

Sprawozdanie

Inżynieria Układów Programowalnych

Nazwa ćwiczenia	Licznik w kodzie Graya
Numer ćwiczenia	3
Imię	Mateusz
Nazwisko	Gabryel
Numer indeksu	181329
Kierunek	Elektronika i Telekomunikacja
Numer grupy dziekańskiej	3

Opis działania zadania:

Układ po zaprogramowaniu, służy jako trzybitowy licznik, który liczy w kodzie Graya. Zliczanie odbywa się, gdy pojawia się narastające zbocze zegara. Impulsy zegara realizowane są za pomocą przycisku BTN0 wciskanego przez użytkownika. Układ posiada reset asynchroniczny, który jest realizowany za pomocą przycisku BTN3. Wciśnięcie tego przycisku skutkuje wyzerowaniem licznika.

W przeprowadzonej symulacji oraz weryfikacji na płycie. Reset licznika został wykonany dwa razy – na początku symulacji oraz weryfikacji i w jej trakcie. Drugi reset został wykonany w taki sposób, aby było widać jego działanie na wyjściu licznika. Symulacja oraz weryfikacja przedstawia wszystkie możliwe stany licznika.

Wyniki weryfikacji na płycie są przedstawione na wyjściach układu, czyli diodach świecących LED0, LED1 oraz LED2.

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 ( clk_i : in STD_LOGIC;
11            rst_i : in STD_LOGIC;
12            led_o : out STD_LOGIC_VECTOR (2 downto 0)
13            );
14  END top;
15
16  ARCHITECTURE Behavioral OF top IS
17
18     signal a: STD_LOGIC_VECTOR (2 downto 0);
19     signal b: STD_LOGIC_VECTOR (2 downto 0);
20
21     BEGIN
22
23     PROCESS (clk_i, rst_i,a,b)
24
25     BEGIN
26
27     if(rst_i='1') then a<="000";
28         elsif(rising_edge(clk_i)) then
29             a<=a+1;
30     END if;
31
32     --przepisujemy MSB
33     --kazdy kolejny bit realizowany jest za pomoca
34     --funkcji XOR
35
36     b(2)<=a(2);
37     b(1)<=a(1) xor a(2);
38     b(0)<=a(1) xor a(0);
39
40     --przepisanie wyników na diody led
41
42     led_o(2)<=b(2);
43     led_o(1)<=b(1);
44     led_o(0)<=b(0);
45
46     END PROCESS;
47  END Behavioral;
```

1. Deklaracja użycia biblioteki IEEE oraz deklaracja użycia pakietów i ich składników.

2. Deklaracja przydziału portów:

- clk_i: zegar
- rst_i: reset
- led_o: wyjścia układu, diody świecące

3. Utworzenie sygnałów „a” oraz „b” potrzebnych do zrealizowania licznika w kodzie Graya.

4. Utworzenie pętli odpowiadającej za liczenie, gdy pojawia się narastające zbocze oraz resetowanie licznika gdy został wciśnięty reset

5. Zrealizowanie kodowania w kodzie Graya. Przepisywany jest MSB, a każdy kolejny bit otrzymywany jest za pomocą funkcji XOR.

6. Przepisanie wyników z punktu 5 na wyjścia układu.

Do prawidłowego działania układu na płytce FPGA jest potrzebny plik constraint iup3.xdc:

```
1 #virtual Buttons
2 set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { clk_i }]; # virtual BTN0
3 set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { rst_i }]; # virtual BTN3
4 #LEDs
5 set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { led_o[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
6 set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { led_o[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
7 set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { led_o[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
8 #the following property allows the use of a regular pin for clock input:
9 set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets clk_i]
10 ## Configuration options, can be used for all designs
11 set_property CONFIG_VOLTAGE 3.3 [current_design]
12 set_property CFGBVS VCCO [current_design]
```

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 ( clk_i : in STD_LOGIC;
12           rst_i : in STD_LOGIC;
13           led_o : out STD_LOGIC_VECTOR (2 downto 0)
14         );
15 END COMPONENT top;
16
17 signal clk_i : STD_LOGIC := '0';
18 signal rst_i : STD_LOGIC := '0';
19 signal led_o : STD_LOGIC_VECTOR (2 downto 0);
20 constant PERIOD : TIME := 200ms;
21
22 BEGIN
23     uut: top PORT MAP (
24         clk_i => clk_i,
25         rst_i => rst_i,
26         led_o => led_o
27     );
28
29     clk_i <= not clk_i after PERIOD/2; -- realizacja prostego zegara
30
31     tb: PROCESS
32     BEGIN
33         rst_i <= '1'; -- I reset na początku
34         wait for 200ms;
35         rst_i <= '0'; -- wyłączenie resetu
36         wait for 400ms;
37         rst_i <= '1'; -- II reset w trakcie
38         wait for 200ms;
39         rst_i <= '0'; -- wyłączenie resetu
40         wait;
41     END PROCESS;
42 END Behavioral;
```

Wyniki symulacji:

