



Universitatea Tehnică “Gheorghe Asachi” din Iași



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

ELECTRONICĂ DIGITALĂ

Proiect

Tema: REG paralel – inel – v1

Student:

- Holban Mihnea – Bogdan

Grupa: 1210B

Coordonator:

Asist.Drd. Marius Obreja

2021

1. Specificatiile proiectului:

REG paralel – inel - v1

Să se implementeze în FPGA prin descriere în limbaj VHDL, un sistem secvențial : cu reset prioritar activ pe 0; cu două intrări de selecție din care să se stabilească funcționare de registru paralel, respectiv , registru în inel cu deplasare stânga/dreapta.

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

2. REG paralel – inel - v1

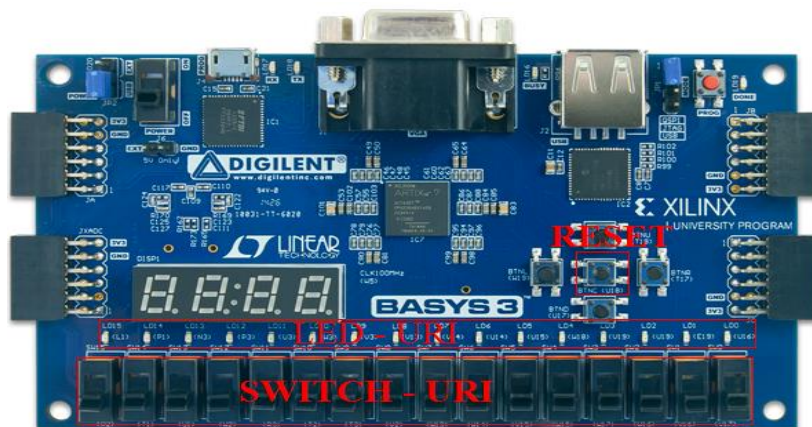
Proiectul contine entitatea REG paralel – inel – v1 in care avem declarate variabile folosite in executarea programului. Structura contine un vectori pentru intrari (D) si inca unul pentru iesiri (Q) de tip bit_vector pe 8 biti, o variabila pentru semnalul de clock si un vector de tip std_logic_vector pentru intrarea de selectie a programului. Rezultatele vor fi afisate pe LED-uri.

Se foloseste o instructiune if pentru a verifica daca clock-ul este pe tactul ascendent (butonul U18 sa fie apasat). In cazul in care conditia este adevarata, se verifica intrarea de selectie, iar pe baza acesteia se decide operatia ce urmeaza a se efectua: resetul activ pe 0, registrul paralel sau registru inel cu deplasare stanga/dreapta.

3. Metoda de implementare

Implementarea s-a realizat pe o placuță BASYS 3 iar ca mediu de programare s-a folosit limbajul VHDL.

Plăcuța BASYS 3:



Librariile utilizate in implementare sunt: **IEEE,IEEE.STD_LOGIC_1164.ALL** și **IEEE.NUMERIC_STD.ALL**

Aceste librării conțin tipuri de date și funcții. Cu ajutorul lor putem declara vectori, constante, variabile de numere întregi, reale, etc.

Exemple de declarații conținute pe care le vom folosi:

- **std_logic**
- **std_logic_vector, bit_vector**
- funcții logice pentru std_logic, std_logic_vector
- funcții de conversie între std_logic_vector și bit_vector

In implementarea problemei s-au folosit urmatoarele:

- 1) CLK- clock-ul circuitului acționat de la unul din butoanele externe ale plăcuței(middle button);
- 2) Switch-uri- 10 Switch-uri dintre care 2 sunt folosite pentru selecția funcționării(reset / registru paralel / registru inel dreapta/ registru inel stânga) și 8 pentru setarea intrării sistemului;
- 3) LED-uri - 8 Led-uri folosite pentru afișare

4. Descrierea (scurta) 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 ocolectie de porturi USB, VGA si altele, placa de dezvoltare BASYS 3 permite proiectarea unor design-uri variate, atat circuite introductorii combinationale, cat si circuite secventiale complexe ca procesoarele si controllerele embedded.

