

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних технологій, автоматики та метрології

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 - Spartan 3A FPGA»

Виконав: ст. гр. КІ-202

Аксьонов Р.О.

Прийняв: старший викладач

Козак Н.Б.

Мета: 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. Про моделювати 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

Вхідне завдання:

Варіант 1:

- 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	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	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 ВИЩОЮ 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**).

Хід роботи:

1. У середовищі Xilinx ISE створив новий проєкт. Налаштував цільову FPGA, обрав інструменти для синтезу і симуляції.
2. Додав VHDL файл Output_Logic до проєкту та імплементував інтерфейс логіки формування вихідних сигналів, а також логічні вирази для формування кожного вихідного сигналу, залежно від поточного стану автомата.

```
1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity Output_Logic_intf is
6 Port (IN_BUS : in std_logic_vector( 2 downto 0 );
7       OUT_BUS : out std_logic_vector ( 7 downto 0 )
8       );
9 end Output_Logic_intf;
10
11 architecture Output_Logic_arch of Output_Logic_intf is
12
13 begin
14     OUT_BUS(0) <= (not(IN_BUS(2)) and not(IN_BUS(1)) and not(IN_BUS(0)));
15     OUT_BUS(1) <= (not(IN_BUS(2)) and not(IN_BUS(1)) and IN_BUS(0));
16     OUT_BUS(2) <= (not(IN_BUS(2)) and IN_BUS(1) and not(IN_BUS(0)));
17     OUT_BUS(3) <= (not(IN_BUS(2)) and IN_BUS(1) and IN_BUS(0));
18     OUT_BUS(4) <= (IN_BUS(2) and not(IN_BUS(1)) and not(IN_BUS(0)));
19     OUT_BUS(5) <= (IN_BUS(2) and not(IN_BUS(1)) and IN_BUS(0));
20     OUT_BUS(6) <= (IN_BUS(2) and IN_BUS(1) and not(IN_BUS(0)));
21     OUT_BUS(7) <= (IN_BUS(2) and IN_BUS(1) and IN_BUS(0));
22
23 end Output_Logic_arch;
```

3. За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.



4. Додав до проекту VHDL файл Transition_Logic, в якому реалізував логіку формування переходів.

