Architektura Systemów Komputerów
Laboratoria

Sprawozdanie z układów logicznych	Rok 2021
1.Nazwisko i imię – sprawozdanie pierwsze: Wiktor Sadowy (260373) Filip Strózik (260377)	**Ćwiczenie nr: 2**
2.Nazwisko i imię – sprawozdanie drugie: Wiktor Sadowy (260373) Filip Strózik (260377)	**Temat ćwiczenia:** Układy kombinacyjne
Grupa laboratoryjna nr (u prowadzącego): Z01-45i	**Dzień tygodnia:** Piątek
Płyta montażowa nr (z tyłu nadajnika):	**Godziny zajęć (od-do):** 17:05-18:45

Spis treści

1. Cel zadania ... 3
2. Wstęp (zagadnienia do samodzielnego opracowania) ... 3
3. Schemat multipleksera 74151 ... 5
4. Blok logiczny multipleksera ... 6
5. Specyfikacja układu oraz sposób działania ... 6
6. Tabela funkcji multipleksera 74151 ... 7
7. Zastosowanie multipleksera 74151 ... 7
8. Analiza funkcji A (minimalizacja) ... 8
9. Przedstawienie funkcji A za pomocą bramek NAND ... 9
10. Przedstawienie funkcji A za pomocą multipleksera ... 9
11. Przedstawienie funkcji A za pomocą bramek NAND oraz multipleksera ... 11
12. Schemat podłączenia układu A ... 12
13. Analiza funkcji B (minimalizacja) ... 14
14. Przedstawienie funkcji B za pomocą bramek NAND ... 15
15. Przedstawienie funkcji B za pomocą multipleksera ... 15
16. Przedstawienie funkcji B za pomocą bramek NAND oraz multipleksera ... 17
17. Schemat podłączenia układu B ... 18
18. Wnioski ... 19
19. Spis użytych układów ... 20
20. Bibliografia oraz link do układów ... 21

1. Cel zadania

Naszym zadaniem jest zrealizować następujące funkcje $f(a,b,c,d) = \overline{\overline{\overline{(a + \bar{b} + \bar{c})} + d}}$ oraz $f(a,b,c) = \overline{ab\bar{c} + \bar{a}\bar{b}c}$ w minimalnym (dwupoziomowym) układzie wykorzystującym bramki NAND oraz układzie wykorzystującym multiplekser 74151 (MUX). Musimy też udowodnić, że oba układy realizują tą samą funkcję czyli dla każdego stanu na wejściach przyjmują takie same stany na wyjściach.

2. Wstęp (zagadnienia do samodzielnego opracowania)

Przed wykonaniem zadania zdefiniujmy pojęcia z których będziemy korzystać podczas tego zadania:

a) Wyrażenie boolowskie – dowolne odwzorowanie f: B^n -> B gdzie B = {0, 1} a n to liczba argumentów. Przykładowym wyrażeniem boolowskim jest f(a, b) = $\bar{a}b$ gdzie zmienna a jest zanegowana (czyli dla wartości a=0 przyjmuje wartość 1 i odwrotnie). Następująca funkcji jest też zapisana w postaci iloczynu zmiennych co oznacza, że wartość $\bar{a}$ i b musi być równa 1, żeby funkcja przyjmowała wartość 1. W wyrażeniach boolowskich można spotkać się też z sumą argumentów np. f(a, b) = a + b. Wówczas wyrażenie przyjmuje wartość 1 gdy co najmniej jeden argument przyjmuje wartość 1.

b) Funkcja przełączająca – takie odwzorowanie, które dla kombinacji argumentów $x_1, x_2, \ldots, x_n$ przyjmujących wartości „0" lub „1" przyporządkowuje rozwiązanie ze zbioru {0, 1}. Należy zauważyć tutaj powiązanie z wyrażeniem boolowskim, którego argumenty też przyjmują wartości „0" lub „1" oraz którego rozwiązanie też należy do zbioru {0, 1}

c) Tablica prawdy – tablica w której wypisujemy wszystkie kombinacje zmiennych wejściowych oraz odpowiadające im wartości funkcji. Dla funkcji f(a, b) = ab tak będzie wyglądała tabela prawdy:

a	b	f
0	0	0
0	1	0
1	0	0
1	1	1

d) Prawa De Morgana – twierdzenia w logice matematycznej i teorii mnogości. Głównie korzystamy z następujących praw De Morgana
   a. Prawo do zaprzeczania sumy: negacja sumy jest równoważna iloczynowi negacji. To prawo jest przedstawiane w następujący sposób

      $$\sim (a \cup b) = (\sim a) \cap (\sim b)$$

      Gdzie a i b to wyrażenia boolowskie
   b. Prawo do zaprzeczenia iloczynu: negacja iloczynu jest równoważna sumie negacji.

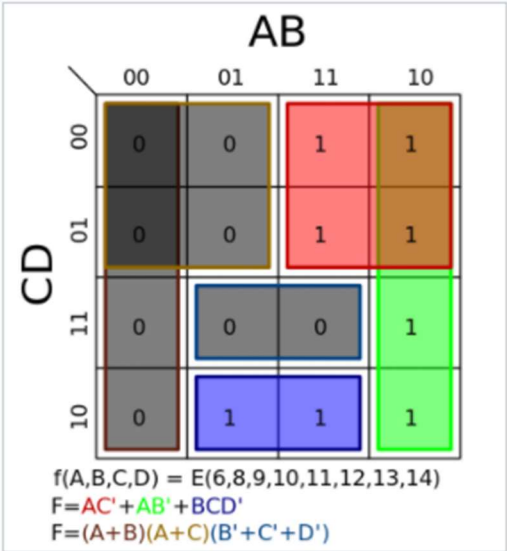
      $$\sim (a \cap b) = (\sim a) \cup (\sim b)$$

e) Tablica Karnaugha – Jest to przekształcona tablica prawdy przedstawiona w postaci prostokątnej tablicy tak aby spełniała własność kodu Graya. Dla funkcji f(A, B, C, D) tak prezentuje się tablica Karnaugha

AB \ CD	00	01	11	10
00	F(0,0,0,0)	F(0,0,0,1)	F(0,0,1,1)	F(0,0,1,0)
01	F(0,1,0,0)	F(0,1,0,1)	F(0,1,1,1)	F(0,1,1,0)
11	F(1,1,0,0)	F(1,1,0,1)	F(1,1,1,1)	F(1,1,1,0)
10	F(1,0,0,0)	F(1,0,0,1)	F(1,0,1,1)	F(1,0,1,0)

Celem minimalizacji funkcji boolowskiej wpisujemy wartości do tabeli jak powyżej i następnie grupujemy pola o wybranej wartości (1 jeżeli chcemy uzyskać funkcję w postaci sumy oraz 0 jeżeli chcemy uzyskać funkcję w postaci iloczynu). Grupy muszę mieć kształt prostokąta o długościach boku będących potęgami dwójki (przy czym może on przechodzić przez krawędź

tablicy). Zmienne które w danej grupie przyjmują wartość 1 będą występowały w postaci prostej, jeżeli przyjmują wartość 0 to wystąpią w postaci zanegowanej, a jeżeli przyjmują obie wartości to są one pomijane. Przykładowa minimalizacja funkcji została pokazana poniżej.