5. Editarea fisierului VHDL

-----reg_par_inel.vhd(TOP MODULE)-----

Entitatea reg_par_inel:

```
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 entity reg_par_inel is
26     Port (
27         D: in bit_vector(7 downto 0);
28         Q: out bit_vector(7 downto 0);
29         clk: in std_logic;
30         s: in std_logic_vector(1 downto 0)
31     );
32 end reg_par_inel;
33
34 architecture Behavioral of reg_par_inel is
35     signal registru: bit_vector(7 downto 0);
36 begin
37     Q<=registru;
38     process (clk)
39     begin
40         if rising_edge(clk) then
41             if s(1) = '0' and s(0) = '0' then --reset activ pe 0
42                 registru<= "00000000";
43             elsif s(1) = '0' and s(0) = '1' then --registru paralel
44                 registru<=D;
45             elsif s(1) = '1' and s(0) = '0' then --registru in inel cu deplasare dreapta
46                 registru <= registru xor 1;
47             elsif s(1) = '1' and s(0) = '1' then --registru in inel cu deplasare stanga
48                 registru <= registru rol 1;
49             end if;
50         end if;
51     end process;
52
53 end Behavioral;
```

6. Editarea fisierului de constrangeri

Switch-uri

```
11  ### Switches
12  set_property PACKAGE_PIN V17 [get_ports {D[0]}]
13      set_property IOSTANDARD LVCMOS33 [get_ports {D[0]}]
14  set_property PACKAGE_PIN V16 [get_ports {D[1]}]
15      set_property IOSTANDARD LVCMOS33 [get_ports {D[1]}]
16  set_property PACKAGE_PIN W16 [get_ports {D[2]}]
17      set_property IOSTANDARD LVCMOS33 [get_ports {D[2]}]
18  set_property PACKAGE_PIN W17 [get_ports {D[3]}]
19      set_property IOSTANDARD LVCMOS33 [get_ports {D[3]}]
20  set_property PACKAGE_PIN W15 [get_ports {D[4]}]
21      set_property IOSTANDARD LVCMOS33 [get_ports {D[4]}]
22  set_property PACKAGE_PIN V15 [get_ports {D[5]}]
23      set_property IOSTANDARD LVCMOS33 [get_ports {D[5]}]
24  set_property PACKAGE_PIN W14 [get_ports {D[6]}]
25      set_property IOSTANDARD LVCMOS33 [get_ports {D[6]}]
26  set_property PACKAGE_PIN W13 [get_ports {D[7]}]
27      set_property IOSTANDARD LVCMOS33 [get_ports {D[7]}]
40  set_property PACKAGE_PIN T1 [get_ports {s[0]}]
41      set_property IOSTANDARD LVCMOS33 [get_ports {s[0]}]
42  set_property PACKAGE_PIN R2 [get_ports {s[1]}]
43      set_property IOSTANDARD LVCMOS33 [get_ports {s[1]}]
```

Led-uri

```
46 | ### LEDs
47 | set_property PACKAGE_PIN U16 [get_ports {Q[0]}]
48 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[0]}]
49 | set_property PACKAGE_PIN E19 [get_ports {Q[1]}]
50 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[1]}]
51 | set_property PACKAGE_PIN U19 [get_ports {Q[2]}]
52 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[2]}]
53 | set_property PACKAGE_PIN V19 [get_ports {Q[3]}]
54 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[3]}]
55 | set_property PACKAGE_PIN W18 [get_ports {Q[4]}]
56 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[4]}]
57 | set_property PACKAGE_PIN U15 [get_ports {Q[5]}]
58 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[5]}]
59 | set_property PACKAGE_PIN U14 [get_ports {Q[6]}]
60 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[6]}]
61 | set_property PACKAGE_PIN V14 [get_ports {Q[7]}]
62 |     set_property IOSTANDARD LVCMOS33 [get_ports {Q[7]}]
```

