



**Universitatea Tehnică “Gheorghe Asachi” din  
Iași**



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

# **ELECTRONICĂ DIGITALĂ**

## **proiect**

**Tema: COUNTER BCD**

Studenti: Donici Leonardo-Mario, Gherghel Stefan, Radu Daniel

Grupa : 1207A

Coordonator:

Asist. Drd. Marius Obreja

**2023**

## **1. Specificațiile proiectului:**

### **COUNTER BCD**

Să se implementeze în FPGA prin descriere în limbaj VHDL, un sistem secvențial de 4 biți : cu reset prioritari activ pe 1; cu două intrări de selecție din care să se stabilească funcționare de registru serie sau numărător BCD direct/invers.

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

## **2. Modulul**

### **ENTITY:**

1. Variabila “**schimba**” stabilește modul de funcitonare (registru serie sau numerotor BCD)
2. sursa de clock “**counter\_clk**” va fi generată de un buton extern
3. Variabila “**ud\_in**” determină sensul de numărare
4. variabila “**reset\_in**” determină resetarea numaratorului la valoarea “0000”
5. afisarea se face cu ajutorul a 4 leduri, variabila “**Data\_out**”
6. variabila “**D\_IN**” este pe 4 biti și reprezintă data care se încarcă în registrul serie
7. variabila “**load**” determină încărcarea lui “**D\_IN**” în registru
8. variabila “**shift\_n**” stabilește dacă va fi executata shiftarea sau nu
9. modulul are ca sursa de clock extern butonul din centru

### **ARCHITECTURE:**

1. Dacă “**schimba = = 0**” se intră în modul COUNTER BCD
  - Dacă restul este setat pe 1, vectorul **a** se zerorizează
  - Altfel, la fiecare tact al clockului “**a**” crește sau scade, în funcție de “**ud\_in**”.  
(1-crește, 0-scade)

## 2. Daca “schimba == 1” se intra in modul REGISTRU SERIE

- Daca “load” este setat pe 0, atunci se va încarca in registru valoarea din “D\_IN”
- Altfel, dacă “shift\_n” este 0, a va fi shiftata

In final, valoarea lui **a** este asignata iesirii.

## 3. Metoda de implementare

Pentru implementarea acestui modul s-au folosit programul de sinteza Vivado si limbajul VHDL. Implementarea proiectului a fost facuta printr-o descriere comportamentalala. S-a proiectat entitatea reg\_v1.

Fisierul bitstream creeat de programul Vivado a fost testat cu ajutorul placii BASYS 3 Artix-7 xc7a35tcpg236-1.

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

Placa de dezvoltare BASYS 3 este un circuit de dezvoltare complet si ready-to-use bazat pe ultimele Artix-7 Field Programmable Gate Array(FPGA) produse de Xilinx. Cu o mare capacitate de FPGA si cu o colectie de porturi USB, VGA si altele, placa de dezvoltare BASYS 3 permite proiectarea unor design-uri variate, atat circuite introductoare combinationale, cat si circuite secventiale complexe ca procesoarele si controlerile embedded.

