

Sprawozdanie - laboratorium 3

Maksim Birszel, Piotr Karon

11 grudnia 2019

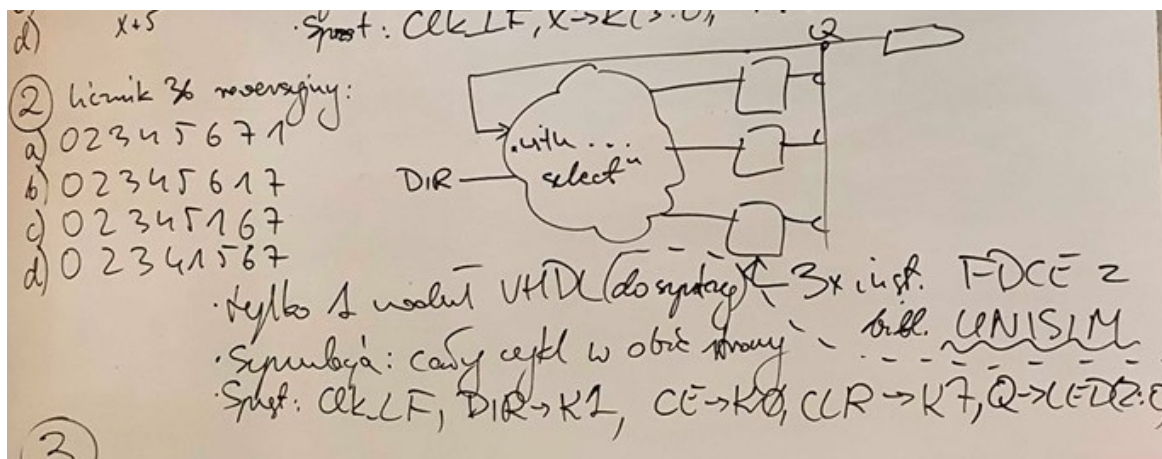
1 Zadanie

Zaprojektować moduł, zasymulować i zaimplementować na płycie ZL-9572 3-bitowy licznik rewersyjny o następującej sekwencji:

0->2->3->4->5->6->7->1

2 Założenia

- Projekt ma zawierać tylko 1 moduł VHDL
- Symulacja: cały cykl w obie strony
- Sprzęt: CLK_LF, DIR->K1, CE->K0, CLR->K7, Q->LED(2:0)



3 Moduł VHDL

```
1 entity licznik3b_rewers is
2     Port ( CE : in  STD_LOGIC;
3           CLR : in  STD_LOGIC;
```

```

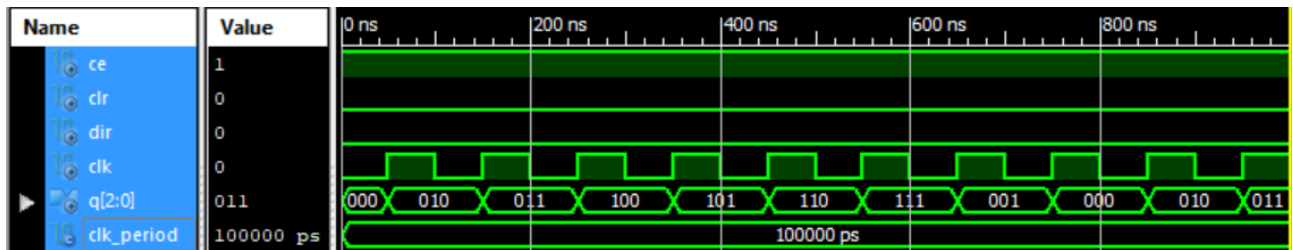
4          DIR : in STD_LOGIC;
5          CLK : in STD_LOGIC;
6          Q : out STD_LOGIC_VECTOR (2 downto 0)
7              );
8  end licznik3b_rewers;
9
10 architecture Behavioral of licznik3b_rewers is
11     signal intQ : STD_LOGIC_VECTOR (2 downto 0);
12     signal D : STD_LOGIC_VECTOR (2 downto 0);
13 begin
14
15     FDCE0 : FDCE port map ( D => D(0), CE => CE, Q => intQ(0),
16                             CLR => CLR, C => CLK );
17
18     FDCE1 : FDCE port map ( D => D(1), CE => CE, Q => intQ(1),
19                             CLR => CLR, C => CLK );
20
21     FDCE2 : FDCE port map ( D => D(2), CE => CE, Q => intQ(2),
22                             CLR => CLR, C => CLK );
23
24     Q <= intQ;
25
26     with DIR & intQ select
27         D <= "010" when "0000",
28             "000" when "0001",
29             "011" when "0010",
30             "100" when "0011",
31             "101" when "0100",
32             "110" when "0101",
33             "111" when "0110",
34             "001" when "0111",
35             "001" when "1000",
36             "111" when "1001",
37             "000" when "1010",
38             "010" when "1011",
39             "011" when "1100",
40             "100" when "1101",
41             "101" when "1110",
42             "110" when "1111",
43             "XXX" when others;
44
45 end Behavioral;

