
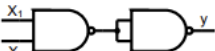
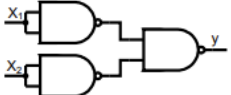

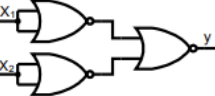
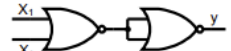


<h1>Sprawozdanie z układów logicznych</h1>	
	Ćwiczenie nr: 2
	Temat ćwiczenia: Układy kombinacyjne
1. Imię i nazwisko – student 1: Wojciech Krzos	
2. Imię i nazwisko – student 2: Natalia Marszałek	
Grupa laboratoryjna nr (u prowadzącego): 5	Dzień tygodnia: Czwartek
Płyta montażowa nr (z tyłu zadajnika): NA	Godziny zajęć (od-do): 13:15 – 15:00

# 1 BODY

## 1.1 METODA ZMIANY UKŁADU AND-OR NA NAND

Możliwie najprostszą i najszybszą metodą jest traktowanie pojedynczych bramek jako bloków bramek NAND według poniższego schematu:

Funkcja logiczna	NOT	AND	OR
Zapis	$y = \overline{x_1}$	$y = x_1 \cdot x_2$	$y = x_1 + x_2$
NAND			
NOR			

## 1.2 FUNKCJA 1

### 1.2.1 Wzór i przekształcenie funkcji

$$f(a, b, c, d) = \overline{\overline{(a + b + c)} + d} = \overline{\overline{cd} + \overline{abd}} = (\neg a \uparrow \neg b \uparrow \neg d) \uparrow (\neg c \uparrow \neg d)$$

### 1.2.2 Tablica prawdy

a	b	c	d	f(a, b, c, d)
0	0	0	0	T
0	0	0	1	F
0	0	1	0	T
0	0	1	1	F
0	1	0	0	T
0	1	0	1	F
0	1	1	0	F
0	1	1	1	F
1	0	0	0	T
1	0	0	1	F
1	0	1	0	F
1	0	1	1	F
1	1	0	0	T
1	1	0	1	F
1	1	1	0	F
1	1	1	1	F

### 1.2.3 Mapa Karnaugh'a

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

### 1.2.4 Analiza względem multiplexera

BCD	000	001	010	011	100	101	110	111
A	A	B	C	D	E	F	G	H
0	1	0	1	0	1	0	0	0
1	1	0	0	0	1	0	0	0
	1	0	A'	0	1	0	0	0

### 1.2.5 Schematy powyższych układów

Schematy powyższych układów znajdują się na końcu pracy (patrz 3.1 Schematy funkcji 1)

## 1.3 FUNKCJA 2

### 1.3.1 Wzór i przekształcenie funkcji

$$f(a, b, c) = abc + abc$$

### 1.3.2 Tablica prawdy

a	b	c	$abc$	$abc + abc$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

### 1.3.3 Analiza względem multipleksera:

	WEJŚCIE ADRESOWE A	WEJŚCIE ADRESOWE B	WEJŚCIE ADRESOWE C	OUT
000	1	1	1	0
001	1	1	1	0
010	1	1	1	0
011	1	1	1	0
100	1	1	1	0
101	1	1	1	0
110	1	1	1	0
111	1	1	1	1

### 1.3.4 Mapa Karnaugh'a

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

Funkcję można zminimalizować do postaci:  $f(a, b, c) = abc$

### 1.3.5 Minimalizacja funkcji 2

Zapis funkcji z użyciem wyłącznie nand:

$$a \text{ and } b \text{ and } c = [(a \text{ nand } a) \text{ nand } (b \text{ nand } b)] \text{ nand } (c \text{ nand } c)$$

## 2 KONKLUZJE

Multiplexery są jednymi z najbardziej użytecznych narzędzi w dziedzinie elektroniki cyfrowej, umożliwiając implementację funkcji za pomocą łatwych i wydajnych układów bramek logicznych. Są to układy scalone, które mają kilka wejść danych i jedno wyjście, a ich działanie polega na wybieraniu jednego z wejść, które zostaje przekazane na wyjście.

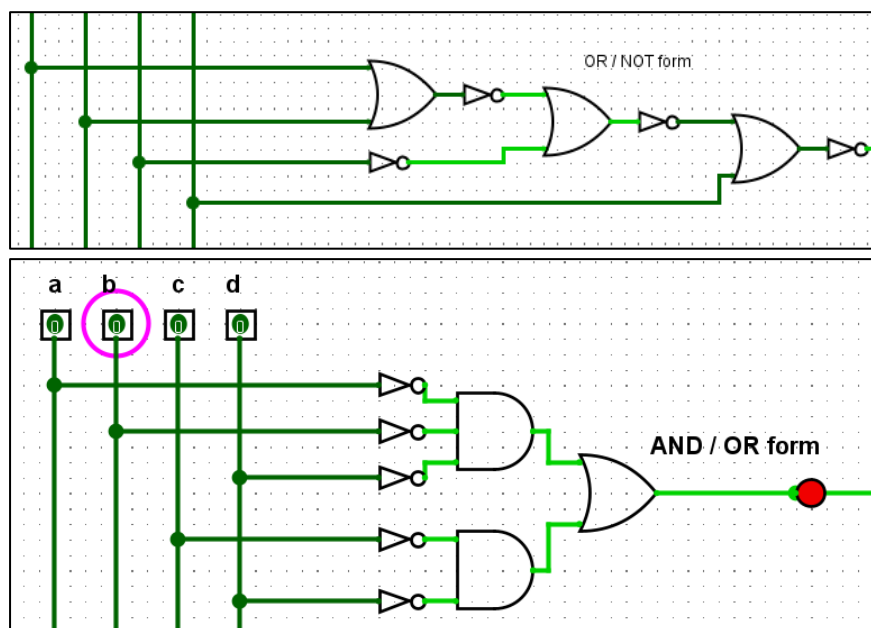
Multiplexery są szczególnie przydatne w systemach cyfrowych, w których wymagana jest zmiana kierunku przepływu danych w zależności od sygnałów sterujących. Mogą być również wykorzystane do selekcji sygnałów wejściowych lub do przekazywania danych do różnych komponentów w zależności od ich adresu.

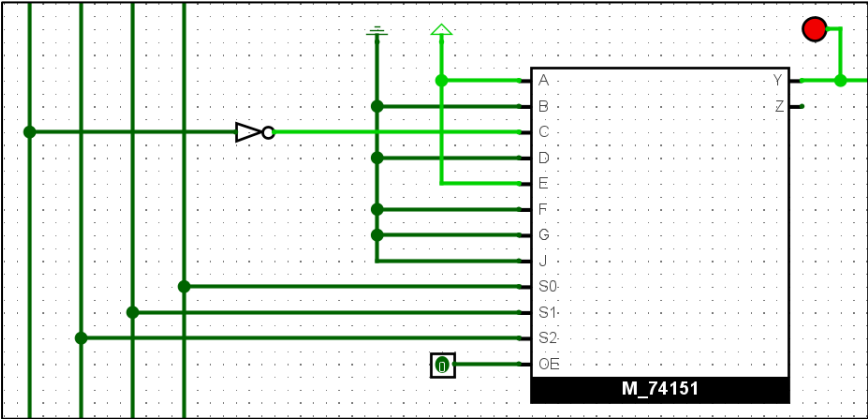
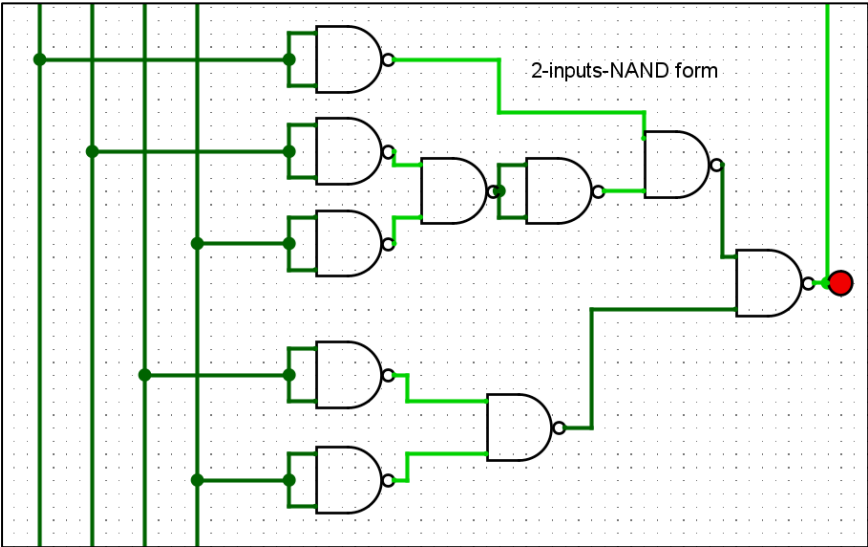
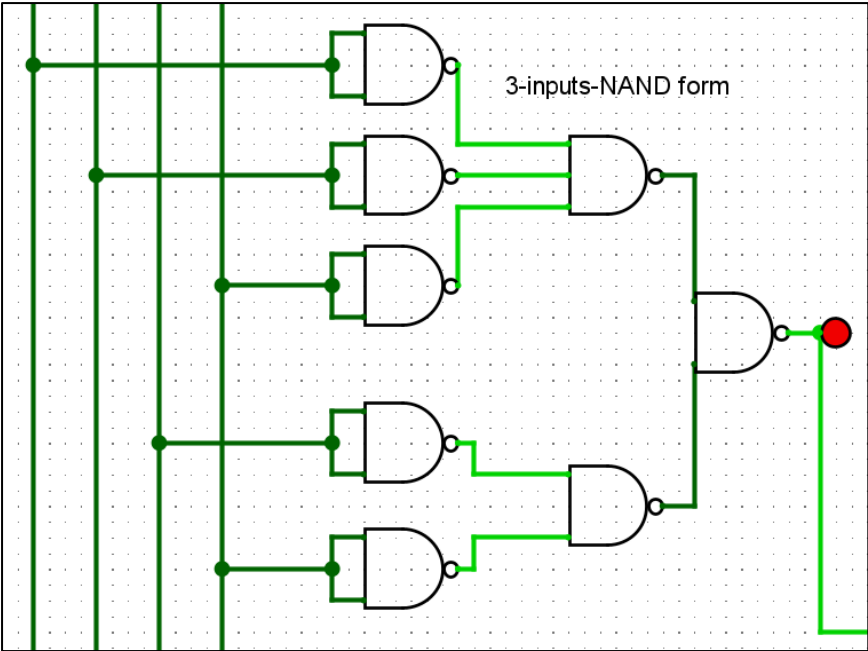
Mapy Karnaugh są również bardzo przydatnym narzędziem w dziedzinie elektroniki cyfrowej. Pozwalają na minimalizację funkcji logicznych za pomocą graficznej reprezentacji ich tablic prawdy. Dzięki temu możliwe jest uzyskanie bardziej zoptymalizowanych układów, które wymagają mniejszej liczby bramek logicznych.

Kolejnym interesującym narzędziem w dziedzinie elektroniki cyfrowej jest wykorzystanie bramek NAND do implementacji dowolnych funkcji logicznych, składających się z bramek OR lub AND. Brama NAND jest połączeniem dwóch bramek NOT i jednej bramki AND. Pozwala to na zaimplementowanie dowolnej funkcji logicznej za pomocą tylko bramek NAND.

## 3 APPENDIX

### 3.1 SCHEMATY FUNKCJI 1





## 3.2 SCHEMATY FUNKCJI 2

