Student: Sâmboan Miruna

An: IIB

Grupa: 2125

Documentație

PROIECT CIRCUITE INTEGRATE DIGITALE

Temă proiect:

Automatul de tranziții: 4

Bistabil: D, Bistabil D

Implementare: I.doar MUX 2:1

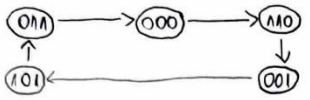
Instrucțiune concurentă: i. CASE

Reprezentarea proiectului pe hârtie

PROIECT CID

AUTOMATUL DE TRANZIȚII: 4
BISTABIL / NUMĂRĂTOR: D, Bistabil D
IMPLEMENTARE: I. doar MUX 2:1
INSTRUCȚIUNE CONCURENTĂ: Int. CASE

AUTOMATUL

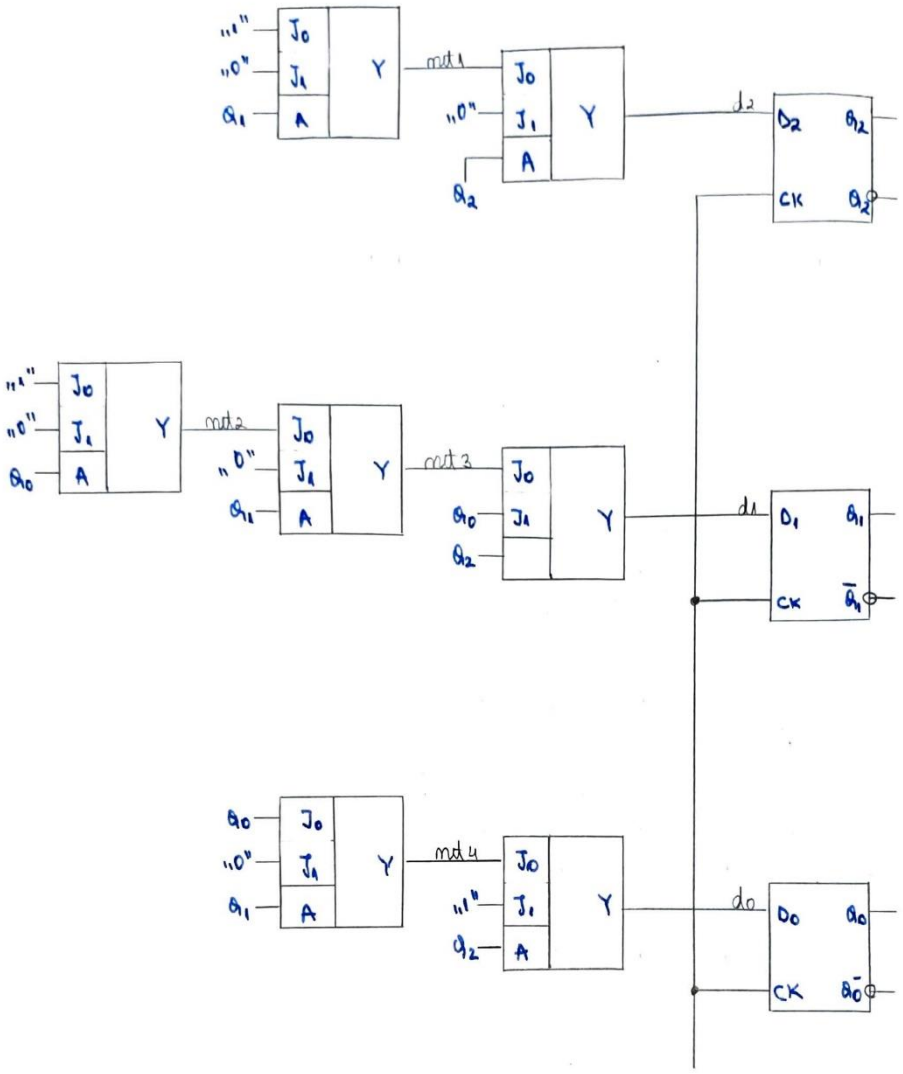


BISTABIL D

R	clk	Action
1	x	Reset
0	↓	$Q^+ = D$
altceva		Wait

TABELUL DE ADEVĂR

	Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
J_0	0	0	0	J_0: 1	J_0: J_1: 1	0
	0	0	1	1	J_0: 0	1
	0	1	0	J_2: x(0)	J_1: x(0)	x(0)
	0	1	1	0	0	0
J_2	1	0	0	x(0)	x(0)	x(1)
	1	0	1	0	1	1
	1	1	0	0	0	1
	1	1	1	x(0)	x(1)	x(1)


