Universitatea Tehnică "Gheorghe Asachi" din Iaşi

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE

ELECTRONICĂ DIGITALĂ

Proiect

Tema: COUNTER BCD – v6

Studenţi:

Papă Teodora

Grupa : 1212B

Coordonator:

Asist. Drd. Ionica P*****

2024

Tema proiectului:

COUNTER BCD – v6

1. Specificaţiile proiectului:

COUNTER BCD – v6

Să se implementeze în FPGA prin descriere în limbaj VHDL, un sistem secvenţial de 4 biţi : cu reset prioritar activ pe 1; cu o intrare de selecţie, din care să se stabilească funcţionarea de registru paralel sau numărător binar direct. Perioada sa fie de 1 secundă. Afișarea se va face pe Displayul 7 segmente de pe placă de dezvoltare.

Fişierul bitstream rezultat în urma procesului de implementare va fi verificat utilizând placa de dezvoltare BASYS3.

2. Modulul COUNTER BCD – v6

Modulul COUNTER BCD (Binary-Coded Decimal) – v6 este utilizat pentru a număra și afișa valori numerice într-un format BCD. Acesta permite în funcție de selecție incrementarea unui contor, cu opțiunea de resetare la valoarea inițială sau afișarea unei valori citite direct. Modulul este proiectat să fie ușor de integrat în sisteme digitale, oferind funcționalități precum control precis al secvenței de numărare și conversia rezultatelor în format compatibil pentru afișare pe display-uri digitale.

3. Metoda de implementare

Utilizarea resurselor: circuit FPGA, limbajul VHDL, programul de sinteză Vivado.

4. Descrierea (scurtă) a sistemului de dezvoltare BASYS 3

BASYS 3 este o placă de dezvoltare FPGA destinată în special învățării și prototipurilor rapide. Aceasta utilizează un FPGA Xilinx Artix-7.

Circuitele FPGA (Field Programmable Gate Array) sunt circuite reconfigurabile prin intermediul unor limbaje de descriere hardware (Hardware Description Languages – HDL) cu scopul de a obține anumite structuri logice dedicate. Celula de bază a unui circuit FPGA este blocul logic configurabil (Configurabe Logic Block). Mai multe celule configurabile pot sa fie conectate între ele prin intermediul unei matrice de conexiuni.

Proiectarea unor structuri logice în tehnologia FPGA presupune implicarea a trei resurse: circuitul FPGA, tool-ul de sinteză și limbajul de descriere VHDL în acest proiect.

BASYS 3 include un set de periferice, cum ar fi interfețe pentru intrări și ieșiri (LED-uri, butoane, switch-uri, display-uri 7 segmente), precum și un conector pentru plăci externe, facilitând astfel integrarea cu diverse sisteme hardware.

5. Editarea fişierului VHDL

Folosim semnalele :

- counter_clk și out_div penru divizorul de frecvență;
- reset_in pentru a reseta valoarea la 0;
- select_in pentru a selecta funcționalitatea dorită: numărător sau registru paralel;
- Val_in este valoarea citită când se dorește funcționarea registrului paralel;
- an și seg cu care activăm display-ul și afișăm cifrele în zecimal pe display-ul cu 7 segmente;
- a conține valoarea ce trebuie afișată;

și variabila n pentru calcularea perioadei de 1s.

Când intrarea de selecție (select_in) ia valoarea 1, va fi activat numărătorul binar direct, iar când select_in ia valoarea 0, va funcționa registrul paralel (valorile ce trebuie afișate sunt cele primite prin Val_in). La activarea semnalului de reset (reset_in), indiferent de valoarea de selecție, valoarea afișată va fi 0. Valoarea care se va afișa va fi cea stocată în semnalul a.

