Universitatea Tehnică "Gheorghe Asachi" din Iaşi

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

ELECTRONICĂ DIGITALĂ

proiect

Tema: COMPARATOR 4-biți

Studenţi: Curuliuc Cosmin-Ștefan, Diaconu Rareș-George, Obrocea Traian

Grupa : 1207A

Coordonator:

Asist. Drd. Marius Obreja

2023

1. Specificaţiile proiectului:

COMPARATOR 4-biți

Să se implementeze în FPGA prin descriere în limbaj VHDL, două comparatoare de câte 2 vectori de 4 biţi care să furnizeze la ieşire rezultatul mai mic, egal sau mai mare; unul dintre vectorii de intrare va fi comun la cele două comparatoare;

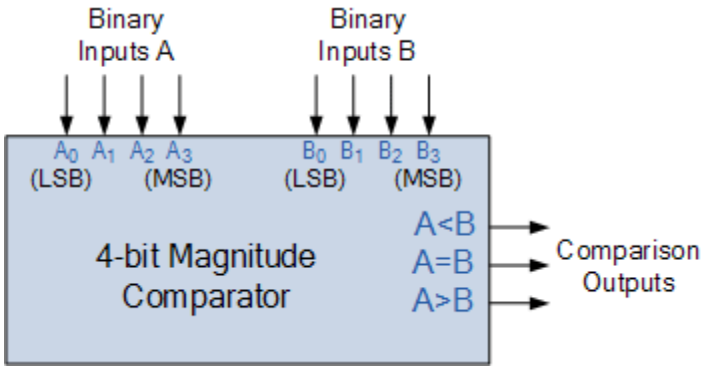
Fişierul bitstream rezultat în urma procesului de implementare va fi verificat utilizând placa de dezvoltare BASYS3.

2. Comparator 4-biți

Comparatoarele numerice permit compararea rapidă a două numere binare A şi B şi determinarea valorii relative a acestora (se determină dacă între cele două numere există una din relaţiile A=B, A>B, A<B).

Un comparator numeric (figura de mai jos) este prevăzut cu:

1. 2*4 = 8 intrări pentru cele 2 numere de 4 biţi;
2. 3 ieşiri cu rezultatul comparaţiei celor 2 numere (A=B, A<B, A>B);



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 comportamentala. S-a proiectat entitatea Proiect. Aceasta are mai multe “procese” și “proceduri”:

1. O procedură pentru un comparator de un singur bit, cu intrări suplimentare pentru înlănțuire.
2. O procedură pentru un comparator de 4-biți, care se folosește de 4 ori de comparatorul pe un singur bit (sunt înlănțuite), și se inițializează două intrări suplimentare, Mare(0) și Mic(0) cu valoarea 0.
3. Două procese, unul pentru primii 2 vectori x și y, astfel încât unul dintre primele 3 becuri(U16, E19, U19) se va aprinde în funcție de rezultatul comparării. Al doilea proces compară vectorii x și z, astfel încât unul dintre celelalte 3 becuri (V19, W18, U15) se va aprinde în funcție de rezultatul comparării.
4.

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 introductorii combinationale, cat si circuite secventiale complexe ca procesoarele si controllerele embedded.

5. Editarea fişierului VHDL

----------Proiect.vhd(TOP MODULE)----------

Entitatea Proiect:

```
entity Proiect is
    port(
        x: in std_logic_vector(3 downto 0);
        y: in std_logic_vector(3 downto 0);
        z: in std_logic_vector(3 downto 0);
        ma_1: out std_logic;
        eg_1: out std_logic;
        mi_1: out std_logic;
        ma_2: out std_logic;
        eg_2: out std_logic;
        mi_2: out std_logic
        );
end Proiect;
```

Procedură comparator pe un bit:

```
architecture Behavioral of Proiect is
    procedure comp1bit (
                    x: in std_logic;
                    y: in std_logic;
                    Mare_In: in std_logic;
                    Mic_In: in std_logic;
                    Mare_Out: out std_logic;
                    Egal_Out: out std_logic;
                    Mic_Out: out std_logic) is

    begin
        Mare_Out := (x and not y) or (x and Mare_In) or (not y and Mare_In);
        Egal_Out := not (x xor y) and not Mare_In and not Mic_In;
        Mic_Out := (not x and y) or (not x and Mic_In) or (y and Mic_In);
    end procedure;
```

Procedură comparator pe 4-biți:

```
procedure comp4bit (
                x: in std_logic_vector(3 downto 0);
                y: in std_logic_vector(3 downto 0);
               signal ma: out std_logic;
               signal eg: out std_logic;
               signal mi: out std_logic) is
                variable Mare, Mic, Egal: STD_LOGIC_VECTOR(4 downto 0);
begin
    Mare(0) := '0';
    Mic(0) := '0';
    for i in 0 to 3 loop
        comp1bit(x(i), y(i), Mare(i), Mic(i), Mare(i+1), Mic(i+1), Egal(i+1));
    end loop;
    ma <= Mare(4);
    eg <= Egal(4);
    mi <= Mic(4);
end procedure;
```

