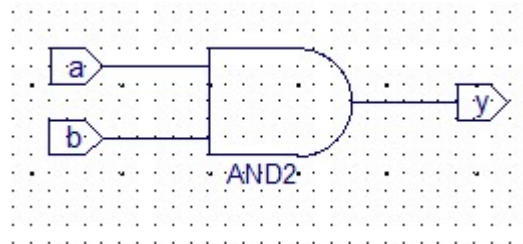


Termin zajęć Wtorek NP 7:30 – 11:00	Układy cyfrowe i systemy wbudowane	
Osoby wykonujące ćwiczenie: Jakub Suski 264028, Adam Czekalski 264488		Grupa: D
Tytuł ćwiczenia: Układy kombinacyjne		Laboratorium nr: 2
Data wykonania ćwiczenia	10.10.2023	Ocena:
Data oddania sprawozdania	24.10.2023	

Na pierwszych zajęciach laboratoryjnych zapoznano się ze środowiskiem Xilinx. Nauczono się tworzyć schematy, symulować projekt i implementować go na płytce CPLD ZL-9572.

1. Dowolna bramka – funktor – 2 wyjścia, 1 wejście

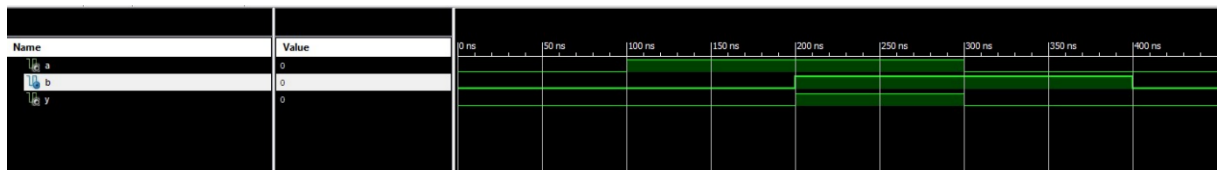
Pierwszy skonstruowany układ był zasadniczo prosty. Składał się z 2 wejść, bramki AND oraz wyjścia.



Następnie wzbogacono poniższy kod języka VHDL o pobudzenia testowe:

```
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 schel_schel_sch_tb IS
21  END schel_schel_sch_tb;
22  ARCHITECTURE behavioral OF schel_schel_sch_tb IS
23
24      COMPONENT schel
25      PORT( a : IN STD_LOGIC;
26            b : IN STD_LOGIC;
27            y : OUT STD_LOGIC);
28      END COMPONENT;
29
30      SIGNAL a : STD_LOGIC;
31      SIGNAL b : STD_LOGIC;
32      SIGNAL y : STD_LOGIC;
33
34      BEGIN
35
36      UUT: schel PORT MAP(
37          a => a,
38          b => b,
39          y => y
40      );
41
42      -- *** Test Bench - User Defined Section ***
43      a <= '0', '1' after 100 ns, '0' after 300 ns;
44      b <= '0', '1' after 200 ns, '0' after 400 ns;
45      -- *** End Test Bench - User Defined Section ***
46
47  END;
48
```

Kolejnym krokiem było uruchomienie symulatora ISIM:



Jak widać na powyższej ilustracji, na wyjściu otrzymano „1” tylko gdy na oba wejścia podano „1”. Układ działał poprawnie, w związku z tym, można było zaimplementować projekt na płycie CPLD.

Upřednio dołączono plik z rozszerzeniem *.ucf* do projektu i przypisano w nim wejścia i wyjścia do wyprowadzeń.

Dioda przestawała świecić gdy na wejściach podano dwie „1”, w przeciwnym wypadku świeciła.

2. Implementacja funkcji logicznej G

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

$$G(w, x, y, z) = \sum(1,5,8,10,14)$$

w	X	y	Z	G
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

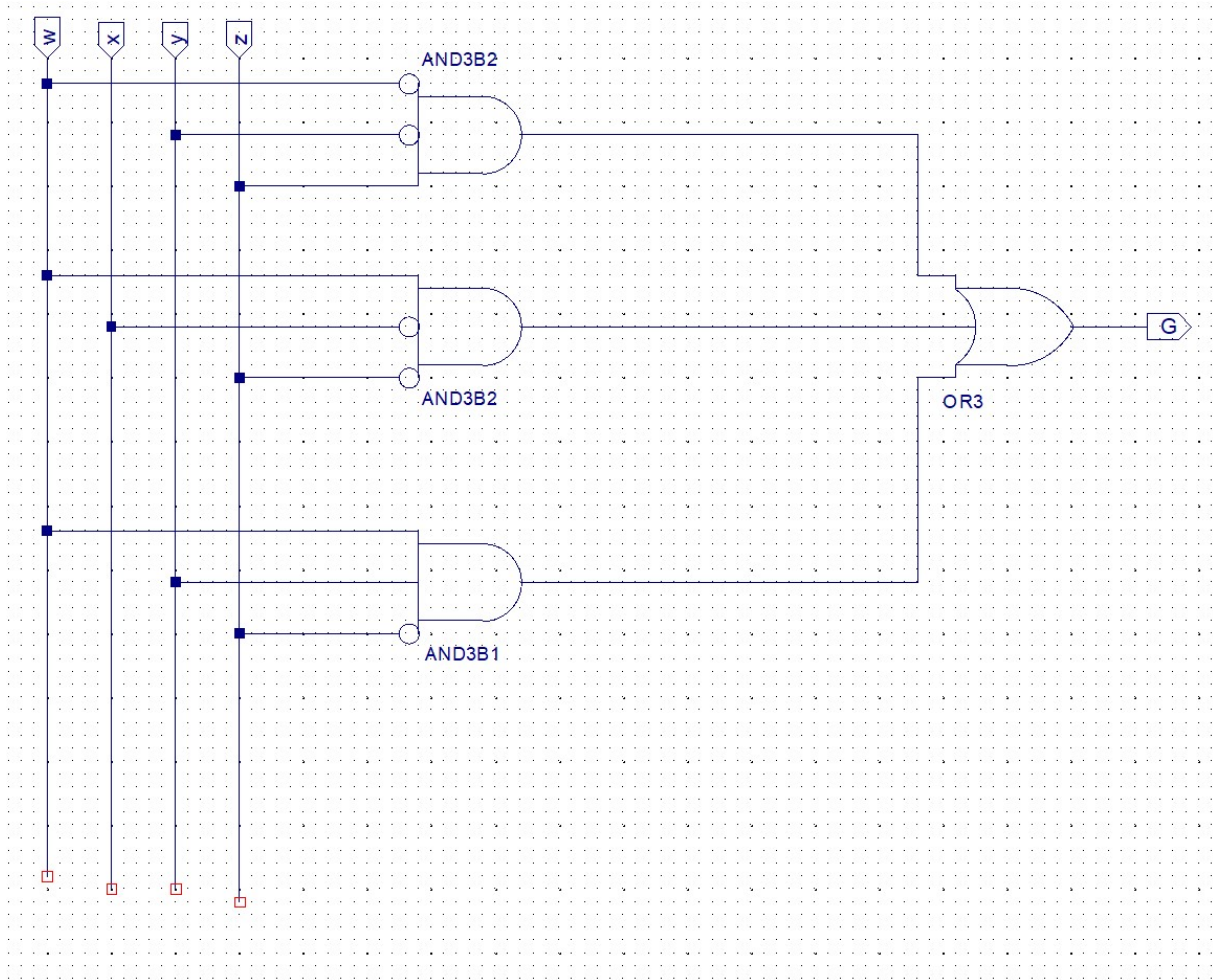
Minimalizacja metodą siatek Karnaugh

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

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

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

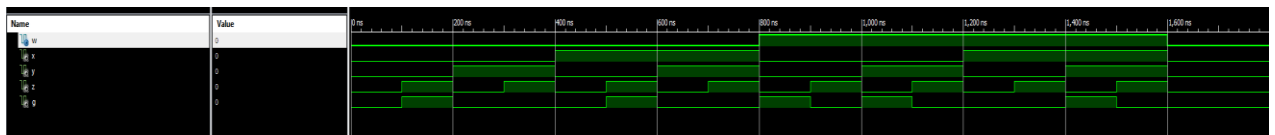
$$G = \bar{w}\bar{y}z + w\bar{x}\bar{z} + wy\bar{z}$$



