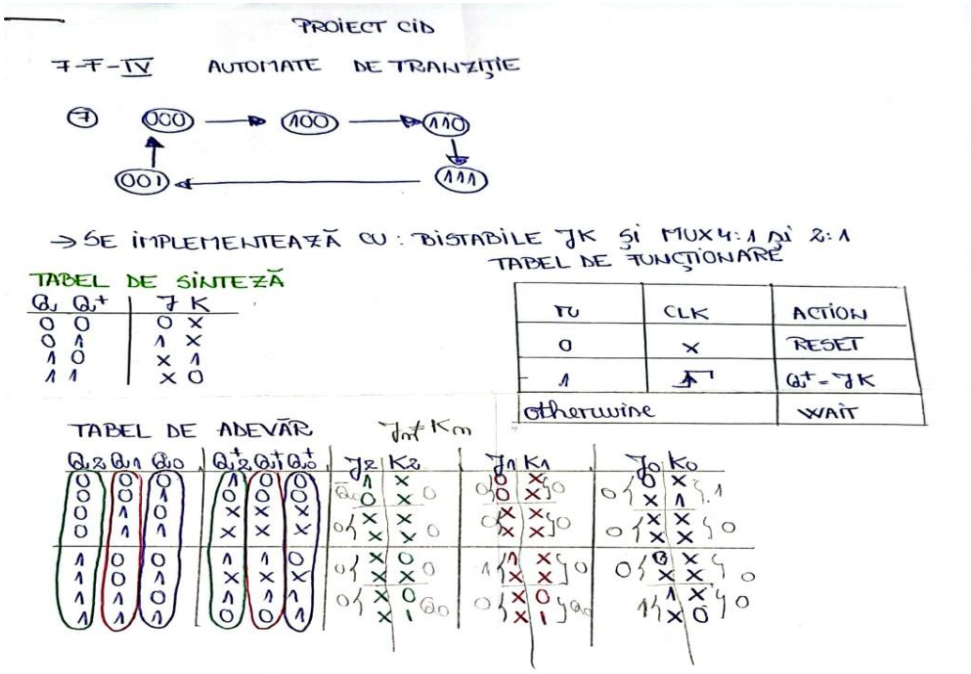
PROIECT
CIRCUITE INTEGRATE DIGITALE

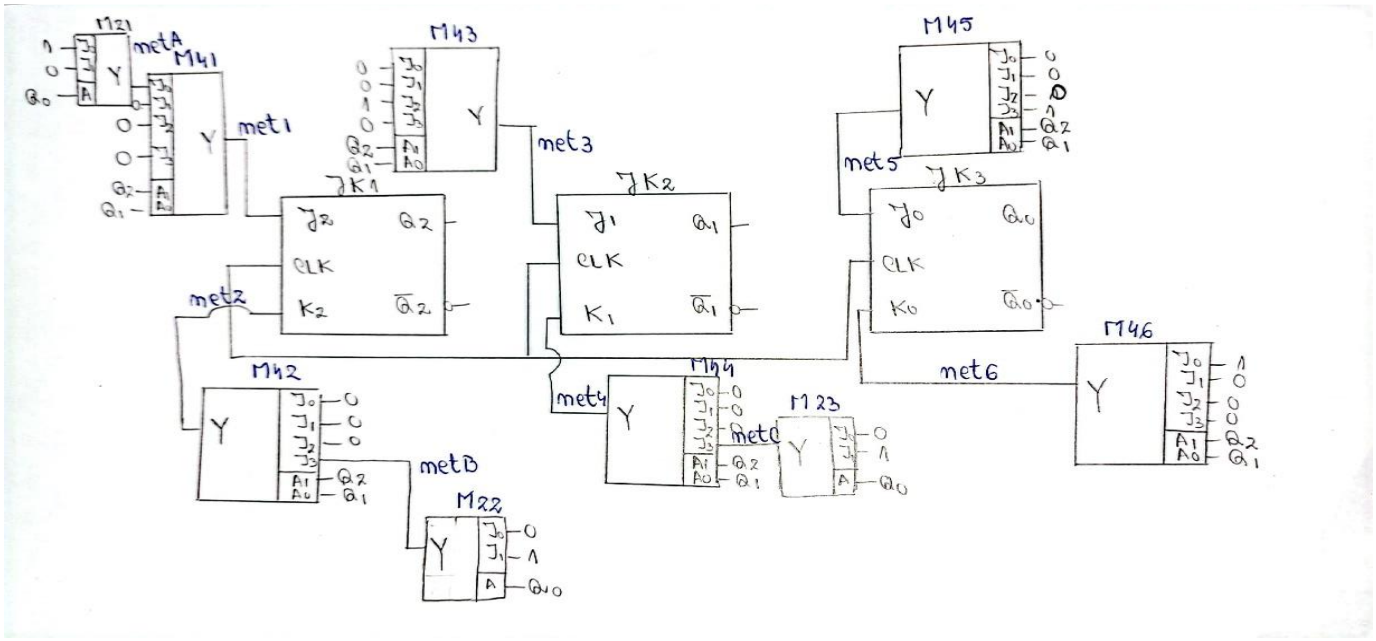
PROIECTUL CU CERINTELE:

7-F-IV

Hentes Cristiana-Maria
Seria:A
Grupa: 2123
Semigrupa: 1

1.Rezolvarea pe hartie a automatului: bistabil JK cu urmatoarea diagrama de tranzitii:

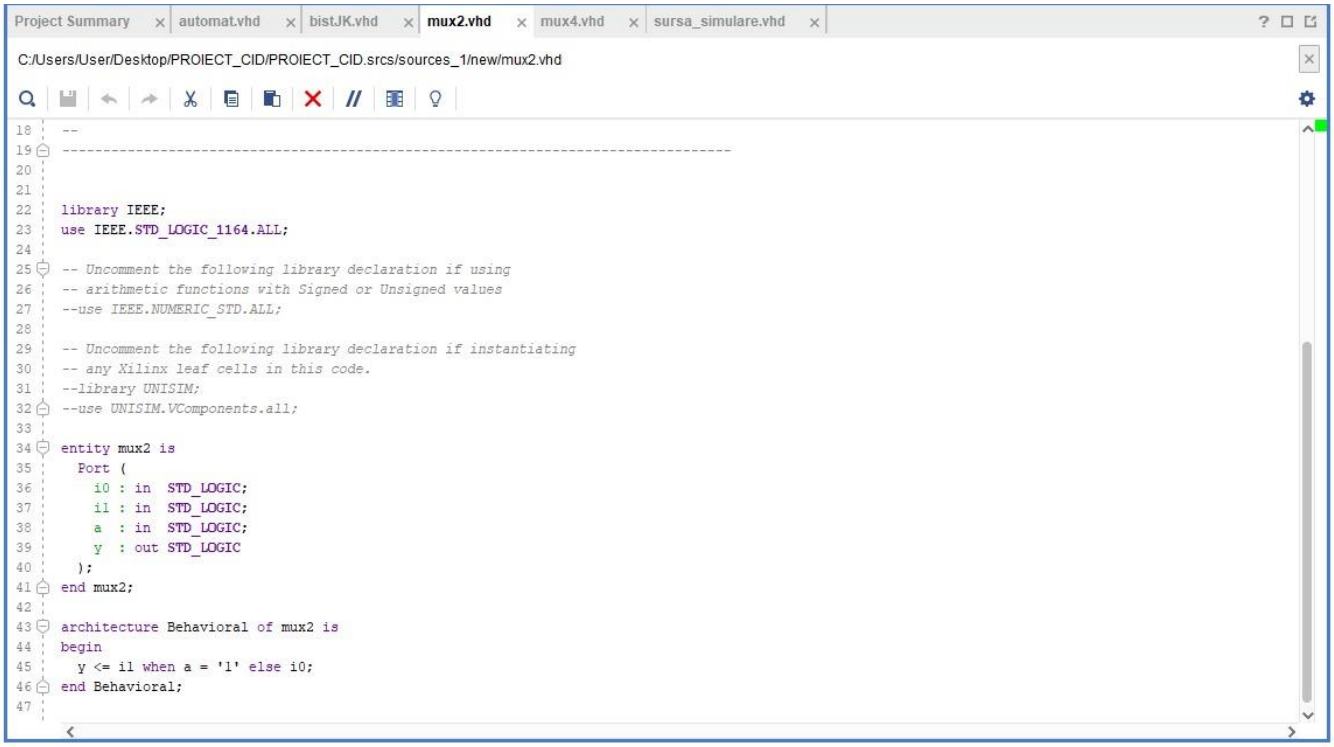




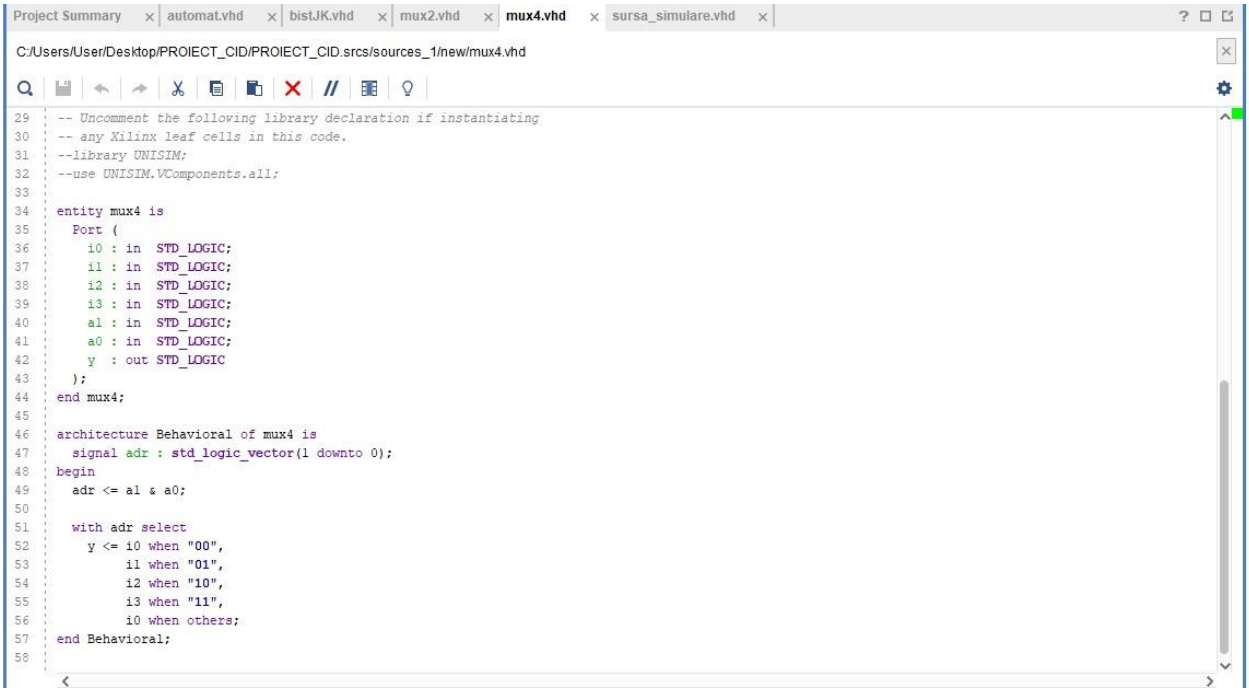
Pentru rezolvarea automatului am folosit 3 bistabile JK, 6 MUX 4:1 si 3 MUX 2:1.

2. Circuitul combinational- format din MUX 4:1 SI MUX 2:1.

MUX 2:1



MUX 4:1



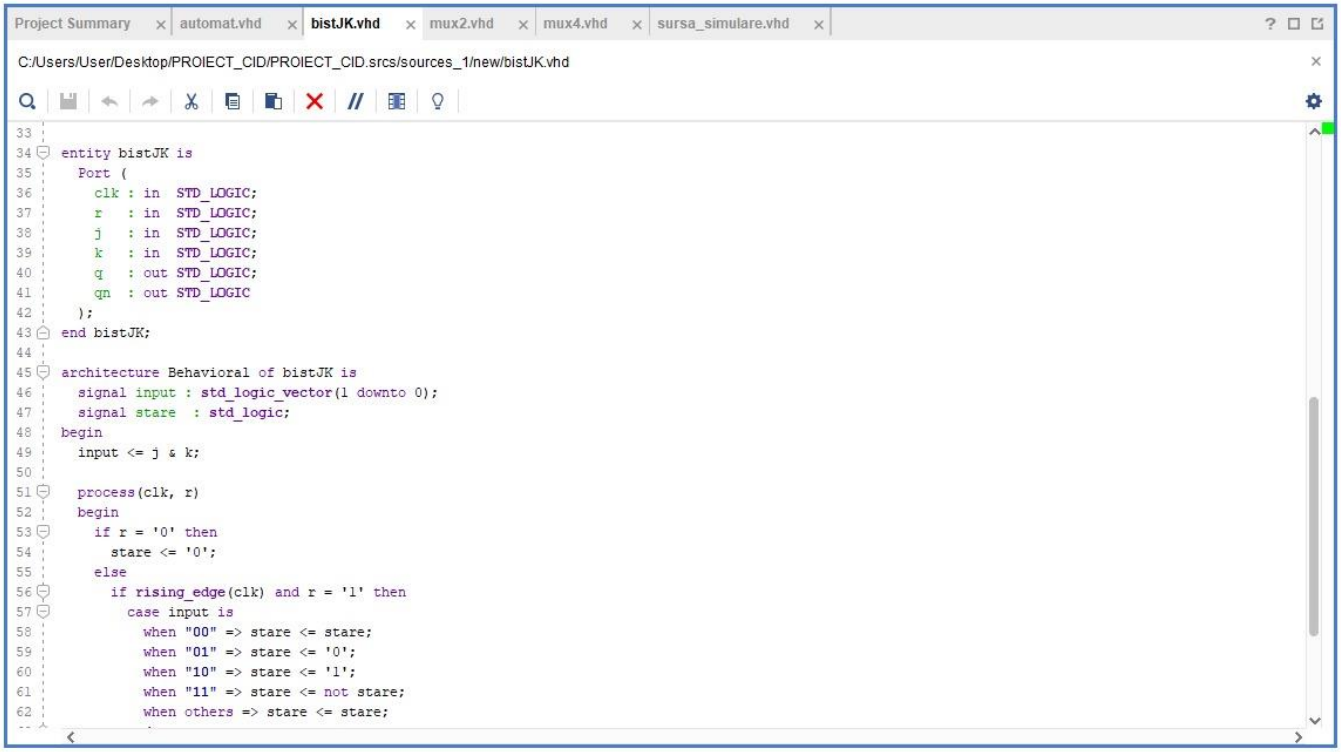

```
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity mux4 is
  Port (
    i0 : in  STD_LOGIC;
    i1 : in  STD_LOGIC;
    i2 : in  STD_LOGIC;
    i3 : in  STD_LOGIC;
    a1 : in  STD_LOGIC;