Butoane

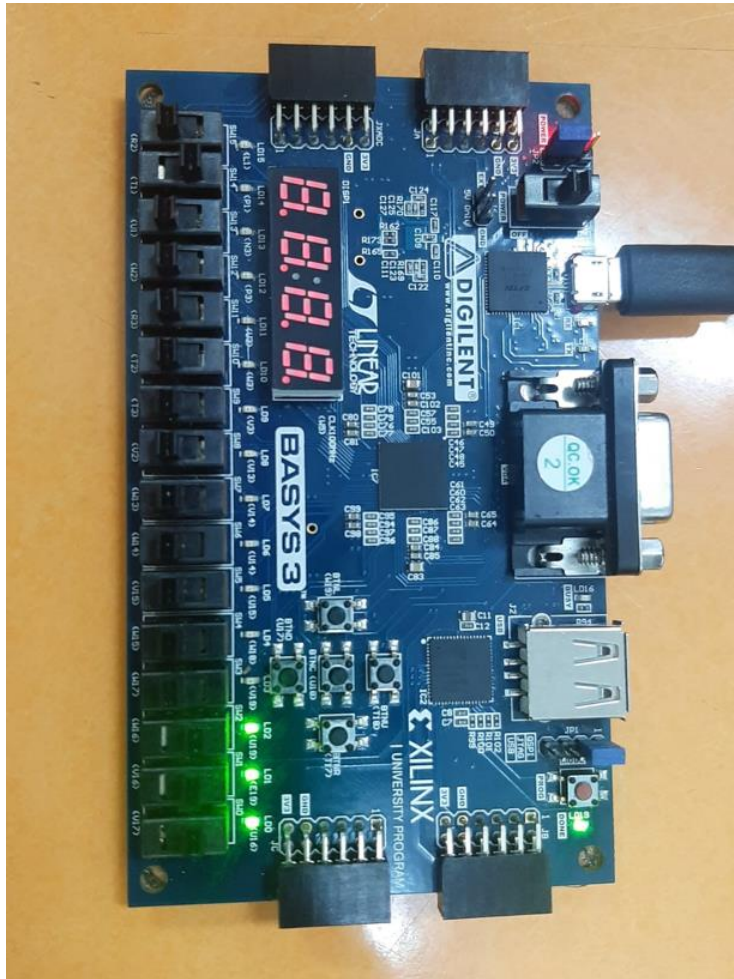
```
110 | ##Buttons
111 | set_property PACKAGE_PIN U18 [get_ports clk]
112 |     set_property IOSTANDARD LVCMOS33 [get_ports clk]
113 |     set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {clk}]
```

7. Descrierea pasilor de sinteza si testarea circuitului rezultat

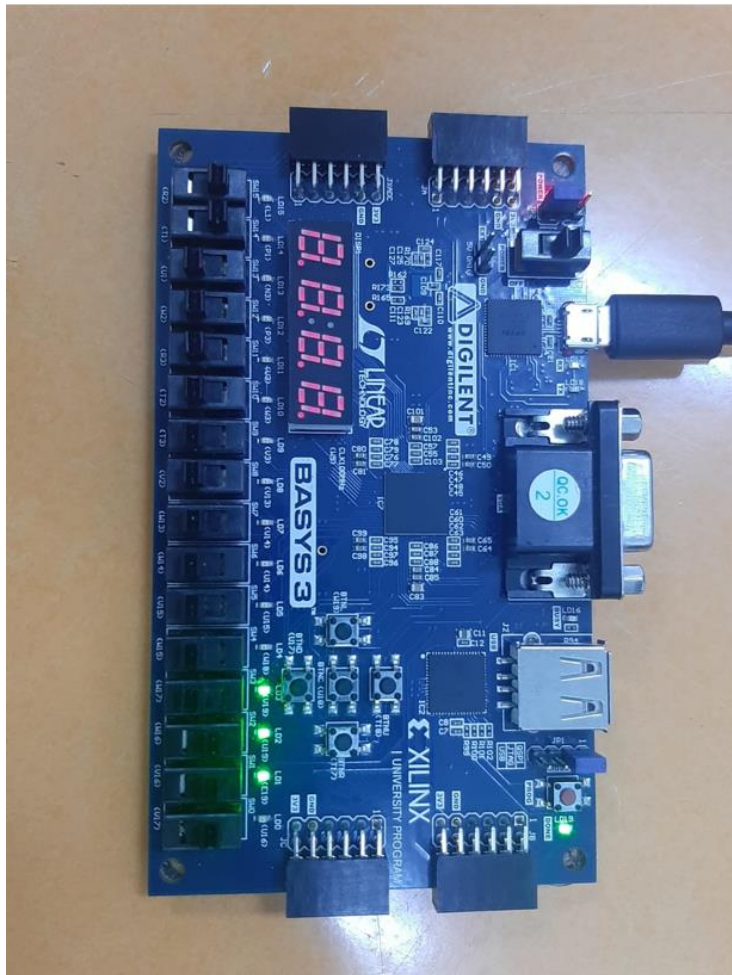
1. S-a creat un proiect nou în programul Vivado
2. S-a implementat modulul “Reg_paralel_ine_v11” printr-o descriere comportamentală
3. S-a editat fișierul de constrângeri în vederea realizării legăturilor între cele 16 switch-uri, led-uri și butoane
4. S-a realizat analiza RTL(Register Transfer Level)
5. S-a sintetizat modulul(pentru se vedea design-ul sintetizat)
6. S-a lansat implementarea proiectului care a avut ca efect final generarea fișierului bitstream
7. S-a programat placa de dezvoltare BASYS 3 cu fișierul bitstream și s-a testat funcționarea corespunzătoare a modulului implementat

