



Politechnika Łódzka

Instytut Fizyki

Laboratorium elektroniki

Ćwiczenie E52IS

Realizacja logicznych układów kombinacyjnych z
bramek NOR

Spis treści:

1. Cel ćwiczenia.....	3
2. Zagrożenia	3
3. Wprowadzenie teoretyczne.....	3
3.1. Podstawowe funkcje logiczne i bramki logiczne	3
3.2. Metoda minimalizacji i syntezy układu kombinacyjnego	5
4. Dostępna aparatura	8
4.1. Moduł zadajnika stanów logicznych	8
4.2. Moduł bramek logicznych.....	8
4.3. Moduł testera stanów logicznych	10
4.4. Zasilacz.....	10
5. Przebieg doświadczenia.....	11
5.1. Sprawdzenie prawa De Morgana	11
5.2. Projektowanie i realizacja układu kombinacyjnego	11
5.3. Propozycje zadań realizowanych przez układ kombinacyjny	12
6. Wskazówki do raportu.....	13
7. Literatura	14
7.1. Literatura podstawowa	14
7.2. Literatura uzupełniająca	15

Przed zapoznaniem się z instrukcją i przystąpieniem do wykonywania ćwiczenia należy opanować następujący materiał teoretyczny:

1. Czym zajmuje się algebra Boole'a i jakie są jej podstawowe prawa i tożsamości? [1-9]
2. Minimalizacja funkcji logicznej przy użyciu tablicy Karnaugh. [1-10]
3. Symbole graficzne funktorów realizujących podstawowe funkcje logiczne AND, OR, NOT, NAND, NOR, EX-OR i EX-NOR. [1-3,5-9].
4. Realizacja pozostałych funktorów logicznych przy użyciu samych funktorów NAND albo NOR. [1-4,8]
5. Podział układów przełączających na układy kombinacyjne i sekwencyjne. [1,3,5-8]

1. Cel ćwiczenia

Celem ćwiczenia jest:

1. Zapoznanie się z funktorami realizującymi podstawowe funkcje logiczne.
2. Zaprojektowanie, wykonanie i przetestowanie kombinacyjnego układu logicznego realizującego postawione zadanie w możliwie najprostszy sposób.

2. Zagrożenia

Rodzaj	Brak	Małe	Średnie	Duże
zagrożenie elektryczne		+		
zagrożenie optyczne	+			
zagrożenie mechaniczne (w tym akustyczne, hałas)	+			
zagrożenie polem elektro-magnetycznym (poza widmem optycznym)	+			
zagrożenie biologiczne	+			
zagrożenie radioaktywne (jonizujące)	+			
zagrożenie chemiczne	+			
zagrożenie termiczne (w tym wybuch i pożar)	+			

Przewody z wtykami bananowymi są przeznaczone wyłącznie do użytku w obwodach niskiego napięcia – nie wolno podłączać ich do gniazda sieci zasilającej 230 V.

3. Wprowadzenie teoretyczne

3.1. Podstawowe funkcje logiczne i bramki logiczne

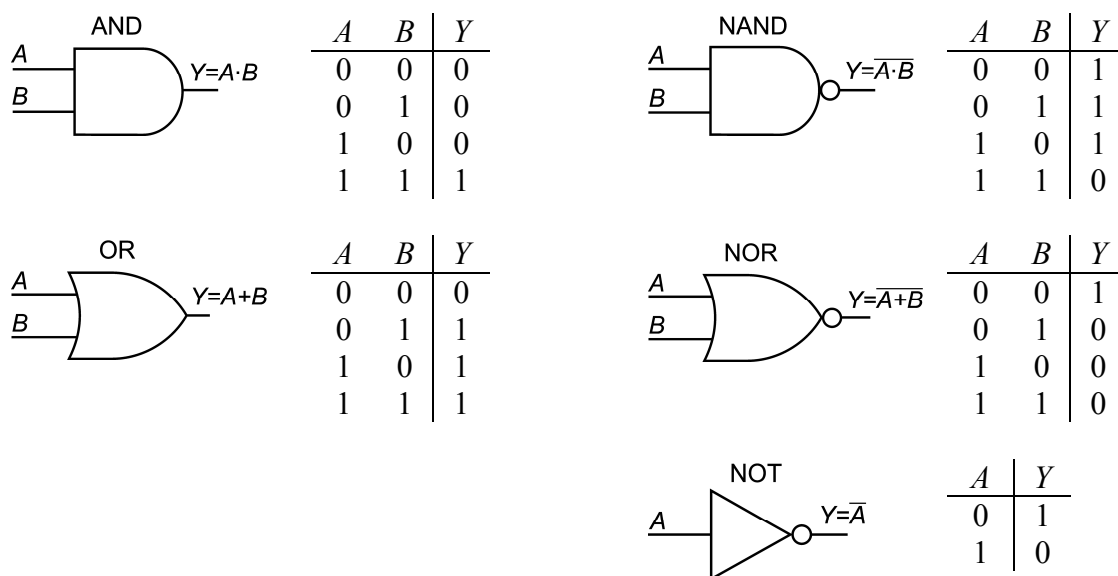
Technika cyfrowa posługuje się głównie algebrą Boole'a określoną dla zmiennych przyjmujących wartości ze zbioru dwóch elementów $\{0; 1\}$. Spotyka się również układy dopuszczające trzy stany, jednakże nie będą one przedmiotem tego ćwiczenia. Na zmiennych dwustanowych definiuje się następujące działania:

- iloczyn logiczny nazywany funkcją AND: $Y = A \cdot B$,
- sumę logiczną OR: $Y = A + B$,
- negację argumentu NOT: $Y = \overline{A}$.

