

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Кафедра ЕОМ
З лабораторної роботи № 2
З дисципліни « МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ »
На тему: « Структурний опис цифрового автомата. Перевірка роботи автомата за
допомогою стенда Elbert V2-Spartan 3A FPGA»
Варіант 8

Виконала: ст. гр. КІ-201
Домашовець С. І.

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 20, тому номер варіант 2.

Варіант – 2:

- 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	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**).

Виконання роботи:

(VHDL OutputLogic)

```
-----  
-- Company:  
-- Engineer:  
--  
-- Create Date: 17:41:48 03/26/2021  
-- Design Name:  
-- Module Name: out_logic_intf_2v - out_logic_arch_2v  
-- Project Name:  
-- Target Devices:  
-- Tool versions:  
-- Description:  
--  
-- Dependencies:  
--  
-- Revision:  
-- Revision 0.01 - File Created  
-- Additional Comments:  
--  
-----
```

```
library IEEE;
```

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

```
entity out_logic_intf_2v is
```

```
port ( in_bus : in std_logic_vector(2 downto 0) ;
```

```
      out_bus : out std_logic_vector(7 downto 0));
```

```
end out_logic_intf_2v;
```

```
architecture out_logic_arch_2v of out_logic_intf_2v is
```

```
signal A_OUT, B_OUT, C_OUT, D_OUT,
```

```
      E_OUT, F_OUT, G_OUT, H_OUT : std_logic;
```

```
begin
```

```
    A_OUT <= (not(in_bus(2)) and not(in_bus(1)) and not(in_bus(0)));
```

```
    B_OUT <= (not(in_bus(2)) and not(in_bus(1)) and in_bus(0) );
```

```
    C_OUT <= (not(in_bus(2)) and in_bus(1) and not(in_bus(0)));
```

```
    D_OUT <= (not(in_bus(2)) and in_bus(1) and in_bus(0) );
```

```
    E_OUT <= ( in_bus(2) and not(in_bus(1)) and not(in_bus(0)));
```

```

F_OUT <= ( in_bus(2) and not(in_bus(1)) and in_bus(0) );
G_OUT <= ( in_bus(2) and in_bus(1) and not(in_bus(0)));
H_OUT <= ( in_bus(2) and in_bus(1) and in_bus(0) );

```

```

out_bus(0) <= A_OUT or B_OUT or C_OUT or D_OUT or E_OUT or F_OUT or
G_OUT or H_OUT;

```

```

out_bus(1) <= B_OUT or C_OUT or D_OUT or E_OUT or F_OUT or G_OUT or
H_OUT;

```

```

out_bus(2) <= C_OUT or D_OUT or E_OUT or F_OUT or G_OUT or H_OUT;

```

```

out_bus(3) <= D_OUT or E_OUT or F_OUT or G_OUT or H_OUT;

```

```

out_bus(4) <= E_OUT or F_OUT or G_OUT or H_OUT;

```

```

out_bus(5) <= F_OUT or G_OUT or H_OUT;

```

```

out_bus(6) <= G_OUT or H_OUT;

```

```

out_bus(7) <= H_OUT;

```

```

end out_logic_arch_2v;

