$$

Przykład 15.5

Przedstawić w postaci sumy funkcję $f = (ab + bc)a$.

$$f = (ab + bc)a = ab \cdot a + bc \cdot a = ab + abc =$$

$$= \overline{\overline{ab}} + \overline{\overline{abc}} = (\overline{\bar{a} + \bar{b}}) + (\overline{\bar{a} + \bar{b} + \bar{c}}). \qquad (15.29)$$

15.4. Synteza funkcji logicznych

Do algebraicznego zapisu dowolnych funkcji logicznych używa się symboli funkcji podstawowych, czyli znaku "+" dla alternatywy (sumy logicznej) i znaku „•" (razy) dla koniunkcji (iloczynu logicznego) oraz znaku „-" (nie) nad symbolem zmiennej lub funkcji kilku zmiennych dla funkcji negacji (zaprzeczenia logicznego).

Znane są 2 metody zapisu funkcji:

1) metoda postaci kanonicznej alternatywnej (składa się z elementarnych koniunkcji),

2) metoda postaci kanonicznej koniunkcyjnej (składa się z elementarnych alternatyw).

Omówimy je na przykładzie

Przykład 15.6

Dana jest funkcja $y = f(a, b, c)$ w postaci tabelki stanów

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

Zapis funkcji w postaci kanonicznej alternatywnej (metoda 1) przedstawia się następująco

$$y = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,c + a\,\bar{b}\,\bar{c} + a\,\bar{b}\,c + a\,b\,\bar{c} + a\,b\,c. \qquad (15.30)$$

Jak widać poszczególne składniki alternatywy są iloczynami logicznymi zmiennych a, b, c (lub ich negacji) ułożonych na podstawie tabeli stanów dla wartości funkcji $y = 1$ w sposób taki, że jeżeli wartość zmiennej w tabeli jest 0, to w iloczynie logicznym odpowiada mu zapis zmiennej zanegowanej, natomiast jeśli w tabeli wartość zmiennej jest równa 1, to w iloczynie logicznym jest po prostu zapis tej zmiennej. Każdy iloczyn logiczny ma w ten sposób wartość logiczną równą 1, ponieważ przedstawia on sobą wyrażenie typu $1 \cdot 1 \cdot 1$.

Zapis funkcji y w postaci kanonicznej koniunkcyjnej jest następujący

$$y = (a + \bar{b} + c)(a + \bar{b} + \bar{c}). \qquad (15.31)$$

Zapis ten powstaje jako koniunkcja sum logicznych. Każda suma logiczna w tym zapisie odpowiada stanowi 0 funkcji y w tabeli stanów i powstaje w sposób taki, że jeżeli w tabeli stanów zmienna przyjmuje wartość 0, to w sumie logicznej odpowiada jej zapis zmiennej, natomiast jeśli posiada wartość 1, to należy w sumie logicznej ją zanegować. Każda suma logiczna ma w ten sposób postać $(0 + 0 + 0)$.

Zapisy funkcji (15.30) i (15.31) są tożsamościowo równe, co można łatwo udowodnić jednocześnie skracając zapis do minimalnego.

Z funkcji (15.30) wynika

$$\begin{aligned} y &= (\bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c) + (a\bar{b}\bar{c} + a\bar{b}c) + (ab\bar{c} + abc) = \\ &= \bar{a}\bar{b}(\bar{c} + c) + a\bar{b}(\bar{c} + c) + ab(\bar{c} + c) = \\ &= \bar{a}\bar{b} + a\bar{b} + ab = (\bar{a}\bar{b} + a\bar{b}) + (a\bar{b} + ab) = \\ &= (\bar{a} + a)\bar{b} + a(\bar{b} + b) = \bar{b} + a. \end{aligned} \qquad (15.32)$$

Z funkcji (15.31) wynika przy wykorzystaniu wzoru (15.21)

$$y = (a + \bar{b} + c)(a + \bar{b} + \bar{c}) = a + \bar{b} + (c \cdot \bar{c}) = a + \bar{b}. \qquad (15.33)$$

Zapisu funkcji f(a, b, c) podanej w przykładzie można dokonać również w tzw. tablicy Karnaugha. Ma ona postać:
[karno]

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

lub

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

W tablicy umieszcza się wartość funkcji f(a, b, c) dla wartości zmiennych a, b, c zaznaczonych na zewnątrz tabeli. Przeprowadzając następnie minimalizację zapisu funkcji na podstawie tablicy Karnaugha otrzymujemy:

a) minimalizując wg stanów 1

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

cztery zewnętrzne jedynki „sklejone" razem

$y = a + \overline{b}$

b) minimalizując wg stanów 0

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

$y = a + \overline{b}$

Jak widać zapisy te są identyczne tożsamościowo z obiema postaciami kanonicznymi funkcji, co już wcześniej wykazano we wzorach (15.32) i (15.33). W celu zorientowania czytelnika w sposobie zapisu minimalnego funkcji na podstawie tablicy Karnaugha podane są dalej inne przykłady.

Uwaga: Należy też zwrócić uwagę, że wartości par zmiennych nie są przyjęte jako kolejne liczby dwójkowe (00, 01, 10, 11), ale wg kodu 00, 01, 11, 10 (tzw. kod Graya), gdzie sąsiednie kody (również skrajne) różnią się tylko na jednej pozycji. Wynika to ze stosowania reguł sklejania.

Przykład 15.7

Zminimalizować funkcję f(a, b, c) wg podanej tablicy Karnaugha.