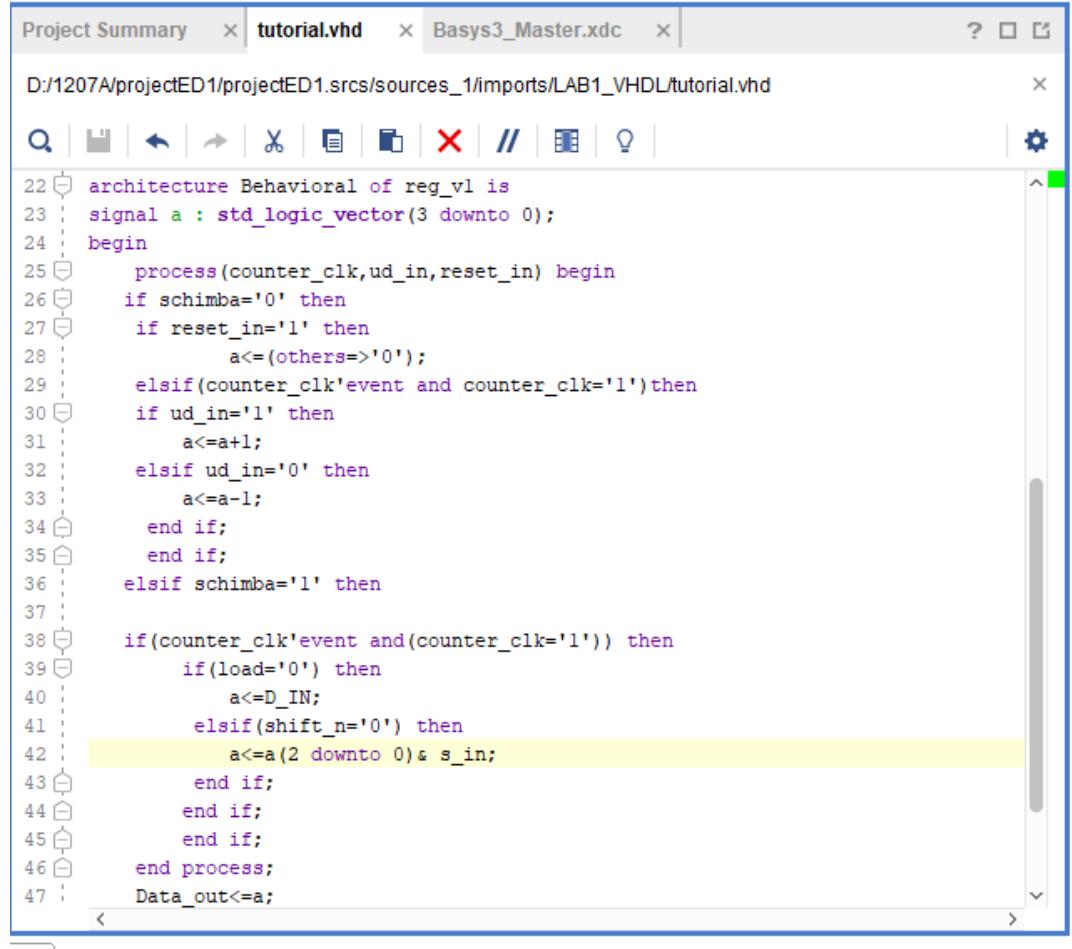
## 5. Editarea fișierului VHDL

Entity:

The screenshot shows a VHDL editor window with the file "tutorial.vhd" open. The code displays the entity declaration for "reg\_v1". The entity has several input ports: counter\_clk, ud\_in, reset\_in, Data\_out, D\_IN, load, schimba, s\_in, and shift\_n. It also has an output port, a generic port, and a local time variable. The code is color-coded for syntax highlighting.

```
library ieee;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_1164.all;
entity reg_v1 is
port(   counter_clk: in STD_LOGIC;
        ud_in: in STD_LOGIC;
        reset_in:in STD_LOGIC;
        Data_out: out STD_LOGIC_VECTOR(3 downto 0);
        D_IN: in std_logic_vector(3 downto 0);
        load:in std_logic;
        schimba:in std_logic;
        s_in:in std_logic;
        shift_n:in std_logic
      );
end reg_v1;
```

## Architecture:



The screenshot shows a VHDL editor window with the following project summary:

- Project Summary
- tutorial.vhd
- Basys3\_Master.xdc

The code editor displays the following VHDL code:

```
22 architecture Behavioral of reg_vl is
23 signal a : std_logic_vector(3 downto 0);
24 begin
25 process(counter_clk,ud_in,reset_in) begin
26 if schimba='0' then
27 if reset_in='1' then
28 a<=(others=>'0');
29 elsif(counter_clk'event and counter_clk='1')then
30 if ud_in='1' then
31 a<=a+1;
32 elsif ud_in='0' then
33 a<=a-1;
34 end if;
35 end if;
36 elsif schimba='1' then
37
38 if(counter_clk'event and(counter_clk='1')) then
39 if(load='0') then
40 a<=D_IN;
41 elsif(shift_n='0') then
42 a<=a(2 downto 0)& s_in;
43 end if;
44 end if;
45 end if;
46 end process;
47 Data_out<=a;
```

The line `a<=a(2 downto 0)& s_in;` is highlighted with a yellow background.

## 6. Editarea fișierului de constrângeri

Clock-ul dat de buton:

```
##Buttons
set_property PACKAGE_PIN U18 [get_ports counter_clk]
set_property IO_STANDARD LVCMOS33 [get_ports counter_clk]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets counter_clk]
```

## Switch-uri:

```
## Switches
set_property PACKAGE_PIN V17 [get_ports {ud_in}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {ud_in}]
set_property PACKAGE_PIN V16 [get_ports {reset_in}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {reset_in}]
set_property PACKAGE_PIN W16 [get_ports {D_IN[0]}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {D_IN[0]}]
set_property PACKAGE_PIN W17 [get_ports {D_IN[1]}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {D_IN[1]}]
set_property PACKAGE_PIN W15 [get_ports {D_IN[2]}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {D_IN[2]}]
set_property PACKAGE_PIN W15 [get_ports {D_IN[3]}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {D_IN[3]}]
set_property PACKAGE_PIN W14 [get_ports {load}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {load}]
set_property PACKAGE_PIN W13 [get_ports {shift_n}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {shift_n}]
set_property PACKAGE_PIN V2 [get_ports {s_in}]
    set_property IOSTANDARD LVC莫斯33 [get_ports {s_in}]

set_property PACKAGE_PIN R2 [get_ports schimba]
    set_property IOSTANDARD LVC莫斯33 [get_ports schimba]
```

## LED-uri:

```
## LEDs
set_property PACKAGE_PIN U16 [get_ports {Data_out[0]}]
set_property IOSTANDARD LVC莫斯33 [get_ports {Data_out[0]}]

set_property PACKAGE_PIN E19 [get_ports {Data_out[1]}]
set_property IOSTANDARD LVC莫斯33 [get_ports {Data_out[1]}]

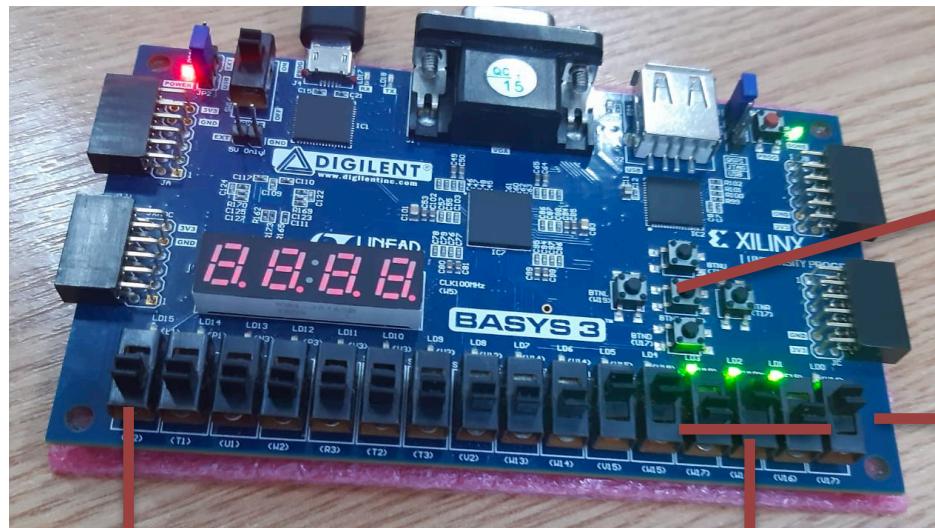
set_property PACKAGE_PIN U19 [get_ports {Data_out[2]}]
set_property IOSTANDARD LVC莫斯33 [get_ports {Data_out[2]}]

set_property PACKAGE_PIN V19 [get_ports {Data_out[3]}]
set_property IOSTANDARD LVC莫斯33 [get_ports {Data_out[3]}]
```

## 7. Descrierea pașilor de sinteză și testarea circuitului rezultat

1. S-a creeat un proiect nou in programul Vivado
2. S-a implementat modulul printr-o descriere comportamentalala
3. S-a editat fisierul de constrangeri in vederea realizarii legaturilor intre switch-uri si intrari, butonului din mijloc si clock si 4 leduri
4. S-a realizat analiza RTL (Register Transfer Level)
5. S-a sintetizat modulul (pentru se vedea design-ul sintetizat)
6. S-a lansat implementarea proiectului care a avut ca efect final generarea fisierului bitstream
7. S-a programat placa de dezvoltare BASYS 3 cu fisierul bitstream si s-a testat functionarea corespunzatoare a modulului implementat

