

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”

Виконав: ст. гр. КІ-201

Давида В.Р.

Прийняв:
Козак Н.Б.

Львів 2023

Мета: “На базі стенда реалізувати цифровий автомат світлових ефектів”.

Завдання до варіанту № 6:

Варіант – 6:

- Пристрій повинен реалізувати 8 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	1	1	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0
3	1	1	1	1	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	1	1	0	0
6	0	0	0	0	1	1	1	0
7	0	0	0	0	1	1	1	1

- Пристрій повинен використовувати 12MHz тактовий сигнал від мікроконтролера IC1 і знижувати частоту за допомогою внутрішнього подільника. Мікроконтролер IC1 є частиною стенда *Elbert V2 – Spartan 3A FPGA*. Тактовий сигнал заведено на вхід *LOC = P129 FPGA* (див. **Додаток – 1**).
- Інтерфейс пристрою повинен мати вхід синхронного скидання (*RESET*).
- Інтерфейс пристрою повинен мати вхід керування режимом роботи (*MODE*):
 - Якщо *MODE=0* то стан пристрою інкрементується по зростаючому фронту тактового сигналу пам'яті станів (0->1->2->3->4->5->6->7->0...).
 - Якщо *MODE=1* то стан пристрою декрементується по зростаючому фронту тактового сигналу пам'яті станів (0->7->6->5->4->3->2->1->0...).
- Інтерфейс пристрою повинен мати однорозрядний вхід (*TEST*) для подачі логічної «1» на всі непарні виходи одночасно:
 - Якщо *TEST=0* то автомат перемикає сигнали на виходах згідно заданого алгоритму.
 - Якщо *TEST=1* то на непарних виходах (7, 5, 3, 1) повинна бути логічна «1» (непарні *LED* увімкнені).
- Для керування сигналом *MODE* використати будь який з 8 *DIP* перемикачів (див. **Додаток – 1**).
- Для керування сигналами *RESET/TEST* використати будь які з *PUSH BUTTON* кнопок (див. **Додаток – 1**).

Хід виконання:

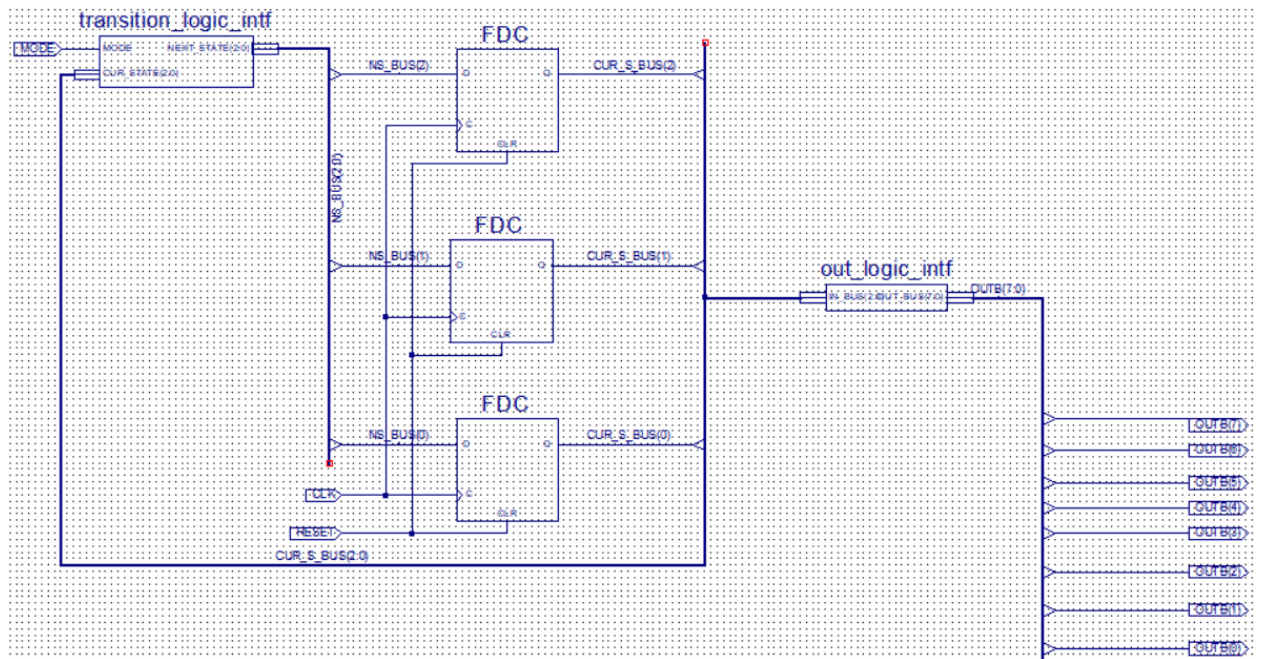
1) Створюю TransitionLogic.vhd

```
32 entity transition_logic_intf is
33   Port( CUR_STATE : in std_logic_vector (2 downto 0);
34         MODE : in std_logic;
35         NEXT_STATE : out std_logic_vector (2 downto 0)
36       );
37 end transition_logic_intf;
38
39 architecture transition_logic_arch of transition_logic_intf is
40
41 begin
42
43   NEXT_STATE(0) <= (not(MODE) and not(CUR_STATE(1)) and not (CUR_STATE(0))) or
44                   (not(MODE) and CUR_STATE(1) and not(CUR_STATE(0))) or
45                   (MODE and not(CUR_STATE(1)) and not (CUR_STATE(0))) or
46                   (MODE and CUR_STATE(1) and not (CUR_STATE(0)));
47
48   NEXT_STATE(1) <= (not(MODE) and not(CUR_STATE(1)) and CUR_STATE(0)) or
49                   (not(MODE) and CUR_STATE(1) and not (CUR_STATE(0))) or
50                   (MODE and not(CUR_STATE(1)) and not(CUR_STATE(0))) or
51                   (MODE and CUR_STATE(1) and CUR_STATE(0));
52
53   NEXT_STATE(2) <= ((not(MODE) and CUR_STATE(2)) and not (CUR_STATE(1) and CUR_STATE(0))) or
54                   ((not(MODE) and not (CUR_STATE(2))) and (CUR_STATE(1) and CUR_STATE(0))) or
55                   ((MODE and CUR_STATE(2)) and (CUR_STATE(1) or CUR_STATE(0))) or
56                   (MODE and not CUR_STATE(2) and not CUR_STATE(1) and not CUR_STATE(0));
57
58 end transition_logic_arch;
59
60
```

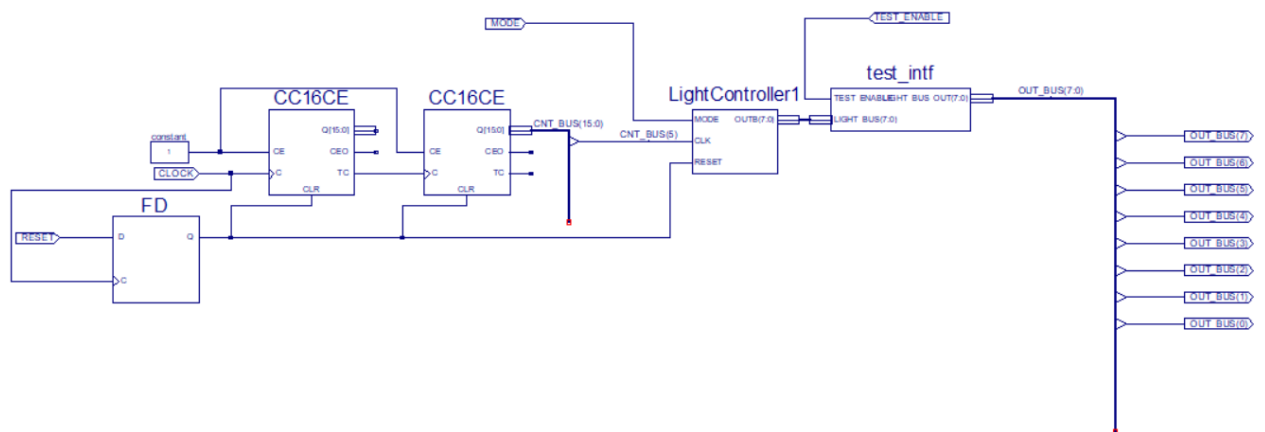
2) Створюю OutputLogic.vhd

```
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity out_logic_intf is
33   Port( IN_BUS : in std_logic_vector(2 downto 0);
34         OUT_BUS : out std_logic_vector(7 downto 0)
35       );
36 end out_logic_intf;
37
38 architecture out_logic_arch of out_logic_intf is
39
40 begin
41
42   OUT_BUS(0) <= not IN_BUS(2);
43
44   OUT_BUS(1) <= not IN_BUS(2) and (IN_BUS(1) or IN_BUS(0));
45
46   OUT_BUS(2) <= not IN_BUS(2) and IN_BUS(1);
47
48   OUT_BUS(3) <= not IN_BUS(2) and IN_BUS(1) and IN_BUS(0);
49
50   OUT_BUS(4) <= IN_BUS(2);
51
52   OUT_BUS(5) <= IN_BUS(2) and (IN_BUS(1) or IN_BUS(0));
53
54   OUT_BUS(6) <= IN_BUS(2) and IN_BUS(1);
55
56   OUT_BUS(7) <= IN_BUS(2) and IN_BUS(1) and IN_BUS(0);
57
58 end out_logic_arch;
59
60
```

3) Створюю схему LightController.sch



4) Створюю файл TopLevel.sch



Також, щоб забезпечити можливість зменшення вхідної частоти в чотири рази – умова завдання, додаю мультіплексор.

5) Добавлю Constraints.ucf файл

```

8  #*****
9  #
10 #*****
11 #*****
12 #*****
13 #*****
14 #*****
15 #*****
16 #*****
17 #*****
18 #*****
19 #*****
20 #*****
21 #*****
22 #*****
23 #*****
24 #*****
25 #*****
26 #*****
27 #*****
28 #*****
29 #*****
30 #*****
31 #*****
32 #*****
33 #*****
34 #*****
35 #*****
36 #*****
37 #*****
38 #*****
39 #*****
40 #*****
41 #*****

```

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 = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "OUT_BUS[7]"      LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

#####
#
# DP Switches
#####

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

#####
#
# Switches
#####

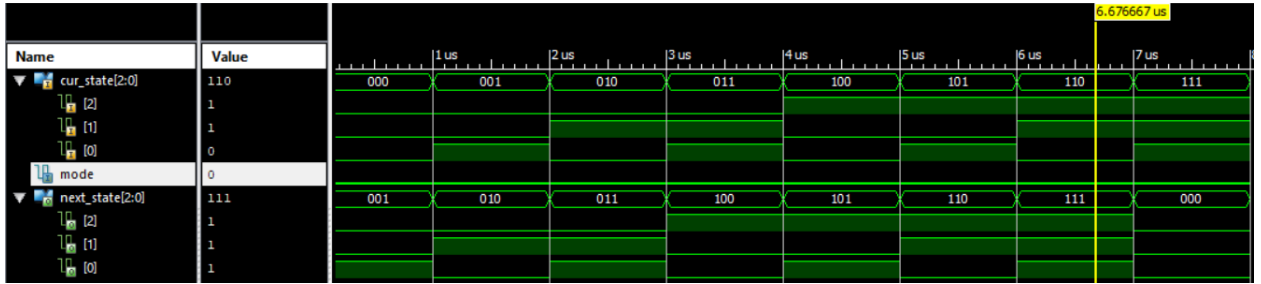
NET "TEST_ENABLE"     LOC = P80 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
NET "RESET"           LOC = P79 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

```

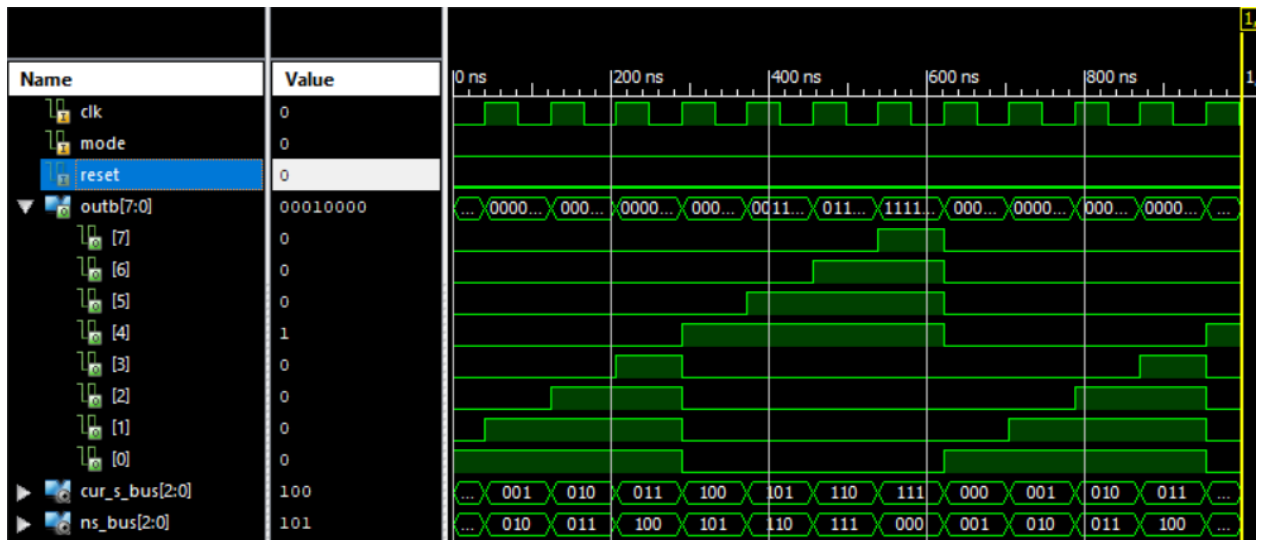
6) Симулюю работу OutputLogic :



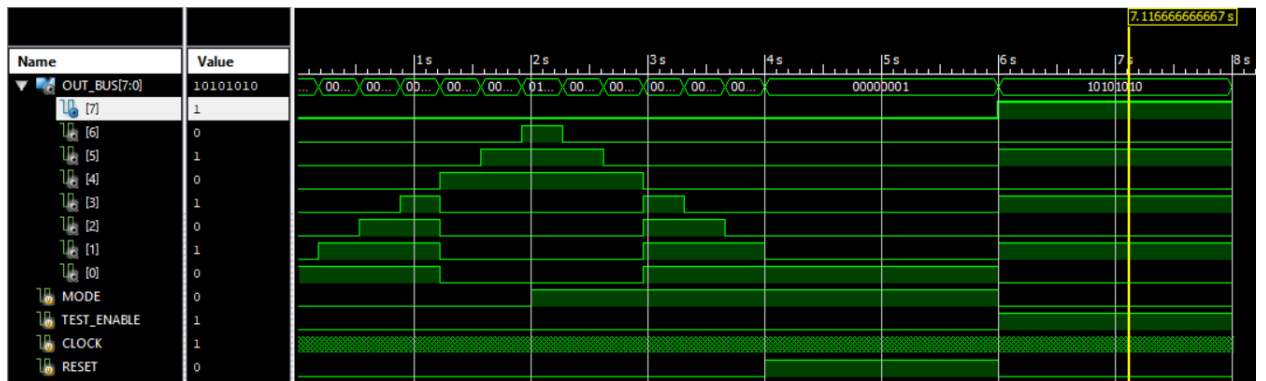
7) Симулюю работу TransitionLogic :



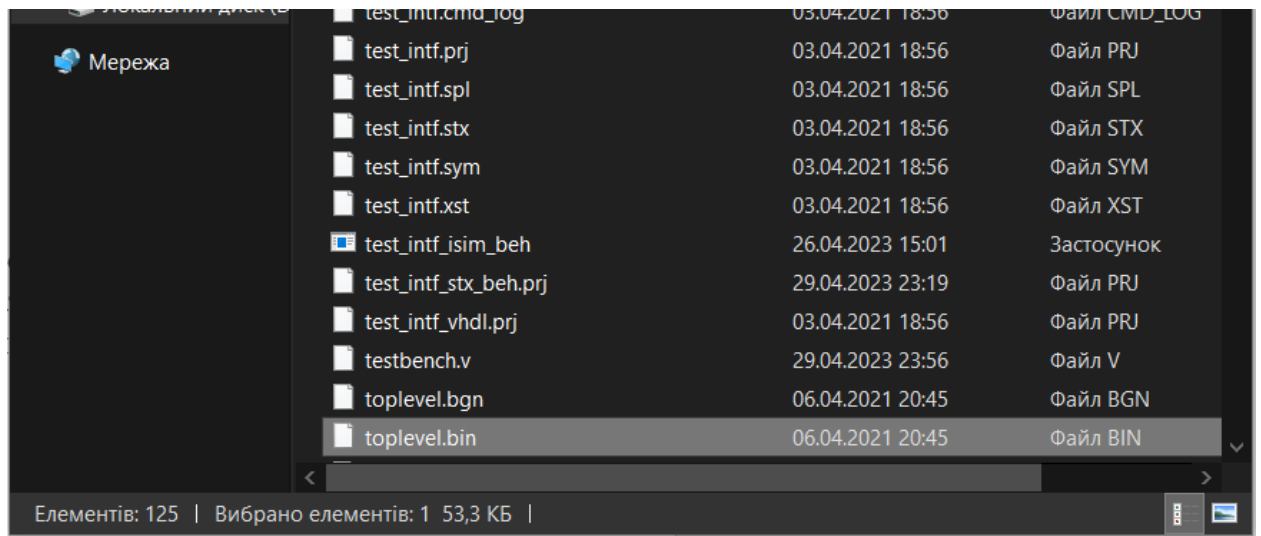
8) Симулюю роботу LightController.sch



9) Симулюю работу TopLevel.sch :



10) Генерую BIN файл :



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