    a0 : in  STD_LOGIC;
    y  : out STD_LOGIC
  );
end mux4;

architecture Behavioral of mux4 is
  signal adr : std_logic_vector(1 downto 0);
begin
  adr <= a1 & a0;

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

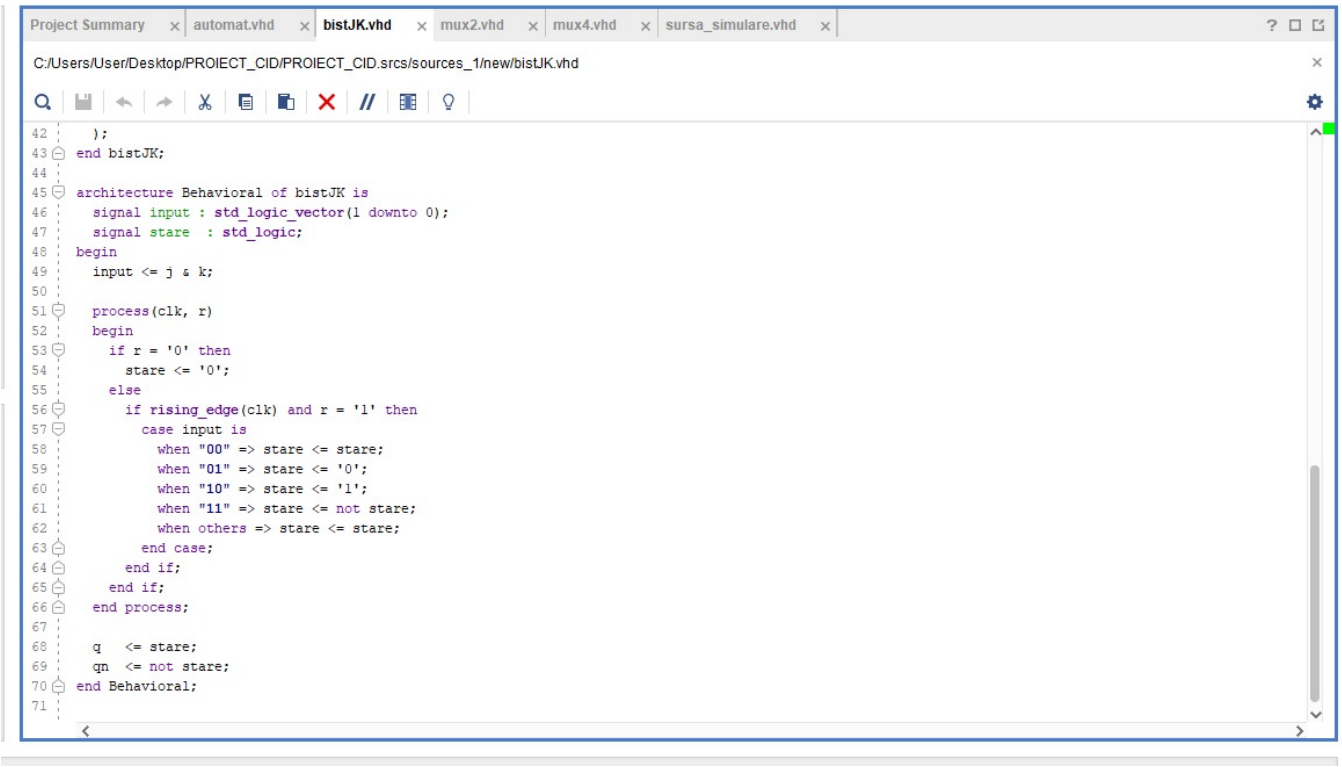
3.Circuitul secvential- bistabil JK.




