



**Звіт**

до лабораторної роботи № 2

з дисципліни: **«МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ»**

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

Варіант 2

Виконав:

ст. гр. КІ-201

Волинець М.М.

Прийняв:

Козак Н. Б.

**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. Промодельовати роботу окремих частин автомата та автомата в цілому за допомогою симулятора
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
9. Підготувати і захистити звіт

## Завдання:

### Варіант – 1:

- 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     | 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 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 керування швидкістю роботи (*SPEED*):
  - Якщо *SPEED=0* то автомат працює зі швидкістю, визначеною за замовчуванням.
  - Якщо *SPEED=1* то автомат працює зі швидкістю, **В 2 РАЗИ ВИЩОЮ** ніж в режимі (*SPEED=0*).
- Для керування сигналом *MODE* використати будь який з 8 *DIP* перемикачів (див. **Додаток – 1**).
- Для керування сигналами *RESET/SPEED* використати будь які з *PUSH BUTTON* кнопок (див. **Додаток – 1**).

## Хід роботи:

1. У середовищі Xilinx ISE створив новий проєкт. Налаштував цільову FPGA, обрала інструменти для синтезу і симуляції.
2. Додав VHDL файл Output\_Logic до проєкту та імплементував інтерфейс логіки формування вихідних сигналів, а також логічні вирази для формування кожного вихідного сигналу, залежно від поточного стану автомата.

```
1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5
6 entity out_logic_intf is
7 Port ( IN_BUS : in  std_logic_vector(2 downto 0);
8       OUT_BUS : out std_logic_vector(7 downto 0)
9       );
10 end out_logic_intf;
11
12 architecture out_logic_arch of out_logic_intf is
13
14 begin
15     OUT_BUS(0) <= (not(IN_BUS(2)) and not(IN_BUS(1)) and not(IN_BUS(0))); --000 0
16     OUT_BUS(1) <= (not(IN_BUS(2)) and not(IN_BUS(1)) and(IN_BUS(0))); --001 1
17     OUT_BUS(2) <= (not(IN_BUS(2)) and (IN_BUS(1)) and not(IN_BUS(0))); --010 2
18     OUT_BUS(3) <= (not(IN_BUS(2)) and (IN_BUS(1)) and (IN_BUS(0))); --011
19     OUT_BUS(4) <= ((IN_BUS(2)) and not(IN_BUS(1)) and not(IN_BUS(0))); --100
20     OUT_BUS(5) <= ((IN_BUS(2)) and not(IN_BUS(1)) and (IN_BUS(0))); --101
21     OUT_BUS(6) <= ((IN_BUS(2)) and (IN_BUS(1)) and not(IN_BUS(0))); --110
22     OUT_BUS(7) <= ((IN_BUS(2)) and (IN_BUS(1)) and (IN_BUS(0))); --111 7
23 end out_logic_arch;
24
25
```

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

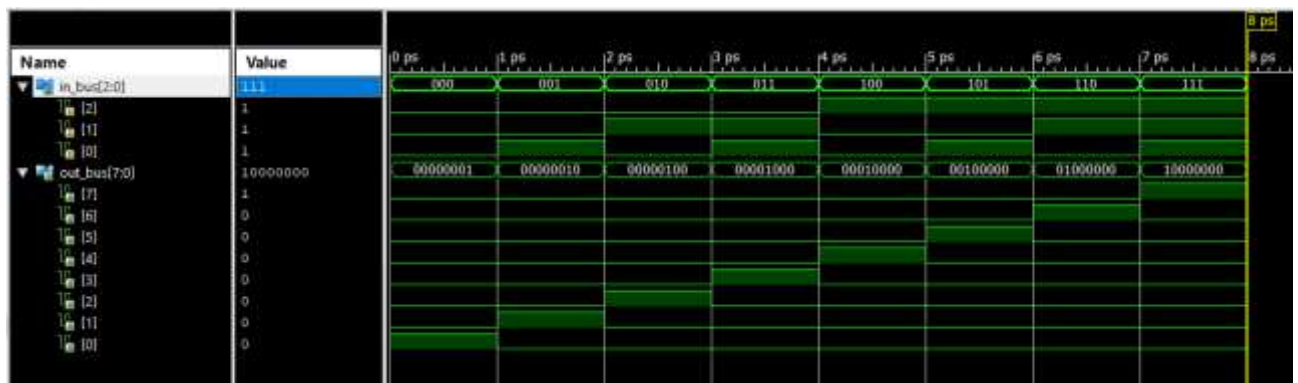


Рис.1 Таблиця істинності виводу сигналів

4. Додав до проєкту VHDL файл Transition\_Logіc, в якому реалізував логіку формування переходів.

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity transition_logic_intf is
5  Port ( CUR_STATE : in  std_logic_vector(2 downto 0);
6        MODE : in  std_logic;
7        NEXT_STATE : out std_logic_vector(2 downto 0)
8        );
9  end transition_logic_intf;
10
11 architecture transition_logic_arch of transition_logic_intf is
12
13 begin
14     NEXT_STATE(0) <= (not(MODE) and not(CUR_STATE(2))and not(CUR_STATE(1)) and not(CUR_STATE(0))) or
15                     (not(MODE) and not(CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) or
16                     (not(MODE) and (CUR_STATE(2))and not(CUR_STATE(1)) and not(CUR_STATE(0))) or
17                     (not(MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) or
18                     ((MODE) and not(CUR_STATE(2))and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
19                     ((MODE) and not(CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) or
20                     ((MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
21                     ((MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) ;
22
23     NEXT_STATE(1) <= (not(MODE) and not(CUR_STATE(2))and not (CUR_STATE(1)) and (CUR_STATE(0))) or
24                     (not(MODE) and not(CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) or
25                     (not(MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and (CUR_STATE(0))) or
26                     (not(MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not(CUR_STATE(0))) or
27                     ((MODE) and not(CUR_STATE(2))and not(CUR_STATE(1)) and not (CUR_STATE(0))) or
28                     ((MODE) and not(CUR_STATE(2))and (CUR_STATE(1)) and (CUR_STATE(0))) or
29                     ((MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and not(CUR_STATE(0))) or
30                     ((MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and (CUR_STATE(0))) ;
31
32     NEXT_STATE(2) <= (not(MODE) and not(CUR_STATE(2))and (CUR_STATE(1)) and (CUR_STATE(0))) or
33                     (not(MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
34                     (not(MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and (CUR_STATE(0))) or
35                     (not(MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) or
36                     ((MODE) and not(CUR_STATE(2))and not(CUR_STATE(1)) and not (CUR_STATE(0))) or
37                     ((MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and (CUR_STATE(0))) or
38                     ((MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not (CUR_STATE(0))) or
39                     ((MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and (CUR_STATE(0))) ;
40
41 end transition_logic_arch;
42
43
44

```

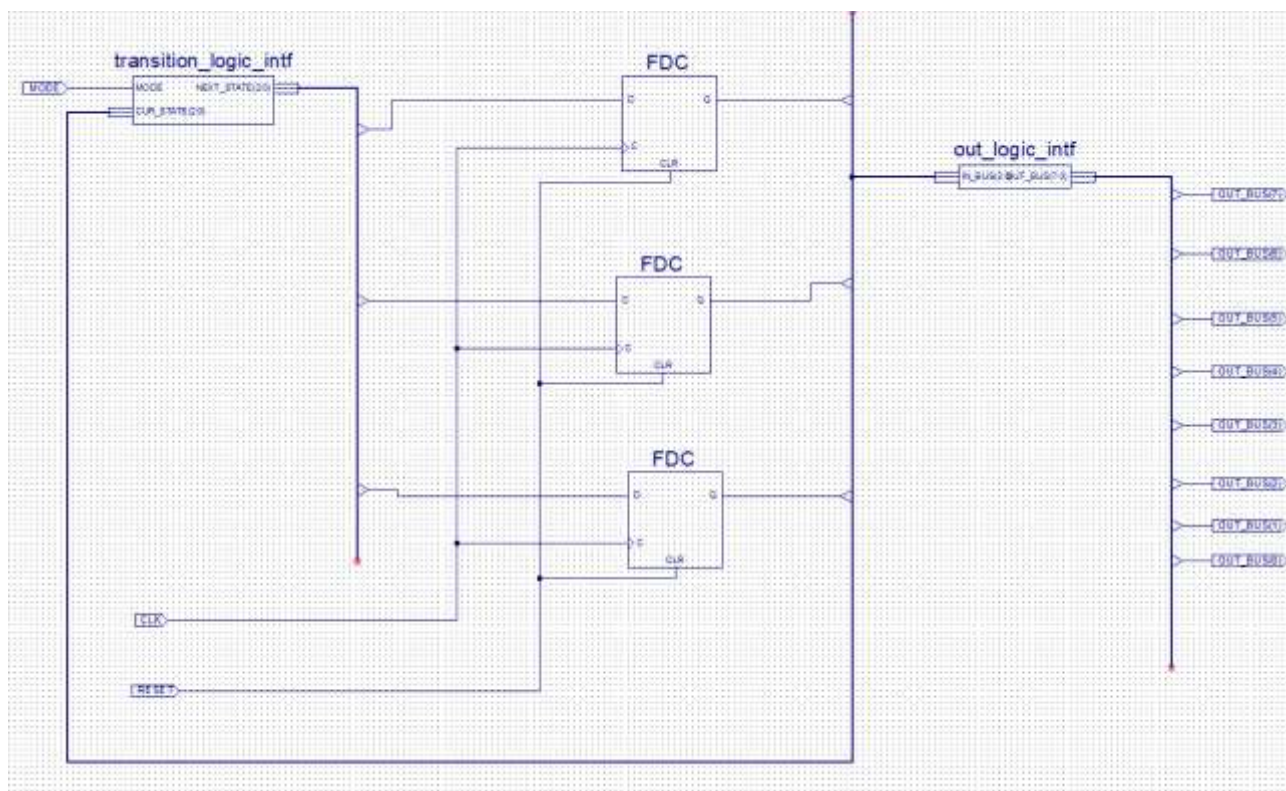
5. За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.





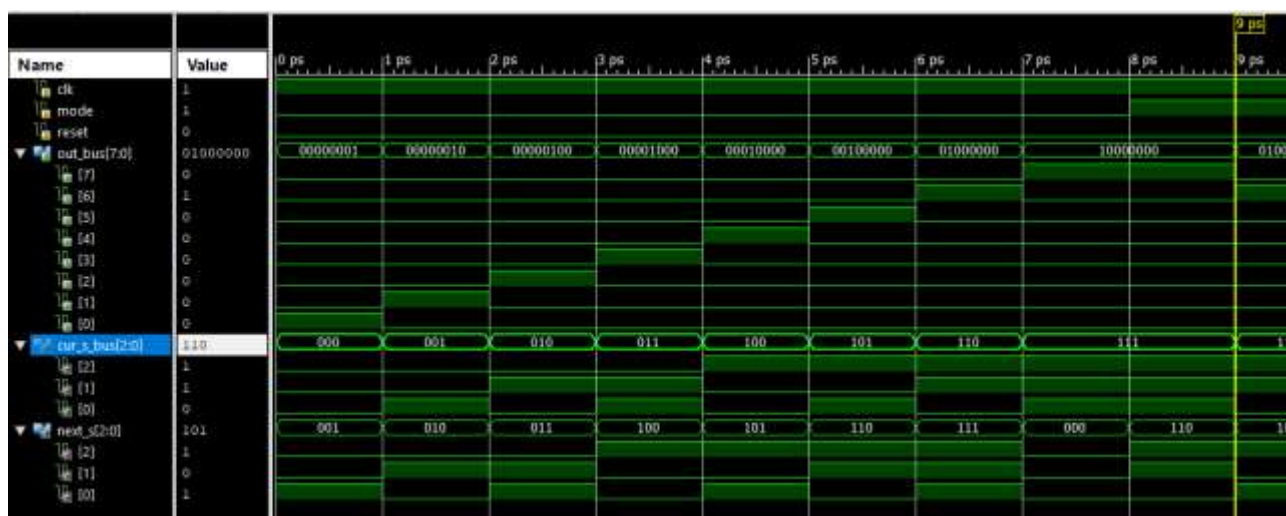
6. Додав до проекту Schematic файл Light\_Controller, виконав для нього команду Set as Top Module. Згенерував Schematic символи для файлів Output\_Logic і Transition\_Logic. Використовуючи новостворені символи та елементи з бібліотеки, реалізував у файлі Light\_Controller.sch пам'ять стану автомата.

out\_logic\_intf  
transition\_logic\_intf



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

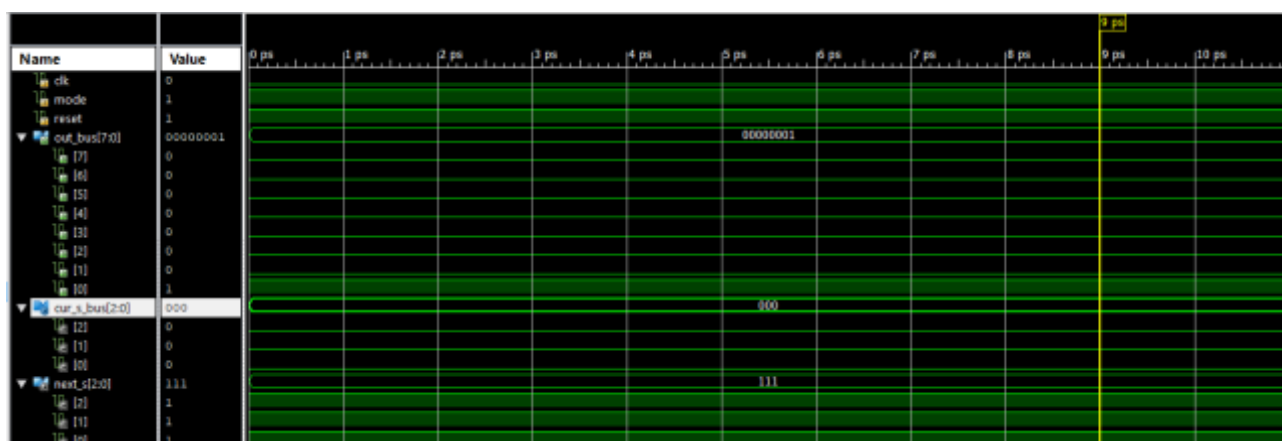
При MODE 0:



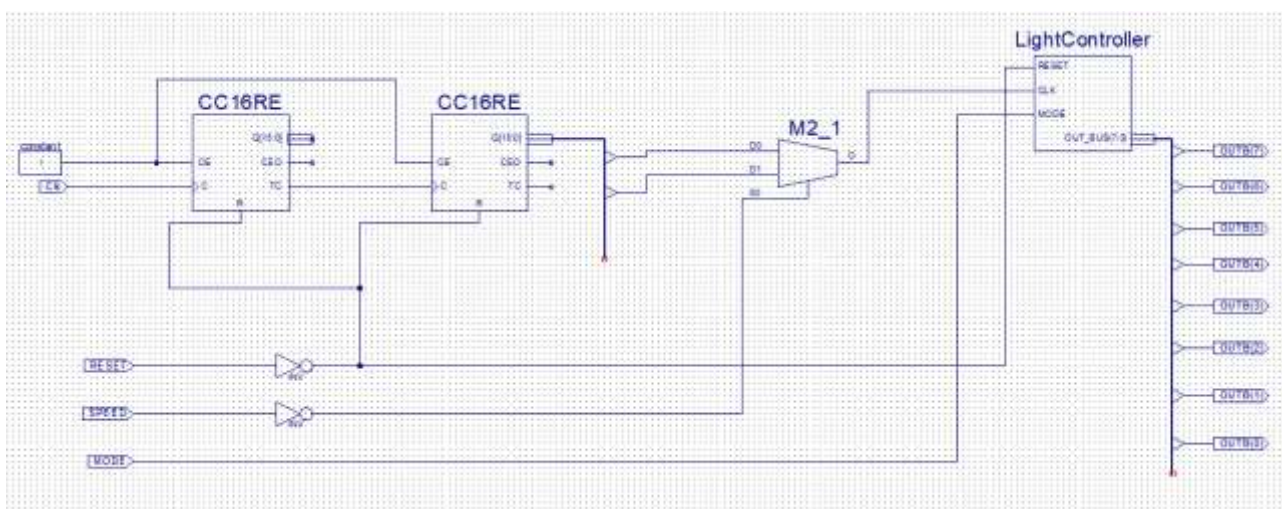
При MODE 1:



RESET:



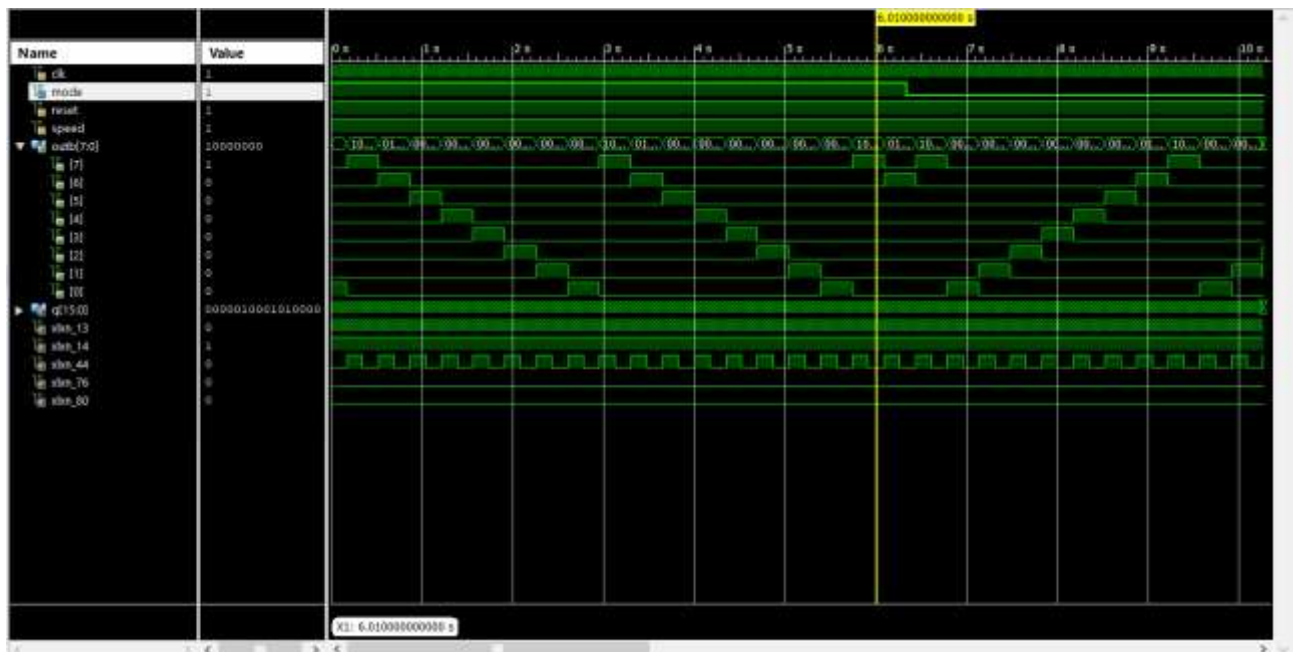
8. Додав до проекту Schematic файл Top\_Level, виконав для нього команду Set as Top Module. Згенерував Schematic символ для файлу Light\_Controller. Використовуючи новостворений символ та елементи з бібліотеки, реалізував у файлі подільник входної частоти та логіку сигналу SPEED.





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

SPEED =1, MODE = 1(до лінії), MODE = 0(після лінії).



Автомат працює відповідно до завдання.

10. Додала до проєкту Constraint файл та призначила виводам схеми фізичні виводи цільової FPGA.

```

#
# 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;

```



11. Згенерував файли прошиття.

|  |                  |          |       |
|--|------------------|----------|-------|
|  toplevel     | 06.05.2023 14:49 | Файл BIN | 54 КБ |
|  toplevel.bit | 06.05.2023 14:49 | Файл BIT | 54 КБ |

**Висновок:** під час виконання цієї лабораторної роботи я реалізувала цифровий автомат світлових ефектів у середовищі Xilinx ISE і стендом Elbert V2 - Spartan 3A FPGA. Я реалізувала схему автомату та провела симуляцію його роботи.