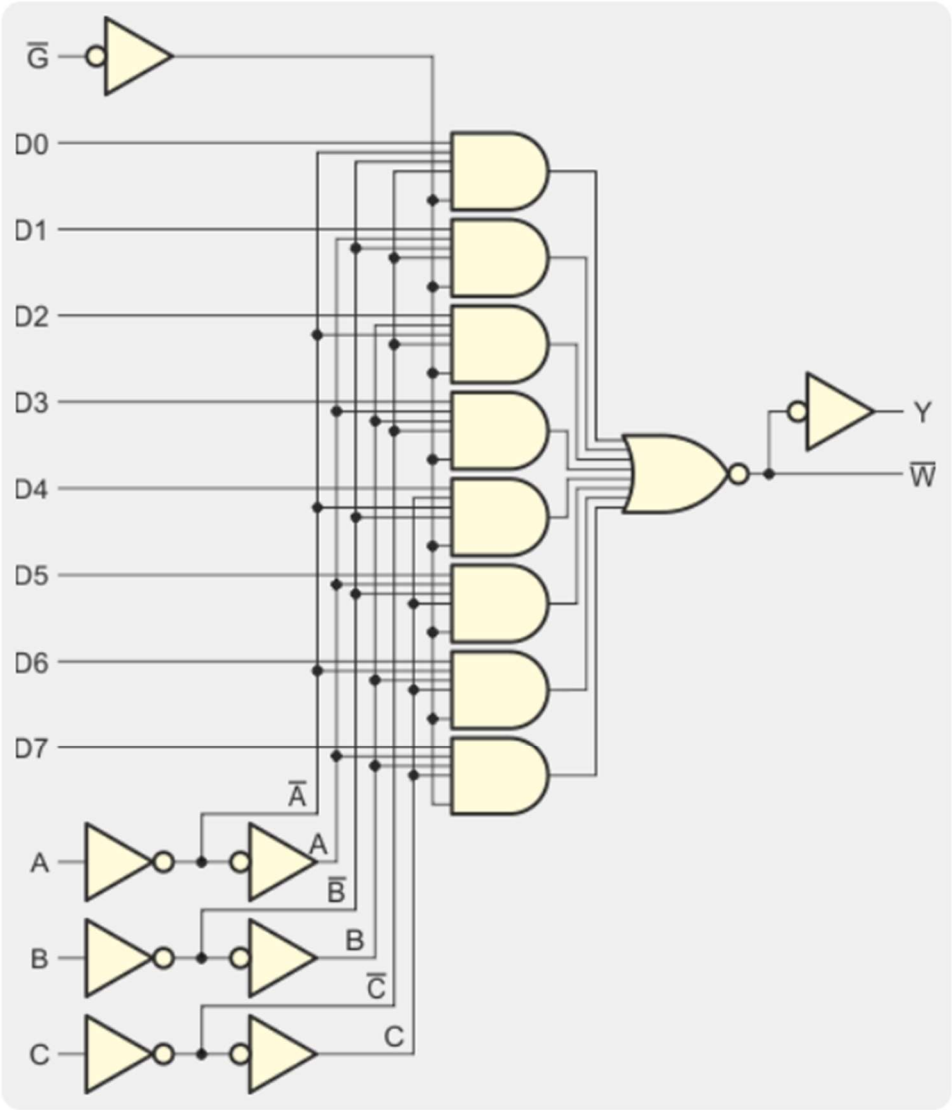
f) Metoda zamiany układu dwupoziomego AND-OR na układ wykorzystujący bramki NAND – Zauważmy, że możemy skorzystać z prawa zaprzeczania koniunkcji oraz, z tego, że podwójna negacja wejścia jest równa temu wejściowi. Załóżmy, że mamy następującą funkcję: f(a, b) = a + $\bar{b}$. Możemy podwójnie zanegować wejścia a i otrzymamy f(a, b) = $\bar{\bar{a}}$ + $\bar{b}$. Teraz korzystając z prawa zaprzeczania koniunkcji otrzymamy f(a, b) = $\overline{\bar{a} * b}$. Można zauważyć, że ta funkcja wykorzystuje bramki NAND.

g) Otrzymywanie układów wykorzystujących tylko multipleksery (MUX) – żeby otrzymać układ wykorzystujący multiplekser potrzebujemy mieć funkcję w postaci kanonicznej sumy. Załóżmy, że mamy następującą funkcję f(a, b) = a + $\bar{a} * b$. Zróbmy prostą tabelę prawdy dla tej funkcji

a	b	f(a, b)
0	0	0
0	1	1
1	0	1
1	1	1

Powyższa funkcja w kanonicznej postaci będzie miała postać $f(a,b) = \sum(1,2,3)$. Mając funkcję w takiej postaci możemy stworzyć układ wykorzystujący tylko multiplekser.

3. Schemat multipleksera 74151



Rysunek 1. Układ 74151

D0, D1, …, D7 – wejścia danych

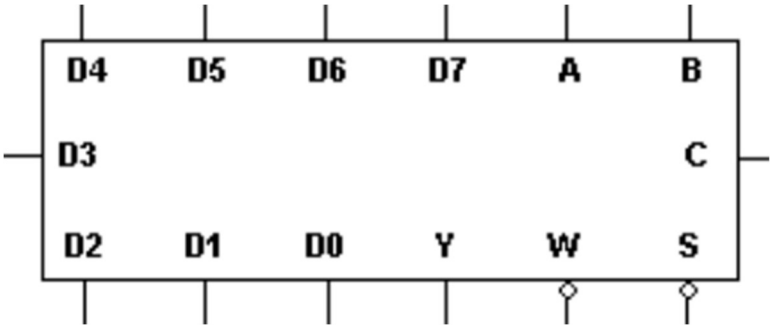
$\bar{G}$ – wejście sterujące działaniem układu

