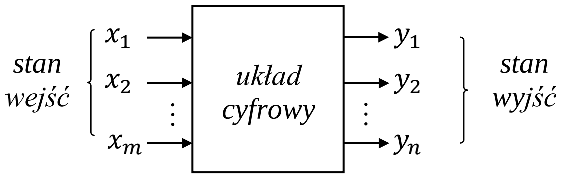
3. Układy kombinacyjne

3.1. Wprowadzenie

Układy kombinacyjne są układami cyfrowymi, w których każdej kombinacji *sygnałów wejściowych* (tzw. *stanowi wejść*) odpowiada dokładnie jedna kombinacja *sygnałów wyjściowych* (tzw. *stan wyjść*) – rys. 3.1. Układy tego typu realizują więc funkcje logiczne opisane w poprzednim rozdziale.

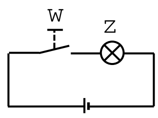


Rys.3.1. Układ cyfrowy jako „czarna skrzynka”

Podstawowymi elementami pozwalającymi na realizację układu cyfrowego są elementy przełączające. Elementy tego typu pracują w dwóch stanach:

- włączenia (zwarcia, przewodzenia),
- wyłączenia (rozwarcia, blokowania).

Przełączanie stanu elementu może odbywać się z wykorzystaniem różnych zjawisk fizycznych. Stosowane są np. elementy elektryczne, elektroniczne, pneumatyczne, hydrauliczne itp. Na rys. 3.2 został pokazany schemat prostego obwodu elektrycznego z ręcznym łącznikiem (z zestykiem zwiernym) pozwalającym na włączenie lampki sygnalizacyjnej.



Rys.3.2. Obwód elektryczny realizujący funkcję powtórzenia

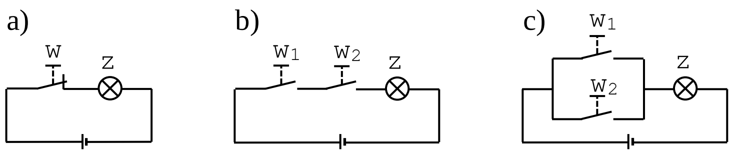
W sytuacji przedstawionej na rys. 3.2 łącznik jest w stanie rozwarcia w obwodzie nie płynie prąd, więc żarówka nie świeci, po przełączeniu łącznika w stan zwarcia w obwodzie zaczyna płynąć prąd, co powoduje, że lampka zaczyna świecić. Przyjmując, że stan łącznika i żarówki opisują zmienne boolowskie w i z zdefiniowane w taki sposób, że:

- zmienna w ma wartość 0, jeśli łącznik jest rozwarty, a wartość 1, gdy łącznik jest zwarty,
- zmienna z jest ustawiana na 0, kiedy żarówka nie świeci i ustawiana na 1, kiedy żarówka zaczyna świecić,

działanie układu można zapisać funkcją logiczną opisującą stan żarówki

$$z = w.$$

Układ realizuje funkcję powtórzenia: zmienna z otrzymuje wartość wynikającą z wartości zmiennej w. W podobny sposób można zaprojektować obwody realizujące podstawowe funkcje logiczne: negację, koniunkcję i alternatywę (zob. rys. 3.3).



Rys.3.3. Obwody elektryczne realizujące funkcje: a) negacji, b) koniunkcji, c) alternatywy

W układzie z rys. 3.3a założono, że zmienna w ma wartość 0, gdy łącznik jest zwarty, a wartość 1, gdy jest rozwarty (w obwodzie wykorzystany został łącznik ręczny z zestykiem rozwiernym). Zakładając, że wartości zmiennej z są ustawiane w taki sam sposób jak w układzie poprzednim (tzn. 0 – nie świeci, 1 – świeci), funkcję opisującą stan żarówki wyraża logiczna negacja

$$z = \overline{w}.$$

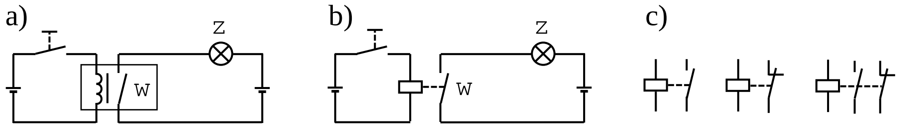
Żarówka z rys. 3.3b świeci tylko wtedy, kiedy obydwa łączniki w_1 i w_2 są zwarte, jej stan można więc opisać logiczną koniunkcją

$$z = w_1 \cdot w_2.$$

W układzie z rys. 3.3c żarówka nie świeci tylko w sytuacji, gdy obydwa łączniki w_1 i w_2 są rozwarte, co można opisać logiczną alternatywą

$$z = w_1 + w_2.$$

Zmiana stanu żarówki w obwodach z rys. 3.2 i 3.3 następuje w wyniku oddziaływania operatora modyfikującego ręcznie stan łączników. Zmianę stanu łączników można zrealizować zdalnie, zastępując łączniki przekaźnikami elektromagnetycznymi. Rysunek 3.4 pokazuje obwód z rys. 3.2 po wprowadzeniu przekaźnika.



Rys.3.4. Obwód elektryczny realizujący funkcję powtórzenia

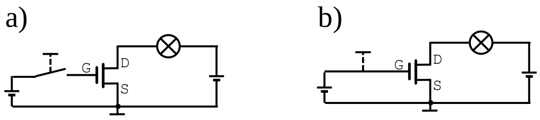
Przekaźnik elektromagnetyczny zbudowany jest z cewki, w której po zamknięciu obwodu sterującego (obwodu, w którym znajduje się cewka) zaczyna płynąć prąd. Przepływający przez cewkę prąd wytwarza pole magnetyczne, które oddziałuje na styk roboczy umieszczony w obwodzie głównym (obwód, w którym znajduje się żarówka). W wyniku tego oddziaływania obwód główny zostaje zamknięty i lampka sygnalizacyjna zaczyna świecić. Otwarcie obwodu sterującego powoduje zanik pola i w efekcie rozwarcie obwodu głównego, co przerywa świecenie lampki. Rysunek 3.4a pokazuje szczegóły budowy przekaźnika (cewkę w obwodzie sterującym i styk roboczy w obwodzie głównym), na rys. 3.4b wykorzystany został symbol przekaźnika, rys. 3.4c przedstawia symbole przekaźników z różnymi stykami roboczymi (z zestykiem zwiernym, rozwiernym, z parą zestyków).

Przekaźniki elektromechaniczne zajmują stosunkowo dużo miejsca, mają stosunkowo małe prędkości przełączania i ograniczony cykl życia styków. Wady te powodują, że nie są wykorzystane do budowy złożonych układów cyfrowych. Układy te budowane są najczęściej z wykorzystaniem elementów scalonych wytwarzanych w technologiach TTL i CMOS. W obydwu przypadkach funkcje logiczne budowane są z wykorzystaniem tranzystorów: w technologii TTL wykorzystywane są tranzystory bipolarne, a w CMOS komplementarne tranzystory unipolarne z kanałami typu N (NMOS) i typu P (PMOS).

Unipolarne tranzystory NMOS i PMOS mają po trzy elektrody: bramkę (ang. gate), źródło (ang. source) oraz dren (ang. drain). Symbole tranzystorów, w jednym ze stosowanych standardów, zostały pokazane na rys. 3.5. Litery G, S i D oznaczają odpowiednio bramkę, źródło i dren tranzystora.

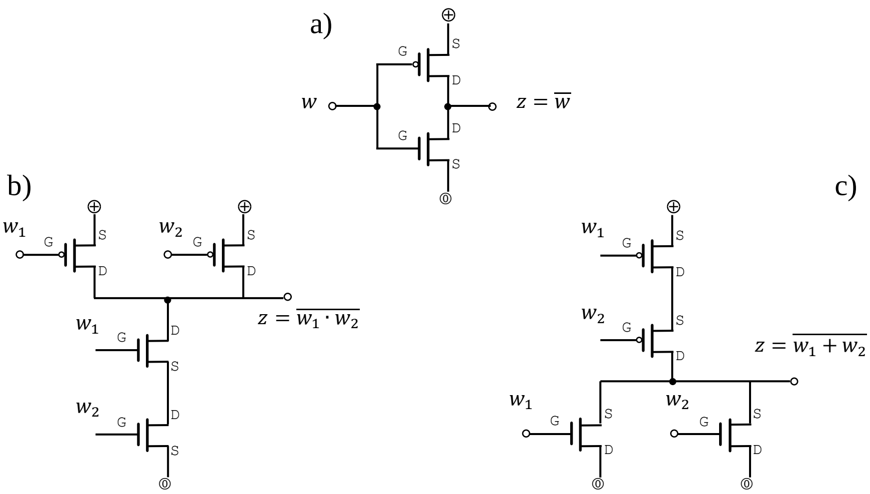
Rys.3.5. Symbole tranzystorów unipolarnych: a) NMOS, b) PMOS

Tranzystory CMOS mogą pełnić funkcje przełączające – ich stan zależy od wielkości napięcia bramka-źródło V_{GS}. Tranzystor NMOS jest w stanie wyłączenia, jeśli napięcie V_{GS} jest mniejsze od napięcia progowego V_{TH}, jeśli potencjał bramki jest odpowiednio większy od potencjału źródła, tzn. napięcie V_{GS} jest znacznie większe od napięcia progowego V_{TH}, to tranzystor jest w stanie włączenia, a pomiędzy źródłem i drenem tworzy się kanał, który przewodzi prąd. Tranzystor na rys. 3.6a jest w stanie wyłączenia ($V_{GS} = 0$), co powoduje, że w obwodzie roboczym nie płynie prąd. Po zamknięciu obwodu sterującego, tak jak pokazano to na rys. 3.6b, różnica potencjałów pomiędzy bramką a źródłem jest dodatnia ($V_{GS} \gg 0$), tranzystor jest włączany i w obwodzie roboczym zaczyna płynąć prąd.



Rys.3.6. Obwód z tranzystorem NMOS w stanach a) wyłączenia, b) włączenia

Tranzystor NMOS dla niskiego napięcia bramka-źródło jest w stanie wyłączenia, wysoki stan napięcia powoduje włączenie tranzystora. Tranzystor PMOS działa przeciwnie, jest w stanie włączenia dla niskiego napięcia bramka-źródło, a w stanie wyłączenia dla wysokiego napięcia bramka-źródło. Obwody z tranzystorami NMOS i PMOS mogą być wykorzystane do realizacji funkcji logicznych [9] (zob. rys. 3.7).



Rys.3.7. Obwody realizujące funkcje: a) negacji, b) zanegowanej koniunkcji, c) zanegowanej alternatywy (⊚, ⊕ stany: niski i wysoki)

3.2. Układy z elementów NAND lub NOR

Operacje *koniunkcji, alternatywy* i *negacji* pozwalają na zapisanie **każdej** funkcji boolowskiej (np. w kanonicznej postaci dysjunkcyjnej lub koniunkcyjnej). System składający się z tych trzech operacji nazywany jest *podstawowym systemem funkcjonalnie pełnym*. Korzystając z praw de Morgana, można wykazać, że *podstawowy system funkcjonalnie pełny* **nie jest minimalny** (np. alternatywę można zastąpić koniunkcją i negacją).
Minimalnymi systemami funkcjonalnie pełnymi są np.:

- system wykorzystujący wyłącznie *negację koniunkcji* (NAND),
- system wykorzystujący wyłącznie *negację alternatywy* (NOR).

Zaletą wykorzystania systemu jednorodnego (opartego wyłącznie na jednej operacji logicznej) jest to, że cały układ logiczny może być zbudowany z wykorzystaniem elementów tylko jednego typu (wyłącznie z elementów NAND czy NOR). Dodatkowo tranzystorowe realizacje funkcji NAND i NOR są prostsze i w konsekwencji szybciej działają od swoich niezanegowanych odpowiedników konstruowanych z obwodów NAND czy NOR uzupełnionych obwodem realizującym negację na wyjściu.

W przypadku zamiany wyrażenia zapisanego przy użyciu operacji podstawowego systemu funkcjonalnie pełnego na opis zawierający wyłącznie negację koniunkcji czy negację alternatywy niezbędne jest wykonanie odpowiednich przekształceń. Do wykonania tych przekształceń wystarcza znajomość następujących praw algebry Boole'a (zob. s. 8):

prawo podwójnej negacji $a = \bar{\bar{a}}$,

prawa de Morgana $\overline{a + b} = \bar{a} \cdot \bar{b}$,

$\overline{a \cdot b} = \bar{a} + \bar{b}$.

Przykład 1

Realizacja funkcji $z = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$ z elementów NAND.

Z funkcji, za pomocą *prawa de Morgana*, przede wszystkim należy wyeliminować alternatywy. Zastosowanie tego podejścia wymaga jednak wprowadzenia do wyrażenia negacji. Korzystając z *prawa podwójnej negacji*, funkcję z można zapisać w równoważnej postaci jako

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

po zastosowaniu *prawa de Morgana* $\overline{a + b} = \bar{a} \cdot \bar{b}$ otrzymuje się

$$z = \overline{\overline{a\bar{b}\bar{c}} \cdot \overline{a\bar{b}c} \cdot \overline{ab\bar{c}}}\,.$$

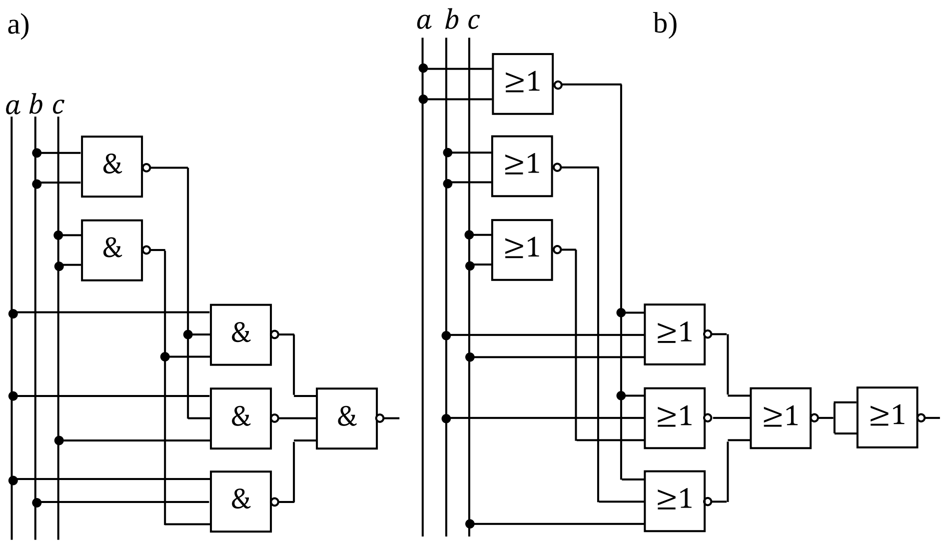
Otrzymane w ten sposób wyrażenie jest już negacją koniunkcji, której argumentami są wyrażenia:

$$\overline{\bar{a}\bar{b}\bar{c}}, \quad \overline{a\bar{b}c} \quad \text{i} \quad \overline{ab\bar{c}}.$$

Wyrażenia te są również zanegowanymi koniunkcjami, więc funkcję

$$z = \overline{\overline{a\bar{b}\bar{c}} \cdot \overline{a\bar{b}c} \cdot \overline{ab\bar{c}}}$$

można zrealizować z wykorzystaniem elementów NAND biorąc pod uwagę, że koniunkcja jest operacją *idempotentną*, co pozwala zapisać $\bar{b}$ i $\bar{c}$ odpowiednio jako $\overline{b \cdot b}$ i $\overline{c \cdot c}$ (zob. rys. 3.8a).



Rys.3.8. Funkcja $z = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$ *zrealizowana z wykorzystaniem elementów a) NAND, b) NOR*

Przykład 2

Realizacja funkcji $z = a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$ z elementów NOR.

W tym przypadku, za pomocą *prawa de Morgana*, z funkcji przede wszystkim należy wyeliminować koniunkcje. Korzystając z *prawa podwójnej negacji* funkcję można zapisać w postaci

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

po zastosowaniu *prawa de Morgana* $\overline{a \cdot b} = \bar{a} + \bar{b}$ otrzymuje się wyrażenie będące alternatywą trzech składników

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

Wprowadzenie negacji (funkcja powinna być zbudowana wyłącznie z elementów realizujących negację alternatywy), umożliwia *prawo podwójnej negacji*, po jego zastosowaniu funkcję można zapisać w postaci

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

Negacje argumentów a, b i c otrzymuje się, wykorzystując fakt, że alternatywa, tak jak koniunkcja, jest operacją *idempotentną*. Realizacja funkcji z wykorzystaniem elementów NOR została przedstawiona na rys. 3.8b.

Przykład 3

Realizacja funkcji $z = (a + b) \cdot (a + \bar{b}) \cdot (\bar{b} + \bar{c})$ z elementów NAND.

Wprowadzenie podwójnych negacji

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

umożliwia skorzystanie z *prawa de Morgana*

$$z = \overline{\bar{a}\,\bar{b}} \cdot \overline{\bar{a}\,b} \cdot \overline{b\,c}$$

i po ponownym skorzystaniu z *prawa podwójnej negacji*

$$z = \overline{\overline{\overline{\bar{a}\,\bar{b}} \cdot \overline{\bar{a}\,b} \cdot \overline{b\,c}}}$$

umożliwia realizację układu z elementów NAND (zob. rys. 3.9a).

Przykład 4

Realizacja funkcji $z = (a + b) \cdot (a + \bar{b}) \cdot (\bar{b} + \bar{c})$ z elementów NOR.