"1"
"0"
Q1
J0
J1
A
Y
mot1
"0"
Q2
d2
D2
Q2
CK
Q2
"1"
"0"
Q0
mot2
"0"
Q1
mot3
Q0
Q2
d1
D1
Q1
CK
Q1
Q0
"0"
Q1
mot4
"1"
Q2
d0
D0
Q0
CK
Q0

Rezolvarea temei de proiect în VIVADO

Surse de design

MUX 2:1

mux2.vhd

C:/Temp/project1/project1.srcs/sources_1/new/mux2.vhd

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity mux2 is
5      Port ( i0 : in STD_LOGIC;
6             i1 : in STD_LOGIC;
7             a : in STD_LOGIC;
8             y : out STD_LOGIC);
9  end mux2;
10
11 architecture Behavioral of mux2 is
12 begin
13     mux2: process (i1, i0, A)
14     begin
15         case(A) is
16             when '0' => y <= i0;
17             when '1' => y <= i1;
18             when others => y <= '0';
19         end case;
20     end process;
21 end Behavioral;
22
```

Bistabil D

dff.vhd

C:/Temp/project1/project1.srcs/sources_1/new/dff.vhd

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

entity dff is
    Port ( d : in STD_LOGIC;
           clk : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC;
           qn : out STD_LOGIC);
end dff;

architecture Behavioral of dff is

signal qint : std_logic;

begin

flipflop: process(r,clk)
begin
    if r = '1' then
        qint <= '0';
    elsif falling_edge (clk) then
        qint <= d;
    else
        qint <= qint;
    end if;
end process;

q <= qint;
qn <= not qint;

end Behavioral;
```

Automatul de tranziții

automat1.vhd

C:/Temp/project1/project1.srcs/sources_1/new/automat1.vhd

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

entity automat1 is
    Port ( clk : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC_VECTOR (2 downto 0));
end automat1;

architecture Behavioral of automat1 is

  component dff is
    Port ( d : in STD_LOGIC;
           clk : in STD_LOGIC;
           r : in STD_LOGIC;
           q : out STD_LOGIC;
           qn : out STD_LOGIC);
  end component dff;

  component mux2 is
    Port ( i0 : in STD_LOGIC;
           i1 : in STD_LOGIC;
           a : in STD_LOGIC;
           y : out STD_LOGIC);
  end component mux2;

  signal net1, net2, net3, net4: STD_LOGIC;
  signal qint: std_logic_vector(2 downto 0);
  signal d: std_logic_vector(2 downto 0);

begin

  q <= qint;

  U1: mux2 port map (i0=>'1', i1=>'0', a=>qint(1), y=>net1);
  U2: mux2 port map (i0=>net1, i1=>'0', a=>qint(2), y=>d(2));
  U3: mux2 port map (i0=>'1', i1=>'0', a=>qint(0), y=>net2);
  U4: mux2 port map (i0=>net2, i1=>'0', a=>qint(1), y=>net3);
  U5: mux2 port map (i0=>net3, i1=>qint(0), a=>qint(2), y=>d(1));
  U6: mux2 port map (i0=>qint(0), i1=>'0', a=>qint(1), y=>net4);
  U7: mux2 port map (i0=>net4, i1=>'1', a=>qint(2), y=>d(0));

  UT0: dff port map (q => qint(0),
                     d => d(0),
                     clk => clk,
                     r => r);

  UT1: dff port map (q => qint(1),
                     d => d(1),
                     clk => clk,
                     r => r);

  UT2: dff port map (q => qint(2),
                     d => d(2),
                     clk => clk,
                     r => r);
end Behavioral;
```

Sursele de simulare si rezultatele simulării

MUX 2:1

-sursa simulării

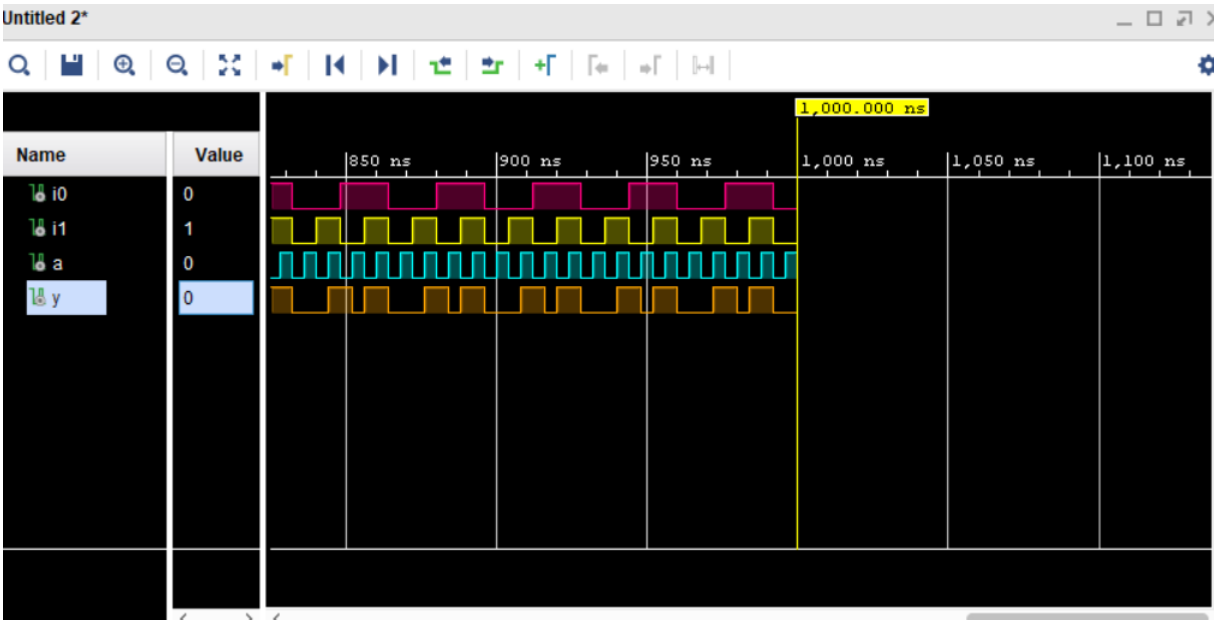
test_mux.vhd

C:/Temp/project1/project1.srcs/sim_1/new/test_mux.vhd

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity test_mux is
5  --  Port ( );
6  end test_mux;
7
8  architecture Behavioral of test_mux is
9
10 component mux2 is
11     Port ( i0 : in STD_LOGIC;
12            i1 : in STD_LOGIC;
13            a : in STD_LOGIC;
14            y : out STD_LOGIC);
15 end component mux2;
16
17 signal i0, i1, a, y: STD_LOGIC;
18
19 begin
20
21 UUT: mux2 port map (i0 => i0,
22                     i1 => i1,
23                     a => a,
24                     y => y);
25 process
26
27 begin
28     i0 <= '0';
29     i1 <= '0';
30     a <= '0';
31     wait for 4 ns;
32
33     i0 <= '0';
34     i1 <= '0';
35     a <= '1';
36     wait for 4 ns;
37
```

```
38      i0 <= '0';
39      il <= '1';
40      a <= '0';
41      wait for 4 ns;
42
43      i0 <= '0';
44      il <= '1';
45      a <= '1';
46      wait for 4 ns;
47
48      i0 <= '1';
49      il <= '0';
50      a <= '0';
51      wait for 4 ns;
52
53      i0 <= '1';
54      il <= '0';
55      a <= '1';
56      wait for 4 ns;
57
58      i0 <= '1';
59      il <= '1';
60      a <= '0';
61      wait for 4 ns;
62
63      i0 <= '1';
64      il <= '1';
65      a <= '1';
66      wait for 4 ns;
67      end process;
68
69  end Behavioral;
```

-rezultatul simulării



Automatul de tranziții

-sursa simulării

test_automat1.vhd *

C:/Temp/project1/project1.srcs/sim_1/new/test_automat1.vhd

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity test_automat1 is
5  --  Port ( );
6  end test_automat1;
7
8  architecture Behavioral of test_automat1 is
9
10 component automat1 is
11     Port ( clk : in STD_LOGIC;
12            r : in STD_LOGIC;
13            q : out STD_LOGIC_VECTOR (2 downto 0));
14 end component automat1;
15
16 signal clk, r: std_logic;
17 signal q: std_logic_vector (2 downto 0);
18
19 begin
20
21 UUT: automat1 port map (clk => clk,
22                         r => r,
23                         q => q);
24
25 generate_clk: process --genereaza clk
26
27 begin
28 clk <= '0'; wait for 1 ns;
29 clk <= '1'; wait for 1 ns;
30 end process;
31
32 r <= '1' after 0 ns, '0' after 0.5 ns;
33
34 end Behavioral;
```

-rezultatul simulării