LAB1: Synteza układów kombinacyjnych

I. CEL ĆWICZENIA

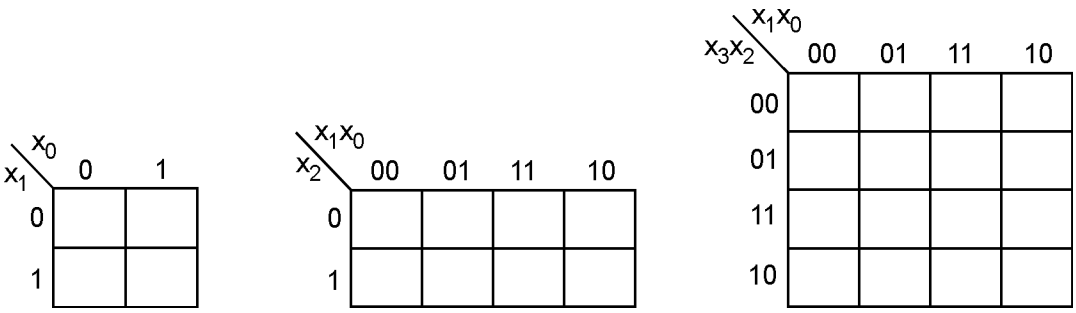
Celem ćwiczenia jest utrwalenie wiedzy dotyczącej praw algebry Boole'a, podstawowych funkcji logicznych i symboli bramek logicznych oraz procesów minimalizacji funkcji boolowskich i syntezy układów kombinacyjnych.

II. WPROWADZENIE

SIATKA KARNAUGH

Jest to graficzny zapis przedstawiający wartości funkcji logicznej dla poszczególnych kombinacji zmiennych wejściowych, wpisanych w odpowiednie pola siatki. Jeśli funkcja nie jest określona dla pewnych kombinacji tych zmiennych, to w odpowiednie pola wpisujemy kreskę (–) lub krzyżyk (×). Na rysunku 1 pokazano siatki Karnaugh dla 2, 3 i 4 zmiennych. W ogólnym przypadku siatka dla funkcji *N* zmiennych zawiera 2^N kratek.

Siatki Karnaugh tworzy się w oparciu o kod Gray'a. Jego cechą jest to, że sąsiednie słowa (kody) różnią się tylko na jednej pozycji. Przypisanie kolejnych słów kodu Gray'a kolejnym wierszom i kolumnom siatki Karnaugh powoduje, że sąsiednie kratki odpowiadają kombinacjom zmiennych wejściowych, które różnią się wartościami tylko na jednej pozycji. Pozwala to na zastosowanie prawa sklejania do uproszczenia funkcji logicznej, czyli jej minimalizacji.



Rys. 1 Siatki Karnaugh dla 2, 3 i 4 zmiennych wejściowych

MINIMALIZACJA FUNKCJI LOGICZNYCH

Zagadnienie minimalizacji funkcji logicznych powstało z konieczności zmniejszenia liczby elementów logicznych potrzebnych do realizacji układowej tych funkcji. Dla prostych funkcji, zwykle do 5 zmiennych, można stosować **metodę sklejania**, z użyciem siatek Karnaugh, w oparciu o prawo sklejania algebry Boole'a:

$$x_1\bar{x}_2 + x_1x_2 = x_1.$$

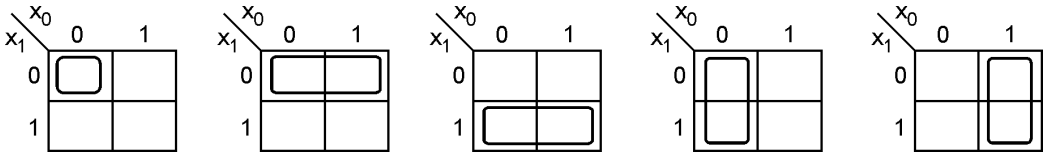
Metoda ta polega na zakreślaniu sąsiednich kratek na siatce Karnaugh o takiej samej wartości logicznej 0 lub 1. Ponadto kratki zawierające symbol wartości nieokreślonej, czyli kreskę (–) lub krzyżyk (×), można dołączyć zarówno do „grup zer", jak i do „grup jedynek".

