



Universitatea Tehnică “Gheorghe Asachi” din Iași



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

ELECTRONICĂ DIGITALĂ

Proiect

Tema: COUNTER BINAR-GRAY

Studenti:

1. Șalaru Ioana
2. Lefter Alesia
3. Chira Ioana

Grupa : 1210B

Coordonator:

Asist. Drd. Ionica Pletea

2024

Tema proiectului:

COUNTER BINAR-GRAY – v2

1. Specificațiile proiectului:

COUNTER BINAR-GRAY – v2

Să se implementeze în FPGA prin descriere în limbaj VHDL, un numărător de 4 biți, care prin intrare de selecție să poată să funcționeze fie în cod BCD, fie în cod Gray. Afisarea sa se faca pe leduri. Frecventa sa fie de 1Hz.

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

2. Modulul COUNTER BINAR-GRAY – v2

Modulul implementat este un numărător pe 4 biți cu funcționalități extinse care includ:

Specificații:

1. Tipuri de numărare:

-Numărare în cod BCD (binar zecimal).

-Numărare în cod Gray.

2. Semnal de control:

-Clock intern (frecvența standard a plăcuței: 10^8 Hz) este redus la o frecvență de 1 Hz utilizând un divizor de frecvență.

3. Direcția de numărare:

Numărătorul poate funcționa în două direcții:

-Incrementare (1).

-Decrementare (0).

4. Alte funcționalități :

Reset: Valoarea numărătorului poate fi resetată la 0.

Selecție mod: ieșirea poate fi afișată fie în format BCD, fie în format Gray.

Intrările modulului:

1. counter_clk: Semnal de clock extern (de pe plăcuță), utilizat pentru divizorul de frecvență.

2. count_direction: Semnal de control care determină direcția numărătorului (1 = incrementare, 0 = decrementare).
3. mode: Selectează modul de afișare (0 = BCD, 1 = Gray).
4. reset_count: Semnal de reset (activ pe nivel logic 1).

Ieșirile modului:

data_out: Vector pe 4 biți utilizat pentru afișarea valorii curente pe LED-uri. = Valoarea curentă a numărătorului, afișată fie în format BCD, fie în format Gray.

Semnalele interne:

1. bcd_value: Vector pe 4 biți care stochează valoarea curentă în cod BCD.
2. gray_value: Vector pe 4 biți care stochează valoarea curentă în cod Gray.
3. clk: Clock divizat, redus la o frecvență de 1 Hz pentru numărătoare.

Funcționalitatea modului:

1. Divizor de frecvență:

-Clock-ul de pe plăcuță este redus de la 10^8 Hz la 1 Hz utilizând un contor intern.

-Semnalul rezultat este utilizat ca clock pentru procesul de numărătoare.

2. Numărător în cod BCD:

-Valoarea este incrementată/decrementată la fiecare front pozitiv al clock-ului divizat.

-Se asigură tranziția între valorile 0 și 9.

3. Convertor BCD → Gray:

-La fiecare modificare a valorii în cod BCD, se calculează valoarea corespunzătoare în cod Gray utilizând expresiile logice:

$$\text{gray_value}(0) = \text{bcd_value}(1) \oplus \text{bcd_value}(0)$$

$$\text{gray_value}(1) = \text{bcd_value}(2) \oplus \text{bcd_value}(1)$$

$$\text{gray_value}(2) = \text{bcd_value}(2) \oplus \text{bcd_value}(3)$$

$$\text{gray_value}(3) = \text{bcd_value}(3)$$

4. Selecția modului de afișare:

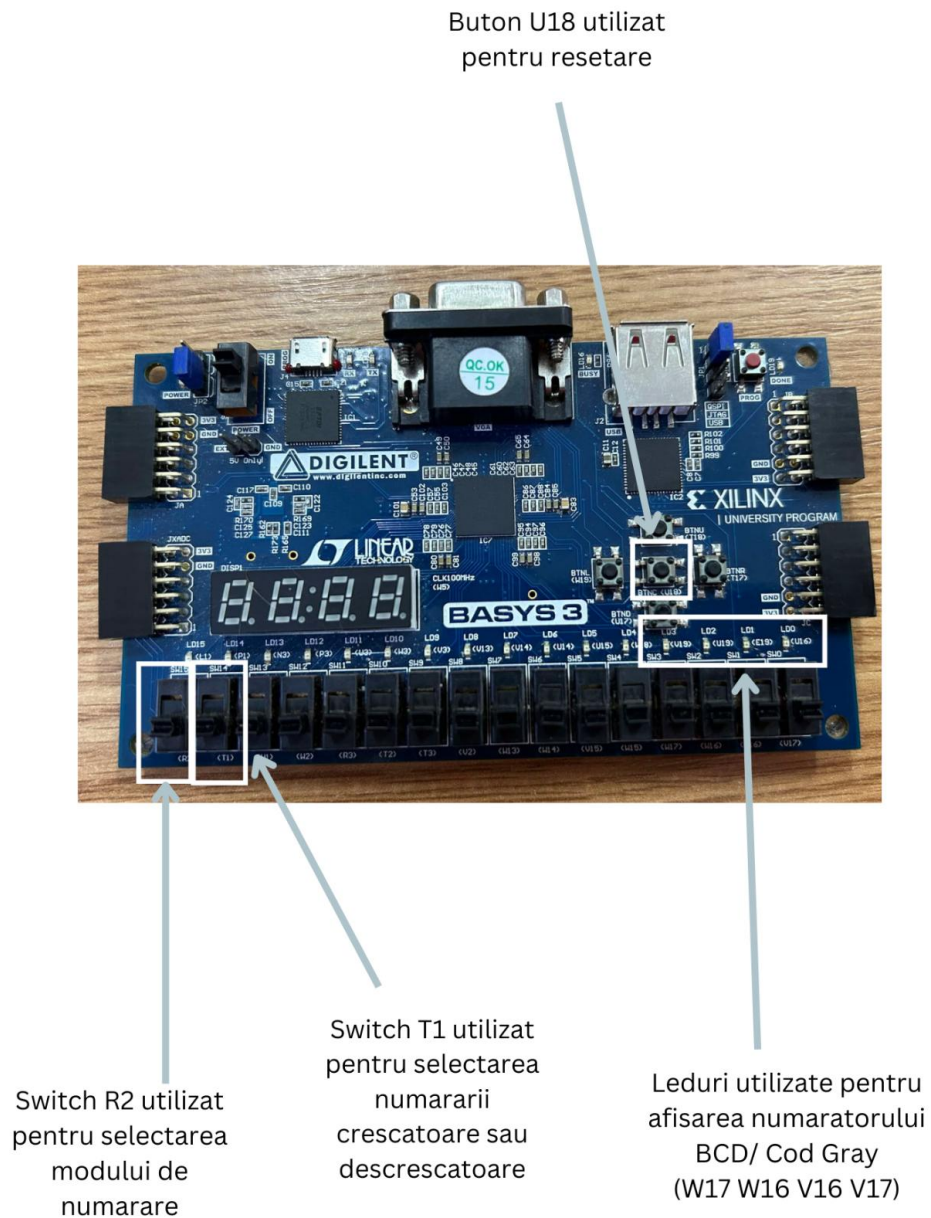
-Dacă mode = 0, ieșirea este afișată în cod BCD.

-Dacă mode = 1, ieșirea este afișată în cod Gray.

5. Reset:

-Dacă semnalul reset_count este activ (1 logic) , numărătorul este resetat la 0.

Legenda:



3. Metoda de implementare

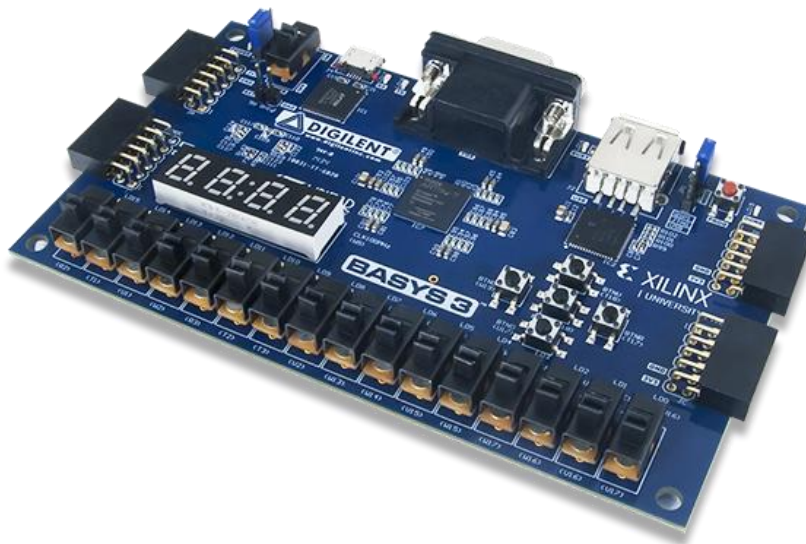
Utilizarea resurselor: circuit FPGA, limbajul VHDL, programul de sinteză Vivado

Implementarea proiectului a fost făcută printr-o descriere comportamentală. S-a proiectat entitatea „numarator”. Aceasta are mai multe „procese”:

- procesul de reducere a frecvenței de la 10^8 Hz la 1 Hz
- procesul de numărare în BCD, convertit ulterior în Gray
- procesul de selecție a modului de afișare

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

BASYS3 este o placă de dezvoltare FPGA (Field-Programmable Gate Array) produsă de Digilent. Placa BASYS3 conține un cip FPGA XILINX Artix-7, care oferă capacități de logică programabilă și resurse pentru implementarea diverselor proiecte digitale. Aceasta dispune de o gamă diversificată de resurse I/O, inclusiv butoane, LED-uri, conectori Pmod (pentru extensii modulare), conectori pentru afișaj cu șapte segmente, porturi USB, VGA și altele. Sistemul de dezvoltare este compatibil cu XILINX Vivado, un mediu integrat de dezvoltare (IDE) folosit pentru sinteza, implementarea și programarea FPGA-urilor și permite integrarea unui procesor MicroBlaze, care oferă o soluție de procesare soft-core. Aceste resurse facilitează interacțiunea cu mediul exterior și permite testarea proiectelor, fiind foarte accesibilă atât pentru studenți cât și pentru profesioniști.



5. Editarea fișierului VHDL

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```

entity numarator is
    Port(
        counter_clk : in std_logic;
-- Semnal de clock de pe placa cu frecvența 10^8 Hz
        count_direction : in std_logic;
-- Direcția de numărare (1: incrementare, 0: decrementare)
        mode : in std_logic;
-- Mod de numărare (1: Gray, 0: BCD)
        reset_count : in std_logic;
-- Resetare (1: resetează, 0: continuă)
        data_out : out std_logic_vector(3 downto 0)
-- Datele de ieșire pe --LED [3:0]
    );
end numarator;

architecture Behavioral of numarator is
    signal bcd_value : std_logic_vector(3 downto 0) := (others => '0');
--Valoarea în BCD
    signal gray_value : std_logic_vector(3 downto 0);
-- Valoarea în Gray
    signal clk : std_logic := '0';
-- Clock cu frecvență redusă la 1 Hz
    signal counter : integer range 0 to 100000000 := 0;
-- Divizor de frecvență
begin
    -- Divizor de frecvență
    process(counter_clk)
    begin
        if rising_edge(counter_clk) then
            if counter < 100000000 - 1 then
                counter <= counter + 1;
            else
                counter <= 0;
                clk <= not clk; -- Generează un clock cu perioadă de 1 secundă
            end if;
        end if;
    end process;

