UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

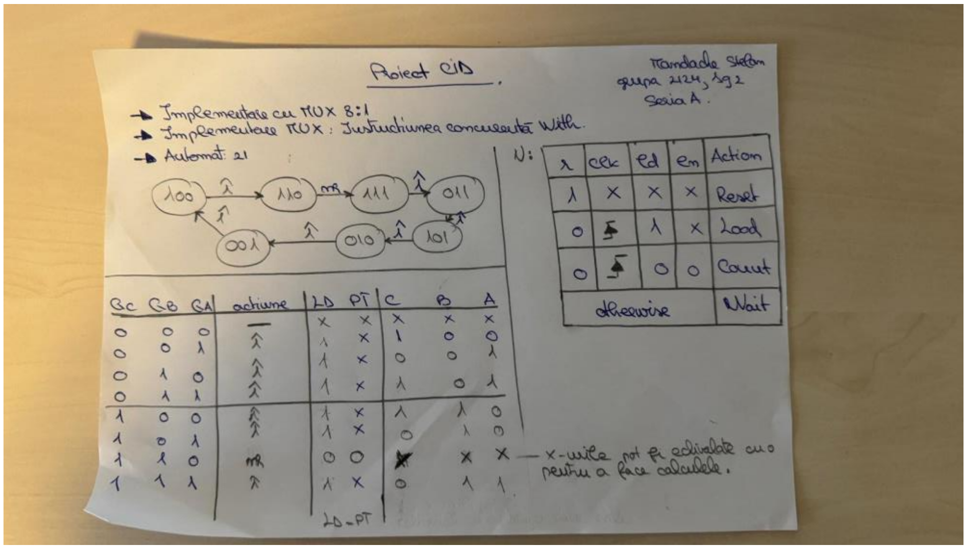
# *Proiect CID:*

# *Proiectarea unui Automat de Tranziții folosind un numărător, utilizând MUX 8:1*

***Proiect realizat de***

***Mandache Stefan , grupa 2124 , semigrupa 2***

## *Rezolvarea pe hartie a temei si desenele pentru MUX si NUMARATOR*



Proiect CID.

Mandache Stefan
grupa 2124, sg 2
Seria A.

- Implementare cu MUX 8:1
- Implementare MUX: Instrucțiunea concurentă With.
- Automat: 21

100
110
111
011
101
010
001
$\hat{1}$
mR
$\hat{1}$
$\hat{1}$
$\hat{1}$
$\hat{1}$
$\hat{1}$

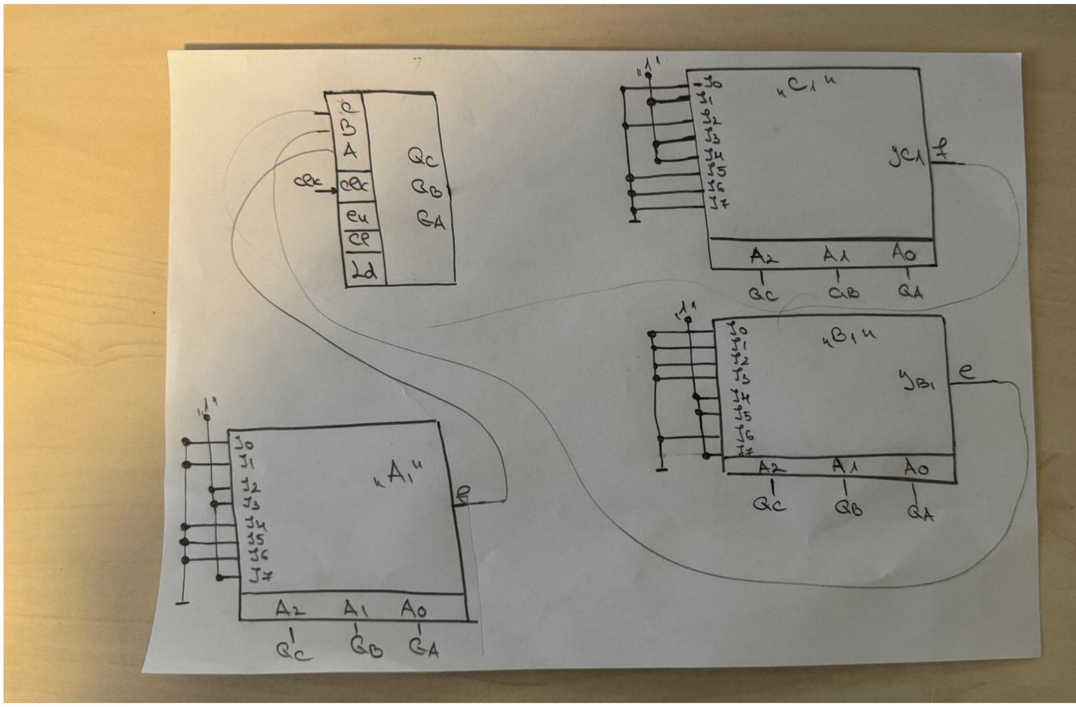
N:

| 1 | clk | ld | en | Action |
|---|---|---|---|---|
| 1 | x | x | x | Reset |
| 0 | ↑ | 1 | x | Load |
| 0 | ↑ | 0 | 0 | Count |
| otherwise | | | | Wait |

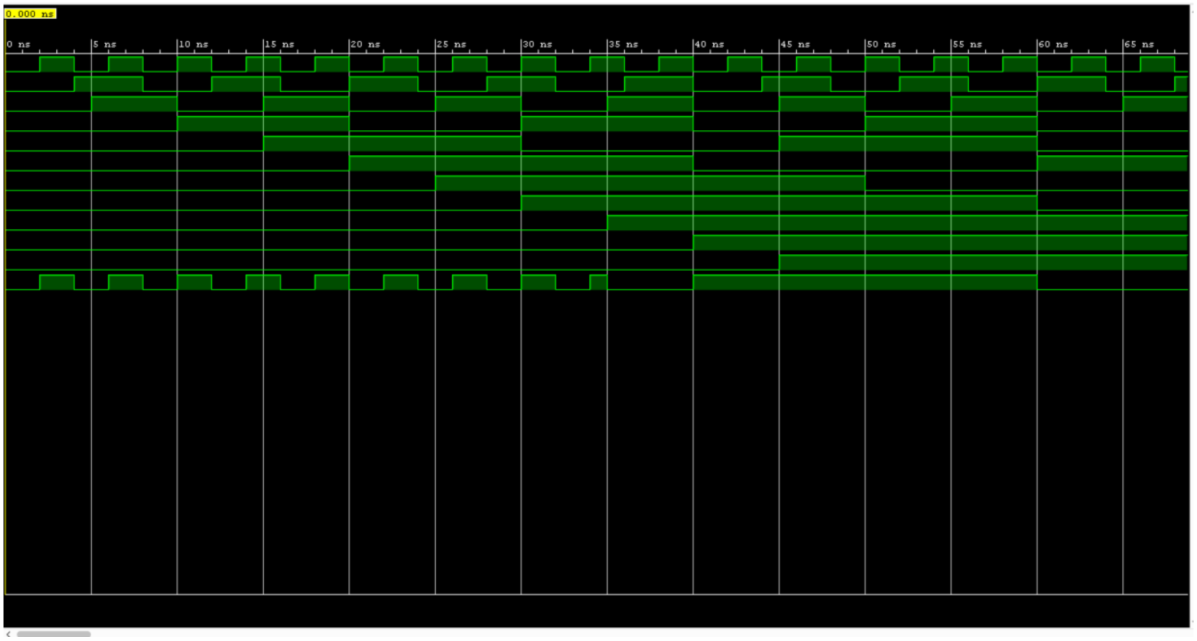
| Qc | QB | QA | actiune | LD | PT | C | B | A |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | — | x | x | x | x | x |
| 0 | 0 | 1 | $\hat{1}$ | 1 | x | 1 | 0 | 0 |
| 0 | 1 | 0 | $\hat{1}$ | 1 | x | 0 | 0 | 1 |
| 0 | 1 | 1 | $\hat{1}$ | 1 | x | 1 | 0 | 1 |
| 1 | 0 | 0 | $\hat{1}$ | 1 | x | 1 | 1 | 0 |
| 1 | 0 | 1 | $\hat{1}$ | 1 | x | 0 | 1 | 0 |
| 1 | 1 | 0 | mR | 0 | 0 | ~~x~~ | x | x |
| 1 | 1 | 1 | $\hat{1}$ | 1 | x | 0 | 1 | 1 |

— x-urile pot fi echivalate cu 0 pentru a face calculele.

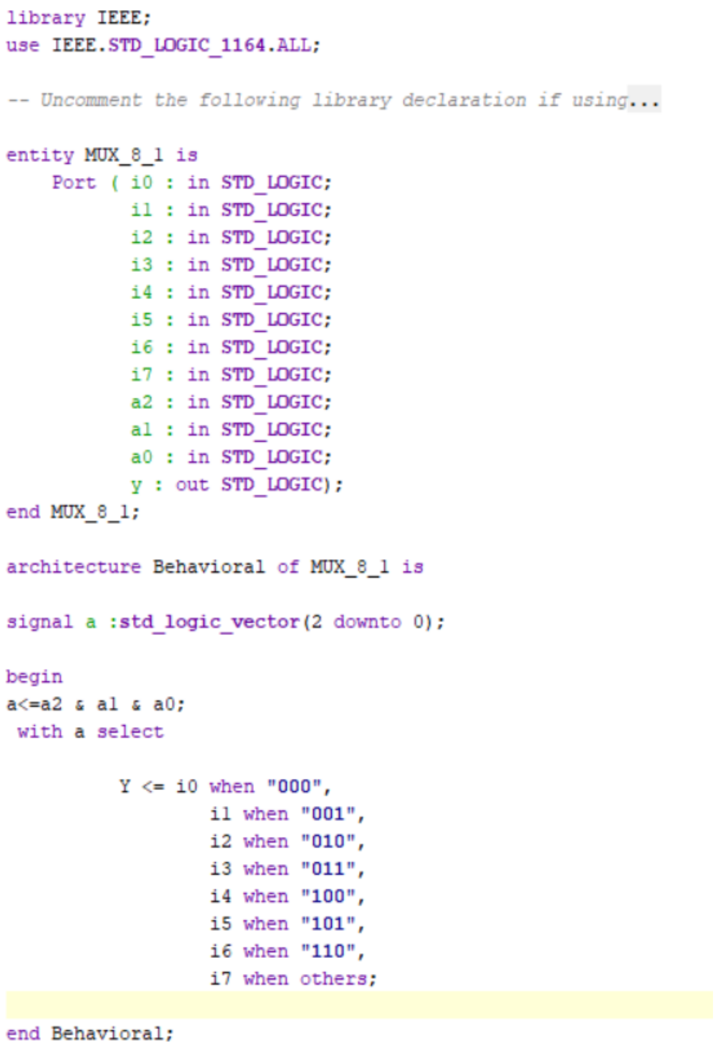
LD = PT



## *Testarea MUX-ului 8:1 ;*

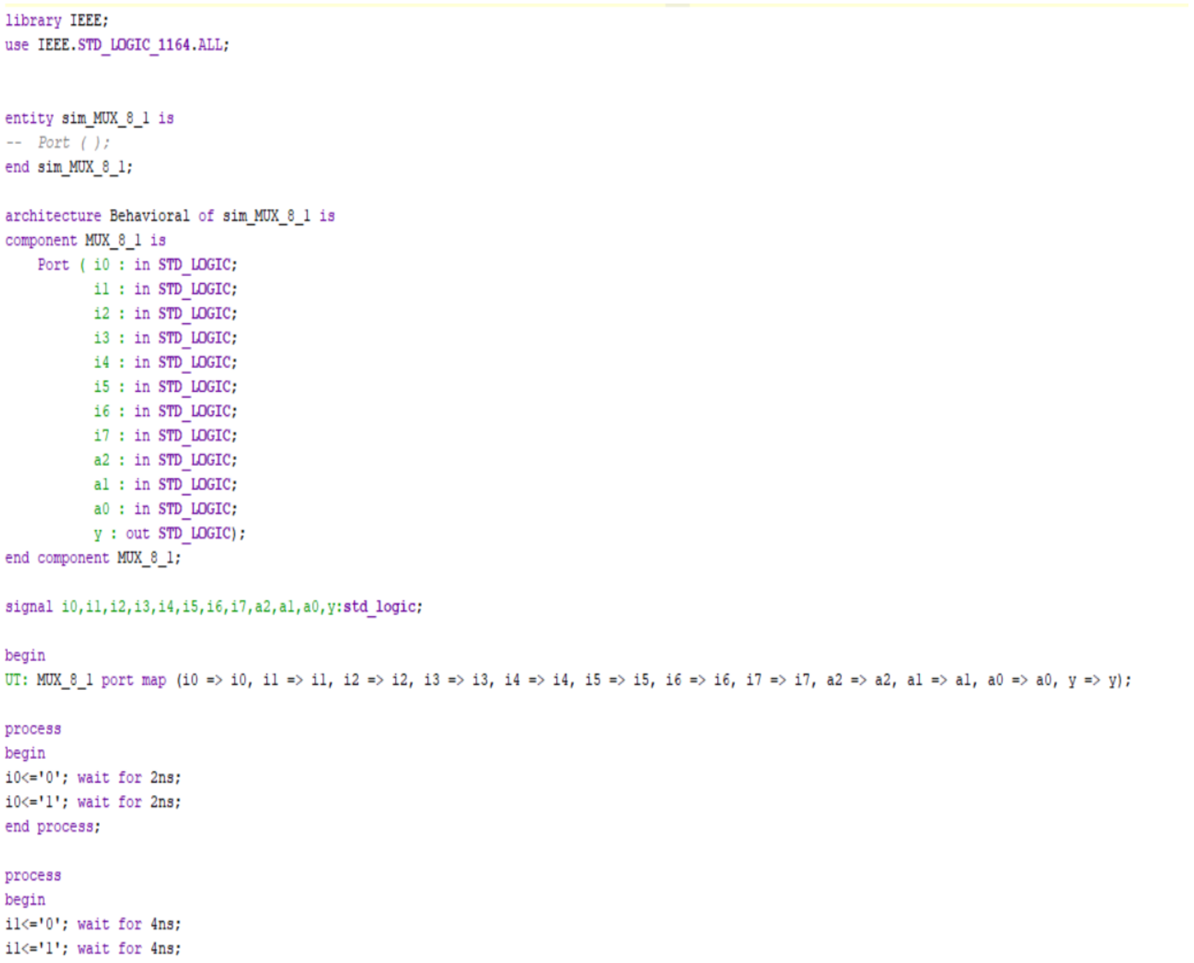


## ***Circuit Combinational MUX 8:1 ;***

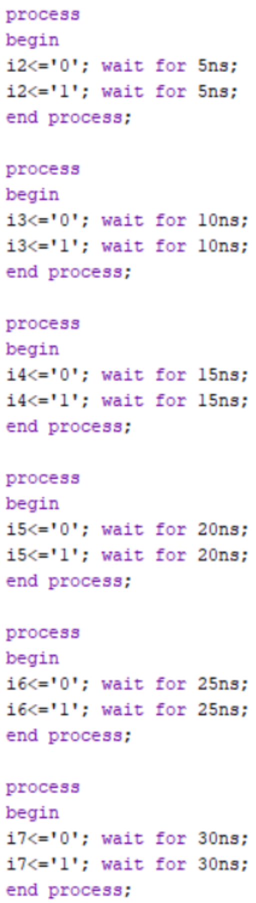


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using...

entity MUX_8_1 is
    Port ( i0 : in STD_LOGIC;
           i1 : in STD_LOGIC;
           i2 : in STD_LOGIC;
           i3 : in STD_LOGIC;
           i4 : in STD_LOGIC;
           i5 : in STD_LOGIC;
           i6 : in STD_LOGIC;
           i7 : in STD_LOGIC;
           a2 : in STD_LOGIC;
           a1 : in STD_LOGIC;
           a0 : in STD_LOGIC;
           y : out STD_LOGIC);
end MUX_8_1;

architecture Behavioral of MUX_8_1 is

signal a :std_logic_vector(2 downto 0);

begin
a<=a2 & a1 & a0;
 with a select

          Y <= i0 when "000",
                  i1 when "001",
                  i2 when "010",
                  i3 when "011",
                  i4 when "100",
                  i5 when "101",
                  i6 when "110",
                  i7 when others;

