**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп’ютерних технологій, автоматики та метрології**

**кафедра “Електронних обчислювальних машин”**

****

**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 16***

Виконав: ст. гр. КІ-202

Петренко В.А.

Прийняв:

Козак Н.Б

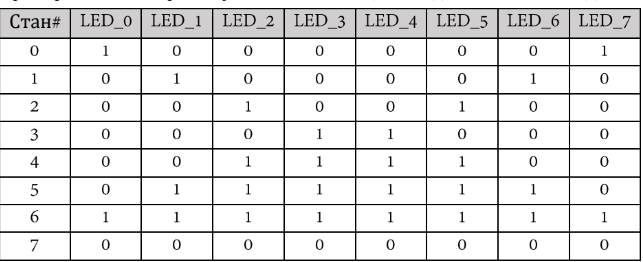
**Львів 2024**

**ЛАБОРАТОРНА РОБОТА №2**

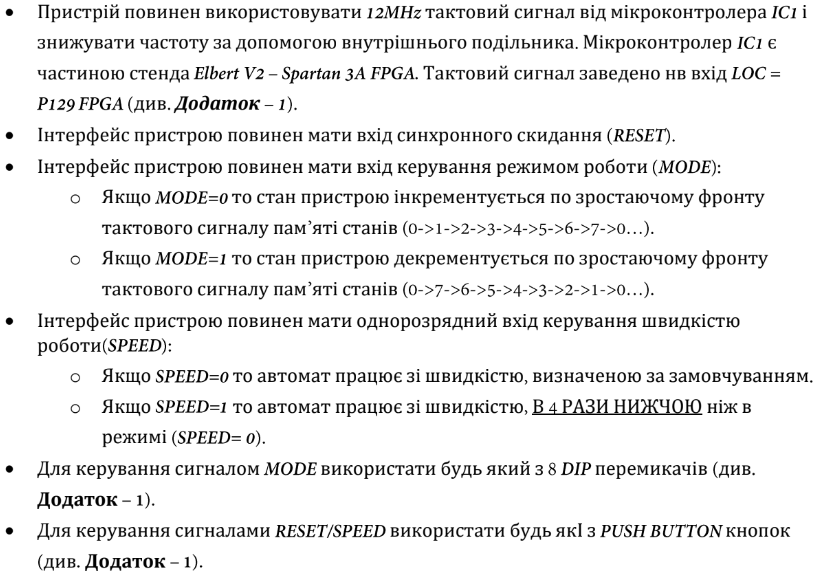
**Структурний опис цифрового автомата Перевірка роботи автомата за допомогою стенда Elbert V2 – Spartan3A FPGA**

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

**Вхідні параметри**

****

**Варіант 4.**

****

**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. Промоделювати роботу окремих частин автомата та автомата вцілому за допомогою симулятора ISim.
7. Інтегрувати створений автомат зі стендом додати подільник частоти для вхідного тактовового сигналу призначити фізичні виводи на FPGA.
8. Згенерувати файал 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**

NEXT\_STATE(0) = not(CURR\_STATE(0));

NEXT\_STATE(1) = ((not(MODE) and not(CURR\_STATE(1)) and CURR\_STATE(0)) or (not(MODE) and CURR\_STATE(1) and not(CURR\_STATE(0))) or (MODE and not(CURR\_STATE(1)) and not(CURR\_STATE(0))) or (MODE and CURR\_STATE(1) and CURR\_STATE(0)));

NEXT\_STATE(2) <= ((not(MODE) and CURR\_STATE(2) and not(CURR\_STATE(1))) or (CURR\_STATE(2) and CURR\_STATE(1) and not(CURR\_STATE(0))) or (MODE and CURR\_STATE(2) and CURR\_STATE(0)) or (not(MODE) and not(CURR\_STATE(2)) and CURR\_STATE(1) and CURR\_STATE(0)) or (MODE and not(CURR\_STATE(2)) and not(CURR\_STATE(1)) and not(CURR\_STATE(0))));

**Логіка формування вихідних сигналів**

OUT\_BUS(0) <= (((not (IN\_BUS(2)) and not (IN\_BUS(1)) and not (IN\_BUS(0))) or (IN\_BUS(2) and IN\_BUS(1) and not (IN\_BUS(0))))) after 1 ns;

OUT\_BUS(1) <= (((not (IN\_BUS(1)) and IN\_BUS(0)) or (IN\_BUS(2) and IN\_BUS(1) and not (IN\_BUS(0))))) after 1 ns;

OUT\_BUS(2) <= (((IN\_BUS(1) and not(IN\_BUS(0))) or (IN\_BUS(2) and not (IN\_BUS(1))))) after 1 ns;

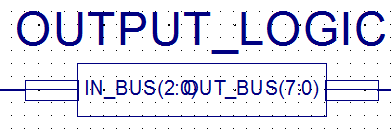
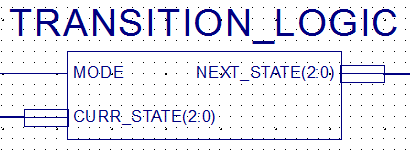
OUT\_BUS(3) <= (((IN\_BUS(2) and not(IN\_BUS(1))) or (IN\_BUS(2) and not(IN\_BUS(0))) or (not(IN\_BUS(2)) and IN\_BUS(1) and IN\_BUS(0)))) after 1 ns;

OUT\_BUS(4) <= (((IN\_BUS(2) and not(IN\_BUS(1))) or (IN\_BUS(2) and not(IN\_BUS(0))) or (not(IN\_BUS(2)) and IN\_BUS(1) and IN\_BUS(0)))) after 1 ns;

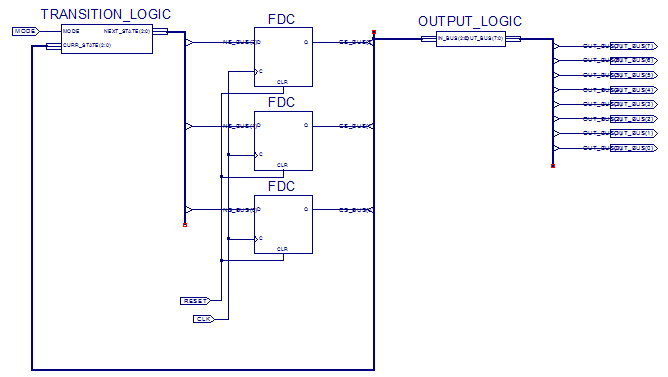
OUT\_BUS(5) <= (((IN\_BUS(1) and not(IN\_BUS(0))) or (IN\_BUS(2) and not (IN\_BUS(1))))) after 1 ns;

OUT\_BUS(6) <= (((not (IN\_BUS(1)) and IN\_BUS(0)) or (IN\_BUS(2) and IN\_BUS(1) and not (IN\_BUS(0))))) after 1 ns;

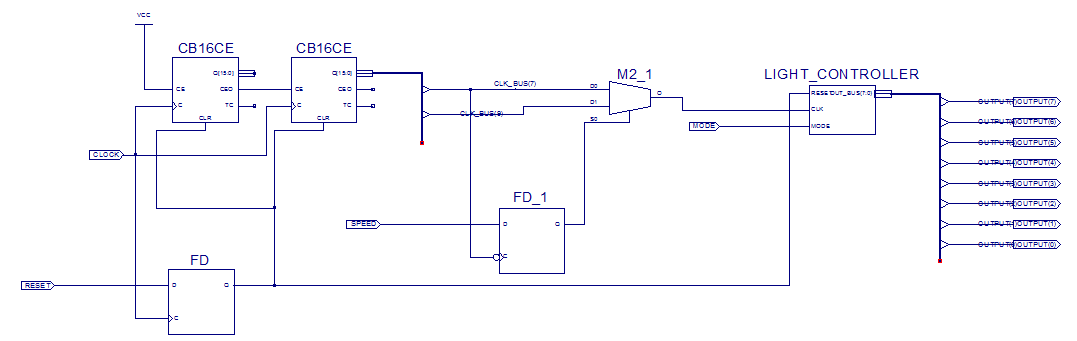
OUT\_BUS(7) <= (((not (IN\_BUS(2)) and not (IN\_BUS(1)) and not (IN\_BUS(0))) or (IN\_BUS(2) and IN\_BUS(1) and not (IN\_BUS(0))))) after 1 ns;



