Maciej Byczko Bartosz Matysiak	Prowadzący: dr inż. Jacek Mazurkiewicz	Numer ćwiczenia 2
PN 10:50 TP	Temat ćwiczenia: Układy Kombinacyjne	Ocena:
Grupa: B	Data wykonania: 10 Października 2021	

Spis treści

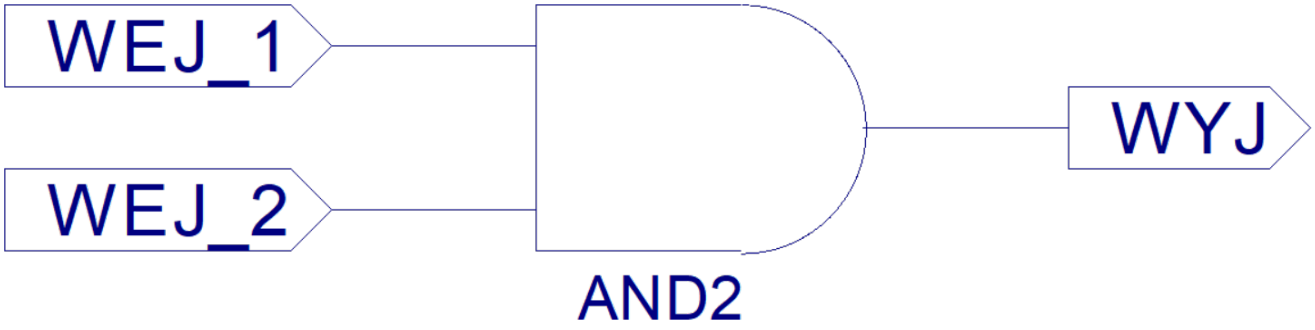
1 Zadanie 1

1.1 Polecenie

Wykonać dowolną bramkę - funktor: 2 wejścia, 1 wyjście

1.2 Rozwiązanie

1.2.1 Schemat układu

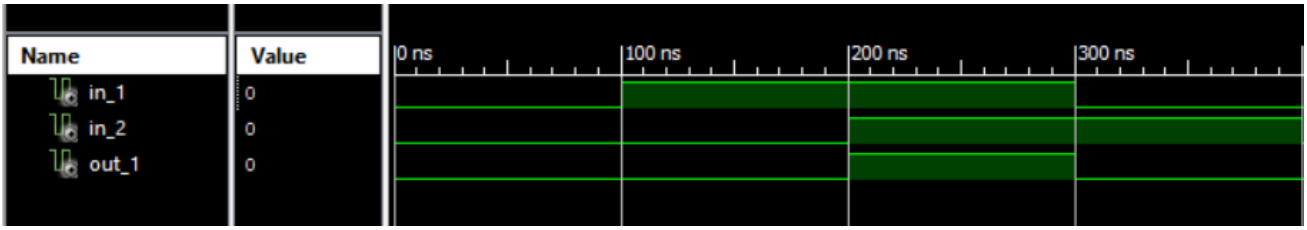


1.2.2 Kod VHDL

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4  LIBRARY UNISIM;
5  USE UNISIM.Vcomponents.ALL;
6  ENTITY schematic_zad1_schematic_zad1_sch_tb IS
7  END schematic_zad1_schematic_zad1_sch_tb;
8  ARCHITECTURE behavioral OF schematic_zad1_schematic_zad1_sch_tb IS
9
10    COMPONENT schematic_zad1
11    PORT(   IN_1   : IN   STD_LOGIC;
12            IN_2   : IN   STD_LOGIC;
13            OUT_1:   OUT STD_LOGIC);
14    END COMPONENT;
15
16    SIGNAL IN_1   : STD_LOGIC;
17    SIGNAL IN_2   : STD_LOGIC;
18    SIGNAL OUT_1:   STD_LOGIC;
19
20 BEGIN
21
22    UUT: schematic_zad1 PORT MAP(
23      IN_1 => IN_1,
24      IN_2 => IN_2,
25      OUT_1 => OUT_1
26    );
27
28 IN_1 <= '0', '1' after 100 ns, '0' after 300 ns;
29 IN_2 <= '0', '1' after 200 ns, '0' after 400 ns;
30
```

```
31 END;
```

1.2.3 Symulacja



2 Zadanie 2

2.1 Polecenie

Implementacja funkcji logicznej $G(w,x,y,z) = \prod(0,2,3,4,6,7,9,11,12,13,15)$

2.2 Rozwiązanie

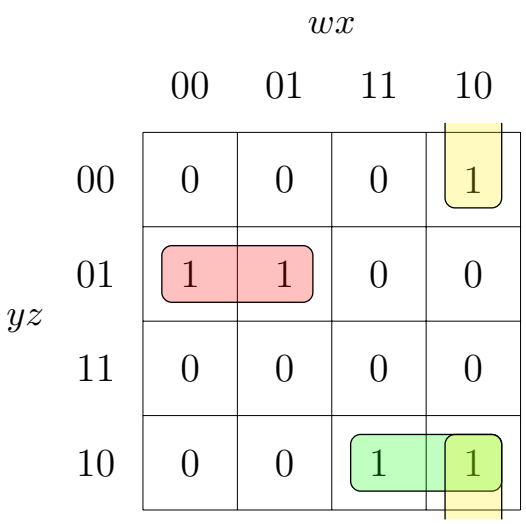
2.2.1 Wyprowadzenie

$$G(w,x,y,z) = \prod(0,2,3,4,6,7,9,11,12,13,15) \tag{1}$$
$$= \sum(1,5,8,10,14) = \sum(0001,0101,1000,1010,1110) \tag{2}$$
$$= \overline{w}x\overline{y}z + \overline{w}x\overline{y}z + w\overline{x}\overline{y}\overline{z} + w\overline{x}y\overline{z} + wxy\overline{z} \tag{3}$$
$$= \overline{w}\overline{y}z(\overline{x}+x) + w\overline{z}(\overline{x}\overline{y} + \overline{x}y + xy) \tag{4}$$
$$= \overline{w}\overline{y}z + w\overline{z}(\overline{x}(\overline{y}+y) + xy) \tag{5}$$
$$= \overline{w}\overline{y}z + w\overline{z}(\overline{x}+xy) \tag{6}$$
$$= \overline{w}\overline{y}z + w\overline{z}((\overline{x}+x)(\overline{x}+y)) \tag{7}$$
$$= \overline{w}\overline{y}z + w\overline{z}((\overline{x}+y)) \tag{8}$$
$$= \overline{w}\overline{y}z + w\overline{x}\overline{z} + w\overline{z}y \tag{9}$$

2.2.2 Tabela prawdy

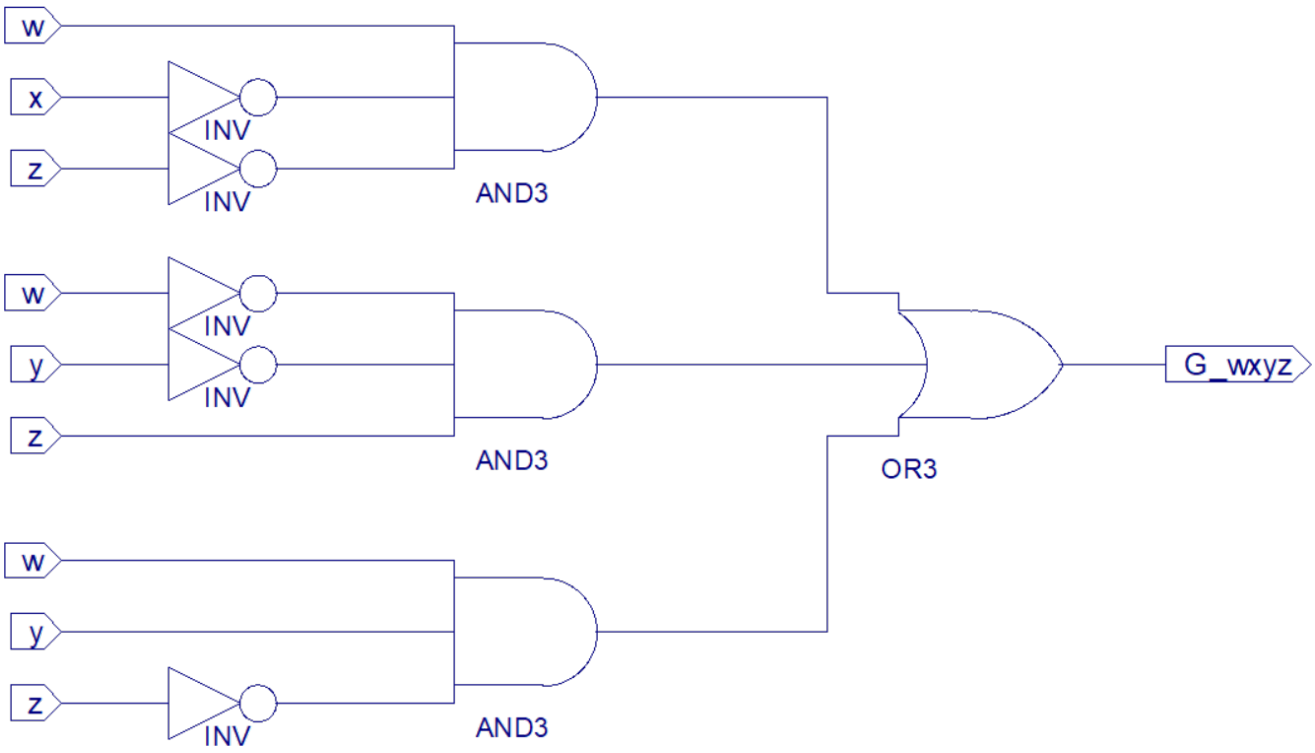
Kod dziesiętny	w	x	y	z	G
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	0

2.2.3 Siatka Karnaugh



Rysunek 1: $G_{wxyz} = w\overline{xz} + \overline{wy}z + wy\overline{z}$

2.2.4 Schemat układu

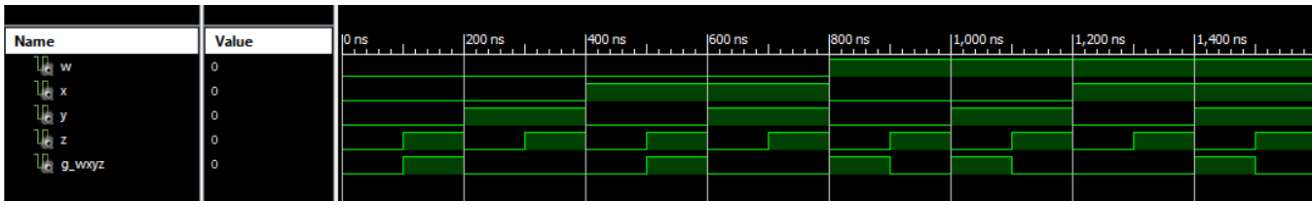


2.2.5 Kod VHDL

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4  LIBRARY UNISIM;
5  USE UNISIM.Vcomponents.ALL;
6  ENTITY schematic_zad2_schematic_zad2_sch_tb IS
7  END schematic_zad2_schematic_zad2_sch_tb;
8  ARCHITECTURE behavioral OF schematic_zad2_schematic_zad2_sch_tb IS
9
10     COMPONENT schematic_zad2
11     PORT(   w : IN   STD_LOGIC;
12             x : IN   STD_LOGIC;
13             y : IN   STD_LOGIC;
14          z   : IN   STD_LOGIC;
15             G_wxyz   : OUT STD_LOGIC);
16     END COMPONENT;
17
18     SIGNAL w : STD_LOGIC := '0';
19     SIGNAL x : STD_LOGIC := '0';
20     SIGNAL y : STD_LOGIC := '0';
21    SIGNAL z   : STD_LOGIC := '0';
22     SIGNAL G_wxyz   : STD_LOGIC;
23
24 BEGIN
25
26     UUT: schematic_zad2 PORT MAP(
27       w => w,
```

```
28      x => x,
29      y => y,
30      z => z,
31      G_wxyz => G_wxyz
32     );
33
34     w <= not w after 800ns;
35     x <= not x after 400ns;
36     y <= not y after 200ns;
37    z <= not z after 100ns;
38  END;
```

2.2.6 Symulacja



3 Zadanie 3

3.1 Polecenie

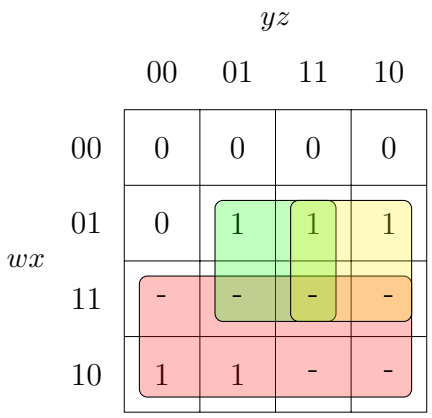
Implementacja układu translatora kodu **4-bit kod NKB na 4-bit kod Aikena**

3.2 Rozwiązanie

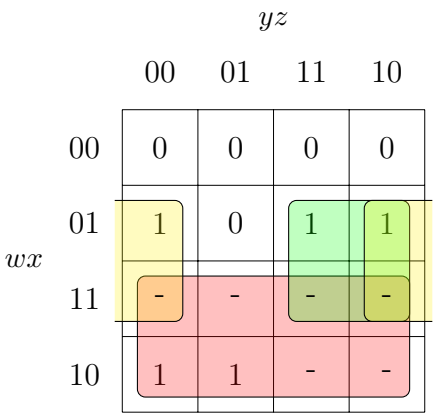
3.2.1 Tabela Prawdy

Kod dziesiętny	NKB				Kod Aikena			
	w	x	y	z	w	x	y	z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	1	0	0
7	0	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1
10	1	0	1	0	-	-	-	-
11	1	0	1	1	-	-	-	-
12	1	1	0	0	-	-	-	-
13	1	1	0	1	-	-	-	-
14	1	1	1	0	-	-	-	-
15	1	1	1	1	-	-	-	-

3.2.2 Siatki Karnaugh



$$w_A = xz + xy + w$$



$$x_A = x\overline{z} + xy + w$$

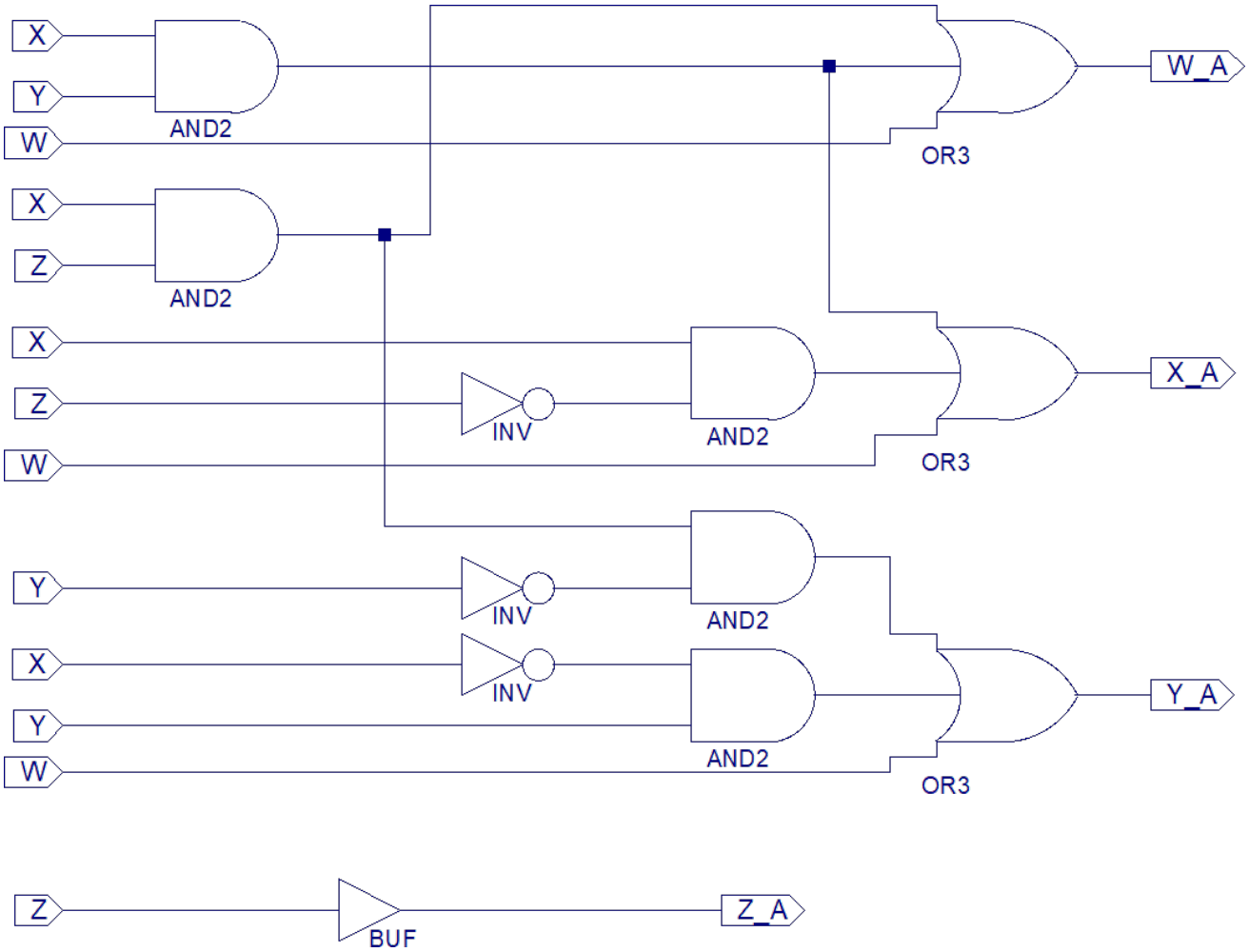
wx \ yz	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	-	-	-	-
10	1	1	-	-

