PROIECT
LABORATOR
CIRCUITE INTEGRATE DIGITALE

student: ADRIAN ALEXANDRU BALUSA

SERIA A

GRUPA 2123

îndrumător: LAURA IOANA MIHĂILĂ

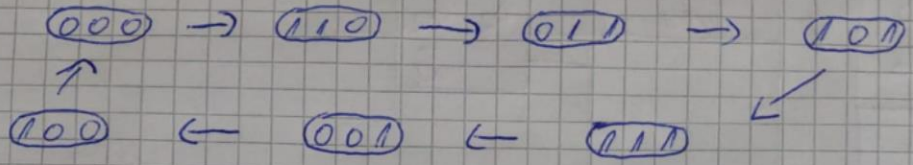
CUPRINS

1. Rezolvarea corectă a temei de proiect pe hârtie
2. Circuitul combinational (Verificarea funcționalității circuitului)
3. Circuitul secvențial (Verificarea funcționalității circuitului)
4. Implementarea finală (Arhitectura se va descrie structural) (Verificarea funcționalității automatului)
5. Aspect documentație (foi de capăt, pagini numerotate etc.)

I. Întrebările legate de proiect se pot pune numai la sfârșitul orei de laborator
II. Simulările trebuie să scoată în evidență funcționalitatea circuitului conform tabelului de adevăr
III. Documentația este un document PDF care va conține cerințele de mai sus, cât și rezolvarea pe hârtie a automatului, capturi de ecran la sursele de proiectare, sursele de simulare, rezultatele simulărilor și explicații
IV. Se va realiza o arhivă cu proiectul final din Vivado
V. Documentul PDF și arhiva cu proiectul final se vor trimite pe Teams profesorului de laborator până la data de 10.01.2023ora 23:59 (Nerespectarea termenului limită duce la scăderea punctajului cu 2p)
VI. Proiectul trebuie susținut personal, iar răspunsurile la întrebări vor influența nota finală

Rezolvare

→ automat de tranziții:

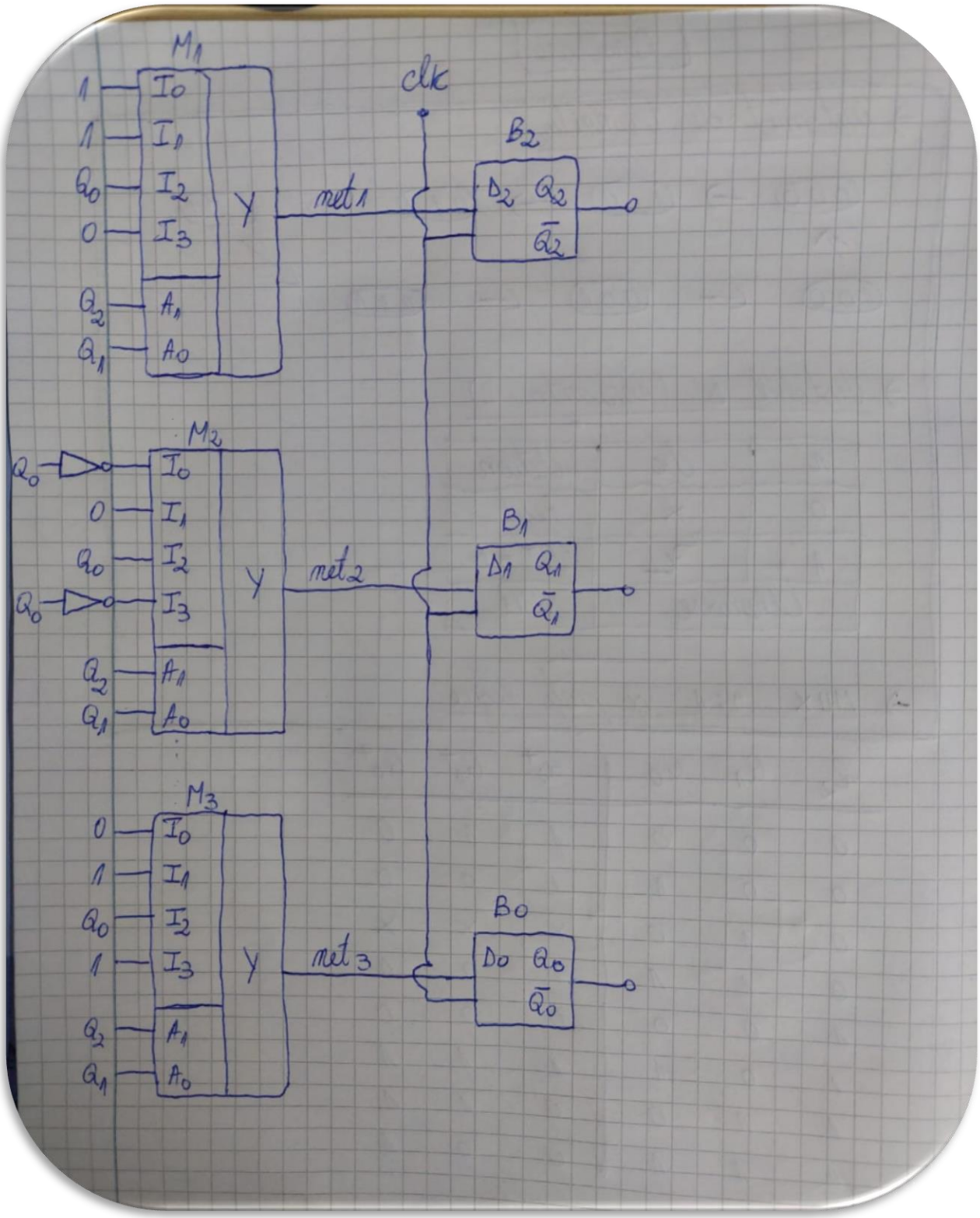


→ bistabil D (varianta C)

R	clk	Action
0	X	Reset
1	↓	$Q^+ = D$
Otherwise		Wait

→ MUX 4:1 și porți logice

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	1	1	0
0	0	1	1	0	0
0	1	0	X	X	X
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	1	0	0	1


M1
1
1
Q0
0
Q2
Q1
I0
I1
I2
I3
A1
A0
Y
net1
clk
B2
D2
Q2
Q̄2
M2
Q0
0
Q0
Q2
Q1
net2
B1
D1
Q1
Q̄1
M3
0
1
Q0
1
Q2
Q1
net3
B0
D0
Q0
Q̄0

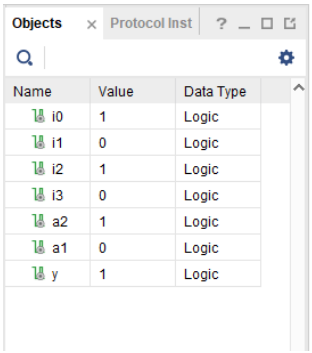
Sursa de design pentru Mux 4:1 si sursa de simulare a acestuia

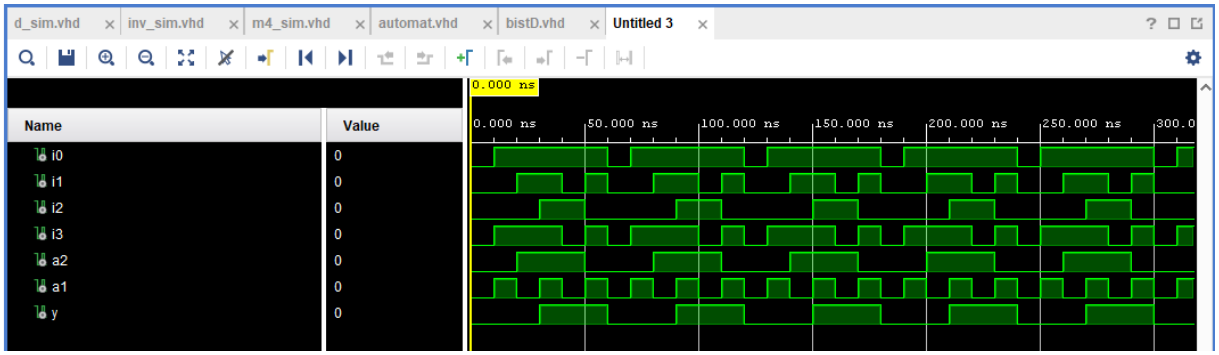
```vhdl
entity mux4 is
    Port ( i0 : in STD_LOGIC;
           i1 : in STD_LOGIC;
           i2 : in STD_LOGIC;
           i3 : in STD_LOGIC;
           a2 : in STD_LOGIC;
           a1 : in STD_LOGIC;
           y : out STD_LOGIC);
end mux4;

architecture Behavioral of mux4 is

signal adr : std_logic_vector(1 downto 0);

begin

adr <= a2 & a1;

  with adr select
    y <= i0 when "00",
         i1 when "01",
         i2 when "10",
         i3 when "11",
         i0 when others;

end Behavioral;
```

```vhdl
entity m4_sim is
    --  Port ( );
end m4_sim;

architecture Behavioral of m4_sim is
    signal i0: std_logic;
    signal i1: std_logic;
    signal i2: std_logic;
    signal i3: std_logic;
    signal a2: std_logic;
    signal a1: std_logic;
    signal y: std_logic;
begin

    dut: entity work.mux4
        port map (
            i0 => i0,
            i1 => i1,
            i2 => i2,
            i3 => i3,
            a2 => a2,
            a1 => a1,
            y => y
        );
```

```vhdl
process
begin

    i0 <= '0';
    i1 <= '0';
    i2 <= '0';
    i3 <= '0';
    a2 <= '0';
    a1 <= '0';
    wait for 10 ns;

    i0 <= '1';
    i1 <= '0';
    i2 <= '0';
    i3 <= '1';
    a2 <= '0';
    a1 <= '1';
    wait for 10 ns;

    i0 <= '1';
    i1 <= '1';
    i2 <= '0';
    i3 <= '1';
    a2 <= '1';
    a1 <= '0';
    wait for 10 ns;
```

```vhdl
    i0 <= '1';
    i1 <= '1';
    i2 <= '1';
    i3 <= '1';
    a2 <= '1';
    a1 <= '1';
    wait for 10 ns;

    i0 <= '1';
    i1 <= '0';
    i2 <= '1';
    i3 <= '0';
    a2 <= '1';
    a1 <= '0';
    wait for 10 ns;

    i0 <= '1';
    i1 <= '1';
    i2 <= '0';
    i3 <= '1';
    a2 <= '0';
    a1 <= '1';
    wait for 10 ns;

end process;

end Behavioral;
```



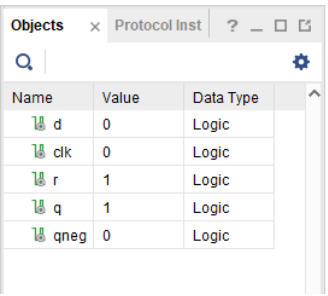


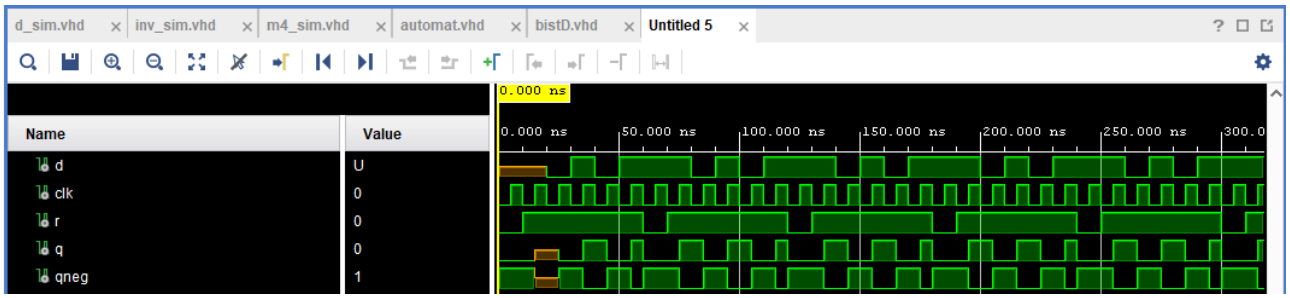
Sursa de design si sursa de simulare pentru bistabilul D

```vhdl
entity bistD is
    Port ( clk : in STD_LOGIC;
           d : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC := '0';
           qneg : out STD_LOGIC := '1');
end bistD;

architecture Behavioral of bistD is
signal qaux: std_logic;

begin

q <= qaux;
qneg <= not qaux;
process(r, clk)
begin
    if r = '0' then
        qaux <= '0';
    elsif falling_edge (clk) then
        qaux <= d;
    else
        qaux <= qaux;
    end if;
end process;

end Behavioral;
```

```vhdl
entity d_sim is
--  Port ( );
end d_sim;

architecture Behavioral of d_sim is
signal d: std_logic;
    signal clk: std_logic;
    signal r: std_logic;
    signal q: std_logic;
    signal qneg: std_logic;

begin

T: entity work.bistD port map (
            d => d,
            clk => clk,
            r => r,
            q => q,
            qneg => qneg
        );

clk_gen: process
    begin
        clk <= '0';
        wait for 5 ns;
        clk <= '1';
        wait for 5 ns;
    end process;
```

```vhdl
stim: process
    begin

        r <= '0';
        wait for 10 ns;
        r <= '1';
        wait for 10 ns;

        d <= '0';
        wait for 10 ns;
        d <= '1';
        wait for 10 ns;
        d <= '0';
        wait for 10 ns;
        d <= '1';
        wait for 10 ns;

    end process;

end Behavioral;
```