end Behavioral;
```

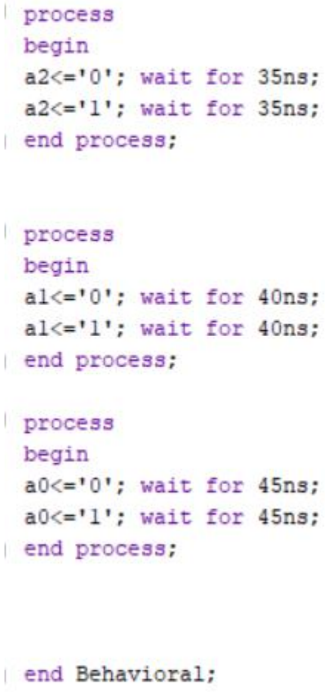
# *Simulare MUX 8:1:*



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


entity sim_MUX_8_1 is
--  Port ( );
end sim_MUX_8_1;

architecture Behavioral of sim_MUX_8_1 is
component MUX_8_1 is
    Port ( i0 : in STD_LOGIC;
           i1 : in STD_LOGIC;
           i2 : in STD_LOGIC;
           i3 : in STD_LOGIC;
           i4 : in STD_LOGIC;
           i5 : in STD_LOGIC;
           i6 : in STD_LOGIC;
           i7 : in STD_LOGIC;
           a2 : in STD_LOGIC;
           a1 : in STD_LOGIC;
           a0 : in STD_LOGIC;
           y : out STD_LOGIC);
end component MUX_8_1;

signal i0,i1,i2,i3,i4,i5,i6,i7,a2,a1,a0,y:std_logic;

begin
UT: MUX_8_1 port map (i0 => i0, i1 => i1, i2 => i2, i3 => i3, i4 => i4, i5 => i5, i6 => i6, i7 => i7, a2 => a2, a1 => a1, a0 => a0, y => y);

process
begin
i0<='0'; wait for 2ns;
i0<='1'; wait for 2ns;
end process;

process
begin
i1<='0'; wait for 4ns;
i1<='1'; wait for 4ns;
```

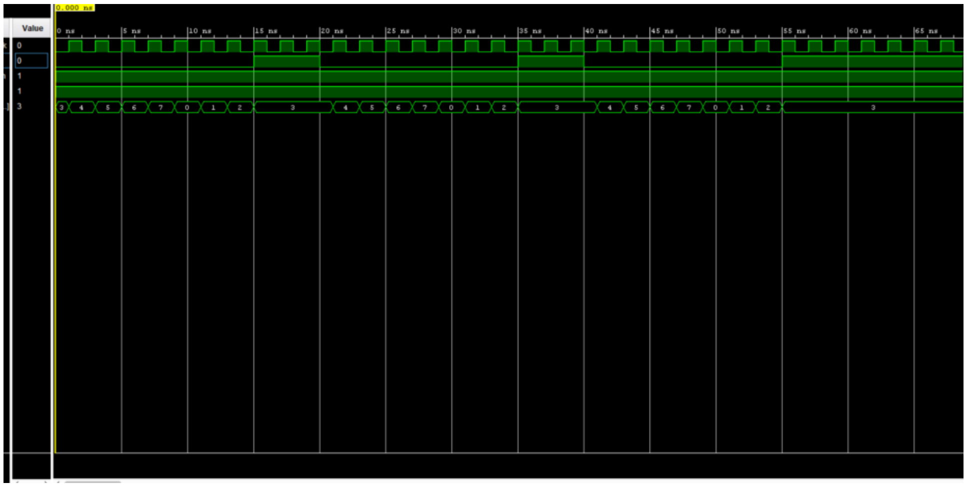


```
process
begin
i2<='0'; wait for 5ns;
i2<='1'; wait for 5ns;
end process;

process
begin
i3<='0'; wait for 10ns;
i3<='1'; wait for 10ns;
end process;

process
begin
i4<='0'; wait for 15ns;
i4<='1'; wait for 15ns;
end process;

process
begin
i5<='0'; wait for 20ns;
i5<='1'; wait for 20ns;
end process;

process
begin
i6<='0'; wait for 25ns;
i6<='1'; wait for 25ns;
end process;

process
begin
i7<='0'; wait for 30ns;
i7<='1'; wait for 30ns;
end process;
```

