

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

<b>1</b>	<b>Zadanie 1</b>	<b>2</b>
1.1	Polecenie . . . . .	2
1.2	Rozwiązanie . . . . .	2
1.2.1	Schemat układu . . . . .	2
1.2.2	Kod VHDL . . . . .	2
1.2.3	Symulacja . . . . .	3
<b>2</b>	<b>Zadanie 2</b>	<b>3</b>
2.1	Polecenie . . . . .	3
2.2	Rozwiązanie . . . . .	3
2.2.1	Wyprowadzenie . . . . .	3
2.2.2	Tabela prawdy . . . . .	4
2.2.3	Siatka Karnaugh . . . . .	4
2.2.4	Schemat układu . . . . .	5
2.2.5	Kod VHDL . . . . .	5
2.2.6	Symulacja . . . . .	6
<b>3</b>	<b>Zadanie 3</b>	<b>6</b>
3.1	Polecenie . . . . .	6
3.2	Rozwiązanie . . . . .	7
3.2.1	Tabela Prawdy . . . . .	7
3.2.2	Siatki Karnaugh . . . . .	7
3.2.3	Schemat układu . . . . .	8
3.2.4	Kod VHDL . . . . .	8
3.2.5	Symulacja . . . . .	10
<b>4</b>	<b>Wnioski</b>	<b>10</b>

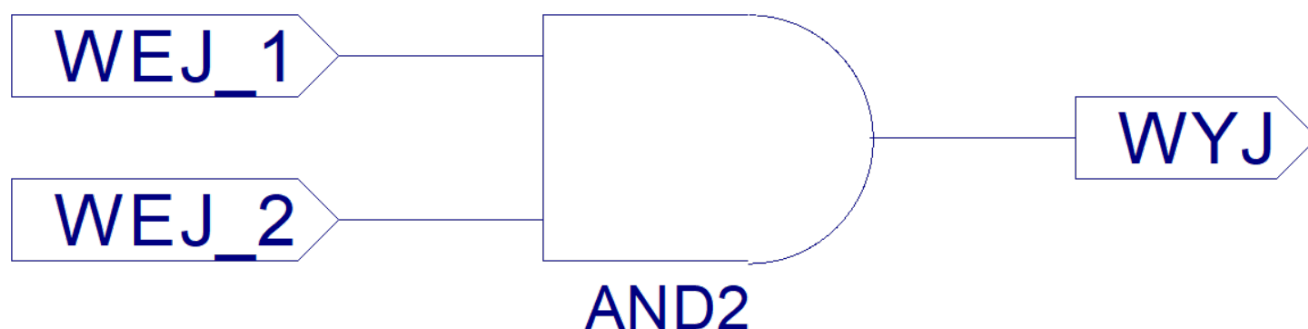
# 1 Zadanie 1

## 1.1 Polecenie

Wykonać dowolną bramkę - funktor: 2 wejścia, 1 wyjście

## 1.2 Rozwiązanie

### 1.2.1 Schemat układu

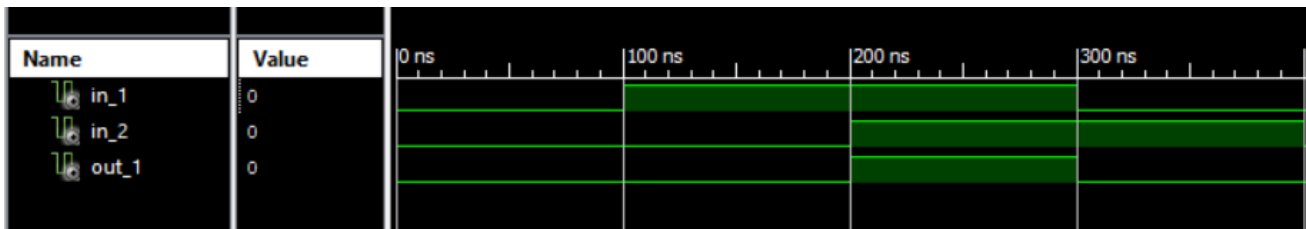


### 1.2.2 Kod VHDL

```
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

$$G(w, x, y, z) = \prod(0, 2, 3, 4, 6, 7, 9, 11, 12, 13, 15) \quad (1)$$

$$= \sum(1, 5, 8, 10, 14) = \sum(0001, 0101, 1000, 1010, 1110) \quad (2)$$

$$= \overline{w}x\overline{y}z + \overline{w}x\overline{y}z + w\overline{x}y\overline{z} + w\overline{x}y\overline{z} + wxy\overline{z} \quad (3)$$

$$= \overline{w}\overline{y}z(\overline{x} + x) + w\overline{z}(\overline{x}\overline{y} + \overline{x}y + xy) \quad (4)$$

$$= \overline{w}\overline{y}z + w\overline{z}(\overline{x}(\overline{y} + y) + xy) \quad (5)$$

$$= \overline{w}\overline{y}z + w\overline{z}(\overline{x} + xy) \quad (6)$$

$$= \overline{w}\overline{y}z + w\overline{z}((\overline{x} + x)(\overline{x} + y)) \quad (7)$$

$$= \overline{w}\overline{y}z + w\overline{z}((\overline{x} + y)) \quad (8)$$

$$= \overline{w}\overline{y}z + w\overline{x}\overline{z} + w\overline{z}y \quad (9)$$

2.2.2 Tabela prawdy

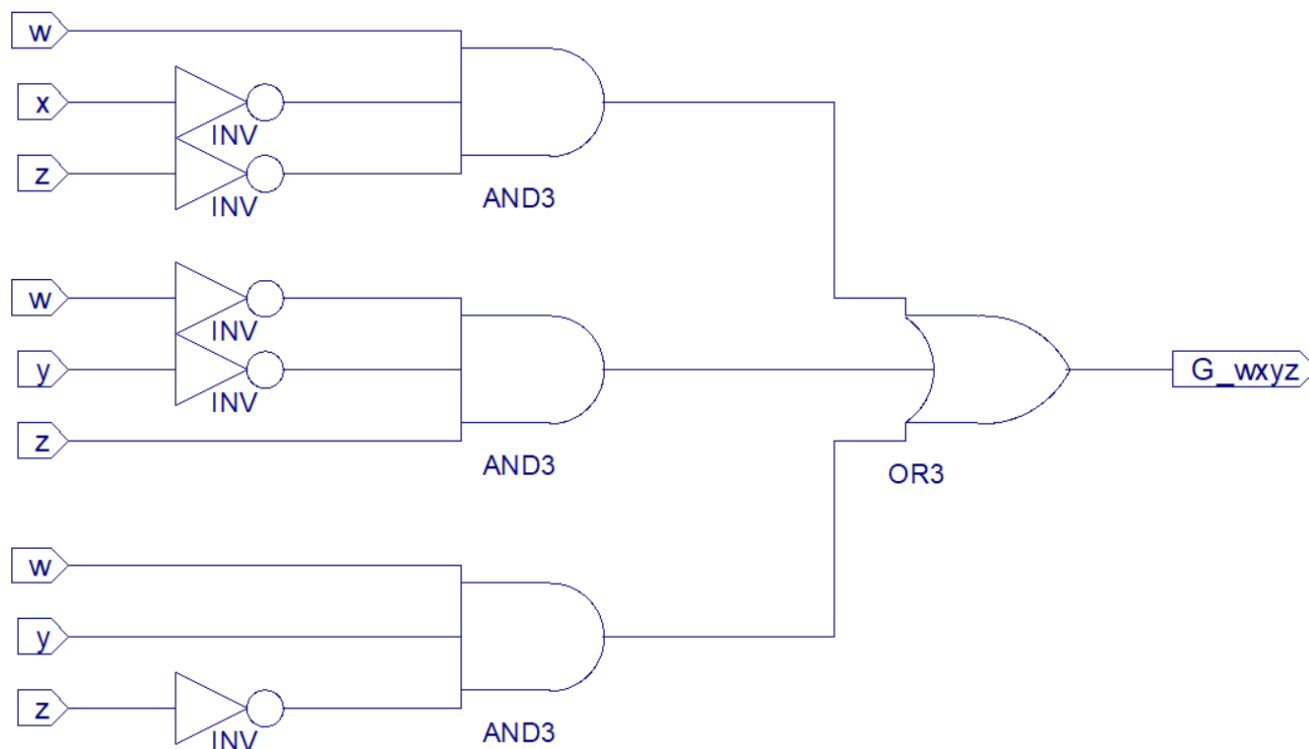
Kod dziesiętny	w	x	y	z	G
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	0

2.2.3 Siatka Karnaugh

		<i>w</i> <i>x</i>			
		00	01	11	10
<i>y</i> <i>z</i>	00	0	0	0	1
	01	1	1	0	0
	11	0	0	0	0
	10	0	0	1	1

Rysunek 1:  $G_{wxyz} = w\bar{x}\bar{z} + \bar{w}yz + wy\bar{z}$

### 2.2.4 Schemat układu



### 2.2.5 Kod VHDL

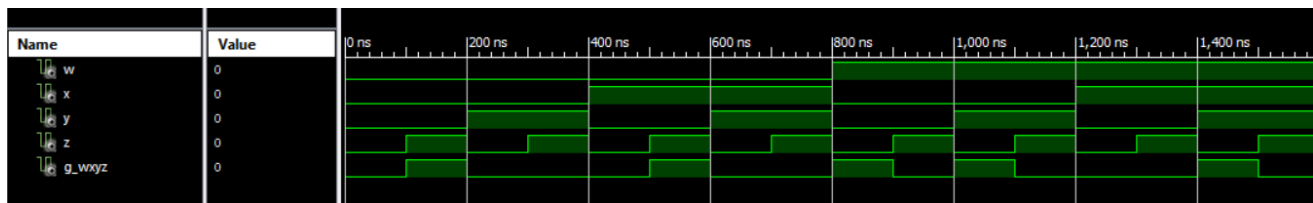
```

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(
12         w : IN  STD_LOGIC;
13         x : IN  STD_LOGIC;
14         y : IN  STD_LOGIC;
15         z : IN  STD_LOGIC;
16         G_wxyz : OUT STD_LOGIC);
17     END COMPONENT;
18
19     SIGNAL w : STD_LOGIC := '0';
20     SIGNAL x : STD_LOGIC := '0';
21     SIGNAL y : STD_LOGIC := '0';
22     SIGNAL z : STD_LOGIC := '0';
23     SIGNAL G_wxyz : STD_LOGIC;
24
25 BEGIN
26
27     UUT: schematic_zad2 PORT MAP(
28         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

		<i>yz</i>			
		00	01	11	10
<i>wx</i>	00	0	0	0	0
	01	0	1	1	1
	11	-	-	-	-
	10	1	1	-	-

$w_A = xz + xy + w$

		<i>yz</i>			
		00	01	11	10
<i>wx</i>	00	0	0	0	0
	01	1	0	1	1
	11	-	-	-	-
	10	1	1	-	-

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

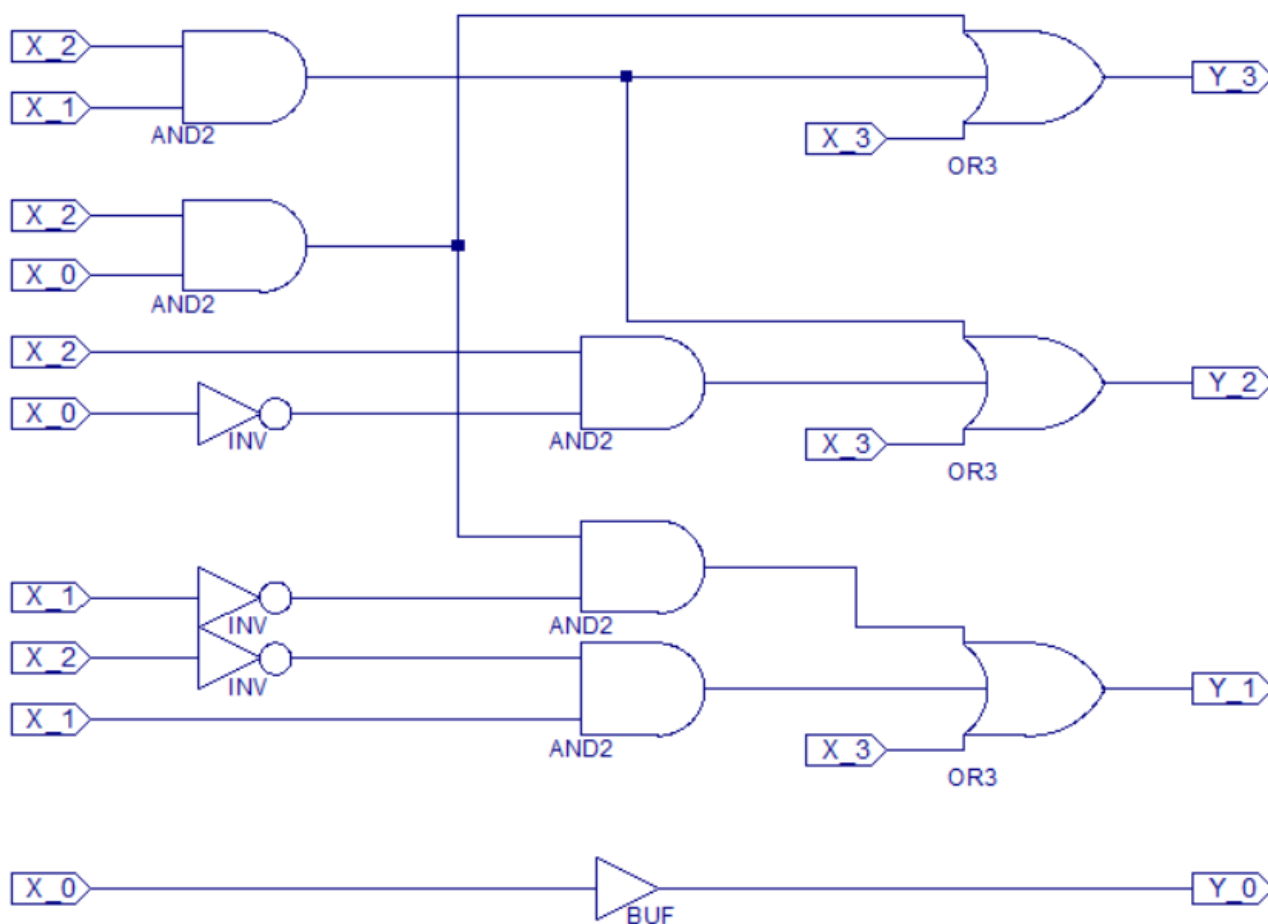
		$yz$			
		00	01	11	10
$wx$	00	0	0	1	1
	01	0	1	0	0
	11	-	-	-	-
	10	1	1	-	-