```

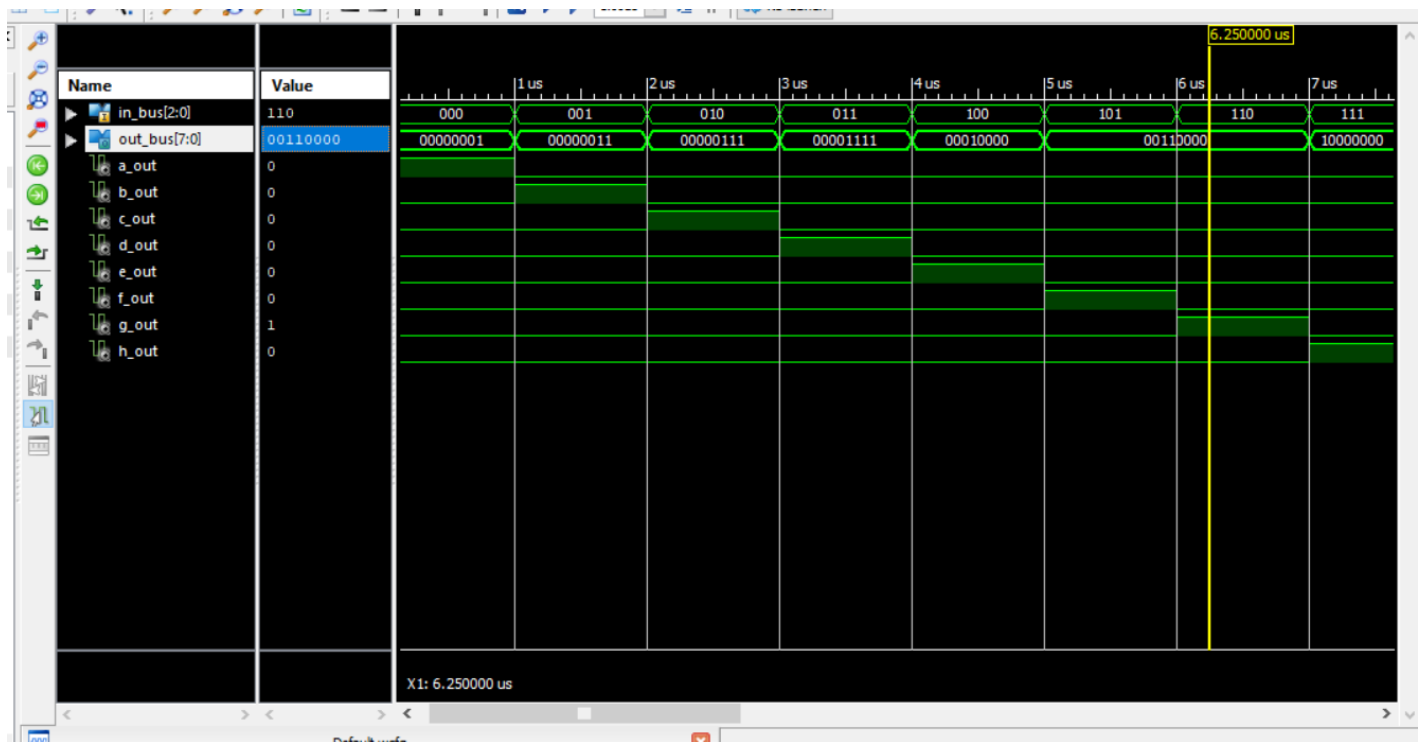


Рис.1: виконання роботи схеми

(VHDL TransitionLogic)

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

entity transition_logic_intf_2v 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_2v;

architecture transition_logic_arch_2v of transition_logic_intf_2v is

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

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

```

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)) and
not(CUR_STATE(0))) or
(not(MODE) and CUR_STATE(2) and not(CUR_STATE(1)) and
CUR_STATE(0)) or
(not(MODE) and CUR_STATE(2) and CUR_STATE(1) and
not(CUR_STATE(0))) or
( MODE and CUR_STATE(2) and not(CUR_STATE(1)) and
CUR_STATE(0)) or
( MODE and CUR_STATE(2) and CUR_STATE(1) and
not(CUR_STATE(0))) or
( MODE and CUR_STATE(2) and CUR_STATE(1) and
CUR_STATE(0)) or
( MODE and not(CUR_STATE(2)) and not(CUR_STATE(1)) and
not(CUR_STATE(0))) ;

end transition_logic_arch_2v;

```



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

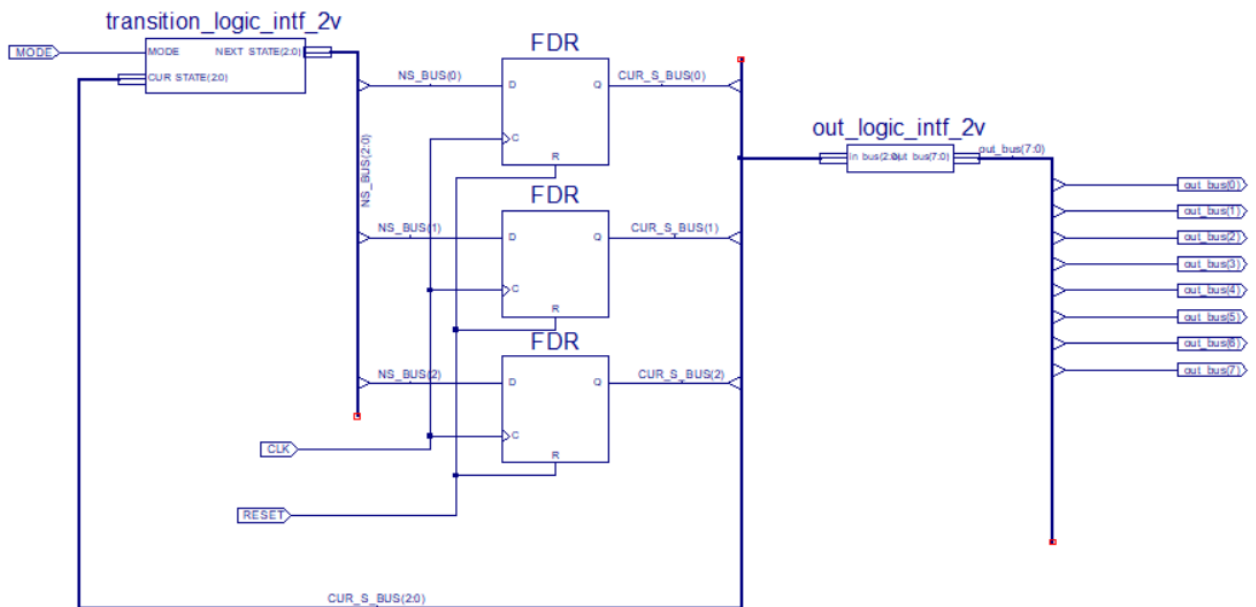


Рис.3: Schematic файл (LightController.sch)