## 8. Fotografii cu functionarea modulului:

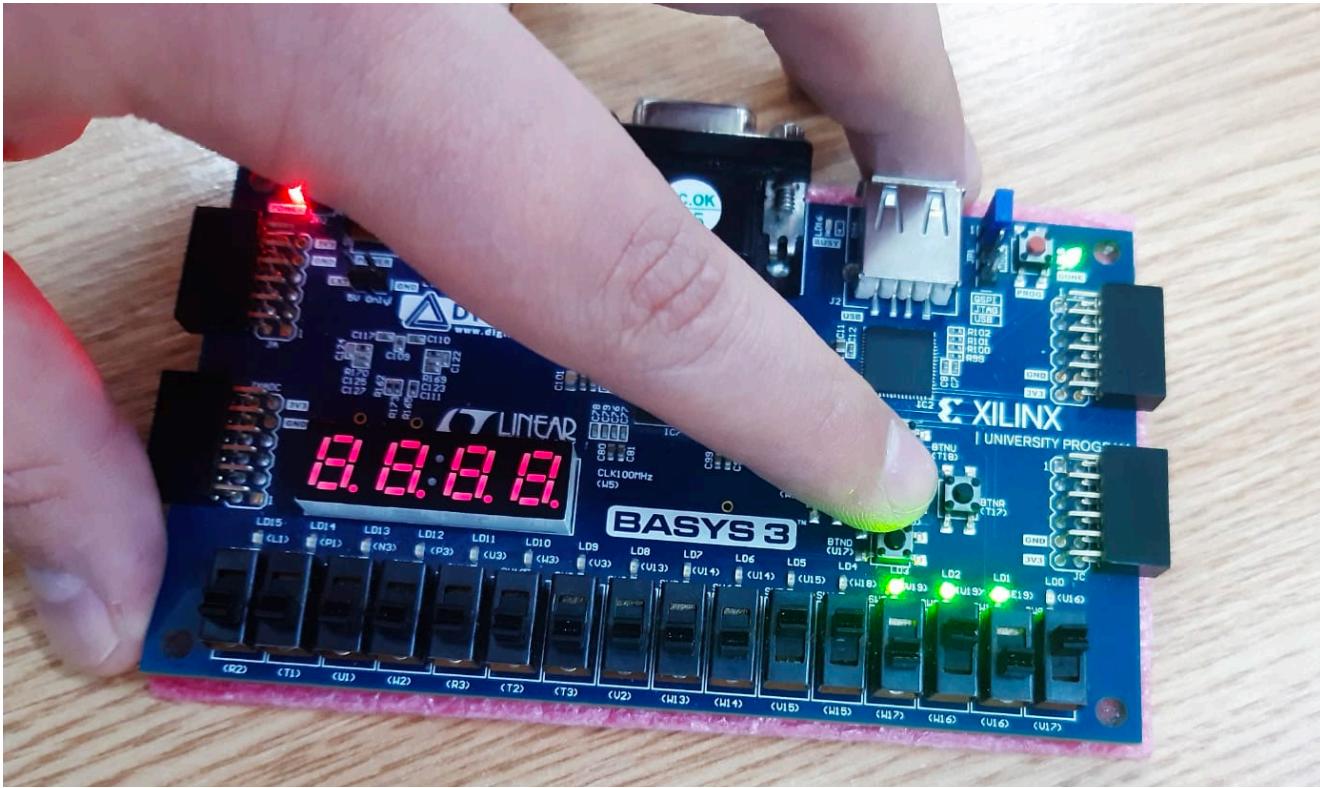


Selectare mod operare

Switch-uri Pt D\_IN

Clock pe buton

Switch dir. de num. BCD



## 9. Concluzii

In concluzie s-a implementat ca proiect un numarator pe 4 biti, cu posibilitate de resetare si de selectare a directiei de numarare si un registru serie, cu posibilitatea de a comuta intre cele doua functionalitati.

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>