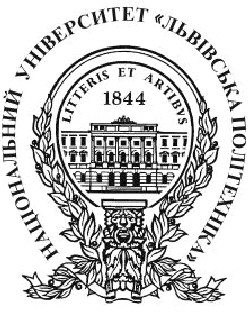
Міністерство освіти і науки України НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Кафедра ЕОМ

З лабораторної роботи № 2

З дисципліни « МОДЕЛЮВАННЯ КОМПЮТЕРНИХ СИСТЕМ »

На тему: « Структурний опис цифрового автомата. Перевірка роботи автомата за допомогою стенда Elbert V2-Spartan 3A FPGA»

Варіант X

Виконав: студент гр.

Прийняв:

Львів – 2023

# Мета роботи:

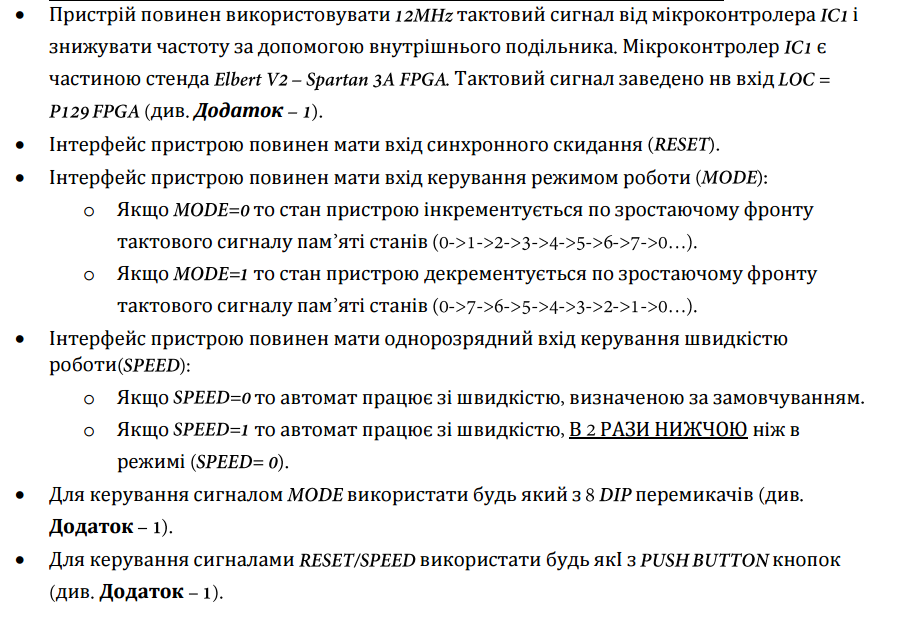
На базі стенда Elbert V2 – Spartan 3A FPGA реалізувати цифровий автомат світлових ефектів згідно наступних вимог:

1. Інтерфейс пристрою та функціонал реалізувати згідно отриманого варіанту завдання
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів VHDL. Заборонено використовувати оператори if, switch, for, when
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів VHDL. Заборонено використовувати оператори if, switch, for, when
4. Згенерувати Schematic символи для VHDL описів логіки переходів та логіки формування вихідних сигналів
5. Зінтегрувати всі компоненти (логіку переходів логіку формування вихідних сигналів та пам'ять станів) в єдину систему за допомогою ISE WebPACK Schematic Capture. Пам'ять станів реалізувати за допомогою графічних компонентів з бібліотеки
6. Промоделювати роботу окремих частин автомата та автомата в цілому за допомогою симулятора ISim
7. Інтегрувати створений автомат зі стендом Elbert V2 – Spartan 3A FPGA (додати подільник частоти для вхідного тактового сигналу призначити фізичні виводи на FPGA)
8. Згенерувати BIT файл та перевірити роботу за допомогою стенда Elbert V2 – Spartan 3A FPGA
9. Підготувати і захистити звіт

**Завдання:**

Мій номер по списку 20, тому номер варіант 2.