$$y_A = \overline{x}y + x\overline{y}z + w$$

wx \ yz	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	-	-	-	-
10	0	1	-	-

$$z_A = z$$

3.2.3 Schemat układu

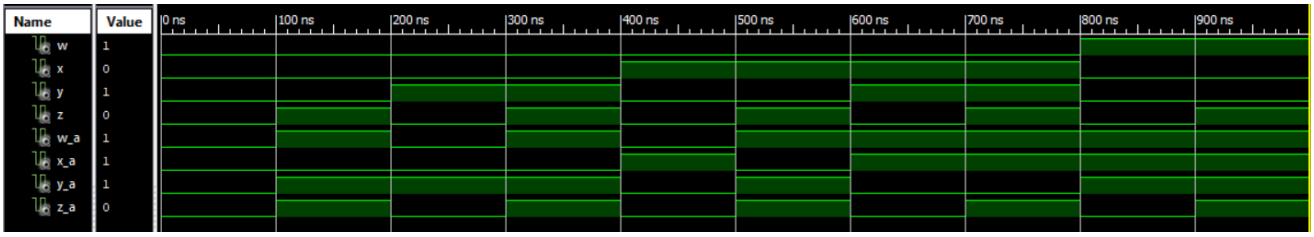


3.2.4 Kod VHDL

```
1 LIBRARY ieee;
2 USE ieee.std_logic_1164.ALL;
3 USE ieee.numeric_std.ALL;
4 LIBRARY UNISIM;
5 USE UNISIM.Vcomponents.ALL;
```

```
6  ENTITY schematic_zad3_schematic_zad3_sch_tb IS
7  END schematic_zad3_schematic_zad3_sch_tb;
8  ARCHITECTURE behavioral OF schematic_zad3_schematic_zad3_sch_tb IS
9
10     COMPONENT schematic_zad3
11     PORT(
12        W : IN  STD_LOGIC;
13            Y   : IN  STD_LOGIC;
14            X   : IN  STD_LOGIC;
15            Z   : IN  STD_LOGIC;
16            Y_A   : OUT STD_LOGIC;
17            X_A   : OUT STD_LOGIC;
18            W_A   : OUT STD_LOGIC;
19            Z_A   : OUT STD_LOGIC
20          );
21     END COMPONENT;
22
23     SIGNAL W : STD_LOGIC := '0';
24     SIGNAL Y : STD_LOGIC := '0';
25     SIGNAL X : STD_LOGIC := '0';
26     SIGNAL Z : STD_LOGIC := '0';
27     SIGNAL Y_A : STD_LOGIC;
28     SIGNAL X_A : STD_LOGIC;
29     SIGNAL W_A : STD_LOGIC;
30     SIGNAL Z_A : STD_LOGIC;
31
32 BEGIN
33
34     UUT: schematic_zad3 PORT MAP(
35      W => W,
36      Y => Y,
37      X => X,
38      Z => Z,
39      Y_A => Y_A,
40      X_A => X_A,
41      W_A => W_A,
42      Z_A => Z_A
43     );
44      W <= not W after 800ns;
45      X <= not X after 400ns;
46      Y <= not Y after 200ns;
47     Z <= not Z after 100ns;
48 END;
```

3.2.5 Symulacja



4 Wnioski