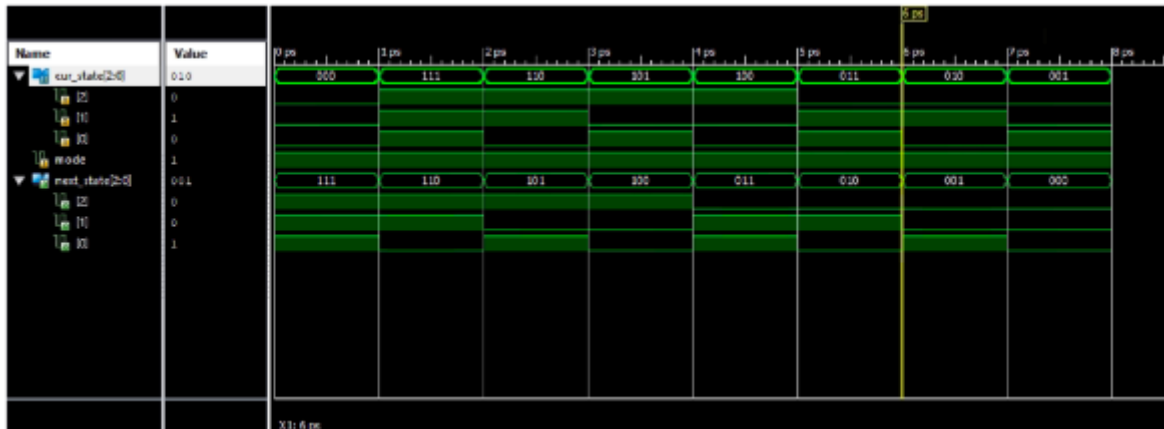
```

5 entity Transition_Logic_intf is
6   Port (CUR_STATE : in std_logic_vector (2 downto 0);
7         MODE : in std_logic;
8         NEXT_STATE : out std_logic_vector (2 downto 0)
9         );
10 end Transition_Logic_intf;
11
12 architecture Transition_Logic_arch of Transition_Logic_intf is
13
14 begin
15   NEXT_STATE(0) <= (MODE and not (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
16   (MODE and not (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0))) or
17   (MODE and (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
18   (MODE and (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0))) or
19   (not (MODE) and not (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
20   (not (MODE) and not (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0))) or
21   (not (MODE) and (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
22   (not (MODE) and (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0)));
23
24   NEXT_STATE(1) <= (MODE and not (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
25   (MODE and (CUR_STATE(2)) and (CUR_STATE(1)) and (CUR_STATE(0))) or
26   (MODE and (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
27   (MODE and not (CUR_STATE(2)) and (CUR_STATE(1)) and (CUR_STATE(0))) or
28   (not (MODE) and not (CUR_STATE(2)) and not (CUR_STATE(1)) and (CUR_STATE(0))) or
29   (not (MODE) and not (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0))) or
30   (not (MODE) and (CUR_STATE(2)) and not (CUR_STATE(1)) and (CUR_STATE(0))) or
31   (not (MODE) and (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0)));
32
33   NEXT_STATE(2) <= (MODE and not (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
34   (MODE and (CUR_STATE(2)) and (CUR_STATE(1)) and (CUR_STATE(0))) or
35   (MODE and (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0))) or
36   (MODE and (CUR_STATE(2)) and not (CUR_STATE(1)) and (CUR_STATE(0))) or
37   (not (MODE) and not (CUR_STATE(2)) and (CUR_STATE(1)) and (CUR_STATE(0))) or
38   (not (MODE) and (CUR_STATE(2)) and not (CUR_STATE(1)) and not (CUR_STATE(0))) or
39   (not (MODE) and (CUR_STATE(2)) and not (CUR_STATE(1)) and (CUR_STATE(0))) or
40   (not (MODE) and (CUR_STATE(2)) and (CUR_STATE(1)) and not (CUR_STATE(0)));
41
42 end Transition_Logic_arch;
43
44

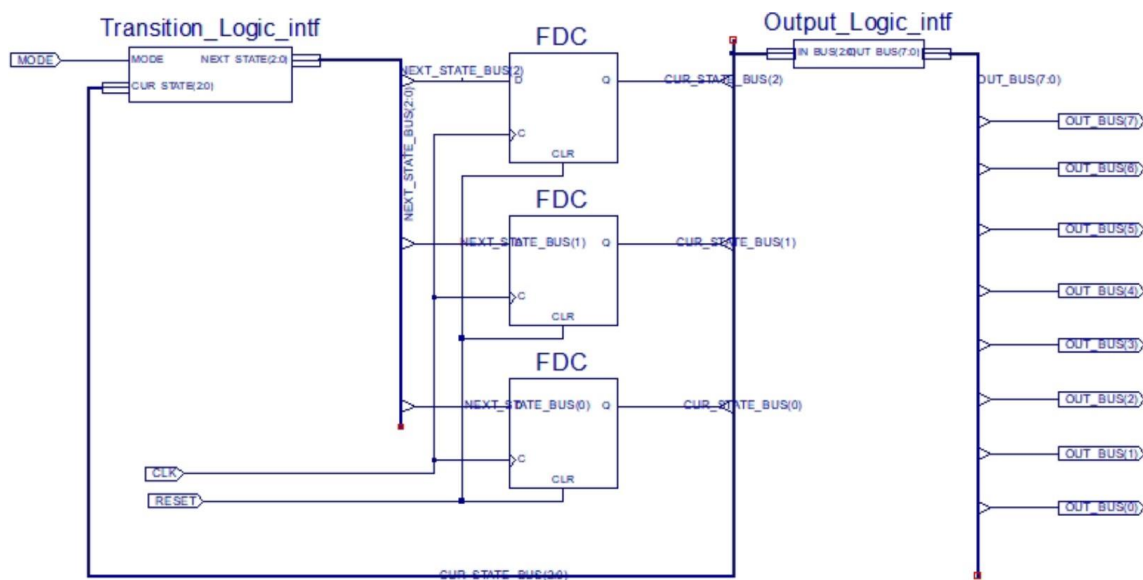
```

5. За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.

