

Sprawozdanie

Inżynieria Układów Programowalnych

Nazwa ćwiczenia	Generator parzystości
Numer ćwiczenia	1
Imię	Mateusz
Nazwisko	Gabryel
Numer indeksu	181329
Kierunek	Elektronika i Telekomunikacja
Numer grupy dziekańskiej	3

Opis działania zadania:

Układ po zaprogramowaniu, służy jako generator parzystości. Na podstawie wprowadzonego słowa 8-bitowego, wyświetla informacje na siedmiosegmentowym wyświetlaczu znajdującego się na płytce FPGA, o parzystej lub nieparzystej liczbie „jedynek” w słowie wejściowym.

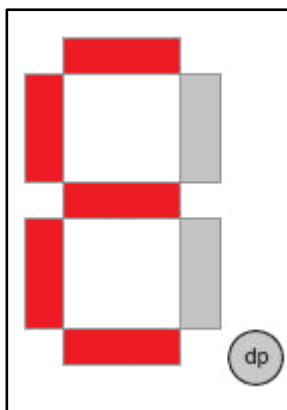
W sytuacji gdy wprowadzone słowo zawiera parzystą liczbę „jedynek”, układ sygnalizuje to na wyświetlaczu siedmiosegmentowym za pomocą wyświetlanej litery „E”.

W sytuacji gdy wprowadzone słowo zawiera nieparzystą liczbę „jedynek”, układ sygnalizuje to na wyświetlaczu siedmiosegmentowym za pomocą wyświetlanej litery „O”.

W przeprowadzonej symulacji układu przełączniki są zainicjalizowane na OFF, a następnie co 100 ms przełączane na ON do momentu gdy wszystkie przełączniki są ON.

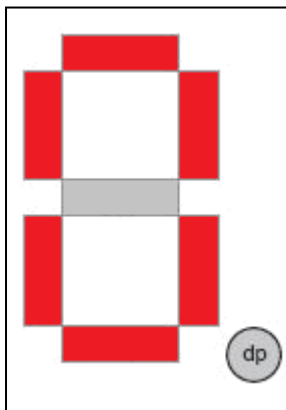
Aktywowanie poszczególnych segmentów wyświetlacza odbywa się poprzez podanie stanu niskiego „0” na wyprowadzenie AN0 - AN3.

Aktywowanie poszczególnych segmentów pojedynczego wyświetlacza odbywa się poprzez podanie stanu niskiego „0” na wyprowadzeniach A-G.



Do uzyskania na wyświetlaczu litery „E” potrzebne jest podanie następujących sygnałów: A=’0’, B=’1’, C=’1’, D=’0’, E=’0’, F=’0’, G=’0’, DP=’1’ („01100001”). Podanie sygnałów w takiej kolejności spowoduje zmianę wyświetlacza na sytuację tak jak na zdjęciu obok.

Kolor czerwony na zdjęciu symbolizuje świeceniu się segmentu, a kolor szary symbolizuje wygaszenie segmentu.



Do uzyskania na wyświetlaczu litery „O” potrzebne jest podanie następujących sygnałów: A='0', B='0', C='0', D='0', E='0', F='0', G='1', DP='1' („00000011”). Podanie sygnałów w takiej kolejności spowoduje zmianę wyświetlacza na sytuację tak jak na zdjęciu obok.

Kolor czerwony na zdjęciu symbolizuje świecenie się segmentu, a kolor szary symbolizuje wygaszenie segmentu.

Opis pliku top.vhd:

1.

2.

3.

4.

5.

6.

```

1  -- Mateusz Gabryel 181329 EiT 3
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.std_logic_signed.all;
5  use IEEE.std_logic_unsigned.all;
6  use IEEE.std_logic_arith.all;
7  use std.standard.all;
8
9  ENTITY top IS
10     PORT ( sw_i : in STD_LOGIC_VECTOR (7 downto 0);
11            led7_an_o : out STD_LOGIC_VECTOR (3 downto 0);
12            led7_seg_o : out STD_LOGIC_VECTOR (7 downto 0)
13          );
14  END top;
15
16  ARCHITECTURE Behavioral OF top IS
17     signal a : STD_LOGIC;
18  BEGIN
19     -- podanie 1 skutkuje wygaszeniem (AN3 -1, AN2 -1, AN1 -1)
20     -- podanie 0 skutkuje działaniem (ANO -0)
21     led7_an_o(3) <= '1';
22     led7_an_o(2) <= '1';
23     led7_an_o(1) <= '1';
24     led7_an_o(0) <= '0';
25
26     -- podanie 1 skutkuje wygaszeniem (dp - 1)
27     -- podanie 0 skutkuje działaniem (SEG7 -0, SEG4 -0, SEG3 -0, SEG2 -0)
28     -- stan zmienia tylko 6,5,1
29     -- reszta (7,4,3,2,0) jest stała
30     a <= sw_i(0) xor sw_i(1) xor sw_i(2) xor sw_i(3) xor sw_i(4) xor sw_i(5) xor sw_i(6) xor sw_i(7);
31
32     led7_seg_o(7) <= '0';
33     led7_seg_o(6) <= not a;
34     led7_seg_o(5) <= not a;
35     led7_seg_o(4) <= '0';
36
37     led7_seg_o(3) <= '0';
38     led7_seg_o(2) <= '0';
39     led7_seg_o(1) <= a;
40     led7_seg_o(0) <= '1';
41  END Behavioral;