Kod VHDL:

```
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 sche2_sche2_sch_tb IS
21  END sche2_sche2_sch_tb;
22  ARCHITECTURE behavioral OF sche2_sche2_sch_tb IS
23
24      COMPONENT sche2
25      PORT( w : IN STD_LOGIC;
26            x : IN STD_LOGIC;
27            y : IN STD_LOGIC;
28            z : IN STD_LOGIC;
29            G : OUT STD_LOGIC);
30      END COMPONENT;
31
32      SIGNAL w : STD_LOGIC;
33      SIGNAL x : STD_LOGIC;
34      SIGNAL y : STD_LOGIC;
35      SIGNAL z : STD_LOGIC;
36      SIGNAL G : STD_LOGIC;
37
38  BEGIN
39
40      UUT: sche2 PORT MAP(
41          w => w,
42          x => x,
43          y => y,
44          z => z,
45          G => G
46      );
47
48      -- *** Test Bench - User Defined Section ***
49      z <= '0', '1' after 100 ns, '0' after 200 ns, '1' after 300 ns, '0' after 400 ns,
50        '1' after 500 ns, '0' after 600 ns, '1' after 700 ns, '0' after 800 ns,
51        '1' after 900 ns, '0' after 1000 ns, '1' after 1100 ns, '0' after 1200 ns,
52        '1' after 1300 ns, '0' after 1400 ns, '1' after 1500 ns, '0' after 1600 ns;
53      y <= '0', '1' after 200 ns, '0' after 400 ns, '1' after 600 ns, '0' after 800 ns,
54        '1' after 1000 ns, '0' after 1200 ns, '1' after 1400 ns, '0' after 1600 ns;
55      x <= '0', '1' after 400 ns, '0' after 800 ns, '1' after 1200 ns, '0' after 1600 ns;
56      w <= '0', '1' after 800 ns, '0' after 1600 ns;
57      -- *** End Test Bench - User Defined Section ***
58
59  END;
```

Symulacja ISIM:



Na płytce CPLD lampka LED paliła się gdy funkcja przyjmowała wartość „0”, zaś gasła gdy miała wartość „1”.