8.Fotografii cu functionarea modelului

Registru paralel copiaza informatia data la intrare catre iesire



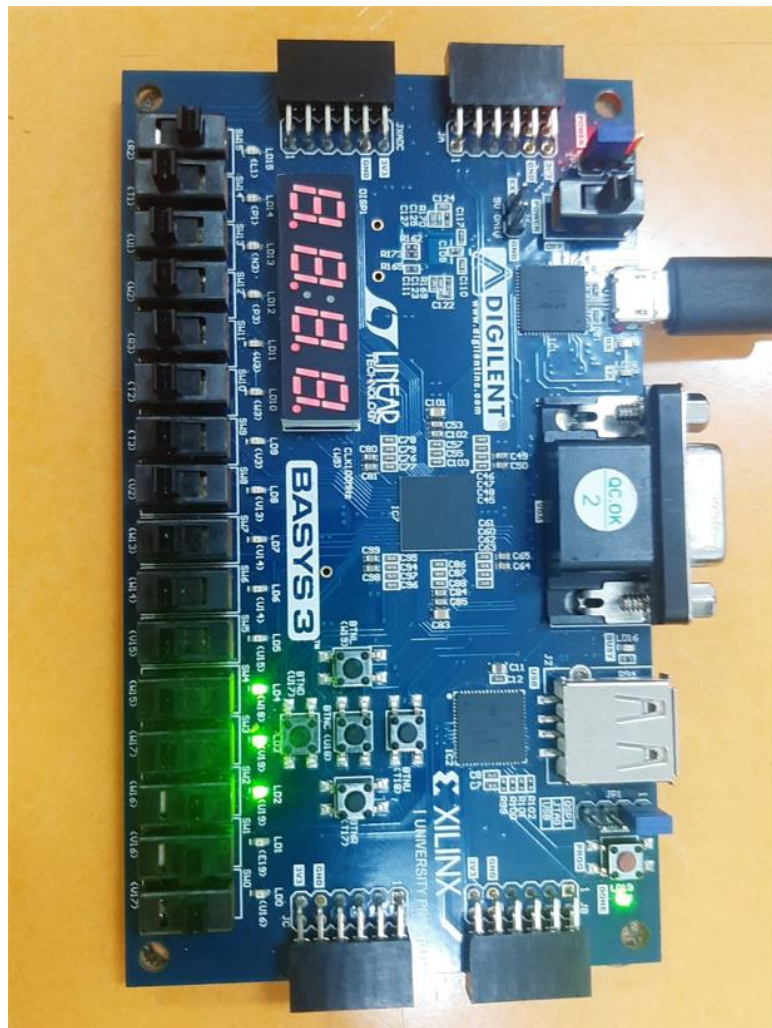
Registrul inel cu deplasare stanga se executa prin apasarea butonului U18 o singura data