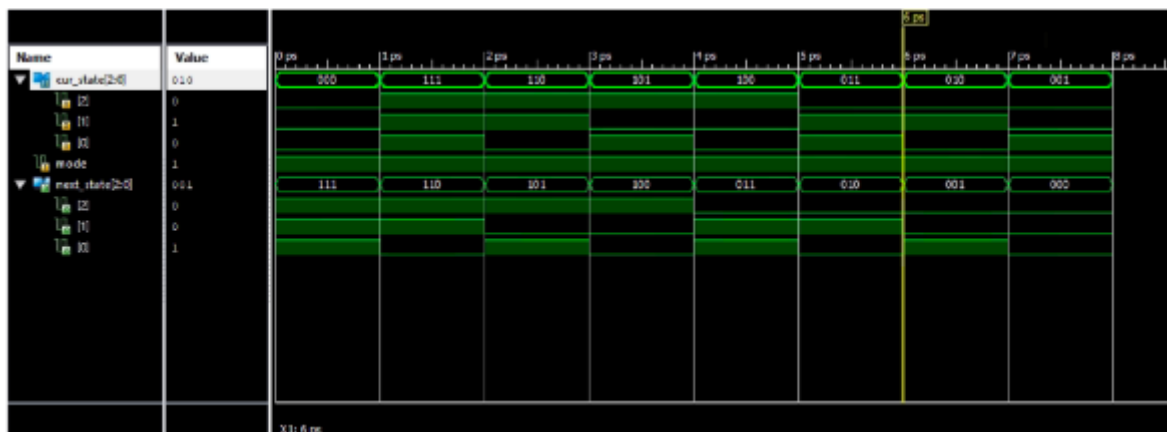




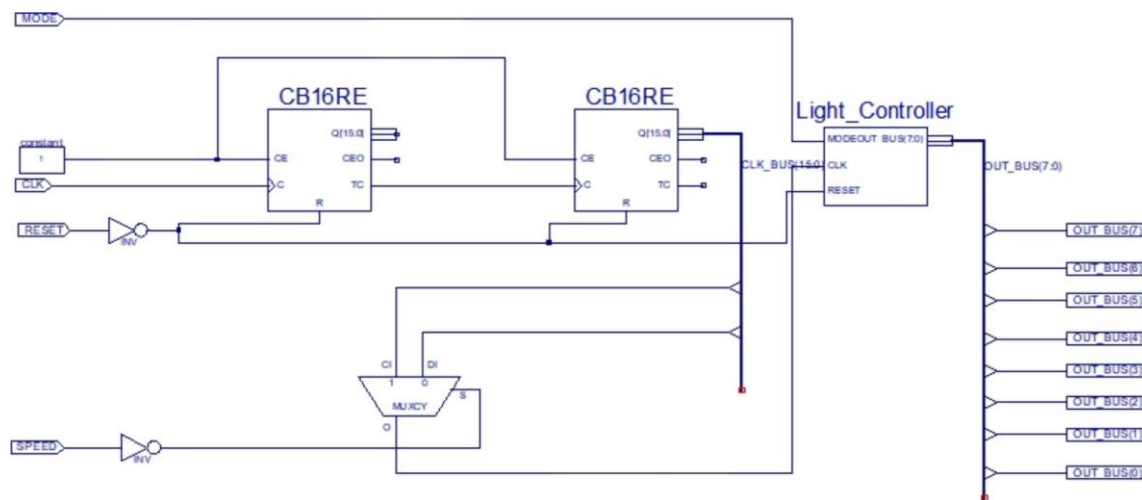
6. Додав до проекту Schematic файл Light_Controller, виконав для нього команду Set as Top Module. Згенерував Schematic символи для файлів Output_Logic і Transition_Logic. Використовуючи новостворені символи та елементи з бібліотеки, реалізував у файлі Light_Controller.sch пам'ять стану автомата.



