

Міністерство освіти і науки України
Національний університет „Львівська політехніка”



Звіт з лабораторної роботи № 2
З дисципліни:
“Моделювання комп’ютерних систем”

Виконав:
студент групи КІ-201
Бенітез Гектор

Прийняв:
Козак Н.Б.

Львів – 2023

Тема роботи:

Структурний опис цифрового автомата. Перевірка роботи автомата за допомогою стенда Elbert V2 – Spartan 3A FPGA

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. Промодельовувати 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

варіант – 4.

- 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 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

- 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 (TEST) для подачі логічної «1» на всі виходи одночасно:
 - Якщо $TEST=0$ то автомат перемикає сигнали на виходах згідно заданого алгоритму.
 - Якщо $TEST=1$ то на всіх виходах повинна бути логічна «1» (всі LED увімкнені).
- Для керування сигналом MODE використати будь який з 8 DIP перемикачів (див. **Додаток – 1**).
- Для керування сигналами RESET/TEST використати будь які з PUSH BUTTON кнопок (див. **Додаток – 1**).

Виконання роботи:

1. Створення файлу VHDL , який реалізовує логіку формування сигналів

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity output_logic_intf is
    Port (
        IN_BUS : in  std_logic_vector(2 downto 0);
        CHECK : in  std_logic;
        OUT_BUS : out std_logic_vector(7 downto 0)
    );
end output_logic_intf;

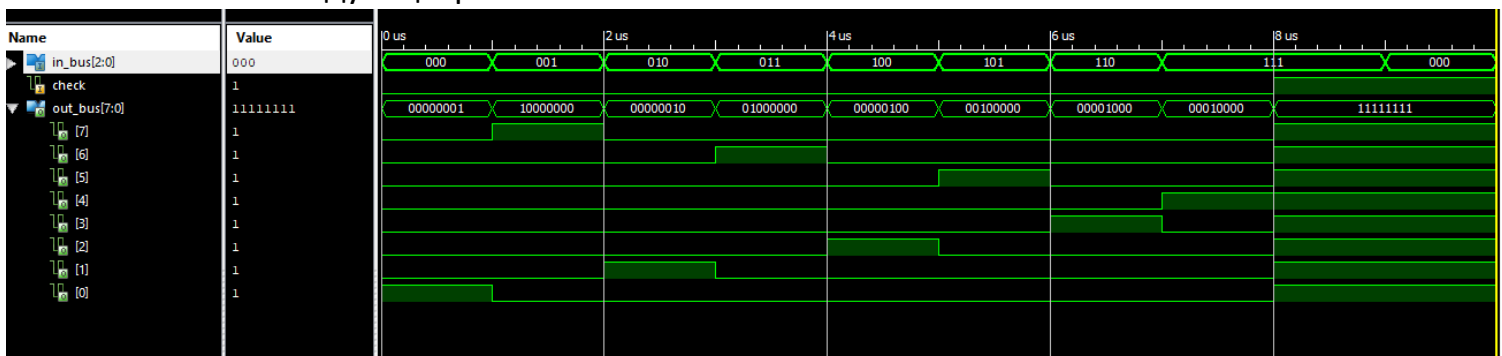
architecture output_logic_arch of output_logic_intf is

begin

    OUT_BUS(0) <= (not(IN_BUS(0)) and not(IN_BUS(1)) and not(IN_BUS(2))) or CHECK;
    OUT_BUS(7) <= (IN_BUS(0) and not(IN_BUS(1)) and not(IN_BUS(2))) or CHECK;
    OUT_BUS(1) <= (not(IN_BUS(0)) and IN_BUS(1) and not(IN_BUS(2))) or CHECK;
    OUT_BUS(6) <= (IN_BUS(0) and IN_BUS(1) and not(IN_BUS(2))) or CHECK;
    OUT_BUS(2) <= (not(IN_BUS(0)) and not(IN_BUS(1)) and IN_BUS(2)) or CHECK;
    OUT_BUS(5) <= (IN_BUS(0) and not(IN_BUS(1)) and IN_BUS(2)) or CHECK;
    OUT_BUS(3) <= (not(IN_BUS(0)) and IN_BUS(1) and IN_BUS(2)) or CHECK;
    OUT_BUS(4) <= (IN_BUS(0) and IN_BUS(1) and IN_BUS(2)) or CHECK;