***Рис.1.2 (Згенеровані схематичні схеми)***



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

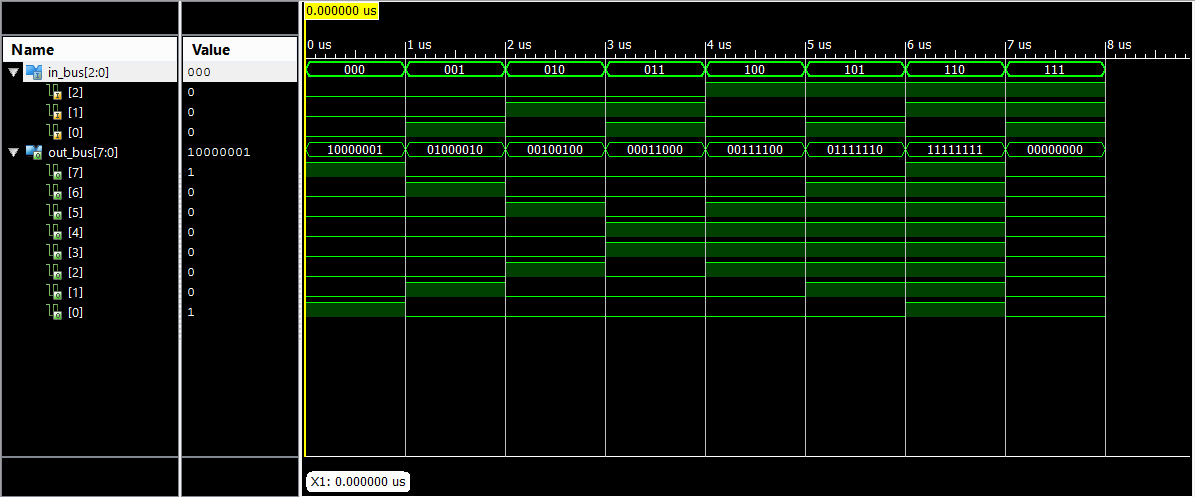
******

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

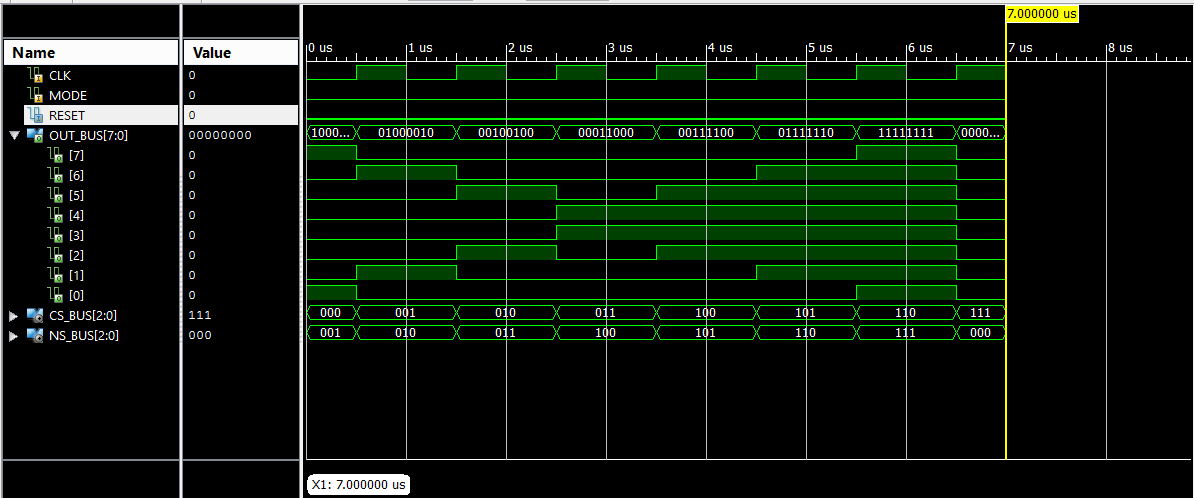
**Демонстрація симуляції схем наведених зверху**



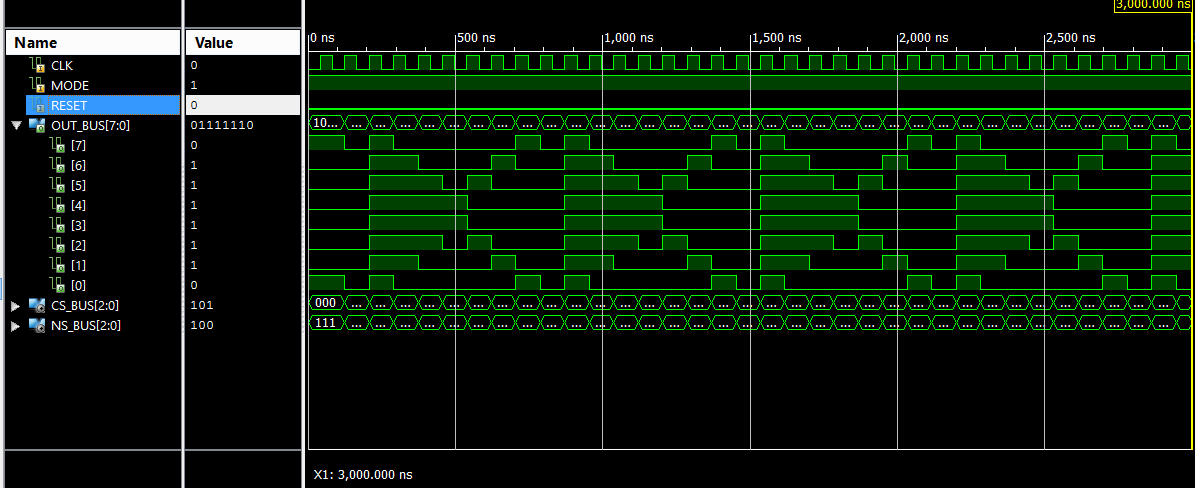
***Рис.1.5 (Результати симуляції логіки переходів в ISim)***

******

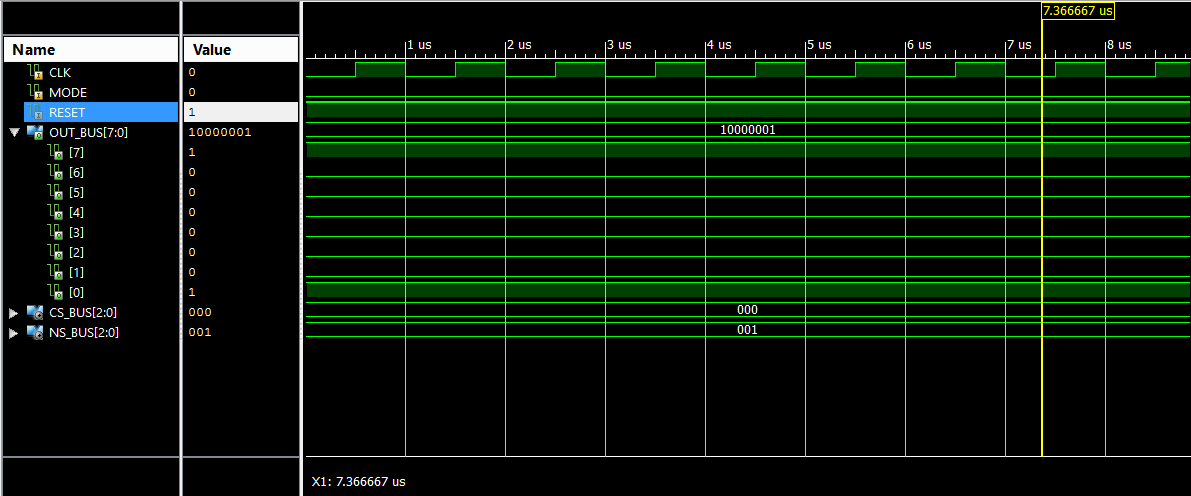
**Рис.1.6 (Результати симуляції логіки вихідних сигналів в ISim)**

******

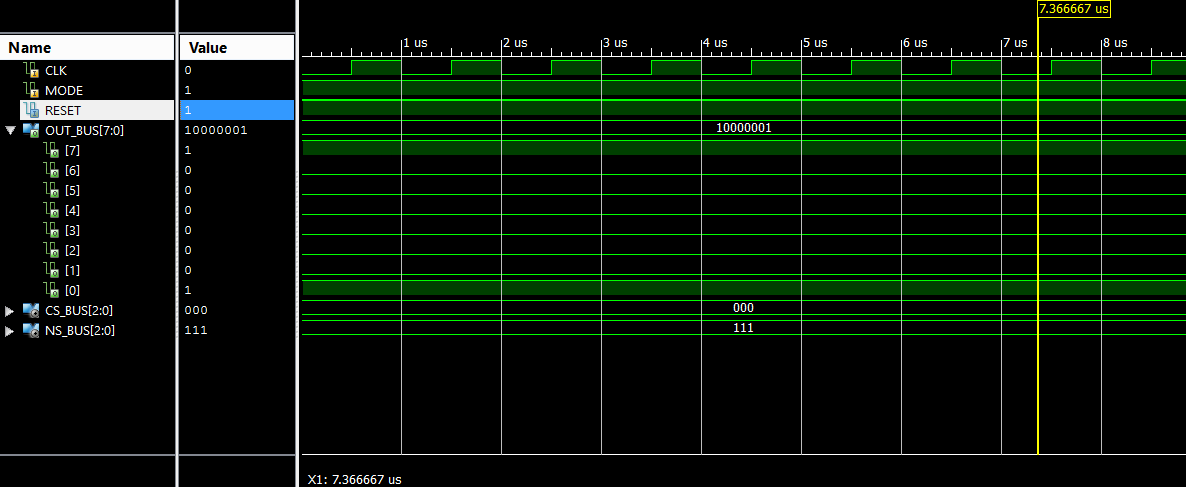
**Рис.1.7 (Результати симуляції автомата (MODE = 0, RESET = 0))**

******

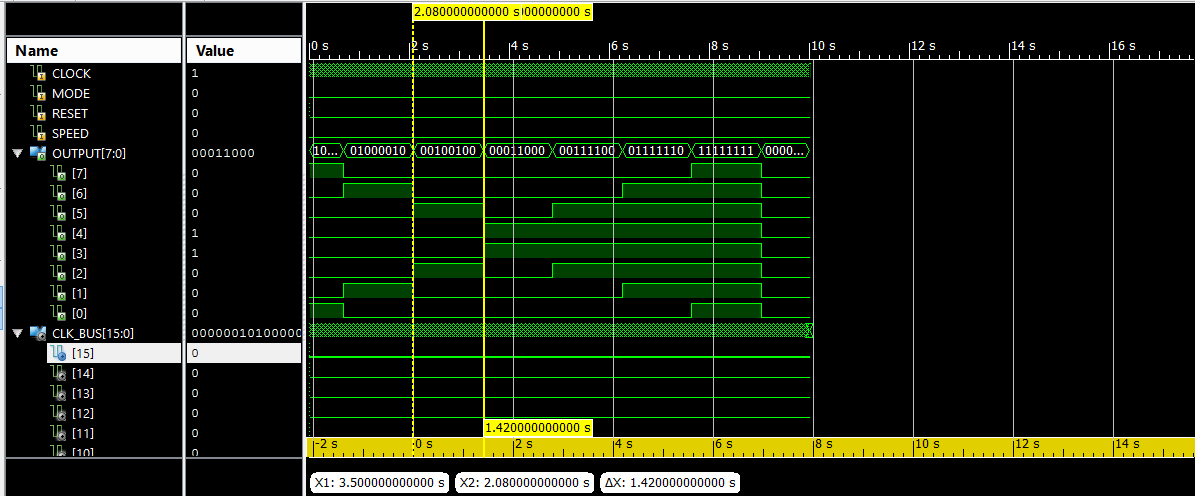
**Рис.1.8 (Результати симуляції автомата (MODE = 1, RESET = 0))**

******

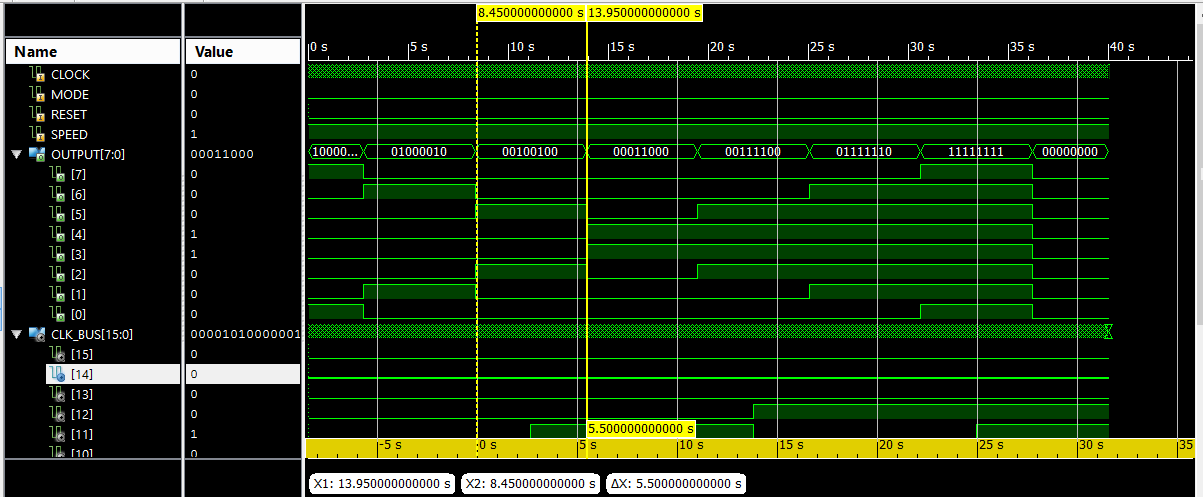
**Рис.1.9 (Результати симуляції автомата (MODE = 0, RESET = 1))**

****

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

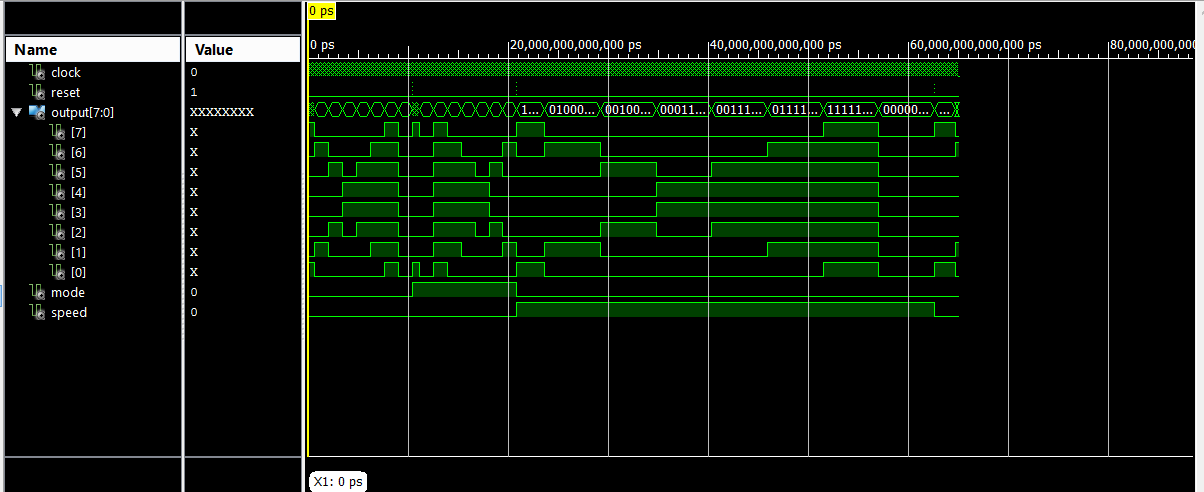
****

**Рис.2.1.1 (**Р**езультати симуляції фінальної схеми (MODE = 0, RESET = 0, SPEED = 0))**



**Рис.2.1.2 (**Р**езультати симуляції фінальної схеми (MODE = 0, RESET = 0, SPEED = 1))**

**Реалізація Test Brench**

****

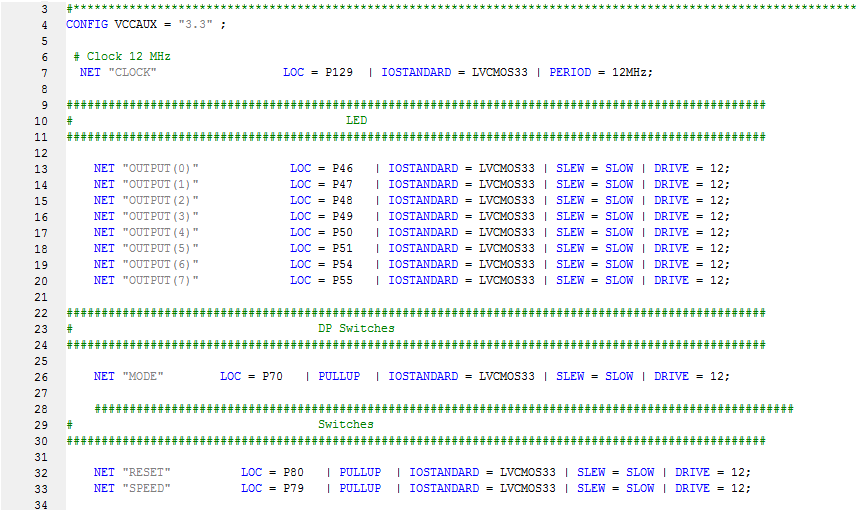
**Рис.2.2 (Часова діаграма)**

****

**Рис.2.3 (Консоль під час тестування)**



***Рис.2.4 (Граф переходів автомата між станами)***

****

**Рис.2.2 (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.