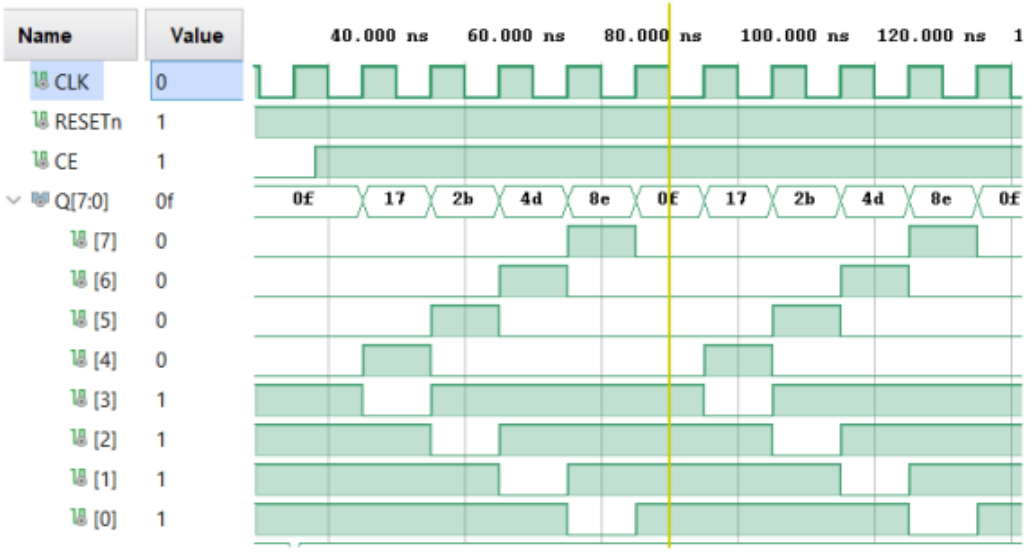
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

SFPGA

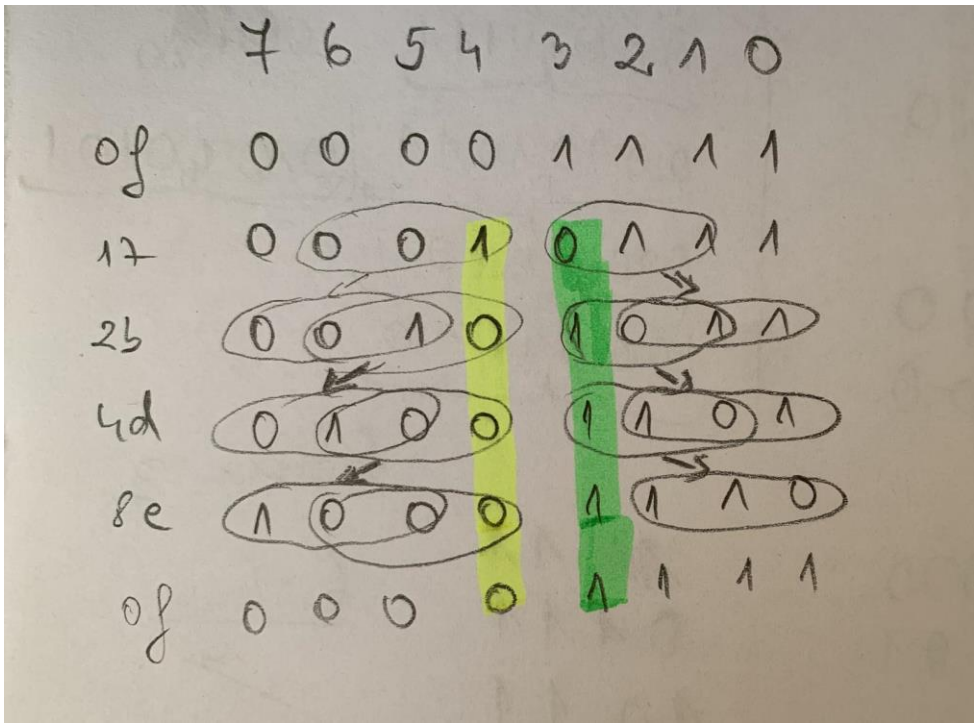
SECVENTA LFSR

Student: Achiriloaei Ioana Isabela
Profesor îndrumător: Albert Csaba Fazakas

V)



Walk 1 OUT to left, Walk0 OUT to right. States **INCLUDE 0F**



Cod:

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 10/20/2020 11:18:02 AM
-- Design Name:
-- Module Name: DataConcat4To12 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
use IEEE.std_logic_arith.all;

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

entity Pattern_Gen is
      Port ( CLK        : in        STD_LOGIC                                ;
               RESETn    : in  STD_LOGIC                              ;
               CE        : in  STD_LOGIC                              ;
           Q         : out   STD_LOGIC_VECTOR (7 downto 0)
    );
end Pattern_Gen;

architecture Behavioral of Pattern_Gen is
constant INIT_VAL : std_logic_vector ( 7 downto 0) := X"17";
constant END_VAL : std_logic_vector ( 7 downto 0) := X"0F";
signal Q_Intern : STD_LOGIC_VECTOR (7 downto 0):= X"0F";

begin
Q <= Q_Intern;

process (CLK, RESETn, Q_intern)
begin
if rising_edge(CLK) then
```

```
      if RESETn = '0' then
            Q_intern <= INIT_VAL;
      elsif CE = '1' then
            if Q_Intern = END_VAL then
                  Q_Intern <= INIT_VAL;
            else
               Q_Intern(7 downto 0) <=  Q_Intern (6 downto 4) & '0' & '1' &
Q_Intern (3 downto 1);
            end if;
      end if;
end if;
end process;

end Behavioral;
```

Simulare: