Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “ЕОМ”



**Звіт з лабораторної роботи № 2**

Виконав:

ст.гр. КІ-202

Куйбіда Н.В.

Прийняв

Ст.викладач

Козак Н.Б.

Львів – 2023

**Тема:** Структурний опис цифрового автомата. Перевірка роботи автомата за допомогою стенда Elbert V2-Spartan 3A FPGA

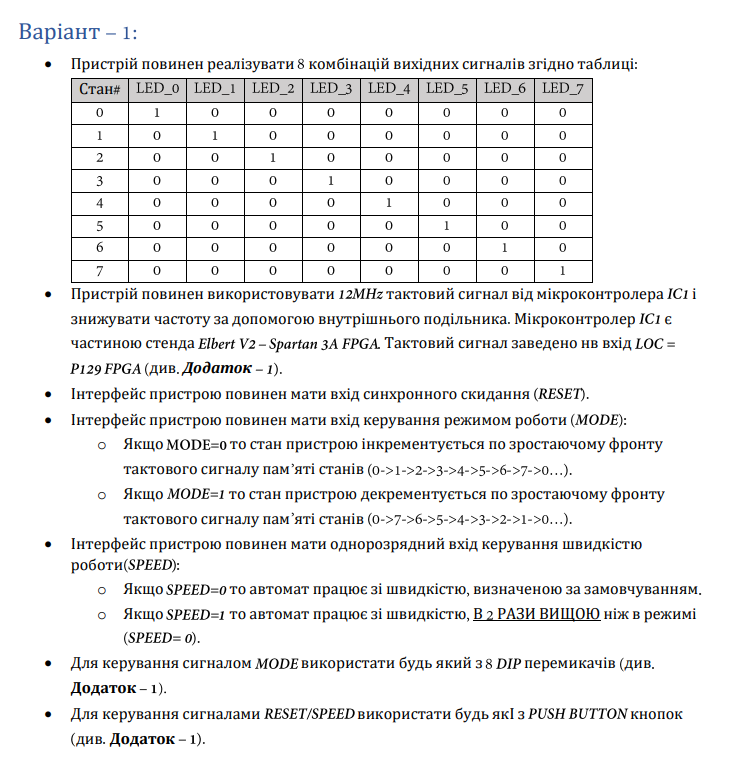
**Мета роботи:**

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

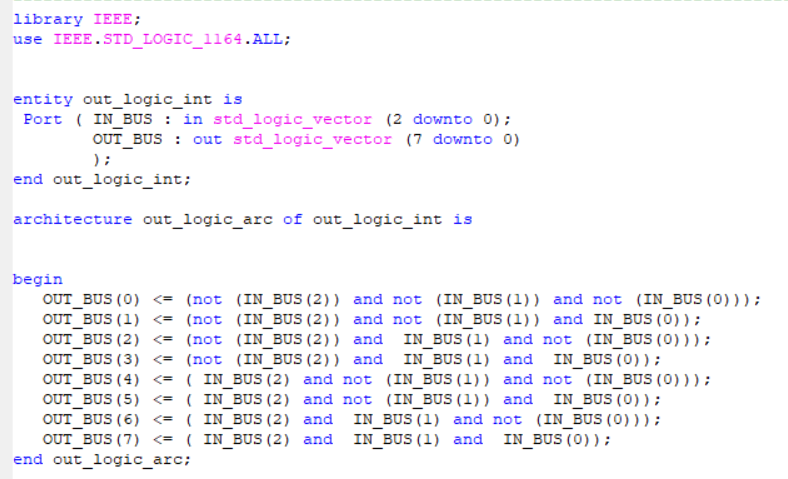
1. Інтерфейс пристрою та функціонал реалізувати згідно отриманого варіанту завдання
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів VHDL. Заборонено використовувати оператори if, switch, for, when
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису 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:

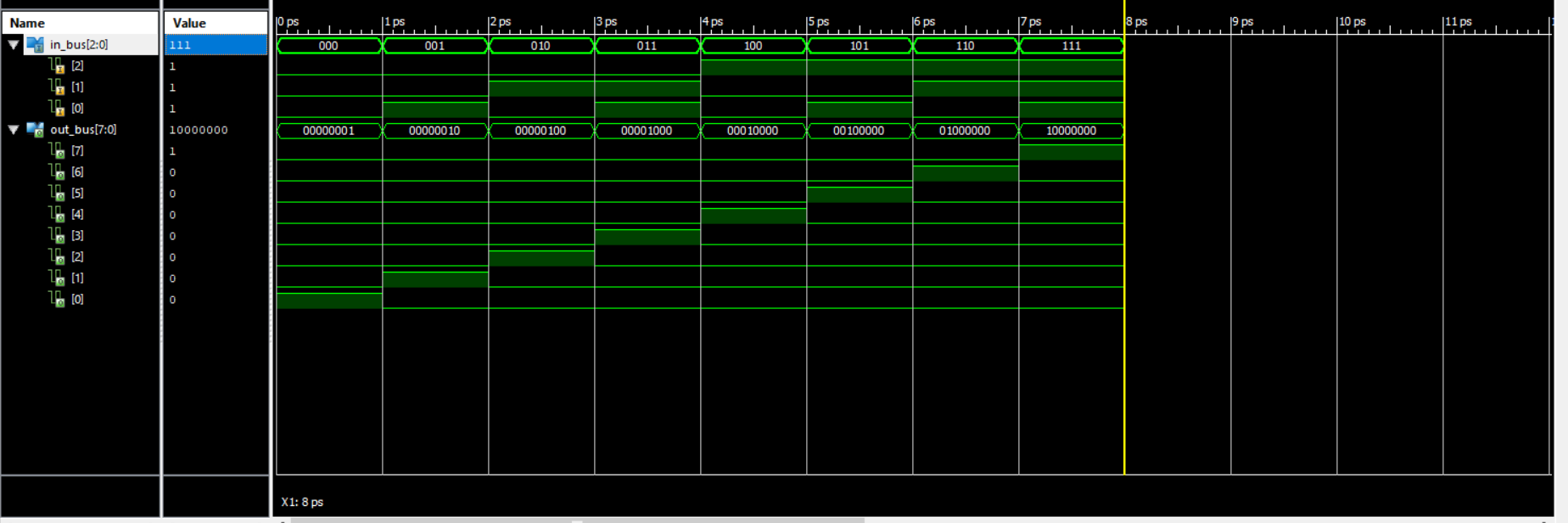
Варіант 7



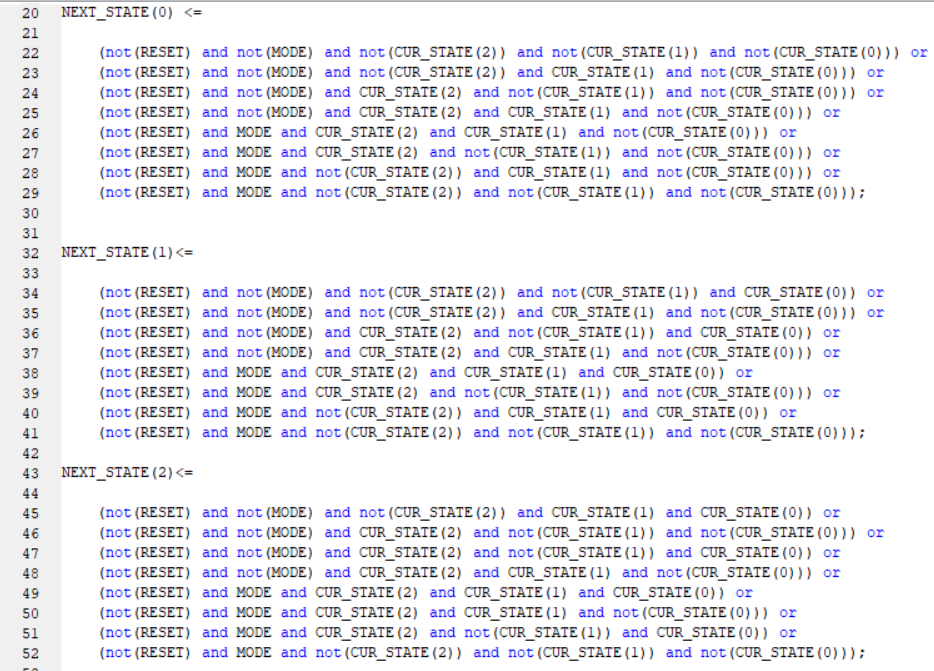
**Виконання роботи:**



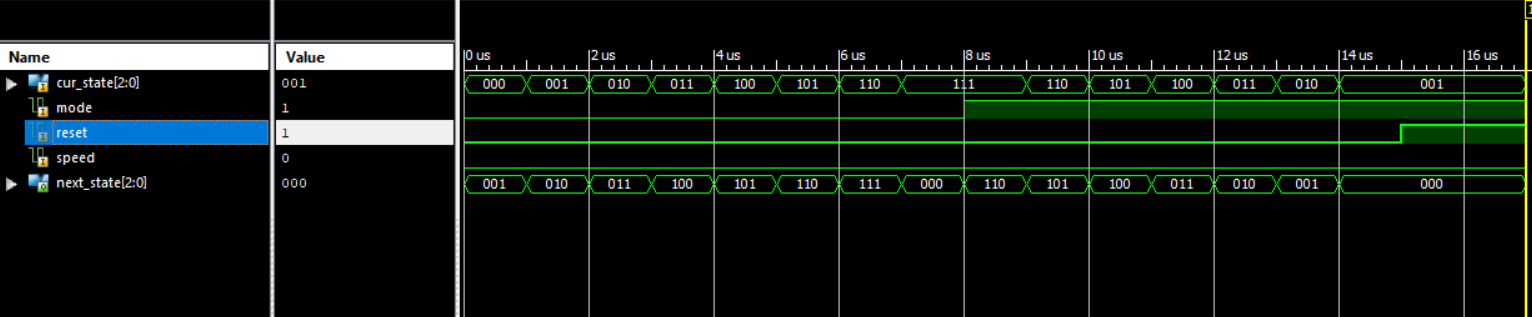
*Рис.1 VHDL файл, який реалізує