

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: Licznik synchroniczny sterowany w Spartanie		Laboratorium nr: 7
Data wykonania ćwiczenia	19.12.2023	Ocena:
Data oddania sprawozdania	9.01.2024	

Na zajęciach zapoznano się z płytką FPGA po raz pierwszy. Przerobiono licznik wykonany na płycie CPLD na tą właśnie płytkę.

1. Plik modułowy VHDL

```
5  -- Create Date:      07:53:07 12/05/2023
6  -- Design Name:
7  -- Module Name:      counter - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.NUMERIC_STD.ALL;
23
24 -- Uncomment the following library declaration if using
25 -- arithmetic functions with Signed or Unsigned values
26 --use IEEE.NUMERIC_STD.ALL;
27
28 -- Uncomment the following library declaration if instantiating
29 -- any Xilinx primitives in this code.
30 --library UNISIM;
31 --use UNISIM.VComponents.all;
32
33 entity counter is
34     Port ( START_STOP : in  STD_LOGIC;
35           PAUSE       : in  STD_LOGIC;
36           LOAD        : in  STD_LOGIC;
37           RESET       : in  STD_LOGIC;
38           OUT_O       : out  STD_LOGIC_VECTOR (7 downto 0);
39           IN_I        : in  STD_LOGIC_VECTOR (7 downto 0);
40           REVERSE     : in  STD_LOGIC;
41           CLOCK       : in  STD_LOGIC);
42 end counter;
43
44 architecture Behavioral of counter is
45
46     signal number : UNSIGNED(7 downto 0) := (others => '0');
47     signal isStarted : STD_LOGIC := '0';
48     signal slow_counter : INTEGER range 0 to 200_000_000 := 0;
```

```

50
51 begin
52
53     process (START_STOP)
54     begin
55         if START_STOP = '1' then
56             isStarted <= '1';
57
58         elsif START_STOP = '0' then
59             isStarted <= '0';
60
61         end if;
62
63     end process;
64
65     process (CLOCK, PAUSE, LOAD, RESET)
66     begin
67         if RESET = '1' then
68             number <= (others => '0');
69
70         elsif rising_edge(CLOCK) then
71             if slow_counter >= 100_000_000 then
72                 if PAUSE = '0' and isStarted = '1' then
73                     if LOAD = '1' then
74                         number <= UNSIGNED(IN_I);
75                     elsif slow_counter >= 100_000_000 then
76                         if REVERSE = '0' then
77                             number <= number + 1;
78                         elsif REVERSE = '1' then
79                             number <= number - 1;
80                         end if;
81                     end if;
82                 end if;
83                 slow_counter <= 0;
84             else
85                 slow_counter <= slow_counter + 1;
86             end if;
87         end if;
88     end process;
89
90     OUT_O <= STD_LOGIC_VECTOR(number);
91
92 end Behavioral;

```

Względem poprzedniego laboratorium, dodano zmienną *slow_counter* typu INT. Cała logika działania licznika została „otoczona” warunkiem, który sprawdza czy zmienna jest większa bądź równa 100 mln. Jeśli nie to dodajemy do niej 1. Gdy zmienna osiągnie wartość 100mln, licznik „dopuszczany” jest do swojej logiki po czym znów przypisujemy do zmiennej wartość 0. Taka operacja pozwala na zwolnienie tempa liczenia licznika.

2. Plik testowy VHDL

Plik testowy nie uległ zmianie względem poprzedniego laboratorium.

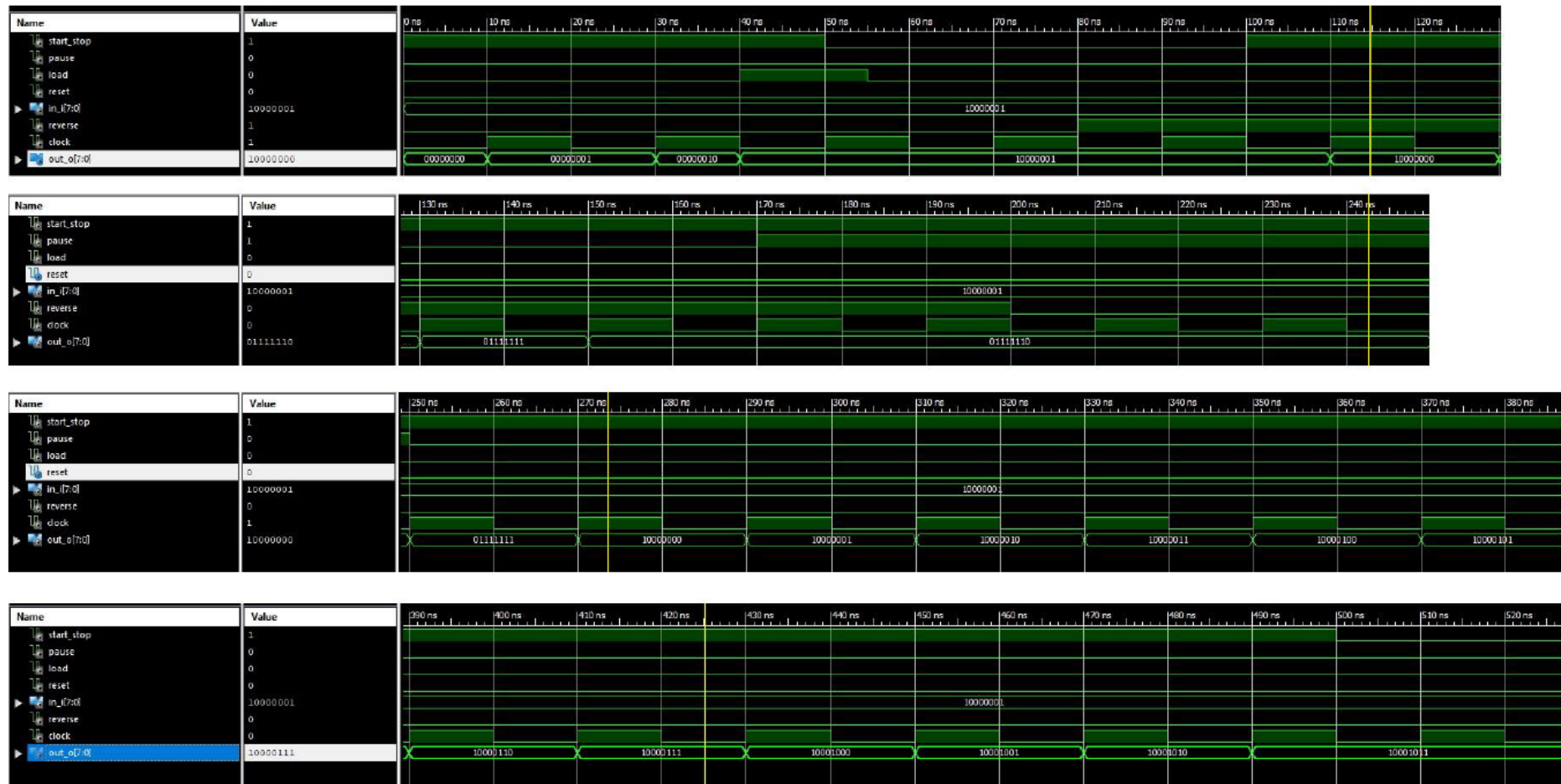
```

28     LIBRARY ieee;
29     USE ieee.std_logic_1164.ALL;
30
31     -- Uncomment the following library declaration if using
32     -- arithmetic functions with Signed or Unsigned values
33     --USE ieee.numeric_std.ALL;
34
35     ENTITY test IS
36     END test;
37
38     ARCHITECTURE behavior OF test IS
39
40         -- Component Declaration for the Unit Under Test (UUT)
41
42         COMPONENT counter
43         PORT(
44             START_STOP : IN std_logic;
45             PAUSE : IN std_logic;
46             LOAD : IN std_logic;
47             RESET : IN std_logic;
48             OUT_O : OUT std_logic_vector(7 downto 0);
49             IN_I : IN std_logic_vector(7 downto 0);
50             REVERSE : IN std_logic;
51             CLOCK : IN std_logic
52         );
53         END COMPONENT;
54
55
56         --Inputs
57         signal START_STOP : std_logic := '0';
58         signal PAUSE : std_logic := '0';
59         signal LOAD : std_logic := '0';
60         signal RESET : std_logic := '0';
61         signal IN_I : std_logic_vector(7 downto 0) := (others => '0');
62         signal REVERSE : std_logic := '0';
63         signal CLOCK : std_logic := '0';
64
65         --Outputs
66         signal OUT_O : std_logic_vector(7 downto 0);
67
68     BEGIN
69
70         -- Instantiate the Unit Under Test (UUT)
71         uut: counter PORT MAP (
72             START_STOP => START_STOP,
73             PAUSE => PAUSE,
74             LOAD => LOAD,
75             RESET => RESET,
76             OUT_O => OUT_O,
77             IN_I => IN_I,
78             REVERSE => REVERSE,
79             CLOCK => CLOCK
80         );
81
82         CLOCK <= not CLOCK after 10 ns;
83         START_STOP <= '1', '0' after 50 ns, '1' after 100 ns, '0' after 500 ns;
84         IN_I <= "10000001";
85         LOAD <= '1' after 40 ns, '0' after 55 ns;
86         PAUSE <= '1' after 170 ns, '0' after 250 ns;
87         RESET <= '1' after 1000 ns, '0' after 1050 ns;
88         REVERSE <= '1' after 80 ns, '0' after 200 ns;
89
90     END;
91