Minimalizacja „jedynkowa"

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

$$y = \underset{\mathrm{I}}{\bar a\bar b} + \underset{\mathrm{II}}{ab} + \underset{\mathrm{III}}{\bar a c}$$

Każdej parze sąsiadujących jedynek obwiedzionych owalną linią na tablicy Karnaugha odpowiada jeden składnik w zapisie alternatywnym funkcji będący iloczynem dwóch zmiennych, które dla danej pary mają stałe wartości logiczne zaznaczone na zewnątrz tabeli Karnaugha, i tak:

- dla I pary stałe wartości posiadają zmienne a = 0 i b = 0; I parze odpowiada zatem składnik $\bar a\bar b$; jest to po prostu graficzna metoda sklejania 2 koniunkcji (wzór 15.20) $\bar a\bar b\bar c + \bar a\bar b c = \bar a\bar b(\bar c + c) = \bar a\bar b$;
- dla II pary stałe wartości mają zmienne a = 1 i b = 1 - odpowiada jej zatem składnik ab, ponieważ $ab\bar c + abc = ab(\bar c + c) = ab$;
- dla III pary stałe wartości mają zmienne a = 0 i c = 1 - odpowiada jej zatem składnik $\bar a c$, ponieważ $\bar a\bar b c + \bar a bc = \bar a c(\bar b + b) = \bar a c$.

Alternatywa, czyli suma logiczna wymienionych 3 składników jest minimalnym zapisem funkcji przedstawionej tablicą Karnaugha w powyższym przykładzie.

Przykład 15.8

Zminimalizować funkcję czterech zmiennych f(a, b, c, d) wg tablicy Karnaugha.

Minimalizacja „jedynkowa"

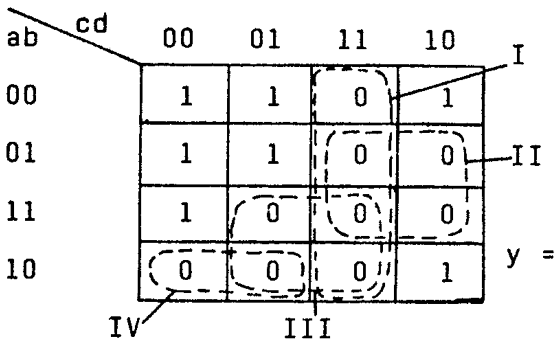
ab \ cd	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	1	0	0	0
10	0	0	0	1

$$y = \underset{\mathrm{I}}{\bar a\bar c} + \underset{\mathrm{II}}{b\bar c\bar d} + \underset{\mathrm{III}}{\bar b c\bar d}$$

Objaśnienie: każdej grupie sąsiadujących jedynek obwiedzionych linią przerywaną na tablicy Karnaugha odpowiada jeden składnik w zapisie alternatywnym będący iloczynem logicznym zmiennych, które dla danej grupy mają stałe wartości logiczne; i tak:

- dla grupy I (cztery jedynki) stałe wartości posiadają zmienne $a = 0$ i $c = 0$, zatem odpowiada jej zapis $\bar{a}\bar{c}$;
- dla grupy II (dwie jedynki) stałe wartości posiadają zmienne $b = 1$, $c = 0$ i $d = 0$, zatem zapis odpowiadający tej grupie będzie $b\bar{c}\bar{d}$;
- dla grupy III (dwie jedynki zewnętrzne) stałe wartości posiadają zmienne $b = 0$, $c = 1$ i $d = 0$, zatem odpowiada jej zapis $\bar{b}c\bar{d}$.

Minimalizacja „zerowa"



$$y = \underset{\mathrm{I}}{(\bar{c} + \bar{d})}\underset{\mathrm{II}}{(\bar{b} + \bar{c})}\underset{\mathrm{III}}{(\bar{a} + \bar{d})}\underset{\mathrm{IV}}{(\bar{a} + b + c)}$$

Objaśnienie: każdej grupie sąsiadujących zer obwiedzionych linią przerywaną na tablicy Karnaugha odpowiada jeden czynnik w zapisie koniunkcyjnym funkcji, będący sumą logiczną zmiennych, które mają stałe wartości logiczne dla danej grupy oznaczonej na zewnątrz tablicy Karnaugha; tak więc w powyższym przykładzie:

- dla grupy I zmienne posiadające stałe wartości są: $c = 1$ i $d = 1$, odpowiadający jej zapis jest więc $(\bar{c} + \bar{d})$; wynika to z reguły sklejania, według wzoru (15.21)

wg [15.19]

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

$$= [(a + \bar{c} + \bar{d}) + (b \cdot \bar{b})][(\bar{a} + \bar{c} + \bar{d}) + (b \cdot \bar{b})] =$$

$$= (a + \bar{c} + \bar{d})(\bar{a} + \bar{c} + \bar{d}) =$$

$$= (\bar{c} + \bar{d}) + (a \cdot \bar{a}) = (\bar{c} + \bar{d}),$$

- dla II grupy zmienne o stałych wartościach są: $b = 1$ i $c = 1$ – zapis więc będzie $(\bar{b} + \bar{c})$,

- dla III grupy zmienne o stałych wartościach są: a = 1 i d = 1 - zapis więc będzie $(\bar{a} + \bar{d})$,
- dla IV grupy zmienne o stałych wartościach są: a = 1, b = 0 i c = 0 - zapis będzie $(\bar{a} + b + c)$.

Przykład 15.9

Przedstawić minimalizację zapisu funkcji f(a, b, c, d) dla podanej tablicy Karnaugha.

Minimalizacja „jedynkowa"

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

$$y = \underset{\mathrm{I}}{\bar{a}b} + \underset{\mathrm{II}}{\bar{b}\bar{d}} + \underset{\mathrm{III}}{bcd}$$

Na zwrócenie uwagi zasługuje w tym przykładzie grupa II złożona z czterech jedynek umieszczonych w rogach tablicy, dla której zmienne posiadające stałe wartości logiczne to b = 0 i d = 0.

Minimalizacja „zerowa"

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

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

Funkcje uzyskane z minimalizacji „zerowej" i „jedynkowej" są tożsamościowo równe, co można zawsze wykazać stosując reguły rachunku Boole´a.