Procesele pentru cele două comparatoare:

```
begin
    process(x, y)
        begin
        comp4bit(x, y, ma_1, eg_1, mi_1);
    end process;

    process(x, z)
        begin
        comp4bit(x, z, ma_2, eg_2, mi_2);
    end process;
end Behavioral;
```

6. Editarea fişierului de constrângeri

Switch-uri:

```
##Switches
set_property PACKAGE_PIN V17 [get_ports {x[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {x[0]}]
set_property PACKAGE_PIN V16 [get_ports {x[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {x[1]}]
set_property PACKAGE_PIN W16 [get_ports {x[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {x[2]}]
set_property PACKAGE_PIN W17 [get_ports {x[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {x[3]}]
set_property PACKAGE_PIN W15 [get_ports {y[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {y[0]}]
set_property PACKAGE_PIN V15 [get_ports {y[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {y[1]}]
set_property PACKAGE_PIN W14 [get_ports {y[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {y[2]}]
set_property PACKAGE_PIN W13 [get_ports {y[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {y[3]}]
set_property PACKAGE_PIN V2 [get_ports {z[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {z[0]}]
set_property PACKAGE_PIN T3 [get_ports {z[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {z[1]}]
set_property PACKAGE_PIN T2 [get_ports {z[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {z[2]}]
set_property PACKAGE_PIN R3 [get_ports {z[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {z[3]}]
```

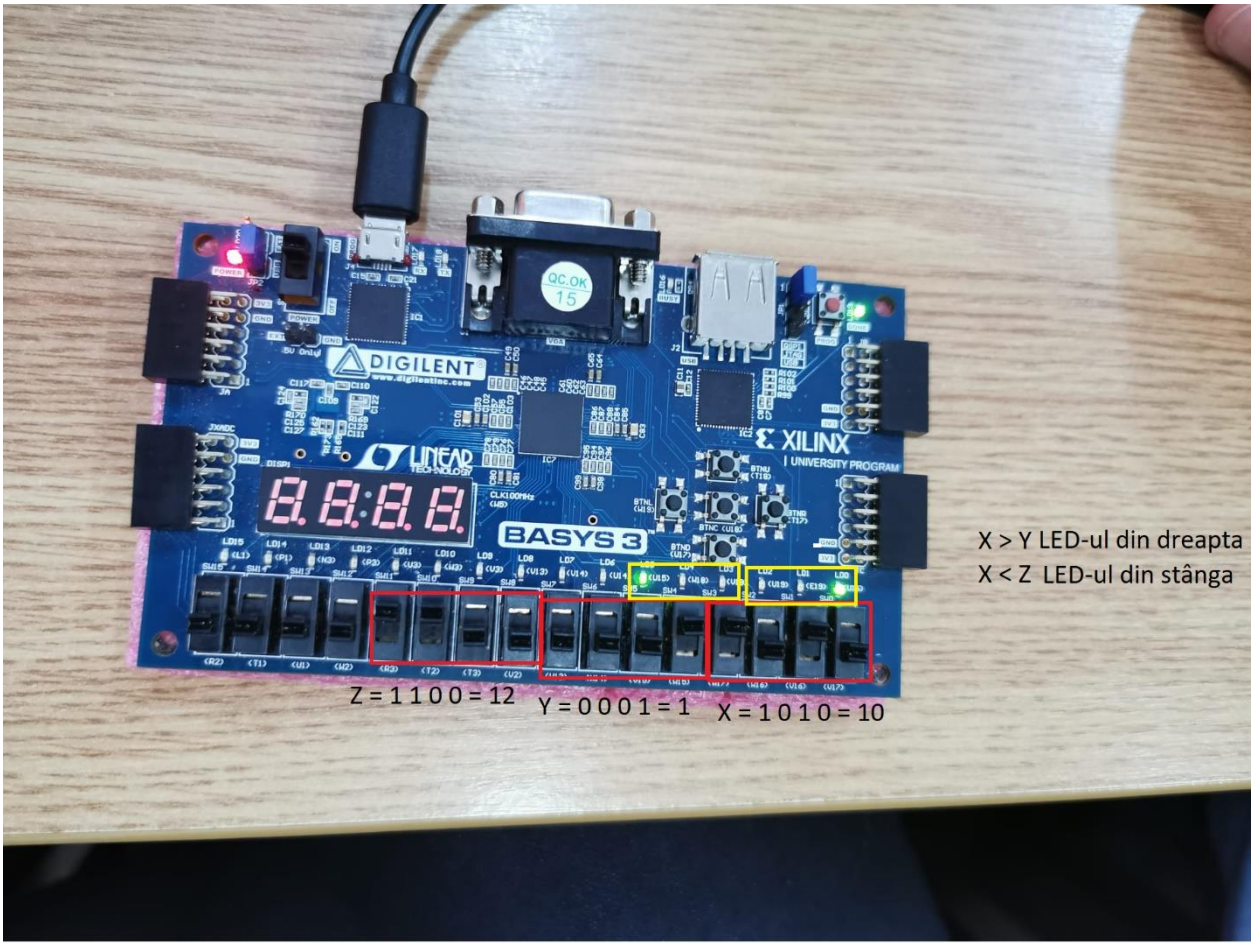
LED-uri:

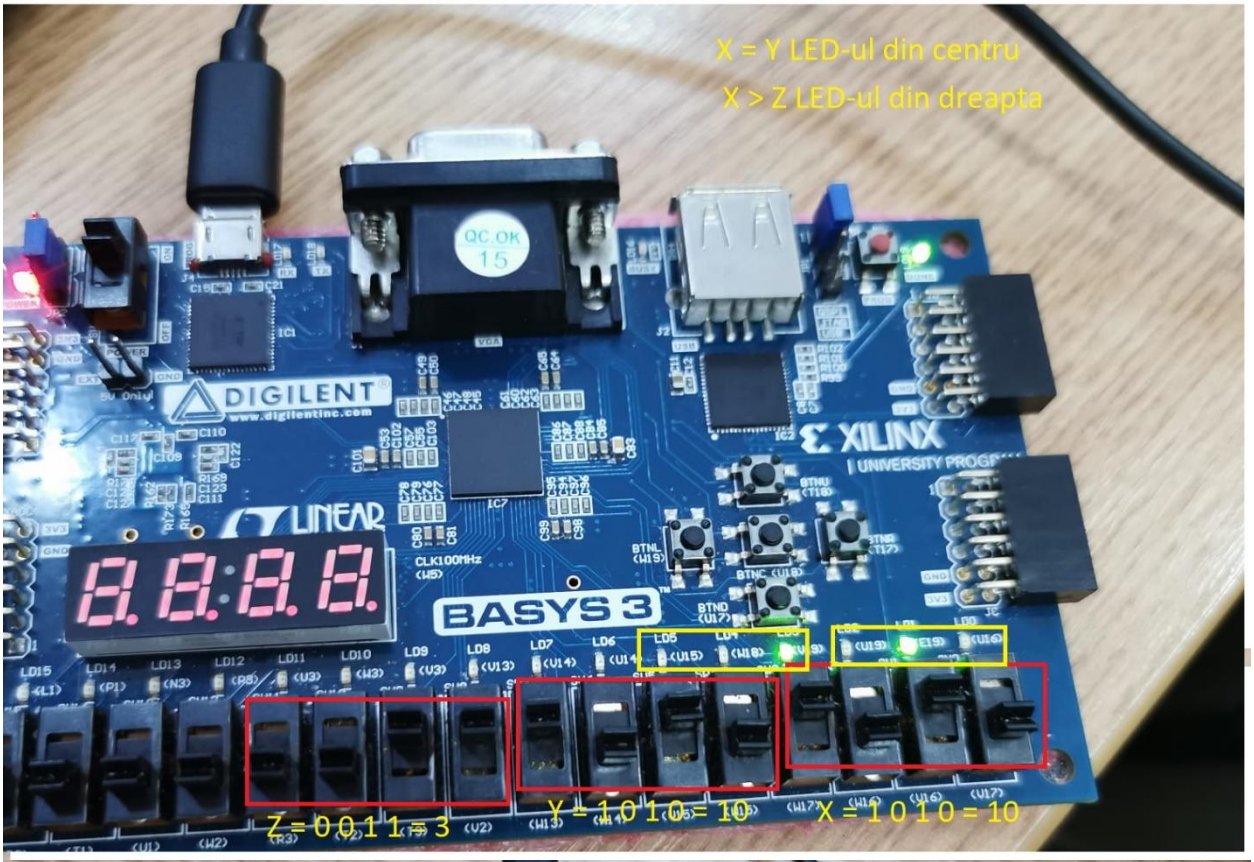
```
## LEDs
set_property PACKAGE_PIN U16 [get_ports ma_1]
    set_property IOSTANDARD LVCMOS33 [get_ports ma_1]
set_property PACKAGE_PIN E19 [get_ports eg_1]
    set_property IOSTANDARD LVCMOS33 [get_ports eg_1]
set_property PACKAGE_PIN U19 [get_ports mi_1]
    set_property IOSTANDARD LVCMOS33 [get_ports mi_1]
set_property PACKAGE_PIN V19 [get_ports ma_2]
    set_property IOSTANDARD LVCMOS33 [get_ports ma_2]
set_property PACKAGE_PIN W18 [get_ports eg_2]
    set_property IOSTANDARD LVCMOS33 [get_ports eg_2]
set_property PACKAGE_PIN U15 [get_ports mi_2]
    set_property IOSTANDARD LVCMOS33 [get_ports mi_2]
```

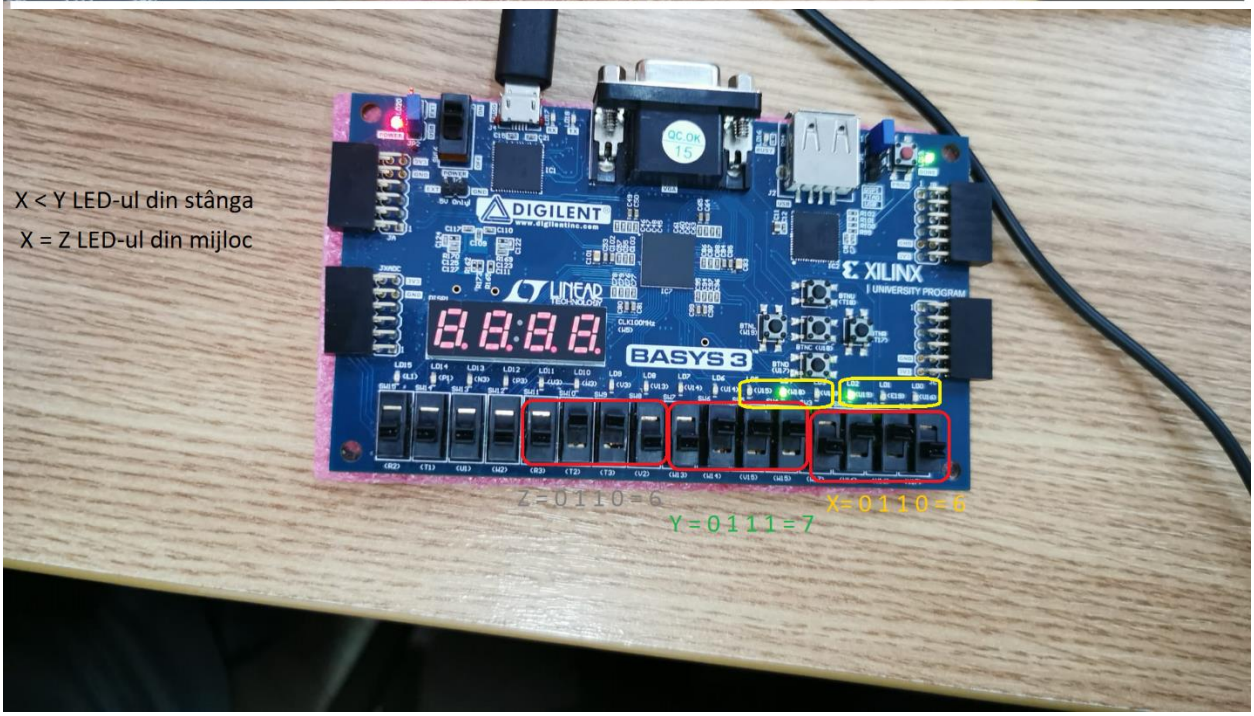
7. Descrierea paşilor de sinteză şi testarea circuitului rezultat

1. S-a creeat un proiect nou in programul Vivado
2. S-a implementat modulul “Proiect” printr-o descriere comportamentala(avand drept “proceduri“ comparatoarele de un bit, respectiv 4-biți, și procese pentru fiecare comparator în parte).
3. S-a editat fisierul de constrangeri in vederea realizarii legaturilor intre switch-uri si intrari(x,y,z), 6 led-uri și ieșiri (ma_1, eg_1, mi_1, ma_2, eg_2, mi_2).
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:




X = Y LED-ul din centru
X > Z LED-ul din dreapta
DIGILENT
XILINX
LINEAR
BASYS 3
Z = 0 0 1 1 = 3
Y = 1 0 1 0 = 10
X = 1 0 1 0 = 10


X < Y LED-ul din stânga
X = Z LED-ul din mijloc
DIGILENT
XILINX
LINEAR
BASYS 3
Z=0110=6
Y=0111=7
X=0110=6

9. Concluzii

In concluzie s-a implementat ca proiect 2 comparatoare pe 4 biți cu 3 vectori, în care se compară simultan vectorul x prima oară cu vectorul y, respectiv a doua oară cu vectorul z.

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

3. https://www.deldsim.com/study/material/15/implementation-of-4-bit-magnitude-comparator-using-ic-74ls85/ (diagrama comparator)