```
entity bistJK is
  Port (
    clk : in  STD_LOGIC;
    r   : in  STD_LOGIC;
    j   : in  STD_LOGIC;
    k   : in  STD_LOGIC;
    q   : out STD_LOGIC;
    qn  : out STD_LOGIC
  );
end bistJK;

architecture Behavioral of bistJK is
  signal input : std_logic_vector(1 downto 0);
  signal stare  : std_logic;
begin
  input <= j & k;

  process(clk, r)
  begin
    if r = '0' then
      stare <= '0';
    else
      if rising_edge(clk) and r = '1' then
        case input is
          when "00" => stare <= stare;
          when "01" => stare <= '0';
          when "10" => stare <= '1';
          when "11" => stare <= not stare;
          when others => stare <= stare;
```



4.Automatul de stare:

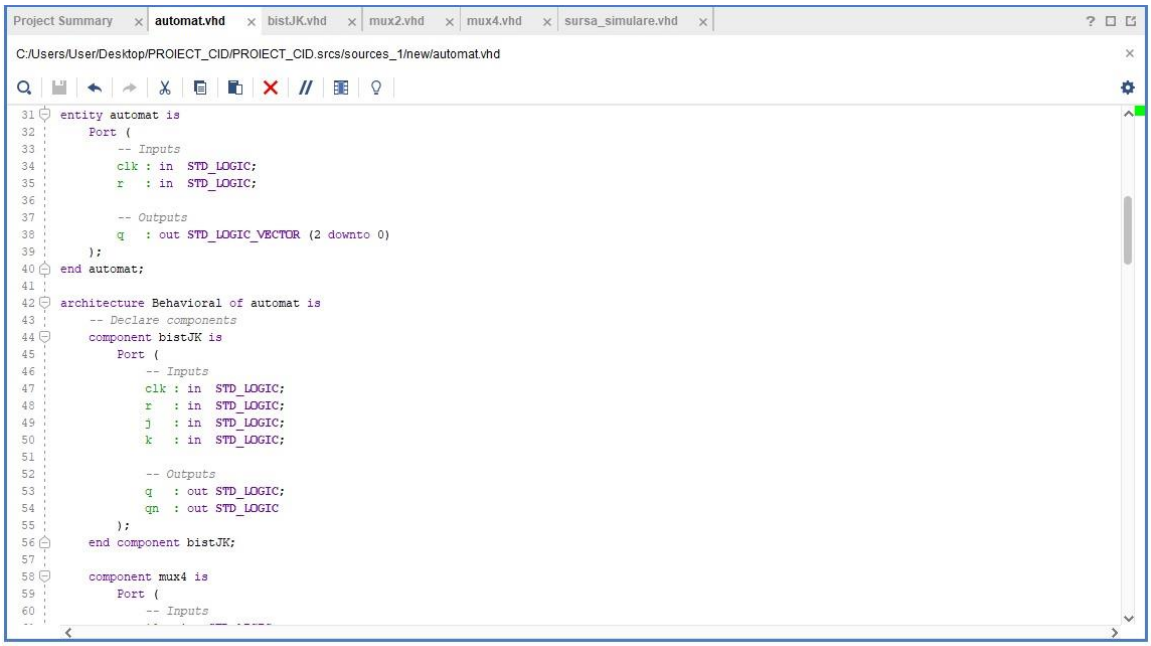
- am declarant in cadrul automatului de stare structurile de tip *entity* din cadrul MUX 4:1,

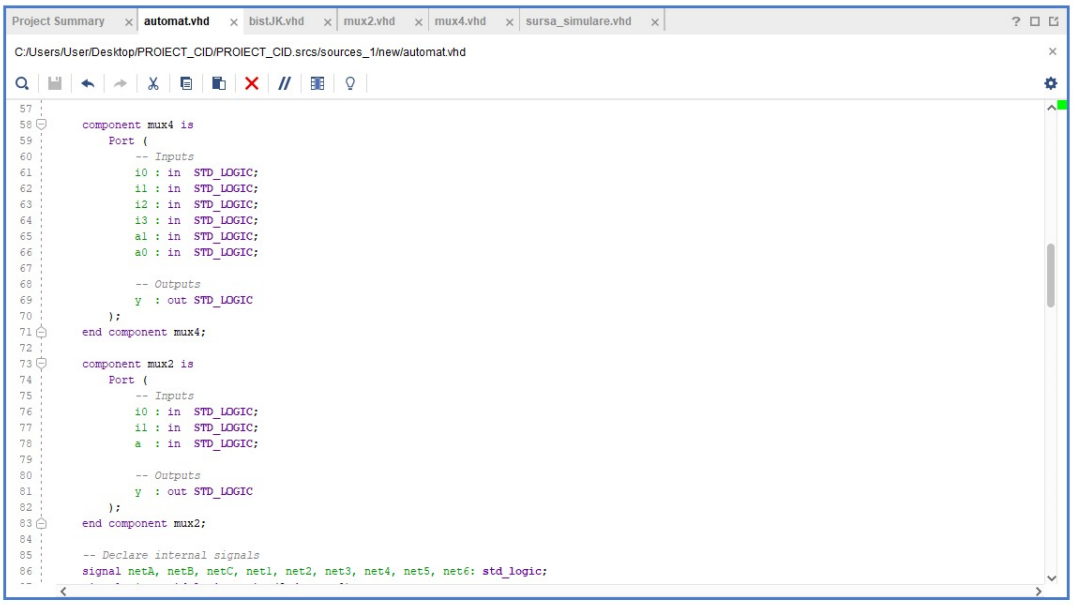
MUX 2:1 si bistabilului JK ca si structuri de tip *component*;

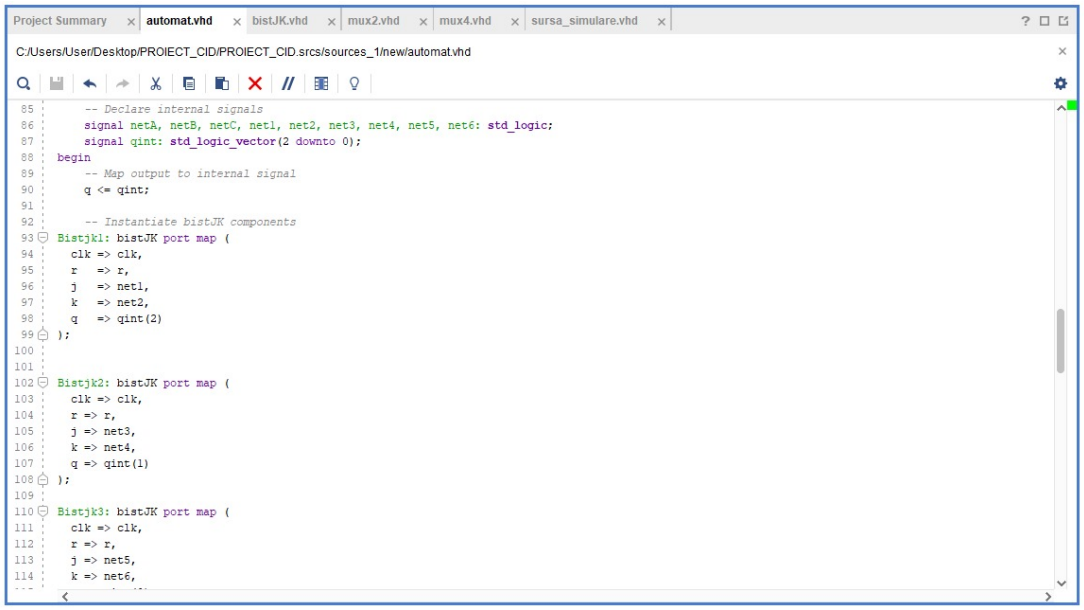
-ca si porturi de intrare in cadrul automatului avem: semnalul de clk si r, iar ca port de iesire avem q;

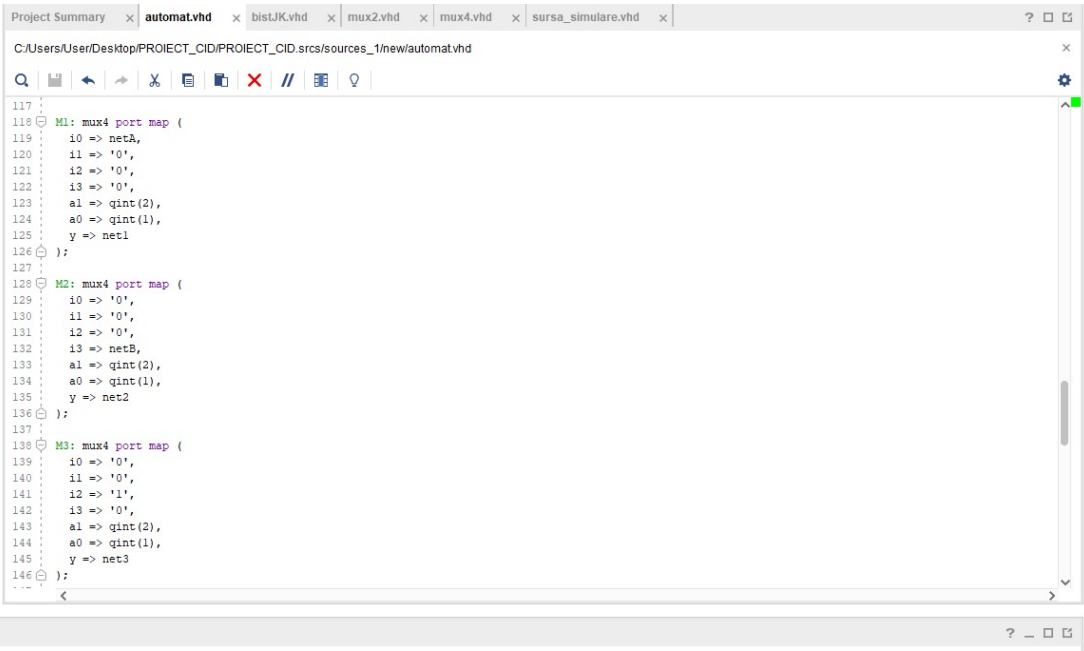
- am definit cele 6 componente ale circuitului de tip MUX 4:1 si cele 3 de tip MUX 2:1;

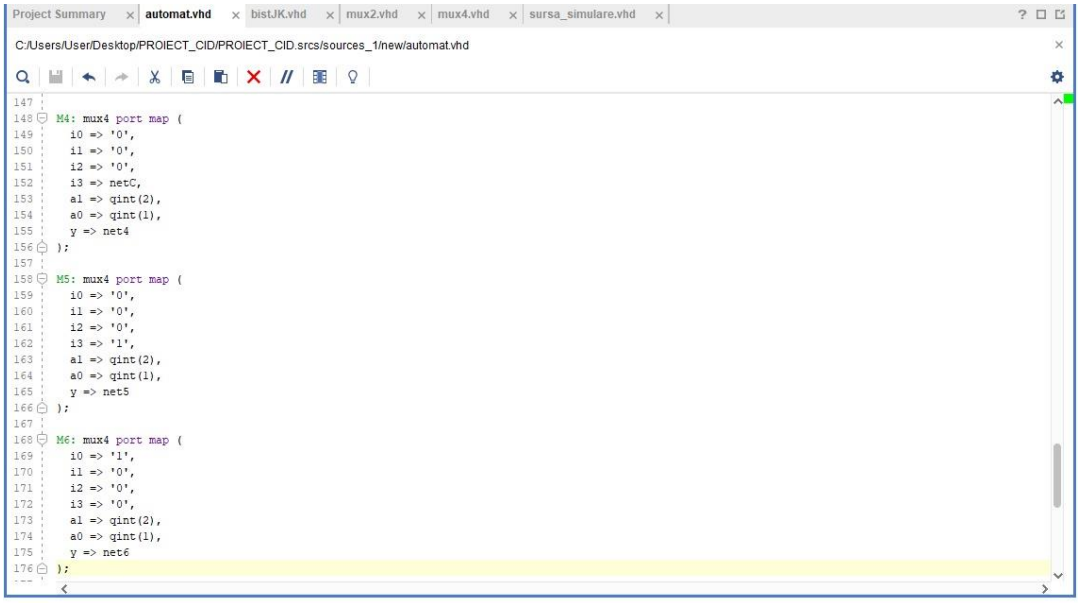
-am declarant semnalele care leaga componentele intre ele (net-urile);

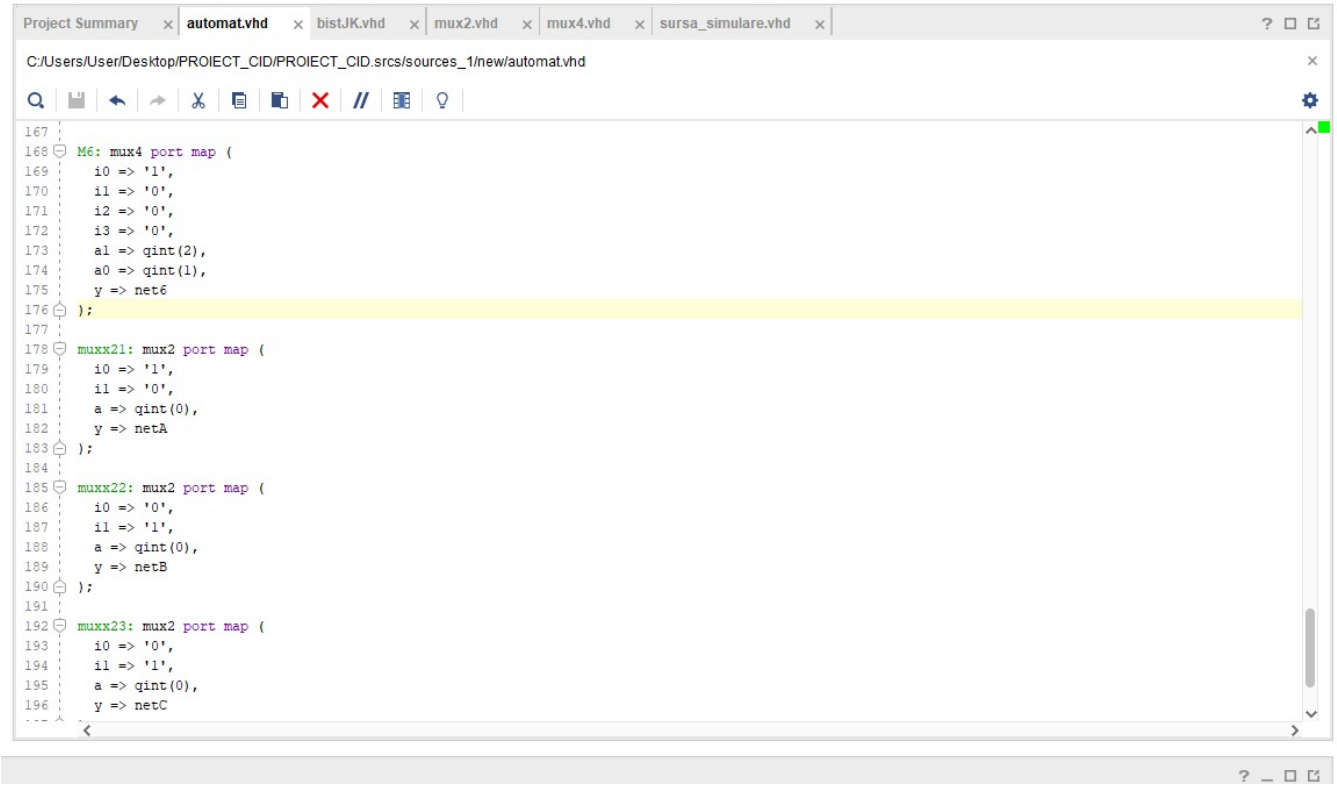