Na podstawie przytoczonych przykładów można zorientować się, że grupy jedynek lub zer mogą obejmować regularne pola złożone wyłącznie z 2, 4 lub 8*) kratek tablicy Karnaugha, będące prostokątami lub kwadra-

*) Należy odnaleźć pola obejmujące możliwie najwięcej kratek, bo wtedy minimalizacja jest najlepsza, a uzyskana funkcja najprostsza.

tami bądź też mogą być one rozdzielone na połówki i wówczas te połówki grupy muszą przylegać do zewnętrznych boków tablicy naprzeciwko siebie albo też może być to grupa obejmująca cztery kratki rogowe tablicy Karnaugha (jak w przykładzie 15.9). *)

Przy okazji można dostrzec, że pola te często zachodzą na siebie. Fakt ten ma swoje pozytywne znaczenie bowiem zachodzenie na siebie pól obejmujących grupy jednakowych wartości logicznych jest zjawiskiem korzystnym - przyczynia się bowiem do eliminacji niekorzystnych zjawisk, takich jak wyścig lub hazard. Brak takiego zachodzenia na siebie pól może być niekiedy sztucznie likwidowany przez tworzenie dodatkowych grup łączących sąsiadujące, lecz nie zachodzące na siebie pola, czyli innymi słowy wzbogacenie minimalnego zapisu funkcji o dodatkowy składnik lub czynnik nie wprowadzający zmian w opisie funkcji za to likwidujący niejednoznaczność funkcji w momencie zmiany stanu jednej ze zmiennych; np. dla przykładu 15.9 takimi składnikami przy minimalizacji jedynkami byłby składnik $\bar{a}\bar{d}$;

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

wówczas funkcja (już nie minimalna) miałaby zapis $y = \bar{a}b + \bar{b}\bar{d} + bcd + \bar{a}\bar{d}$;

Przy minimalizacji zerami rolę takiego „łącznika" pełniłby czynnik $(\bar{a} + \bar{b} + c)$;

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

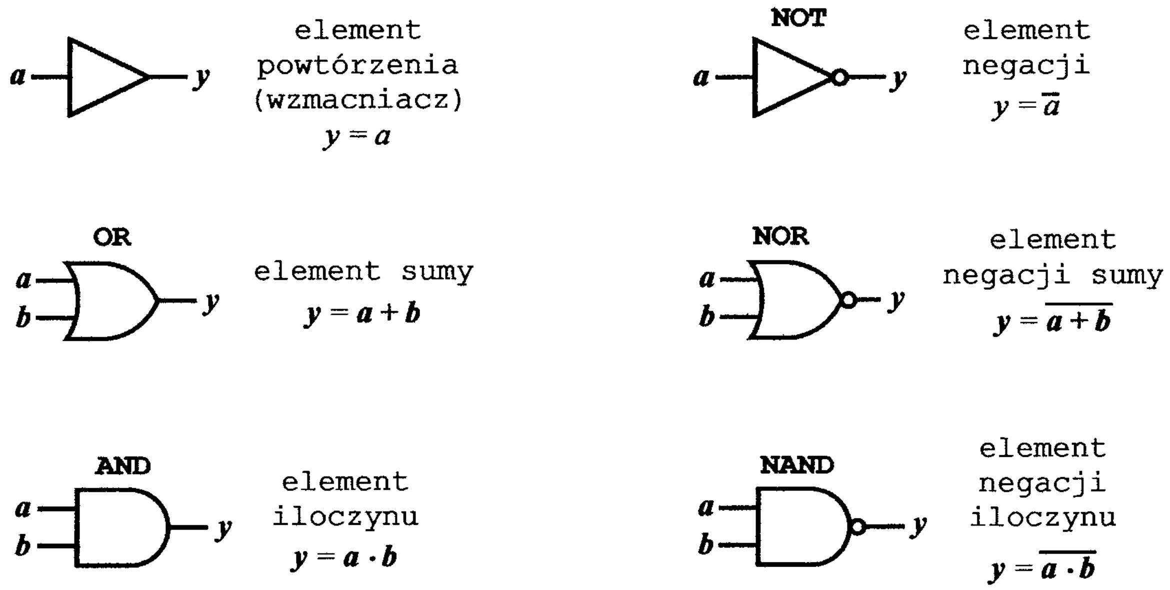
zapis funkcji byłby wtedy
$y = (b + \bar{d})(\bar{a} + \bar{b} + d)(\bar{a} + c + \bar{d})(\bar{a} + \bar{b} + c)$.

W przypadku opisu funkcji y pięcioma lub więcej zmiennymi minimalizacja jej za pomocą tabeli Karnaugha jest bardziej skomplikowana i nie będzie tu omówiona (patrz: Literatura).

---

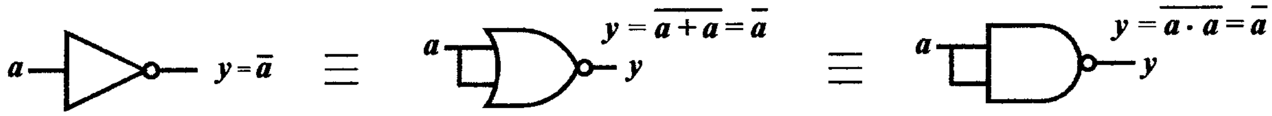
* pola nie mogą więc obejmować np. 3, 5, 6 czy 7 jedynek lub zer. Ale może wystąpić przypadek, że będzie tylko jedna jedynka lub jedno zero.

Do realizacji funkcji logicznych stosuje się elementy funkcji podstawowych, które oznacza się jak na rysunku 15.4.



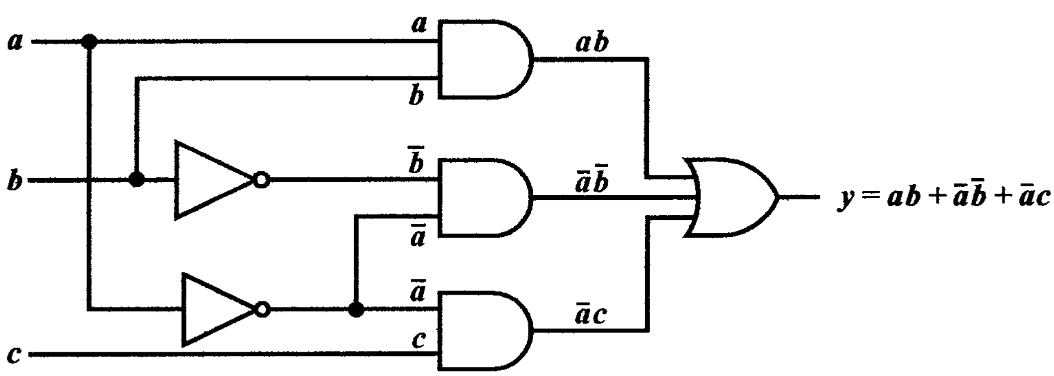
Rys. 15.4 - Podstawowe elementy logiczne (bramki logiczne)

Na powyższym rysunku elementy (bramki) **OR**, **NOR**, **AND** i **NAND** przedstawiono jako elementy 2-wejściowe. W praktyce są również stosowane elementy z większą liczbą wejść, np. 3, 4 itp., co będzie pokazane na dalszych rysunkach. Jak łatwo zauważyć, kółko przy wyjściu graficznego symbolu elementu oznacza negację realizowanej funkcji. Należy też zauważyć, że negację **NOT** można też zrealizować np. z dwuwejściowych elementów **NOR** lub **NAND**, łącząc razem ich wejścia, jak to pokazuje poniższy rys.15.4a (jest to ważne, gdyż w ćwiczeniu mamy do dyspozycji tylko NAND-y).



Rys. 15.4a - równoważne realizacje negacji z bramek NOR lub NAND

Schemat realizacji funkcji z przykładu 15.7 (uzyskanej z minimalizacji "jedynkowej") z wykorzystaniem podstawowych elementów logicznych przedstawia poniższy rys. 15.5.



Rys. 15.5 - realizacja funkcji z przykładu 15.7

Do realizacji funkcji logicznych stosuje się elementy funkcji podstawowych, które oznacza się tak jak na rysunku 15.4:

a) element powtórzenia $y = x$,

b) element negacji $y = \bar{x}$ (NOT),

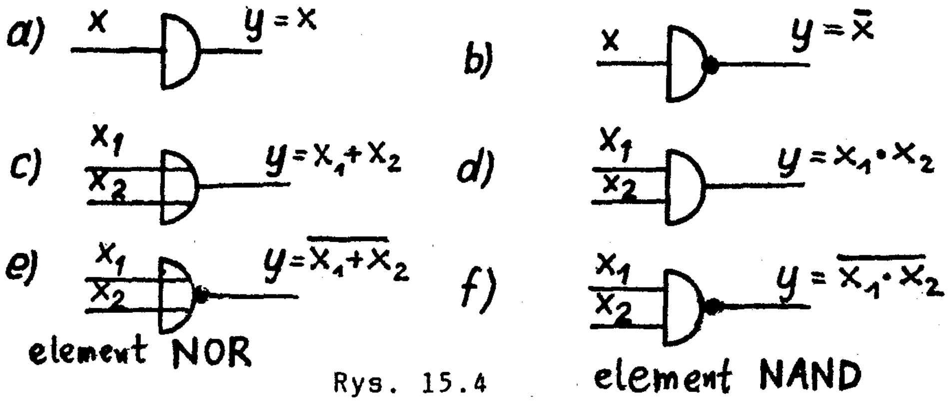
c) element sumy $y = x_1 + x_2$ (OR),

d) element iloczynu $y = x_1 \cdot x_2$ (AND),

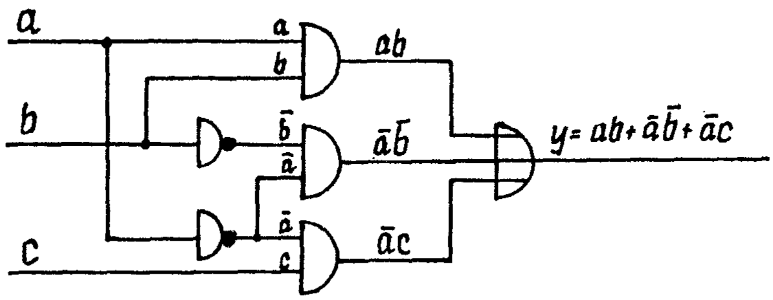
e) element negacji sumy (NOR) $y = \overline{x_1 + x_2}$,

f) element negacji iloczynu (NAND) $y = \overline{x_1 \cdot x_2}$.

Jak łatwo zauważyć, kropki na wyjściach elementów oznaczają negację realizowanej funkcji.