A, B, C – wejścia adresowe

$\overline{W}$ – wyjście odwracające

Y – wyjście nieodwracające

4. Blok logiczny multipleksera



Rysunek 2. Blok logiczny układu 74151

D0, D1, …, D7 – wejścia danych

$\bar{G}$ – wejście sterujące działaniem układu

A, B, C – wejścia adresowe

$\overline{W}$ – wyjście odwracające

Y – wyjście nieodwracające

S – wejście strobujące

5. Specyfikacja układu oraz sposób działania

Multiplekser 74151 jest układem kombinacyjnym składającym się z 8 wejść. Gdy do $\bar{G}$ zostanie przyłożony stan wysoki to wówczas na wyjściu nieodwracającym będzie stan niski, a na wyjściu odwracającym stan wysoki. Gdy do $\bar{G}$ zostanie przyłożony stan niski to wówczas na wyjściu nieodwracającym będzie stan równy jednemu z wejść danych a na wyjściu odwracającym będzie stan równy negacji tego wejścia danych. Należy zauważyć, że wejścia adresowe wyznaczają, które wejście danych będzie na wyjściu np. gdy przyłożymy stan niski do każdego do każdego wejścia adresowego to na wyjściu otrzymamy sygnał z wejścia danych nr. 0, gdy przyłożymy stan wysoki do każdego wejścia adresowego to na wyjściu otrzymamy sygnał z wejścia danych nr. 7 itp. Wszystkie możliwe ustawienia multipleksera obrazuje poniższa tabela funkcji.

6. Tabela funkcji multipleksera 74151

Wejścia				Wyjścia	
Wybór			STROBE		
C	B	A	$\overline{G}$	Y	$\overline{W}$
X	X	X	1	0	1
0	0	0	0	D0	$\overline{D0}$
0	0	1	0	D1	$\overline{D1}$
0	1	0	0	D2	$\overline{D2}$
0	1	1	0	D3	$\overline{D3}$
1	0	0	0	D4	$\overline{D4}$
1	0	1	0	D5	$\overline{D5}$
1	1	0	0	D6	$\overline{D6}$
1	1	1	0	D7	$\overline{D7}$

Rysunek 3. Tabela funkcji układu 74151

0 – stan niski

1 – stan wysoki

X – nie ma znaczenia jaki stan przyłożymy

D0, D1, …, D7 – wyjście

$\overline{D0}, \overline{D1}, \ldots, \overline{D7}$ – zanegowane wyjście

7. Zastosowanie multipleksera 74151

Multipleksera 74151 często używa się do usprawnienia komunikacji pomiędzy kanałami. Wówczas przeważnie podłącza się wyjście multipleksera do wejścia demultipleksera, aby z powrotem odzyskać dane przekazane multiplekserowi. Taka implementacja przekazu danych jest tańsza i sprawniejsza niż implementowanie oddzielnych kanałów do przekazu pojedyńczych danych. Praktyczne zastosowania tego możemy znaleźć np. w:

a) systemach komunikacyjnych. Wygodniej jest przekazywać takie dane jak audio czy video z różnych kanałów za pomocą jednego przewodu
b) komputerach. Często pamięć komputerowa korzysta z multipleksera po to, aby zwiększyć jej pojemność, ale także po to, aby zmniejszyć liczbę przewodów potrzebnych do podłączenia pamięci do innych części komputera

8. Analiza funkcji A (minimalizacja)

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

Aby zminimaliować funkcję skorzystam z tablicy Karnougha. Jednak do tego potrzebuję mieć tablicę prawdy, aby móc wpisać poprawne wartości to tablicy Karnougha.

a	b	c	d	$\overline{a+\ \overline{b}}$	$\overline{\overline{(a+\ \overline{b}\ +\ \overline{c})}}$	$\overline{\overline{\overline{(a+\ \overline{b}\ +\ \overline{c})}} + d}$
0	0	0	0	1	0	1
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	0
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	0	1	0	0	0
1	1	1	0	0	1	0
1	1	1	1	0	1	0

Tabela 1. Tabela prawdy funkcji A

Teraz wpisuję odpowiednie wartości do tablicy Karnougha:

cd↓ ab→	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	0

Tabela 2. Tablica Karnaugh dla funkcji A

Należy zauważyć, że możemy połączyć cały pierwszy wiersz (kolor zielony) oraz pierwszy i ostatni wiersz pierwszej kolumny (kolor niebieski). Tak będzie prezentowała się zminimalizowana funkcja:

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

Do przedstawienia powyższej funkcji za pomocą bramek NAND skorzystkam z praw de Morgana, które mówią:

$$\overline{ab} = \bar{a} + \bar{b}$$

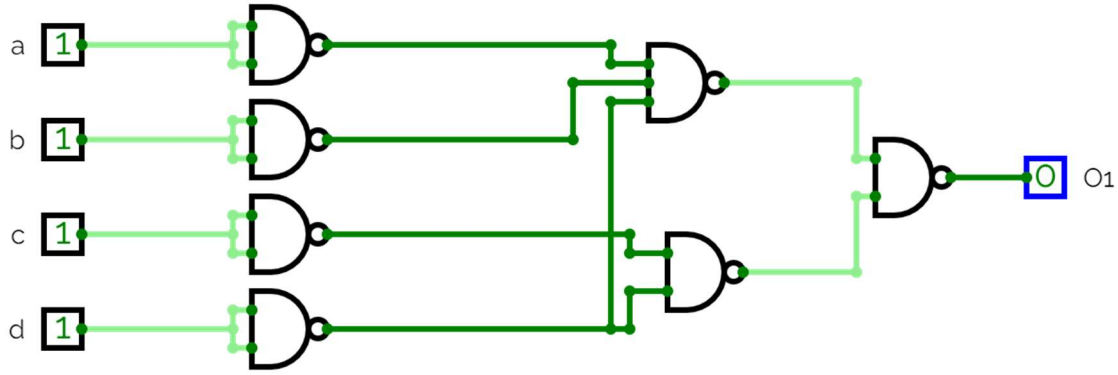
$$\overline{a+\ b} = \bar{a} * \bar{b}$$

Korzystając z poniższych praw możemy sprowadzić funkcję do następującej postaci

$$f(a,b,c,d) = \bar{c}\bar{d} + \bar{a}\bar{b}\bar{d} = \overline{\overline{\bar{c}\bar{d} + \bar{a}\bar{b}\bar{d}}} = \overline{\overline{\bar{c}\bar{d}} * \overline{\bar{a}\bar{b}\bar{d}}}$$

Powyższą funkcję da rady przedstawić za pomocą bramek NAND, więc dalsze modyfikacje są niepotrzebne.

9. Przedstawienie funkcji A za pomocą bramek NAND



Rysunek 4. Schemat logiczny układu funkcji A za pomocą bramek NAND

10. Przedstawienie funkcji A za pomocą multipleksera

Aby przedstawić funkcję używając multipleksera, potrzeba przedstawić funkcję w postaci kanonicznej sumy. Potrzebne nam będą wartości logiczne na wyjściu funkcji.