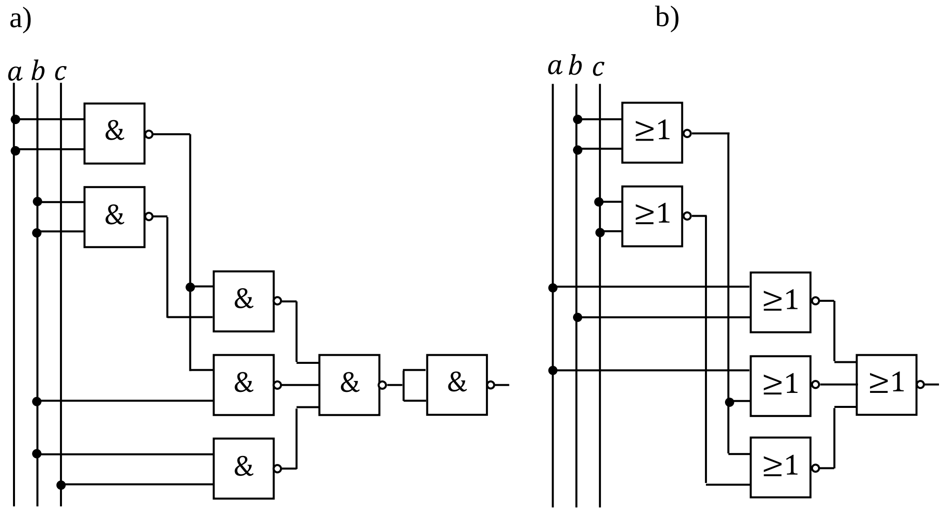
Po podwójnym zaprzeczeniu

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

i skorzystaniu z *prawa de Morgana*

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

otrzymuje się wyrażenie, które pozwala na realizację układu z wykorzystaniem elementów NOR (zob. rys. 3.9b).



Rys.3.9. Funkcja $z = (a + b) \cdot (a + \bar{b}) \cdot (\bar{b} + \bar{c})$ zrealizowana z elementów a) NAND, b) NOR

3.3. Układy wielowyjściowe

W praktyce rzadko układy cyfrowe są wykorzystywane do generowania tylko jednego sygnału wyjściowego. Na ogół, tak jak pokazano na rys. 3.1, układ generuje kilka sygnałów wyjściowych, które zależą od tego samego zbioru sygnałów wejściowych. W takim przypadku pojedyncza funkcja logiczna jest niewystarczająca do opisu działania układu i konieczne jest wykorzystanie układu funkcji logicznych. Projektując układy tego typu, funkcje opisujące poszczególne sygnały wyjściowe można zapisywać niezależnie. Otrzymany w ten sposób opis nie pozwala na uwzględnienie wspólnych członów, które mogą się pojawić w opisach wyjść układu. Powtarzające się elementy opisu można wykryć, analizując tablice Karnaugha lub stosując zmodyfikowaną metodę QMC.

Przykład 5

Układ umożliwia włączanie i wyłączanie dwóch urządzeń sygnałami z_1 i z_2 w zależności od wartości sygnałów z dwóch czujników x_1 i x_2 monitorujących stan pewnego parametru systemu oraz stanu łącznika x_3 pozwalającego na przełączanie trybu pracy układu. W trybie normalnym ($x_3 = 0$) wysoki stan sygnału z czujnika ($x_1 = 1$ lub $x_2 = 1$) powoduje włączenie odpowiedniego urządzenia ($z_1 = 1$ lub $z_2 = 1$). W trybie oszczędzania energii ($x_3 = 1$) urządzenia są włączane tylko w przypadku, gdy stan obydwu czujników jest wysoki.

Tabela prawdy i tablice Karnaugha dla wyjść z_1 i z_2 sporządzone na podstawie opisu układu zostały pokazane na rys. 3.10.

a)

x_1	x_2	x_3	z_1	z_2
0	0	0	0	0
0	0	**1**	0	0
0	**1**	0	0	**1**
0	**1**	**1**	0	0
1	0	0	**1**	0
1	0	**1**	0	0
1	**1**	0	**1**	**1**
1	**1**	**1**	**1**	**1**

b)

x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	0	1	1
1	0	0	1	0

z_1

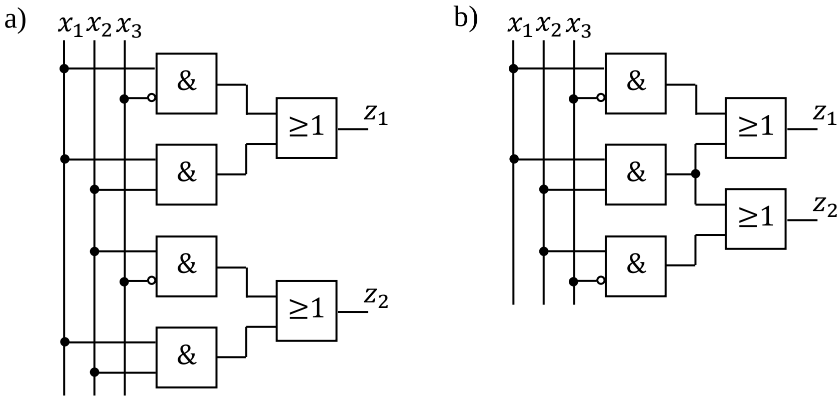
x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	1	1	0
1	0	0	1	0

z_2

Rys.3.10. Synteza układu: a) tabela prawdy, b) tablice Karnaugha

Funkcje logiczne opisujące wartości sygnałów z_1 i z_2 wyznaczone na podstawie grup jedynek, zaznaczonych na rys. 3.10, można zapisać w postaci:

$$z_1 = x_1\bar{x}_3 + x_1\,x_2,$$
$$z_2 = x_2\bar{x}_3 + x_1\,x_2.$$



Rys.3.11. Wyjścia układu zaprojektowane: a) niezależnie, b) z wykorzystaniem wspólnego elementu

Z tablic Karnaugha i otrzymanych ostatecznie wyrażeń wynika, że w obydwu funkcjach z_1 i z_2 występuje ten sam składnik $x_1\,x_2$. Pozwala to na przygotowanie bardziej oszczędnego schematu układu (zob. rys. 3.11a i 3.11b). Warto zauważyć, że próba zapisu wyrażeń opartych na grupach zer nie pozwoliłaby na takie uproszczenie. W przypadku bardziej złożonych układów wyodrębnianie wspólnych elementów nie jest proste i nie zawsze jest możliwe.