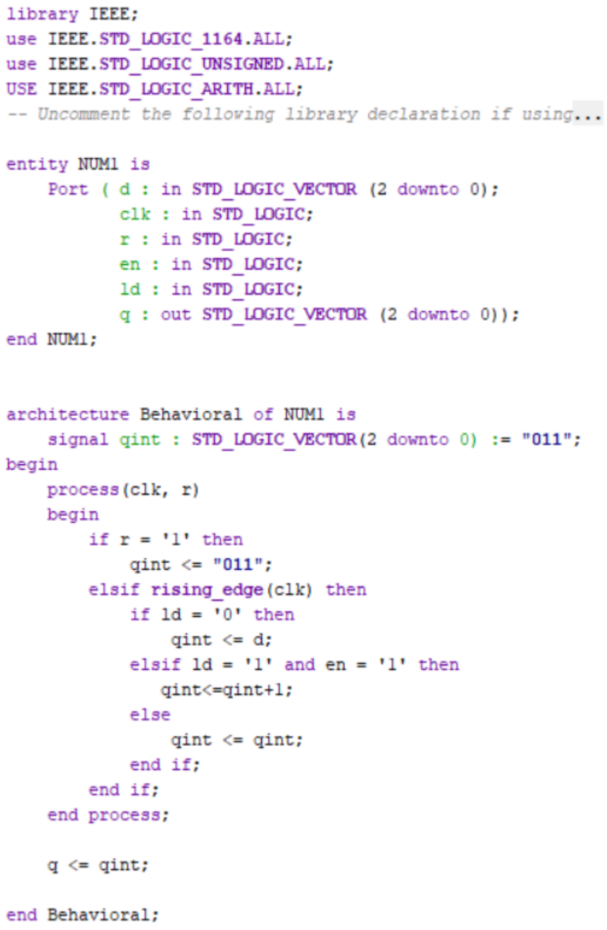


```
process
begin
a2<='0'; wait for 35ns;
a2<='1'; wait for 35ns;
end process;


process
begin
a1<='0'; wait for 40ns;
a1<='1'; wait for 40ns;
end process;

process
begin
a0<='0'; wait for 45ns;
a0<='1'; wait for 45ns;
end process;


end Behavioral;
```