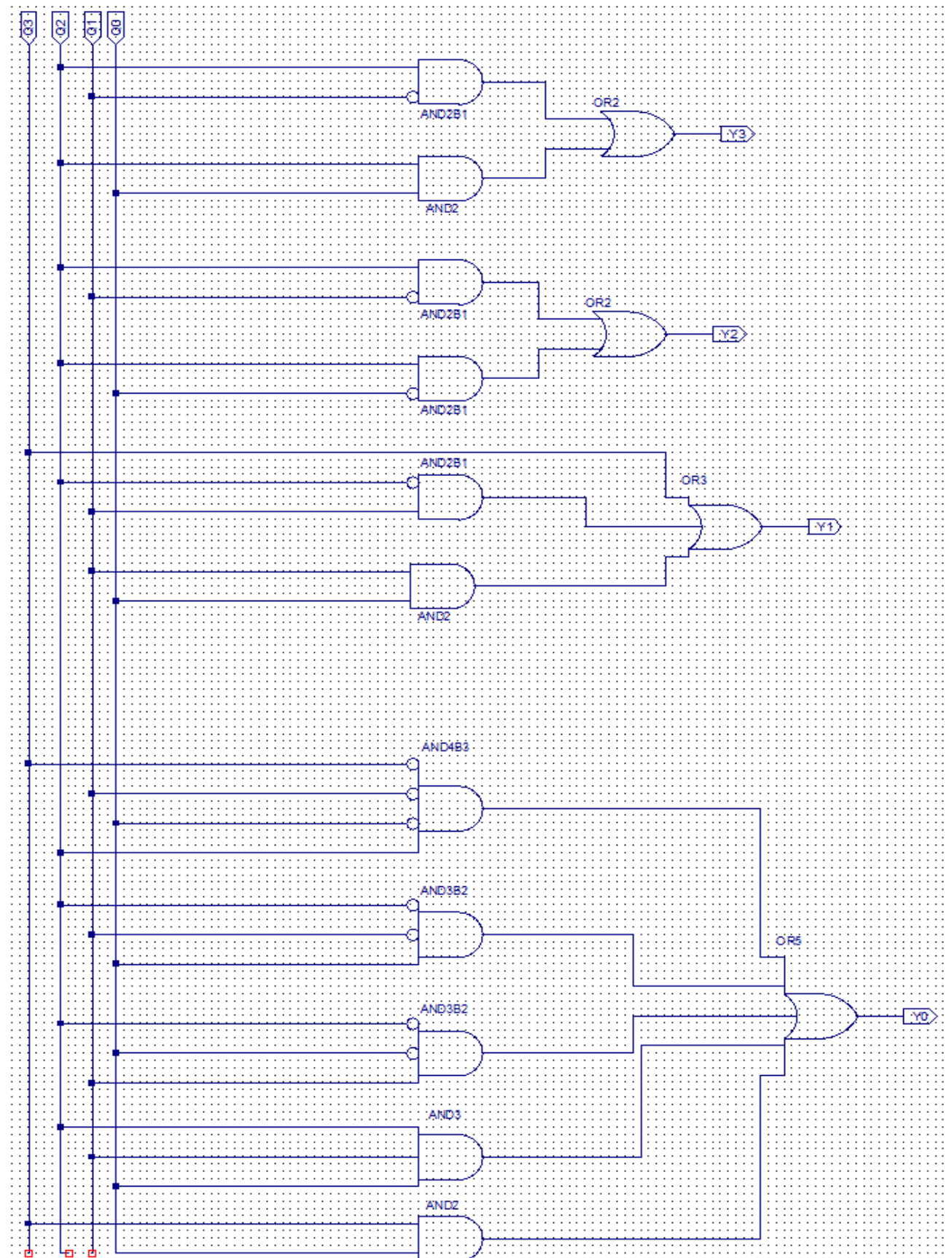
3. Translator kodu

Dany układ jest translatorem kodu Graya na kod Aikena.

Kod Graya charakteryzuje się tym, że dwa kolejne słowa kodowe różnią się tylko stanem jednego bitu.

Kod Aikena charakteryzuje się za to tym, że każdej cyfrze od 0 do 9 w systemie dziesiętnym przypisana jest konkretna 4. bitowa sekwencja liczb w systemie dwójkowym.

Schemat translatora:



Proces minimalizacji:

dwójkowy	Graya	Aikena
0000	0000	0000
0001	0001	0001
0010	0011	0010
0011	0010	0011
0100	0110	0100
0101	0111	1011
0110	0101	1100
0111	0100	1101
1000	1100	1110
1001	1101	1111
1010	1111	\
1011	1110	\
1100	1010	\
1101	1011	\
1110	1001	\
1111	1000	\

Gdzie “\” oznacza stan nieokreślony

Tabela przejść:

	Q(t)				Q(t+1)			
	Q_3	Q_2	Q_1	Q_0	Q'_3	Q'_2	Q'_1	Q'_0
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	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	1	1	0	1
5	0	1	0	1	1	1	0	0
6	0	1	1	0	0	1	0	0
7	0	1	1	1	1	0	1	1
8	1	0	0	0	\	\	\	\
9	1	0	0	1	\	\	\	\
10	1	0	1	0	\	\	\	\
11	1	0	1	1	\	\	\	\
12	1	1	0	0	1	1	1	0
13	1	1	0	1	1	1	1	1
14	1	1	1	0	\	\	\	\
15	1	1	1	1	\	\	\	\

Siatki Karnaugh:

$Q_3 Q_2 \backslash Q_1 Q_0$	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	1	1	\	\
10	\	\	\	\

$$Q'_3 = Q_2 \overline{Q_1} + Q_2 Q_0$$

$Q_3 Q_2 \backslash Q_1 Q_0$	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	1	1	\	\
10	\	\	\	\

$$Q'_2 = Q_2 \overline{Q_1} + Q_2 \overline{Q_0}$$

$Q_3 Q_2 \backslash Q_1 Q_0$	00	01	11	10
00	0	0	1	1
01	0	0	1	0
11	1	1	\	\
10	\	\	\	\

$$Q'_1 = Q_3 + \overline{Q_2} Q_1 + Q_1 \overline{Q_0}$$

$Q_3 Q_2 \backslash Q_1 Q_0$	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	\	\
10	\	\	\	\

$$Q'_0 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_2} \overline{Q_1} Q_0 + \overline{Q_2} Q_1 \overline{Q_0} + Q_2 Q_1 Q_0 + Q_3 Q_0$$

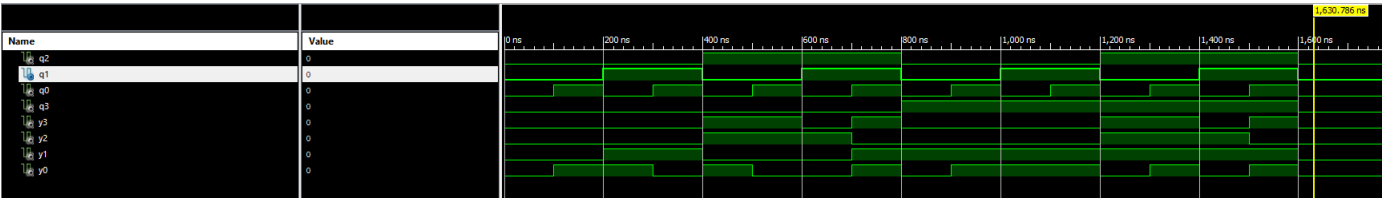
Kod VHDL:

```

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 sche3_sche3_sch_tb IS
21  END sche3_sche3_sch_tb;
22  ARCHITECTURE behavioral OF sche3_sche3_sch_tb IS
23
24  COMPONENT sche3
25  PORT(
26      Q2 : IN STD_LOGIC;
27      Q1 : IN STD_LOGIC;
28      Q0 : IN STD_LOGIC;
29      Q3 : IN STD_LOGIC;
30      Y3 : OUT STD_LOGIC;
31      Y2 : OUT STD_LOGIC;
32      Y1 : OUT STD_LOGIC;
33      Y0 : OUT STD_LOGIC);
34  END COMPONENT;
35
36  SIGNAL Q2 : STD_LOGIC;
37  SIGNAL Q1 : STD_LOGIC;
38  SIGNAL Q0 : STD_LOGIC;
39  SIGNAL Q3 : STD_LOGIC;
40  SIGNAL Y3 : STD_LOGIC;
41  SIGNAL Y2 : STD_LOGIC;
42  SIGNAL Y1 : STD_LOGIC;
43  SIGNAL Y0 : STD_LOGIC;
44
45  BEGIN
46
47  UUT: sche3 PORT MAP(
48      Q2 => Q2,
49      Q1 => Q1,
50      Q0 => Q0,
51      Q3 => Q3,
52      Y3 => Y3,
53      Y2 => Y2,
54      Y1 => Y1,
55      Y0 => Y0
56  );
57
58  -- *** Test Bench - User Defined Section ***
59  Q0 <= '0', '1' after 100 ns, '0' after 200 ns, '1' after 300 ns, '0' after 400 ns,
60      '1' after 500 ns, '0' after 600 ns, '1' after 700 ns, '0' after 800 ns,
61      '1' after 900 ns, '0' after 1000 ns, '1' after 1100 ns, '0' after 1200 ns,
62      '1' after 1300 ns, '0' after 1400 ns, '1' after 1500 ns, '0' after 1600 ns;
63  Q1 <= '0', '1' after 200 ns, '0' after 400 ns, '1' after 600 ns, '0' after 800 ns,
64      '1' after 1000 ns, '0' after 1200 ns, '1' after 1400 ns, '0' after 1600 ns;
65  Q2 <= '0', '1' after 400 ns, '0' after 800 ns, '1' after 1200 ns, '0' after 1600 ns;
66  Q3 <= '0', '1' after 800 ns, '0' after 1600 ns;
67  -- *** End Test Bench - User Defined Section ***
68  END;
69

```

Wyniki działania symulatora:



4. Wnioski

Wszystkie układy działały na płytce CPLD według ustalonych założeń.