Rys. 15.4

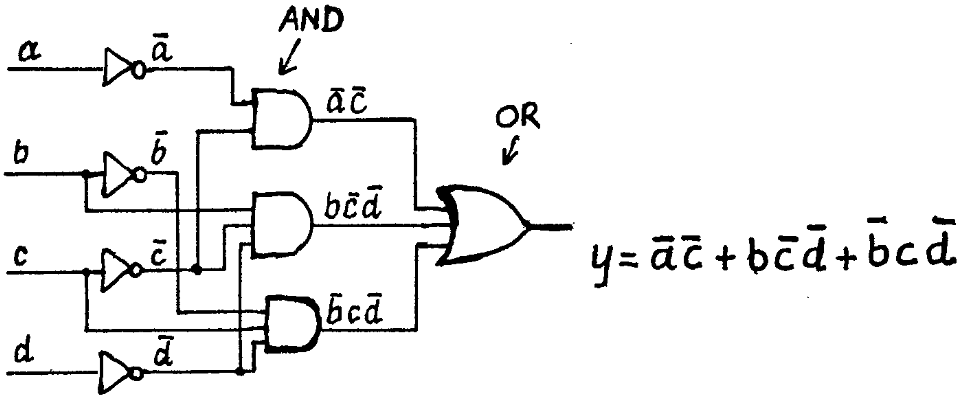


Rys. 15.5

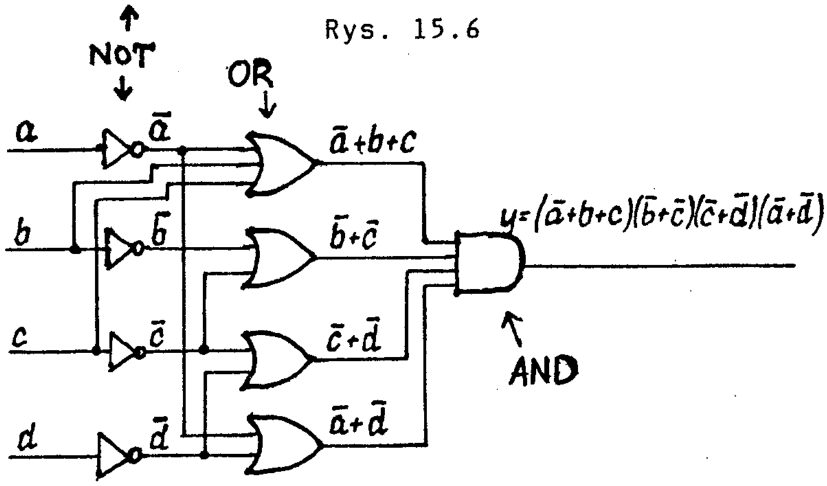
Na rysunku 15.4 c), d), e), f) przedstawiono symbole elementów 2-wejściowych. W praktyce są również stosowane elementy 3, 4 i więcej wejściowe.

Schemat realizacji funkcji z przykładu 15.7 przedstawia rysunek 15.5. Schematy realizacji funkcji z przykładu 15.8 przedstawiają rysunki 15.6 i 15.7.

Schematy realizacji funkcji z przykładu 15.8 przedstawiają rysunki 15.6 (minimalizacja „jedynkowa") oraz 15.7 (minimalizacja „zerowa") .



Rys. 15.6



Rys. 15.7

W technice są stosowane systemy funkcjonalnie pełne, czyli oparte na jednym bazowym elemencie. Do takich elementów zaliczają się NOR i NAND. Aby zbudować schemat używając wyłącznie elementów, np. NOR należy dokonać przekształcenia zapisu funkcji do postaci zawierającej wyłącznie zaprzeczenie sum logicznych; zatem np, funkcja z przykładu 15.9 o postaci

$$y = \bar a b + \bar b \bar d + bcd + \bar a \bar d$$

musi zostać przekształcona przez stosowanie podwójnej negacji i praw de Morgana; zatem

$$y = \overline{\overline{\bar a b}} + \overline{\overline{\bar b \bar d}} + \overline{\overline{bcd}} + \overline{\overline{\bar a \bar d}} = \overline{(\bar{\bar a} + \bar b)} + \overline{(\bar{\bar b} + \bar{\bar d})} +$$
$$+ \overline{(\bar b + \bar c + \bar d)} + \overline{(\bar{\bar a} + \bar{\bar d})} =$$
$$= \overline{\overline{\overline{(a + \bar b)} + \overline{(b + d)} + \overline{(\bar b + \bar c + \bar d)} + \overline{(a + d)}}}$$

zapisowi temu odpowiada schemat z rysunku 15.8.

Funkcję o postaci

$$y = (b + \overline{d})(\overline{a} + \overline{b} + d)(\overline{a} + c + \overline{d})(\overline{a} + \overline{b} + c)$$

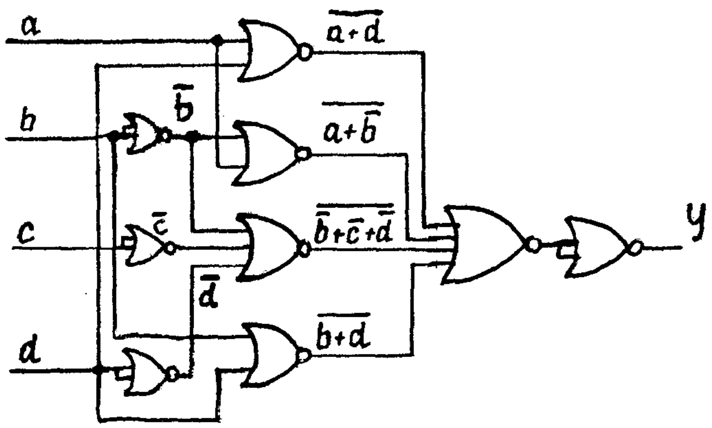
należy przekształcić do postaci

$$y = \overline{\overline{(b + \overline{d})(\overline{a} + \overline{b} + d)(\overline{a} + c + \overline{d})(\overline{a} + \overline{b} + c)}} =$$

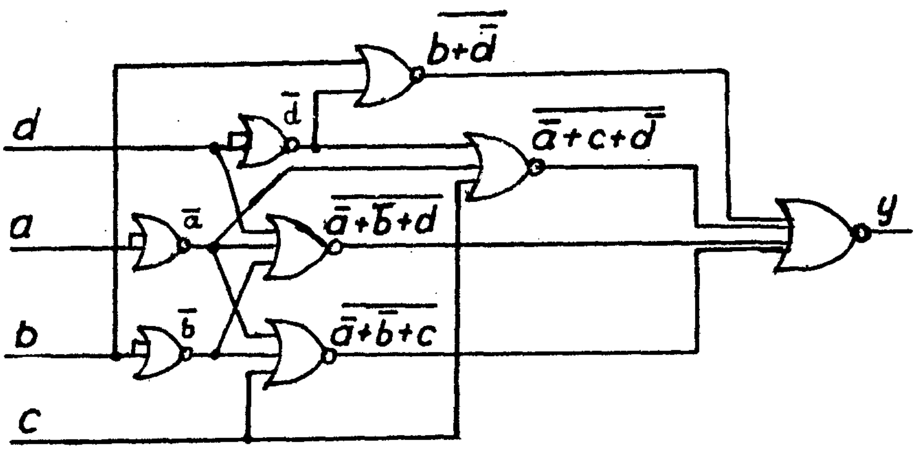
$$= \overline{\overline{(b + \overline{d})} + \overline{(\overline{a} + \overline{b} + d)} + \overline{(\overline{a} + c + \overline{d})} + \overline{(\overline{a} + \overline{b} + c)}}$$

wówczas odpowiada jej schemat z rysunku 15.9.