# Виконання роботи:

# (VHDL OutputLogic)

# ----------------------------------------------------------------------------------

# -- Company:

# -- Engineer:

# --

# -- Create Date: 17:41:48 03/26/2021

# -- Design Name:

# -- Module Name: out\_logic\_intf\_2v - out\_logic\_arch\_2v

# -- Project Name:

# -- Target Devices:

# -- Tool versions:

# -- Description:

# --

# -- Dependencies:

# --

# -- Revision:

# -- Revision 0.01 - File Created

# -- Additional Comments:

# --

# ----------------------------------------------------------------------------------

# library IEEE;

# use IEEE.STD\_LOGIC\_1164.ALL;

# entity out\_logic\_intf\_2v is

# port ( in\_bus : in std\_logic\_vector(2 downto 0) ;

# out\_bus : out std\_logic\_vector(7 downto 0));

# end out\_logic\_intf\_2v;

# architecture out\_logic\_arch\_2v of out\_logic\_intf\_2v is

# signal A\_OUT, B\_OUT, C\_OUT, D\_OUT,

# E\_OUT, F\_OUT, G\_OUT, H\_OUT : std\_logic;

# begin

# A\_OUT <= (not(in\_bus(2)) and not(in\_bus(1)) and not(in\_bus(0)));

# B\_OUT <= (not(in\_bus(2)) and not(in\_bus(1)) and in\_bus(0) );

# C\_OUT <= (not(in\_bus(2)) and in\_bus(1) and not(in\_bus(0)));

# D\_OUT <= (not(in\_bus(2)) and in\_bus(1) and in\_bus(0) );

# E\_OUT <= ( in\_bus(2) and not(in\_bus(1)) and not(in\_bus(0)));

# F\_OUT <= ( in\_bus(2) and not(in\_bus(1)) and in\_bus(0) );

# G\_OUT <= ( in\_bus(2) and in\_bus(1) and not(in\_bus(0)));

# H\_OUT <= ( in\_bus(2) and in\_bus(1) and in\_bus(0) );

# out\_bus(0) <= A\_OUT or B\_OUT or C\_OUT or D\_OUT or E\_OUT or F\_OUT or G\_OUT or H\_OUT;

# out\_bus(1) <= B\_OUT or C\_OUT or D\_OUT or E\_OUT or F\_OUT or G\_OUT or H\_OUT;

# out\_bus(2) <= C\_OUT or D\_OUT or E\_OUT or F\_OUT or G\_OUT or H\_OUT;

# out\_bus(3) <= D\_OUT or E\_OUT or F\_OUT or G\_OUT or H\_OUT;

# out\_bus(4) <= E\_OUT or F\_OUT or G\_OUT or H\_OUT;

# out\_bus(5) <= F\_OUT or G\_OUT or H\_OUT;

# out\_bus(6) <= G\_OUT or H\_OUT;

# out\_bus(7) <= H\_OUT;

# end out\_logic\_arch\_2v;

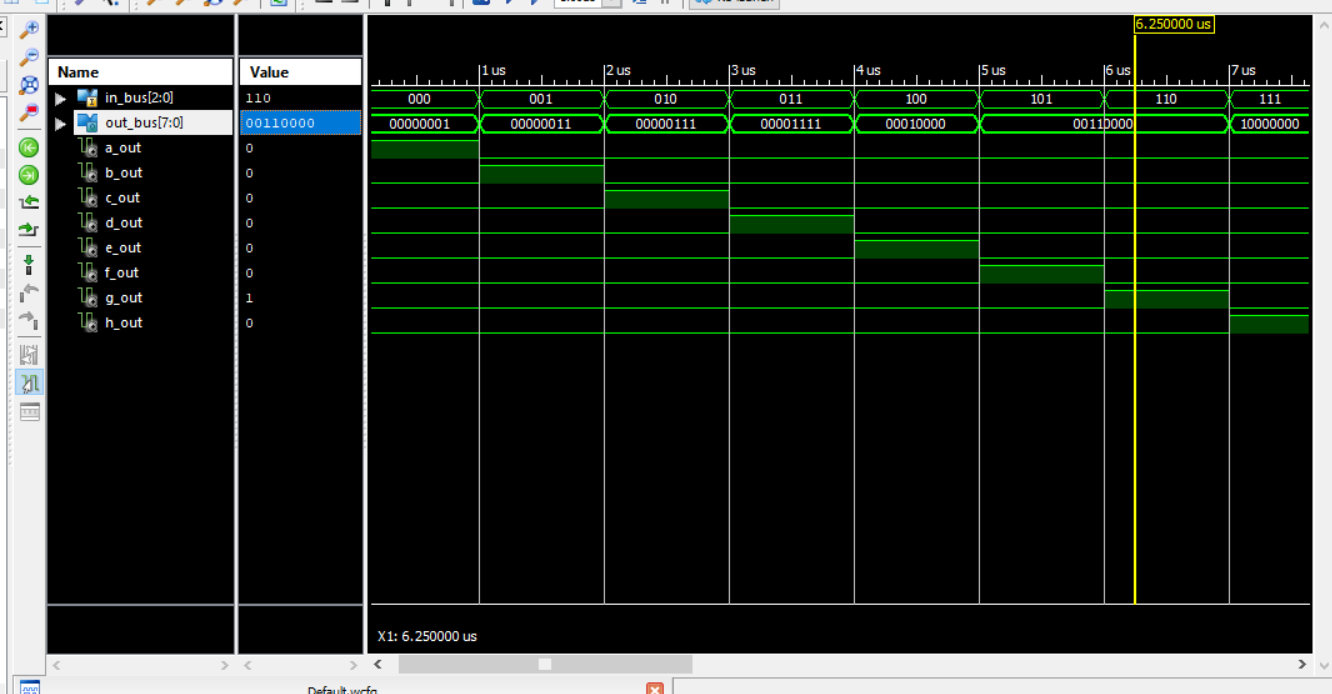


Рис.1: виконання роботи схеми

# (VHDL TransitionLogic)

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity transition\_logic\_intf\_2v is

Port ( CUR\_STATE : in std\_logic\_vector(2 downto 0) ;

MODE : in std\_logic;

NEXT\_STATE : out std\_logic\_vector(2 downto 0));

end transition\_logic\_intf\_2v;

architecture transition\_logic\_arch\_2v of transition\_logic\_intf\_2v is

begin

NEXT\_STATE(0) <= (not(MODE) and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or

(not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

(not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or

(not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

( MODE and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

( MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or

( MODE and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

( MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) ;

NEXT\_STATE(1) <= (not(MODE) and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or

(not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

(not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or

(not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

( MODE and not(CUR\_STATE(2)) and CUR\_STATE(1) and CUR\_STATE(0)) or

( MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or

( MODE and CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0)) or

( MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) ;

NEXT\_STATE(2) <= (not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and CUR\_STATE(0)) or

(not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or

(not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or

(not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

( MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or

( MODE and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or

( MODE and CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0)) or

( MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) ;

end transition\_logic\_arch\_2v;



Рис.2: промодельована робота схеми формування вихідних сигналів з усіма можливими наборами сигналів

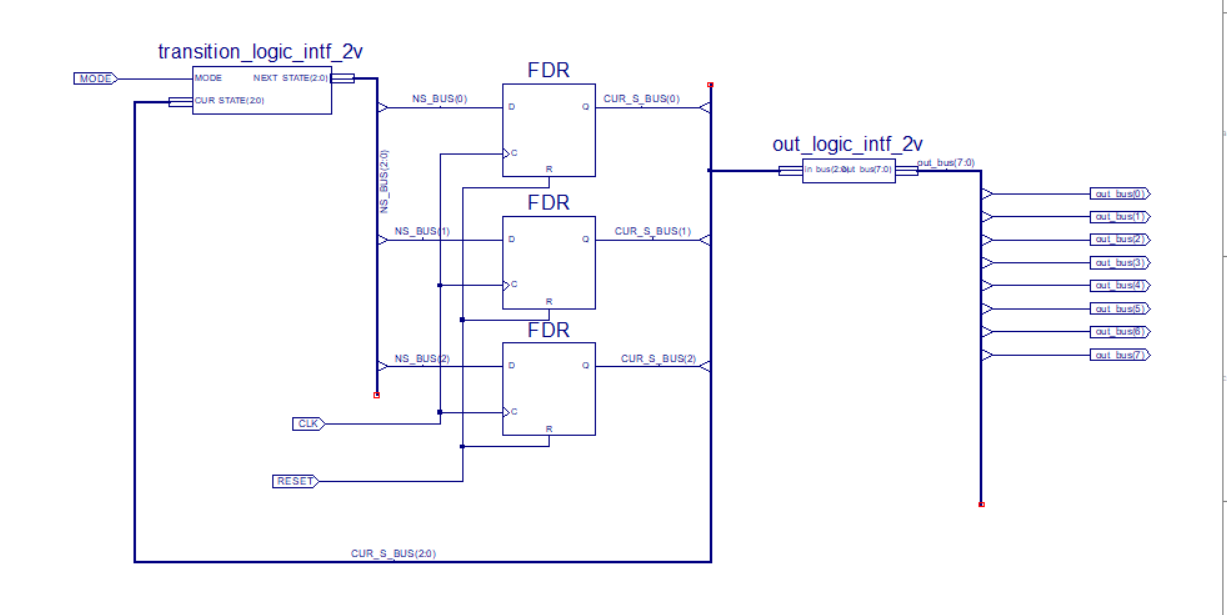


Рис.3: Schematic файл (LightController.sch)

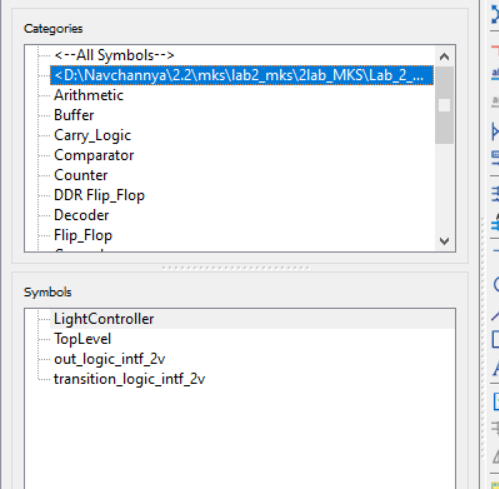


Рис.4: згенеровані Schematic символи

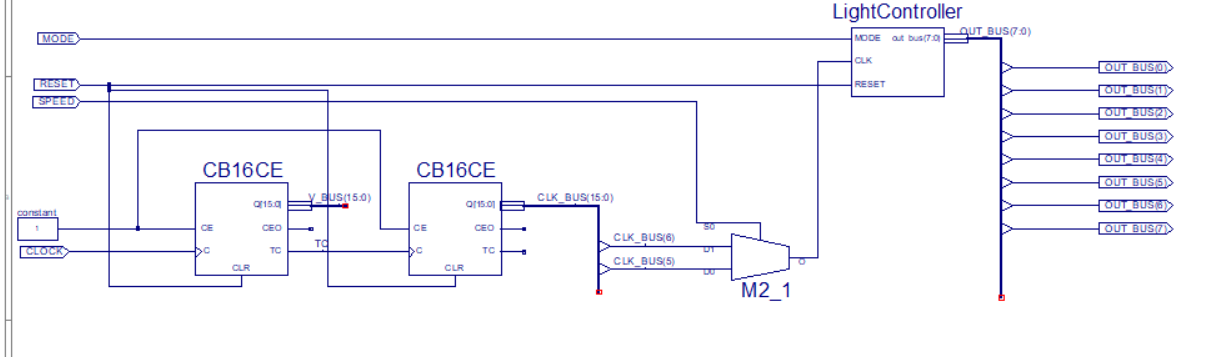


Рис.5: Schematic файл для кінцевої схеми.(TopLevel.sch)

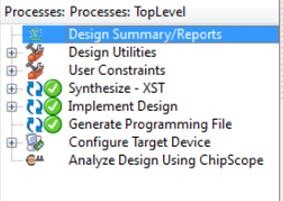


Рис.6: згенерований бінарний файл

**Висновок:** на лабораторній роботі на базі стенда Elbert V2- Spartan 3A FPGA реалізував цифровий автомат світлових ефектів. Навчився створювати нові елементи і описувати логіку їх роботи засобами VHDL.