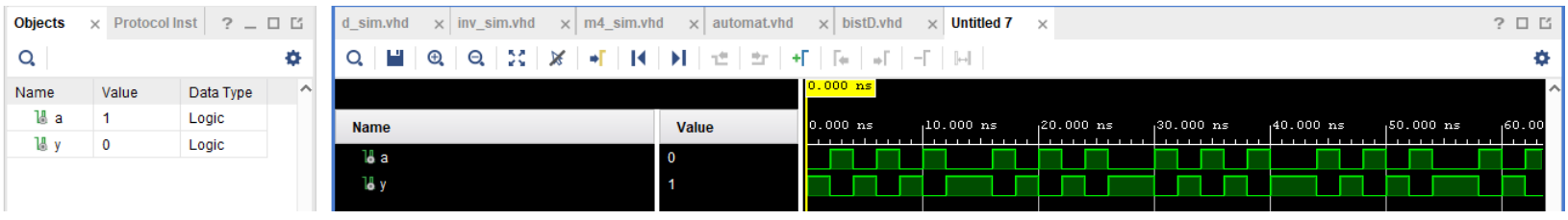




Sursa de design și sursa de simulare a inversorului

```vhdl
entity inversor is
    Port ( a : in STD_LOGIC;
           y : out STD_LOGIC);
end inversor;

architecture Behavioral of inversor is

begin

y <= not a;

end Behavioral;
```

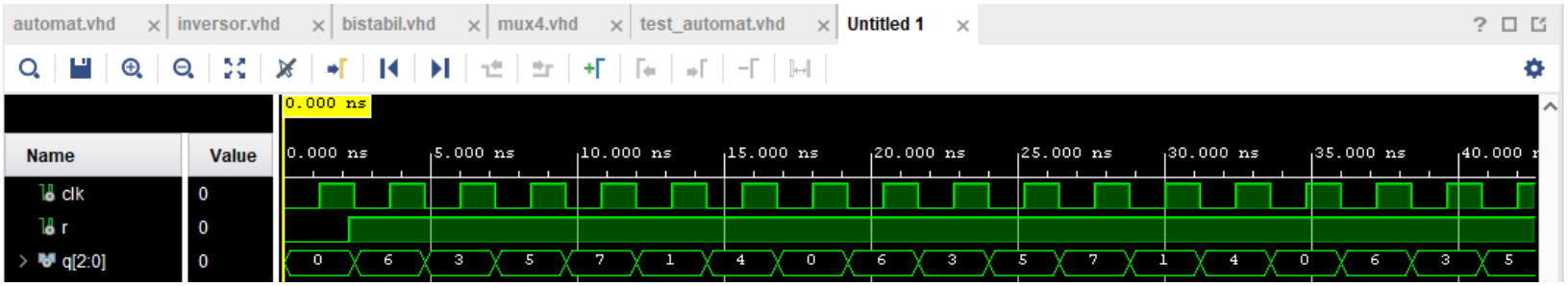
```vhdl
entity inv_sim is
end inv_sim;

architecture Behavioral of inv_sim is
signal a : std_logic;
    signal y : std_logic;
begin
    uut: entity work.inversor
        port map (
            a => a,
            y => y);
    stim_proc: process
    begin
        a <= '0';
        wait for 2 ns;
        a <= '1';
        wait for 2 ns;
        a <= '0';
        wait for 2 ns;
        a <= '1';
        wait for 2 ns;
        a <= '0';
        wait for 2 ns;
        a <= '1';
        wait for 2 ns;
        a <= '0';
        wait for 2 ns;
    end process;
end Behavioral;
```