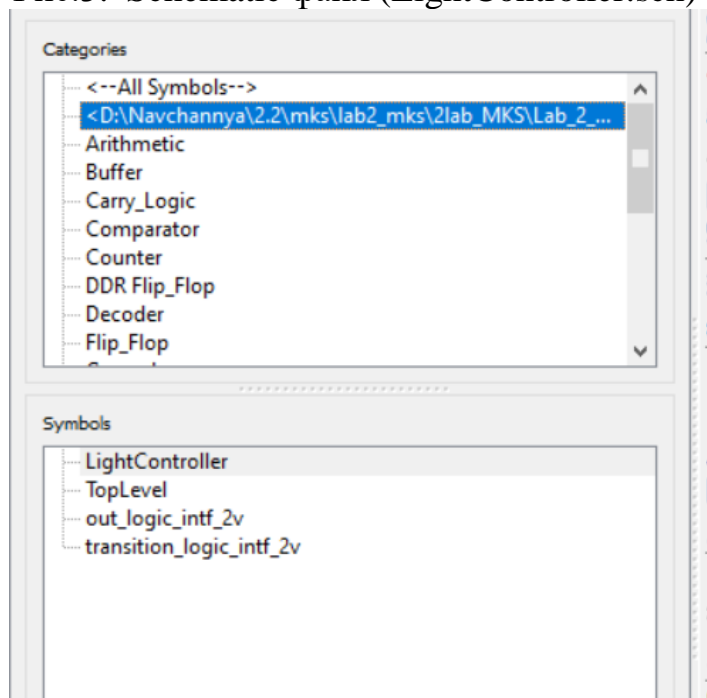


Рис.4: згенеровані Schematic символи

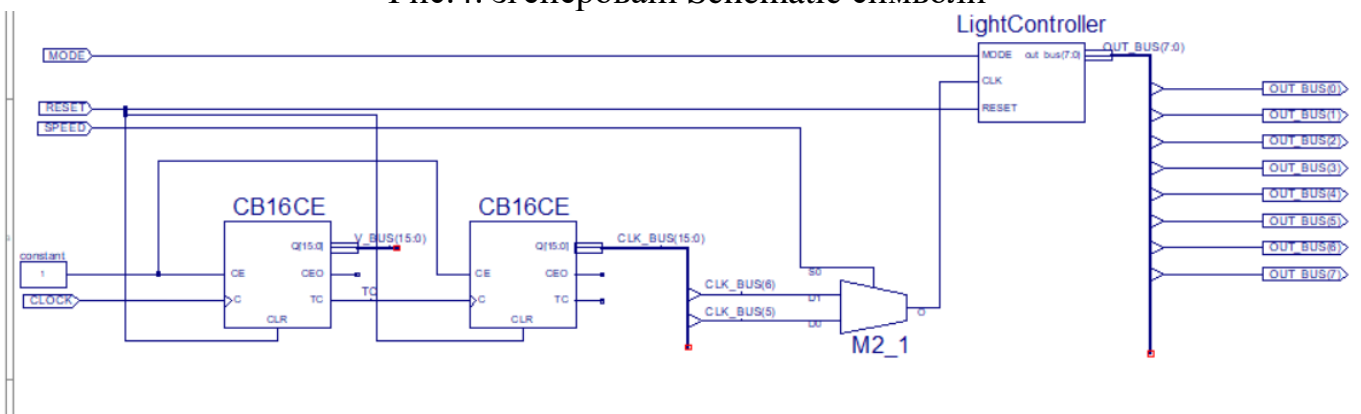


Рис.5: Schematic файл для кінцевої схеми.(TopLevel.sch)

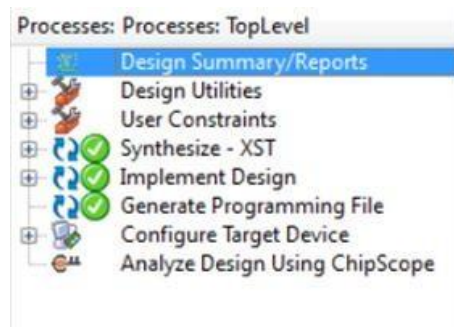


Рис.6: згенерований бінарний файл

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