end output_logic_arch;
```

2. Модуляція роботи



3. Створення файлу VHDL , який реалізовує логіку формування переходів стану автомата

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity transition_logic_intf is
Port (      CUR_STATE : in  std_logic_vector(2 downto 0) := "000";
          MODE : in  std_logic := '0';
          RESET : in  std_logic := '0';
          NEXT_STATE : out std_logic_vector(2 downto 0)
        );
end transition_logic_intf;

architecture transition_logic_arch of transition_logic_intf is
begin

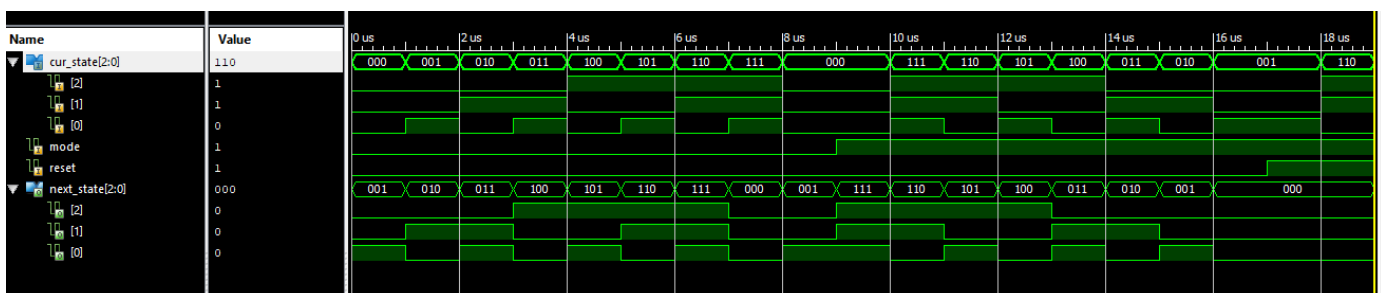
    NEXT_STATE(0) <= not(CUR_STATE(0)) and not(RESET);

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

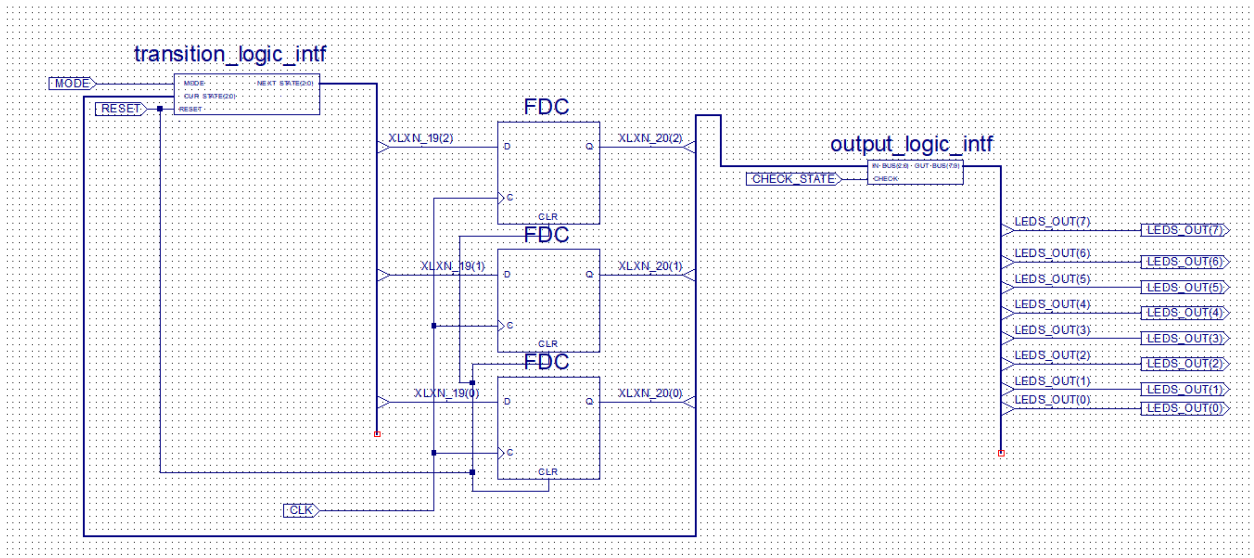
    NEXT_STATE(2) <= ((not(MODE) and not(CUR_STATE(2)) and CUR_STATE(1) and CUR_STATE(0)) or
                      (not(MODE) and CUR_STATE(2) and not(CUR_STATE(1))) or
                      (CUR_STATE(2) and CUR_STATE(1) and not(CUR_STATE(0))) or
                      (MODE and not(CUR_STATE(2)) and not(CUR_STATE(1))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

end transition_logic_arch;
```

