

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Кафедра ЕОМ
З лабораторної роботи № 1
З дисципліни « МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ »
На тему: « Структурний опис цифрового автомата. Перевірка роботи автомата за
допомогою стенда Elbert V2-Spartan 3A FPGA»
Варіант 18

Виконав: студент гр. КІ-201
Панасевич Ю. В.

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
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів VHDL. Заборонено використовувати оператори if, switch, for, when
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів VHDL. Заборонено використовувати оператори if, switch, for, when
4. Згенерувати Schematic символи для VHDL 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 за допомогою ISE WebPACK Schematic Capture. 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 Elbert V2 – Spartan 3A FPGA (додати подільник частоти для вхідного тактового сигналу призначити фізичні виводи на FPGA)
8. Згенерувати BIT файл та перевірити роботу за допомогою стенда Elbert V2 – Spartan 3A FPGA
9. Підготувати і захистити звіт

Завдання:

Мій номер по списку 18, тому номер варіант 6.

- Пристрій повинен реалізувати 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 тактовий сигнал від мікроконтролера і знижувати частоту за допомогою внутрішнього подільника Мікроконтролер є частиною стенда 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 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 для подачі логічної 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.
- Для керування сигналами RESET/TEST використати будь які з PUSH BUTTON кнопок.

Виконання роботи:

(VHDL OutputLogic)

