



Звіт

З лабораторної роботи № 1

З дисципліни «Моделювання комп'ютерних систем»

На тему: «Структурний опис цифрового автомата
Перевірка роботи
автомата за допомогою стенда»

Виконав: ст. гр. КІ-202

Довганюк О. С.

Прийняв: ст. в.

Козак Н. Б.

Мета роботи

На базі стенда реалізувати цифровий автомат світлових ефектів.

Завдання

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

Варіант 1(7):

- Пристрій повинен реалізувати комбінацій вихідних сигналів згідно таблиці:

Стан#	LED_0	LED_1	LED_2	LED_3	LED_4	LED_5	LED_6	LED_7
0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1

- Пристрій повинен використовувати тактовий сигнал від мікроконтролера і знижувати частоту за допомогою внутрішнього подільника. Мікроконтролер є частиною стенда.
- Інтерфейс пристрою повинен мати вхід синхронного скидання
- Інтерфейс пристрою повинен мати вхід керування режимом роботи:

Якщо то стан пристрою інкрементується по зростаючому фронту тактового сигналу пам'яті станів;

Якщо то стан пристрою декрементується по зростаючому фронту тактового сигналу пам'яті станів;

- Інтерфейс пристрою повинен мати однорозрядний вхід керування швидкістю роботи

Якщо SPEED=0 то автомат працює зі швидкістю визначеною за замовчуванням

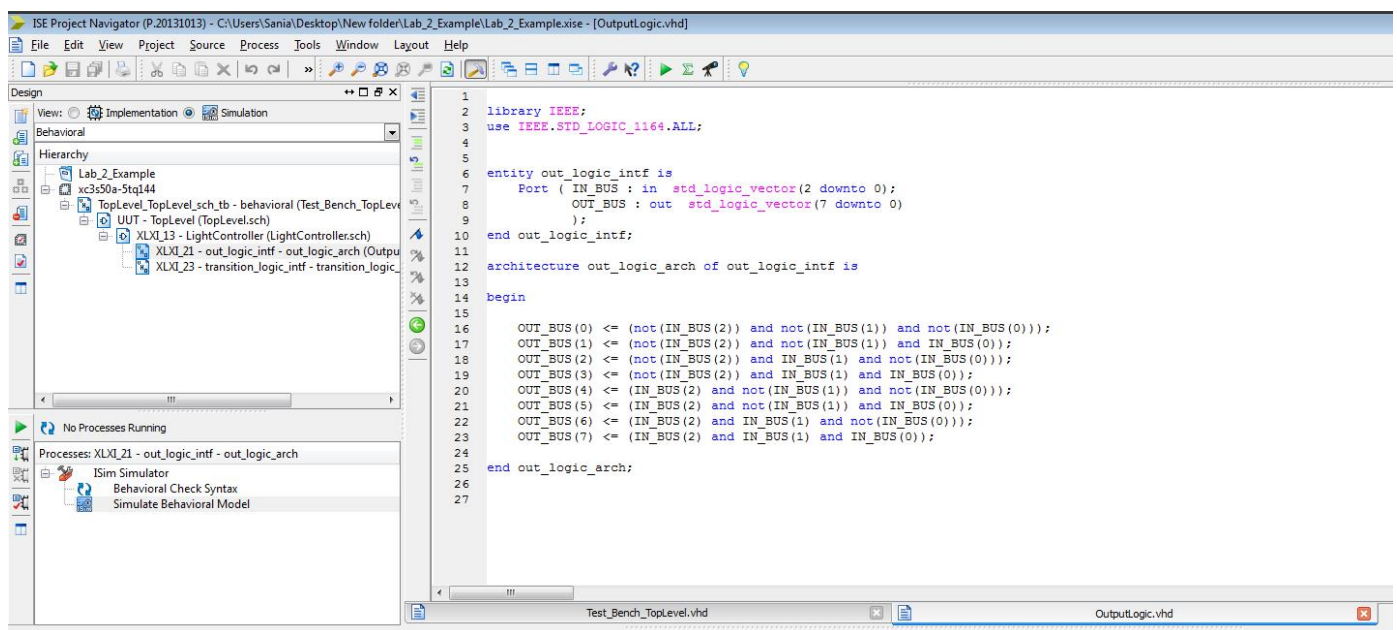
Якщо SPEED=1 то автомат працює зі швидкістю В РАЗИ ВИЩОЮ ніж в режимі SPEED=0

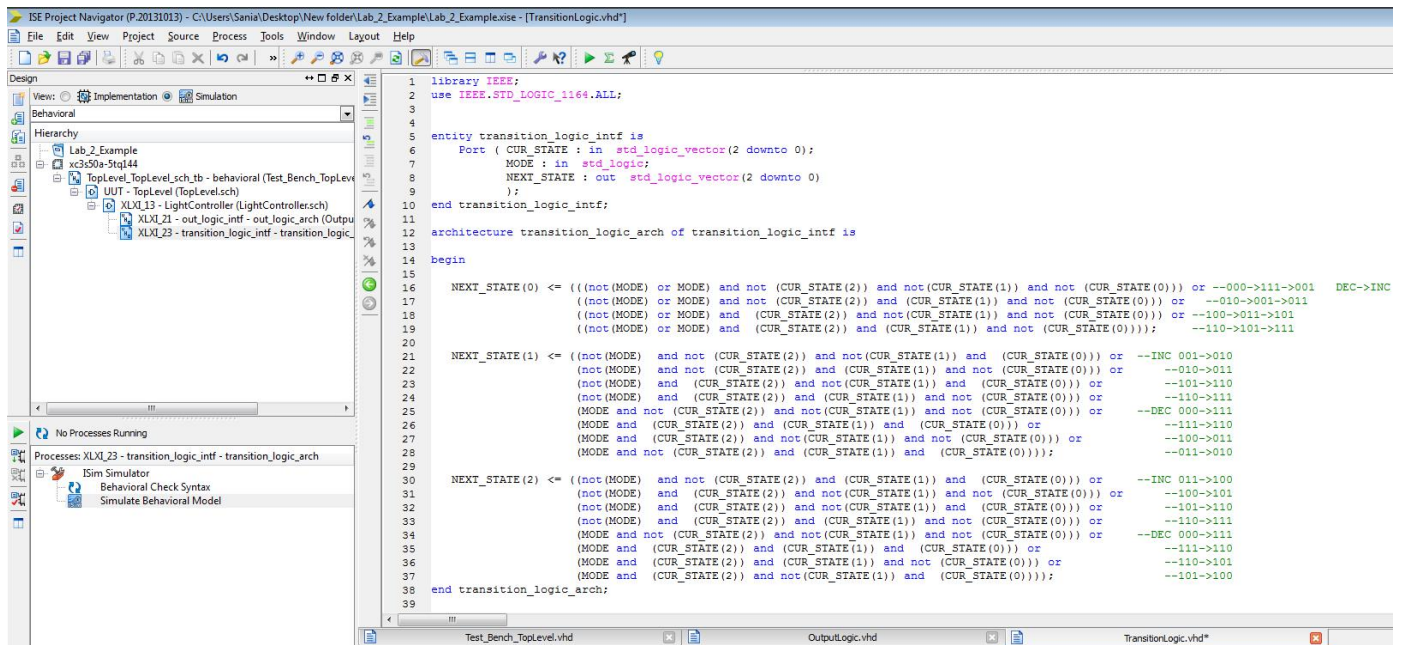
- Для керування сигналом MODE використати будь який з 8 DIP перемикачів
- Додаток

Виконання роботи

1) Спочатку створюю новий проект користуючись методичними вказівками до лабораторної роботи No1.

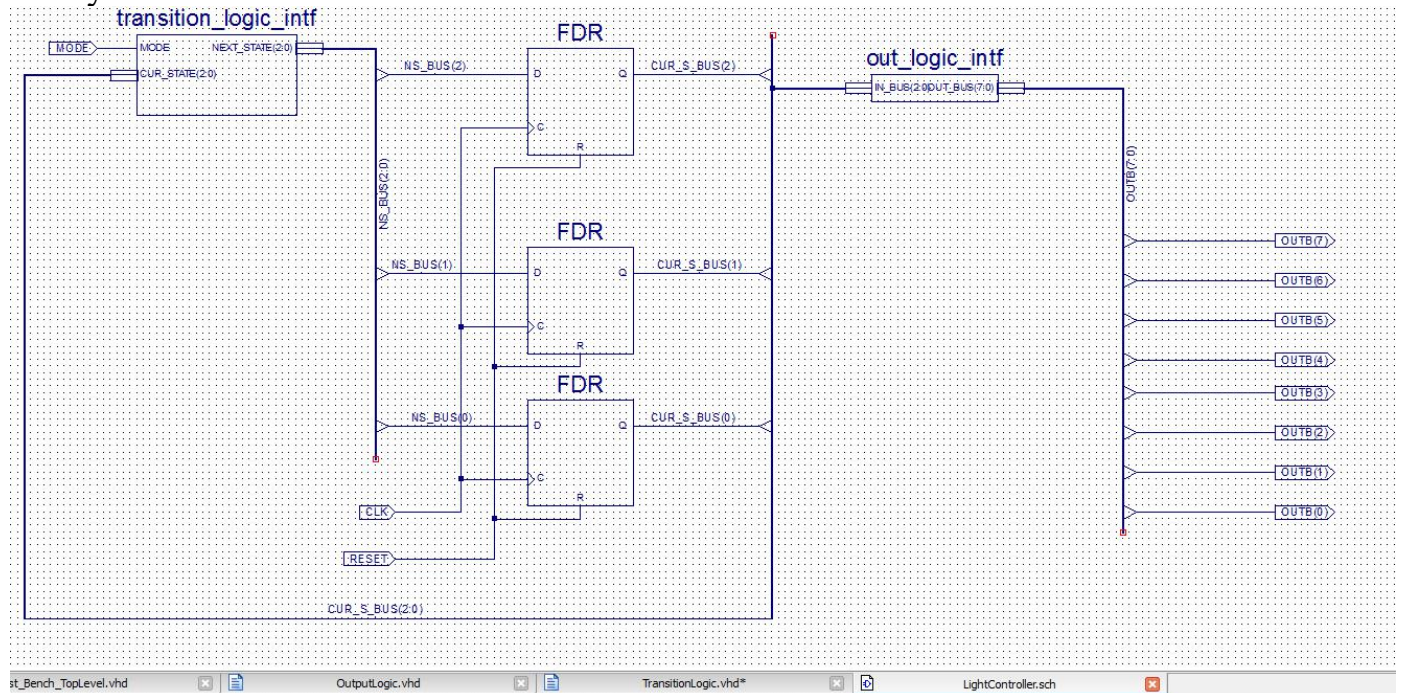
2) Додаю до проекту два нових файли в яких реалізую логіку формування вихідних сигналів та логіку формування переходів:





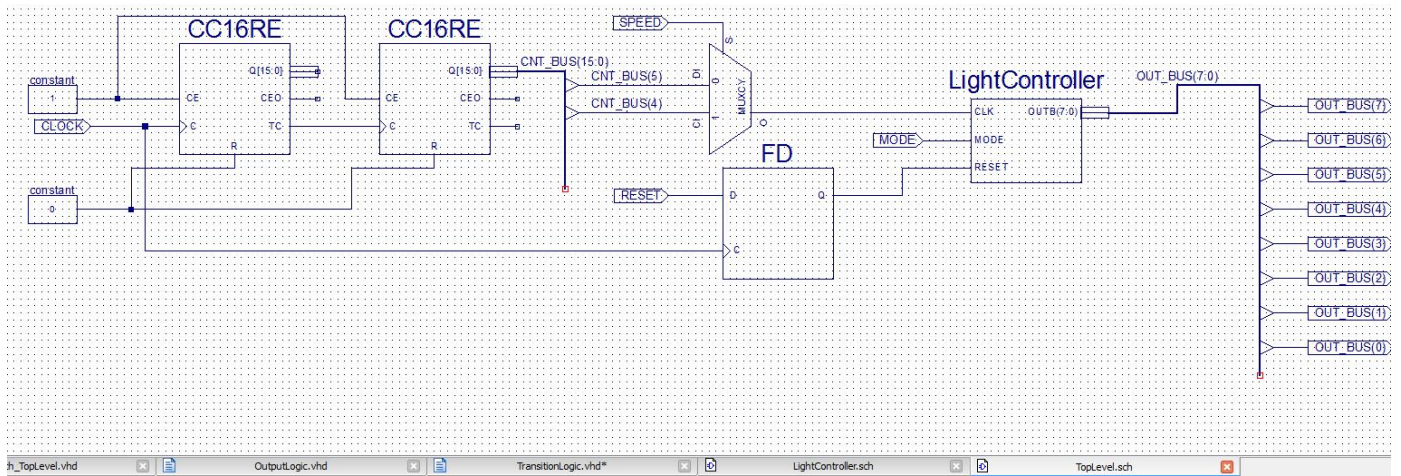
3) Генерую Schematic символи для створених файлів.

4) Додаю до проекту новий файл в якому реалізую пам'ять стану автомата та зв'язую між ябою всі його частини.



5) Генерую Schematic символ для файлу LightController.sch;

6) Створюю файл TopLevel.sch, реалізую подільник входної частоти та інтегрую його зі створеним автоматом (використовуючи створений символ зі схеми автомата):



7) Створюю файл Constraints.ucf та призначаю виводам схеми фізичні виводи цільової FPGA:

```

10
11 CONFIG VCCAUX = "3.3" ;
12
13 # Clock 12 MHz
14 NET "CLOCK" LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
15
16 #####
17 # LED
18 #####
19
20 NET "OUT_BUS(0)" LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
21 NET "OUT_BUS(1)" LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
22 NET "OUT_BUS(2)" LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
23 NET "OUT_BUS(3)" LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
24 NET "OUT_BUS(4)" LOC = P50 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
25 NET "OUT_BUS(5)" LOC = P51 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
26 NET "OUT_BUS(6)" LOC = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
27 NET "OUT_BUS(7)" LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
28
29 #####
30 # DP Switches
31 #####
32
33 NET "MODE" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
34 NET "RESET" LOC = P80 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
35 NET "SPEED" LOC = P79 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
36

```

Constraints.ucf

8) Створюю файл TestBenchTopLevel.vhd та прописую в ньому поведінку вхідних сигналів для тестування схеми за допомогою симулятора:

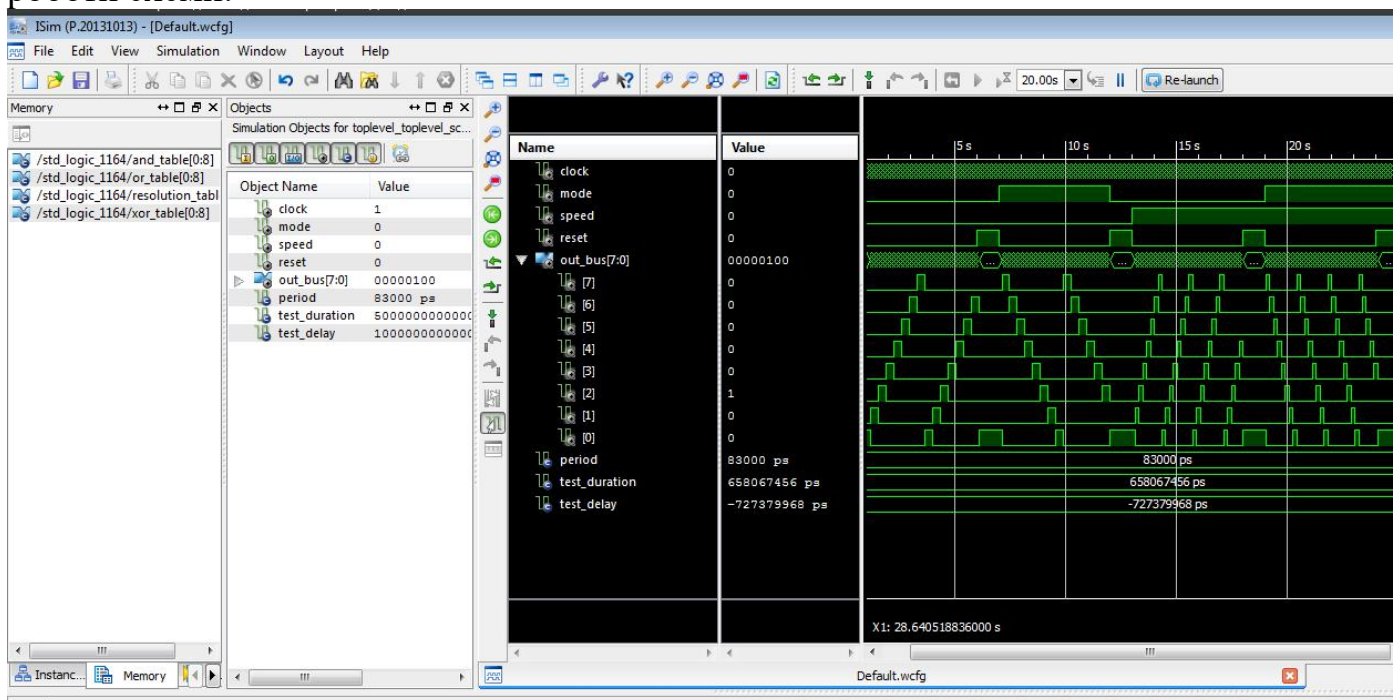

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4  LIBRARY UNISIM;
5  USE UNISIM.vcomponents.ALL;
6  ENTITY TopLevel_TopLevel_sch_tb IS
7  END TopLevel_TopLevel_sch_tb;
8  ARCHITECTURE behavioral OF TopLevel_TopLevel_sch_tb IS
9
10     COMPONENT TopLevel
11     PORT( CLOCK      : IN STD_LOGIC;
12           MODE       : IN STD_LOGIC;
13           SPEED      : IN STD_LOGIC;
14           RESET      : IN STD_LOGIC;
15           OUT_BUS    : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0));
16     END COMPONENT;
17
18     SIGNAL CLOCK      : STD_LOGIC;
19     SIGNAL MODE       : STD_LOGIC;
20     SIGNAL SPEED      : STD_LOGIC;
21     SIGNAL RESET      : STD_LOGIC;
22     SIGNAL OUT_BUS    : STD_LOGIC_VECTOR (7 DOWNTO 0);
23
24     CONSTANT PERIOD    : time := 83 ns;
25     CONSTANT TEST_DURATION : time := 5000 ms;
26     CONSTANT TEST_DELAY : time := 1000 ms;
27
28 BEGIN
29
30     UUT: TopLevel PORT MAP (
31         CLOCK => CLOCK,
32         MODE  => MODE,
33         SPEED => SPEED,
34         RESET => RESET,
35         OUT_BUS => OUT_BUS
36     );
37
38     clock_proc : PROCESS
39     BEGIN
39
40         CLOCK <= '0';
41         wait for PERIOD / 2;
42
43         CLOCK <= '1';
44         wait for PERIOD / 2;
45     END PROCESS;
46
47     main : PROCESS
48     BEGIN
49         RESET <= '1';
50         wait for TEST_DELAY;
51
52         MODE <= '0';
53         SPEED <= '0';
54         RESET <= '0';
55         wait for TEST_DURATION;
56
57         RESET <= '1';
58         wait for TEST_DELAY;
59
60         RESET <= '0';
61         MODE <= '1';
62         wait for TEST_DURATION;
63
64         RESET <= '1';
65         MODE <= '0';
66         wait for TEST_DELAY;
67
68         RESET <= '0';
69         SPEED <= '1';
70         wait for TEST_DURATION;
71
72         RESET <= '1';
73         wait for TEST_DELAY;
74
75         RESET <= '0';
76         MODE <= '1';
77         wait for TEST_DURATION;
78     END PROCESS;
79 END;
80

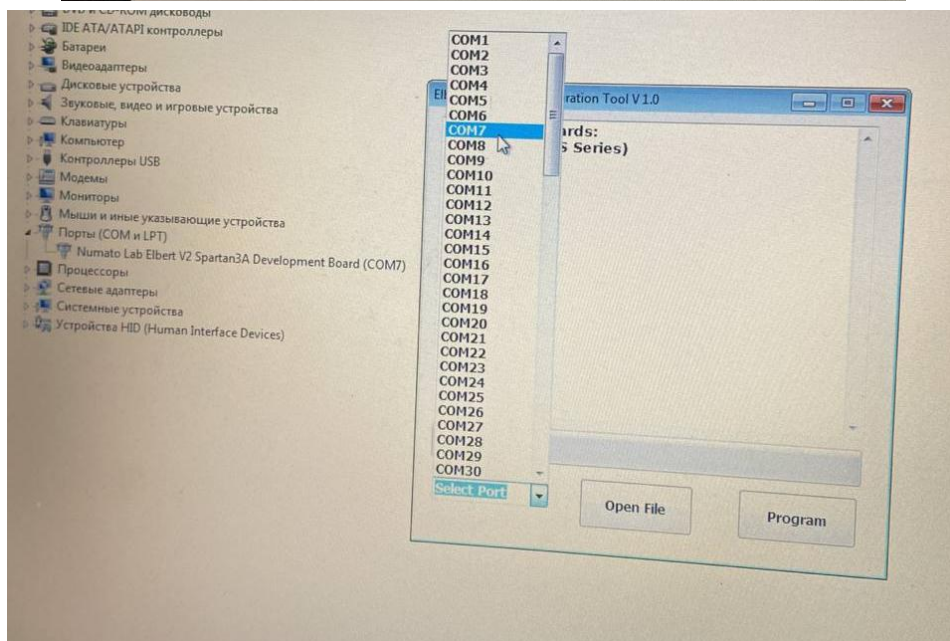
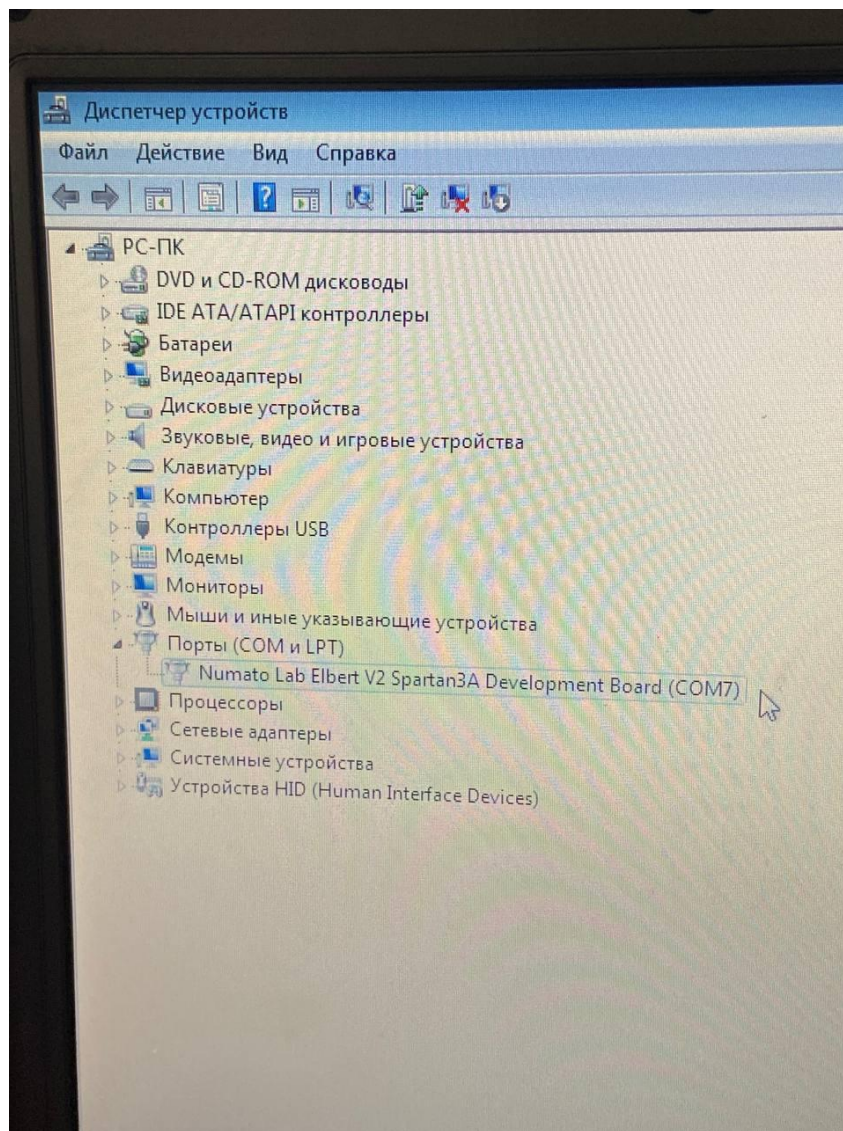
```

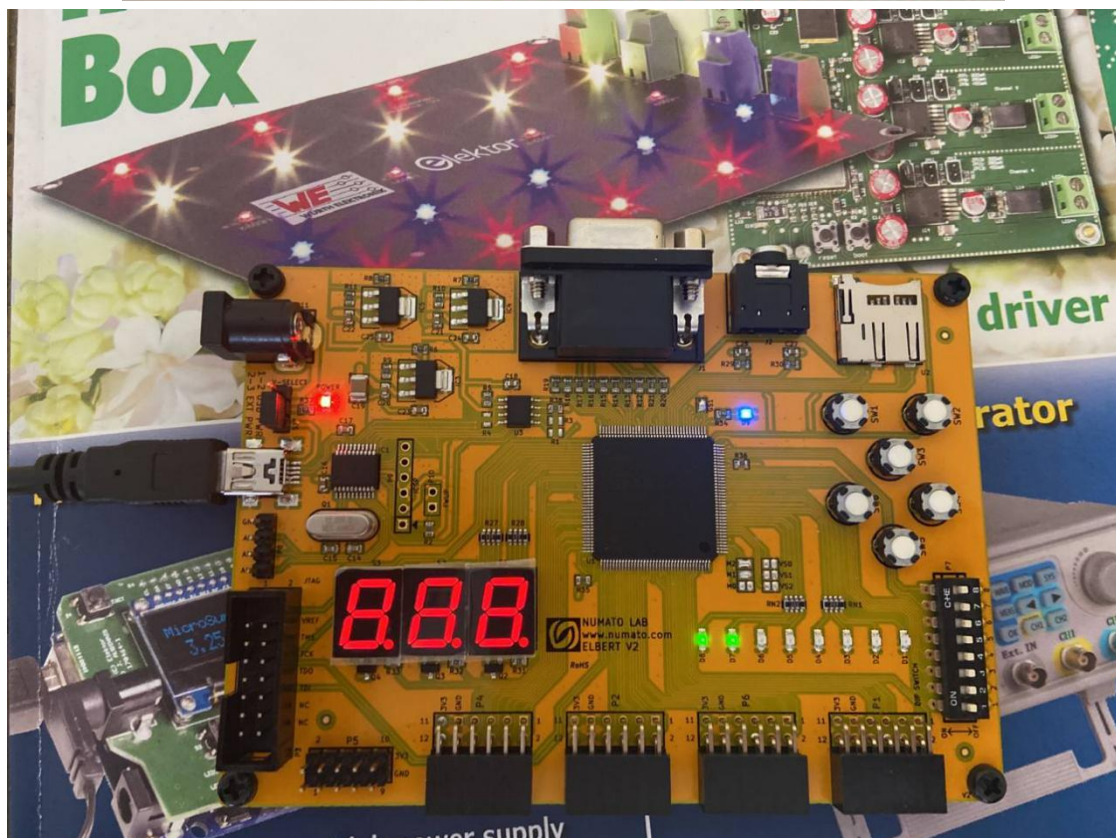
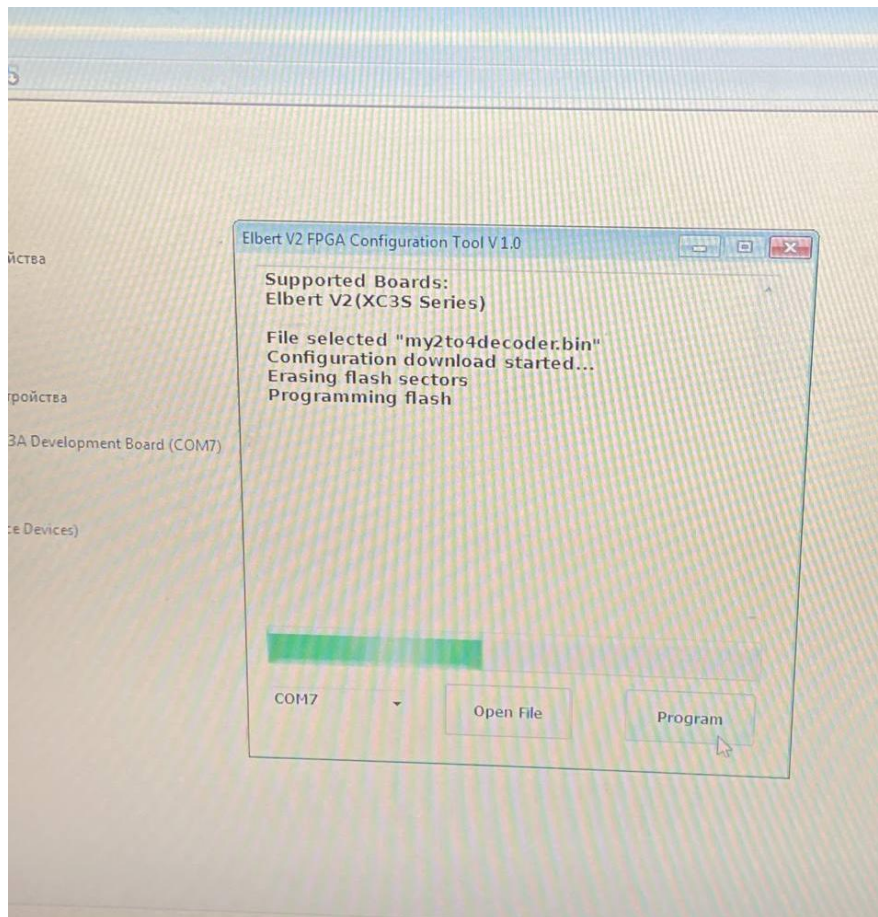
Test_Bench_TopLevel.vhd*

9) Запускаю симулятор для файла TestBenchTopLevel.vhd та перевіряю правильність роботи схеми:



10) Генерую бінарний файл та запрограмовую ним стэнд:





Висновок: у цій лабораторній роботі я реалізував цифровий автомат світлових ефектів та навчився писати Test Bench файл.