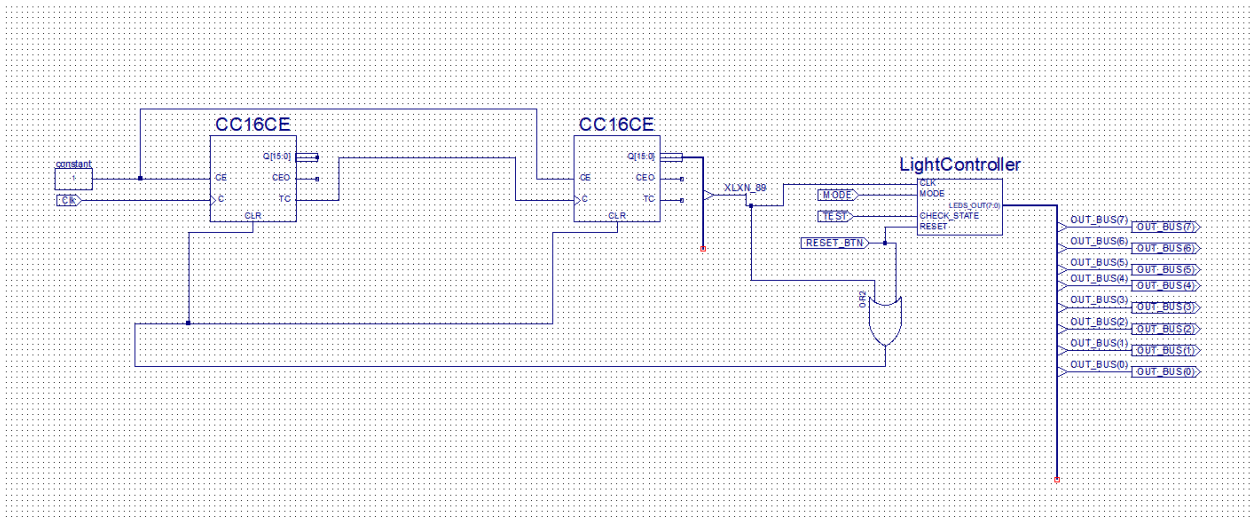
4. Модуляція роботи



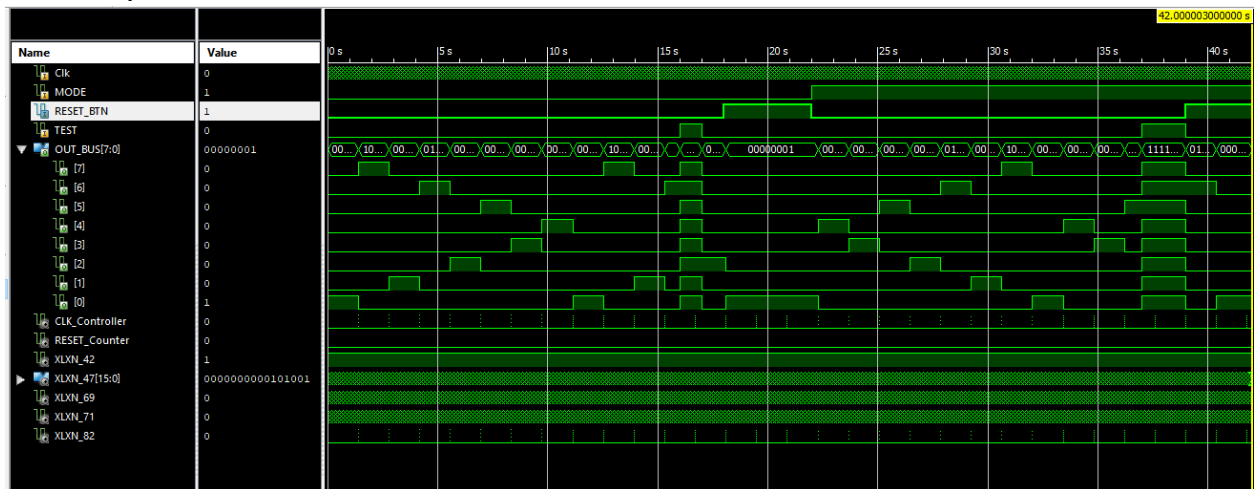
5. Створення схеми з реалізацією пам'яті стану



6. Створення файлу для кінцевої схеми та реалізація подільника частоти.



7. Модуляція кінцевої схеми.



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