Zatem, jeśli w dwóch sąsiednich kratkach siatki Karnaugh znajdują się jednakowe wartości logiczne, to odpowiadające tym kratkom wyrażenia można skleić, co odpowiada usunięciu tej zmiennej, która w ramach zakreślonej grupy przyjmuje obie wartości 0 i 1. Sklejenia oznaczamy poprzez zakreślenie grupy kratek (rys. 2).

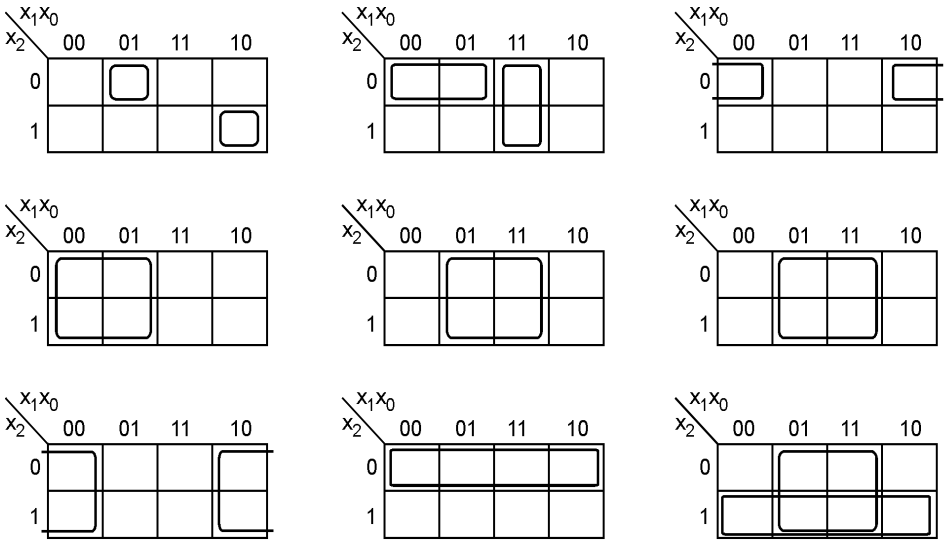
Dwie kratki tworzące parę o takich samych wartościach logicznych, można skleić z inną parą o takich samych wartościach, jeśli obie pary łącznie stanowią kwadrat lub prostokąt (rys. 3). Ta sama zasada dotyczy par czterech kratek pozwalając zakreślić grupy ośmiu kratek (rys. 4). Przeciwległe krawędzie siatki traktuje się jak zetknięte ze sobą, co pozwala na ich zakreślenie.

Zakreślane obszary powinny być jak największe a ich liczba jak najmniejsza. Ponadto należy pamiętać, że można zakreślić tylko taką liczbę kratek, która jest potęgą liczby 2, czyli 2, 4, 8, 16, 32, itd. W przypadku kratki „bez sąsiada do sklejenia", zakreślamy tylko tę kratkę.

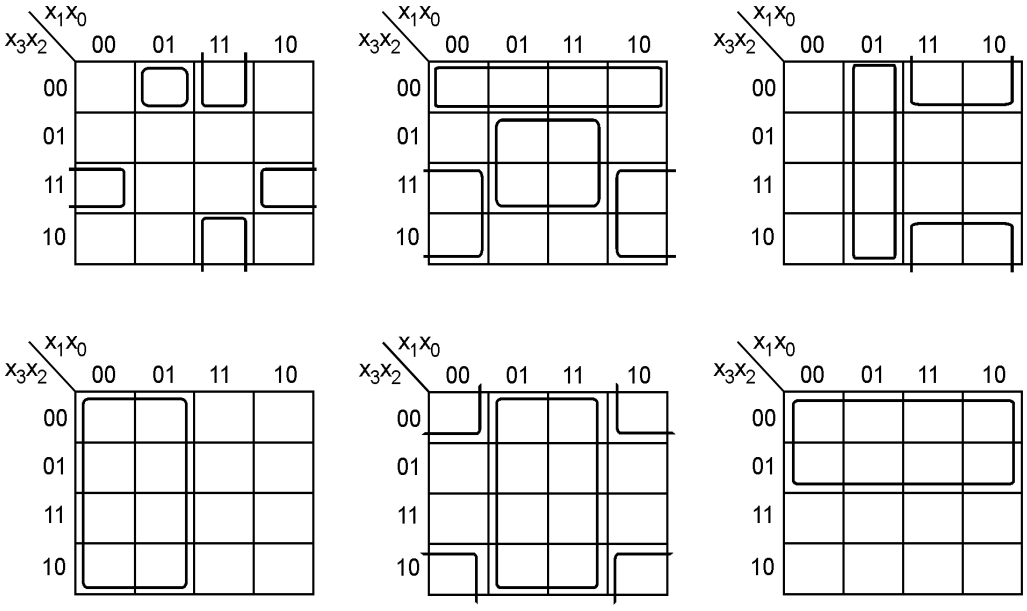
Zakreślając wartości 1 zapisujemy minimalną funkcję w **postaci sumacyjnej**, czyli sumy wyrażeń będących iloczynami logicznymi zmiennych wejściowych. Natomiast zakreślając wartości 0 uzyskujemy **postać iloczynową**, czyli iloczyn wyrażeń będących sumami logicznymi.



Rys. 2 Przykłady zakreśleń dla dwóch zmiennych x_1 x_0



Rys. 3 Przykłady zakreśleń dla trzech zmiennych x_2 x_1 x_0

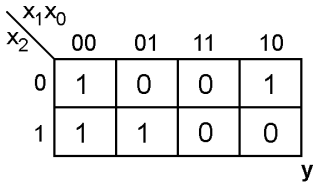


Rys. 4 Przykłady zakreśleń dla czterech zmiennych x_3 x_2 x_1 x_0

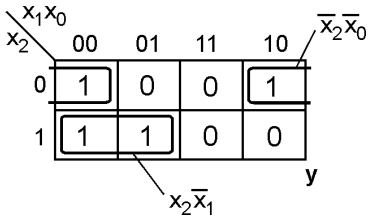
PRZYKŁAD SYNTEZY FUNKCJI LOGICZNEJ

Zminimalizować przy użyciu siatek Karnaugh funkcję logiczną określoną zbiorem T3 = {0, 2, 4, 5}. Narysować schematy logiczne powstałych układów. Ponadto zrealizować podaną funkcję logiczną z użyciem multipleksera.

Każda kratka siatki Karnaugh odpowiada dziesiętnej reprezentacji zmiennych wejściowych zapisanych po uporządkowaniu kierunku indeksowania $x_2x_1x_0$. Zatem w odpowiednie pola wpisujemy wartości funkcji:



Po analizie siatki pod względem implikantów istotnych, dokonujemy sklejeń tylko dla dwóch par o wartościach 1:



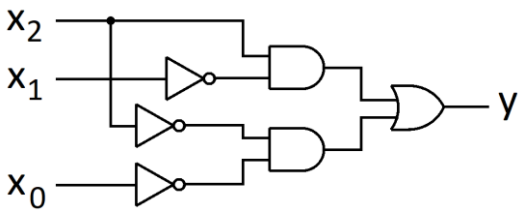
Każdą zakreśloną grupę „jedynek" zapisujemy w postaci iloczynu logicznego zmiennych wejściowych. W danym iloczynie nie występuje ta zmienna, która dla sklejonych kratek przyjmuje obie wartości 0 i 1 (odczytane z krawędzi siatki). Ponadto zmienną mającą wartość 0 zapisujemy ze znakiem negacji, a zmienną równą 1 zapisujemy bez negacji (czyli wprost). Dla powyższej siatki mamy dwie grupy, stąd odpowiednie wyrażenia iloczynowe są następujące:

$$\left.\begin{array}{l} x_2 = 1 \\ x_1 = 0 \\ x_0 = 0,1 \end{array}\right\} \rightarrow x_2\bar{x}_1, \qquad \left.\begin{array}{l} x_2 = 0 \\ x_1 = 0,1 \\ x_0 = 0 \end{array}\right\} \rightarrow \bar{x}_2\bar{x}_0.$$

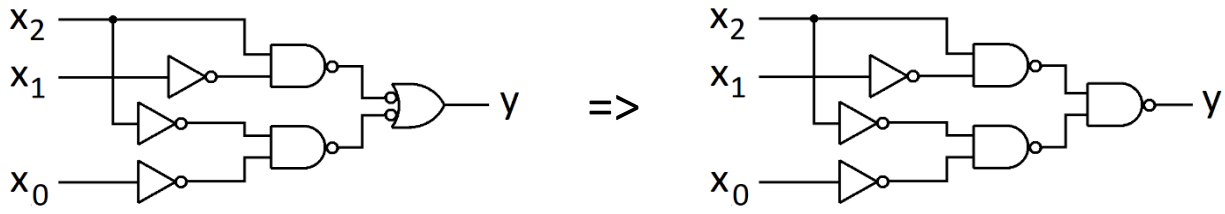
Otrzymane w ten sposób wyrażenia zapisujemy w postaci sumy logicznej:

$$y = x_2\bar{x}_1 + \bar{x}_2\bar{x}_0.$$

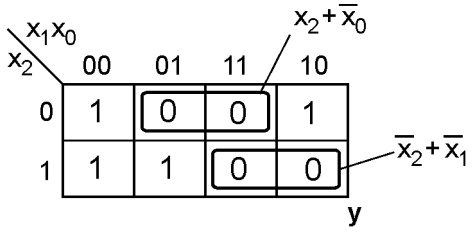
Teraz można narysować schemat układu:



Praktyczna realizacja układu zazwyczaj wykonywana jest z użyciem bramek NAND. Zatem wystarczy dokonać transformacji bramek AND i OR do postaci NAND poprzez „wstawienie" negatorów (dorysowanie kółek) na wyjściach bramek AND i wejściach bramek OR, co przedstawiono poniżej.



Dla uzyskania postaci iloczynowej należy skleić kratki zawierające wartości 0:



Każdą zakreśloną grupę „zer” zapisujemy w postaci sumy logicznej zmiennych wejściowych. W danej sumie nie występuje ta zmienna, która dla sklejonych kratek posiada obie wartości 0 i 1 (odczytane z krawędzi siatki). Ponadto zmienną mającą wartość 1 zapisujemy ze znakiem negacji, a zmienną równą 0 zapisujemy bez negacji (czyli wprost). Dla powyższej siatki mamy dwie grupy, stąd odpowiednie wyrażenia sumacyjne są następujące:

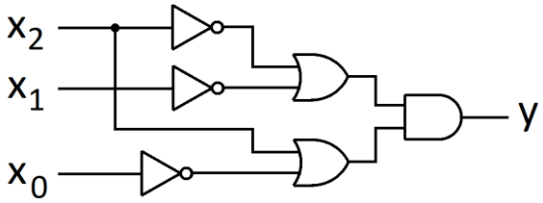
$$\left.\begin{array}{l} x_2 = 0 \\ x_1 = 0,1 \\ x_0 = 1 \end{array}\right\} \rightarrow (x_2 + \overline{x}_0),$$

$$\left.\begin{array}{l} x_2 = 1 \\ x_1 = 1 \\ x_0 = 0,1 \end{array}\right\} \rightarrow (\overline{x}_2 + \overline{x}_1).$$

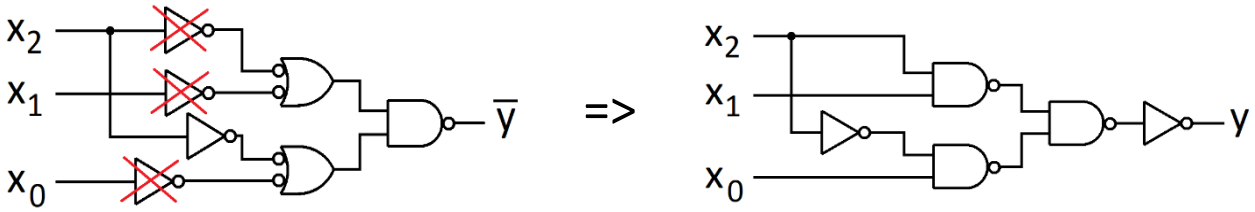
Następnie otrzymane wyrażenia sumacyjne zapisujemy w postaci ich iloczynu logicznego:

$$y = (x_2 + \overline{x}_0) \cdot (\overline{x}_2 + \overline{x}_1).$$

Teraz można narysować schemat układu:



Praktyczna realizacja układu zazwyczaj wykonywana jest z użyciem bramek NAND. Zatem wystarczy dokonać transformacji bramek AND i OR do postaci NAND poprzez „wstawienie” negatorów (dorysowanie kółek) na wyjściach bramek AND i wejściach bramek OR, co przedstawiono poniżej. Należy zauważyć, że nastąpiło zanegowanie wszystkich zmiennych układu w porównaniu do schematu pierwotnego. Spowodowało to konieczność zastosowania dodatkowej bramki NOT na wyjściu.



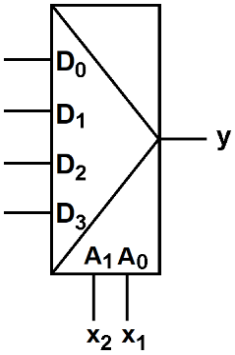
Ponadto każdy układ kombinacyjny można zrealizować z użyciem dowolnego multipleksera. Przy czym w pierwszym kroku należy do jego wejść adresowych dołączyć najbardziej znaczące zmienne wejściowe układu kombinacyjnego. Pozostałe zmienne wejściowe należy zastosować do zrealizowania „podukładów” realizujących funkcje logiczne dołączone do każdego wejścia danych multipleksera. Właśnie te funkcje należy określić w postępowaniu projektowym.

Zatem najpierw zapisujemy tablicę prawdy dla zadanej funkcji logicznej:

x_2	x_1	x_0	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

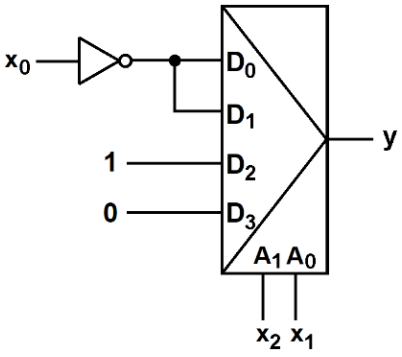
Przykładowo, zastosujmy multiplekser o czterech wejściach danych, czyli o dwóch wejściach adresowych. Zatem w pierwszym kroku dołączamy do wejść adresowych multipleksera A_1 i A_0 odpowiednio zmienne x_2 i x_1. Takie postępowanie powoduje pogrupowanie wierszy w tablicy prawdy w sekcje o stałych wartościach dla par zmiennych x_2x_1, co zilustrowano poniżej. Stąd idea podłączenia bardziej znaczących zmiennych układu kombinacyjnego do wejść adresowych multipleksera, z zachowaniem kolejności indeksowania.

x_2	x_1	x_0	y
0	**0**	0	1
0	**0**	1	0
0	**1**	0	1
0	**1**	1	0
1	**0**	0	1
1	**0**	1	1
1	**1**	0	0
1	**1**	1	0
A_1	A_0		



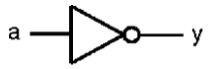
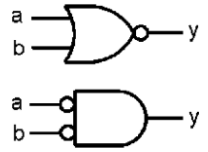
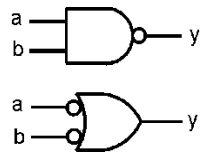
Teraz należy porównać wartości wyjścia y z wartościami zmiennej x0. Jeżeli dla obu wierszy w obrębie jednej sekcji, wyjście y ma wartość stałą 0 lub 1, to do stosownego wejścia danych multipleksera dołączamy odpowiednio stan 0 lub 1, bez uwzględnienia x_0. Natomiast, jeżeli sygnał y przyjmuje wartości 0 i 1, to do wejścia danych podłączamy zmienną x_0. W przeciwnym przypadku dołączamy zanegowaną zmienną x_0. Wynik postępowania przedstawia poniższy obrazek.

x_2	x_1	x_0	y	multiplekser
0	0	**0**	**1**	D_0 <= not x_0
0	0	**1**	**0**	
0	1	**0**	**1**	D_1 <= not x_0
0	1	**1**	**0**	
1	0	0	**1**	D_2 <= 1
1	0	1	**1**	
1	1	0	**0**	D_3 <= 0
1	1	1	**0**	
A_1	A_0			