логіку формування сигналів (OutputLogic.vhd).*



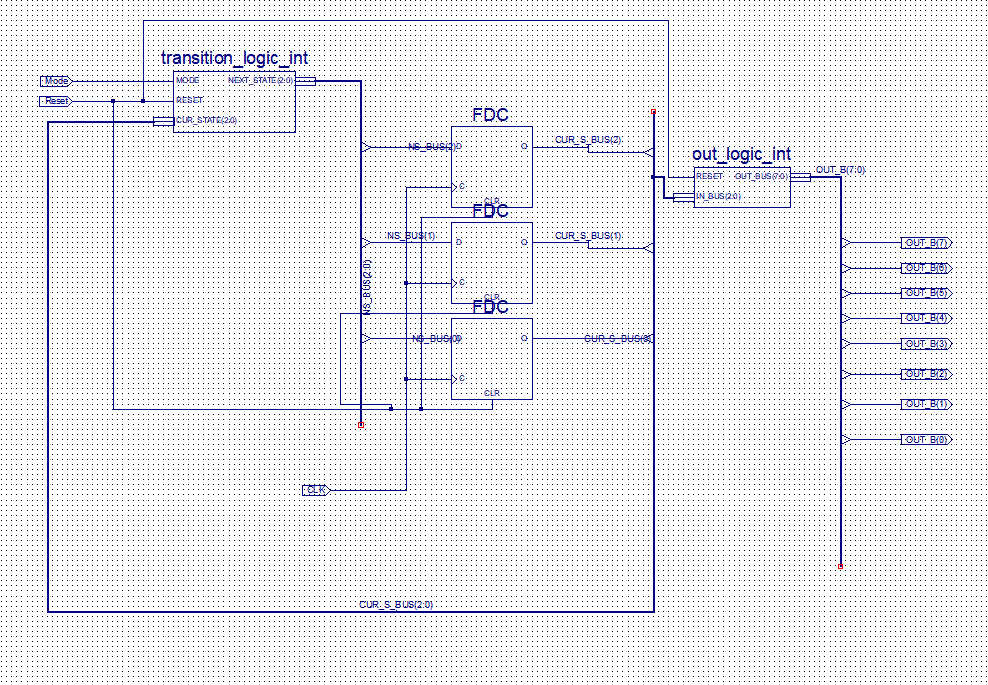
*Рис.2 Промодельована робота схеми формування вихідних сигналів з усіма можливими наборами сигналів.*