Przykład 6

Podstawowa funkcja układu sprowadza się do blokowania pracy urządzenia, jeśli nie są spełnione warunki bezpieczeństwa sygnalizowane za pomocą dwóch czujników x_1 i x_2. Sygnały $x_1 = 1$, $x_2 = 1$ oznaczają naruszenie warunków bezpieczeństwa, co w normalnym trybie pracy układu oznacza, że urządzenie jest blokowane ($u = 0$). Obejście blokady umożliwia łącznik x_3, którego ustawienie ($x_3 = 1$) pozwala na uruchomienie urządzenia ($u = 1$). Układ sygnalizuje naruszenie warunków bezpieczeństwa, włączając lampki ostrzegawcze z_1 i z_2. Lampka z_1 sygnalizuje naruszenie warunków bezpieczeństwa, lampka z_2 sygnalizuje pracę układu w warunkach niebezpiecznych. Tabela prawdy i tablice Karnaugha sporządzone na podstawie opisu układu zostały pokazane na rys. 3.12.

a)

x_1	x_2	x_3	u	z_1	z_2
0	0	0	**1**	0	0
0	0	**1**	**1**	0	0
0	**1**	0	0	**1**	0
0	**1**	**1**	**1**	**1**	**1**
1	0	0	0	**1**	0
1	0	**1**	**1**	**1**	**1**
1	**1**	0	0	**1**	0
1	**1**	**1**	**1**	**1**	**1**

b)

x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	1	0	0	0
1	1	1	1	1

u

x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	1	1	1
1	0	1	1	1

z_1

x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	0	0	0
1	0	1	1	1

z_2

Rys.3.12. Synteza układu: a) tabela prawdy, b) tablice Karnaugha

Wykorzystując fakt, że funkcje z_1 i z_2 mają pierwszą kolumnę tablicy Karnaugha wypełnioną zerami, można zapisać wyrażenia:

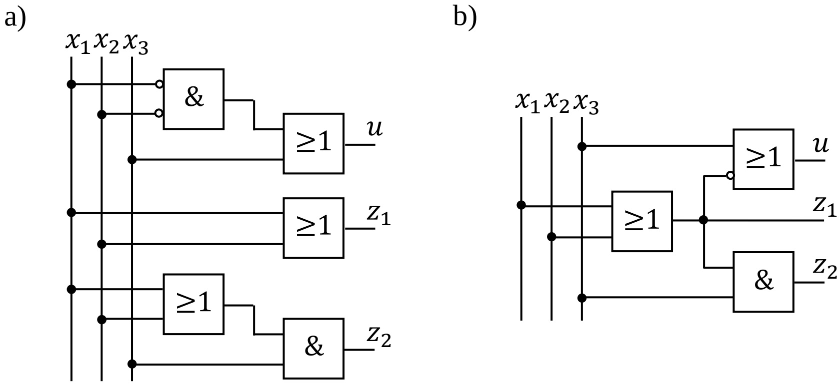
$$z_1 = x_1 + x_2,$$

$$z_2 = (x_1 + x_2)\,x_3 = z_1\,x_3.$$

Funkcję logiczną opisującą sygnał u można zapisać niezależnie, można również wykorzystać fakt, że pierwsza kolumna tablicy w tym przypadku zawiera jedynki, co pozwala na wykorzystanie do jej opisu zanegowanego wyrażenia wykorzystanego wcześniej w funkcjach z_1 i z_2

$$u = \bar{x}_1\bar{x}_2 + x_3 = \bar{z}_1 + x_3.$$

Rozważany układ można więc zrealizować, wykorzystując tylko 3 bramki (zob. rys. 3.13).



Rys.3.13. Wyjścia układu zaprojektowane: a) niezależnie, b) z wykorzystaniem wspólnego elementu

3.4. Stany nieokreślone

W praktyce stany wyjściowe układu nie zawsze są określone dla wszystkich stanów wejściowych. Układy tego typu są opisywane funkcjami logicznymi nie w pełni określonymi (niezupełnymi, zob. p. 2.2). Uwzględnienie stanów nieokreślonych w tabelach prawdy i tablicach Karnaugha wymaga użycia dodatkowego symbolu oznaczającego niezdefiniowaną wartość sygnału wyjściowego. Do oznaczenia nieokreśloności wykorzystywane są różne symbole: kreska (–), krzyżyk (x) itp. Funkcje logiczne mają tylko dwie dopuszczalne wartości: 0 i 1, wartość nieokreślona nie jest więc trzecią wartością funkcji. Minimalizując funkcje logiczne, każdą wartość nieokreśloną należy wykorzystać jako 0 lub 1, starając się uzyskać jak najprostszą postać funkcji.

Przykład 7

Układ umożliwia włączanie i wyłączanie dwóch urządzeń sygnałami z_1 i z_2 w zależności od wartości sygnałów z trzech czujników temperatury x_1, x_2 i x_3. Czujniki generują sygnał o wartości 1, jeśli mierzona temperatura przekroczy zadaną wartość progową. Temperatury progowe są dobrane w taki sposób, że przy wzroście temperatury najpierw reaguje czujnik pierwszy, później drugi, a na końcu trzeci, co pozwala uzależnić pracę urządzeń od czterech zakresów temperatury. W pierwszym zakresie ($x_1 = x_2 = x_3 = 0$) urządzenia nie powinny pracować, w drugim zakresie ($x_1 = 1,\ x_2 = x_3 = 0$) pracuje wyłącznie urządzenie pierwsze, w trzecim zakresie ($x_1 = x_2 = 1,\ x_3 = 0$) pracuje wyłącznie urządzenie drugie, w ostatnim zakresie ($x_1 = x_2 = x_3 = 1$) są włączone obydwa urządzenia. Pozostałe przypadki są fizycznie niemożliwe, więc stan urządzeń można przyjąć jako dowolny (nieokreślony).

Tabela prawdy i tablica Karnaugha sporządzone na podstawie opisu układu zostały pokazane na rys. 3.14. W tablicach zaznaczone zostały grupy pozwalające na zapis wyrażeń w postaci dysjunkcyjnej. W przypadku funkcję z_1 jedna wartość nieokreślona została wykorzystana jako 1 (tym samym pozostałe trzy wartości nieokreślone to 0), w przypadku funkcji z_2 dwie wartości nieokreślone zostały wykorzystane jako 1 (pozostałe dwie wartości to 0).