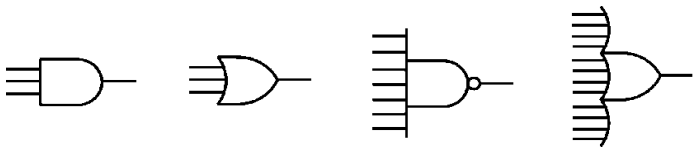


BRAMKI LOGICZNE

Układy kombinacyjne buduje się przy użyciu bramek, czyli elementów elektronicznych realizujących funkcje logiczne. Zestawienie podstawowych oznaczeń graficznych i realizowanych funkcji zawarto w poniższej tabeli. Kółeczko na wyjściu bramki oznacza operację negacji funkcji logicznej realizowanej przez tę bramkę. Ponadto kółeczko na wejściu symbolizuje negację sygnału wejściowego.

Nazwa bramki	Symbol graficzny	Funkcja logiczna	Tablica prawdy
Suma (OR)		$y = a + b$	a b | y 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Iloczyn (AND)		$y = a\,b$	a b | y 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
Negacja (NOT)		$y = \bar{a}$	a | y 0 | 1 1 | 0
Negacja sumy (NOR)		$y = \overline{a + b} = \bar{a}\,\bar{b}$	a b | y 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Negacja iloczynu (NAND)		$y = \overline{a\,b} = \bar{a} + \bar{b}$	a b | y 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
Suma modulo 2 (XOR)		$y = a\bar{b} + \bar{a}b = a \oplus b$	a b | y 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0

Symbol graficzny bramki o większej liczbie wejść rysujemy z odpowiednią liczbą kresek oznaczających wejścia jak na rys. 5. Należy pamiętać, że dla dużej liczby wejść nie zwiększamy obrysu symbolu bramki.



Rys. 5 Symbole graficzne bramek o zwiększonej liczbie wejść

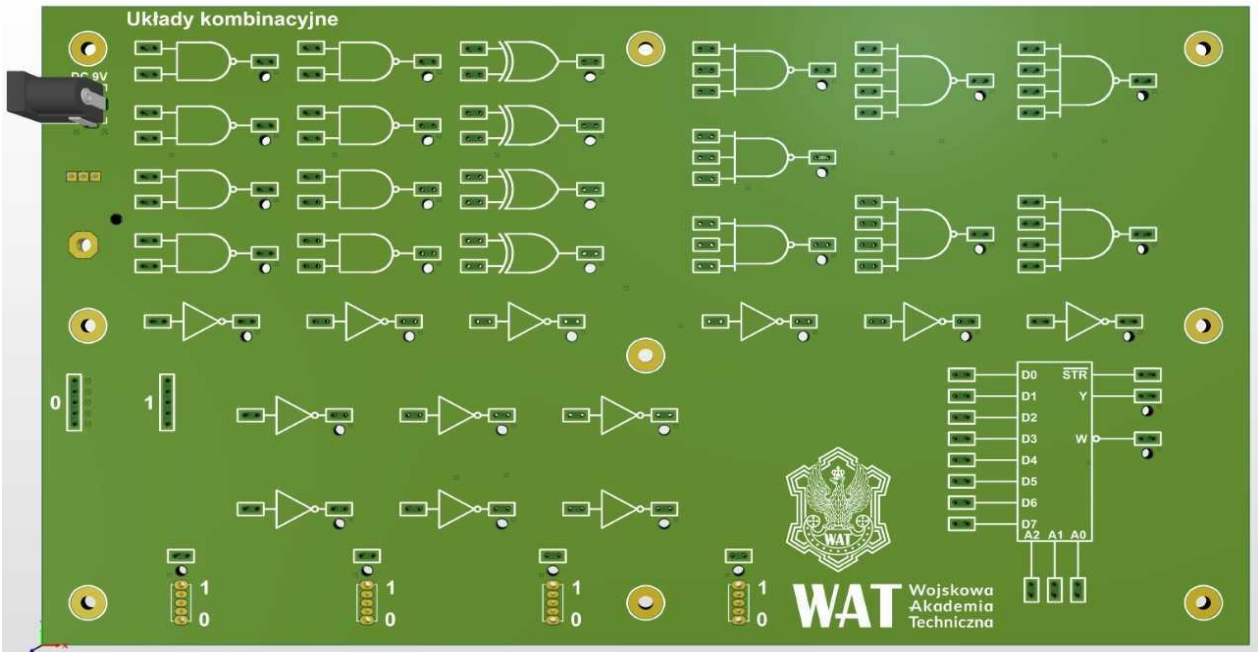
III. REALIZACJA ĆWICZENIA LABORATORYJNEGO