    -- Proces pentru numărare
    process(clk, reset_count, count_direction)
    begin
        if reset_count = '1' then
            bcd_value <= (others => '0'); -- Resetează numărătorul
        elsif rising_edge(clk) then
            if count_direction = '1' then -- Incrementare

```

```

        if bcd_value = "1001" then
            bcd_value <= (others => '0');
        else
            bcd_value <= bcd_value + 1;
        end if;
    else -- Decrementare
        if bcd_value = "0000" then
            bcd_value <= "1001";
        else
            bcd_value <= bcd_value - 1;
        end if;
    end if;
end if;
end process;

```

-- Conversie în cod Gray

```

gray_value(0) <= bcd_value(1) xor bcd_value(0);
gray_value(1) <= bcd_value(2) xor bcd_value(1);
gray_value(2) <= bcd_value(3) xor bcd_value(2);
gray_value(3) <= bcd_value(3);

```

-- Selectarea modului de afișare (BCD sau Gray)

```

process(mode, bcd_value, gray_value)
begin
    if mode = '0' then
        data_out <= bcd_value;
    else
        data_out <= gray_value;
    end if;
end process;

```

end Behavioral;

6. Editarea fișierului de constrângeri

Clock signal

set_property PACKAGE_PIN W5 [get_ports counter_clk]

set_property IOSTANDARD LVCMOS33 [get_ports counter_clk]

Switches

```

set_property PACKAGE_PIN T1 [get_ports {count_direction}]
    set_property IOSTANDARD LVCMOS33 [get_ports {count_direction}]
set_property PACKAGE_PIN R2 [get_ports {mode}]
    set_property IOSTANDARD LVCMOS33 [get_ports {mode}]
## LEDs
set_property PACKAGE_PIN U16 [get_ports {data_out[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {data_out[0]}]
set_property PACKAGE_PIN E19 [get_ports {data_out[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {data_out[1]}]
set_property PACKAGE_PIN U19 [get_ports {data_out[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {data_out[2]}]
set_property PACKAGE_PIN V19 [get_ports {data_out[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {data_out[3]}]
##Buttons
set_property PACKAGE_PIN U18 [get_ports reset_count]
    set_property IOSTANDARD LVCMOS33 [get_ports reset_count]

```

7. Descrierea pașilor de sinteză și testarea circuitului rezultat

- deschidere Vivado și crearea unui proiect nou
- crearea unui fișier cod sursă în VHDL în care a fost inclusă implementarea descrierii comportamentale, proiectarea entității numărător și a proceselor
- actualizarea fișierului de constrangeri BASYS3 pentru a atribui butonului, LED-urilor și switch-urilor funcționalitățile corespunzătoare
- realizarea sintezei pentru a obține design-ul sintetizat
- aplicarea implementării design-ului ce creează un fișier bitstream utilizat în programarea plăcuței xc7a35tcbg236-1
- încărcarea design-ului și programarea plăcuței
- testarea plăcuței pentru a asigura funcționarea corectă a funcționalităților

8. Concluzii

Proiectul numărătorului Binar-Gray pe 4 biți, implementat în VHDL pe placa BASYS3, a fost realizat cu succes. Obiectivele au fost îndeplinite, incluzând implementarea modurilor BCD și Gray, reducerea frecvenței la 1 Hz și numărarea în direcții crescătoare și descrescătoare. Testarea pe placa BASYS3 a confirmat funcționarea corectă a sistemului, iar divizorul de frecvență și conversia BCD → Gray au fost implementate corect. Proiectul a oferit o bună oportunitate de învățare a utilizării FPGA și VHDL. Posibile îmbunătățiri ar include optimizarea resurselor și extinderea numărătorului.

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>