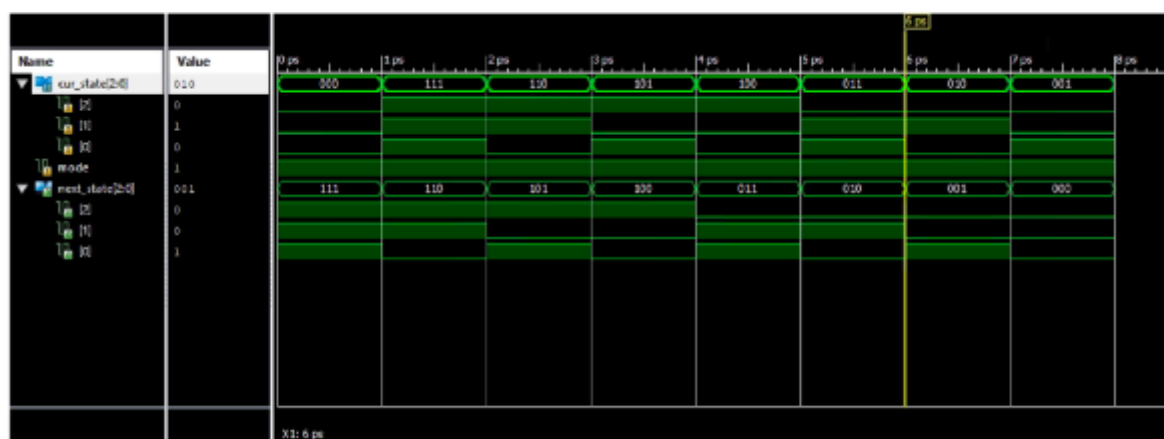
7. За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.



8. Додав до проєкту Schematic файл Top_Level, виконав для нього команду Set as Top Module. Згенерував Schematic символ для файлу Light_Controller. Використовуючи новостворений символ та елементи з бібліотеки, реалізував у файлі подільник входної частоти та логіку сигналу SPEED.





9. За допомогою симулятора Isim провів моделювання роботи схеми з різними значеннями сигналів MODE, RESET та SPEED при подачі на вхід CLOCK тактового сигналу 12 MHz.



10. Додав до проєкту Constraint файл та призначила виводам схеми фізичні виводи цільової FPGA.

```
1 CONFIG VCCAUX = "3.3" ;
2
3 # Clock 12 MHz
4 NET "CLK" LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
5
6 #####
7 # LED
8 #####
9
10 NET "OUT_BUS(0)" LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
11 NET "OUT_BUS(1)" LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
12 NET "OUT_BUS(2)" LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
13 NET "OUT_BUS(3)" LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
14 NET "OUT_BUS(4)" LOC = P50 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
15 NET "OUT_BUS(5)" LOC = P51 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
16 NET "OUT_BUS(6)" LOC = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
17 NET "OUT_BUS(7)" LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
18
19 #####
20 # DP Switches
21 #####
22
23 NET "MODE" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
24
25 #####
26 # Switches
27 #####
28
29 NET "SPEED" LOC = P80 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
30 NET "RESET" LOC = P79 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
31
32 #####
```

11. Відкрив директорію проєкту у файловому менеджері та переконався, що файли прошиття успішно згенерувалися.

 top_level.bin	05.05.2022 16:38	Файл BIN	54 КБ
 top_level.bit	05.05.2022 16:38	Файл BIT	54 КБ

12. Взяв ці файли та через ElbertV2Config завантажив їх у робочий стенд.



Висновок: під час виконання цієї лабораторної роботи я реалізував цифровий автомат світлових ефектів у середовищі Xilinx ISE і стендом Elbert V2 - Spartan 3A FPGA, реалізував схему автомату та провів симуляцію його роботи.