Project Summary
automat.vhd
bistJK.vhd
mux2.vhd
mux4.vhd
sursa_simulare.vhd
C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sources_1/new/automat.vhd
entity automat is
Port (
-- Inputs
clk : in STD_LOGIC;
r : in STD_LOGIC;
-- Outputs
q : out STD_LOGIC_VECTOR (2 downto 0)
);
end automat;
architecture Behavioral of automat is
-- Declare components
component bistJK is
Port (
-- Inputs
clk : in STD_LOGIC;
r : in STD_LOGIC;
j : in STD_LOGIC;
k : in STD_LOGIC;
-- Outputs
q : out STD_LOGIC;
qn : out STD_LOGIC
);
end component bistJK;
component mux4 is
Port (
-- Inputs


Project Summary
automat.vhd
bistJK.vhd
mux2.vhd
mux4.vhd
sursa_simulare.vhd
C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sources_1/new/automat.vhd
component mux4 is
Port (
-- Inputs
i0 : in STD_LOGIC;
i1 : in STD_LOGIC;
i2 : in STD_LOGIC;
i3 : in STD_LOGIC;
a1 : in STD_LOGIC;
a0 : in STD_LOGIC;
-- Outputs
y : out STD_LOGIC
);
end component mux4;
component mux2 is
Port (
-- Inputs
i0 : in STD_LOGIC;
i1 : in STD_LOGIC;
a : in STD_LOGIC;
-- Outputs
y : out STD_LOGIC
);
end component mux2;
-- Declare internal signals
signal netA, netB, netC, net1, net2, net3, net4, net5, net6: std_logic;


Project Summary
automat.vhd
bistJK.vhd
mux2.vhd
mux4.vhd
sursa_simulare.vhd
C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sources_1/new/automat.vhd
-- Declare internal signals
signal netA, netB, netC, net1, net2, net3, net4, net5, net6: std_logic;
signal qint: std_logic_vector(2 downto 0);
begin
-- Map output to internal signal
q <= qint;
-- Instantiate bistJK components
Bistjk1: bistJK port map (
clk => clk,
r => r,
j => net1,
k => net2,
q => qint(2)
);
Bistjk2: bistJK port map (
clk => clk,
r => r,
j => net3,
k => net4,
q => qint(1)
);
Bistjk3: bistJK port map (
clk => clk,
r => r,
j => net5,
k => net6,


Project Summary
automat.vhd
bistJK.vhd
mux2.vhd
mux4.vhd
sursa_simulare.vhd
C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sources_1/new/automat.vhd
M1: mux4 port map (
i0 => netA,
i1 => '0',
i2 => '0',
i3 => '0',
a1 => qint(2),
a0 => qint(1),
y => net1
);
M2: mux4 port map (
i0 => '0',
i1 => '0',
i2 => '0',
i3 => netB,
a1 => qint(2),
a0 => qint(1),
y => net2
);
M3: mux4 port map (
i0 => '0',
i1 => '0',
i2 => '1',
i3 => '0',
a1 => qint(2),
a0 => qint(1),
y => net3
);


Project Summary
automat.vhd
bistJK.vhd
mux2.vhd
mux4.vhd
sursa_simulare.vhd
C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sources_1/new/automat.vhd
M4: mux4 port map (
i0 => '0',
i1 => '0',
i2 => '0',
i3 => netC,
a1 => qint(2),
a0 => qint(1),
y => net4
);
M5: mux4 port map (
i0 => '0',
i1 => '0',
i2 => '0',
i3 => '1',
a1 => qint(2),
a0 => qint(1),
y => net5
);
M6: mux4 port map (
i0 => '1',
i1 => '0',
i2 => '0',
i3 => '0',
a1 => qint(2),
a0 => qint(1),
y => net6
