**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 3

la cursul ***„Proiectarea cu dispozitive programabile”***

**A efectuat:**  **St. gr. CR-221FR Serba Cristina**

**A verificat: Cărbune Viorel**

**Chișinău 2025**

# Tema:

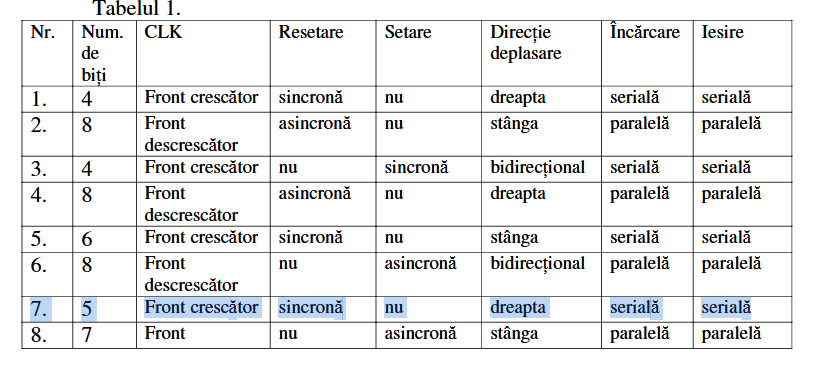
Sinteza si implementarea circuitelor logice secvențiale

# Scopul lucrării:

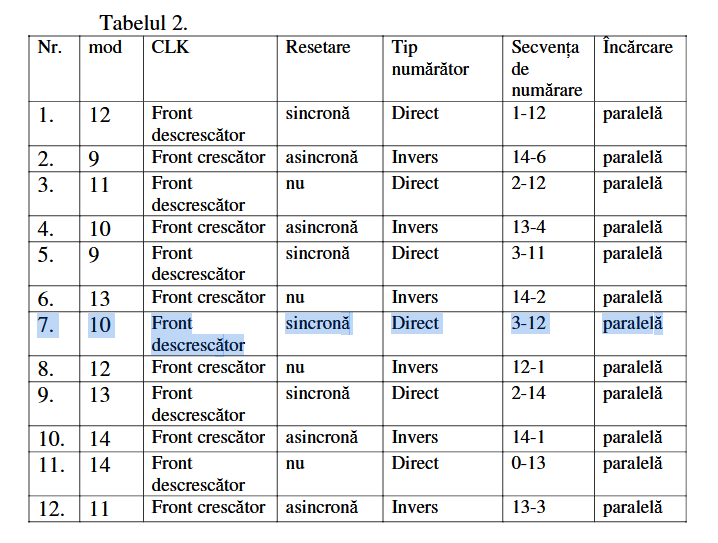
Proiectarea, testarea şi simularea circuitelor logice secvențiale în mediul de dezvoltare software Altera Quartus II. Descrierea circuitelori va fi efectuată în limbajul VHDL, folosind codificarea comportamentală.

# Sarcina lucrării:

1. Se va proiecta un registru conform variantei din tabelul 1.



1. Se va proiecta un numărător conform variantei din tabelul 2.



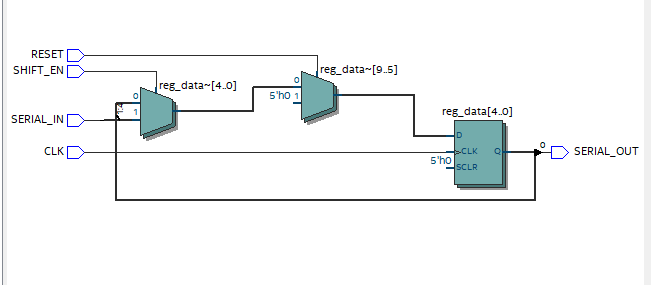
# Mersul lucrării:

Partea 1

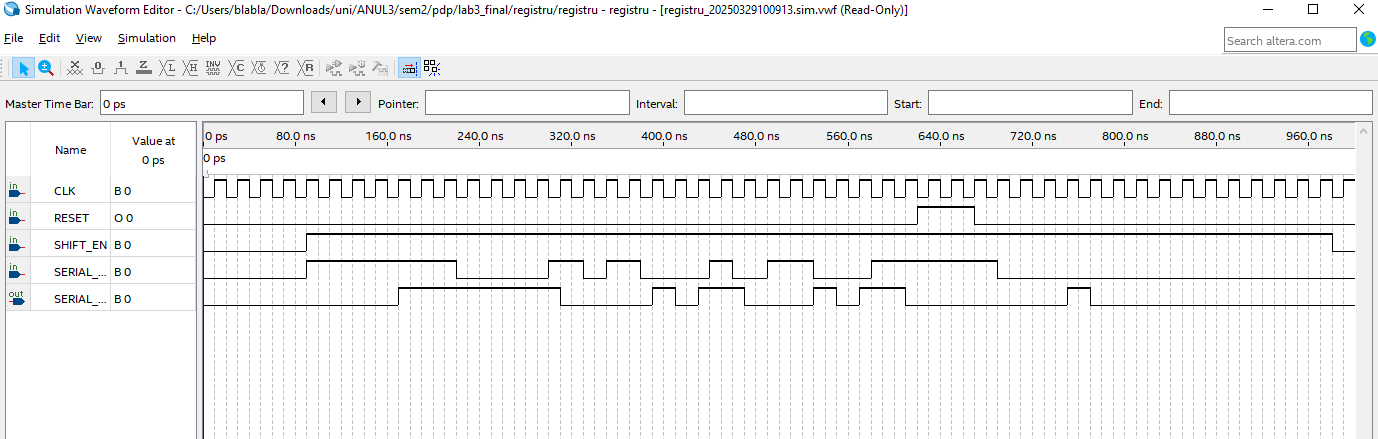
Proiectarea registrului:

|  |
| --- |
| library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  entity registru is  port (  CLK : in std\_logic;  RESET : in std\_logic;  SHIFT\_EN : in std\_logic;  SERIAL\_IN : in std\_logic;  SERIAL\_OUT: out std\_logic  );  end registru;  architecture Behavioral of registru is  signal reg\_data: std\_logic\_vector(4 downto 0);  begin  process (CLK, RESET)  begin  if rising\_edge(CLK) then --front crescator al ceasului  if RESET = '1' then  reg\_data <= (others => '0');  elsif SHIFT\_EN = '1' then  for i in 1 to 4 loop  reg\_data(i-1) <= reg\_data(i); -- deplasare la dreapta  end loop;  reg\_data(4) <= SERIAL\_IN; -- incarcare seriala  end if;  end if;  end process;  SERIAL\_OUT <= reg\_data(0); --iesire seriala, ultimul bit  end Behavioral; |

RTL view:



Waveform simulation:

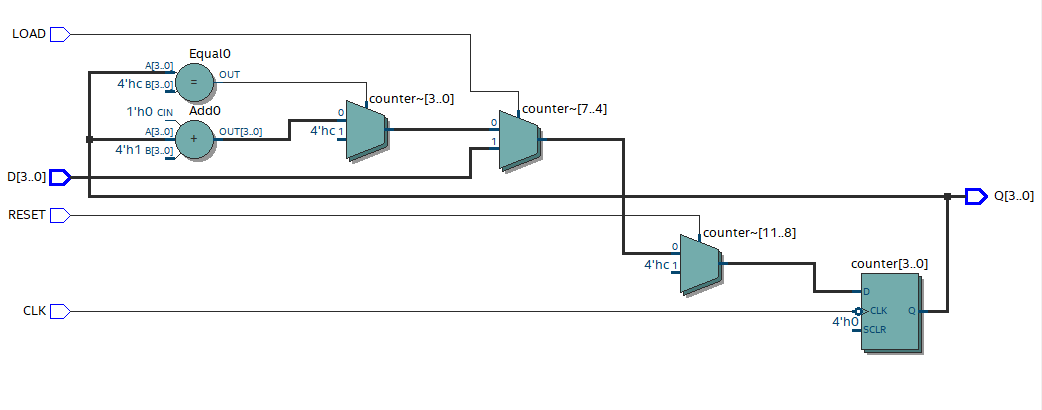


Partea 2

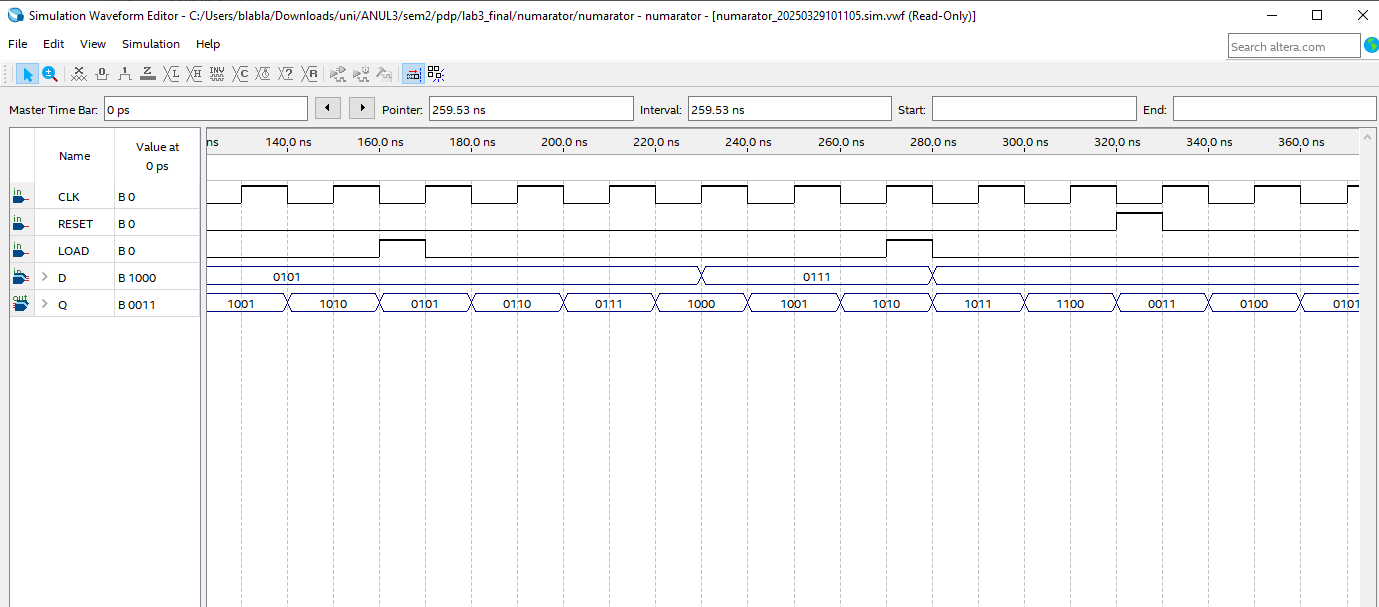
Proiectarea numărătorului:

|  |
| --- |
| library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  use IEEE.STD\_LOGIC\_ARITH.ALL;  use IEEE.STD\_LOGIC\_UNSIGNED.ALL;  entity numarator is  port (  CLK : in std\_logic;  RESET : in std\_logic;  LOAD : in std\_logic;  D : in std\_logic\_vector(3 downto 0);  Q : out std\_logic\_vector(3 downto 0)  );  end numarator;  architecture Behavioral of numarator is  signal counter : std\_logic\_vector(3 downto 0) := "0011"; -- asignam 3 in binar  begin  process(CLK, RESET)  begin  if falling\_edge(CLK) then --resetare sincrona cu falling edge  if RESET = '1' then  counter <= "0011"; --reseteaza la 3  elsif LOAD = '1' then  counter <= D; -- incarca paralel  else  if counter = "1100" then  counter <= "0011";  else  counter <= counter + 1;  end if;  end if;  end if;  end process;  Q <= counter;  end Behavioral; |

RTL view:



Simularea în waveform:



# Concluzii:

În urma realizării acestei lucrări, am dobândit cunoștințe esențiale despre proiectarea și simularea registrelor și numărătoarelor, componente fundamentale ale circuitelor secvențiale. Implementarea acestor structuri conform variantelor specificate ne-a permis să explorăm atât principiile teoretice, cât și aplicarea practică în mediul de dezvoltare. Prin testare și simulare, am verificat funcționalitatea circuitelor și am înțeles importanța sincronizării și optimizării acestora pentru performanță și fiabilitate. Această experiență a contribuit la consolidarea competențelor necesare în proiectarea circuitelor digitale secvențiale.