Proiect

Circuite integrate digitale

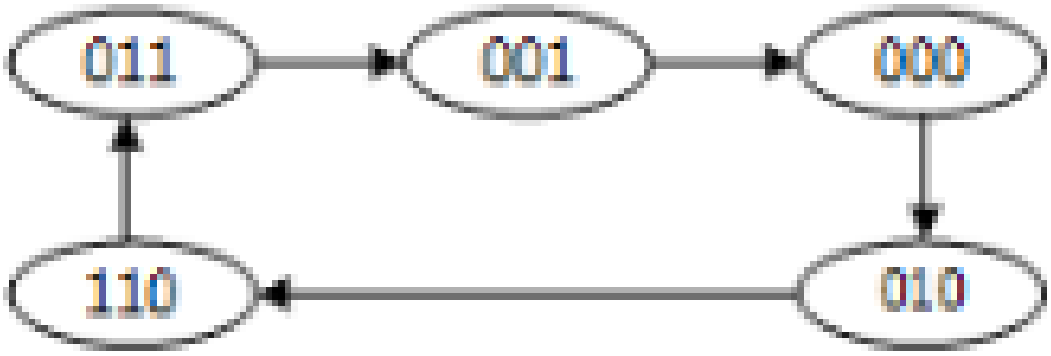
Student: Melnic Anastasia

Grupa:2124

Profesor:Claudia Georgiana Cordos

Cod proiect: 5-F-I

5.

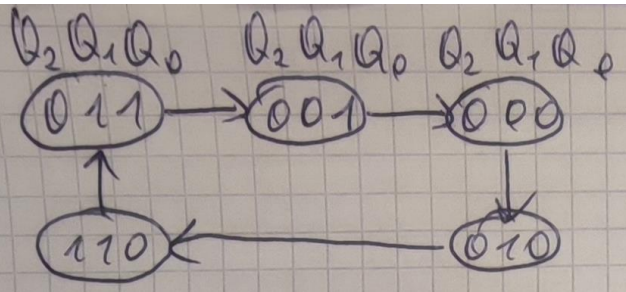


F.

r	clk	Action
0	x	Reset
1	↑	$Q^+ = JK$
otherwise		Wait

I. Doar MUX 2:1

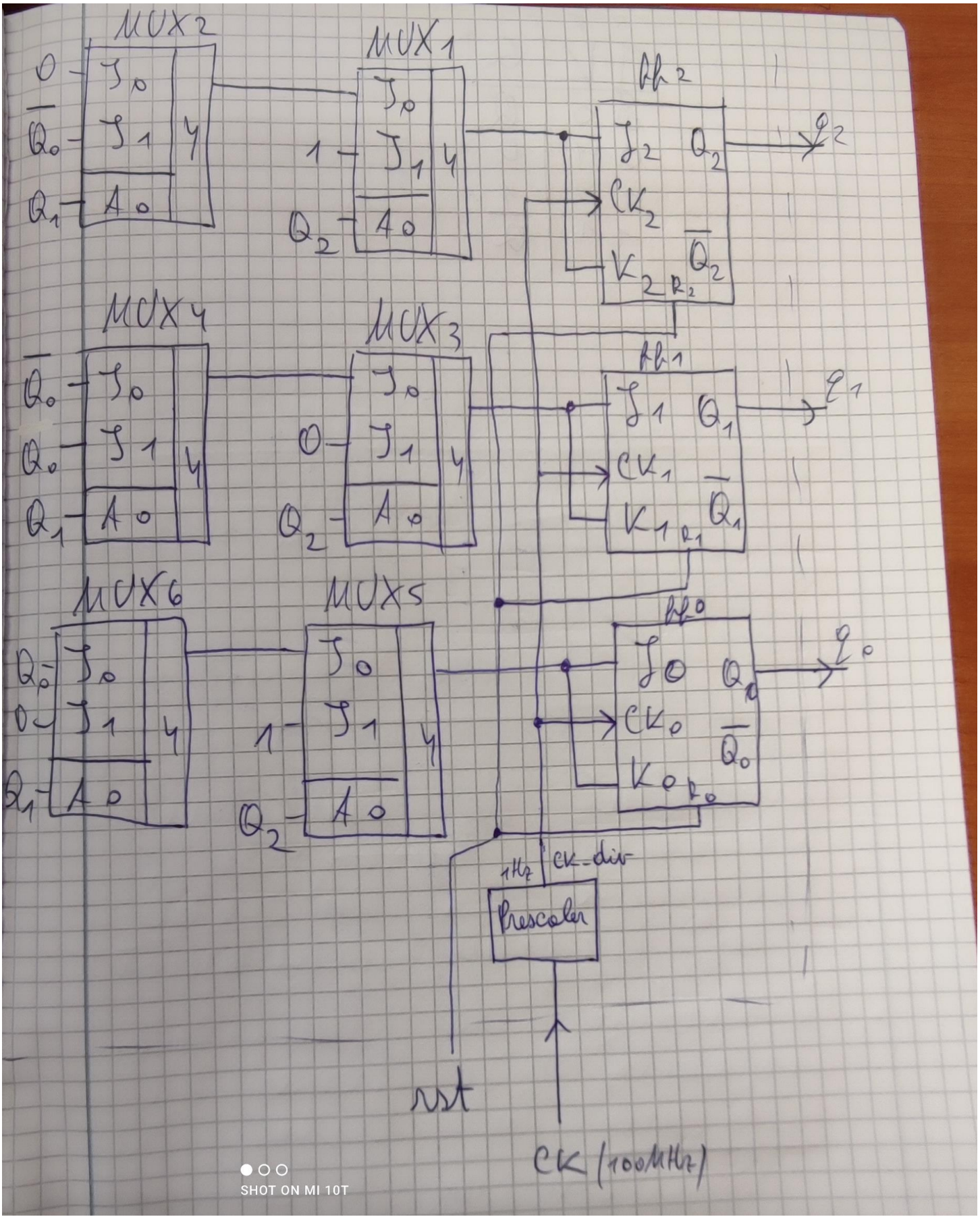
1) Rezolvarea temei de proiect pe hartie



Q	Q^+	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	1	0	0	x	1	x	0	x
0	0	1	0	0	0	0	x	0	x	x	1
0	1	0	1	1	0	1	x	x	0	0	x
0	1	1	0	0	1	0	x	x	1	x	0
1	0	0	x	x	x	x	x	x	x	x	x
1	0	1	x	x	x	x	x	x	x	x	x
1	1	0	0	1	1	x	1	x	0	1	x
1	1	1	x	x	x	x	x	x	x	x	x

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	$J_2=K_2$	$J_1=K_1$	$J_0=K_0$
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	0	1	0	0
0	1	1	0	0	1	0	1	0
1	0	0	x	x	x	x	x	x
1	0	1	x	x	x	x	x	x
1	1	0	0	1	1	1	0	1
1	1	1	x	x	x	x	x	x



2) Codul pentru mux2:1

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

entity mux2_1 is
    Port ( i0 : in STD_LOGI
           i1 : in STD_LOGI
           a0 : in STD_LOGI
           y : out STD_LOGI
end mux2_1;

architecture Behavioral of

begin

with a0 select
y<=i0 when '0',
   i1 when '1',
   i0 when others;

end Behavioral;
```

3) Codul pentru bistabilul JK

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity JK_ff is
    Port ( ck : in STD_LOGIC;
           rst:in std_logic;
           j : in STD_LOGIC;
           k : in STD_LOGIC;
           q : out STD_LOGIC;
           q_neg : out STD_LOGIC);
end JK_ff;
architecture Behavioral of JK_ff is
signal q_int:std_logic;--stare
signal jk:std_logic_vector(1 downto 0);--input
begin
jk<=j&k;
ff:process(ck, rst)
begin
    if rst='1' then
        q_int<='0';
    elsif rising_edge(ck) then
        case jk is
            when "00"=>q_int<=q_int;
            when "01"=>q_int<='0';
            when "10"=>q_int<='1';
            when "11"=>q_int<=not q_int;
            when others=>q_int<=q_int;
        end case;
    end if;
end process;
q<=q_int;
q_neg<=not q_int;
end Behavioral;
```

4) Codul pentru divizorul de frecventa

ck_divider.vhd × | state_machine.vhd × | JK_ff.vhd * × | testbench.vhd × | Untitled 2

D:/Xilinx/state_machine/state_machine.srcs/sources_1/new/ck_divider.vhd

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   use IEEE.NUMERIC_STD.ALL;
4
5   entity ck_divider is
6       Port ( ck : in STD_LOGIC;
7              rst : in STD_LOGIC;
8              ck_div : out STD_LOGIC);
9   end ck_divider;
10
11  architecture Behavioral of ck_divider is
12  constant const_div:integer:=10;--**8;
13  signal cnt:integer;
14  begin
15
16  ck_divide:process(ck, rst)
17  begin
18      if rst='1' then
19          cnt<=0;
20      elsif rising_edge(ck) then
21          if cnt=const_div-1 then
22              cnt<=0;
23              ck_div<='1';
24          else
25              cnt<=cnt+1;
26              ck_div<='0';
27          end if;
28      end if;
29  end process;
30
31  end Behavioral;
```

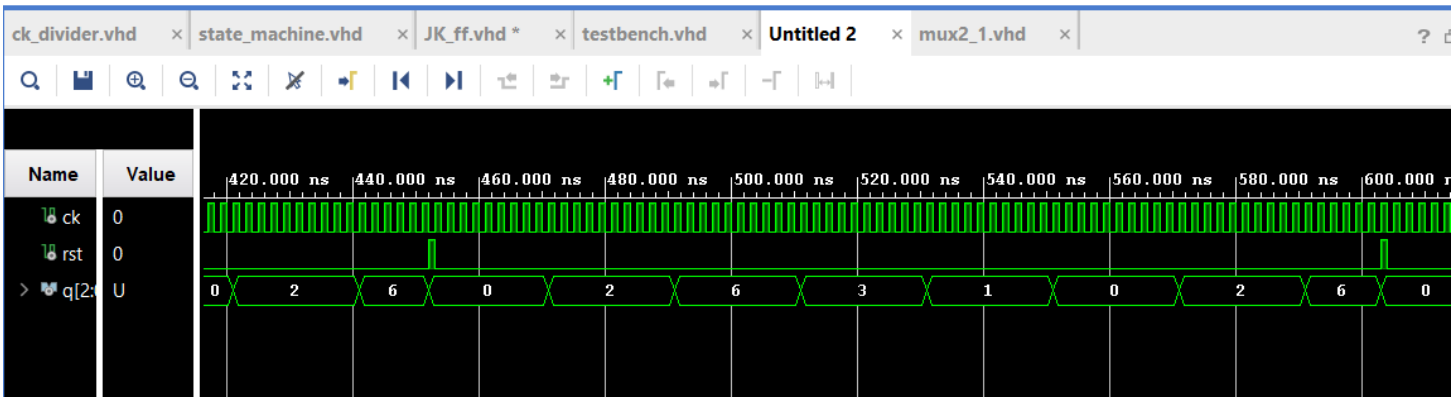
5) Codul pentru automat

ck_divider.vhd | **state_machine.vhd** | JK_ff.vhd * | testbench.vhd | Untitled 2 | mux

D:/Xilinx/state_machine/state_machine.srcs/sources_1/new/state_machine.vhd

```vhdl

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity state_machine is
    Port ( ck : in STD_LOGIC;
           rst : in STD_LOGIC;
           q_out : out std_logic_vector(2 downto 0));
end state_machine;

architecture Behavioral of state_machine is

component ck_divider is
    Port ( ck : in STD_LOGIC;
           rst : in STD_LOGIC;
           ck_div : out STD_LOGIC);
end component;

component mux2_1 is
    Port ( i0 : in STD_LOGIC;
           i1 : in STD_LOGIC;
           a0 : in STD_LOGIC;
           y : out STD_LOGIC);
end component;

component JK_ff is
    Port ( ck : in STD_LOGIC;
           rst:in std_logic;
           j : in STD_LOGIC;
           k : in STD_LOGIC;
           q : out STD_LOGIC;
           q_neg : out STD_LOGIC);
end component;

signal ck_div,q0_neg,y1,y2,y3,y4,y5,y6:std_logic;
signal q_int:std_logic_vector(2 downto 0);

begin

div_ck: ck_divider port map(ck=>ck, rst=>rst, ck_div=>ck_div);
mux1:mux2_1 port map(i0=>y2, i1=>'1', a0=>q_int(2), y=>y1);
mux2:mux2_1 port map(i0=>'0', i1=>q0_neg, a0=>q_int(1), y=>y2);
mux3:mux2_1 port map(i0=>y4, i1=>'0', a0=>q_int(2), y=>y3);
mux4:mux2_1 port map(i0=>q0_neg, i1=>q_int(0), a0=>q_int(1), y=>y4);
mux5:mux2_1 port map(i0=>y6, i1=>'1', a0=>q_int(2), y=>y5);
mux6:mux2_1 port map(i0=>q_int(0), i1=>'0', a0=>q_int(1), y=>y6);
ff0:JK_ff port map( ck=>ck_div, rst=>rst, j=>y5, k=>y5,q=>q_int(0));
ff1:JK_ff port map( ck=>ck_div, rst=>rst,j=>y3, k=>y3, q=>q_int(1));
ff2:JK_ff port map( ck=>ck_div, rst=>rst,j=>y1, k=>y1, q=>q_int(2));

q_out<=q_int;
q0_neg<=not q_int(0);

end Behavioral;
```

6) Simularea:


ck_divider.vhd
state_machine.vhd
JK_ff.vhd *
testbench.vhd
Untitled 2
mux2_1.vhd
Name
Value
ck
0
rst
0
q[2:
U
420.000 ns
440.000 ns
460.000 ns
480.000 ns
500.000 ns
520.000 ns
540.000 ns
560.000 ns
580.000 ns
0
2
6
0
2
6
3
1
0
2
6
0