Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №2

з дисципліни: “Моделювання комп’ютерних систем”

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

Виконав: ст.гр. КІ-201

Терлецький А.А.

Прийняв:

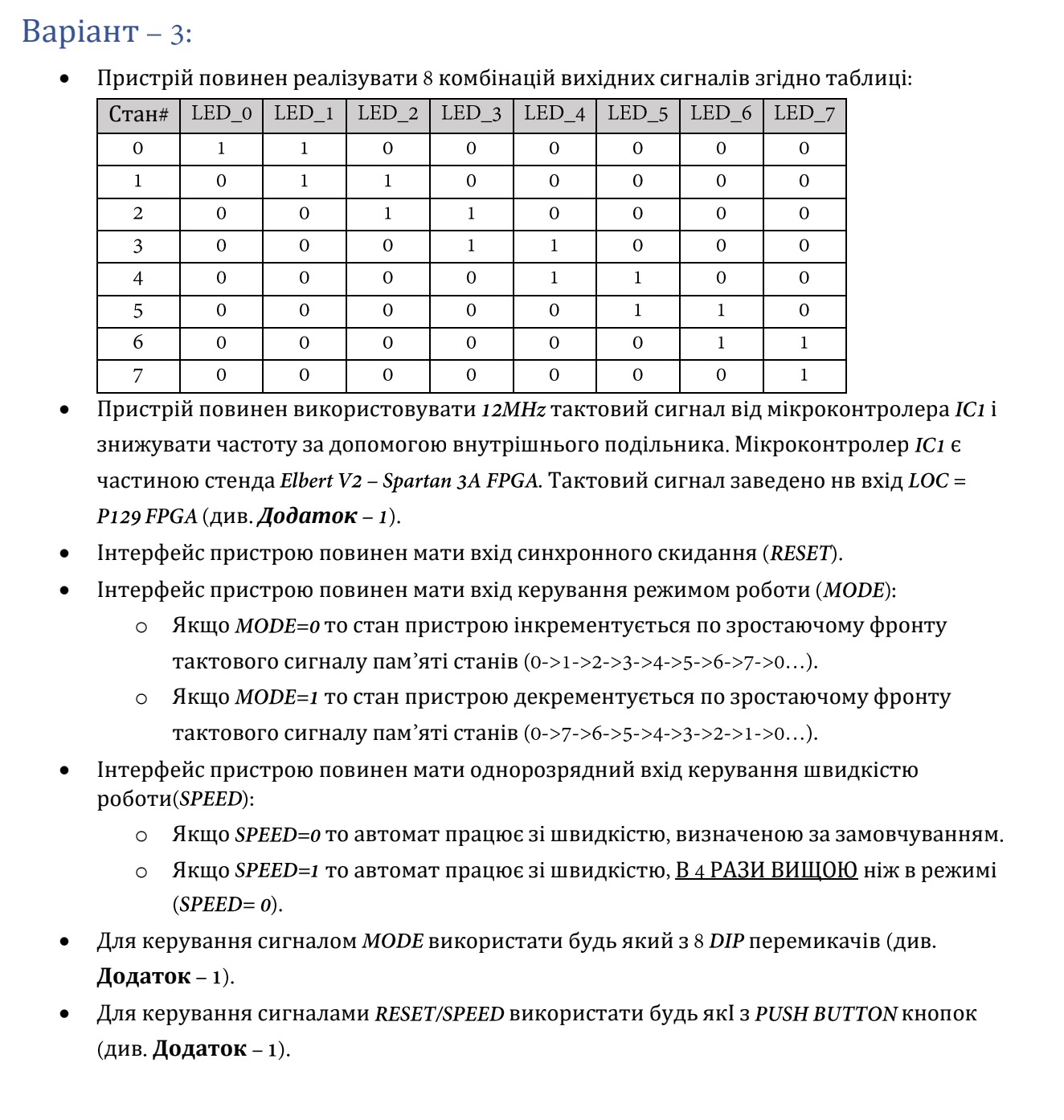
Козак Н.Б.

Львів

2023

**Мета роботи:** на базі стенда **Elbert V2 – Spartan** 3**A FPGA** реалізувати цифровий автомат світлових ефектів згідно вимог.

**Завдання згідно з варіантом:**



**Виконання роботи:**

1. Створив проект.

2. Додав VHDL Module для імплементації логіки формування вихідних сигналів.

Код OutputLogic:

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;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity out\_logic\_intf is

Port ( IN\_BUS : in std\_logic\_vector(2 downto 0);

OUT\_BUS : out std\_logic\_vector(7 downto 0)

);

end out\_logic\_intf;

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

begin

OUT\_BUS(0) <= (not(IN\_BUS(2)) and not(IN\_BUS(1)) and not(IN\_BUS(0)));

OUT\_BUS(1) <= ((not(IN\_BUS(2)) and not(IN\_BUS(1)) and not(IN\_BUS(0))) or (not(IN\_BUS(2)) and not(IN\_BUS(1)) and (IN\_BUS(0))));

OUT\_BUS(2) <= ((not(IN\_BUS(2)) and not(IN\_BUS(1)) and (IN\_BUS(0))) or (not(IN\_BUS(2)) and (IN\_BUS(1)) and not(IN\_BUS(0))));

OUT\_BUS(3) <= ((not(IN\_BUS(2)) and (IN\_BUS(1)) and not(IN\_BUS(0))) or (not(IN\_BUS(2)) and (IN\_BUS(1)) and (IN\_BUS(0))));

OUT\_BUS(4) <= ((not(IN\_BUS(2)) and (IN\_BUS(1)) and (IN\_BUS(0))) or ((IN\_BUS(2)) and not(IN\_BUS(1)) and not(IN\_BUS(0))));

OUT\_BUS(5) <= (((IN\_BUS(2)) and not(IN\_BUS(1)) and not(IN\_BUS(0))) or ((IN\_BUS(2)) and not(IN\_BUS(1)) and (IN\_BUS(0))));

OUT\_BUS(6) <= (((IN\_BUS(2)) and not(IN\_BUS(1)) and (IN\_BUS(0))) or ((IN\_BUS(2)) and (IN\_BUS(1)) and not(IN\_BUS(0))));

OUT\_BUS(7) <= (((IN\_BUS(2)) and (IN\_BUS(1)) and not(IN\_BUS(0))) or ((IN\_BUS(2)) and (IN\_BUS(1)) and (IN\_BUS(0))));

end out\_logic\_arch;



*Рис. 1 Діаграма проведеної симуляції для OutputLogic*

3. Додав VHDL Module для імплементації логіки формування переходів. Код TransitionLogic:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity transition\_logic\_intf is

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

MODE : in std\_logic;

RES : in std\_logic;

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

);

end transition\_logic\_intf;

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

begin

NEXT\_STATE(0) <= (not(RES) and not(MODE) and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 000 -> 001

(not(RES) and not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or -- 010 -> 011

(not(RES) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 100 -> 101

(not(RES) and not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or -- 110 -> 111

(not(RES) and MODE and not(CUR\_STATE(2)) and (CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 001 <- 010

(not(RES) and MODE and (CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 011 <- 100

(not(RES) and MODE and CUR\_STATE(2) and (CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 101 <- 110

(not(RES) and MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))); -- 111 <- 000

NEXT\_STATE(1) <= (not(RES) and not(MODE) and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or -- 001 -> 010

(not(RES) and not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and not(CUR\_STATE(0))) or -- 010 -> 011

(not(RES) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or -- 101 -> 110

(not(RES) and not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or -- 110 -> 111

(not(RES) and MODE and not(CUR\_STATE(2)) and (CUR\_STATE(1)) and (CUR\_STATE(0))) or -- 010 <- 011

(not(RES) and MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 011 <- 100

(not(RES) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0)) or -- 110 <- 111

(not(RES) and MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))); -- 111 <- 000

NEXT\_STATE(2) <= (not(RES) and not(MODE) and not(CUR\_STATE(2)) and CUR\_STATE(1) and CUR\_STATE(0)) or -- 011 -> 100

(not(RES) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or -- 100 -> 101

(not(RES) and not(MODE) and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or -- 101 -> 110

(not(RES) and not(MODE) and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or -- 110 -> 111

(not(RES) and MODE and CUR\_STATE(2) and not(CUR\_STATE(1)) and CUR\_STATE(0)) or -- 100 <- 101

(not(RES) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and not(CUR\_STATE(0))) or -- 101 <- 110

(not(RES) and MODE and CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0)) or -- 110 <- 111

(not(RES) and MODE and not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))); -- 111 <- 000

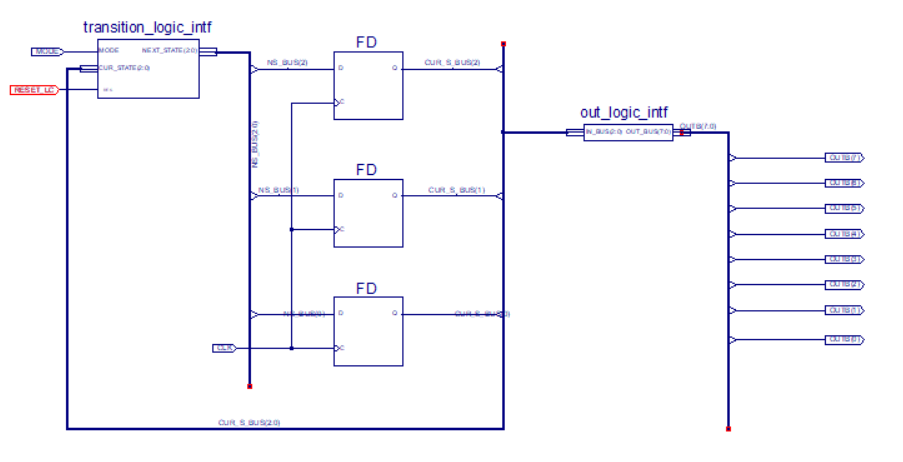
end transition\_logic\_arch;



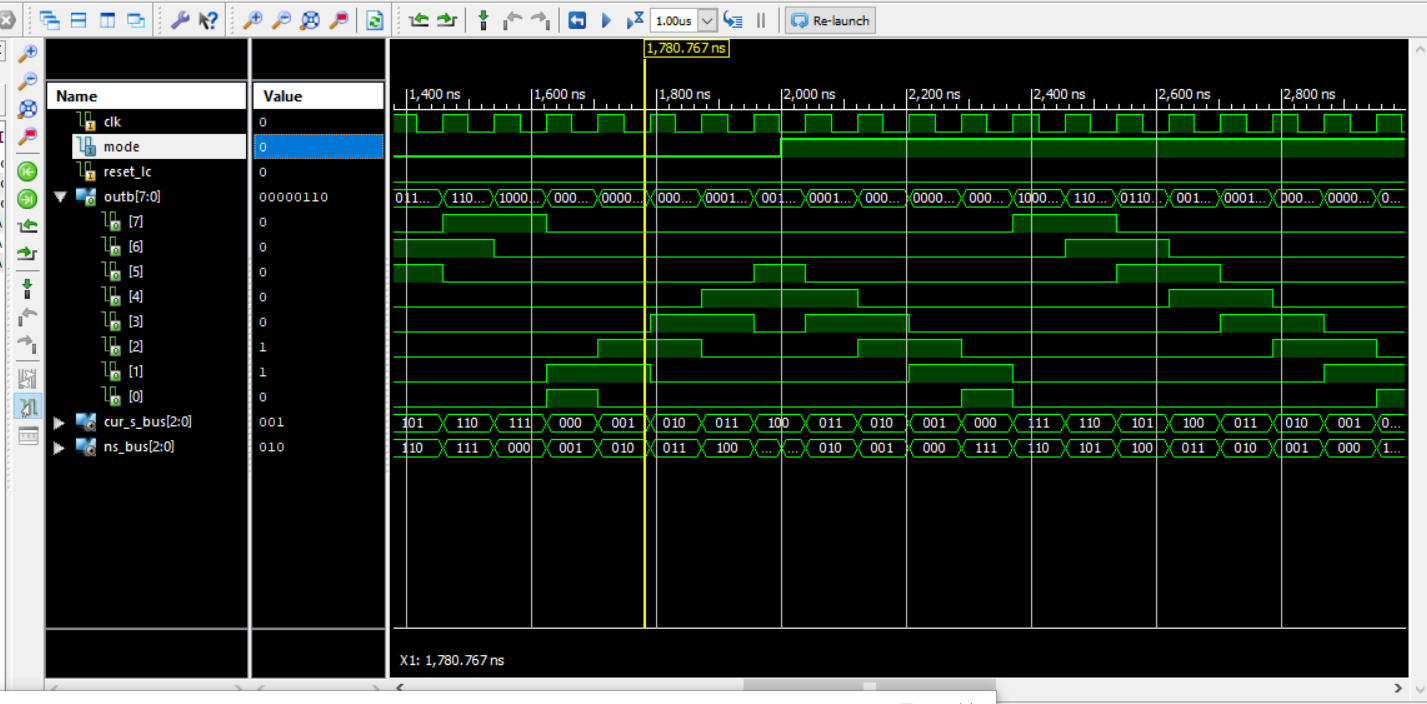
*Рис. 2 Діаграма проведеної симуляції для TransitionLogic*

4. Згенерував Schematic файли для OutputLogic та TransitionLogic.

5. Створив Schematic файл LightController, реалізував в ньому пам’ять стану автомата та зв’язав між собою всі його частини.

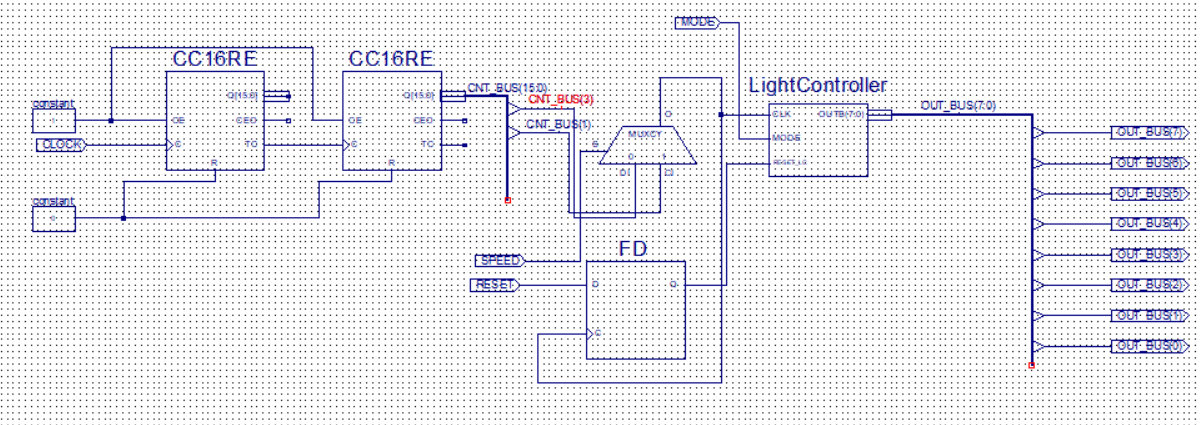


*Рис. 3 Схема LightController*

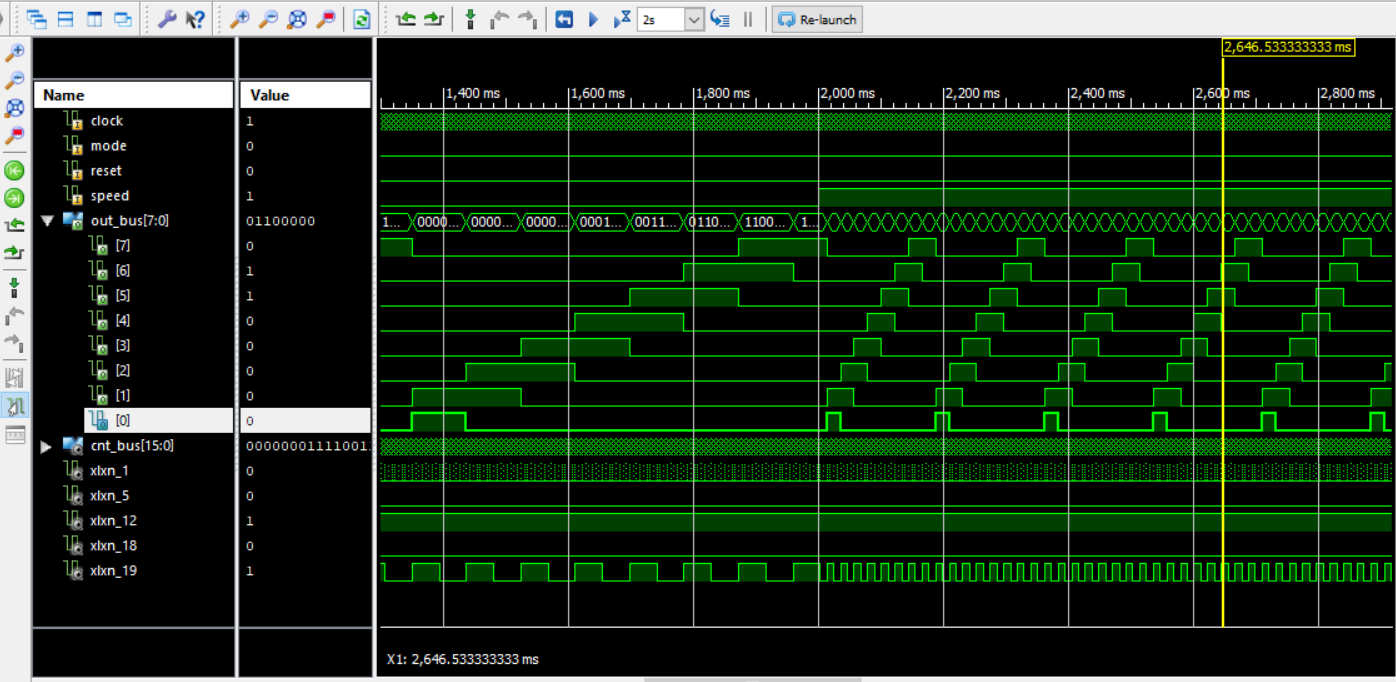


*Рис. 4 Діаграма проведеної симуляції для LightController*

6. Створив Schematic файл TopLevel, в якому реалізував подільник вхідної частоти та інтегрував його зі створеним автоматом, попередньо згенерувавши для нього Schematic файл.



*Рис. 5 Схема TopLevel*



*Рис. 6 Діаграма проведеної симуляції для TopLevel*

7. Створив Constraints файл, зв’язав в ньому виводи схеми та фізичні виводи плати.

Вміст Constraints.ucf:

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

# This file is a .ucf for ElbertV2 Development Board #

# To use it in your project : #

# \* Remove or comment the lines corresponding to unused pins in the project #

# \* Rename the used signals according to the your project #

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

# UCF for ElbertV2 Development Board #

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

CONFIG VCCAUX = "3.3" ;

# Clock 12 MHz

NET "CLOCK" LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;

####################################################################################################

# LED

####################################################################################################

NET "OUT\_BUS(0)" LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(1)" LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(2)" LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(3)" LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(4)" LOC = P50 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(5)" LOC = P51 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(6)" LOC = P52 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUT\_BUS(7)" LOC = P53 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

####################################################################################################

# DP Switches

####################################################################################################

NET "MODE" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

####################################################################################################

# Switches

####################################################################################################

NET "SREED" LOC = P80 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "RESET" LOC = P79 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

8. Згенерував бінарний файл для цифрового автомата світлових ефектів.

**Висновок:** виконавши лабораторну роботу, здобуто навики реалізації цифрових автоматів світлових ефектів.