```

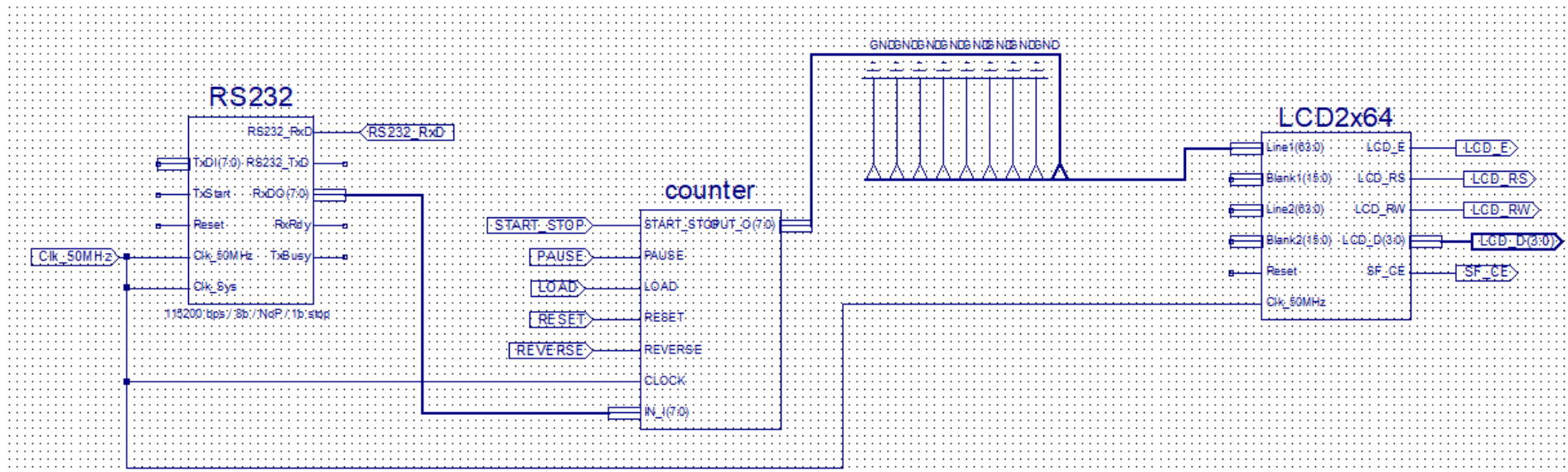
3. Symulacja

Symulacja również przebiegła tak samo jak na poprzednim laboratorium.