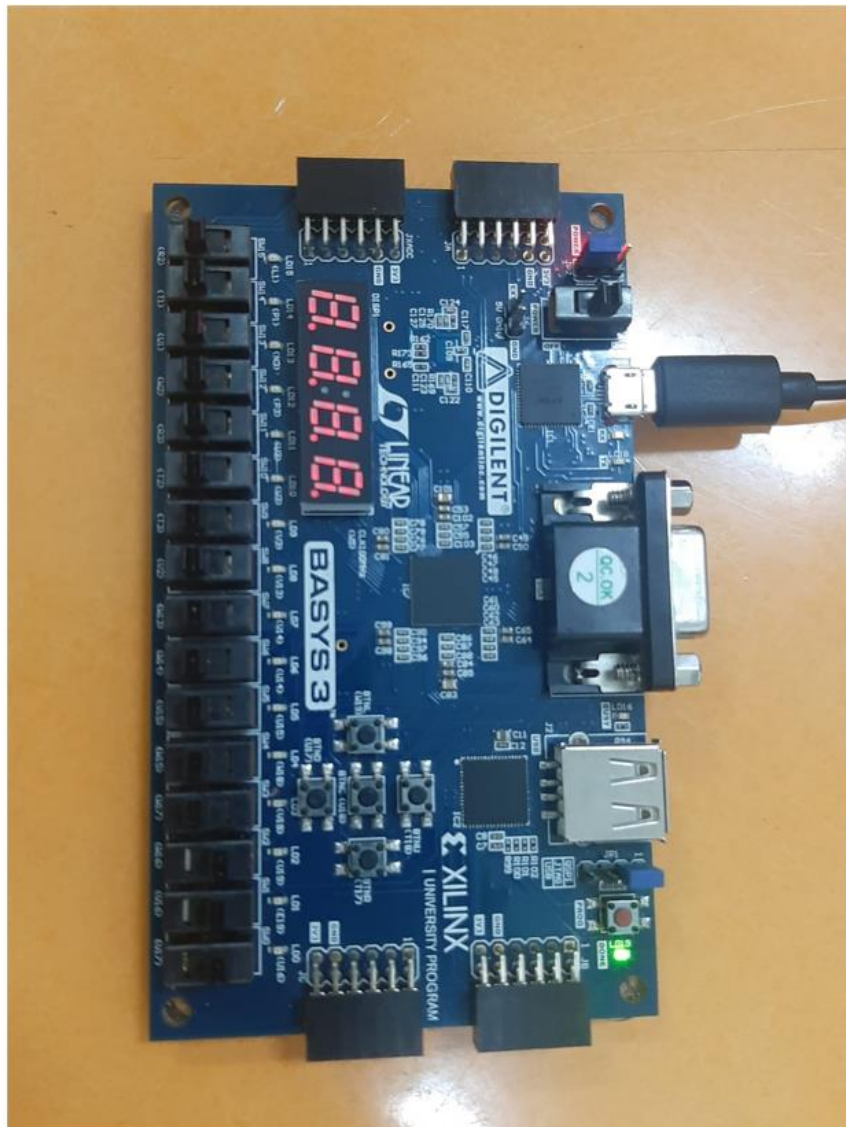
Se apasa butonul U18 de inca 3 ori pentru deplasari la stanga



Se comuta registrul inel pe deplasare dreapta si se apasa butonul de 2 ori



Intrarea de selectie este pe 00 deci avem resetul activ pe 0



9. Concluzii

În concluzie, s-a realizat implementarea unui REG paralel – înel pentru care: registrul paralel copiază informația de la intrare spre ieșire prin intermediul bistabilelor, registrul înel execută deplasări la stânga/dreapta, pe baza intrării de selecție, a numărului deja încărcat prin intermediul registrului paralel, fiecare operație este executată la apăsarea butonului care setează clock-ul pe tactul ascendent, iar afișarea se realizează prin intermediul LED-urilor.

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>