a)

x_1	x_2	x_3	z_1	z_2
0	0	0	0	0
0	0	**1**	–	–
0	**1**	0	–	–
0	**1**	**1**	–	–
1	0	0	**1**	0
1	0	**1**	–	–
1	**1**	0	0	**1**
1	**1**	**1**	**1**	**1**

b)

x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	–	0	1
1	–	–	1	–

z_1

x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	–	1	0
1	–	–	1	–

z_2

Rys.3.14. Synteza układu: a) tabela prawdy, b) tablice Karnaugha

Dla tak przyjętych wartości stanów nieokreślonych działanie układu można zapisać funkcjami w postaci:

$$z_1 = x_1\bar{x}_2 + x_3,$$

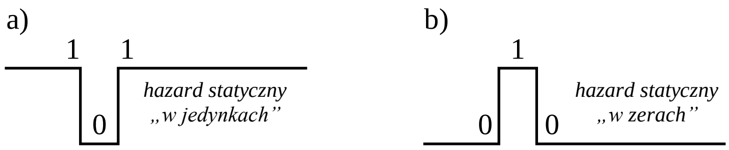
$$z_2 = x_2.$$

3.5. Przyczyny błędów

Obwody elektrycznie realizujące funkcje logiczne, niezależnie od tego czy są zbudowane z przekaźników elektromagnetycznych czy tranzystorów (zob. p. 3.1), nie są w stanie natychmiast zmienić wartości sygnału wyjściowego po zmianie sygnału wejściowego. Każdy element układu wprowadza pewne opóźnienie nazywane czasem propagacji sygnału. Opóźnienia różnych elementów układu są różne i dodatkowo opóźnienia przy przełączaniu z 0 na 1 i z 1 na 0 dla tego samego elementu mogą być różne. Efektem opóźnień mogą być krótkie zakłócenia pojawiające się na wyjściach układów kombinacyjnych.

3.5.1. Hazard statyczny

W układach zbudowanych na podstawie *postaci normalnych funkcji logicznych* (normalnych postaci dysjunkcyjnych lub normalnych postaci koniunkcyjnych) może wystąpić tzw. hazard statyczny [23]. W przypadku hazardu tego typu, zmiana pojedynczego sygnału wejściowego, która, zgodnie z założeniami, nie powinna prowadzić do zmiany wartości sygnału wyjściowego, powoduje chwilowe przełączenie sygnału. W zależności od zakładanej wartości sygnału wyjściowego rozróżniane są dwa rodzaje hazardu: hazard statyczny „w jedynkach" i hazard statyczny „w zerach". W przypadku hazardu „w jedynkach" na wyjściu układu może pojawić się impuls o wartości 0, hazard „w zerach" może prowadzić do impulsu o wartości 1 (zob. rys. 3.15).



Rys. 3.15. Przebiegi czasowe sygnału wyjściowego dla układów z hazardem statycznym: a) „w jedynkach", b) „w zerach"

W przypadku prostych układów hazard można wykryć analizując tablice Karnaugha. Układy, w których występuje hazard mają w tablicach grupy, które nie są połączone z innymi grupami. Hazard „w jedynkach" może powstawać w układach zbudowanych na podstawie postaci dysjunkcyjnej, a hazard w „zerach” w realizacjach opartych na postaci koniunkcyjnej. Projektowanie układów wolnych od hazardu polega na dołączaniu do projektu dodatkowych grup łączących stykające się ze sobą grupy – są to tzw. grupy antyhazardowe. Z jednej strony, składowe wprowadzane przez grupy antyhazardowe, z punktu widzenia zapisu funkcji logicznej w postaci minimalnej, są zbędne i niepotrzebnie rozbudowują ostateczną postać funkcji. Z drugiej strony, nie pozwalają na powstawanie impulsowych zakłóceń pokazanych na rys. 3.15.

3.5.2. Przykład 8

Na rys. 3.16a pokazana została tablica Karnaugha z zaznaczonymi linią ciągłą dwiema grupami jedynek, które pozwalają zapisać funkcję opisującą działanie układu w minimalnej postaci dysjunkcyjnej (funkcja z_h podana na rys. 3.16b). Zaznaczone grupy jedynek stykają się ze sobą (jedynka w komórce $x_1 = 1,\ x_2 = 1,\ x_3 = 1$ sąsiaduje z jedynką w komórce $x_1 = 1,\ x_2 = 0,\ x_3 = 1$), co wskazuje na obecność hazardu statycznego. Hazard może zostać wyeliminowany po uzupełnieniu funkcji o grupę antyhazardową łączącą sąsiadujące i niepołączone jedynki (grupa antyhazardowa została zaznaczona w tablicy linią przerywaną, uzupełniona o dodatkowe wyrażenie funkcja logiczna została zapisana jako z_{bh} i pokazana na rys. 3.16b). Schematy układów realizujących funkcje z_h i z_{bh} zostały przedstawione na rys. 3.16c i 3.16d. Liniami przerywanymi zaznaczone zostały elementy układów wprowadzone w celu wyeliminowania hazardu.

a)

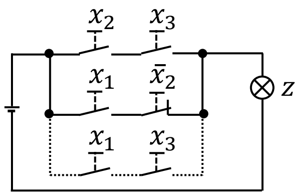
x_3 \ x_1x_2	**00**	**01**	**11**	**10**
0	0	0	0	1
1	0	1	1	1

z

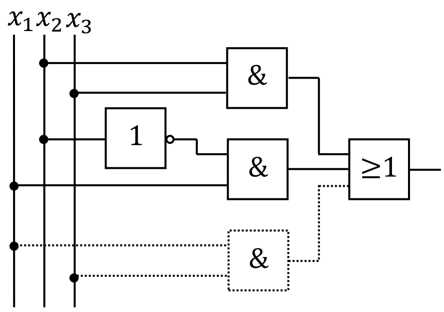
b)

$$z_h = x_2\,x_3 + x_1\,\bar{x}_2,$$
$$z_{bh} = x_2\,x_3 + x_1\,\bar{x}_2 + x_1\,x_3.$$

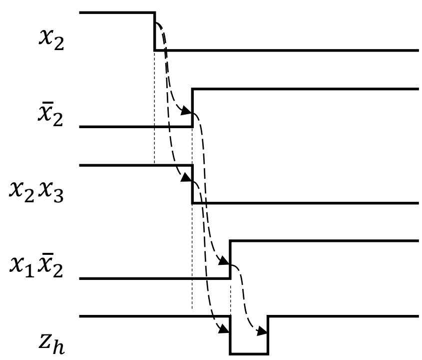
c)



d)



e)



Rys. 3.16. Przykładowy układ z hazardem w „jedynkach”: a) tablica Karnaugha, b) funkcje logiczne, c) realizacja układu z elementów stykowych, d) realizacja układu z bramek e) przebiegi czasowe ilustrujące powstawanie impulsowego zakłócenia na wyjściu

Zakłócenia wynikające z hazardu można zaobserwować na wyjściu układu, obserwując zmianę sygnału wejściowego odpowiadającą sąsiadującym niepołączonym grupom w tablicy Karnaugha (sąsiadujące komórki różnią się wartością tylko jednego sygnału wejściowego). W analizowanym przypadku zakłócenie takie powoduje zmiana sygnału x_2 (pozostałe sygnały są równe 1).