Perioada de 1 secundă se implementează printr-un divizor de frecvență, folosind semnalul counter_clk, care se activează la fiecare 10 ns. Divizorul de frecvență va folosi o variabilă n pentru a număra semnalele counter_clk și a genera un semnal out_div cu o perioadă de 1 secundă.

```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use IEEE.std_logic_unsigned.all;
25
26  -- Uncomment the following library declaration if using
27  -- arithmetic functions with Signed or Unsigned values
28  --use IEEE.NUMERIC_STD.ALL;
29
30  -- Uncomment the following library declaration if instantiating
31  -- any Xilinx leaf cells in this code.
32  --library UNISIM;
33  --use UNISIM.VComponents.all;
34
35  entity bcd is
36      Port (
37              counter_clk : in STD_LOGIC; --semnalul de ceas de 100MHz
38              select_in : in STD_LOGIC;
39              reset_in : in STD_LOGIC;
40              an   : out STD_LOGIC_VECTOR(3 downto 0);
41              seg      : out STD_LOGIC_VECTOR(6 downto 0); --segmentele din display
42              Val_in : in STD_LOGIC_VECTOR(3 downto 0);
43              out_div : buffer STD_LOGIC ---pt divizor
44              );
45  end bcd;
46
47  architecture Behavioral of bcd is
48
49          signal a : std_logic_vector(3 downto 0):="1001"; -- semnal pentru         bcd
50  begin
51      -- activam Displayul 7
52     an <= "0000";
```

```
    process(counter_clk,reset_in) --divizor de frecventa, facem perioada 1 secunda --reducem de la 100MHz la 1Hz
              variable n : integer range 0 to 1000000000;
    begin
         if reset_in ='1' then
                     n:=0;--reset pentru divizor
                     out_div  <= '0';
         elsif counter_clk'event and counter_clk='1' then ------------divizor frecventa
                 if n < 100000000 then
                     n := n + 1;
                 else
                     n := 0;
                 end if;

                 if n < 50000000 then
                     out_div  <= '1';
                 else
                     out_div  <= '0';
                 end if;
          end if;
     end process;


      process(reset_in,select_in, out_div,a,Val_in)
      begin
           if reset_in ='1' then
               a <= (others => '0');
           else
                if select_in='1' then ---bcd/registru paralel
                        if out_div = '1' and out_div'event then
                                if a = "1001" then  -- Dacă se ajunge la 9 (1001), revine la 0
                                    a <= "0000";
                                else
                                    a <= a + 1;  -- Incrementare
                                end if;
                         end if;
                  else
                        a <= Val_in;
                  end if;
           end if;
                end process;


     seg(0) <= (a(2) and not(a(1)) and not(a(0))) or (not(a(3)) and not(a(2)) and not(a(1)) and a(0));
     seg(1) <= (a(2) and not(a(1)) and a(0)) or (a(2) and a(1) and not(a(0)));
     seg(2) <= (not(a(2)) and a(1) and not(a(0)));
     seg(3) <= (a(2) and not(a(1)) and not(a(0))) or (a(2) and a(1) and a(0)) or (not(a(3)) and not(a(2)) and not(a(1)) and a(0));
     seg(4) <= a(0) or (a(2) and not(a(1)));
     seg(5) <= (not(a(3)) and not(a(2)) and a(0)) or (a(1) and a(0)) or (not(a(3)) and not(a(2)) and a(1));
     seg(6) <= (not(a(3)) and not(a(2)) and not(a(1))) or (a(2) and a(1) and a(0));

end Behavioral;
```

6. Editarea fişierului de constrângeri

Placa de dezvoltare BASYS3 are un genetator de clock cu o frecvenţă de 100 MHz a cărui ieşire este conectată la pinul W5 al circuitului FPGA, deci legăm counter_clk de pinul W5 :

```
 #Clock signal
set_property PACKAGE_PIN W5 [get_ports counter_clk]
    set_property IOSTANDARD LVCMOS33 [get_ports counter_clk]
    create_clock -add -name sys_clk_pin -period 10.00  -waveform {0 5} [get_ports counter_clk]
```

Conectăm reset_in și select_in la primele 2 switch-uri din stânga plăcii și cele 4 intrări ale Val_in la cele 4 switch-uri din dreapta plăcii :

```
## Switches
set_property PACKAGE_PIN V17 [get_ports {Val_in[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Val_in[0]}]
set_property PACKAGE_PIN V16 [get_ports {Val_in[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Val_in[1]}]
set_property PACKAGE_PIN W16 [get_ports {Val_in[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Val_in[2]}]
set_property PACKAGE_PIN W17 [get_ports {Val_in[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {Val_in[3]}]
#set_property PACKAGE_PIN W15 [get_ports {swt[4]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {swt[4]}]
#set_property PACKAGE_PIN V15 [get_ports {swt[5]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {swt[5]}]
#set_property PACKAGE_PIN W14 [get_ports {swt[6]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {swt[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
set_property PACKAGE_PIN T1 [get_ports {select_in}]
    set_property IOSTANDARD LVCMOS33 [get_ports {select_in}]
set_property PACKAGE_PIN R2 [get_ports {reset_in}]
    set_property IOSTANDARD LVCMOS33 [get_ports {reset_in}]
```

Legăm și out_div la led-ul LD-8 al plăcii (se va aprinde la trecerea unei secunde) :

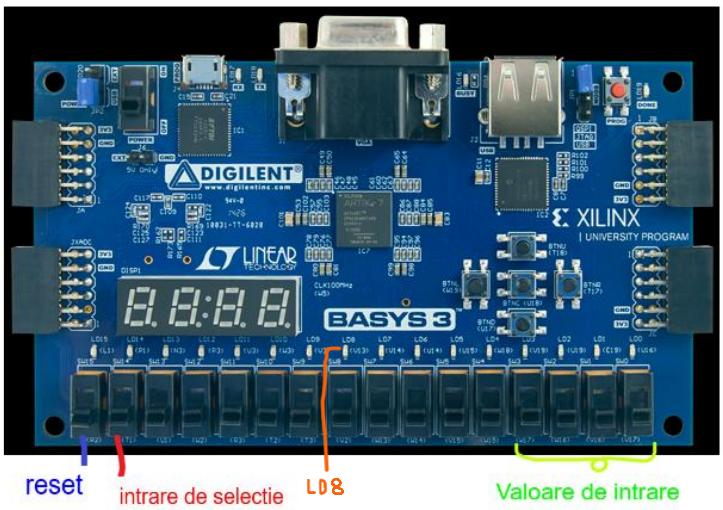
```
set_property PACKAGE_PIN V13 [get_ports {out_div}]
    set property IOSTANDARD LVCMOS33 [get ports {out_div}]
```

Și în cele din urmă conectăm displayul de 7 segmente pentru afișarea cifrelor și vectorul de semnal seg la cele 7 segmente pentru construirea cifrelor :

```
#7 segment display
set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]

#set_property PACKAGE_PIN V7 [get_ports dp]
    #set_property IOSTANDARD LVCMOS33 [get_ports dp]

set_property PACKAGE_PIN U2 [get_ports {an[0]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property PACKAGE_PIN U4 [get_ports {an[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
set_property PACKAGE_PIN V4 [get_ports {an[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
set_property PACKAGE_PIN W4 [get_ports {an[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]
```



7. Descrierea paşilor de sinteză şi testarea circuitului rezultat

Circuitul pentru proiectul COUNTER BCD v6 a fost implementat în VHDL folosind Vivado. După scrierea codului, am utilizat un fișier pentru constrângeri de tip .xdc pentru maparea semnalelor pe pinii plăcii FPGA (Basys 3), specificând conexiunile pentru semnalul de ceas (counter_clk), semnalul de resetare (reset_in), intrarea de selecție (select_in), valoarea citită la funcţionarea de registru paralel (Val_in) și ieșirile circuitului.

Pentru validare, am rulat o simulare comportamentală (Run Behavioral Simulation) în Vivado, care a confirmat că circuitul numără corect în format BCD, iar resetarea funcționează conform așteptărilor. După simulare, am realizat sinteza circuitului, verificând utilizarea eficientă a resurselor hardware și respectarea constrângerilor.

La codul sintetizat am generat fişierul bitstream apoi am făcut verificarea proiectului prin încărcarea fişierului bitstream în circuitul plăcii de dezvoltare BASYS3.

Testarea, folosind displayul 7 segmente de pe placa de dezvoltare pentru ieșire și semnale de intrare aplicate direct, a demonstrat funcționarea corectă a circuitului în condiții reale, confirmând incrementarea în format BCD, a funcționării registrului paralel și resetarea la atingerea limitei sau la activarea semnalului de resetare.

8. Concluzii

Proiectul a fost un exercițiu de învățare foarte util, în care am aplicat cunoștințele teoretice despre circuite digitale dobândite la materia Electronică digitală. Provocările întâmpinate în timpul testării pe hardware real ne-au arătat cât de important este să îți verifici constant designul, să faci ajustări și să optimizezi resursele, pentru a obține performanța dorită.

În ansamblu, proiectul a fost o oportunitate de a aprofunda concepte de sinteză, simulare și implementare pe o placă FPGA și de a înțelege cum se transformă ideile teoretice în soluții funcționale.

Bibliografie:

1. VHDL Reference Manual, http://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf

2. BASYS 3 Reference Manual, https://reference.digilentinc.com/reference/programmable-logic/basys-3/reference-manual