Uwaga: Negacje zbudowano z 2-wejściowych bramek NOR



Rys. 15.8



Rys. 15.9

Budując schemat dla realizacji funkcji z elementów NAND należy zapis funkcji przekształcić do postaci zawierającej wyłącznie zaprzeczenia iloczynów logicznych; zatem funkcje z przykładu 15.9 przekształcamy następująco:

$$y = \bar{a}b + \bar{b}\bar{d} + bcd + \bar{a}\bar{d} = \overline{\overline{\bar{a}b + \bar{b}\bar{d} + bcd + \bar{a}\bar{d}}} =$$

$$= \overline{\overline{(\bar{a}b)}\,\overline{(\bar{b}\bar{d})}\,\overline{(bcd)}\,\overline{(\bar{a}\bar{d})}}, \quad \text{/rys. 15.10}$$

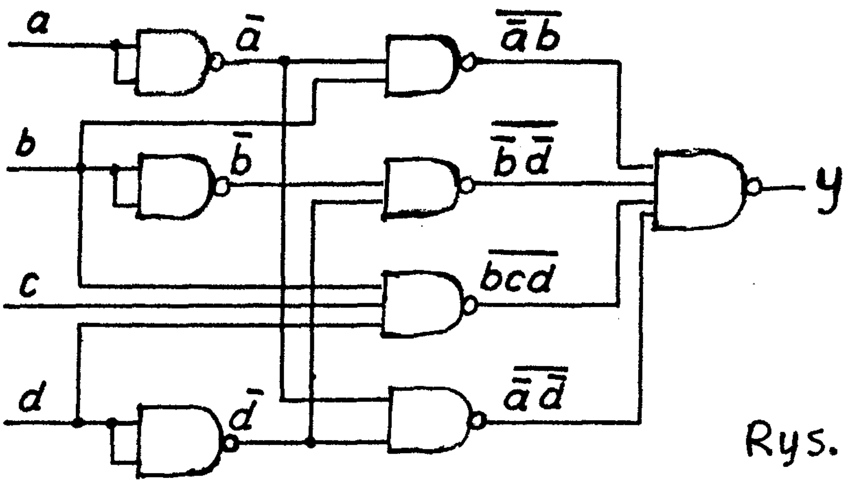
$$y = (b + \bar{d})(\bar{a} + \bar{b} + d)(\bar{a} + c + \bar{d})(\bar{a} + \bar{b} + c) =$$

$$= \overline{\overline{(b + \bar{d})}}\;\overline{\overline{(\bar{a} + \bar{b} + d)}}\;\overline{\overline{(\bar{a} + c + \bar{d})}}\;\overline{\overline{(\bar{a} + \bar{b} + c)}} =$$

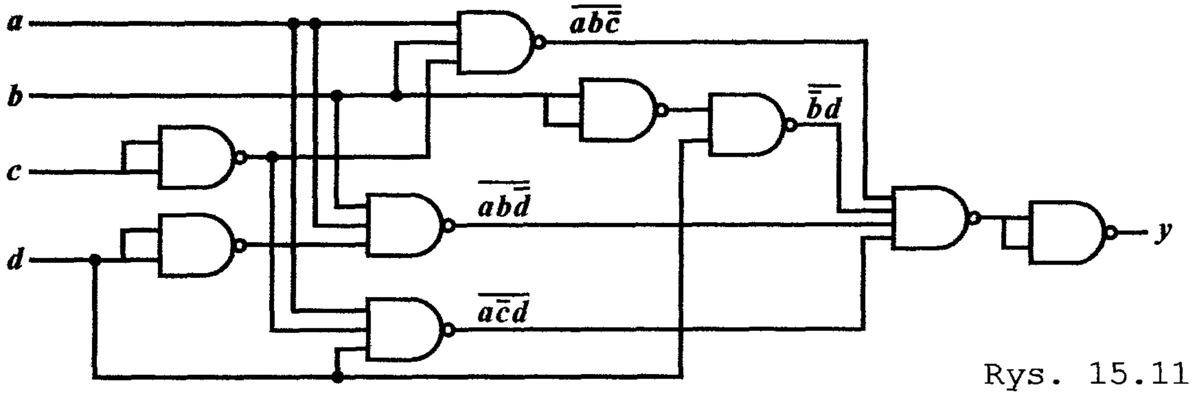
$$= \overline{(\bar{b}\bar{\bar{d}})}\;\overline{(\bar{\bar{a}}\bar{\bar{b}}\bar{d})}\;\overline{(\bar{\bar{a}}\bar{c}\bar{\bar{d}})}\;\overline{(\bar{\bar{a}}\bar{\bar{b}}\bar{c})} =$$

$$= \overline{\overline{\overline{(\bar{b}d)}\,\overline{(ab\bar{d})}\,\overline{(a\bar{c}d)}\,\overline{(ab\bar{c})}}}; \quad \text{/rys. 15.11}$$

ich schematy odpowiadające końcowym zapisom funkcji przedstawiono na rysunkach 15.10 i 15.11.

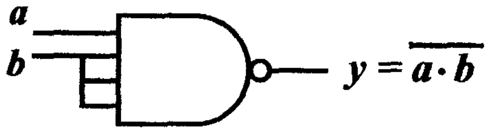


Rys. 15.10



Rys. 15.11

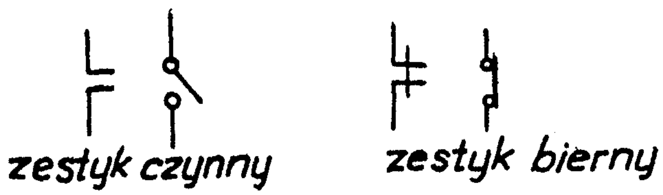
Podczas praktycznej realizacji funkcji z dostępnych elementów logicznych może się okazać, że wejścia niektórych bramek pozostaną niepodłączone (np. mamy do dyspozycji bramkę 4-wejściową, a potrzebne są tylko 2 wejścia). W takiej sytuacji najlepiej zewrzeć niewykorzystane wejścia razem z dowolnym sygnałem podanym na jedno z wykorzystywanych wejść, jak pokazano to na poniższym rysunku.



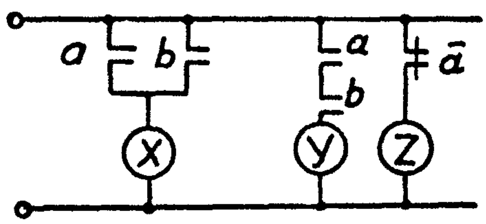
Niewykorzystane wejścia podłączone do jednego z wykorzystanych wejść

W praktyce układy logiczne realizuje się współcześnie w technologii TTL z wymaganym napięciem zasilania 5V (np. układy cyfrowe serii 74LS... o małym poborze mocy i opóźnieniu zbocza 9ns) lub w technologii CMOS (np. popularna seria 40... o dopuszczalnym napięciu zasilania 3-15V i opóźnieniu ok. 40ns). Tablica z układami logicznymi wykorzystywanymi w ćwiczeniu zbudowana jest z wykorzystaniem standardowych układów TTL.

Funkcje logiczne można realizować również na drodze elektromechanicznej, wykorzystując elementy przekaźnikowe stykowe. Pojedynczy element przekaźnikowy może realizować tylko funkcję powtórzenia lub negacji w zależności jak jest zbudowany. Przekaźnik z zestykiem czynnym realizuje funkcję powtórzenia, tzn. zestyk jest zwarty, gdy na cewkę przekaźnika jest podane napięcie. Przekaźnik z zestykiem biernym realizuje funkcję negacji, tzn. zestyk jest zwarty wówczas, gdy na cewce przekaźnika nie ma napięcia. W chwili, gdy na cewce przekaźnika jest napięcie zestyk jest rozwarty.



Rys. 15.12



Rys. 15.13

Stosując elementy przekaźnikowe można zbudować dowolną funkcję logiczną.

Na rysunku 15.12 przedstawiono symbole stosowane na schemacie układów przekaźnikowych zestyków czynnych i biernych. Na rysunku 15.13 przedstawiono schemat realizacji 3 podstawowych funkcji logicznych przy wykorzystaniu elementów przekaźnikowych:

$$x = a + b,$$

$$y = ab,$$

$$z = \bar{a}.$$

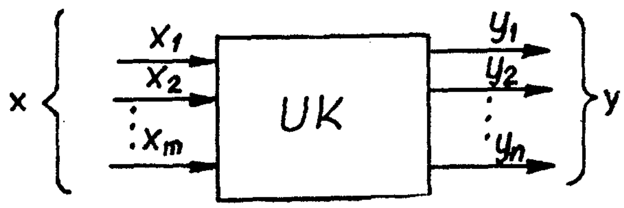
W praktyce elementy przekaźnikowe posiadają zarówno zestyki czynne, jak i zestyki bierne sterowane jedną cewką.

15.5. Układy kombinacyjne

Układem kombinacyjnym nazywa się taki układ przełączający, w którym sygnały wyjściowe, inaczej stan jego wyjść Y, zależą wyłącznie od kombinacji sygnałów wejściowych, tzn. od stanu jego wejść X. Działanie takiego układu nie zależy od czasu. Na aktualny stan wyjścia Y(t) nie ma wpływu jego stan poprzedni Y(t - T). Można więc napisać

$$Y = f(X), \qquad (15.34)$$

gdzie: Y - stan wszystkich wyjść układu $(y_1, y_2, \ldots, y_n)$,
X - stan wszystkich wejść układu $(x_1, x_2, \ldots, x_m)$.



Rys. 15.14

Ogólny schemat układu kombinacyjnego UK przedstawia rysunek 15.14. Projektowanie układu kombinacyjnego składa się z następujących etapów:

1) na podstawie słownego sformułowania zadań wykonywanych przez układ określa się liczbę jego wejść i wyjść,

2) zestawia się tablicę zależności stanów wyjść od stanów wejść,

3) na podstawie tablicy zależności znajduje się wyrażenia strukturalne dla każdego wyjścia; ponieważ w układach przełączających sygnały mogą przyjmować wartości 0 lub 1, wyrażenia te będą funkcjami Boole'owskimi,

4) upraszcza się wyrażenia strukturalne jedną z metod minimalizacji funkcji Boole'a,

5) na podstawie uproszczonych wyrażeń strukturalnych rysuje się schemat połączeń układu,

6) sprawdza się prawidłowość działania układu z założeniami.

Sposób projektowania układu rozpatrzymy na przykładzie przedstawionym poniżej.

Przykład 15.10

Należy zaprojektować układ sterujący silnikami wentylatorów W_1 i W_2 w hali fabrycznej w zależności od temperatury pomieszczenia. Temperatura jest mierzona przez 3 termometry kontaktowe T_1, T_2, T_3 nastawione na 3 różne temperatury: $T_1 = 25^oC$, $T_2 = 30^oC$, $T_3 = 35^oC$.

Moc wentylatorów: $W_1 = 5$ kW, $W_2 = 10$ kW.

Założenia:

- gdy temperatura pomieszczenia t jest mniejsza od 25^oC wentylatory nie pracują,
- gdy $25^oC \leqslant t < 30^oC$ pracuje wentylator W_1,
- gdy $30^oC \leqslant t < 35^oC$ pracuje wentylator W_2,
- gdy $t \geqslant 35^oC$ pracują oba wentylatory,
- nieprawidłowe działanie termometrów powoduje uruchomienie alarmu A.

Rozwiązanie

1. Liczbę wejść układu - 3 (T_1, T_2, T_3) oraz liczbę wyjść układu - - 3 (W_1, W_2, A).

Przyjęto ponadto następujące oznaczenia: sygnał na wyjściu termometru, gdy temperatura osiągnie lub przekroczy wartość nastawioną przyjmuje stan 1; stan 1 oznacza również, że wentylator lub alarm pracują.

2. Tablicę zależności po przyjęciu takich oznaczeń przedstawiono na rysunku 15.15.

Ilość wierszy tablicy zależy od liczby wejść układu, gdyż musi ona przedstawiać działanie układu dla wszystkich kombinacji sygnałów wejściowych. W tym przykładzie ilość wierszy wynosi 2^3.

Stany układu przedstawione w wierszach: 1, 2, 3, 5 nie mogą wystąpić przy prawidłowym działaniu termometrów, więc dla wentylatorów są to stany obojętne; tzn. można je wykorzystywać do minimalizacji jako zera lub jako jedynki, tzn. nie narzucamy układowi żadnych wymagań dotyczących pracy wentylatorów w przypadku uszkodzenia termometrów. Zakładamy, że w tym przypadku sygnalizacja A spowoduje ingerencję człowieka (ręczne sterowanie wentylatorami).

Lp	wejścia układu			wyjścia układu			
	T_1	T_2	T_3	W_1	W_2	A	
0	0	0	0	0	0	0	← $t < 25°C$
1	0	0	1	–	–	1	
2	0	1	0	–	–	1	
3	0	1	1	–	–	1	
4	1	0	0	1	0	0	← $25°C \leq t < 30°C$
5	1	0	1	–	–	1	
6	1	1	0	0	1	0	← $30°C \leq t < 35°C$
7	1	1	1	1	1	0	← $t \geq 35°C$

Rys. 15.15

W_1:

$T_1T_2 \backslash T_3$	0	1
00	0	–
01	–	–
11	0	1
10	1	–

W_2:

$T_1T_2 \backslash T_3$	0	1
00	0	–
01	–	–
11	1	1
10	0	–

A:

$T_1T_2 \backslash T_3$	0	1
00	0	1
01	1	1
11	0	0
10	0	1

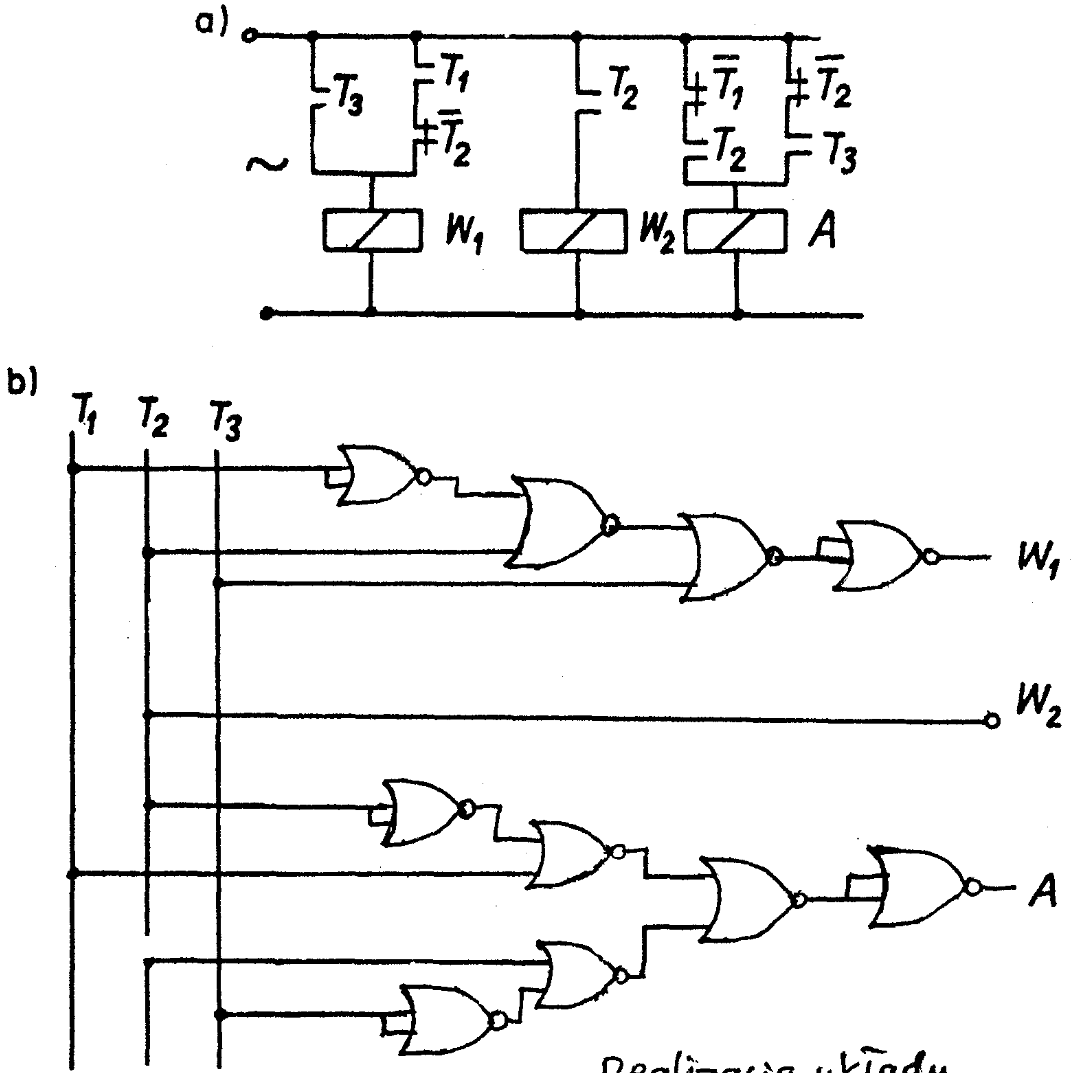
Rys. 15.16

3. Tablice Karnaugh'a dla każdego wyjścia otrzymane na podstawie tablicy zależności przedstawiono na rysunku 15.16. Po minimalizacji wyrażeń otrzymujemy następujące zależności:

$$W_1 = T_3 + T_1\overline{T_2}, \tag{15.35}$$

$$W_2 = T_2, \tag{15.36}$$

$$A = \overline{T}_1 T_2 + \overline{T}_2 T_3. \tag{15.37}$$



Rys. 15.17 – Realizacja układu z 2-wejściowych bramek NOR

4. Na rysunku 15.17a przedstawiono schemat połączeń rozpatrywanego układu zbudowanego z elementów stykowych, a na rysunku 15.17b zbudowano z elementów NOR.

5. Analiza obu schematów potwierdza spełnienie wszystkich założeń do przykładu.

Był to przykład układu kombinacyjnego, którego działanie jest funkcją wyłącznie aktualnego stanu sygnałów wejściowych.