Rysunek 4 Symulacja układu licznika

4. Podpięcie klawiatury i wyświetlacza



Tym razem nie było wymagane oddzielenie zegara dla wyświetlacza i licznika od modułu RS232. Używany jest jeden sygnał zegarowy *Clk_50MHz* o częstotliwości 50MHz.

5. Pliki z rozszerzeniem .ucf

W odróżnieniu od plików dla układu CPLD, tutaj dla każdej czarnej skrzynki zrobiony jest osobny plik z rozszerzeniem .ucf

GenIO.ucf:

```
1  # soldered 50MHz Clock.
2  NET "Clk_50MHz" LOC = "C9" | IOSTANDARD = LVTTTL;
3  NET "Clk_50MHz" PERIOD = 20.0ns HIGH 50%;
4
5  # Ignore cross-clock domain data paths in timing analysis
6  #NET "Clk50" TNM = "TNM_Clk50";
7  #NET "ClkSys" TNM = "TNM_ClkSys";
8  #TIMESPEC "TS_12" = FROM "TNM_ClkSys" TO "TNM_Clk50" TIG;
9  #TIMESPEC "TS_13" = FROM "TNM_Clk50" TO "TNM_ClkSys" TIG;
10
11 #
12 # Push-buttons (Press = Hi)
13 #
14 #NET "btn_north" LOC = "V4" | IOSTANDARD = LVTTTL | PULLDOWN;
15 #NET "btn_east" LOC = "H13" | IOSTANDARD = LVTTTL | PULLDOWN;
16 NET "LOAD" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN;
17 #NET "btn_west" LOC = "D18" | IOSTANDARD = LVTTTL | PULLDOWN;
18
19 #
20 # Slide switches (Up = Hi)
21 #
22 NET "START_STOP" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP;
23 NET "REVERSE" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP;
24 NET "PAUSE" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP;
25 #NET "SW<3>" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP;
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61 # RS-232 Serial Port: DCE
62 NET "RS232_RXD" LOC = "R7" | IOSTANDARD = LVTTTL ;
63 #NET "RS232_TXD" LOC = "M14" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = SLOW ;
64 # RS-232 Serial Port: DTE
65 #NET "RS232_RXD" LOC = "U8" | IOSTANDARD = LVTTTL ;
66 #NET "RS232_TXD" LOC = "M13" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = SLOW ;
```

LCD.ucf:

```
1  # ==== Character LCD (LCD) ====
2  NET "LCD_E" LOC = "M18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
3  NET "LCD_RS" LOC = "L18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
4  NET "LCD_RW" LOC = "L17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
5  # LCD data connections are shared with StrataFlash connections SF_D<11:8>
6  NET "LCD_D<0>" LOC = "R15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
7  NET "LCD_D<1>" LOC = "R16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
8  NET "LCD_D<2>" LOC = "P17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
9  NET "LCD_D<3>" LOC = "M15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
10 NET "SF_CE" LOC = "D16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
```

Wartość licznika można ustawić z klawiatury za pomocą portu RS232, po czym naciskając przycisk z oznaczeniem *btn_south* załadować. Licznik zaczyna liczenie po przełączeniu switcha *SW0* w stan

wysoki. Licznik liczy w przód gdy na switchu *SW1* jest stan niski, a w tył gdy na switchu *SW1* jest stan wysoki. Gdy switch *SW2* ustawi się w tryb wysoki, licznik wstrzymuje liczenie. Na wyświetlaczu jest pokazywana aktualna wartość licznika.

6. Wnioski

Na dobrą sprawę, przerobienie licznika na nowo poznaną płytkę FPGA, nie wymagało wiele modyfikacji. Trzeba było zbudować dzielnik, który spowalniał takt zegara nowo-poznanej płytki oraz użyć plików z rozszerzeniem .ucf stworzonych z myślą o FPGA. Nowa płytka Spartan-3E jest o wiele bardziej rozbudowana niż CPLD pod względem ilości portów czy szerszego wyświetlacza, co pozwala na większe możliwości programistyczne.