);


Project Summary
automat.vhd
bistJK.vhd
mux2.vhd
mux4.vhd
sursa_simulare.vhd
C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sources_1/new/automat.vhd
M6: mux4 port map (
i0 => '1',
i1 => '0',
i2 => '0',
i3 => '0',
a1 => qint(2),
a0 => qint(1),
y => net6
);
muxx21: mux2 port map (
i0 => '1',
i1 => '0',
a => qint(0),
y => netA
);
muxx22: mux2 port map (
i0 => '0',
i1 => '1',
a => qint(0),
y => netB
);
muxx23: mux2 port map (
i0 => '0',
i1 => '1',
a => qint(0),
y => netC

5.Sursa de simulare

Project Summary | automat.vhd | bistJK.vhd | mux2.vhd | mux4.vhd | **sursa_simulare.vhd**

C:/Users/User/Desktop/PROIECT_CID/PROIECT_CID.srcs/sim_1/new/sursa_simulare.vhd

```
33
34  entity sursa_simulare is
35    --  Port ( );
36  end sursa_simulare;
37
38  architecture Behavioral of sursa_simulare is
39    component automat is
40      Port (
41        clk : in  STD_LOGIC;
42        r   : in  STD_LOGIC;
43        q   : out STD_LOGIC_VECTOR (2 downto 0)
44      );
45    end component automat;
46
47    signal clk, r : std_logic;
48    signal q      : std_logic_vector(2 downto 0);
49  begin
50    UUT: automat port map (clk, r, q);
51
52    process
53    begin
54
55      clk <= '0';
56      wait for 1.2 ns;
57      clk <= '1';
58      wait for 1.2 ns;
59    end process;
60
61    r <= '0' after 0 ns, '1' after 2.2 ns;
62  end Behavioral;
```

Functionalitate: