

# Sprawozdanie

## Inżynieria Układów Programowalnych

Nazwa ćwiczenia	Dzielnik częstotliwości
Numer ćwiczenia	5
Imię	Mateusz
Nazwisko	Gabryel
Numer indeksu	181329
Kierunek	Elektronika i Telekomunikacja
Numer grupy dziekańskiej	3

### Opis działania zadania:

Układ po zaprogramowaniu, służy jako dzielnik częstotliwości. Zaprojektowany dzielnik częstotliwości dzieli wejściową częstotliwość 100 MHz przez  $N$ , gdzie  $N$  stanowi sumę cyfr indeksu studenta zaliczającego ćwiczenie. W moim przypadku jest to  $1+8+1+3+2+9 = 24$ . W kodzie została umożliwiona łatwa zmiana wartości  $N$  poprzez constant. Układ posiada reset asynchroniczny, który jest realizowany za pomocą przycisku BTN3. Wciśnięcie tego przycisku skutkuje wyzerowaniem układu.

W przeprowadzonej symulacji oraz weryfikacji na płytce. Reset układu 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 dzielnika. Symulacja oraz weryfikacja przedstawia wszystkie możliwe stany dzielnika.

Wynik weryfikacji na płytce jest przedstawiony na wyjściu układu, czyli diodzie świecącej LED0.

Miganie diody LED0 z częstotliwością 1 Hz, zostało uzyskane poprzez ustawienie wartości  $N$  na wartość 100000000.

## Opis pliku top.vhd:

1.	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
2.	9    ENTITY top IS 10        PORT ( clk_i : in STD_LOGIC; 11               rst_i : in STD_LOGIC; 12               led_o : out STD_LOGIC 13               ); 14    END top; 15
3.	16    ARCHITECTURE Behavioral OF top IS 17 18        -- N = 1+8+1+3+2+9 = 24 19        constant N : integer := 24; -- dla symulacji 20        --constant N : integer := 100000000; -- dla weryfikacji na płytce 21 22        signal a : integer := 1; --licznik 23        signal b : STD_LOGIC := '0'; --zegar po dzieleniu 24 25        BEGIN 26        PROCESS (clk_i, rst_i, a, b) 27        BEGIN 28            if (rst_i = '1') then 29               a <= 1; 30               b <= '0'; 31            elsif (rising_edge(clk_i)) then 32               a <= a + 1; 33               if (a=N) then 34                   b <= NOT b; 35                   a <= 1; 36               END if; 37            END if; 38        --przepisanie wyników na diode led 39        led_o <= b; 40 41        END PROCESS; 42    END Behavioral;
4.	
5.	

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

2. Deklaracja przydziału portów:

- clk\_i: zegar wejściowy
- rst\_i: reset
- led\_o: wyjście układu, dioda świecąca

3. Utworzenie stałej N służącej do ustawiania wartości, do której zliczanie impulsów ma się odbywać. Utworzenie sygnałów „a” oraz „b” potrzebnych do zrealizowania dzielnika częstotliwości.

4. Utworzenie pętli odpowiadającej za „liczenie”, gdy pojawia się narastające zbocze oraz resetowanie dzielnika gdy został wciśnięty reset. W pętli realizowane jest zanegowanie wartości sygnału „b” gdy wartość sygnału „a” będzie równa ustawionej wartości N.

5. Przepisanie wyników z sygnału „b” na wyjście układu.

Do prawidłowego działania układu na płytce FPGA jest potrzebny plik constraint iup5.xdc:

```
1  # Clock signal
2  set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports { clk_i }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
3  create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports { clk_i }];
4  # Reset
5  set_property -dict { PACKAGE_PIN G17     IOSTANDARD LVCMOS33 } [get_ports { rst_i }];
6  # LED
7  set_property -dict { PACKAGE_PIN H17     IOSTANDARD LVCMOS33 } [get_ports { led_o }]; #IO_L18P_T2_A24_15 Sch=led[0]
8  ## Configuration options, can be used for all designs
9  set_property CONFIG_VOLTAGE 3.3 [current_design]
10 set_property CFGBVS VCCO [current_design]
```

W pliku tb.vhd została wykonana symulacja (na potrzeby symulacji musi zostać ustawiona wartość  $N = 24$ ):

```
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
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;
20 constant PERIOD : TIME := 10 ns;
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';
34         wait for 100 ns;
35         rst_i <= '0';
36         wait for 800 ns;
37         rst_i <= '1';
38         wait for 100 ns;
39         rst_i <= '0';
40         wait;
41     END PROCESS;
42 END Behavioral;
```

Wyniki symulacji:

