

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 – Spartan3A FPGA»

***Варіант 2***

Виконав: ст. гр. КІ-202

Соніч Орест

Прийняв:

Козак Н.Б.

**Львів 2024**  
**ЛАБОРАТОРНА РОБОТА №2**  
**Структурний опис цифрового автомата Перевірка роботи автомата за**  
**допомогою стенда Elbert V2 – Spartan3A FPGA**

**Мета роботи:** На базі стенда реалізувати цифровий автомат світлових ефектів

**Вхідні параметри**

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	1	1	1	1	1	0	0	0
5	1	1	1	1	1	1	0	0
6	1	1	1	1	1	1	1	0
7	1	1	1	1	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 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 НИЖЧОЮ 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**).

### Порядок виконання лабораторної роботи.

1. Інтерфейс пристрою та функціонал реалізувати згідно отриманого варіанту завдання.
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів.
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів.
4. Згенерувати символи для 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. Пам'ять станів реалізувати за допомогою графічних компонентів з бібліотеки.
6. Промодельовати 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 на FPGA.
8. Згенерувати файл та перевірити роботу за допомогою стенда Elbert V2 – Spartan3A FPGA.
9. Підготувати і захистити звіт.

### Виконання лабораторної роботи:

MODE	CURR_STATE(2)	CURR_STATE(1)	CURR_STATE(0)	NEXT_STATE(2)	NEXT_STATE(1)	NEXT_STATE(0)
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	0	1	1	0	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

*Рис.1.1 (Логіка переходів для всіх станів автомата)*

## Логіка переходів на мові VHDL

$\text{NEXT\_STATE}(0) = \text{not}(\text{CURR\_STATE}(0));$

$\text{NEXT\_STATE}(1) = ((\text{not}(\text{MODE}) \text{ and } \text{not}(\text{CURR\_STATE}(1)) \text{ and } \text{CURR\_STATE}(0)) \text{ or } (\text{not}(\text{MODE}) \text{ and } \text{CURR\_STATE}(1) \text{ and } \text{not}(\text{CURR\_STATE}(0))) \text{ or } (\text{MODE} \text{ and } \text{not}(\text{CURR\_STATE}(1)) \text{ and } \text{not}(\text{CURR\_STATE}(0))) \text{ or } (\text{MODE} \text{ and } \text{CURR\_STATE}(1) \text{ and } \text{CURR\_STATE}(0)));$

$\text{NEXT\_STATE}(2) \leq ((\text{not}(\text{MODE}) \text{ and } \text{CURR\_STATE}(2) \text{ and } \text{not}(\text{CURR\_STATE}(1))) \text{ or } (\text{CURR\_STATE}(2) \text{ and } \text{CURR\_STATE}(1) \text{ and } \text{not}(\text{CURR\_STATE}(0))) \text{ or } (\text{MODE} \text{ and } \text{CURR\_STATE}(2) \text{ and } \text{CURR\_STATE}(0)) \text{ or } (\text{not}(\text{MODE}) \text{ and } \text{not}(\text{CURR\_STATE}(2)) \text{ and } \text{CURR\_STATE}(1) \text{ and } \text{CURR\_STATE}(0)) \text{ or } (\text{MODE} \text{ and } \text{not}(\text{CURR\_STATE}(2)) \text{ and } \text{not}(\text{CURR\_STATE}(1)) \text{ and } \text{not}(\text{CURR\_STATE}(0))));$

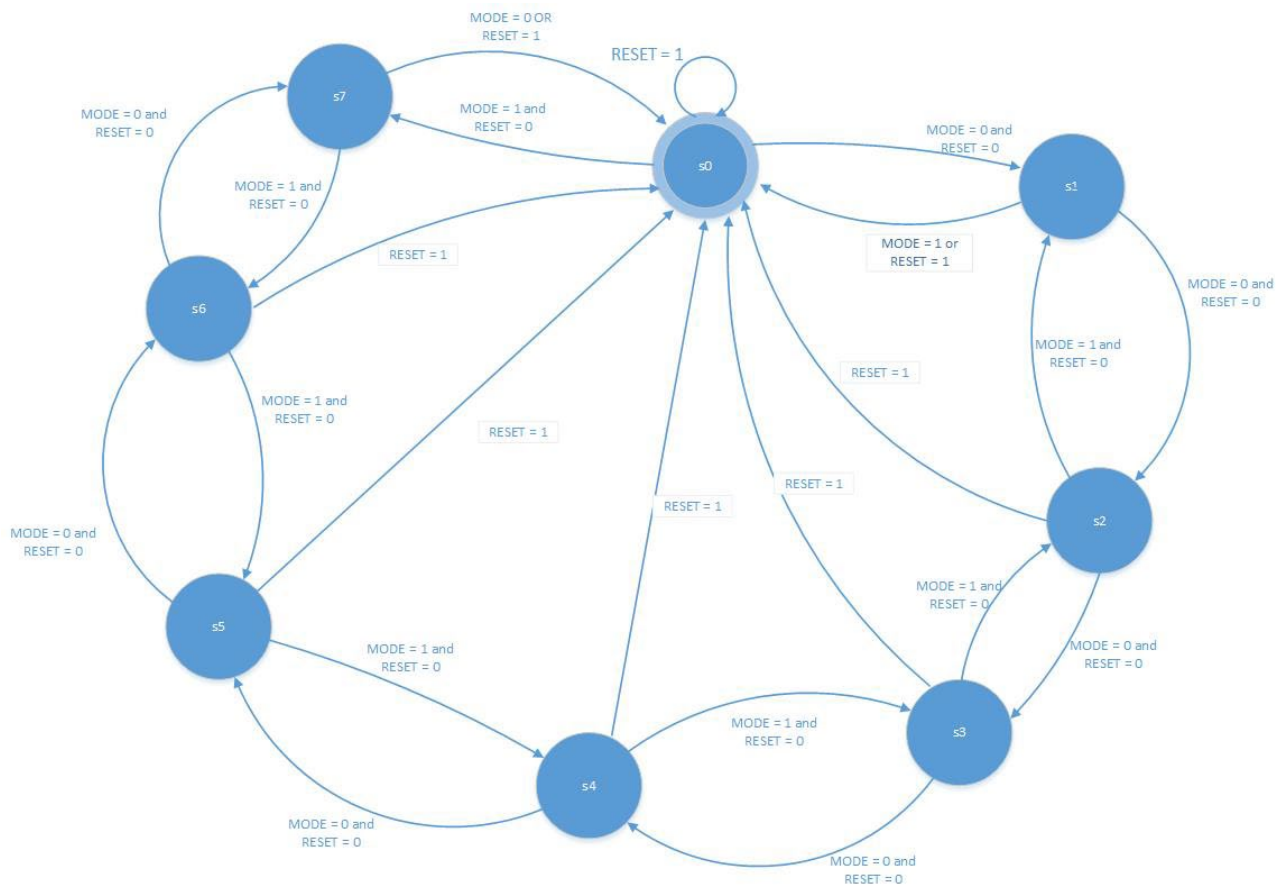


Рис.1.2. Граф переходів автомата між станами.

## Логіка формування вихідних сигналів

$\text{OUT\_BUS}(0) \leq '1'$  after 1 ns;

$\text{OUT\_BUS}(1) \leq (\text{IN\_BUS}(2) \text{ or } \text{IN\_BUS}(1) \text{ or } \text{IN\_BUS}(0))$  after 1 ns;

$\text{OUT\_BUS}(2) \leq (\text{IN\_BUS}(2) \text{ or } \text{IN\_BUS}(1))$  after 1 ns;

$OUT\_BUS(3) \leq ((IN\_BUS(2)) \text{ or } (IN\_BUS(1) \text{ and } IN\_BUS(0)))$  after 1 ns;

$OUT\_BUS(4) \leq (IN\_BUS(2))$  after 1 ns;

$OUT\_BUS(5) \leq ((IN\_BUS(2) \text{ and } IN\_BUS(0)) \text{ or } (IN\_BUS(2) \text{ and } IN\_BUS(1)))$  after 1 ns;

$OUT\_BUS(6) \leq (IN\_BUS(2) \text{ and } IN\_BUS(1))$  after 1 ns;

$OUT\_BUS(7) \leq (IN\_BUS(2) \text{ and } IN\_BUS(1) \text{ and } IN\_BUS(0))$  after 1 ns;

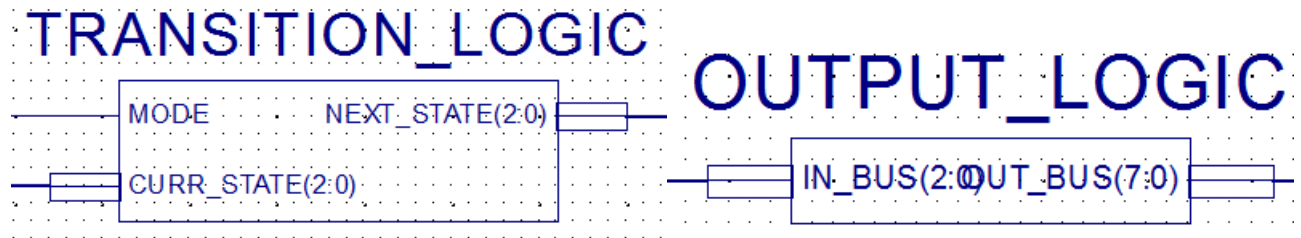


Рис.1.2 (Згенеровані схематичні схеми)

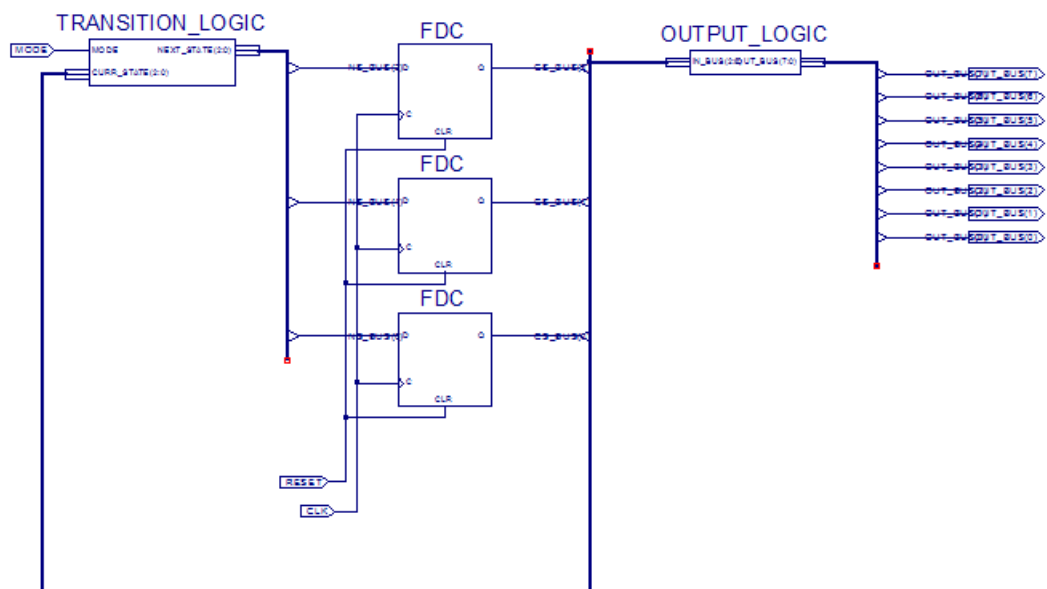
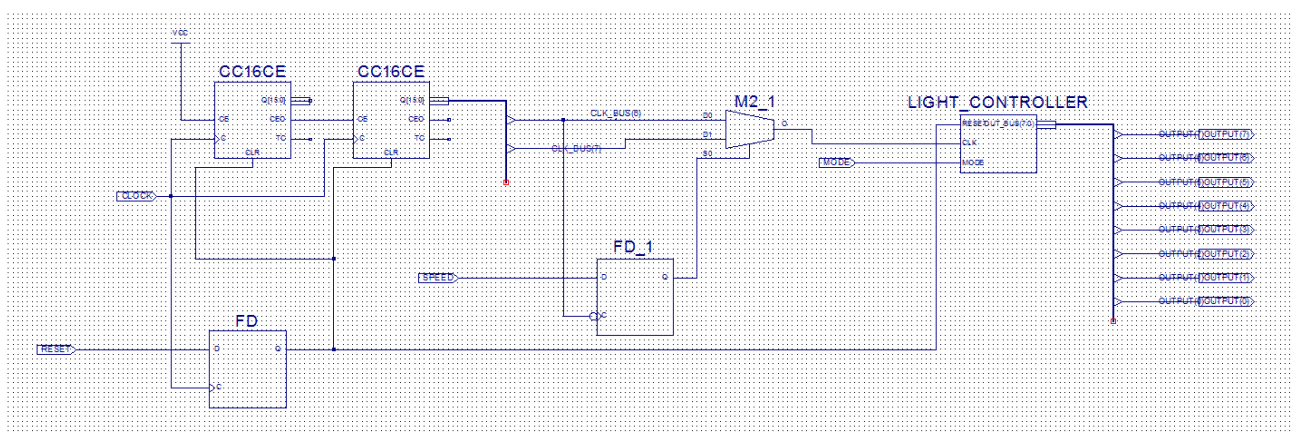
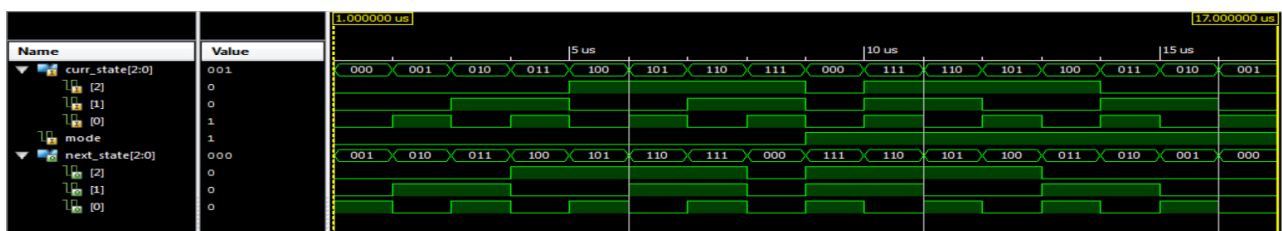


Рис.1.3 (Інтеграція всіх створених компонентів разом з пам'ятю станів автомата)

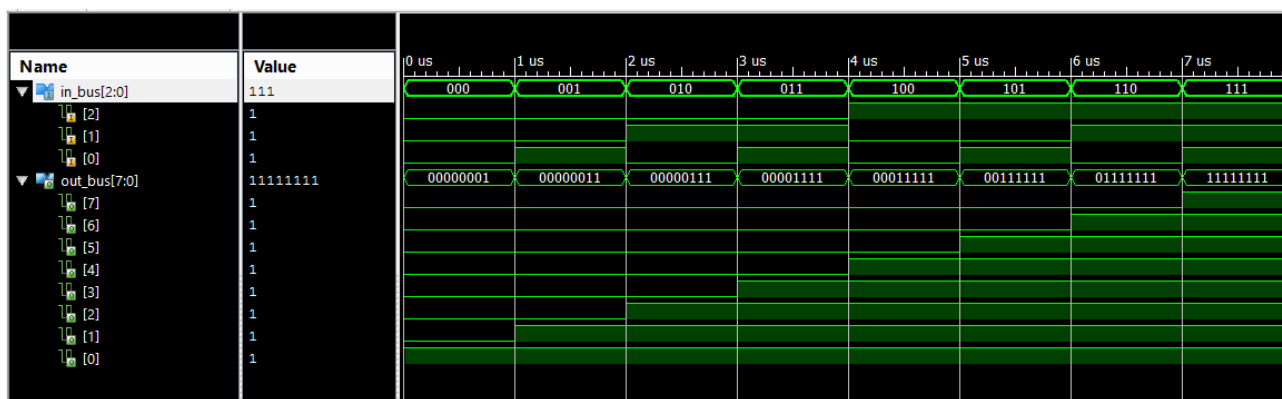


*Рис.1.4 (Автомат світлових сигналів та подільник тактового сигналу)*

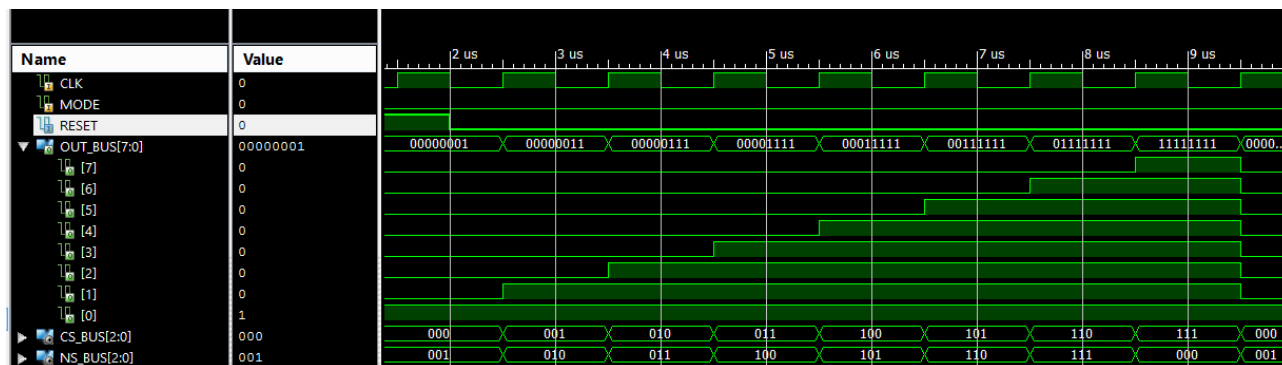
## Демонстрація симуляції схем наведених зверху



*Рис.1.5 Результати симуляції логіки переходів в ISim.*



*Рис. 1.6. Результати симуляції логіки вихідних сигналів в ISim.*



*Рис.1.7. Результати симуляції автомата (MODE = 0, RESET = 0).*

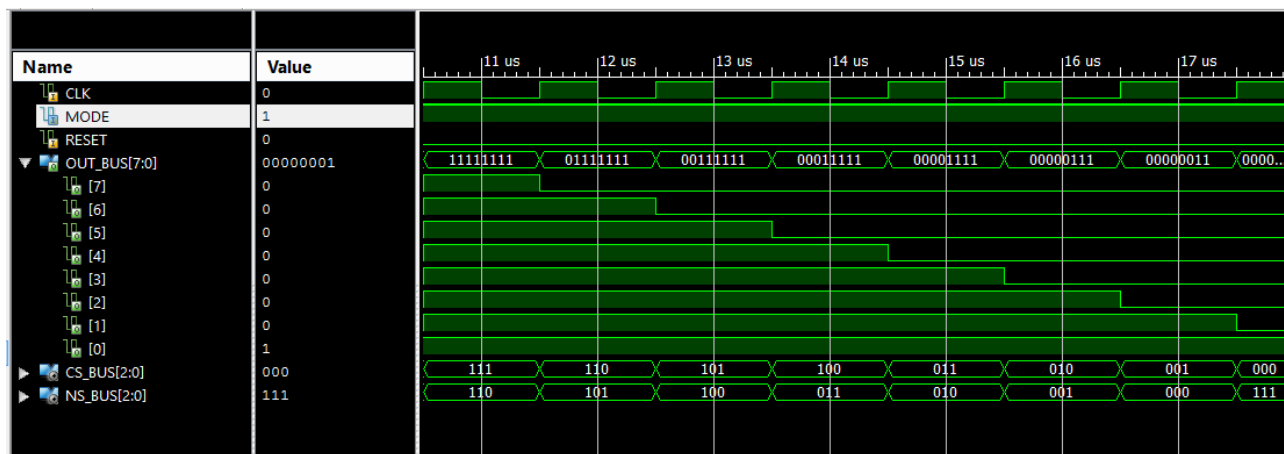


Рис.1.8. Результати симуляції автомата ( $MODE = 1$ ,  $RESET = 0$ ).

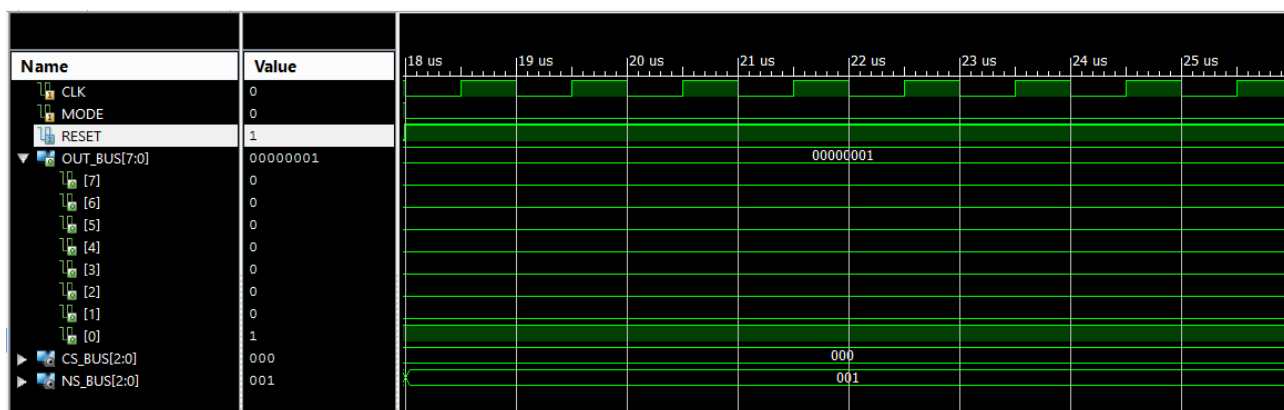


Рис.1.9. Результати симуляції автомата ( $MODE = 0$ ,  $RESET = 1$ ).

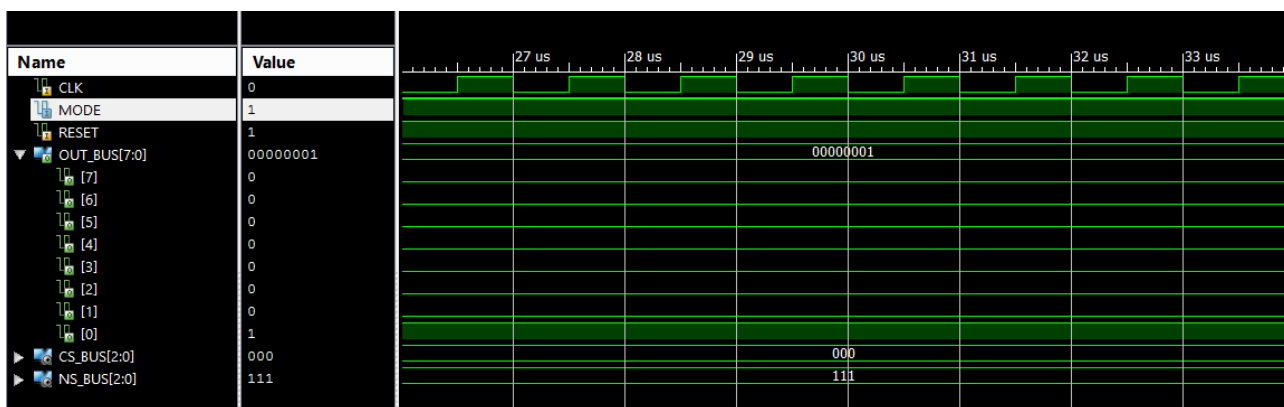


Рис.2.1. Результати симуляції автомата ( $MODE = 1$ ,  $RESET = 1$ ).

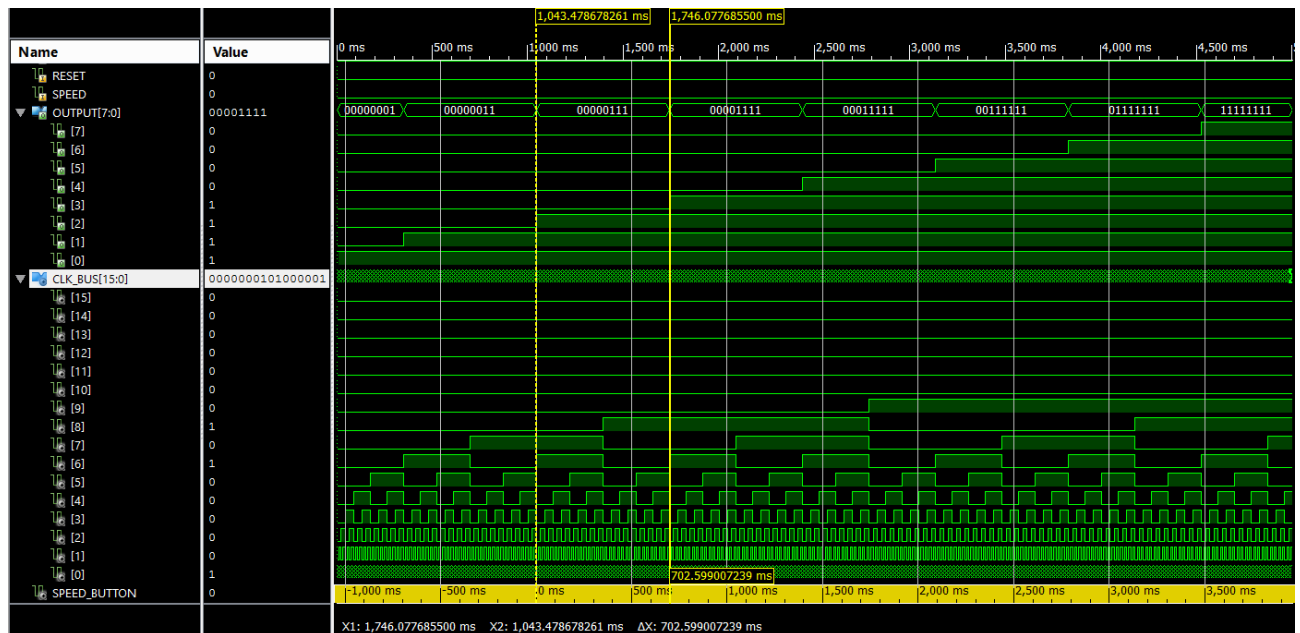


Рис.2.2. Результати симуляції автомата ( $SPEED = 0$ ).

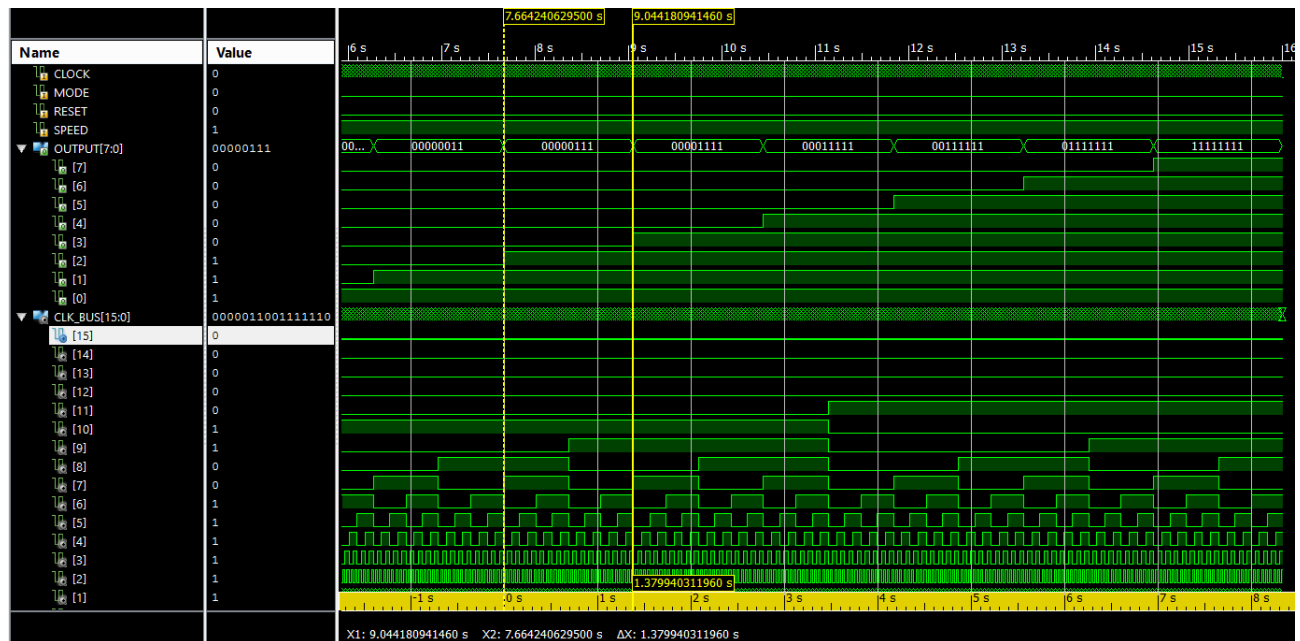


Рис.2.3. Результати симуляції автомата ( $SPEED = 1$ ).

## Реалізація Test Branch



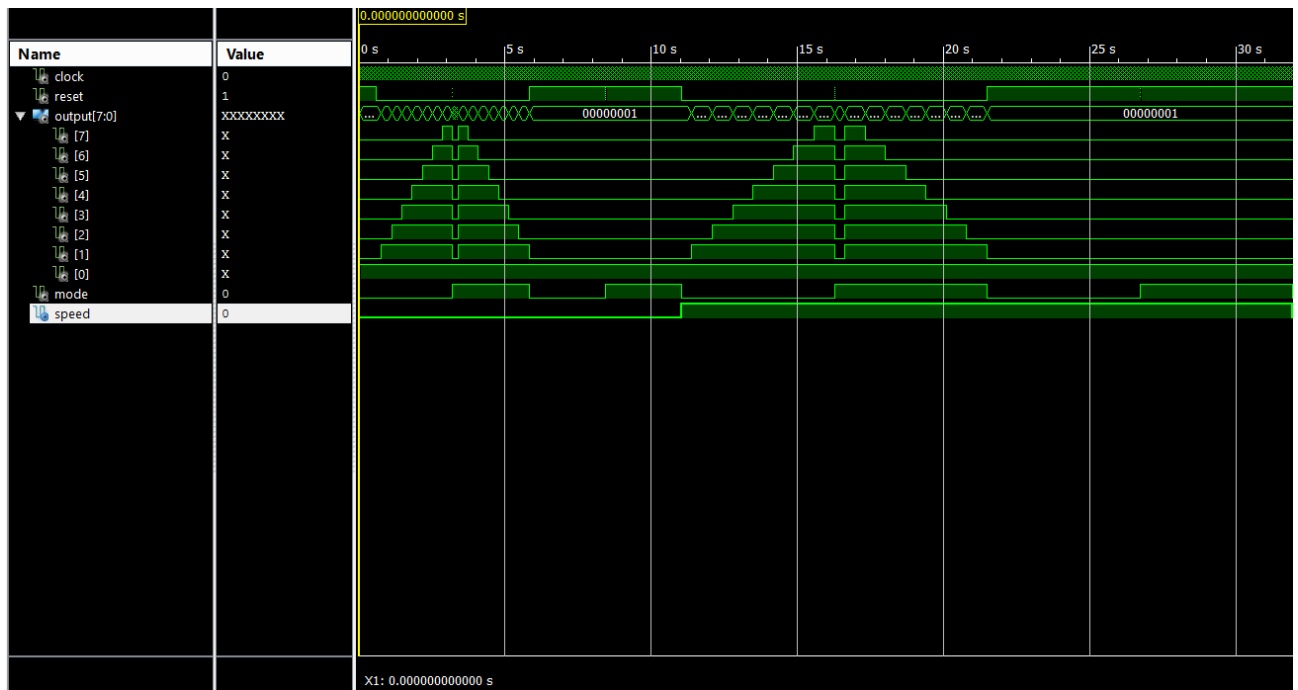


Рис.2.4 (Часова діаграма)

```
ISim>
# run 30s
ISim>
# run 5s
ISim>
```

Рис.2.5 (Консоль під час тестування)

```

1  #-----
2  #
3  #----- UCF for ElbertV2 Development Board -----
4  CONFIG VCCAUX = "3.3" ;
5
6  # Clock 12 MHz
7  NET "CLOCK" LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
8
9  #----- LED -----
10 #
11 #-----
12
13 NET "OUTPUT (0) " LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
14 NET "OUTPUT (1) " LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
15 NET "OUTPUT (2) " LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
16 NET "OUTPUT (3) " LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
17 NET "OUTPUT (4) " LOC = P50 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
18 NET "OUTPUT (5) " LOC = P51 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
19 NET "OUTPUT (6) " LOC = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
20 NET "OUTPUT (7) " LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
21
22 #----- DP Switches -----
23 #
24 #-----
25
26 NET "MODE" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
27
28 #----- Switches -----
29 #
30 #-----
31
32 NET "RESET" LOC = P80 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
33 NET "SPEED" LOC = P79 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
34
35

```

Рис.2.6. Призначення фізичних входів та виходів.

**Висновок:**

В ході виконання цієї лабораторної роботи я реалізував на базі стенда Elbert V2 – Spartan3A FPGA цифровий автомат світлових ефектів згідно заданих вимог.