# *Testare Numarator :*

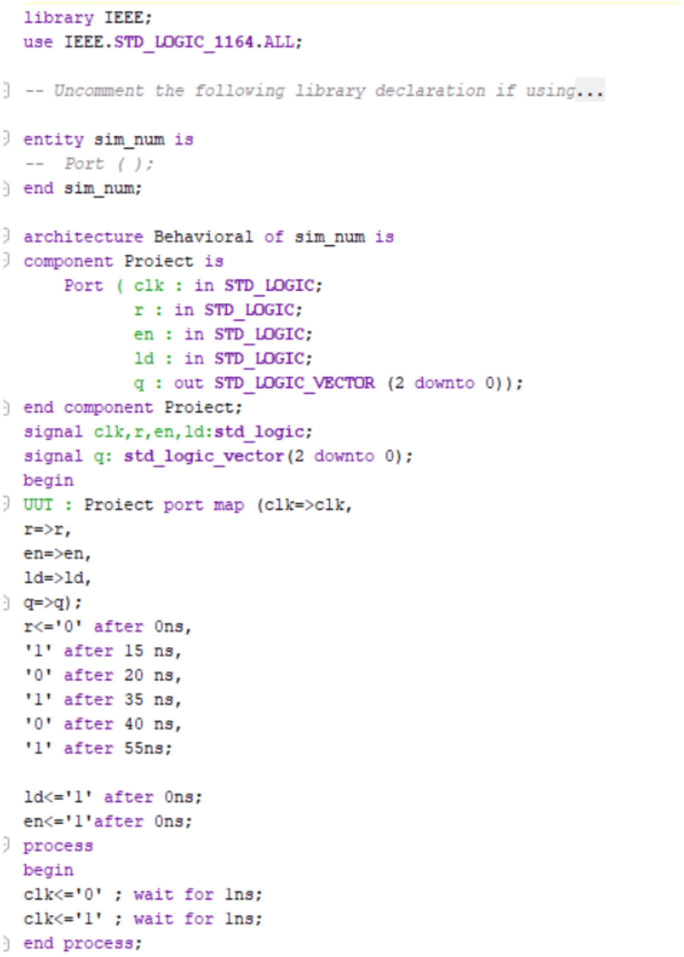


# *Numarator pe 3 Biti :*



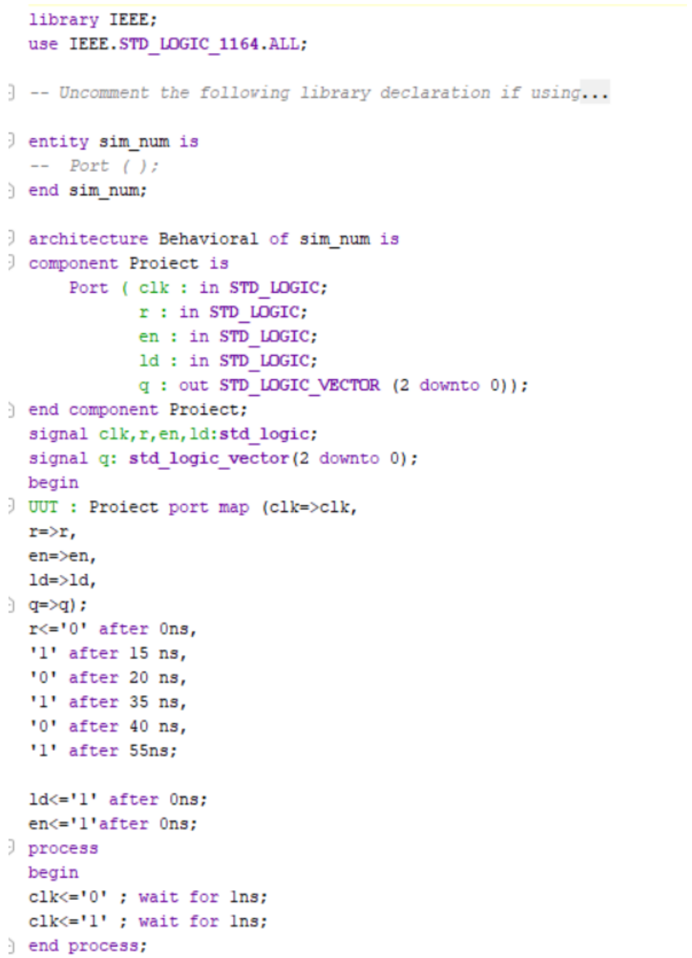
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
-- Uncomment the following library declaration if using...

entity NUM1 is
    Port ( d : in STD_LOGIC_VECTOR (2 downto 0);
           clk : in STD_LOGIC;
           r : in STD_LOGIC;
           en : in STD_LOGIC;
           ld : in STD_LOGIC;
           q : out STD_LOGIC_VECTOR (2 downto 0));
end NUM1;


architecture Behavioral of NUM1 is
    signal qint : STD_LOGIC_VECTOR(2 downto 0) := "011";
begin
    process(clk, r)
    begin
        if r = '1' then
            qint <= "011";
        elsif rising_edge(clk) then
            if ld = '0' then
                qint <= d;
            elsif ld = '1' and en = '1' then
               qint<=qint+1;
            else
                qint <= qint;
            end if;
        end if;
    end process;

    q <= qint;

end Behavioral;
```

# *Simulare Numaratorului :*



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

-- Uncomment the following library declaration if using...

entity sim_num is
--  Port ( );
end sim_num;

architecture Behavioral of sim_num is
component Proiect is
    Port ( clk : in STD_LOGIC;
           r : in STD_LOGIC;
           en : in STD_LOGIC;
           ld : in STD_LOGIC;
           q : out STD_LOGIC_VECTOR (2 downto 0));
end component Proiect;
signal clk,r,en,ld:std_logic;
signal q: std_logic_vector(2 downto 0);
begin
UUT : Proiect port map (clk=>clk,
r=>r,
en=>en,
ld=>ld,
q=>q);
r<='0' after 0ns,
'1' after 15 ns,
'0' after 20 ns,
'1' after 35 ns,
'0' after 40 ns,
'1' after 55ns;

ld<='1' after 0ns;
en<='1'after 0ns;
process
begin
clk<='0' ; wait for 1ns;
clk<='1' ; wait for 1ns;
end process;
```

# *Implementarea in vivado cu ajutorul limbajului de programare VHDL:*



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

-- Uncomment the following library declaration if using...

entity sim_num is
--  Port ( );
end sim_num;

architecture Behavioral of sim_num is
component Proiect is
    Port ( clk : in STD_LOGIC;
           r : in STD_LOGIC;
           en : in STD_LOGIC;
           ld : in STD_LOGIC;
           q : out STD_LOGIC_VECTOR (2 downto 0));
end component Proiect;
signal clk,r,en,ld:std_logic;
signal q: std_logic_vector(2 downto 0);
begin
UUT : Proiect port map (clk=>clk,
r=>r,
en=>en,
ld=>ld,
q=>q);
r<='0' after 0ns,
'1' after 15 ns,
'0' after 20 ns,
'1' after 35 ns,
'0' after 40 ns,
'1' after 55ns;

ld<='1' after 0ns;
en<='1'after 0ns;
process
begin
clk<='0' ; wait for 1ns;
clk<='1' ; wait for 1ns;
end process;
```