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

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

$$z_A = z$$

### 3.2.3 Schemat układu



### 3.2.4 Kod VHDL

```

1  -- Vhdl test bench created from schematic C:\Users\Qucker135\
    Documents\ISE_DS_projects\NKB_to_Aiken\NKB_to_Aiken_schematic.sch
    -- Wed Oct 06 22:49:49 2021
2  --
3  -- Notes:

```



```

4 — 1) This testbench template has been automatically generated using
   types
5 — std_logic and std_logic_vector for the ports of the unit under
   test.
6 — Xilinx recommends that these types always be used for the top-
   level
7 — I/O of a design in order to guarantee that the testbench will
   bind
8 — correctly to the timing (post-route) simulation model.
9 — 2) To use this template as your testbench, change the filename to
   any
10 — name of your choice with the extension .vhd, and use the "Source
    ->Add"
11 — menu in Project Navigator to import the testbench. Then
12 — edit the user defined section below, adding code to generate the
13 — stimulus for your design.
14 —
15 LIBRARY ieee;
16 USE ieee.std_logic_1164.ALL;
17 USE ieee.numeric_std.ALL;
18 LIBRARY UNISIM;
19 USE UNISIM.Vcomponents.ALL;
20 ENTITY NKB_to_Aiken_schematic_NKB_to_Aiken_schematic_sch_tb IS
21 END NKB_to_Aiken_schematic_NKB_to_Aiken_schematic_sch_tb;
22 ARCHITECTURE behavioral OF
23     NKB_to_Aiken_schematic_NKB_to_Aiken_schematic_sch_tb IS
24
25     COMPONENT NKB_to_Aiken_schematic
26     PORT( X_3 : IN STD_LOGIC;
27           X_2 : IN STD_LOGIC;
28           X_1 : IN STD_LOGIC;
29           X_0 : IN STD_LOGIC;
30           Y_3 : OUT STD_LOGIC;
31           Y_2 : OUT STD_LOGIC;
32           Y_1 : OUT STD_LOGIC;
33           Y_0 : OUT STD_LOGIC);
34     END COMPONENT;
35
36     SIGNAL X_3 : STD_LOGIC := '0';
37     SIGNAL X_2 : STD_LOGIC := '0';
38     SIGNAL X_1 : STD_LOGIC := '0';
39     SIGNAL X_0 : STD_LOGIC := '0';
40     SIGNAL Y_3 : STD_LOGIC;
41     SIGNAL Y_2 : STD_LOGIC;
42     SIGNAL Y_1 : STD_LOGIC;
43     SIGNAL Y_0 : STD_LOGIC;
44
45 BEGIN
46
47     UUT: NKB_to_Aiken_schematic PORT MAP(
48         X_3 => X_3,

```

```
49     X_2 => X_2,  
50     X_1 => X_1,  
51     X_0 => X_0,  
52     Y_3 => Y_3,  
53     Y_2 => Y_2,  
54     Y_1 => Y_1,  
55     Y_0 => Y_0  
56 );  
57  
58     X_3 <= not X_3 after 800 ns;  
59     X_2 <= not X_2 after 400 ns;  
60     X_1 <= not X_1 after 200 ns;  
61     X_0 <= not X_0 after 100 ns;  
62  
63 END;
```

3.2.5 Symulacja