ZESTAW LABORATORYJNY

Zadania projektowe realizowane są z użyciem zestawu laboratoryjnego (rys. 6) zawierającego m.in.:

- osiem dwuwejściowych bramek NAND – układy scalone 74HC00D,
- trzy trójwejściowe bramki NAND – układ scalony 74HC10D,

- cztery czterowejściowe bramki NAND – układ scalony 74HC20D,
- cztery dwuwejściowe bramki XOR – układ scalony 74HC86D,
- dwanaście inwerterów – układy scalone 74HC04D,
- multiplekser o ośmiu wejściach danych – układ scalony 74HC151D.



Rys. 6 Zestaw laboratoryjny

Badane układy kombinacyjne realizowane są poprzez wykonanie połączeń pomiędzy wejściami i wyjściami układów scalonych, udostępnionych na płycie zestawu laboratoryjnego. Stany logiczne na wyjściach układów kombinacyjnych mogą być obserwowane z użyciem wbudowanych w zestaw diod elektroluminescencyjnych. Wszystkie układy scalone są wspólnie zasilane, natomiast do wprowadzania stanów logicznych na wejścia układów kombinacyjnych przeznaczone są:

- cztery przełączniki suwakowe wraz z wejściami czterech inwerterów nad tymi przełącznikami,
- dwa rzędy wyprowadzeń stanów logicznych 0 i 1.

UWAGA! Zabrania się łączenia ze sobą wyjść układów scalonych.

WYKONANIE ĆWICZENIA

Przeprowadzić minimalizację funkcji logicznych z użyciem siatek Karnaugh i zbudować układy kombinacyjne realizujące funkcje logiczne według poleceń prowadzącego ćwiczenie.

OPRACOWANIE WYNIKÓW

1. Sprawozdanie z ćwiczenia laboratoryjnego powinno zawierać zwięzłe opisy postawionych zadań projektowych wraz z rozwiązaniami, w szczególności z siatkami Karnaugh i schematami zaprojektowanych układów.
2. W sprawozdaniu należy również zawrzeć rozwiązania dodatkowych problemów projektowych postawionych przez prowadzącego ćwiczenie.

ZALICZENIE ĆWICZENIA

1. Zaliczenie kolokwium wstępnego oraz poprawne wykonanie zadań laboratoryjnych postawionych przez osobę prowadzącą ćwiczenie.
2. Złożenie sprawozdania, zawierającego zwięzły opis wykonanych zadań, wnioski i poprawne odpowiedzi na postawione pytania.

PRZYKŁADOWE ZADANIA PROJEKTOWE

1. Zbudować bramkę AND o liczbie wejść n>2 wskazanej przez prowadzącego ćwiczenie, a następnie sprawdzić działanie tej bramki na zestawie laboratoryjnym.

2. Korzystając z bramek NAND i/lub XOR dostępnych w zestawie, zaprojektować układ kombinacyjny, określony zbiorami: T_4, F_4 i D_4, wskazanymi przez prowadzącego ćwiczenie:
   a) dokonać minimalizacji funkcji logicznej względem jedynek,
   b) dokonać minimalizacji funkcji logicznej względem zer.
3. Układ z zadania 2 wykonać praktycznie z użyciem multipleksera o ośmiu wejściach danych.
4. Korzystając z multipleksera o ośmiu wejściach danych (MUX 8x1), zbudować układ kombinacyjny, określony zbiorami: T_3, F_3 i D_3, wskazanymi przez prowadzącego ćwiczenie.
5. Korzystając z multipleksera o ośmiu wejściach danych, dokonać implementacji określonych bramek logicznych o trzech wejściach.
6. Korzystając z bramek NAND dostępnych w zestawie, wykonać praktycznie schematy zastępcze dwuwejściowej bramki XOR/XNOR.
7. Przy użyciu bramek dostępnych w zestawie zbudować układy kombinacyjne o dwóch wejściach danych:
   a) dekoder,
   b) multiplekser,
   c) demultiplekser.
8. Dokonać syntezy układu kombinacyjnego, opisanego zbiorami:
   a) $T_4 = \{ 2, 3, 6, 7, 8, 9, 12, 13 \}$,
   b) $T_4 = \{ 0, 1, 4, 5, 10, 11, 14, 15 \}$,

   a następnie wskazać, której bramce logicznej odpowiada ten układ.
9. Zbudować układ dekodera o trzech wejściach danych złożonego z dwóch dekoderów o dwóch wejściach danych oraz jednego inwertera.
10. Zbudować układ multipleksera o czterech wejściach danych złożonego z układu trzech multiplekserów o dwóch wejściach danych.

PRZYKŁADOWE ZESTAWY ZBIORÓW OKREŚLAJĄCYCH FUNKCJE LOGICZNE

1. $T_4 = \{ 0, 1, 3, 6, 7, 15 \}$, $F_4 = \{ 4, 5, 8, 9, 10, 13, 14 \}$, $D_4 = \{ 2, 11, 12 \}$.
2. $T_4 = \{ 0, 1, 4, 5, 6, 9 \}$, $F_4 = \{ 2, 3, 7, 8, 10, 11, 12, 14, 15 \}$, $D_4 = \{ 13 \}$.
3. $T_4 = \{ 2, 5, 7, 11, 13, 15\}$, $F_4 = \{ 0, 1, 4, 6, 8, 10, 12, 14 \}$, $D_4 = \{ 3, 9 \}$.
4. $T_4 = \{ 0, 2, 9, 10, 12, 13, 14 \}$, $F_4 = \{ 3, 4, 5, 6, 11, 15 \}$, $D_4 = \{ 1, 7, 8 \}$.
5. $T_4 = \{ 0, 5, 8, 11, 13, 15 \}$, $F_4 = \{ 2, 3, 4, 6, 7, 12, 14 \}$, $D_4 = \{ 1, 9, 10 \}$.
6. $T_4 = \{ 0, 1, 2, 10, 13 \}$, $F_4 = \{ 2, 3, 4, 6, 7, 12, 14 \}$, $D_4 = \{ 4, 5, 8, 15 \}$.
7. $T_4 = \{ 2, 3, 5, 13, 14, 15 \}$, $F_4 = \{ 0, 1, 6, 7, 8, 9, 10, 11 \}$, $D_4 = \{ 4, 12 \}$.
8. $T_4 = \{ 0, 1, 3, 7, 8, 15 \}$, $F_4 = \{ 4, 5, 10, 11, 12, 13, 14 \}$, $D_4 = \{ 2, 6, 9 \}$.

IV. ZAGADNIENIA DO OPRACOWANIA PRZED PRZYSTĄPIENIEM DO ĆWICZENIA

1. Jaki układ logiczny nazywany kombinacyjnym?
2. Definicja funkcji logicznej, postaci kanonicznej funkcji, wyrażenia normalnego: alternatywnego i koniunkcyjnego.
3. Wymienić sposoby przedstawiania funkcji logicznej.
4. Zdefiniować podstawowe funkcje logiczne.
5. Omówić konstrukcję mapy Karnaugh i wykorzystanie jej do minimalizacji funkcji logicznej.
6. Podać algorytm syntezy układu kombinacyjnego metodą symboli dualnych.
7. Na czym polega zjawisko hazardu w układach logicznych?
8. Na czym polega hazard statyczny w zerach?
9. Na czym polega hazard statyczny w jedynkach?
10. Na czym polega hazard dynamiczny?
11. Wymienić przykłady układów wykorzystujących zjawisko skończonego czasu propagacji sygnału przez bramki logiczne.

V. LITERATURA I MATERIAŁY POMOCNICZE

1. Materiały wykładowe.
2. J. Kalisz: Podstawy elektroniki cyfrowej, WKŁ, Warszawa 2007.
3. J. Tyszer: Układy cyfrowe, Wyd. Politechniki Poznańskiej, 2000.
4. J.F. Wakerly: Digital Design, Principles and Practices, 4th Edition, Pearson/Prentice Hall, 2005.
5. B. Wilkinson: Układy cyfrowe, WKŁ, 2000.
6. A. Skorupski: Podstawy techniki cyfrowej, WKŁ, 2001.