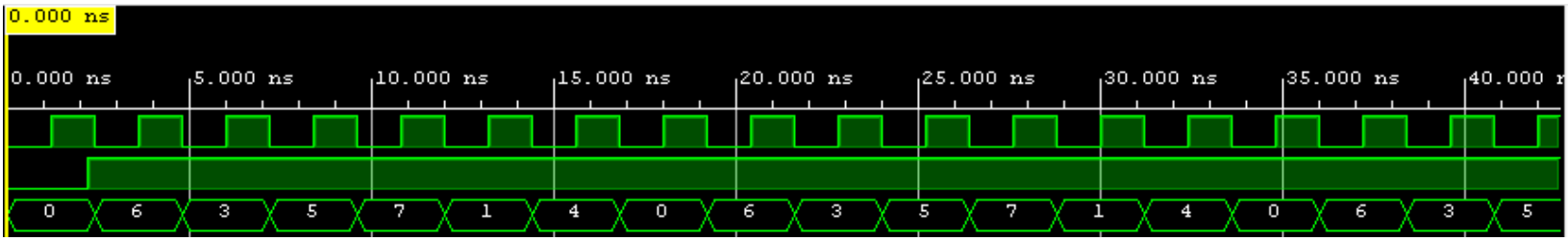


Sursa de design pentru automat și sursa de simulare a acestuia

```
entity automat is
    Port ( clk : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC_VECTOR (2 downto 0));
end automat;

architecture Behavioral of automat is

component mux4 is
    Port ( i0 : in STD_LOGIC;
           i1 : in STD_LOGIC;
           i2 : in STD_LOGIC;
           i3 : in STD_LOGIC;
           a1 : in STD_LOGIC;
           a2 : in STD_LOGIC;
           y : out STD_LOGIC);
end component mux4;

component bistD is
    Port ( clk : in STD_LOGIC;
           d : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC;
           qneg : out STD_LOGIC);
end component bistD;

component inversor is
    Port ( a : in STD_LOGIC;
           y : out STD_LOGIC);
end component inversor;
signal net1, net2, net3, netA:std_logic;
signal qint:std_logic_vector(2 downto 0);

begin

q <= qint;

B2: bistD port map(clk => clk, d =>net1, r => r, q => qint(2));
B1: bistD port map(clk => clk, d =>net2, r => r, q => qint(1));
B0: bistD port map(clk => clk, d =>net3, r => r, q => qint(0));

M1: mux4 port map(i0=>'1', i1=>'1', i2=>qint(0), i3=>'0', a2=>qint(2), a1=>qint(1), y=>net1);
M2: mux4 port map(i0=>netA, i1=>'0', i2=>qint(0), i3=>netA, a2=>qint(2), a1=>qint(1), y=>net2);
M3: mux4 port map(i0=>'0', i1=>qint(0), i2=>qint(0), i3=>'1', a2=>qint(2), a1=>qint(1), y=>net3);

I1: inversor port map ( a => qint(0), y => netA);

end Behavioral;
```

```
entity test_automat is
--  Port ( );
end test_automat;

architecture Behavioral of test_automat is

component automat is
    Port ( clk : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC_VECTOR (2 downto 0));
end component automat;

signal clk, r:std_logic;
signal q:std_logic_vector(2 downto 0);


begin

UT:automat port map(clk, r, q);

process
begin
clk<= '0'; wait for 1.2 ns;
clk<= '1'; wait for 1.2 ns;

end process;

r <= '0' after 0 ns, '1' after 2.2 ns;

end Behavioral;
```


automat.vhd
inversor.vhd
bistabil.vhd
mux4.vhd
test_automat.vhd
Untitled 1
Name
Value
clk
r
q[2:0]
0
0
0
0.000 ns
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
0 6 3 5 7 1 4 0 6 3 5 7 1 4 0 6 3 5


0.000 ns
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
0 6 3 5 7 1 4 0 6 3 5 7 1 4 0 6 3 5