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity out_logic_intf is
```

```
    Port( IN_BUS : in std_logic_vector(2 downto 0);
```

```
          OUT_BUS : out std_logic_vector(7 downto 0)
```

```
          );
```

```
end out_logic_intf;
```

```
architecture out_logic_arch of out_logic_intf is
```

```
begin
```

```
    OUT_BUS(0) <= not IN_BUS(2);
```

```
    OUT_BUS(1) <= not IN_BUS(2) and (IN_BUS(1) or IN_BUS(0));
```

```
    OUT_BUS(2) <= not IN_BUS(2) and IN_BUS(1);
```

```
    OUT_BUS(3) <= not IN_BUS(2) and IN_BUS(1) and IN_BUS(0);
```

```
    OUT_BUS(4) <= IN_BUS(2);
```

```
    OUT_BUS(5) <= IN_BUS(2) and (IN_BUS(1) or IN_BUS(0));
```

```
    OUT_BUS(6) <= IN_BUS(2) and IN_BUS(1);
```

```
    OUT_BUS(7) <= IN_BUS(2) and IN_BUS(1) and IN_BUS(0);
```

```
end out_logic_arch;
```

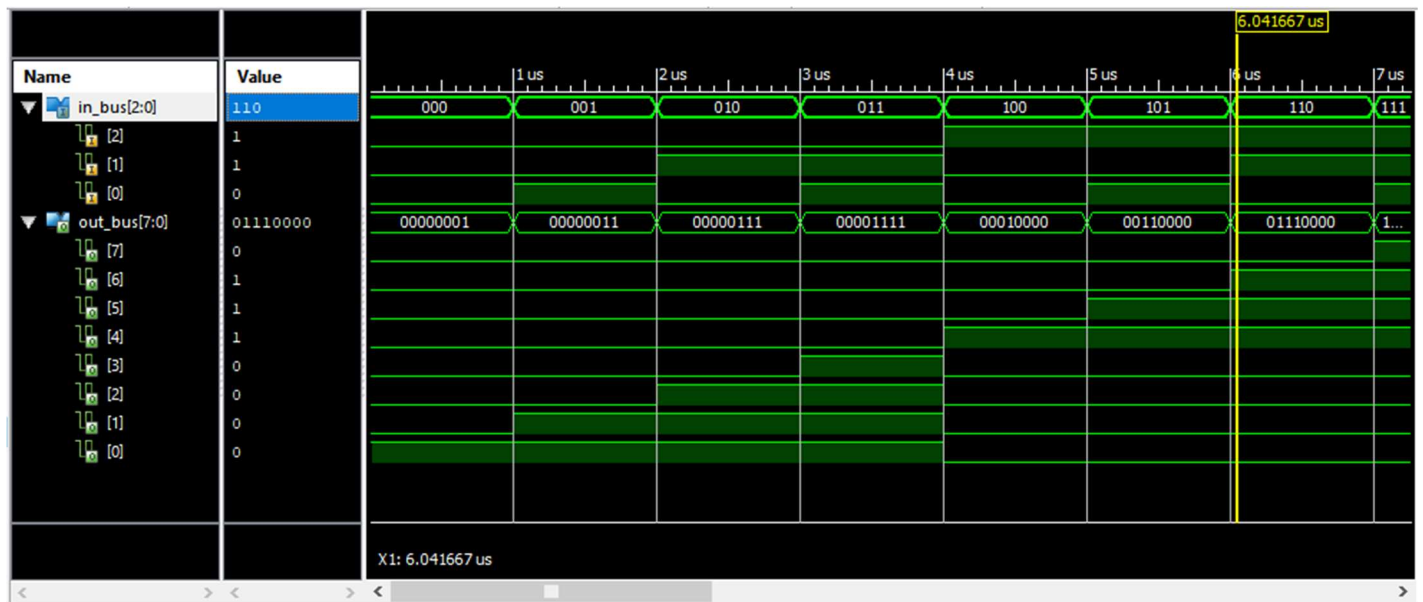


Рис.1: виконання роботи схеми
(VHDL TransitionLogic)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity transition_logic_intf is
    Port( CUR_STATE : in std_logic_vector (2 downto 0);
          MODE : in std_logic;
          NEXT_STATE : out std_logic_vector (2 downto 0)
    );
end transition_logic_intf;

architecture transition_logic_arch of transition_logic_intf is

begin

    NEXT_STATE(0) <= (not(MODE) and not(CUR_STATE(1)) and not
(CUR_STATE(0))) or
(not(CUR_STATE(0))) or
(MODE and not(CUR_STATE(1)) and not
(CUR_STATE(0))) or
(MODE and CUR_STATE(1) and not
(CUR_STATE(0)));

    NEXT_STATE(1) <= (not(MODE) and not(CUR_STATE(1)) and CUR_STATE(0))
or
(not(MODE) and CUR_STATE(1) and not
(CUR_STATE(0))) or
(MODE and not(CUR_STATE(1)) and
not(CUR_STATE(0))) or
(MODE and CUR_STATE(1) and
CUR_STATE(0));

```

```

NEXT_STATE(2) <= ((not(MODE) and CUR_STATE(2)) and not (CUR_STATE(1)
and CUR_STATE(0))) or
((not(MODE) and not (CUR_STATE(2))) and
(CUR_STATE(1) and CUR_STATE(0))) or
((MODE and CUR_STATE(2)) and
(CUR_STATE(1) or CUR_STATE(0))) or
(MODE and not CUR_STATE(2) and not
CUR_STATE(1) and not CUR_STATE(0));
end transition_logic_arch;

```

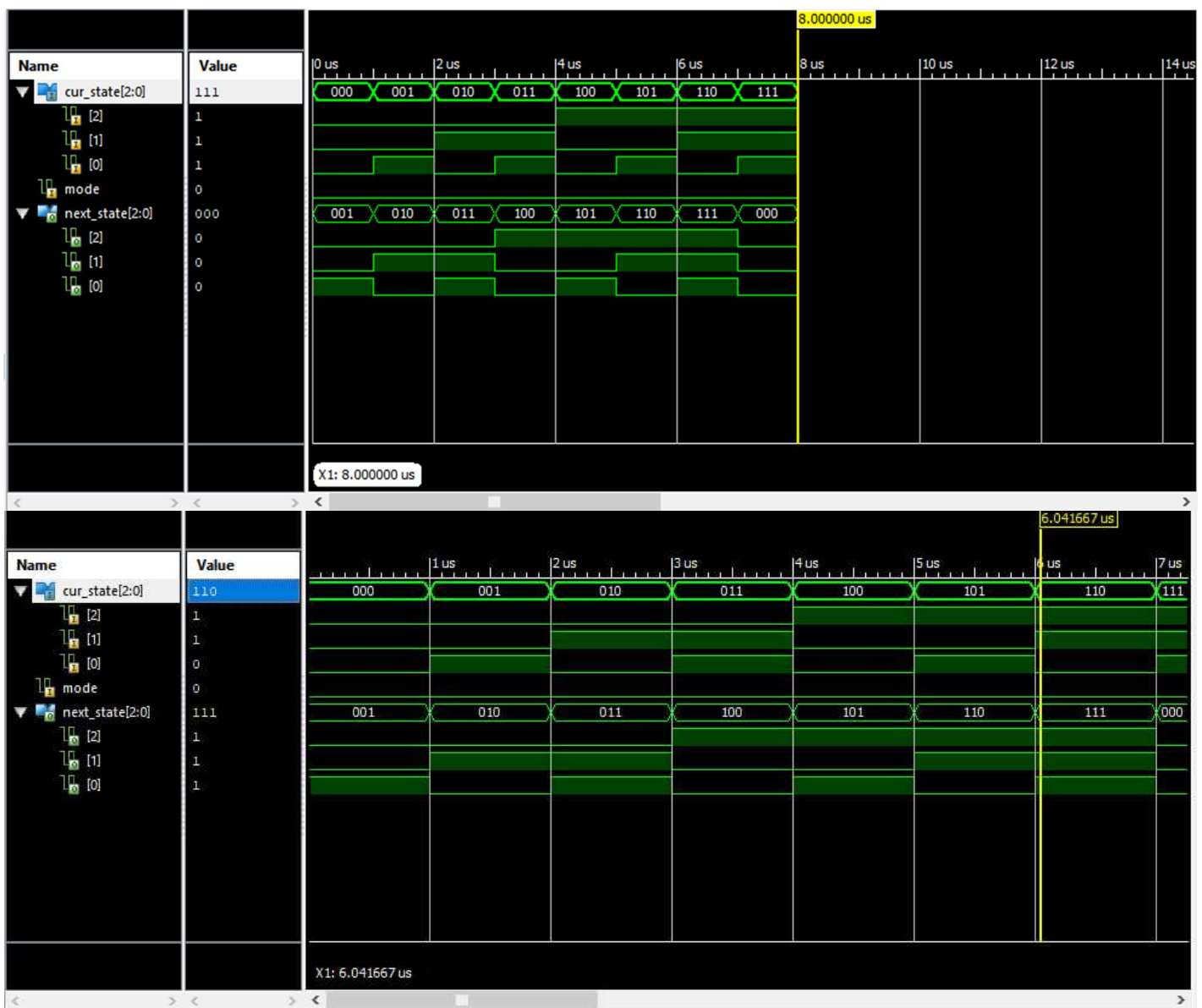


Рис.2: промодельована робота схеми формування вихідних сигналів з усіма можливими наборами сигналів

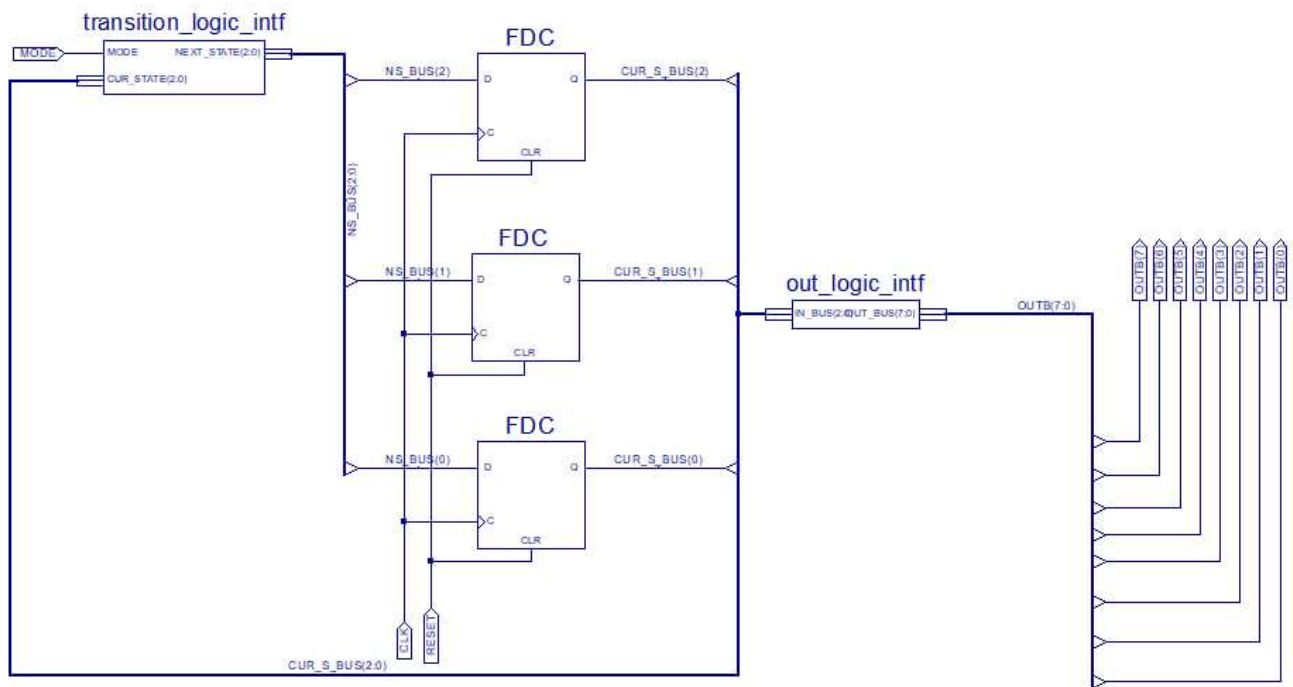


Рис.3: Schematic файл (LightController.sch)