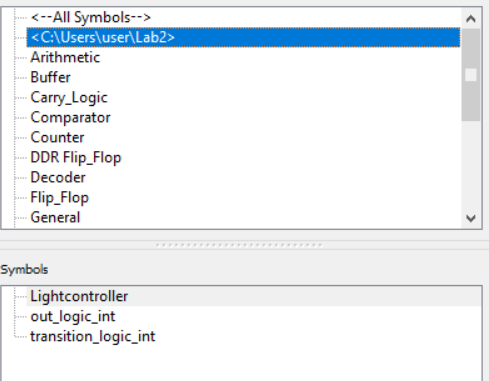
*Рис.3 VHDL файл, який реалізує логіку формування переходів стану автомата(TransitionLogic.vhd).*



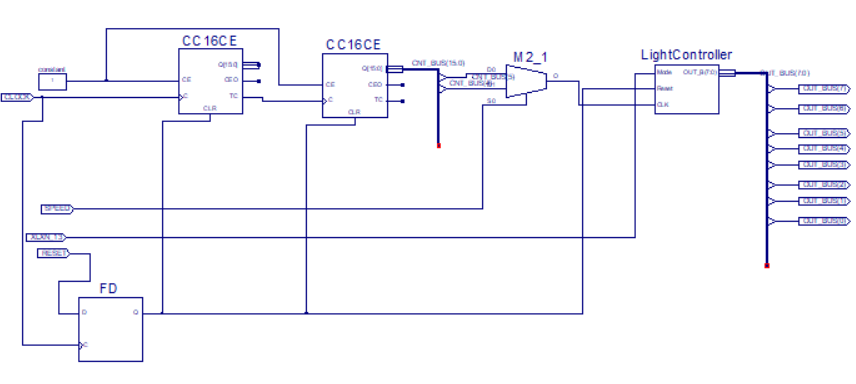
*Рис.4 Промодельована роботу схеми формування вихідних сигналів з усіма можливими наборами сигналів.*



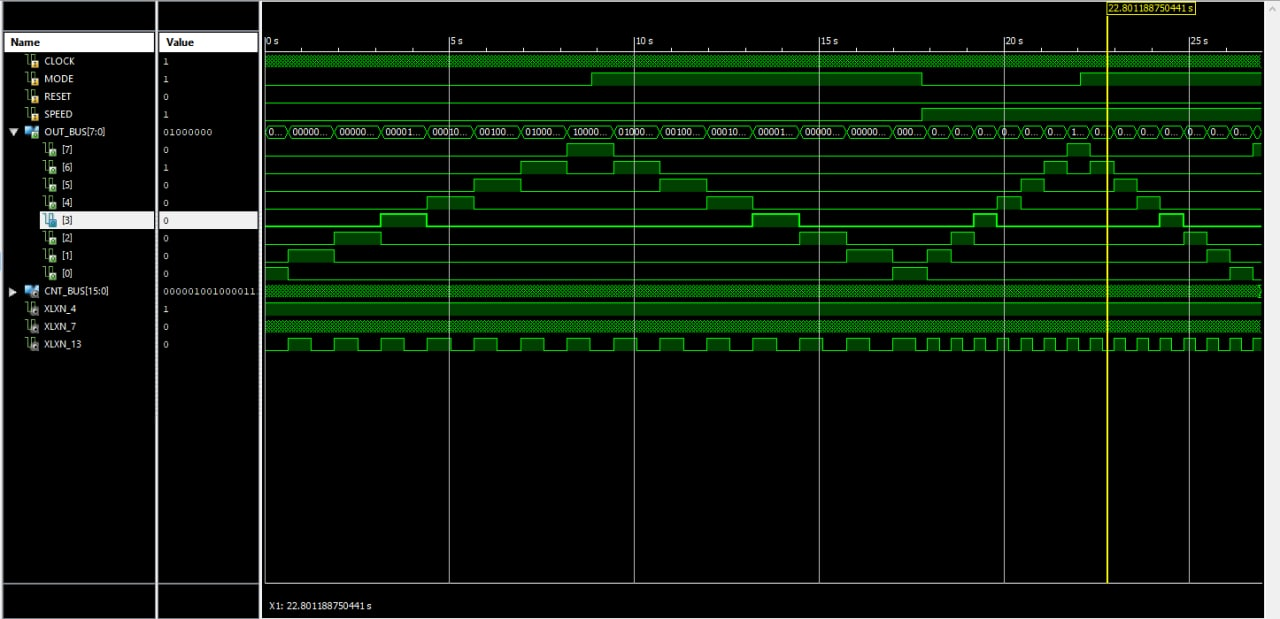
*Рис.5 Schematic файл (LightController.sch), в якому реалізована пам’ять стану автомата.*