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

Mogę skorzystać z poprzednio utworzonej tabeli prawdy i przepisać tylko te wartości a, b, c, d dla których funkcja na wyjściu będzie miała wartość 1

a	b	c	d	$f(a,b,c,d)$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
1	1	0	0	1

Tabela 3. Tabela prawdy dla funkcji A, która zawiera tylko te argumenty dla których funkcja przyjmuje wartość 1

Do multipleksera 74151 musimy podać tylko trzy wartości sterujące, zauważając że wartość zmiennej d jest zawsze 0, można pominąć tą zmienną, lecz trzeba zauważyć, że zawsze gdy zmienna d przyjmuje wartość 1 funkcja zwraca wynik równy 0. Możemy skorzystać z tej zależności i zmienić funkcję na taką, która nie wykorzystuje zmiennej d, lecz zmienna d przyjmująca wartość 1 uaktywnia działanie G' która sprawia, że wynik działania będzie równy 0.

a	b	c	$f(a,b,c)$
0	0	0	1
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	1

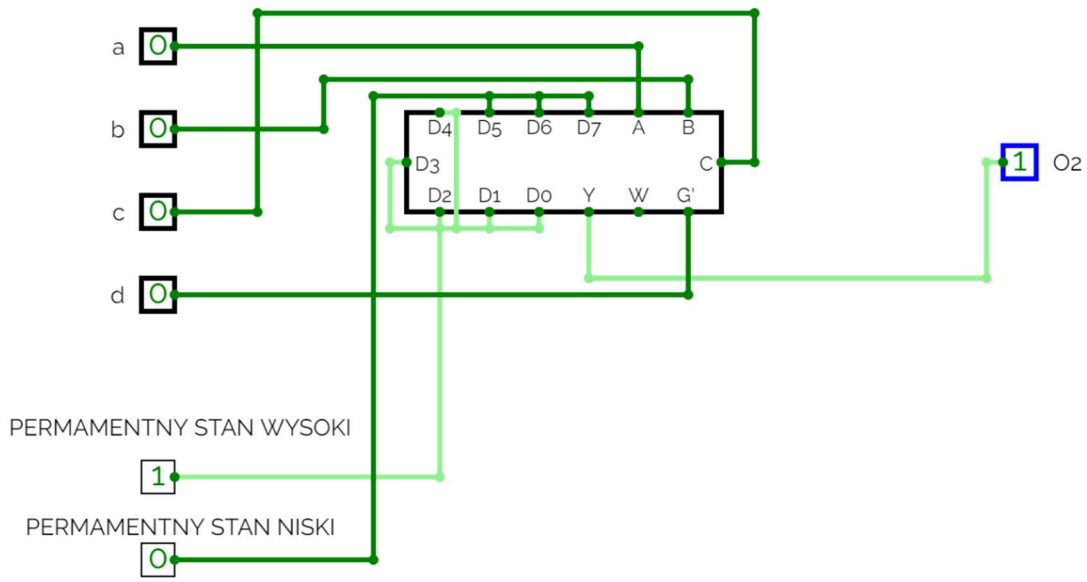
Tabela 4. Tabela prawdy dla funkcji A, która zawiera tylko te argumenty dla których funkcja przyjmuje wartość 1 bez zmiennej d

c	b	a	wartość cba w systemie dziesiętnym
0	0	0	0
1	0	0	4
0	1	0	2
0	0	1	1
0	1	1	3

Tabela 5. Tabela wartości cba w systemie dziesiętnym w zależności od stanów c, b, a.

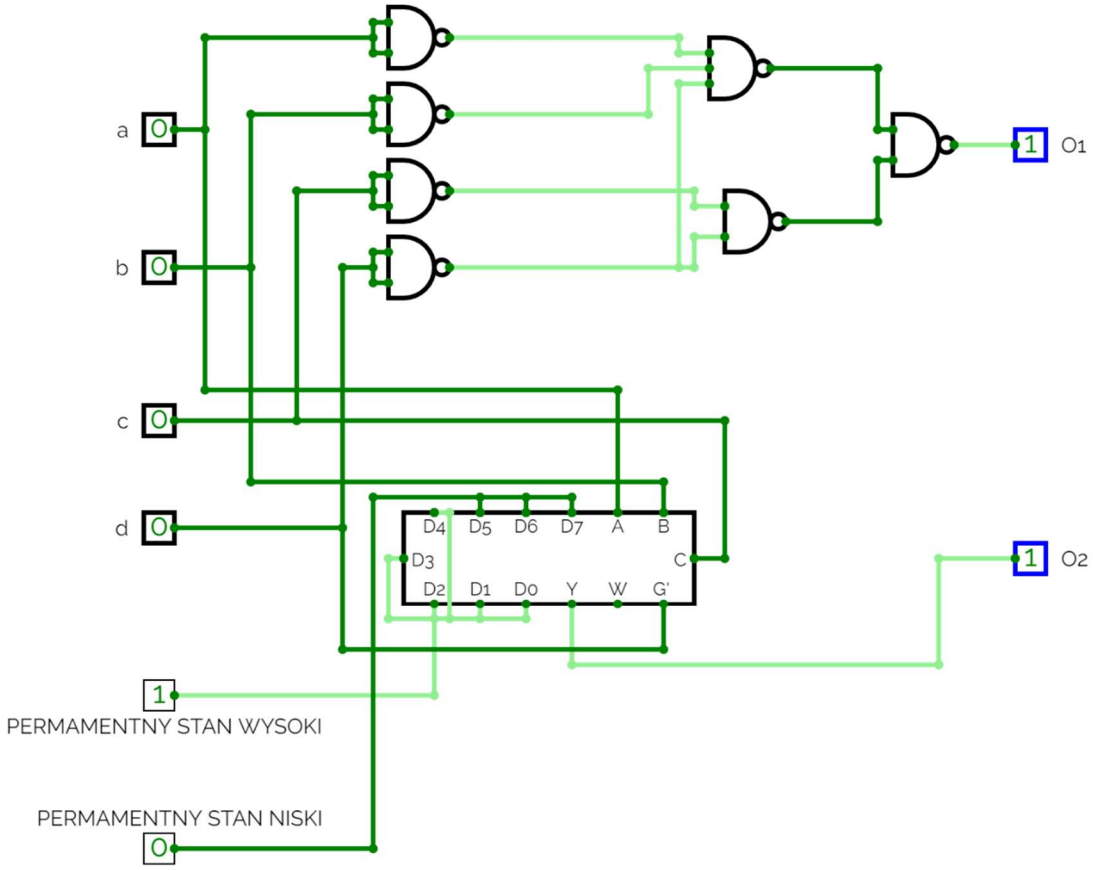
Nasza końcowa funkcja przyjmie więc następującą postać.