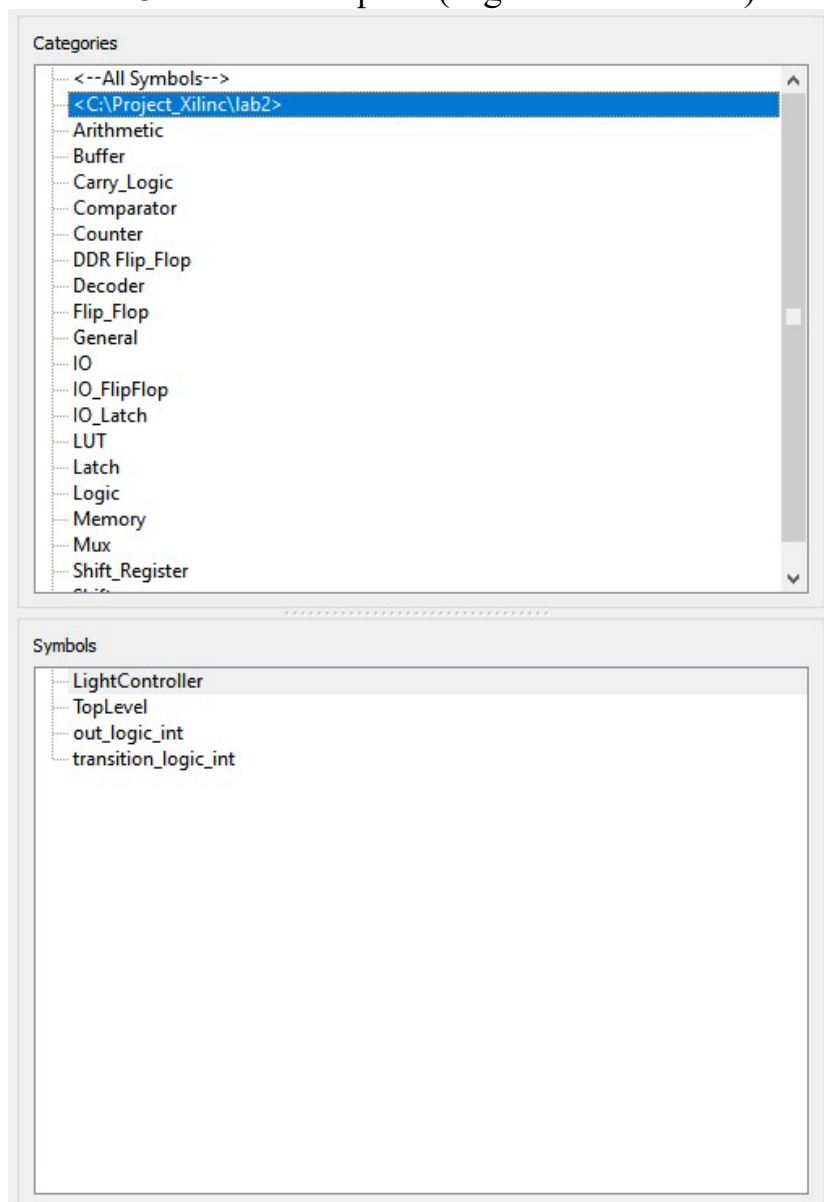


Рис.4: згенеровані Schematic симболи

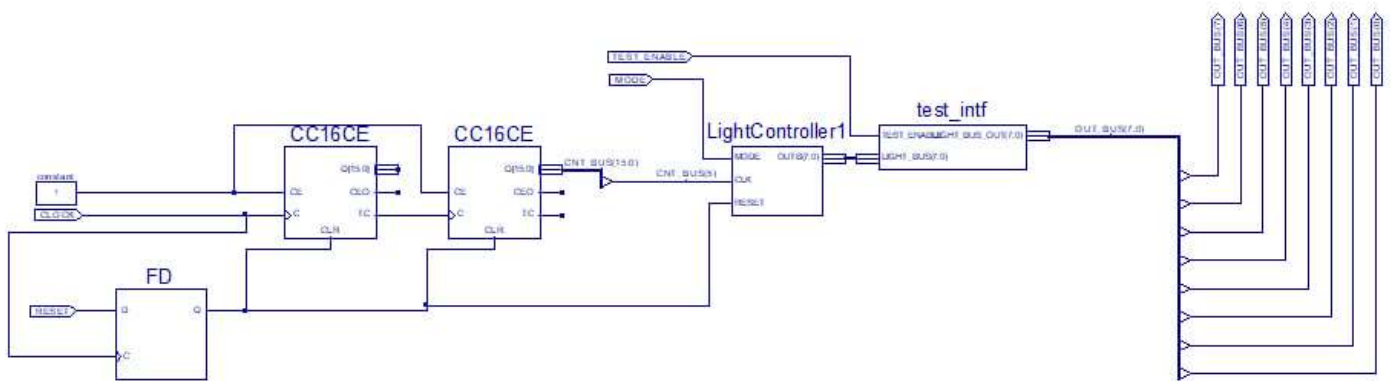


Рис.5: Schematic файл для кінцевої схеми.(TopLevel.sch)

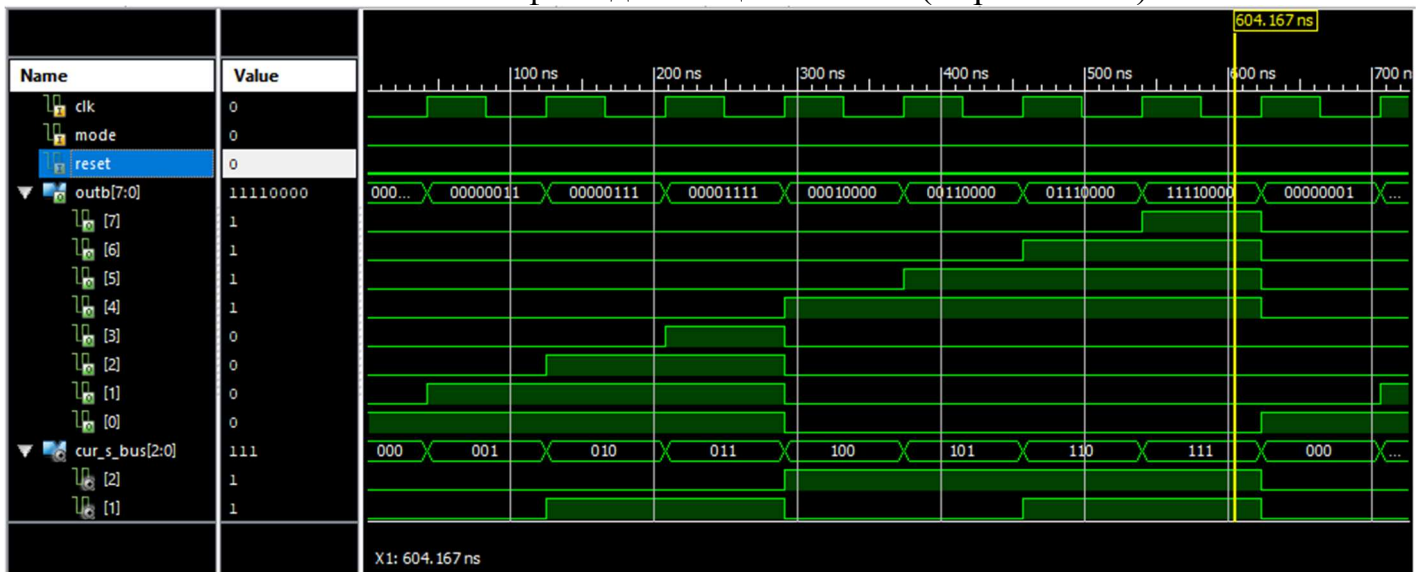


Рис.6: симуляція системи різними значеннями сигналів MODE/RESET/SPEED



Рис.7: згенерований бінарний файл

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