**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



**Звіт**

до лабораторної роботи № 2

з дисципліни: «Моделювання комп’ютерних систем»

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

Варіант 13

Виконав:

ст. гр. КІ-201

Костюк І. В.

Прийняв:

Козак Н. Б.

**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 Дивись розділ Завдання
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів Заборонено використовуючи оператори
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів Заборонено використовуючи оператори
4. Згенерувати символи для описів логіки переходів та логіки формування вихідних сигналів
5. Згрупувати всі компоненти логіку переходів логіку формування вихідних сигналів та пам’ять станів в єдину систему. 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 додати подільник частоти для вхідного тактового сигналу призначити фізичні виводи.
8. Згенерувати файл та перевірити роботу за допомогою стенда Elbert V2 – Spartan 3A FPGA.
9. Підготувати і захистити звіт

**Завдання:**

Изображение выглядит как стол

Автоматически созданное описание

**Хід роботи:**

Створив новий проект користуючись методичними вказівками до лабораторної роботи №1.

Додав до проекту новий VHDL файл Output\_Logic, в якому буде реалізовано логіку формування вихідних сигналів.

**Лістинг:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.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))); -- 0 000

OUT\_BUS(1) <= (not (IN\_BUS(2)) and not (IN\_BUS(1)) and (IN\_BUS(0))); -- 1 001

OUT\_BUS(2) <= (not(IN\_BUS(2)) and (IN\_BUS(1)) and not(IN\_BUS(0))); -- 2 010

OUT\_BUS(3) <= (not(IN\_BUS(2)) and (IN\_BUS(1)) and (IN\_BUS(0))); -- 3 011

OUT\_BUS(4) <= ((IN\_BUS(2)) and not(IN\_BUS(1)) and not(IN\_BUS(0))); -- 4 100

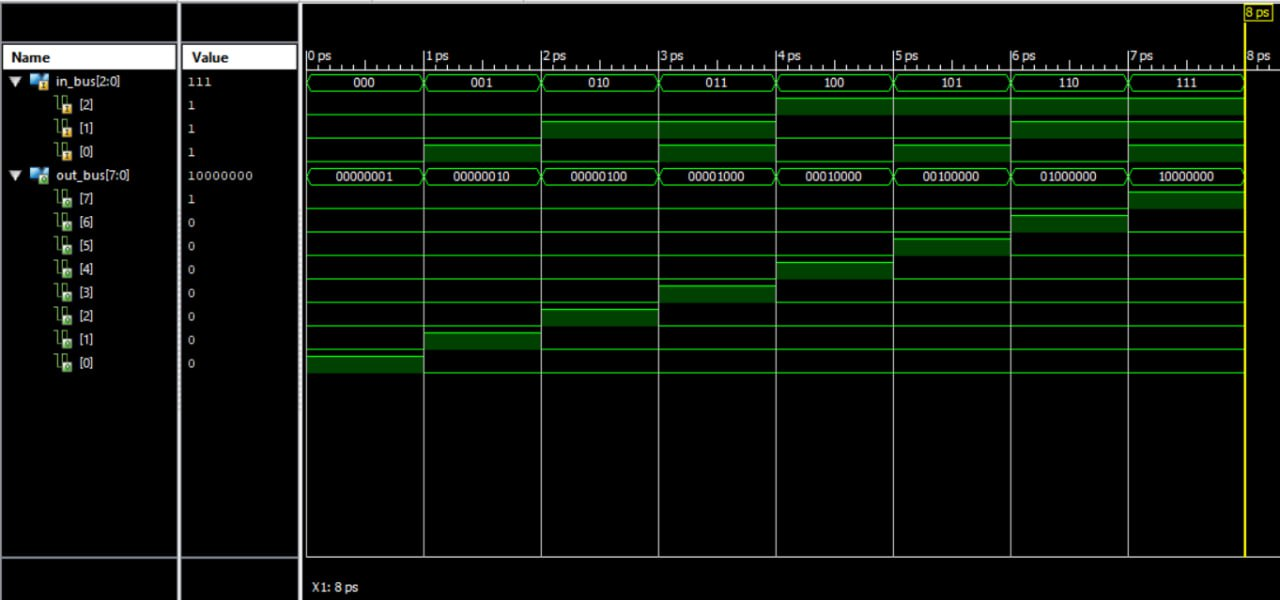
OUT\_BUS(5) <= ((IN\_BUS(2)) and not(IN\_BUS(1)) and (IN\_BUS(0))); -- 5 101

OUT\_BUS(6) <= ((IN\_BUS(2)) and (IN\_BUS(1)) and not(IN\_BUS(0))) ; -- 6 110

OUT\_BUS(7) <= ((IN\_BUS(2)) and (IN\_BUS(1)) and (IN\_BUS(0))); -- 7 111

end out\_logic\_arch;

За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах:



*Рис. 1. Симуляція Output\_Logic*

Додав до проекту VHDL файл Transition\_Logic, в якому реалізував логіку формування переходів:

**Лістинг:**

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;

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

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

(not(MODE) and (CUR\_STATE(2))and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or --4

(not(MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and not (CUR\_STATE(0))) or --6

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

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

((MODE) and (CUR\_STATE(2))and not (CUR\_STATE(1)) and not (CUR\_STATE(0))) or --4

((MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and not (CUR\_STATE(0))) ; --6

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

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

(not(MODE) and (CUR\_STATE(2))and not (CUR\_STATE(1)) and (CUR\_STATE(0))) or --5

(not(MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and not(CUR\_STATE(0))) or --6

((MODE) and not(CUR\_STATE(2))and not(CUR\_STATE(1)) and not (CUR\_STATE(0))) or --4

((MODE) and not(CUR\_STATE(2))and (CUR\_STATE(1)) and (CUR\_STATE(0))) or --7

((MODE) and (CUR\_STATE(2))and not (CUR\_STATE(1)) and not(CUR\_STATE(0))) or --3

((MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and (CUR\_STATE(0))) ; --0

NEXT\_STATE(2) <= (not(MODE) and not(CUR\_STATE(2))and (CUR\_STATE(1)) and (CUR\_STATE(0))) or --3

(not(MODE) and (CUR\_STATE(2))and not (CUR\_STATE(1)) and not (CUR\_STATE(0))) or --4

(not(MODE) and (CUR\_STATE(2))and not (CUR\_STATE(1)) and (CUR\_STATE(0))) or --5

(not(MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and not (CUR\_STATE(0)))or --6

((MODE) and not(CUR\_STATE(2))and not(CUR\_STATE(1)) and not (CUR\_STATE(0))) or --7

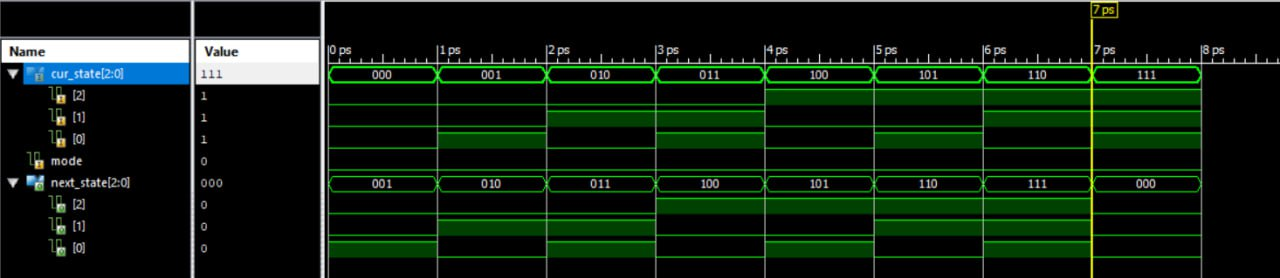
((MODE) and (CUR\_STATE(2))and not (CUR\_STATE(1)) and (CUR\_STATE(0))) or --0

((MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and not (CUR\_STATE(0))) or --5

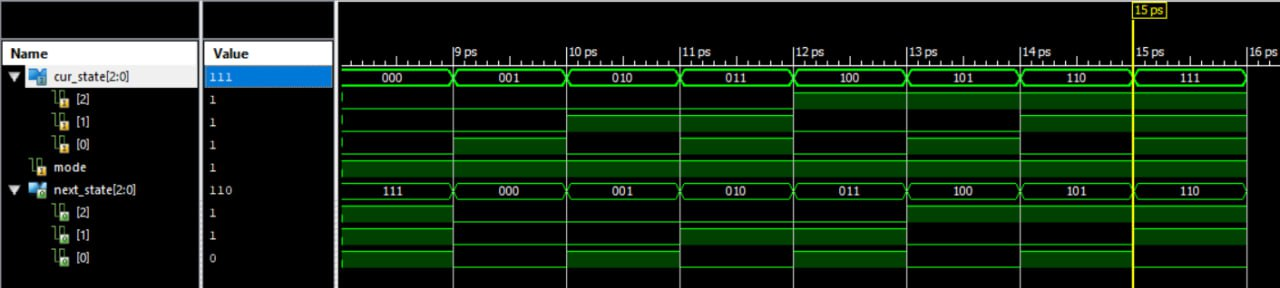
((MODE) and (CUR\_STATE(2))and (CUR\_STATE(1)) and (CUR\_STATE(0))) ; --6

end transition\_logic\_arch;

За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.

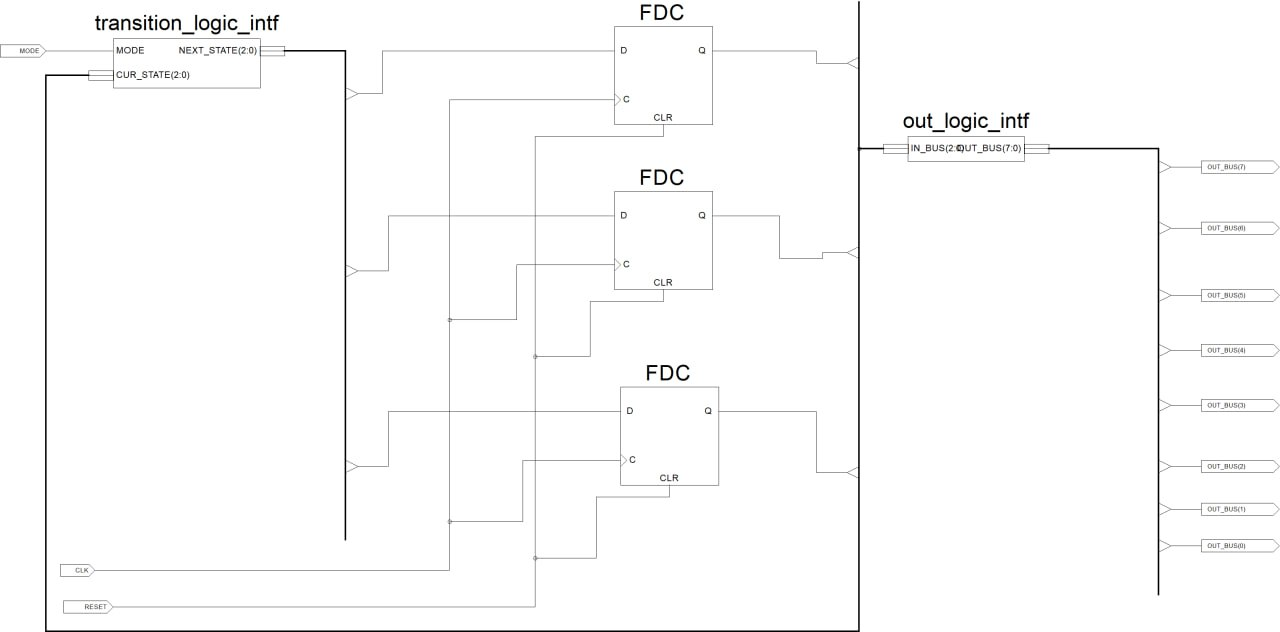


*Рис. 2. Симуляція Transition\_Logic 0 -> 7*

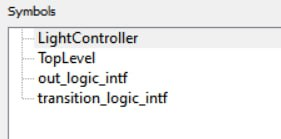


*Рис. 3. Симуляція Transition\_Logic 7 -> 0*

Додав до проекту Schematic файл Light\_Controller, виконав для нього команду Set as Top Module. Згенерував Schematic символи для файлів Output\_Logic і Transition\_Logic. Використовуючи новостворені символи та елементи з бібліотеки, реалізував у файлі Light\_Controller.sch пам’ять стану автомата.



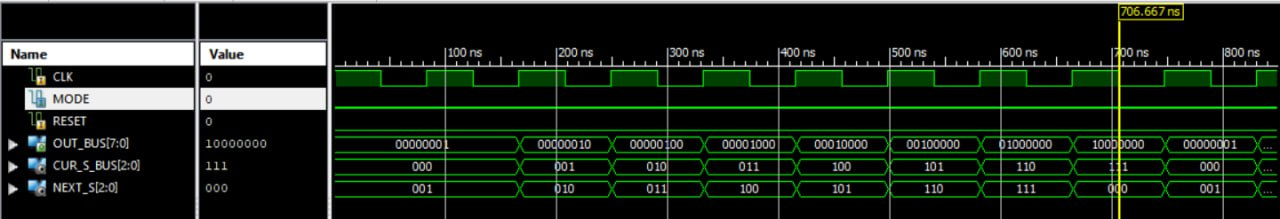
*Рис. 4. Schematic Light\_Controller*



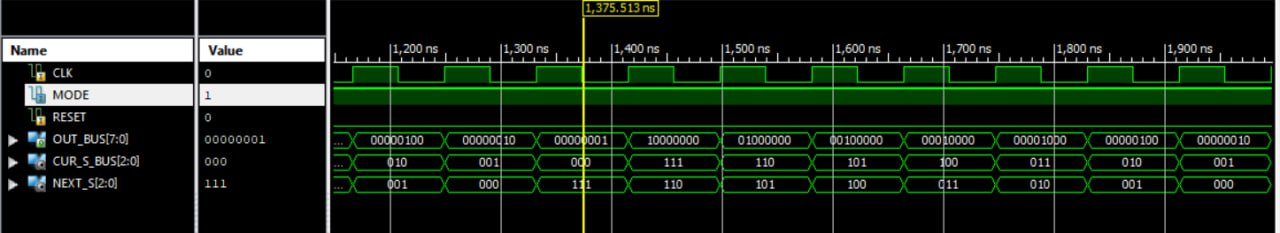
*Рис. 5. Згенеровані Schematic символи для файлів Output\_Logic і Transition\_Logic*

За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.

При MODE 0:



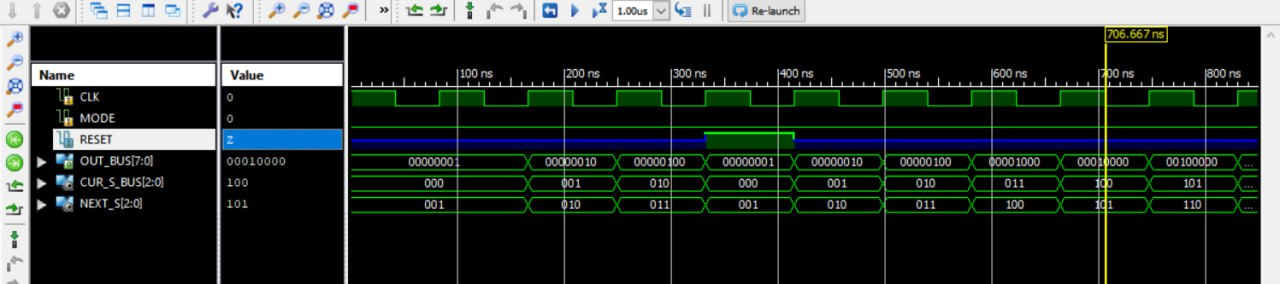
*Рис. 6. MODE 0*



*Рис. 7. MODE 1*

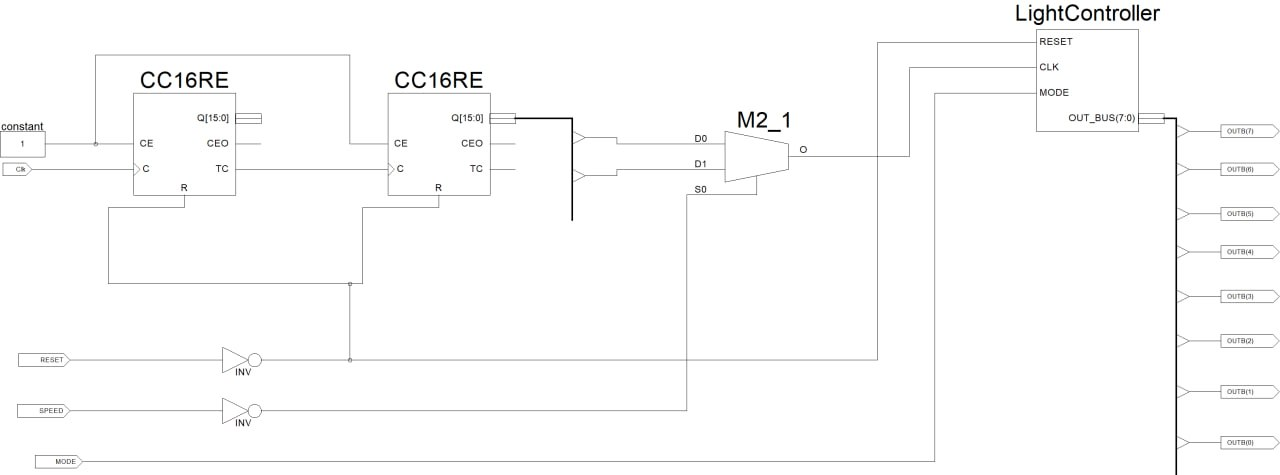


*Рис. 8. RESET*



*Рис. 9. RESET*

Додав до проекту *Schematic* файл *Top\_Level*, виконав для нього команду Set as Top Module. Згенерував *Schematic* символ для файлу *Light\_Controller*. Використовуючи новостворений символ та елементи з бібліотеки, реалізував у файлі подільник вхідної частоти та логіку сигналу *SPEED*.

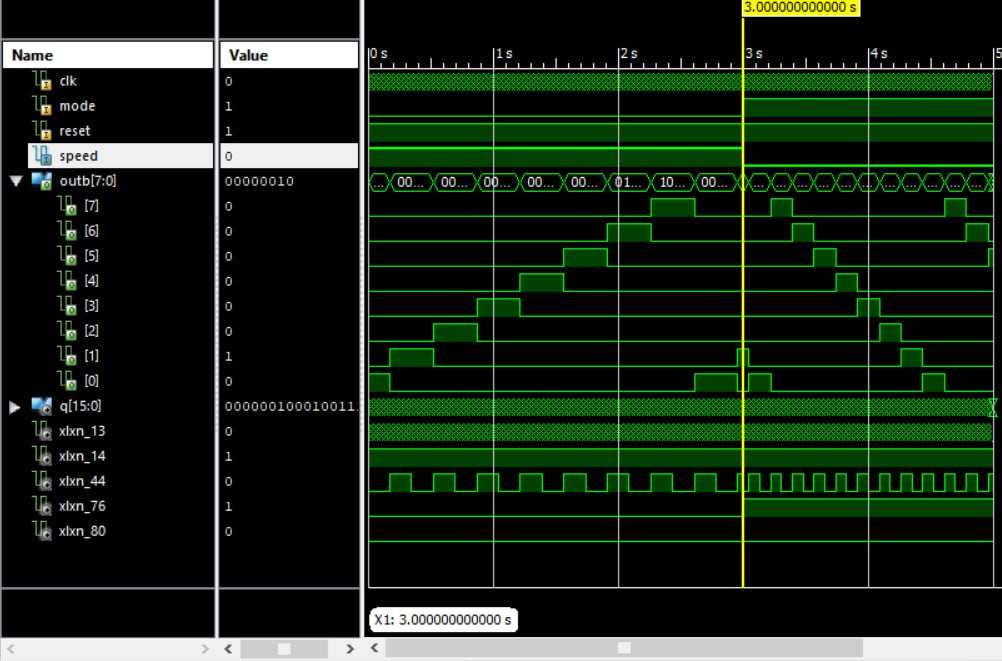


*Рис. 10. Top\_Level*

За допомогою Isim провів моделювання роботи схеми з різними значеннями сигналів MODE, RESET та SPEED при подачі на вхід CLOCK тактового сигналу 12 MHz.

SPEED =1, MODE = 0 (до лінії)

SPEED =0 MODE = 1 (після лінії).



*Рис. 11. Симуляція Top\_Level*

Додав до проекту Constraint файл та призначила виводам схеми фізичні виводи цільової FPGA.

**Лістинг:**

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

# 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 "Clk" LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;

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

# LED

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

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

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

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

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

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

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

NET "OUTB(6)" LOC = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "OUTB(7)" LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

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

# DP Switches

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

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

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

# Switches

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

NET "RESET" LOC = P76 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

NET "SPEED" LOC = P75 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

Згенерував файли прошиття.



*Рис. 12. Файли прошиття*

**Висновок:** під час виконання лабораторної роботи я реалізував цифровий автомат світлових ефектів у середовищі Xilinx ISE і стендом Elbert V2 - Spartan 3A FPGA, реалізував схему автомату, провів симуляцію його роботи.