```

1. Deklaracja użycia biblioteki IEEE oraz deklaracja użycia pakietów i ich składników.
2. Deklaracja przydziału portów:
 - sw_i: słowo wprowadzane, wejściowe
 - led7_an_o: pojedyncze wyświetlacze
 - led7_seg_o: segmenty pojedynczego wyświetlacza
3. Utworzenie sygnału „a” potrzebnego do zrealizowania funkcji parzystości.
4. Podanie odpowiednich stanów na wyprowadzeniach AN0 - AN3 do ich aktywacji lub wygaszenia. W zadaniu wynik ma być wyświetlany na wyświetlaczu AN0, czyli podanie „0” uaktywnia go. Pozostałe wyświetlacze trzeba wygaszyć, podając sygnały „1”.
5. Przypisanie do sygnału „a” funkcji parzystości. Funkcja ta została zrealizowana za pomocą funkcji logicznej XOR oraz słowa wejściowego.
6. Przypisanie odpowiednich wartości sygnałów na poszczególne segmenty pojedynczych wyświetlaczy. W zadaniu swój stan będą zmieniać tylko 3 segmenty („b”, „c”, „g”), które są zależne od słowa wejściowego, reszta segmentów będzie miała cały czas ten sam stan, więc stałe wartości, niezależne od słowa wejściowego. Na segmencie („b”, „g”) sygnał „a” musi zostać zanegowany za pomocą funkcji logicznej NOT.

Do prawidłowego działania układu na płycie FPGA jest potrzebny plik

```

1  #virtual DIP Switches
2  set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { sw_i[0] }];
3  set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { sw_i[1] }];
4  set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { sw_i[2] }];
5  set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { sw_i[3] }];
6  set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { sw_i[4] }];
7  set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { sw_i[5] }];
8  set_property -dict { PACKAGE_PIN E17 IOSTANDARD LVCMOS33 } [get_ports { sw_i[6] }];
9  set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { sw_i[7] }];
10 #7-segment display
11 #segments (active low)
12 set_property -dict { PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[7] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
13 set_property -dict { PACKAGE_PIN R10 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[6] }]; #IO_25_14 Sch=cb
14 set_property -dict { PACKAGE_PIN K16 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[5] }]; #IO_25_15 Sch=cc
15 set_property -dict { PACKAGE_PIN K13 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[4] }]; #IO_L17P_T2_A26_15 Sch=cd
16 set_property -dict { PACKAGE_PIN P15 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[3] }]; #IO_L13P_T2_MRCC_14 Sch=ce
17 set_property -dict { PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[2] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
18 set_property -dict { PACKAGE_PIN L18 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[1] }]; #IO_L4P_T0_D04_14 Sch=cg
19 set_property -dict { PACKAGE_PIN H15 IOSTANDARD LVCMOS33 } [get_ports { led7_seg_o[0] }]; #IO_L19N_T3_A21_VREF_15 Sch=dp
20 #digits (active low)
21 set_property -dict { PACKAGE_PIN J17 IOSTANDARD LVCMOS33 } [get_ports { led7_an_o[0] }]; #IO_L23P_T3_F0E_B_15 Sch=an[0]
22 set_property -dict { PACKAGE_PIN J18 IOSTANDARD LVCMOS33 } [get_ports { led7_an_o[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
23 set_property -dict { PACKAGE_PIN T9 IOSTANDARD LVCMOS33 } [get_ports { led7_an_o[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
24 set_property -dict { PACKAGE_PIN J14 IOSTANDARD LVCMOS33 } [get_ports { led7_an_o[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
25 ## Configuration options, can be used for all designs
26 set_property CONFIG_VOLTAGE 3.3 [current_design]
27 set_property CFGBVS VCCO [current_design]

```

constraint iup1.xdc:

W pliku tb.vhd została wykonana symulacja poprzez podanie odpowiednich wartości słowa wejściowego:

```
1  -- Mateusz Gabryel 181329 EiT 3
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  ENTITY tb IS
6  END tb;
7
8  ARCHITECTURE Behavioral OF tb IS
9
10 COMPONENT top IS
11     PORT ( sw_i : in STD_LOGIC_VECTOR (7 downto 0);
12           led7_an_o : out STD_LOGIC_VECTOR (3 downto 0);
13           led7_seg_o : out STD_LOGIC_VECTOR (7 downto 0)
14         );
15 END COMPONENT top;
16
17 signal sw_i : STD_LOGIC_VECTOR (7 downto 0);
18 signal led7_an_o : STD_LOGIC_VECTOR (3 downto 0);
19 signal led7_seg_o : STD_LOGIC_VECTOR (7 downto 0);
20
21 BEGIN
22     uut: top PORT MAP (
23         sw_i => sw_i,
24         led7_seg_o => led7_seg_o,
25         led7_an_o => led7_an_o
26     );
27
28 tb: PROCESS
29 BEGIN
30     sw_i <= "00000000"; --przelaczniki OFF (00000000)
31     wait for 100 ms;
32     sw_i <= "00000001"; --przelaczniki 7 OFF i 1 ON
33     wait for 100 ms;
34     sw_i <= "00000011"; --przelaczniki 6 OFF i 2 ON
35     wait for 100 ms;
36     sw_i <= "00000111"; --przelaczniki 5 OFF i 3 ON
37     wait for 100 ms;
38     sw_i <= "00001111"; --przelaczniki 4 OFF i 4 ON
39     wait for 100 ms;
40     sw_i <= "00011111"; --przelaczniki 3 OFF i 5 ON
41     wait for 100 ms;
42     sw_i <= "00111111"; --przelaczniki 2 OFF i 6 ON
43     wait for 100 ms;
44     sw_i <= "01111111"; --przelaczniki 1 OFF i 7 ON
45     wait for 100 ms;
46     sw_i <= "11111111"; --przelaczniki ON (11111111)
47     wait;
48 END PROCESS;
49
50 END Behavioral;
```

Wyniki symulacji:

