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 23***

Виконав

Студент групи КІ-201

Равчак В.А.

Перевірив:  
Козак Н.Б.

**Львів 2024**

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

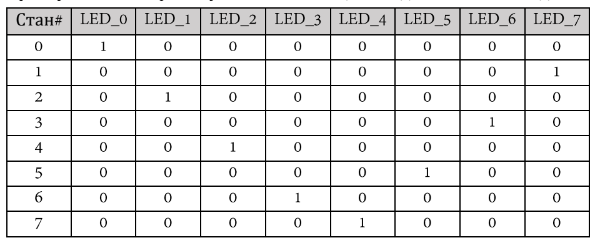
згідно заданих вимог.

**Етапи 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.

**Варіант виконання роботи:**

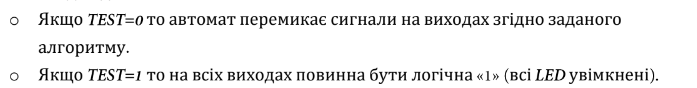
Пристрій повинен реалізувати комбінацій вихідних сигналів згідно таблиці:



* Пристрій повинен використовувати тактовий сигнал 12MHz від мікроконтролера і знижувати частоту за допомогою внутрішнього подільника Мікроконтролер є частиною стенда Elbert V2 – Spartan3A FPGA. Тактовий сигнал заведено на вхід LOC = P129 FPGA.
* Інтерфейс пристрою повинен мати вхід синхронного скидання (RESET).
* Інтерфейс пристрою повинен мати вхід керування режимом роботи (MODE):



* Інтерфейс пристрою повинен мати однорозрядний вхід (TEST) для подачі логічної <1> на всі виходи:



* Для керування сигналом MODE використати будь який з 8 DIP перемикачів.
* Для керування сигналами RESET/TEST використати будь які з PUSH BUTTON кнопок.

**Виконання роботи:**

1. Логіку переходів реалізувати з використанням мови опису апаратних засобів.

1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODE** | **CUR\_STATE(2)** | **CUR\_STATE(1)** | **CUR\_STATE(0)** | **NEXT\_STATE(0)** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODE** | **CUR\_STATE(2)** | **CUR\_STATE(1)** | **CUR\_STATE(0)** | **NEXT\_STATE(0)** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODE** | **CUR\_STATE(2)** | **CUR\_STATE(1)** | **CUR\_STATE(0)** | **NEXT\_STATE(0)** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

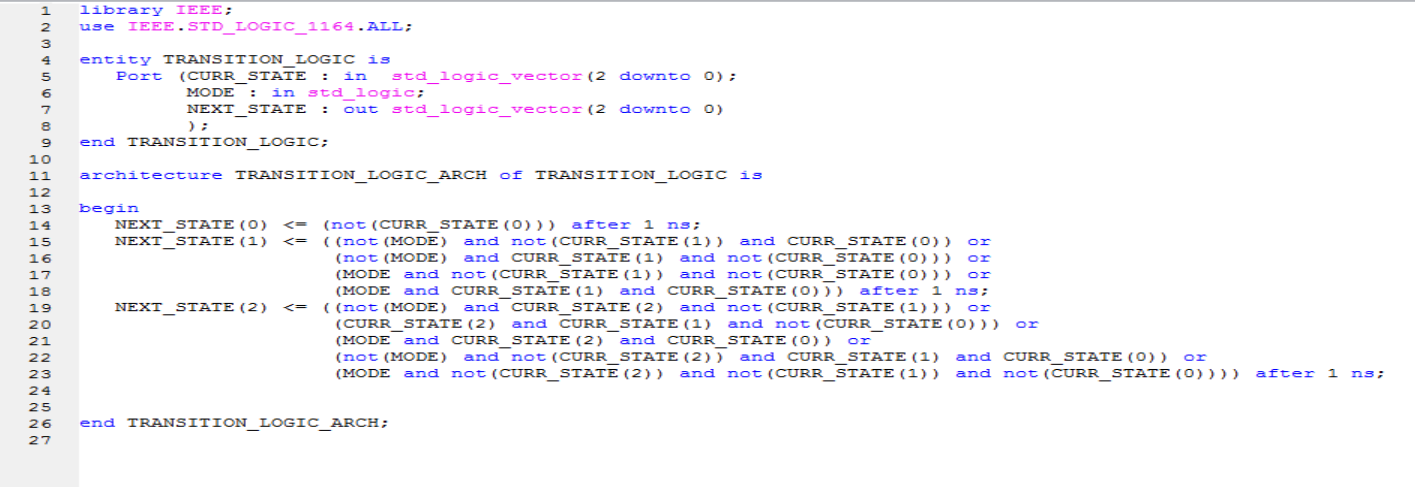
***Мінімізовані функції наступних станів автомата:***

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))));

*VHDL опис логіки переходів:*



**Граф переходів між станами:**



2) Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів VHDL.

***Логічні вирази для вихідних сигналів:***

OUT\_BUS(0) = ((not(IN\_BUS(2)) and not(IN\_BUS(1)) and not(IN\_BUS(0))) or TEST);

OUT\_BUS(1) = ((not(IN\_BUS(2)) and IN\_BUS(1) and not(IN\_BUS(0))) or TEST);

OUT\_BUS(2) = ((IN\_BUS(2) and not(IN\_BUS(1)) and not(IN\_BUS(0))) or TEST);

OUT\_BUS(3) = ((IN\_BUS(2) and IN\_BUS(1) and not(IN\_BUS(0))) or TEST);

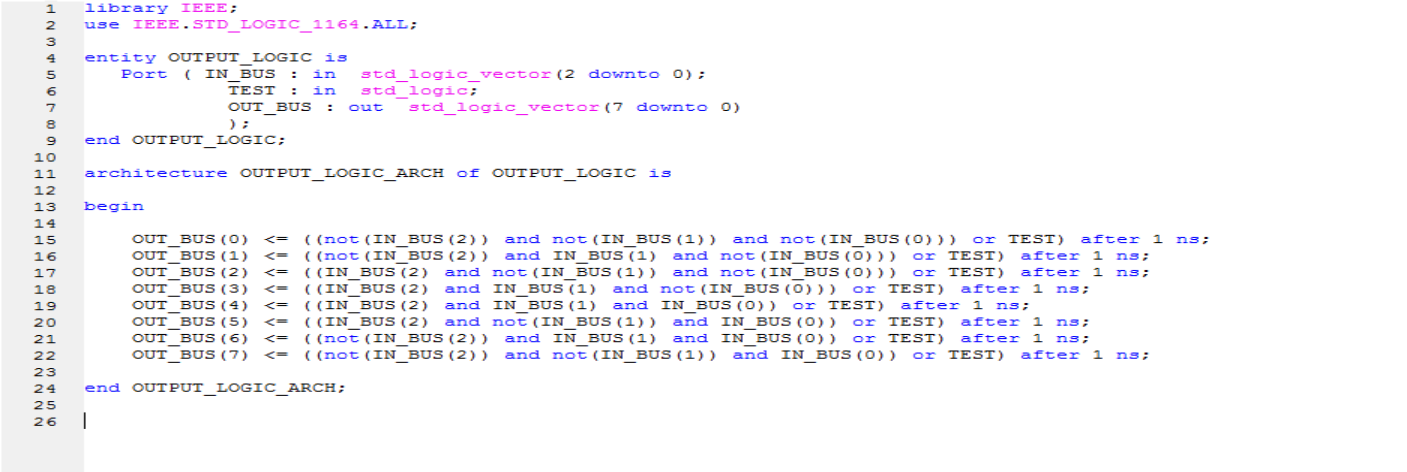
OUT\_BUS(4) = ((IN\_BUS(2) and IN\_BUS(1) and IN\_BUS(0)) or TEST);

OUT\_BUS(5) = ((IN\_BUS(2) and not(IN\_BUS(1)) and IN\_BUS(0)) or TEST);

OUT\_BUS(6) = ((not(IN\_BUS(2)) and IN\_BUS(1) and IN\_BUS(0)) or TEST);

OUT\_BUS(7) = ((not(IN\_BUS(2)) and not(IN\_BUS(1)) and IN\_BUS(0)) or TEST);

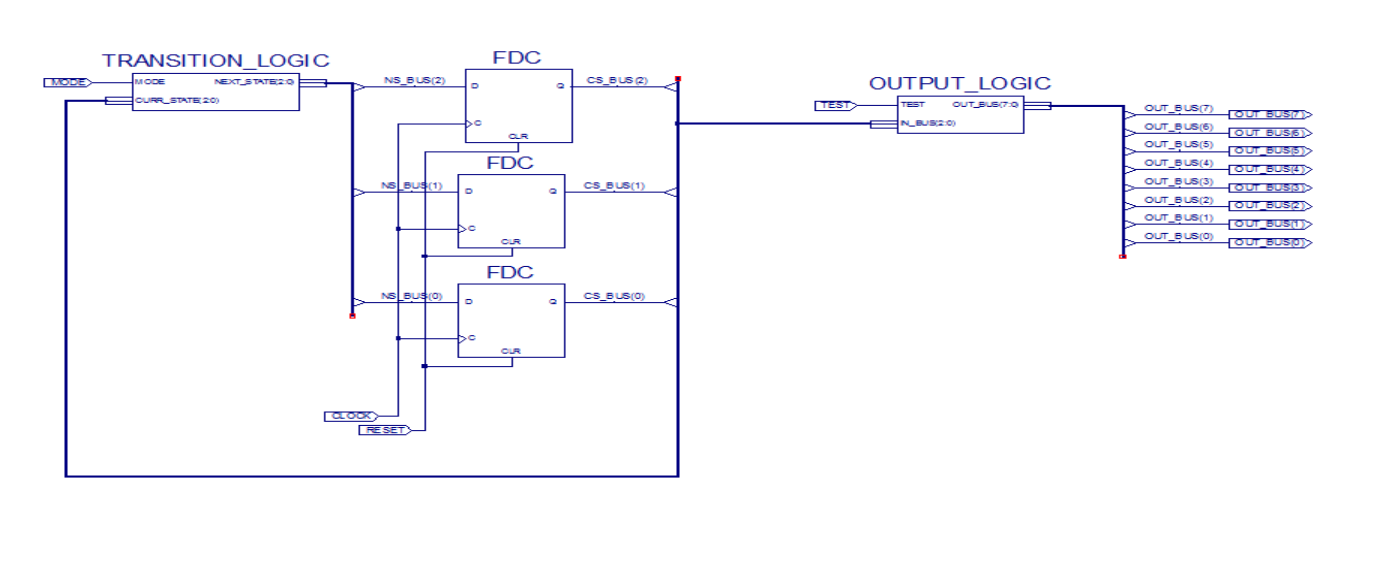
VHDL опис вихідних сигналів:



3) Згенерувати символи для описів логіки переходів та логіки формування вихідних сигналів.

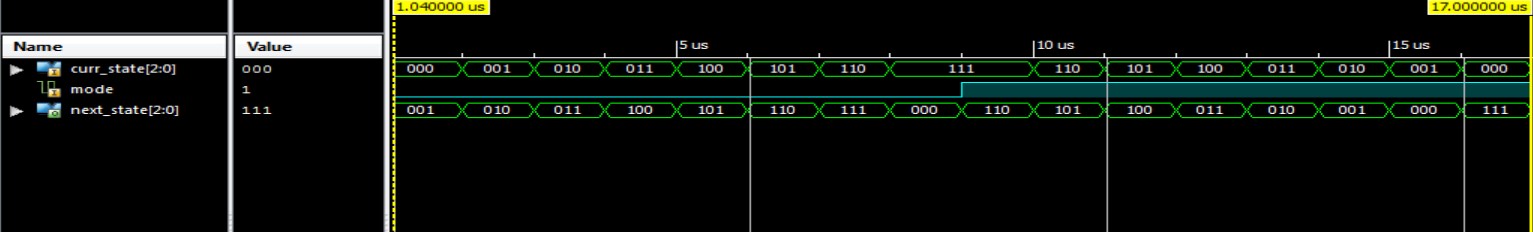
4) Зінтегрувати всі компоненти логіку переходів логіку формування вихідних сигналів та пам ять станів в єдину систему за допомогою ISE WebPACK Schematic Capture. Пам ять станів реалізувати за допомогою графічних компонентів з бібліотеки.

***Інтеграція всіх створених компонентів разом з пам’ятю стану автомата:***

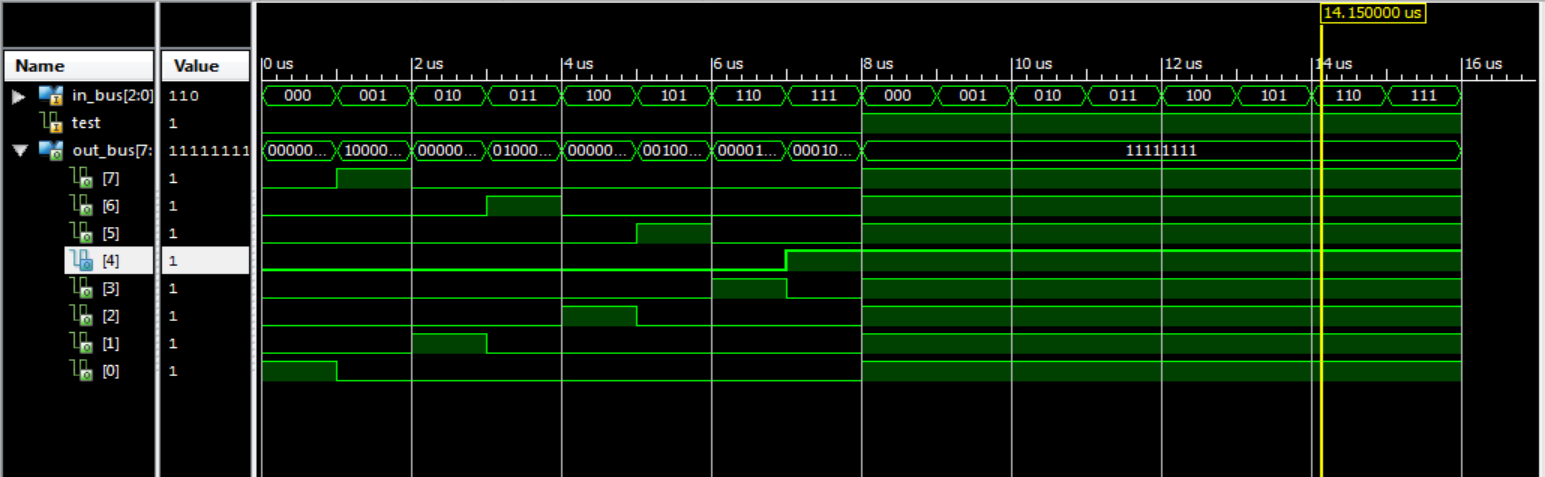


5) Промоделював роботу окремих частин автомата та автомата вцілому за допомогою симулятора ISim.

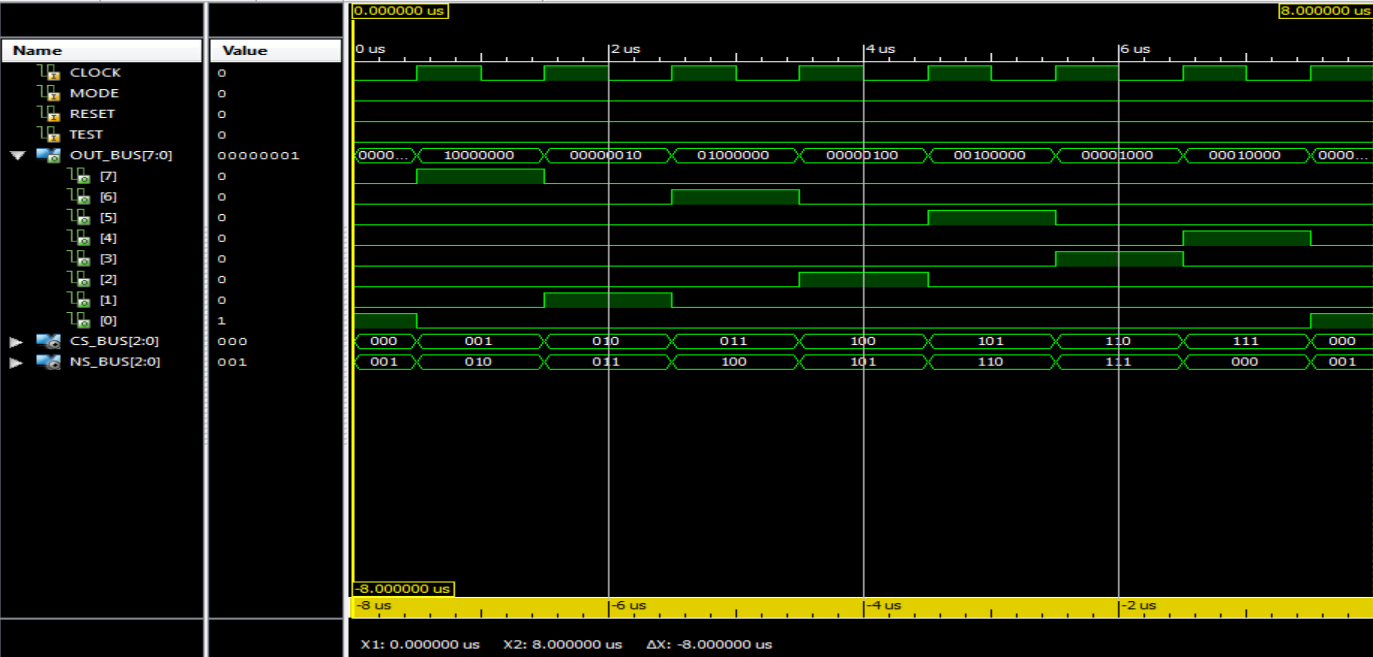
*Симуляція логіки переходів:*



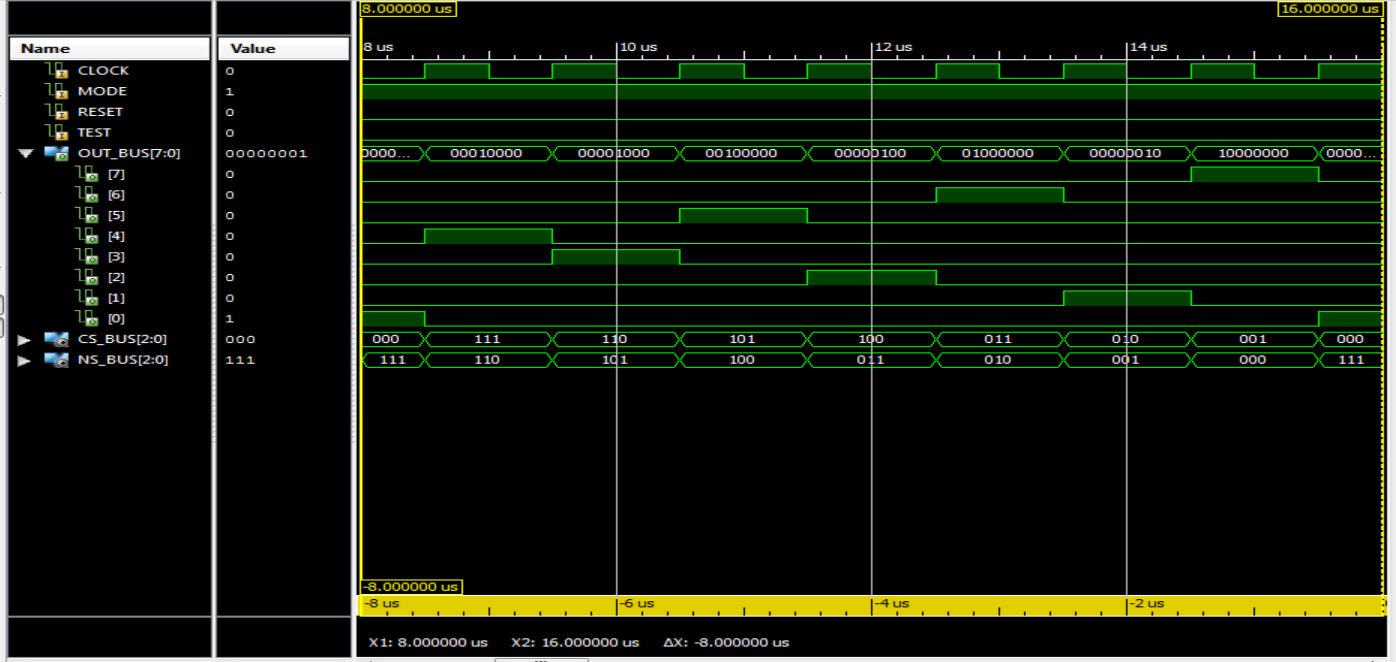
*Симуляція логіки вихідних сигналів(TEST = 0):*



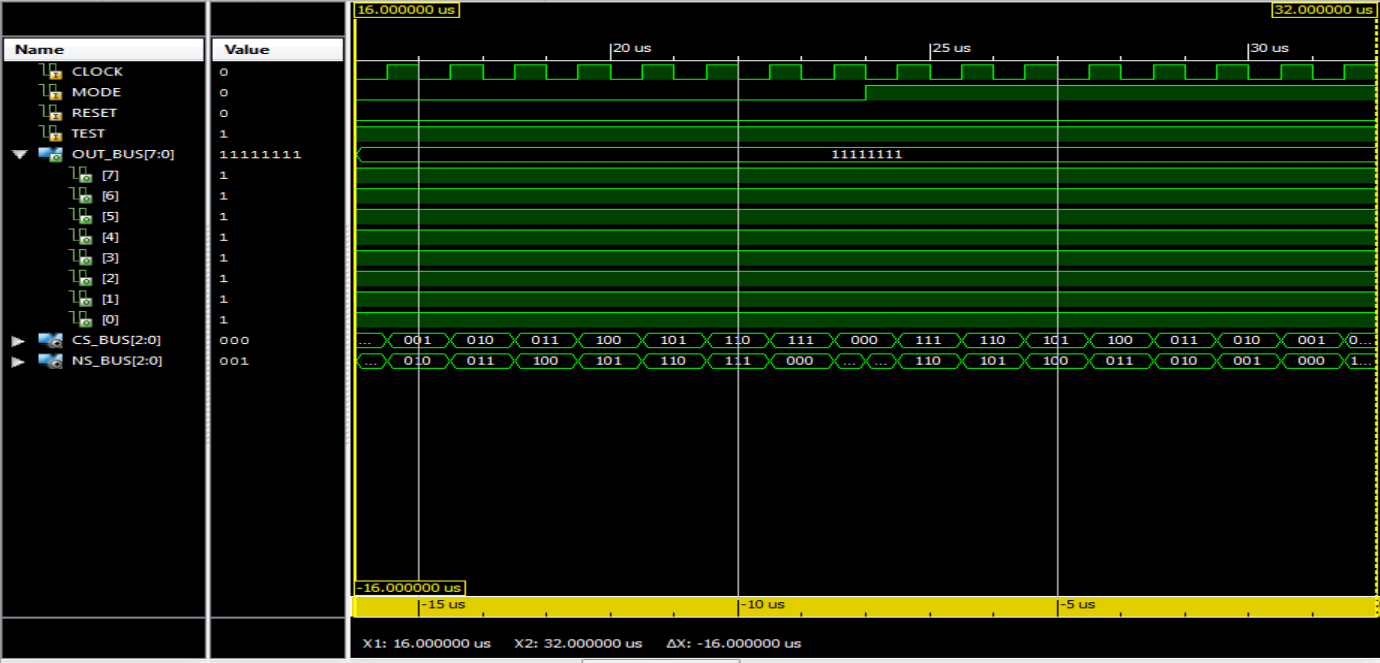
*Результати симуляції автомата(MODE = 0, TEST = 0, RESET = 0)*



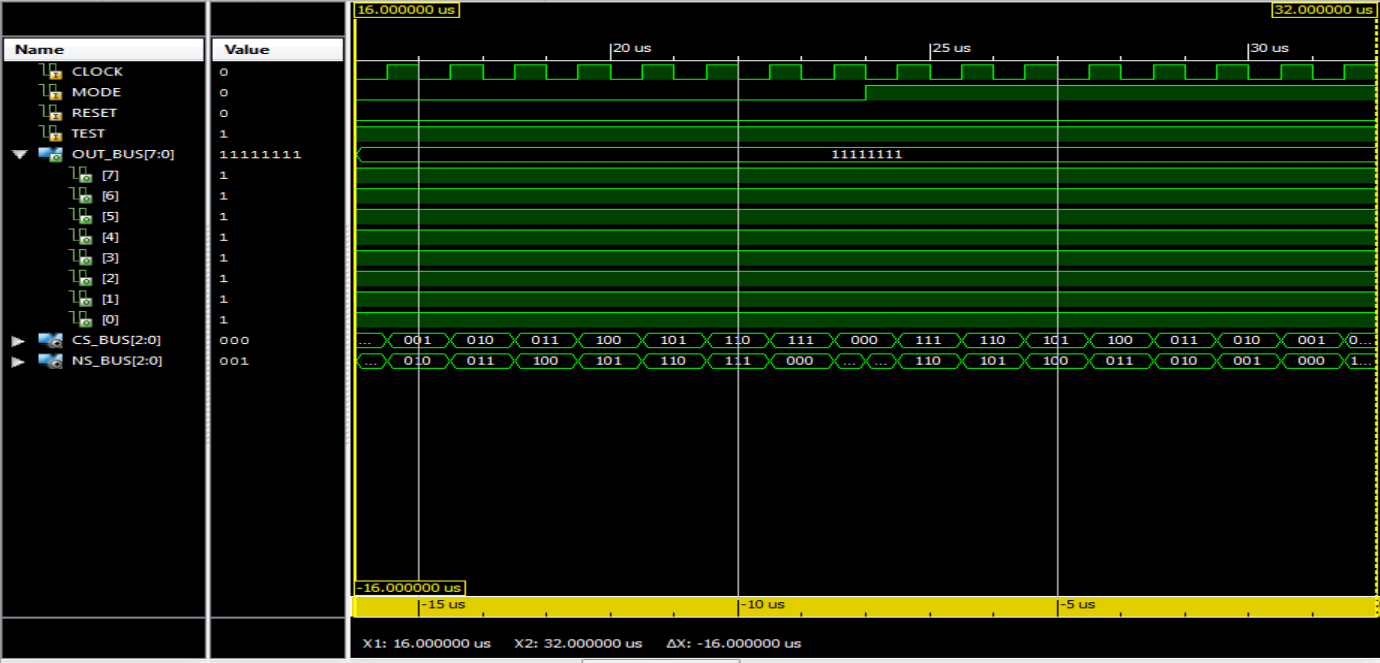
*Результати симуляції автомата(MODE = 1, TEST = 0, RESET = 0)*



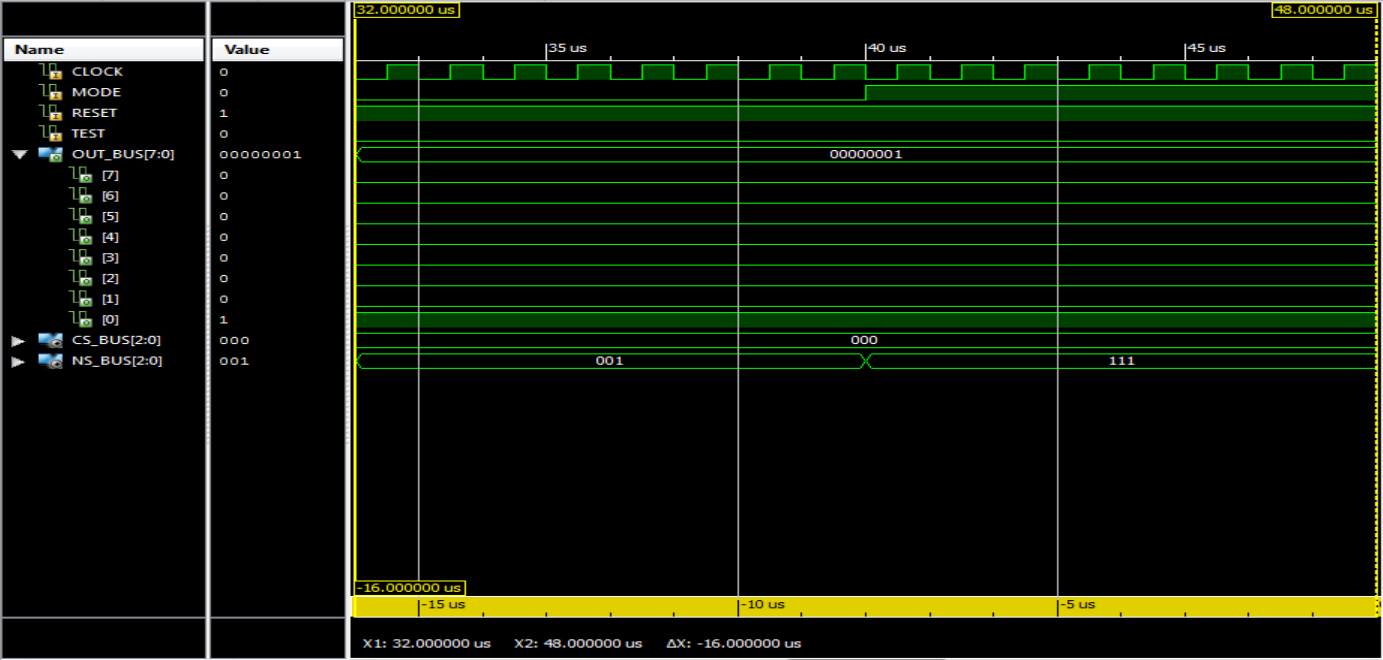
*Результати симуляції автомата(MODE = 0, TEST = 1, RESET = 0)*



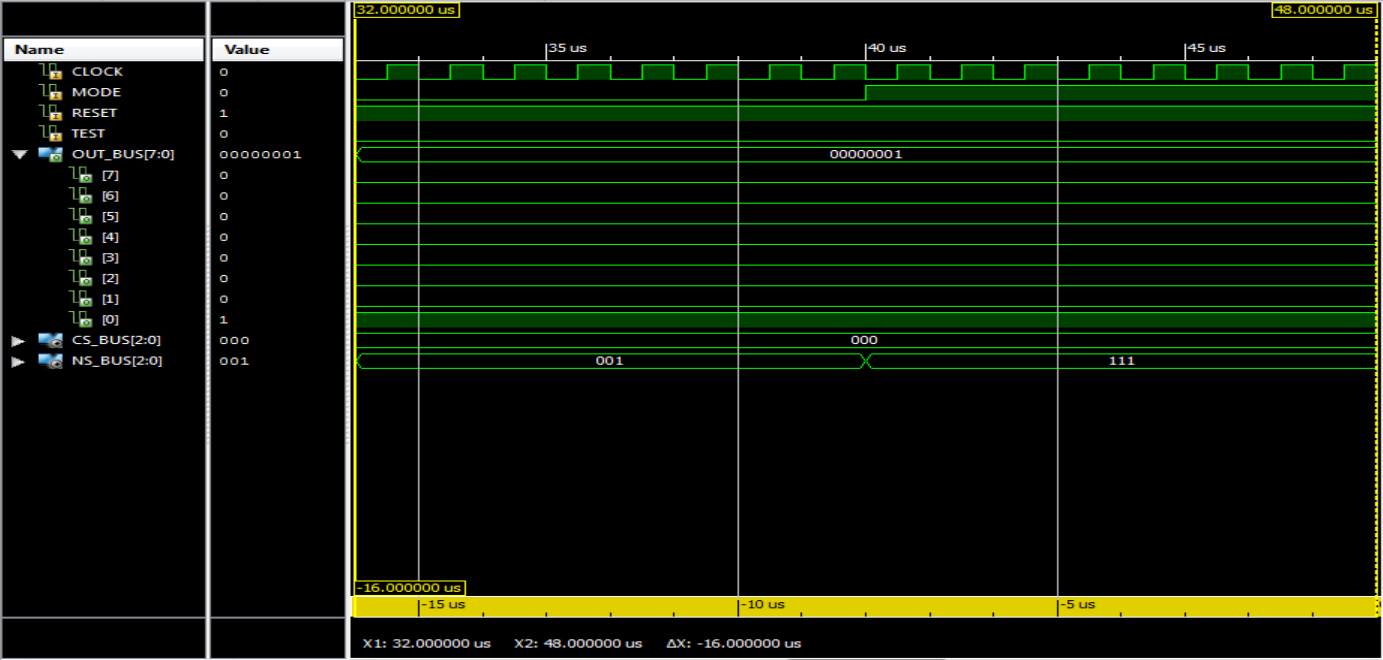
*Результати симуляції автомата(MODE = 1, TEST = 1, RESET = 0)*



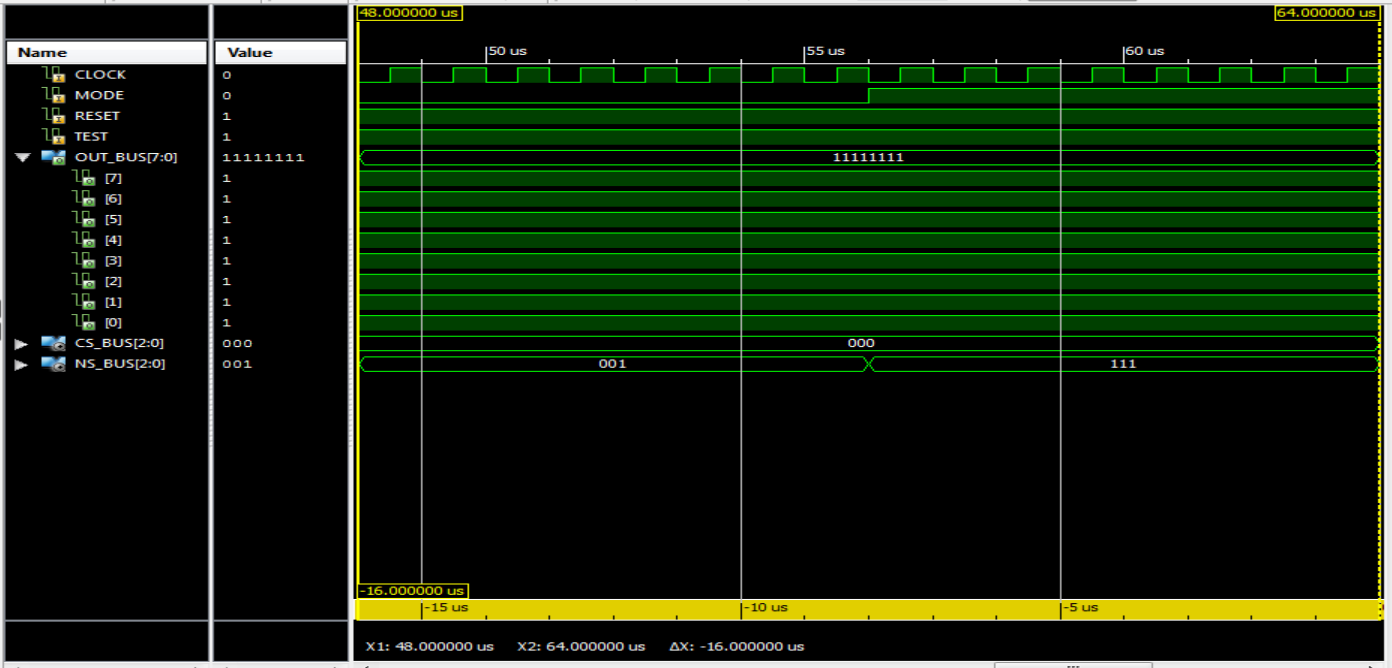
*Результати симуляції автомата(MODE = 0, TEST = 0, RESET = 1)*



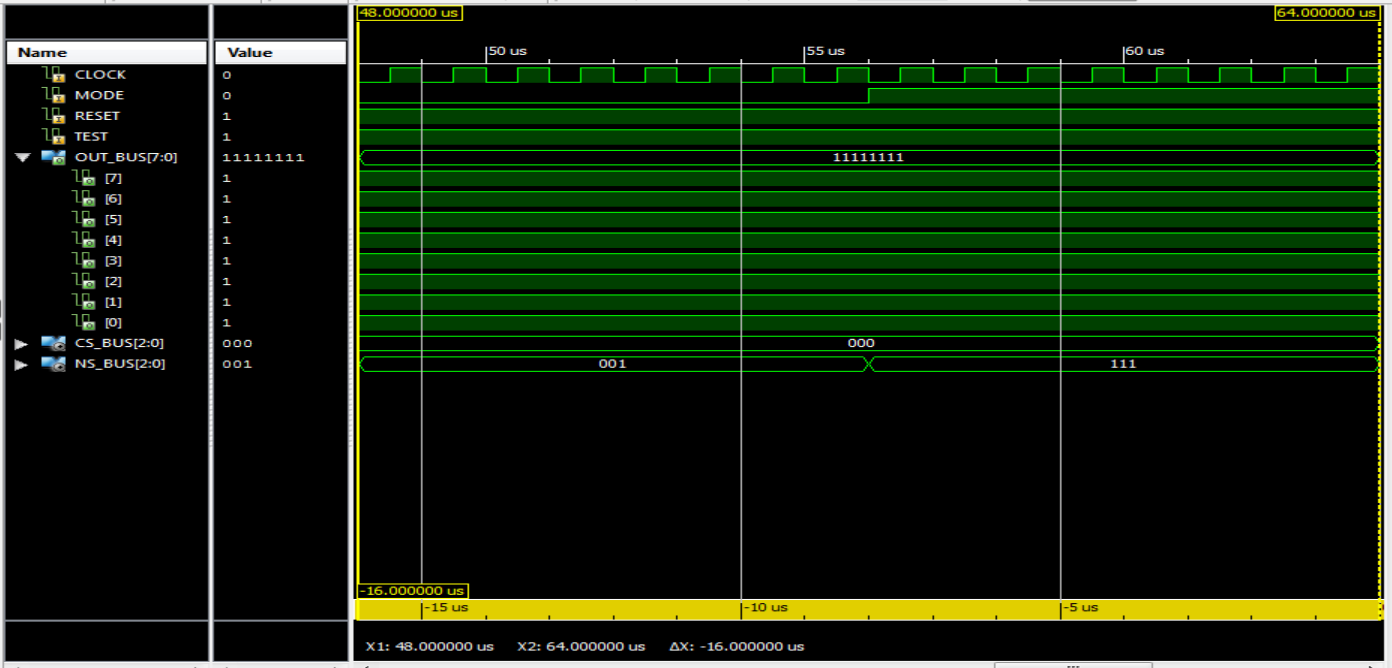
*Результати симуляції автомата(MODE = 1, TEST = 0, RESET = 1)*



*Результати симуляції автомата(MODE = 0, TEST = 1, RESET = 1)*



*Результати симуляції автомата(MODE = 1, TEST = 1, RESET = 1)*



***TEST BENCH:***

-- \*\*\* Test Bench - User Defined Section \*\*\*

tb : PROCESS

BEGIN

MODE <= '1';

TEST <= '1';

RESET <= '1', '0' after 200 ms;

wait until RESET = '0';

assert OUTPUT = "11111111";

wait for 2786028us;

TEST <= '0';

RESET <= '1', '0' after 200 ms;

wait until RESET = '0';

assert OUTPUT = "00000001";

wait for 174864us;

assert OUTPUT = "00010000";

wait for 349625us;

assert OUTPUT = "00001000";

wait for 349625us;

assert OUTPUT = "00100000";

wait for 349625us;

assert OUTPUT = "00000100";

wait for 349625us;

assert OUTPUT = "01000000";

wait for 349625us;

assert OUTPUT = "00000010";

wait for 349625us;

assert OUTPUT = "10000000";

wait for 349625us;

assert OUTPUT = "00000001";

wait for 349625us;

TEST <= '1';

MODE <= '0';

RESET <= '1', '0' after 200 ms;

wait until RESET = '0';

assert OUTPUT = "11111111";

wait for 2786028us;

TEST <= '0';

RESET <= '1', '0' after 200 ms;

wait until RESET = '0';

assert OUTPUT = "00000001";

wait for 174864us;

assert OUTPUT = "10000000";

wait for 349625us;

assert OUTPUT = "00000010";

wait for 349625us;

assert OUTPUT = "01000000";

wait for 349625us;

assert OUTPUT = "00000100";

wait for 349625us;

assert OUTPUT = "00100000";

wait for 349625us;

assert OUTPUT = "00001000";

wait for 349625us;

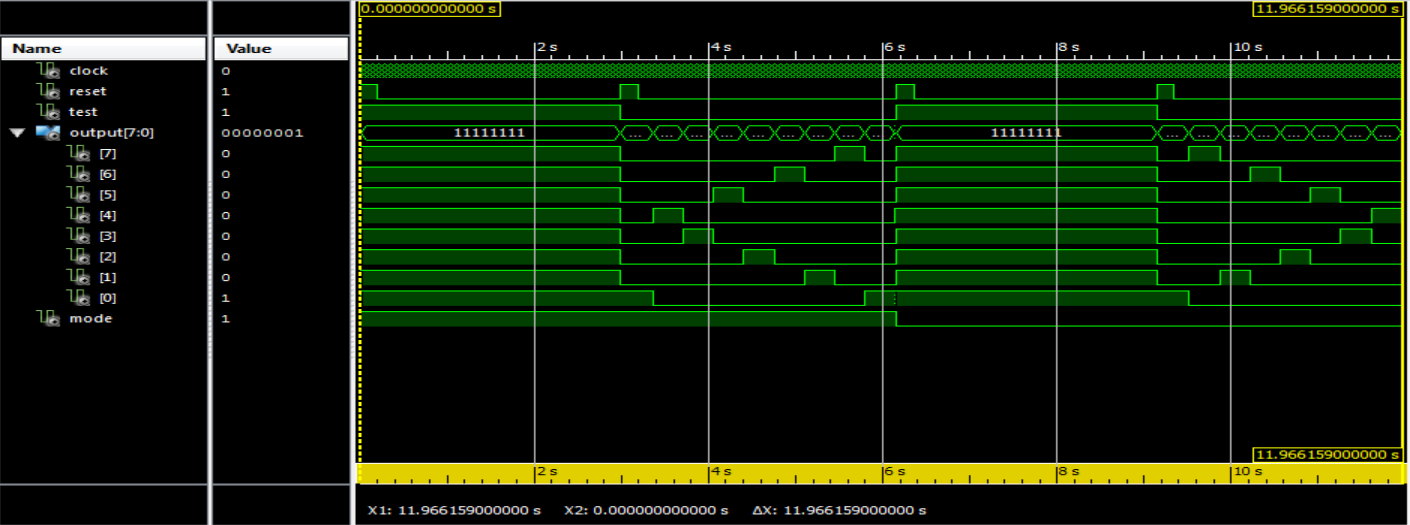
assert OUTPUT = "00010000";

wait for 349625us;

END PROCESS;

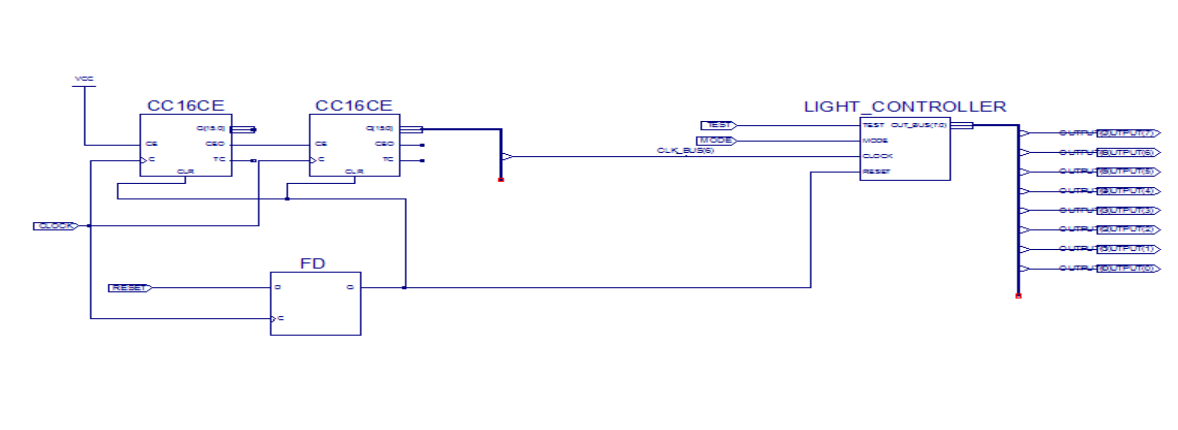
-- \*\*\* End Test Bench - User Defined Section \*\*\*

*Результати TEST BENCH:*

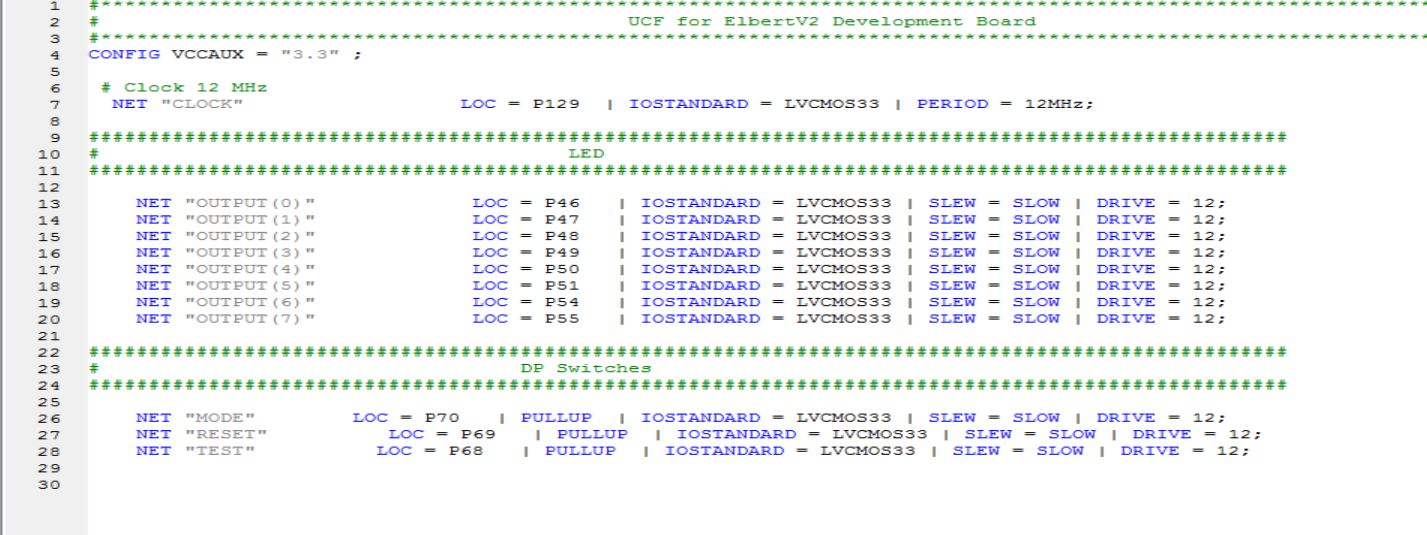


6) Інтегрувати створений автомат зі стендом Elbert V2 – Spartan3A FPGA. Додати подільник частоти для вхідного тактовового сигналу призначити фізичні виводи на FPGA.

*Схема автомата світлових сигналів та подільник тактового сигналу:*



*Призначення фізичних входів та виходів:*



**Висновок:** виконуючи дану лабораторну роботу я реалізував цифровий автомат світлових ефектів на базі стенда Elbert V2 – Spartan3A FPGA згідно заданих вимог.