W układzie zbudowanym z elementów stykowych źródłem zakłócenia jest niejednoczesna zmiana stanu łączników x_2 i $\bar{x}_2$ (grupa antyhazardowa eliminuje ten problem). W układzie zbudowanym z bramek zakłócenia powstają ze względu na różnice w czasach propagacji sygnału. Rysunek 3.16e ilustruje mechanizm powstawania impulsu odzwierciedlającego hazard „w jedynkach” przy założeniu, że opóźnienia wprowadzane przez wszystkie bramki są jednakowe. Zmiana sygnału x_2 po opóźnieniu wynikającym z działania bramek realizujących $\bar{x}_2$ i $x_2 x_3$ wpływa na zmianę sygnałów wyjściowych z tych bramek. Po kolejnym opóźnieniu zmiana sygnału x_2 wpływa na wartość wyjścia układu (sygnał z bramki $x_2 x_3$ jest już równy 0, a sygnał z wyjścia bramki $x_1 \bar{x}_2$ jest jeszcze równy 0), zmiana ta wpływa też na wartość sygnału wyjściowego bramki $x_1 \bar{x}_2$, ale dopiero po kolejnym opóźnieniu bramka wyjściowa przetworzy tę zmianę i da na wyjściu sygnał o wartości 1 (podobnie jak w układzie zbudowanym z elementów stykowych grupa antyhazardowa eliminuje możliwość powstania impulsu). Warto zauważyć, że układ z hazardem nie generowałby zakłóceń, gdyby czasy przetwarzania sygnałów $x_2\, x_3$ i $x_1\, \bar{x}_2$ były sobie równe.

Po podstawieniu do wyrażeń opisujących funkcje logiczne z_h i z_{bh} wartości wejść, które pozostają niezmienione podczas przejścia generującego hazard (tzn. $x_1 = x_3 = 1$) i pozostawiając zmienną odpowiedzialną za powstanie zakłócenia (tzn. x_2), można zaobserwować ciekawe zależności:

$$\begin{aligned} z_h &= x_2\, x_3 + x_1\, \bar{x}_2 &&= x_2 + \bar{x}_2, \\ z_{bh} &= x_2\, x_3 + x_1\, \bar{x}_2 + x_1\, x_3 &&= x_2 + \bar{x}_2 + 1 = 1. \end{aligned}$$

Funkcja z_{bh} w postaci bezhazardowej ma wartość 1, co oznacza, że na wyjściu układu nie może pojawić się zakłócenie. Wartość funkcji w postaci z hazardem z_h zależy od wartości wyrażenia $x_2 + \bar{x}_2$. Wyrażenie to powinno zawsze dawać wartość 1, z faktu pojawiania się na wyjściu chwilowego sygnału o wartości 0 wynika, że ze względu na opóźnienia wprowadzane przez elementy układu warunek ten przez chwilę nie jest spełniony.

Powyższe spostrzeżenie można wykorzystać do skonstruowania alternatywnej metody wykrywania hazardu. W układach z hazardem zaprojektowanych na podstawie *postaci normalnych funkcji logicznych* łamane jest prawo istnienia dopełnienia. W układach z hazardem „w jedynkach" nie jest spełniony warunek

$$x + \bar{x} = 1,$$

a w układach z hazardem „w zerach" warunek

$$x \cdot \bar{x} = 0.$$

Jeżeli dla pewnego sygnału wejściowego, przy ustalonych wartościach pozostałych sygnałów, funkcję logiczną, którą realizuje układ można zapisać w postaci: $x + \bar{x}$ lub $x \cdot \bar{x}$ to w układzie występuje hazard statyczny.

Zjawisko hazardu statycznego może być niebezpieczne, jeżeli układ jest częścią większego systemu lub wchodzi w skład asynchronicznego układu sekwencyjnego (zob. p. 4). Oprócz hazardu statycznego w wielopoziomowych układach kombinacyjnych może również wystąpić hazard dynamiczny charakteryzujący się kilkukrotną zmianą wartości sygnału wyjściowego, w sytuacji gdy sygnał ten powinien zmienić się tylko raz (z 0 na 1 lub z 1 na 0). Należy też podkreślić, że omówiony sposób eliminowania zakłóceń wywołanych hazardem statycznym dotyczy tylko przypadków, w których zmianie ulega wartość tylko jednego wejścia. W rozważonym przykładzie układ zaprojektowany na podstawie funkcji z_{bh} generowałby na wyjściu chwilowy sygnał o wartości zero po zmianie stanu wejść z $x_1 = 0,\ x_2 = 1,\ x_3 = 1$ na $x_1 = 1,\ x_2 = 0,\ x_3 = 1$. Hazard będący konsekwencją jednoczesnej zmiany kilku sygnałów wejściowych jest nazywany hazardem funkcyjnym i może generować na wyjściu takie same zakłócenia jak hazard statyczny i dynamiczny. Zagadnienia związane z eliminowaniem hazardu dynamicznego i funkcyjnego zostały pominięte w tym podręczniku.

3.6. Zadania

3.1. Zapisz funkcje logiczne, używając systemu jednorodnego opartego na negacji koniunkcji (NAND), narysuj schematy logiczne przekształconych funkcji, używając symboli z normy DIN 40700 lub IEEE Std 91/91a-1991:

a) $f(a,b,c,d) = b\bar{c}d + ad + \bar{a}\bar{b}d$,
b) $g(x,y,z,q) = (\bar{a} + d)(c + \bar{b} + \bar{d})(a + d)$.

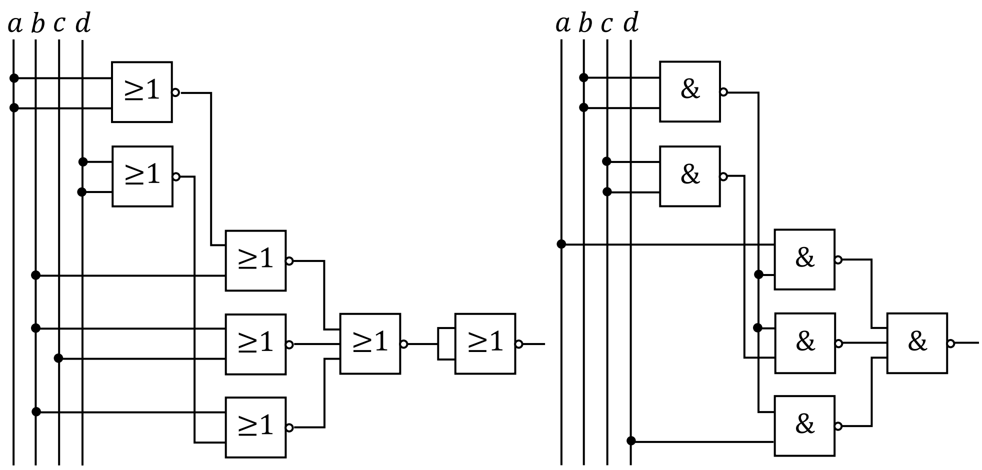
3.2. Zapisz funkcje logiczne z zadania 3.1, używając systemu jednorodnego opartego na negacji alternatywy (NOR), narysuj schematy logiczne przekształconych funkcji używając symboli z normy DIN 40700 lub IEEE Std 91/91a-1991.

3.3. Wykonaj analizę funkcji i zapisz je w minimalnej postaci bezhazardowej

a) $f(x,y,z,q) = \bar{x}\bar{z} + yz + x\bar{z}$,
b) $g(x,y,z,q) = (x + y + z)(\bar{x} + y)(\bar{y} + z + q)(x + y + \bar{z} + q)$.

Wskazówka: skorzystaj z podejścia przedstawionego w p. 2.5.5.

3.4. Przeprowadź analizę funkcji przedstawionych na schematach logicznych, zapisz wyrażenia logiczne opisujące te funkcję oraz ich tabela prawdy.



3.5. Zaprojektuj układ sterowania mechanizmem automatycznego otwierania drzwi obrabiarki CNC realizującej pewne operacje technologiczne. Obrabiarka sygnalizuje pracę sygnałem p ($p = 1$ praca, $p = 0$ zatrzymanie) oraz operację, które jest wykonywana sygnałami a_1, a_2, a_3 (1 oznacza realizację danej operacji). Należy przyjąć, że w danej chwili może być wykonywana tylko jedna operacja (co najwyżej jeden z sygnałów a_1, a_2, a_3 może mieć wartość 1), a obrabiarka może zatrzymać się po zakończeniu dowolnej z nich. Gdy obrabiarka przechodzi od wykonania jednej operacji do kolejnej, wszystkie sygnały mają wartość 0, ale urządzenie zgłasza tryb pracy ($p = 1$). Drzwi otwierane są sygnałem d ($d = 1$ otwarcie, $d = 0$ blokada) i powinny pozostać zamknięte dopóki obrabiarka zgłasza tryb pracy. Należy je otworzyć, gdy urządzenie zatrzyma się po zakończeniu realizacji danej operacji (jeden z sygnałów a_1, a_2, a_3 ma wartość 1). Jeżeli zatrzymanie nastąpi, gdy żadna z operacji nie jest zgłaszana, oznacza to błąd i drzwi powinny pozostać zablokowane.

Przygotuj tabelę prawdy opisującą działanie układu. Uwzględnij, że pewne przypadki nie mogą wystąpić i przyjmij stan sygnału wyjściowego jako nieokreślony. Na podstawie tabeli prawdy przygotuj tablicę Karnaugha i zapisz wyrażenie opisujące działanie układu w dysjunkcyjnej i koniunkcyjnej bezhazardowej postaci minimalnej.

3.6. Zaprojektuj układ sterowania ruchem prasy hydraulicznej. Prasa jest załączana dwoma przyciskami obsługiwanymi przez operatora: przycisk w_1 włącza ruch w górę, przycisk w_2 ruch w dół (stan 1 oznacza ruch). O aktualnym położeniu prasy informują dwa czujniki: p_1 sygnalizuje położenie górne, p_2 położenie dolne. Jedynka na jednym z czujników oznacza, że prasa znajduje się w danym położeniu, 0 na obydwu czujnikach oznacza, że jest w trakcie ruchu, jedynka na obydwu czujnikach nie może wystąpić. Układ sterowania, na podstawie sygnałów z przycisków w_1, w_2 oraz czujników p_1, p_2, uruchamia ruch prasy w odpowiednim kierunku. Sygnał r_1 uruchamia siłownik wywołujący ruch w górę, sygnał r_2 siłownik wywołujący ruch w dół (stan 1 oznacza ruch). Prasa powinna poruszać się dopóki naciśnięty jest przycisk w_1 lub w_2 i nie osiągnęła położenia górnego lub dolnego zgłaszanego przez czujniki p_1 lub p_2. Jeżeli prasa znajduje się w położeniu górnym i naciśnięty zostanie przycisk w_1, ruch w górę nie jest uruchamiany, analogicznie jeżeli prasa znajduje się w położeniu dolnym i naciśnięty zostanie przycisk w_2, nie jest uruchamiany ruch w dół. W przypadku naciśnięcia obydwu przycisków lub zwolnieniu przycisku przed osiągnięciem skrajnego położenia prasa powinna być zatrzymana.

Przygotuj tabelę prawdy opisującą działanie układu. Uwzględnij, że pewne przypadki nie mogą wystąpić i przyjmij stan sygnałów wyjściowych jako nieokreślony. Na podstawie tabeli prawdy przygotuj tablicę Karnaugha i zapisz wyrażenie opisujące działanie układu w dysjunkcyjnej i koniunkcyjnej bezhazardowej postaci minimalnej.

3.7. Zaprojektuj układ sterowania z zadania 3.6, przyjmując, że przyciski w_1, w_2 skonstruowane są tak, że nie jest możliwe ich jednoczesne wciśnięcie.