W praktyce wygodnie jest wprowadzić również funkcje realizujące złożenie funkcji AND z NOT oraz OR z NOT

- negacja iloczynu NAND: $Y = \overline{A \cdot B}$,
- negacja sumy NOR: $Y = \overline{A + B}$.

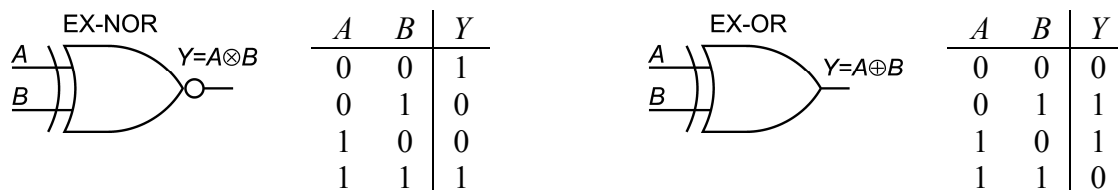
Symbole układów elektrycznych (zwanymi dalej bramkami) realizujących wymienione powyżej funkcje zestawiono na Rys. 1. Definicje funkcji AND, OR, NAND i NOR można łatwo rozszerzyć na dowolną liczbę argumentów przez złożenie ich z funkcji dwuargumentowych, np. $A \cdot B \cdot C = (A \cdot B) \cdot C$.



Rys. 1. Symbole podstawowych bramek logicznych i ich tablice stanów.

Ze względu na duże znaczenie praktyczne definiuje się również funkcję nierównoważności EX-OR (skrót od EXCLUSIVE-OR) odpowiadającą polskiemu operatorowi ALBO, czyli WYŁĄCZNE LUB, a także funkcję równoważności EX-NOR (pol. ALBO-NIE) określoną jako złożenie EX-OR i NOT

- nierównoważność EX-OR: $Y = A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$,
- równoważność EX-NOR: $Y = A \otimes B = A \cdot B + \overline{A} \cdot \overline{B}$.



Rys. 2. Symbole dodatkowych bramek logicznych i ich tablice stanów.

Podczas projektowania układów logicznych największe znaczenie mają następujące prawa algebry Boole'a: przemienności, łączności, rozdzielności i De Morgana. Zauważmy, że prawo rozdzielności sumy względem iloczynu oraz obydwa prawa De Morgana nie mają swoich odpowiedników w algebrze liczb rzeczywistych.

Nazwa przekształcenia	Dla iloczynu logicznego	Dla sumy logicznej
Prawa przemienności	$A \cdot B = B \cdot A$	$A + B = B + A$
Prawa łączności	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
Prawa rozdzielności	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + B \cdot C = (A + B) \cdot (A + C)$
Prawa De Morgana	$\overline{A \cdot B \cdot \dots} = \overline{A} + \overline{B} + \dots$	$\overline{A + B + \dots} = \overline{A} \cdot \overline{B} \cdot \dots$
Tożsamości podstawowe	$A \cdot 0 = 0$ $A \cdot 1 = A$ $A \cdot A = A$ $A \cdot \overline{A} = 0$	$A + 1 = 1$ $A + 0 = A$ $A + A = A$ $A + \overline{A} = 1$
Tożsamości dodatkowe	$A \cdot (A + B) = A$ $A + \overline{A} \cdot B = A + B$ $(A + B) \cdot (\overline{A} + \overline{B}) = B$	$A + A \cdot B = A$ $A \cdot (\overline{A} + B) = A \cdot B$ $A \cdot B + \overline{A} \cdot B = B$

Tabela 1. Podstawowe prawa i tożsamości algebry Boole'a.

Posługując się prawami De Morgana można stwierdzić, że spośród funktorów realizujących podstawowe działania logiczne jedynie funktory NAND i NOR są uniwersalne, tzn. łącząc funktory tylko jednego z tych dwóch typów można zrealizować dowolną funkcję logiczną, w tym także pozostałe funkcje elementarne AND, OR, NOT, NAND, NOR, EX-OR i EX-NOR. Ograniczenie zbioru funktorów stosowanych do realizacji dowolnej funkcji logicznej ma wiele zalet i jest często stosowane w praktyce.

3.2. Metoda minimalizacji i syntezy układu kombinacyjnego

Załóżmy, że w wyniku analizy treści zadania ustalono następującą tablicę stanów układu o czterech wejściach A, B, C, D i jednym wyjściu W .

A	B	C	D	W
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	–

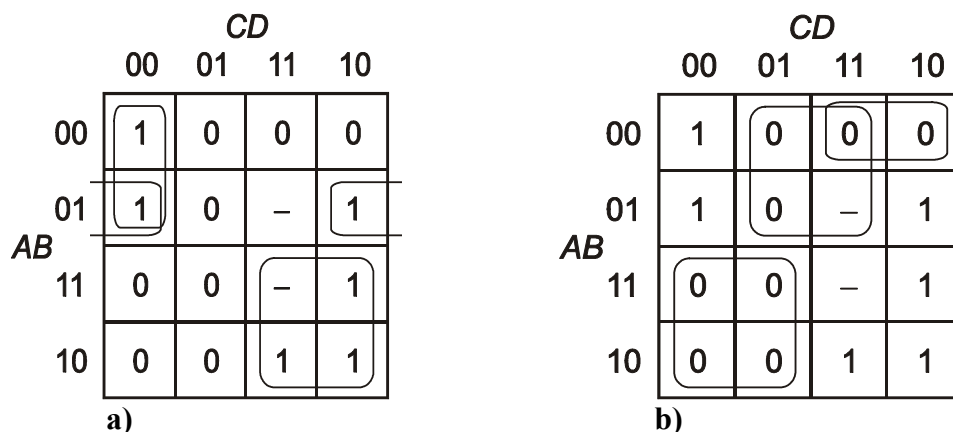
A	B	C	D	W
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	–

Tabela 2. Zapis funkcji logicznej w postaci tablicy prawdy. Kreskami (–) oznaczono przypadki, w których stan logiczny wejścia nie ma znaczenia.