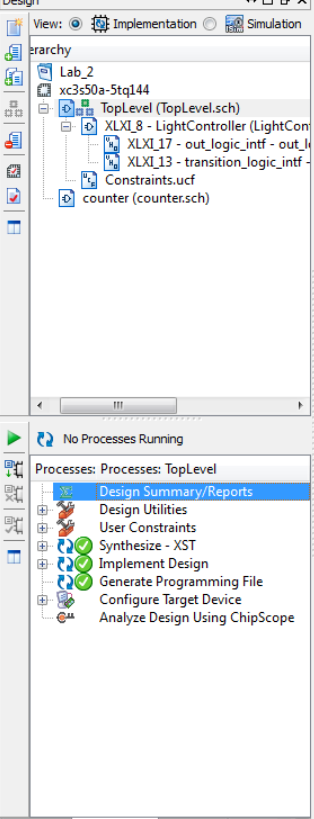
*Рис.6 Згенеровані Schematic символи.*



*Рис.7 Schematic файл для кінцевої схеми.(TopLevel.sch). Реалізований подільник вхідної частоти.*



*Рис.8 Симуляція системи різними значеннями сигналів MODE/RESET/SPEED.*



*Рис.9 Згенерований бінарний файл.*

*Лістинг1 (VHDL OutputLogic)*

|  |
| --- |
| library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  entity out\_logic\_int is  Port ( IN\_BUS : in std\_logic\_vector(2 downto 0);  OUT\_BUS : out std\_logic\_vector(7 downto 0);  RESET:in std\_logic  );  end out\_logic\_int;  architecture out\_logic\_arc of out\_logic\_int is  begin  OUT\_BUS(0) <= (not(RESET) and not (IN\_BUS(2)) and not (IN\_BUS(1)) and not(IN\_BUS(0)));  OUT\_BUS(1) <= (not(RESET) and not (IN\_BUS(2)) and not (IN\_BUS(1)) and IN\_BUS(0));  OUT\_BUS(2) <= (not(RESET) and not (IN\_BUS(2)) and IN\_BUS(1) and not (IN\_BUS(0)));  OUT\_BUS(3) <= (not(RESET) and not (IN\_BUS(2)) and IN\_BUS(1) and IN\_BUS(0));  OUT\_BUS(4) <= (not(RESET) and IN\_BUS(2) and not (IN\_BUS(1)) and not (IN\_BUS(0)));  OUT\_BUS(5) <= (not(RESET) and IN\_BUS(2) and not (IN\_BUS(1)) and IN\_BUS(0));  OUT\_BUS(6) <= (not(RESET) and IN\_BUS(2) and IN\_BUS(1) and not (IN\_BUS(0)));  OUT\_BUS(7) <= (not(RESET) and IN\_BUS(2) and IN\_BUS(1) and IN\_BUS(0));  end out\_logic\_arc; |

*Лістинг2 (VHDL TransitionLogic)*

|  |
| --- |
| library IEEE;  use IEEE.STD\_LOGIC\_1164.ALL;  entity transition\_logic\_int is  Port ( CUR\_STATE : in std\_logic\_vector(2 downto 0);  MODE : in std\_logic;  RESET : in std\_logic;    NEXT\_STATE : out std\_logic\_vector(2 downto 0)  );    end transition\_logic\_int;  architecture transition\_logic\_arc of transition\_logic\_int is  begin    NEXT\_STATE(0) <=  (not(RESET) and not(MODE) and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or  (not(RESET) and not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or    (not(RESET) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or  (not(RESET) and MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or  (not(RESET) and MODE and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or  (not(RESET) and MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0)));          NEXT\_STATE(1)<=  (not(RESET) and not(MODE) and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or  (not(RESET) and not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or      (not(RESET) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0)) or  (not(RESET) and MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or  (not(RESET) and MODE and not(CUR\_STATE(2)) and CUR\_STATE(1) and CUR\_STATE(0)) or  (not(RESET) and MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0)));    NEXT\_STATE(2)<=  (not(RESET) and not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and CUR\_STATE(0)) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or  (not(RESET) and not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or    (not(RESET) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0)) or  (not(RESET) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or  (not(RESET) and MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or  (not(RESET) and MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0)));      end transition\_logic\_arc; |

**Висновок:** На даній 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.