```

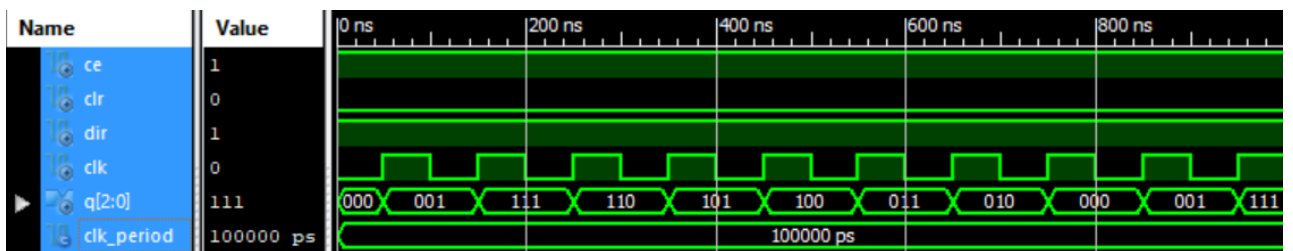
4 Test Bench

```
1 ENTITY licznik3bit_testbench IS
2 END licznik3bit_testbench;
3
4 ARCHITECTURE behavior OF licznik3bit_testbench IS
5
6     COMPONENT licznik3b_rewers
7     PORT(
8         CE : IN  std_logic;
9         CLR : IN  std_logic;
10        DIR : IN  std_logic;
11        CLK : IN  std_logic;
12        Q : OUT std_logic_vector(2 downto 0));
13    END COMPONENT;
14
15    signal CE : std_logic := '1';
16    signal CLR : std_logic := '0';
17    signal DIR : std_logic := '1';
18    signal CLK : std_logic := '0';
19
20    signal Q : std_logic_vector(2 downto 0);
21
22    constant CLK_period : time := 100 ns;
23
24 BEGIN
25
26    uut: licznik3b_rewers PORT MAP (
27        CE => CE,
28        CLR => CLR,
29        DIR => DIR,
30        CLK => CLK,
31        Q => Q);
32
33    CLK_process : process
34    begin
35
36        CLK <= '0';
37        wait for CLK_period/2;
38        CLK <= '1';
39        wait for CLK_period/2;
40
41    end process;
42
43 END;
```

5 Symulacja behawioralna



Symulacja dla DIR = '0', liczenie w przód.



Symulacja dla DIR = '1', liczenie w tył.

6 Wnioski

Moduł udało się poprawnie zaimplementować, a symulacja przebiegła według wstępnych założeń.

Najwięcej problemów sprawiło odpowiednie przypisanie wartości wejściom, wyjściom oraz sygnałom, aby licznik był rewersyjny.

Nie udało nam się skończyć licznika w czasie zajęć i musieliśmy dokończyć w domu. Z tego powodu jego działanie nie zostało przetestowane na płytce, ale z racji tego, że symulacja przebiegała według założeń, możemy przypuszczać, że przy odpowiednim przypisaniu komponentów płytki do wejść/wyjść modułu działałaby on poprawnie.