Bezpośrednio na podstawie tablicy stanów funkcję można przedstawić np. w postaci kanonicznej sumy, czyli sumy składników, z których każdy stanowi iloczyn pełny wszystkich zmiennych wejściowych lub ich negacji i odpowiada innej linii w tabeli prawdy, dla której $W = 1$

$$W = \overline{A} \overline{B} \overline{C} \overline{D} + \overline{A} B \overline{C} \overline{D} + \overline{A} B C \overline{D} + A \overline{B} C \overline{D} + A \overline{B} C D + A B C \overline{D}. \quad (1)$$

Uproszczenie powyższej funkcji $W = f(A, B, C, D)$ przy wykorzystaniu zasad algebry Boole'a jest zadaniem żmudnym a prostota końcowej postaci zależy od intuicji i szczęścia projektanta. Znacznie efektywniejsza metoda wykorzystuje zdolność ludzi do rozpoznawania geometrycznych wzorów w tablicy Karnaugh'a (Rys. 3), która jest dwuwymiarową specyficznym ułożoną tablicą stanów wyjściowych. Stany wejściowe podane są na zewnątrz tablicy i uporządkowane zgodnie z kodem Graya, w którym dwa kolejne słowa różnią się dokładnie jednym bitem. W przypadku funkcji trzech zmiennych buduje się tablicę o wymiarach 2×4 , zaś dla dwóch zmiennych tablicę 2×2 . Minimalizacja funkcji sprowadza się do zgrupowaniu sąsiednich jedynek w prostokątne obszary (Rys. 3.a), przy czym liczba pól w danych obszarze musi być potęgą liczby 2. Przeciwnie krawędzie tablicy traktuje się przy tym jak sklezione ze sobą. Aby otrzymać najprostsze rozwiązanie należy zakreślić wszystkie jedynek wybierając możliwie najmniejszą liczbę obszarów o maksymalnie dużych rozmiarach. Alternatywnie można łączyć same zera (Rys. 3.b). Wybrane pola zawierające stan nieokreślony (-) można łączyć zarówno z jedynekami, jak i zerami.



Rys. 3. Zapis funkcji logicznej danej Tabelą 2 w postaci tablicy Karnaugh'a.
 a) minimalizacja funkcji przez łączenie jedynek, b) minimalizacja funkcji przez łączenie zer.

Na podstawie tablicy na Rys. 3.a zminimalizowaną funkcję logiczną tworzymy jako sumę iloczynów, przy czym każdy iloczyn musi przyjmować wartość 1 dla wszystkich pól w danej grupie. Przykładowo, obszar zakreślony poprzez krawędzie tablicy odpowiada iloczynowi $\overline{A} \overline{B} \overline{D}$, gdzie pominięte C nie zachowuje jednego stanu na całym obszarze. Dla wszystkich grup otrzymujemy

$$W = \overline{A} \overline{C} \overline{D} + \overline{A} B \overline{D} + AC. \quad (2)$$

W przypadku, gdy na tablicy Karnaugh'a połączono pola o stanach 0 (Rys. 3.b), funkcję logiczną tworzymy jako iloczyn sum, z których każda przyjmuje wartość 0 w jednym zakreślonym obszarze

$$W = (\overline{A} + C)(A + \overline{D})(A + B + \overline{C}). \quad (3)$$

Załóżmy, że funkcję trzeba zrealizować przy użyciu samych funktorów NOT i NOR. Funkcję iloczynu logicznego, występującą we wzorach (2) i (3), przekształcamy zgodnie z prawami De Morgana

$$XY = \overline{\overline{X} + \overline{Y} + \dots}, \quad \overline{X} \overline{Y} \dots = \overline{X + Y + \dots} \quad (4)$$

otrzymując ze wzorów (2) i (3) odpowiednio

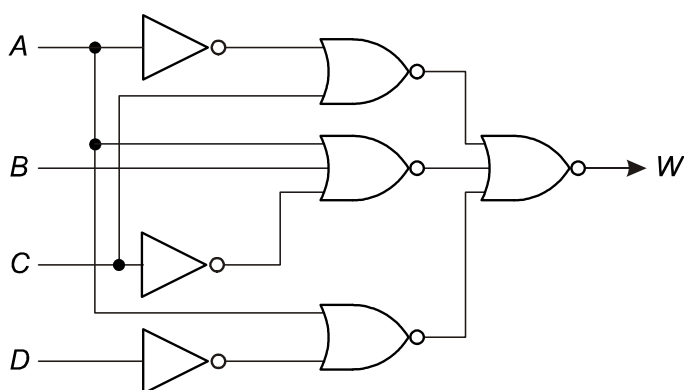
$$W = \overline{(A + C + D)} + \overline{(A + \bar{B} + D)} + \overline{(\bar{A} + \bar{C})}, \quad (5)$$

$$W = \overline{(\bar{A} + C)} + \overline{(A + \bar{D})} + \overline{(A + B + \bar{C})}. \quad (6)$$

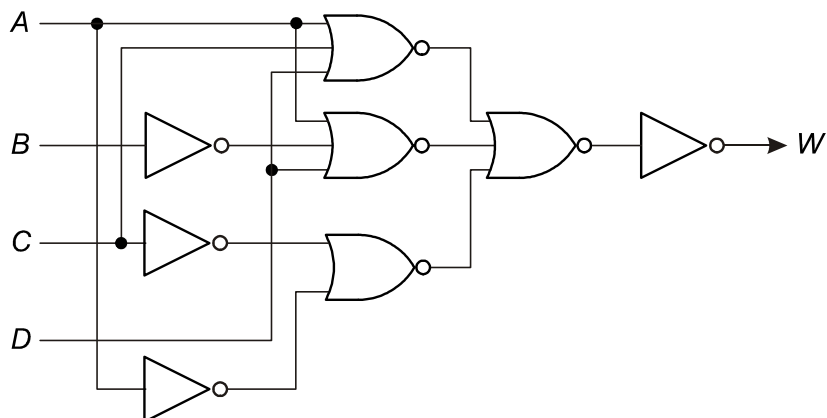
Wszystkie działania występując we wzorze (6) można zrealizować bezpośrednio przy użyciu założonych funktorów. We wzorze (5) najbardziej zewnętrzną funkcję sumy logicznej (OR) trzeba przedstawić jako złożenie negacji (NOT) i zanegowanej sumy (NOR)

$$W = \overline{\overline{(A + C + D)} + \overline{(A + \bar{B} + D)} + \overline{(\bar{A} + \bar{C})}}. \quad (7)$$

Jak widać na rysunkach 4 i 5 realizacja sprzętowa funkcji danej wzorem (7) wymaga większej liczby bramek NOT i większej liczby wejść bramek NOR niż w przypadku funkcji (6). Ponadto rozwiązanie (7) jest także mniej korzystne pod względem czasu propagacji sygnału z wejść na wyjście.



Rys. 4. Schemat układu realizującego funkcję (6).

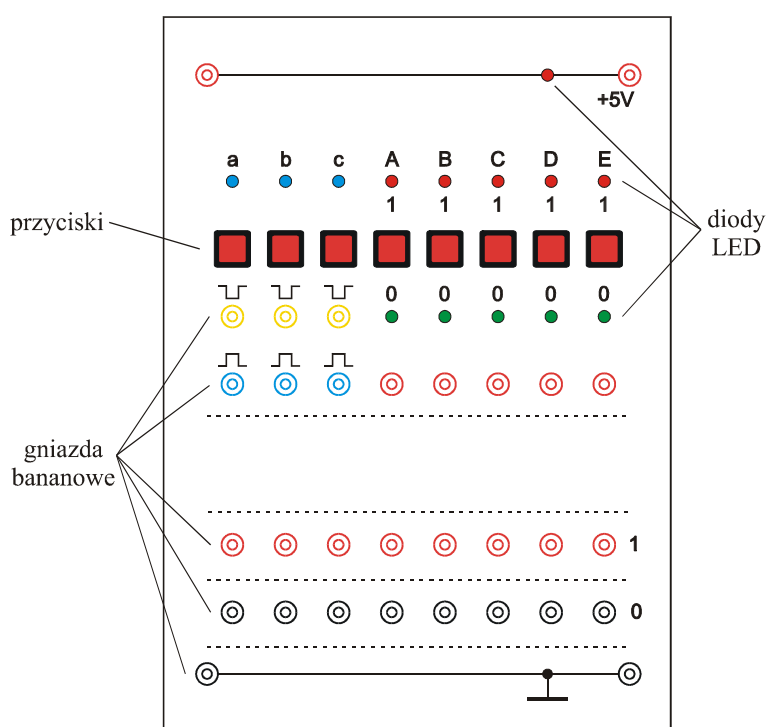


Rys. 5. Schemat układu realizującego funkcję (7).

4. Dostępna aparatura

4.1. Moduł zadajnika stanów logicznych

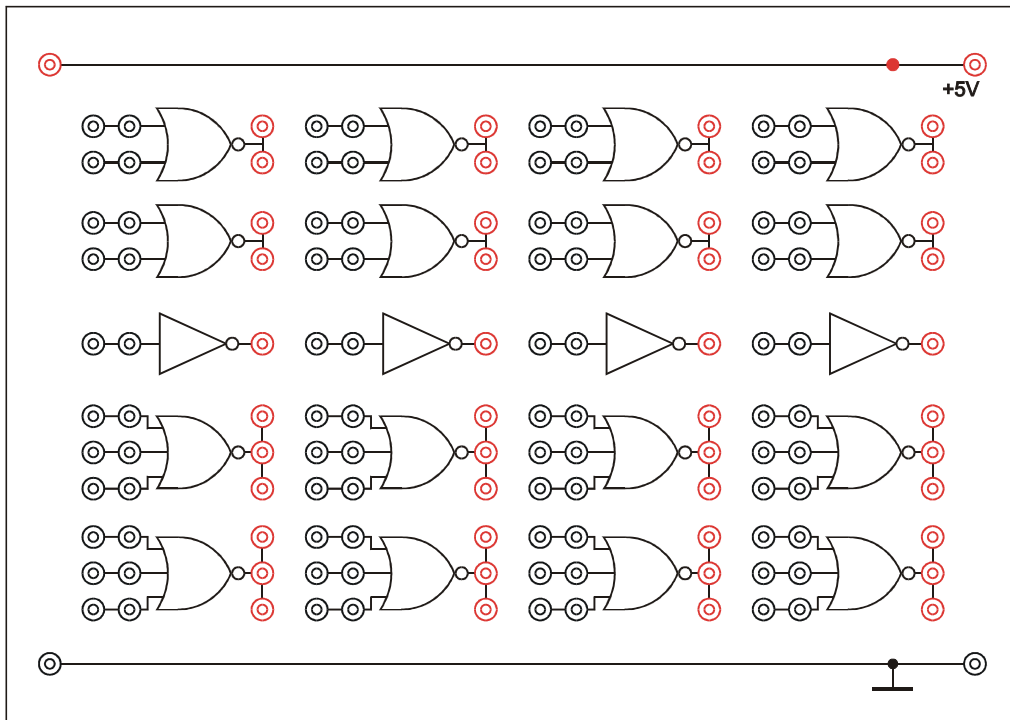
Moduł zadajnika stanów logicznych składa się z pięciu przełączników umożliwiających wybór stanu logicznego 0 albo 1 w czerwonych gniazdach umieszczonych pod przełącznikami (rys. 6). W przypadku gdy konieczne jest niezależne zadawanie stanów na więcej niż pięciu wejściach układu kombinacyjnego, najrzadziej przełączane wejścia należy podłączyć do gniazd o ustalonym stanie 0 albo 1 umieszczonych w dolnej części urządzenia. Ponadto moduł zawiera trzy generatory pojedynczego impulsu (gniazda żółte i niebieskie) wykorzystywane tylko podczas badania układów sekwencyjnych, które wykraczają poza zakres ćwiczenia przewidzianego w niniejszej instrukcji.



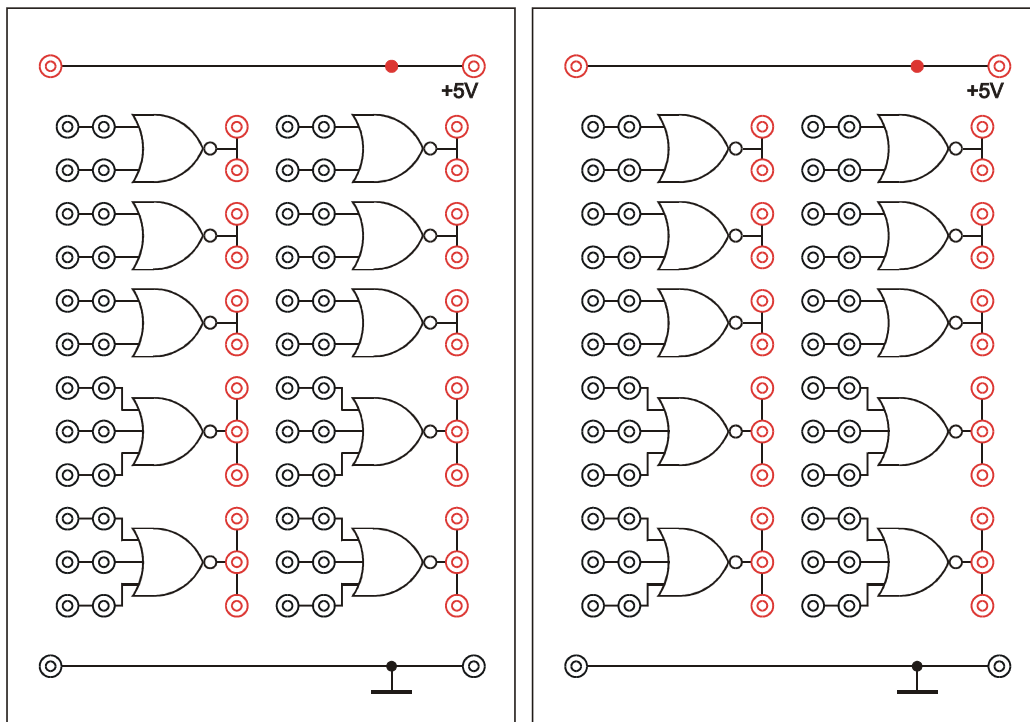
Rys. 6. Panel czołowy modułu zadajnika stanów logicznych.

4.2. Moduł bramek logicznych

Moduł bramek logicznych zawiera 4 bramki logiczne NOT, 8 dwuwejściowych bramek NOR oraz 8 trzywejściowych bramek NOR (rys. 7a). Wszystkie pozostałe funkcje logiczne należy realizować przez połączenia dostępnych bramek NOR i NOT. Alternatywnie na stanowisku może znajdować się zestaw bez bramek NOT złożony z 12 dwuwejściowych bramek NOR oraz 8 trzywejściowych bramek NOR (rys. 7b). Brakujące bramki NOT można zrealizować np. przez połączenie zworką wejść dwuwejściowych bramek NOR.



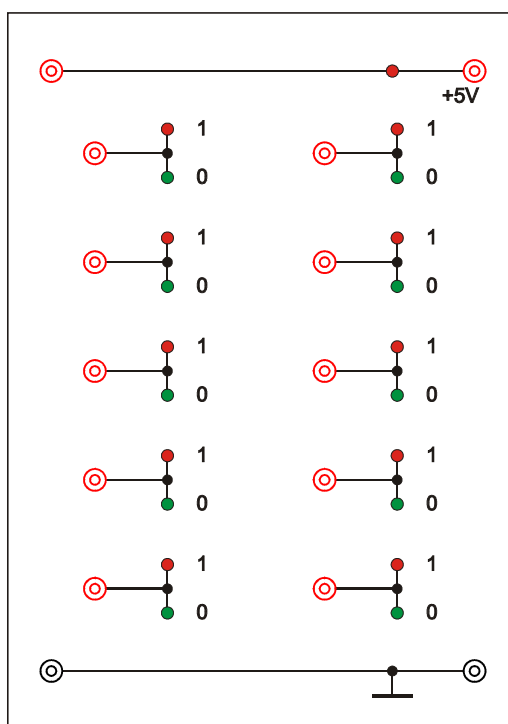
Rys. 7a. Panel czołowy modułu bramek logicznych NOR i NOT.



Rys. 7b. Panele czołowe zestawu dwóch modułów bramek logicznych NOR.

4.3. Moduł testera stanów logicznych

Moduł testera stanów logicznych zawiera 10 niezależnych testerów (rys. 8). Każdy tester wyposażony jest w jedno wejście pomiarowe oraz diody czerwoną i zieloną, których zapalenie symbolizuje stan logiczny odpowiednio 1 i 0. Jednoczesne świecenie diody czerwonej i zielonej oznacza wzbudzenie oscylacji w badanym układzie. Oscylacje nie powinny występować w poprawnie połączonym układzie kombinacyjnym, zatem ich obecność wskazuje na błąd w sieci połączeń. W przypadku gdy na wejściu pomiarowym występuje stan wysokiej rezystancji lub doprowadzone napięcie względem masy nie odpowiada żadnemu stanowi logicznemu obie diody są zgaszone. Wystąpienie takiego stanu na wejściu pomiarowym podłączonym do wyjścia bramki logicznej lub modułu zadawania stanów wskazuje na złamanie przewodu, brak kontaktu, awarię przyrządów lub brak zasilania.



Rys. 8. Panel czołowy modułu testera stanów logicznych.

4.4. Zasilacz

Zasilanie wszystkich modułów doświadczalnych opisanych powyżej zrealizowano przy użyciu jednego zasilacza dogniazdkowego +5 V z przewodem zakończonym wtyczkami bananowymi.

UWAGA: w tym ćwiczeniu nie należy zasilać modułów doświadczalnych z zasilacza laboratoryjnego SIGLENT SPD3303D, nawet z kanału o ustalonym napięciu +5 V prądu stałego. Przepięcia występujące na wyjściach tego zasilacza podczas załączania zasilania stwarzają zagrożenie dla modułów doświadczalnych.

5. Przebieg doświadczenia

Pierwszym krokiem powinno być przygotowanie własnego projektu układu kombinacyjnego. Następnie można przystąpić do połączenia zaprojektowanego układu wykorzystując dostępne bramki logiczne.

Odpowiedź na wyjściach układu bada się testerem stanów logicznych kolejno dla wszystkich kombinacji stanów zadawanych na wejściach. Wyznaczoną doświadczalnie tablicę stanów porównuje się z tablicą teoretyczną.

5.1. Sprawdzenie prawa De Morgana

1. Wykorzystując zanegowane obustronnie prawo De Morgana dla iloczynu logicznego (Tabela 1) narysować schemat elektryczny układu realizującego funkcję bramki AND przy wykorzystaniu dostępnych bramek NOR i NOT.
2. Połączyć zaprojektowany układ, do wejść układu doprowadzić sygnały z modułu zadawania stanów, zaś wyjście układu połączyć z próbnikiem stanów.
3. Połączyć z zasilaczem dogniazdkowym wszystkie używane moduły doświadczalne. W tym celu w skrajnym prawym albo lewym module podłączyć czarną wtyczkę zasilacza do linii masy (na dole modułu) oraz czerwoną wtyczkę zasilacza do linii zasilania +5 V (na górze modułu). Zasilanie pozostałych modułów realizuje się poprzez połączenie analogicznych linii w sąsiednich modułach.

UWAGA:

- a) nie używać żadnych innych dostępnych w pracowni zasilaczy niż zasilacze dogniazdkowe o ustalonym napięciu wyjściowym +5 V,
- b) nie wolno łączyć zasilania +5 V z wyjściami bramek lub wyjściami zadajnika stanów logicznych.

Nieprzestrzeżenie powyższych zaleceń grozi uszkodzeniem urządzeń.

4. Po uzyskaniu zezwolenia włączyć zasilacz i sprawdzić czy świecą się czerwone diody umieszczone na linii +5 V. W module testera stanów logicznych wszystkie diody symbolizujące stan 0 lub 1 powinny być zgaszone przy rozwartych wejściach testerów.
5. Zadając różne kombinacje sygnałów wejściowych sporządzić tablicę prawdy układu. Porównać tablicę otrzymaną eksperymentalnie z tablicą teoretyczną dla bramki AND umieszczoną na Rys. 1. W przypadku wystąpienia rozbieżności sprawdzić ponownie układ. Jeśli próby usunięcia rozbieżności nie powiodą się zawiadomić obsługę pracowni.

5.2. Projektowanie i realizacja układu kombinacyjnego

1. W porozumieniu z prowadzącym zajęcia wybrać zadania do realizacji. Przykładowe zadania zebrano w następnym rozdziale.
2. Sporządzić teoretyczną tablicę prawdy (patrz przykład w Tabeli 2). Przypadki w których stan logiczny wyjścia nie ma znaczenia oznaczyć myślnikiem.
3. Podjąć próbę minimalizacji funkcji logicznej posługując się tablicą Karnaugh'a. Rozważyć przypadek łączenia jedynek (patrz przykład na Rys. 3.a) oraz przypadek łączenia zer (Rys. 3.b).
4. Wykorzystując prawa algebry Boole'a przekształcić zminimalizowaną funkcję logiczną tak by wyrazić ją bezpośrednio przy użyciu funkcji logicznych realizowanych przez dostępną na stanowisku tablicę bramek logicznych (np. tablica zawierająca bramki NOT

oraz dwu- i trzywejściowe bramki NOR umożliwia skorzystanie wyłącznie z funkcji typu \overline{A} , $A+B$ oraz $\overline{A+B+C}$. W przypadku gdy rozwiązanie zadania wymaga zbudowania układu kombinacyjnego o kilku wyjściach porównać funkcje przełączające dla każdego z wyjść i podjąć próbę znalezienia wspólnych wyrażeń, które można wykorzystać do dalszego uproszczenia układu przez użycie tych samych bramek do realizacji więcej niż jednej funkcji.

5. Włączyć zasilacz.
6. Połączyć układ kombinacyjny zgodnie z funkcjami zbudowanymi w punkcie 4. Nie zakładać, że niepodłączone wejścia bramek posiadają pewien ustalony stan logiczny. Najprostszą metodą ustalenia stanu na nadmiarowym wejściu bramki jest połączenie go zworką z sąsiednim używanym wejściem. Alternatywnie nadmiarowe wejście można połączyć przewodem z modułem zadawania stanów logicznych. Zmiany w układzie połączeń logicznych mogą być bezpiecznie wykonywane przy włączonym zasilaniu. Układy elektroniczne są zabezpieczone przez przypadkowymi błędnymi połączeniami obejmującymi kilka wyjść bramek logicznych lub wyjść modułu zadawania stanów logicznych.
7. Sporządzić doświadczalną tablicę prawdy dla wszystkich wyjść połączonego układu.
8. Sprawdzić zgodność tablicy wykonanej w punkcie 7 z tablicą z punktu 2. W przypadku stwierdzenia rozbieżności podjąć próbę ich wyjaśnienia i usunięcia przez sprawdzenie luzów w połączeniach realizowanych przy użyciu przewodów i zwerek, przeanalizowanie przekształceń wykonanych w punktach 3 i 4 oraz sprawdzenie poprawności połączeń wykonanych w punkcie 6. Po dokonaniu poprawek sporządzić ponownie tablicę prawdy dla układu kombinacyjnego. Jeśli próby usunięcia rozbieżności nie powiodą się zawiadomić obsługę pracowni.
9. Wyłączyć zasilanie i narysować schemat układu elektrycznego.
10. Rozłączyć połączenia i uprzątnąć stanowisko.

5.3. Propozycje zadań realizowanych przez układ kombinacyjny

Liczba gwiazdek w nawiasach opisuje stopień trudności zadania. Realizacja zadań o zbyt małej łącznej liczbie gwiazdek powoduje ograniczenie oceny za kompletny i poprawny raport (zasady oceniania podano na końcu rozdziału 6).

1. (*) Zaprojektować i połączyć układ sterowania grzejnikiem zawierającym jedną grzałkę. W celu zwiększenia niezawodności sterowania zastosowano trzy jednakowe czujniki temperatury, które załączają się gdy temperatura spada poniżej jednego ustalonego poziomu. Układ sterujący powinien być odporny na awarię dowolnego jednego z trzech czujników, tzn. powinien wykrywać stan większości swoich trzech wejść.
2. (*) Zaprojektować i połączyć układ realizujący funkcję czterowejściowej bramki NAND przy użyciu bramek NOT i NOR posiadających co najwyżej trzy wejścia.
3. (*) W kodzie dwójkowo-dziesiętnym BCD każda cyfra dziesiętna jest reprezentowana za pomocą 4-bitowego słowa. Przyjmując, że do kodowania kolejnych cyfr dziesiętnych od 0 do 9 wykorzystywane są tylko kolejne wartości binarne od 0000 do 1001. Zaprojektować i połączyć układ sygnalizujący na swoim wyjściu wystąpienie niedozwolonego stanu w pojedynczym 4-bitowym słowie reprezentującym jedną cyfrę dziesiętną.
4. (**) Zaprojektować i połączyć układ inteligentnego sterowania roletą okienną. Użytkownik rolety ma do dyspozycji dwa przyciski opisane „w górę” oraz „w dół” (stan logiczny 1 po przyciśnięciu), przy czym roleta przesuwa się tylko podczas przytrzymania przycisku. Ponadto układ powinien reagować na sygnały z czujnika pozycji skrajnej górnej oraz czujnika pozycji skrajnej dolnej (stan logiczny 1 dla pozycji

- skrajnej). Układ sterujący ma dwa wyjścia załączające przesuw rolety odpowiednio do góry i w dół, przy czym stan 1 nie może pojawić się jednocześnie na obu wyjściach. Rozważyć samodzielnie reakcję układu na wystąpienie niepoprawnej kombinacji sygnałów wejściowych. Opisać przyjęte założenia oraz symbole wejść i wyjść.
5. (**) Zaprojektować i połączyć układ dokonujący konwersji trzybitowej liczby binarnej ABC na trzybitową liczbę XYZ w kodzie Graya. *Wskazówka:* każde dwie kolejne liczby zapisane w kodzie Graya różnią się dokładnie jednym bitem. Kolejne wartości binarne 000, 001, 010, 011, 100, 101, 110, 111 mają następującą reprezentację w kodzie Graya: 000, 001, 011, 010, 110, 111, 101, 100.
 6. (**) Zaprojektować i połączyć układ, który na dwubitowym wyjściu XY wybiera mniejszą z dwóch dwubitowych liczb binarnych AB oraz CD podanych na wejścia.
 7. (**) Zaprojektować i połączyć układ dokonujący konwersji sygnałów z czterech termometrów kontaktowych na trzybitową liczbę binarną XYZ z przedziału od 000 do 100. Wartość tej liczby powinna wzrastać ze wzrostem temperatury. Sygnały wejściowe z termometrów kontaktowych A, B, C, D , przy wzrastającej temperaturze zmieniają swój stan z 0 na 1 w wymienionej kolejności. Podczas optymalizacji wykorzystać założenie, że układ może zareagować w sposób dowolny na awarię termometru prowadzącą do niekolejnego przełączania stanów wejść A, B, C, D .
 8. (***) Zaprojektować i połączyć układ, który przenosi sygnały z trzybitowego wejścia ABC na trzybitowe wyjście XYZ w następujący sposób: dla wejścia sterującego w stanie $R = 0$ układ zwraca wynik $XYZ = ABC$, natomiast dla $R = 1$ dokonuje rotacji cyklicznej w lewo, tzn. zwraca wynik $XYZ = BCA$.
 9. (***) Zaprojektować i połączyć układ programowalnej bramki logicznej o dwóch wejściach dla danych A i B oraz jednym wyjściu W . Dodatkowo układ posiada wejście sterujące C do wyboru funkcji iloczynu logicznego AND dla $C = 0$ i sumy logicznej OR dla $C = 1$. Ponadto drugie wejście sterujące D utrzymywane w stanie 1 powoduje włączenie negacji wyniku, co prowadzi do realizacji funkcja NAND albo NOR. W przypadku $D = 0$ negacja nie jest wykonywana. Zwrócić uwagę na optymalizację czasu propagacji sygnałów z wejść A i B do wyjścia W przy najmniej korzystnej kombinacji stanów na wejściach C i D . Propagacja sygnału przez więcej niż pięć kolejnych bramek elementarnych świadczy o nieoptymalnym rozwiązaniu.
 10. (****) Zaprojektować i połączyć układ dokonujący inkrementacji (tzn. zwiększenia o jeden) czterobitowej liczby binarnej. Założyć, że układ nie posiada specjalnego wyjścia do sygnalizacji przeniesienia arytmetycznego dla argumentu równego 1111 i odpowiada w takiej sytuacji wartością 0000 na czterobitowym wyjściu. UWAGA: w przypadku stwierdzenia zbyt małej liczby bramek dostępnych do realizacji projektu należy ponownie przemyśleć optymalizację układu.

6. Wskazówki do raportu

Raport powinien zawierać:

1. Stronę tytułową (wg wzoru).
2. Sformułowanie celu ćwiczenia.
3. Wykaz użytej aparatury. Dla modułu/modułów bramek logicznych podać także pełną specyfikację dostępnego zestawu bramek logicznych (typy bramek, liczba wejść, liczba dostępnych bramek).
4. Opis przebiegu sprawdzenia prawa De Morgana.
5. Treść zadania, które powinien realizować zbudowany układ kombinacyjny.
6. Teoretyczne tablice prawdy dla każdego wyjścia wynikające z analizy treści zadania.

7. Minimalizację funkcji logicznej, np. metodą tablic Karnauga lub przy użyciu praw algebry Bool’a.
8. Wynik przekształcenia zminimalizowanych funkcji logicznych do postaci możliwej do bezpośredniego zrealizowania przy użyciu dostępnych bramek.
9. Schemat połączeń elektrycznych układu kombinacyjnego.
10. Tablice stanu układu zbadane doświadczalnie.
11. Dyskusję uzyskanych wyników. Przedstawić rezultat sprawdzenia zgodności zbudowanego układu kombinacyjnego z teoretyczną tablicą prawdy. W przypadku wystąpienia rozbieżności opisać środki podjęte w celu ich usunięcia, znalezione błędy i uzyskany ostatecznie rezultat. Czy metoda tablic Karnauga pomogła w znalezieniu nieoczywistych możliwości uproszczenia układu?

W raporcie ocenie podlegać będzie obecność i poprawność wszystkich wymienionych powyżej składników, czytelność prezentacji wyników w postaci tabel, wzorów i schematów wraz z opisami oraz jakość sformułowanych wniosków. Wstęp teoretyczny nie jest wymagany i w przypadku jego zamieszczenia w raporcie nie wpłynie na ocenę.

Ponadto ocena za poprawny i kompletny raport zależy od łącznej liczby gwiazdek opisujących trudność zadań, których wykonanie udokumentowano w zatwierdzonym brudnopisie. Jeżeli prowadzący zajęcia nie poda inaczej obowiązuje następująca tabela:

Suma gwiazdek za wykonane zadania	Maksymalna ocena z raportu w skali 0...5 pkt.
0	nie zaliczone
1	2 pkt.
2	3 pkt.
3	4 pkt.
≥ 4	5 pkt.

7. Literatura

7.1. Literatura podstawowa

- [1] J. Kalisz, *Podstawy elektroniki cyfrowej*, WKiŁ, Warszawa 2002.
- [2] P. Horowitz, W. Hill, *Sztuka elektroniki*, WKiŁ, Warszawa 2001,
- [3] U. Tietze, Ch. Schenk, *Układy półprzewodnikowe*, WNT, Warszawa 2009.
- [4] M. Molski, *Wstęp do techniki cyfrowej*, WKiŁ, Warszawa 1989.
- [5] R. Ćwirko, M. Rusek, W. Marciniak, *Układy scalone w pytaniach i odpowiedziach*, WNT, Warszawa, 1987.
- [6] W. Traczyk, *Układy cyfrowe. Podstawy teoretyczne i metody syntezy*, WNT, Warszawa 1986.
- [7] P. Misiurewicz, *Układy automatyki cyfrowej*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1984.
- [8] A. Rusek, *Podstawy elektroniki*, część 2, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1983.
- [9] W. Głocki, *Układy cyfrowe*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 2008.

7.2. Literatura uzupełniająca

- [10] B. Zbierzchowski, T. Łuba, K. Jasiński, M. A. Markowski, *Synteza logiczna w układach programowalnych*, Wydawnictwa Politechniki Warszawskiej, Warszawa, 1992.