$$f(a,b,c) = \sum(0,1,2,3,4)$$



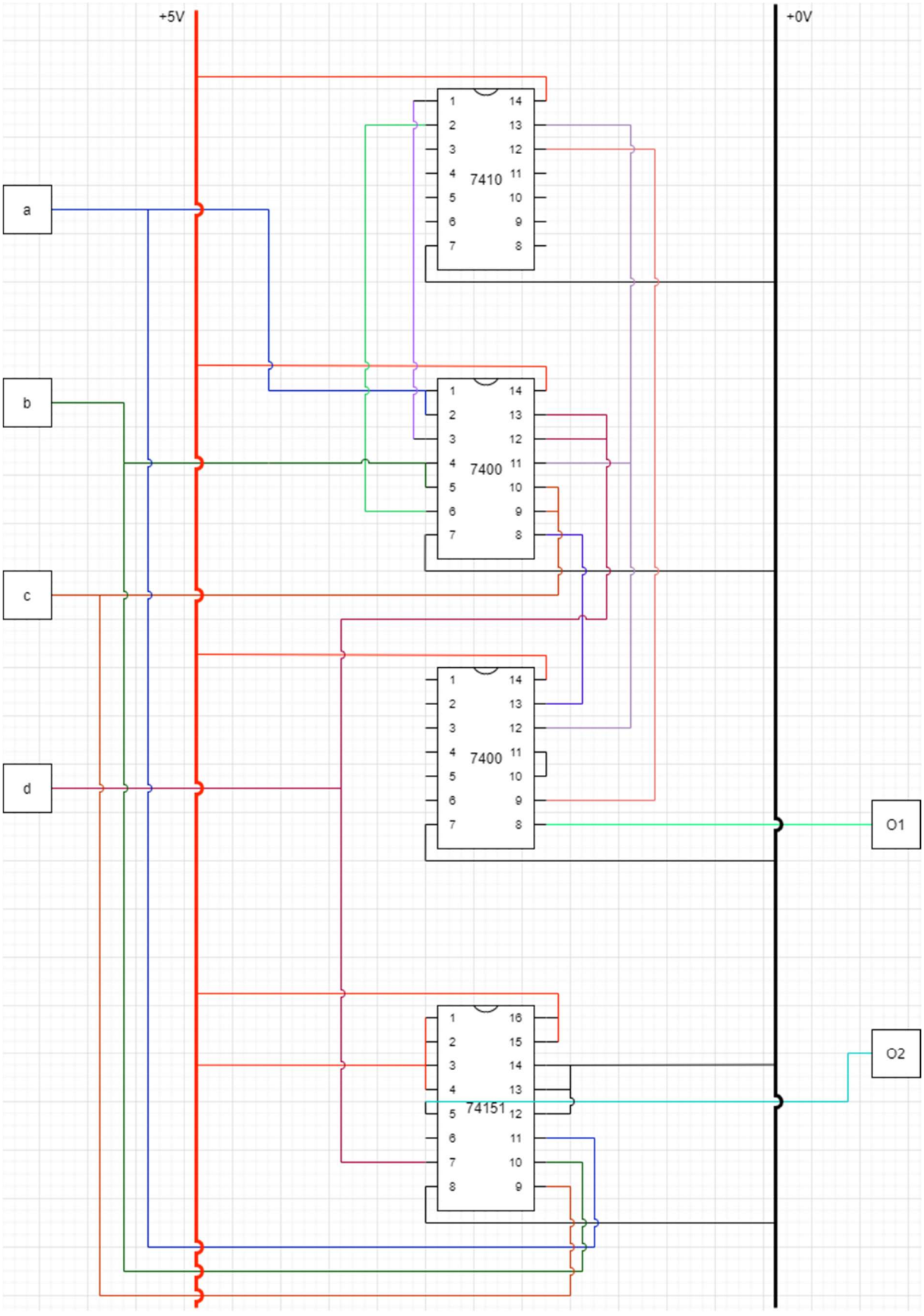
Rysunek 5. Schemat logiczny układu funkcji A za pomocą multipleksera.

11. Przedstawienie funkcji A za pomocą bramek NAND oraz multipleksera



Rysunek 6. Schemat logiczny układu funkcji A za pomocą bramek NAND oraz multipleksera.

12. Schemat podłączenia układu A



Rysunek 7. Schemat podłączenia układu realizującego funkcję A.

Tablica kroków działania dla powyższego schematu.

	WEJŚCIA				WYJŚCIA	
lp.	a	b	c	d	O1	O2
1	0	0	0	0	1	1
2	0	0	0	1	0	0
3	0	0	0	0	1	1
4	0	0	1	0	1	1
5	0	0	0	0	1	1
6	0	1	0	0	1	1
7	0	0	0	0	1	1
8	1	0	0	0	1	1
9	0	0	0	0	1	1
10	0	0	0	1	0	0
11	0	0	1	1	0	0
12	0	0	0	1	0	0
13	0	1	0	1	0	0
14	0	0	0	1	0	0
15	1	0	0	1	0	0
16	0	0	0	1	0	0
17	0	0	0	0	1	1
18	0	0	1	0	1	1
19	0	1	1	0	0	0
20	0	0	1	0	1	1
21	1	0	1	0	0	0
22	0	0	1	0	1	1
23	0	0	0	0	1	1
24	0	1	0	0	1	1
25	1	1	0	0	1	1
26	0	1	0	0	1	1
27	0	0	0	0	1	1
28	0	0	0	1	0	0
29	0	0	1	1	0	0
30	0	1	1	1	0	0
31	0	1	1	0	0	0
32	1	1	1	0	0	0
33	1	1	1	1	0	0
34	1	0	1	1	0	0
35	1	1	1	1	0	0
36	1	1	0	1	0	0

Tabela 6. Tabela kroków działania układu A.

Korzystając z tego, że jest to układ kombinacyjny czyli stan na wyjściu zależy tylko od stanach na wejściu możemy uprościć powyższą tabelę do następującej postaci

WEJŚCIA				WYJŚCIA	
a	b	c	d	O1	O2
0	0	0	0	1	1
0	0	0	1	0	0
0	0	1	0	1	1
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	0	0

1	0	1	1	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Tabela 7. Tabela prawdy dla układu A.

Możemy zauważyć, że tabela prawdy dla układu A jest zgodna z tabelą 1 – tabelą prawdy dla funkcji A. Dodatkowo na wyjściach O1 i O2 mamy te same stany co oznacza, że oba układy, jeden stworzony korzystając z bramek NAND, a drugi korzystający z multipleksera, są identyczne oraz realizują funkcję A.

13. Analiza funkcji B (minimalizacja)

$$f(a,b,c) = \overline{ab\bar{c} + \bar{a}\bar{b}c}$$

Aby zminimaliować funkcję skorzystam z tablicy Karnougha. Jednak do tego potrzebuję mieć tablicę prawdy, aby móc wpisać poprawne wartości to tablicy Karnougha.

a	b	c	$ab\bar{c}$	$\bar{a}\bar{b}c$	$ab\bar{c} + \bar{a}\bar{b}c$	$\overline{ab\bar{c} + \bar{a}\bar{b}c}$
0	0	0	0	0	0	1
0	0	1	0	1	1	0
0	1	0	0	0	0	1
0	1	1	0	0	0	1
1	0	0	0	0	0	1
1	0	1	0	0	0	1
1	1	0	1	0	1	0
1	1	1	0	0	0	1

Tabela 8. Tabela prawdy funkcji B

Teraz wpisuję odpowiednie wartości do tablicy Karnougha:

a↓	bc→	00	01	11	10
00		1	0	1	1
01		1	1	1	0

Tabela 9. Tablica Karnaugh dla funkcji B

Wszystkie połączenia zostały oznaczane na powyższej tablicy Karnougha za pomocą kolorów: żółty, zielony i niebieski. Funkcja zminimalizowana będzie wyglądała w następujący sposób

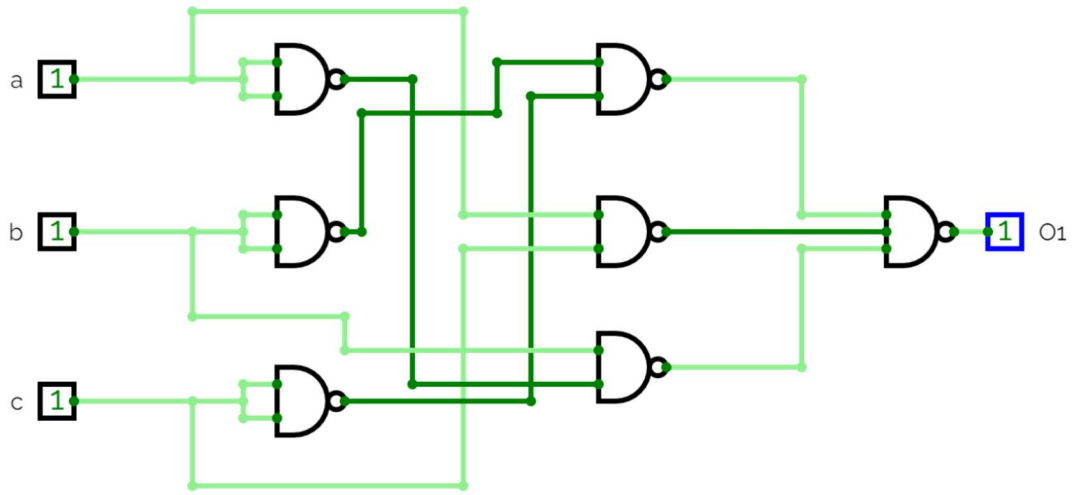
$$f(a,b,c) = \bar{b}\bar{c} + ca + \bar{a}b$$

Korzystając z praw de Morgana sprowadzam funkcję do następującej postaci

$$f(a,b,c) = \bar{b}\bar{c} + ca + \bar{a}b = \overline{\overline{\bar{b}\bar{c} + ca + \bar{a}b}} = \overline{\overline{\bar{b}\bar{c}} * \overline{ca} * \overline{\bar{a}b}}$$

Powyższą funkcję da rady przedstawić za pomocą bramek NAND, więc dalsze modyfikacje są niepotrzebne.

14. Przedstawienie funkcji B za pomocą bramek NAND



Rysunek 8. Schemat logiczny układu funkcji B za pomocą bramek NAND

15. Przedstawienie funkcji B za pomocą multipleksera

Aby przedstawić funkcję używając multipleksera, należy przedstawić funkcję w postaci kanonicznej sumy. Potrzebne nam będą wartości logiczne na wyjściu funkcji.

$$f(a,b,c) = \overline{ab\bar{c} + \bar{a}\bar{b}c}$$

Mogę skorzystać z poprzednio utworzonej tabeli prawdy i przepisać tylko te wartości a, b, c, d dla których funkcja na wyjściu będzie miała wartość 1

a	b	c	$f(a,b,c)$
0	0	0	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	1	1

Tabela 10. Tabela prawdy dla funkcji B, która zawiera tylko te argumenty dla których funkcja przyjmuje wartość 1

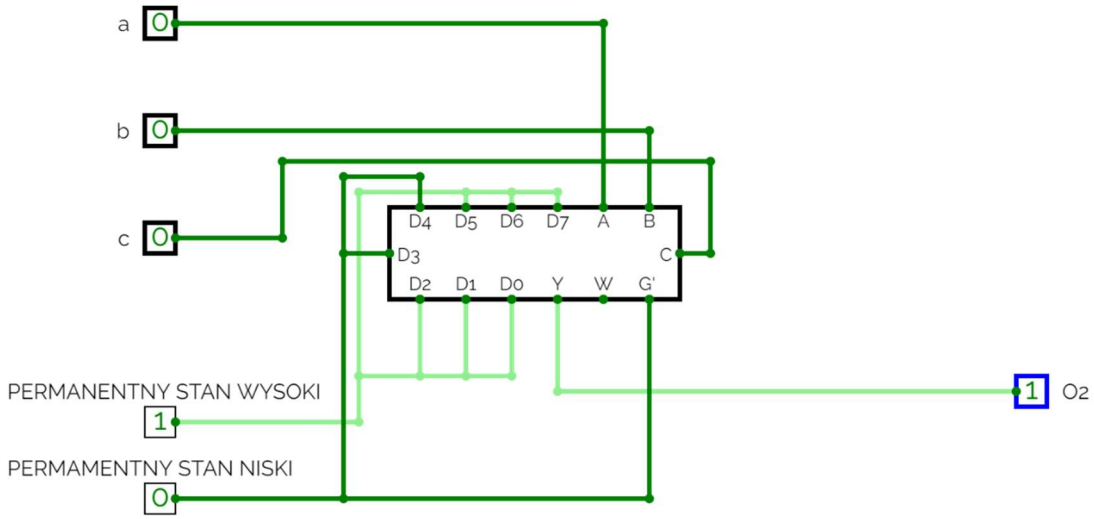
c	b	a	wartość cba w systemie dziesiętnym
0	0	0	0
0	1	0	2
1	1	0	6
0	0	1	1
1	0	1	5
1	1	1	7

Tabela 11. Tabela wartości cba w systemie dziesiętnym w zależności od stanów c, b, a.

Korzystając z powyższej tabeli prawdy mogę zapisać funkcję w postaci kanonicznej sumy

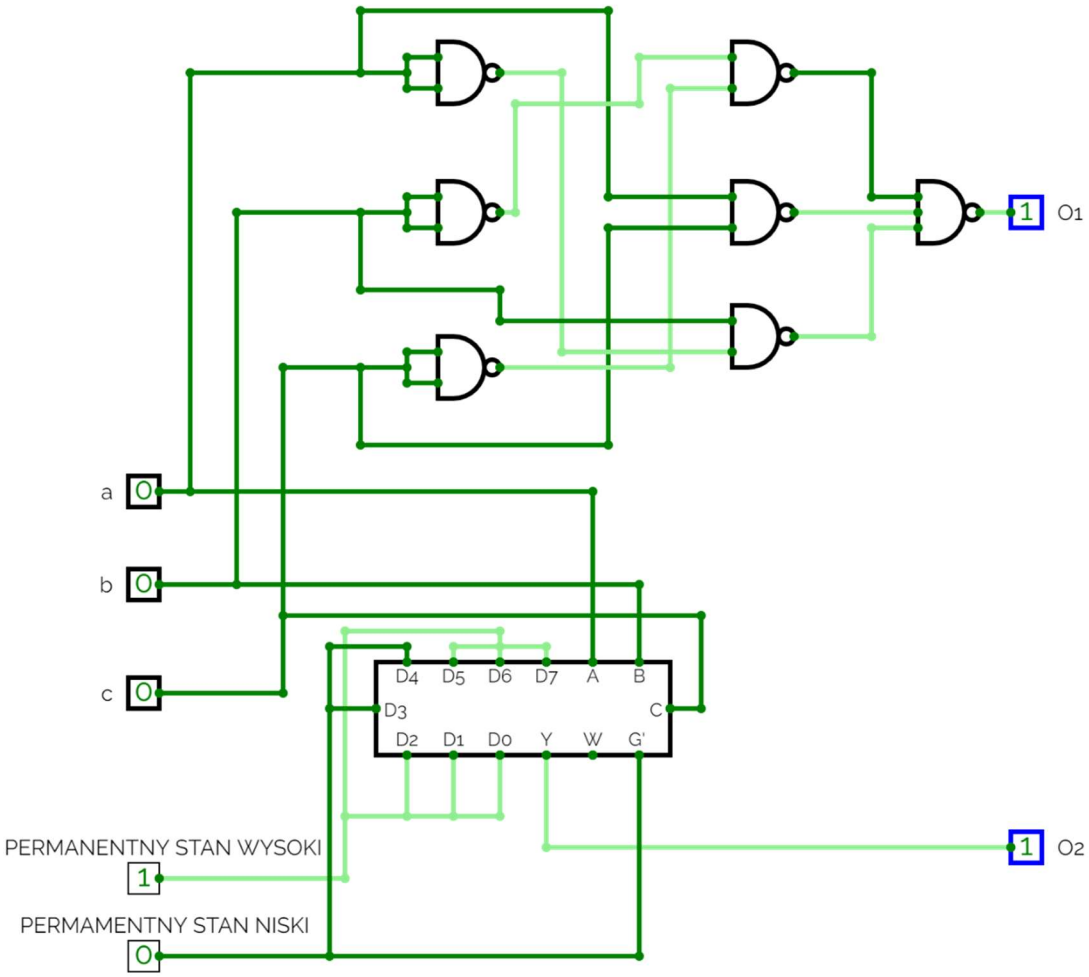
$$f(a,b,c) = \Sigma(0,1,2,5,6,7)$$

Taka funkcja może być przekazana do multipleksera bez modyfikacji.



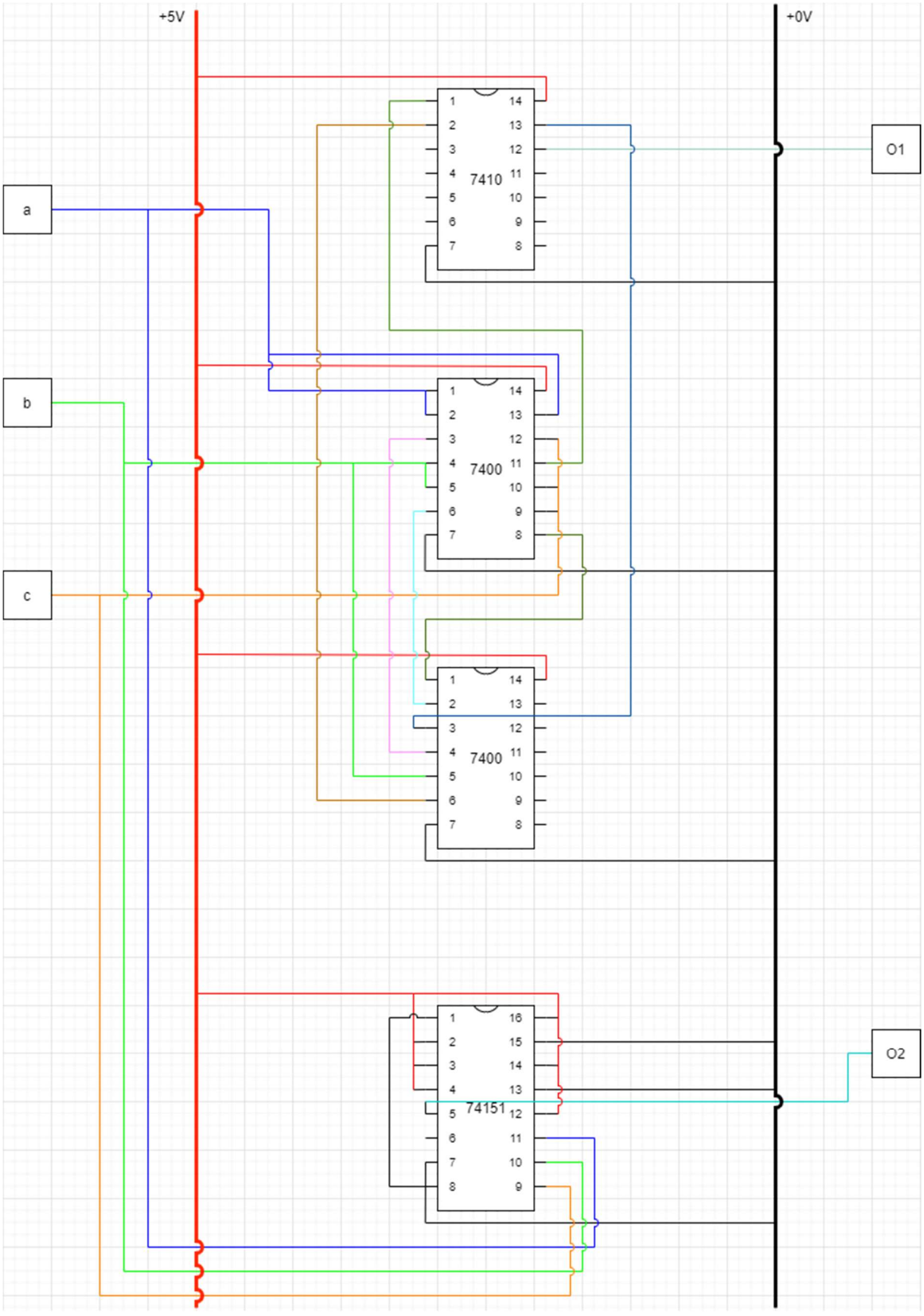
Rysunek 9. Schemat logiczny układu funkcji B za pomocą multipleksera.

16. Przedstawienie funkcji B za pomocą bramek NAND oraz multipleksera



Rysunek 10. Schemat logiczny układu funkcji B za pomocą bramek NAND oraz multipleksera.

17. Schemat podłączenia układu B



Rysunek 11. Schemat podłączenia układu realizującego funkcję B.

Tablica kroków działania dla powyższego schematu.

	WEJŚCIA			WYJŚCIA	
lp.	a	b	c	O1	O2
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	0	1	1
4	0	1	0	1	1
5	0	0	0	1	1
6	1	0	0	1	1
7	0	0	0	1	1
8	0	0	1	0	0
9	0	1	1	1	1
10	0	0	1	0	0
11	1	0	1	1	1
12	1	1	1	1	1
13	1	1	0	0	0
14	1	0	0	1	1
15	0	0	0	1	1

Tabela 12. Tabela kroków działania układu B

Korzystając z tego, że jest to układ kombinacyjny czyli stan na wyjściu zależy tylko od stanach na wejściu możemy uprościć powyższą tabelę do następującej postaci

WEJŚCIA			WYJŚCIA	
a	b	c	O1	O2
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Tabela 13. Tabela prawdy dla układu B.

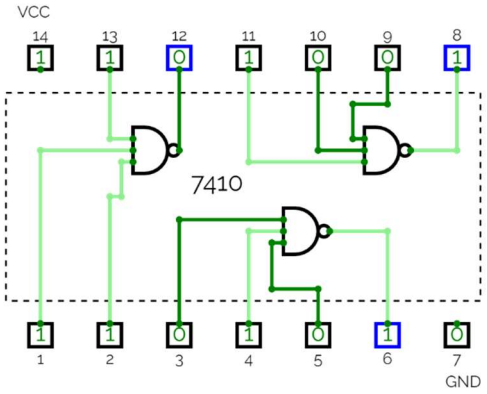
Możemy zauważyć, że tabela prawdy dla układu B jest zgodna z tabelą 8 – tabelą prawdy dla funkcji B. Dodatkowo na wyjściach O1 i O2 mamy te same stany co oznacza, że oba układy, jeden stworzony korzystając z bramek NAND, a drugi korzystający z multipleksera, są identyczne oraz realizują funkcję B.

18. Wnioski

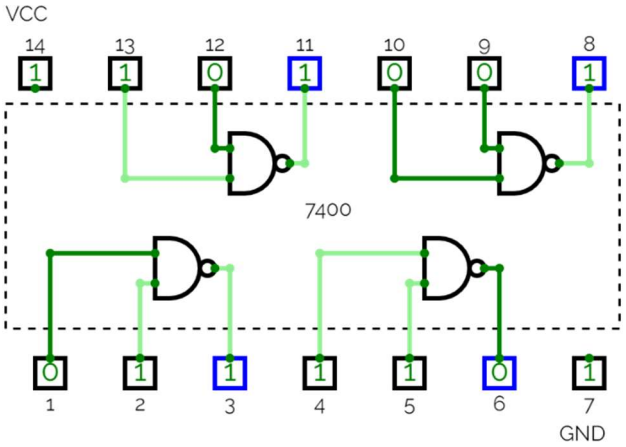
Naszym celem było zrealizować funkcje $f(a,b,c,d) = \overline{\overline{\overline{\overline{a+b}+\bar{c}}}+d}$ oraz $f(a,b,c) = \overline{ab\bar{c} + \bar{a}\bar{b}c}$ w minimalnym (dwupoziomowym) układzie wykorzystującym bramki NAND oraz układzie wykorzystującym multiplekser 74151 (MUX). Byliśmy w stanie to osiągnąć dzięki zastosowaniu tablicy Karnaugh, praw De Morgana oraz postaci kanonicznej sumy. Rezultatem są układy, których schematy podłączenia możemy zobaczyć na rysunku 7 (układ A) oraz rysunku 11 (układ B). Należy zauważyć, że zbudowane przez nas układy realizują podane funkcje. Nie ma też znaczenia czy użyjemy do konstrukcji układów bramek NAND czy multipleksera gdyż oba układy będą identyczne. Udowodniliśmy to porównując kolejno tabelę 1 (tabela prawdy dla funkcji A) z tabelą 7 (tabela prawdy dla zbudowanych układów realizujących funkcję A) i tabelę 8 (tabela prawdy dla funkcji B) z tabelą 13 (tabela prawdy dla układów realizujących funkcję B).

19. Spis użytych układów

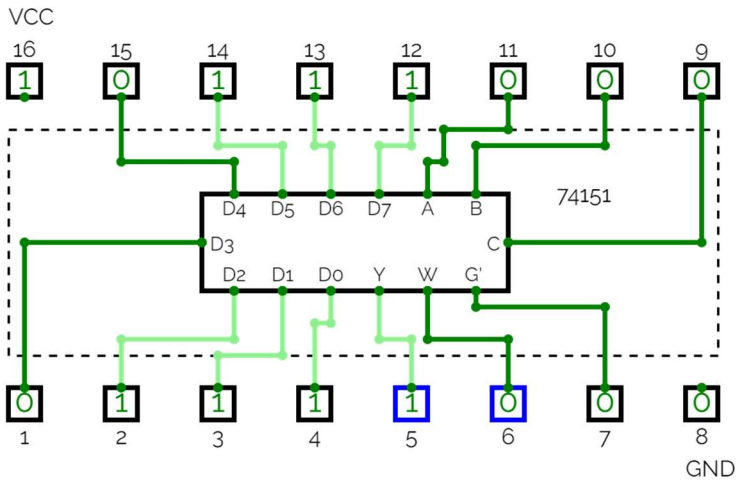
- Układ 7410 – układ zawierający trzy trójwejściowe bramki NAND



- Układ 7400 – układ zawierający 4 niezależne bramki NAND



- Układ 74151 – multiplekser



20. Bibliografia oraz link do układów

a) Schemat oraz specyfikacja układu 74151:
https://www.circuitbasics.com/what-is-a-multiplexer/
https://www.elprocus.com/what-is-multiplexer-and-demultiplexer-types-and-differences/

b) Układy:
https://circuitverse.org/users/71768/projects/sprawozdanie-2