

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 - Spartan 3A FPGA»

Виконав: ст. гр. КІ-201

Олеш С.Б.

Прийняв:

Козак Н.Б.

Львів – 2023

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. 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:

Варіант 5

- 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 $LOC = P129\text{ FPGA}$ (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 ($0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0 \dots$).
 - Якщо $MODE=1$ то стан пристрою декрементується по зростаючому фронту тактового сигналу пам'яті станів ($0 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \dots$).
- 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**).

Хід роботи:

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

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

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



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

```

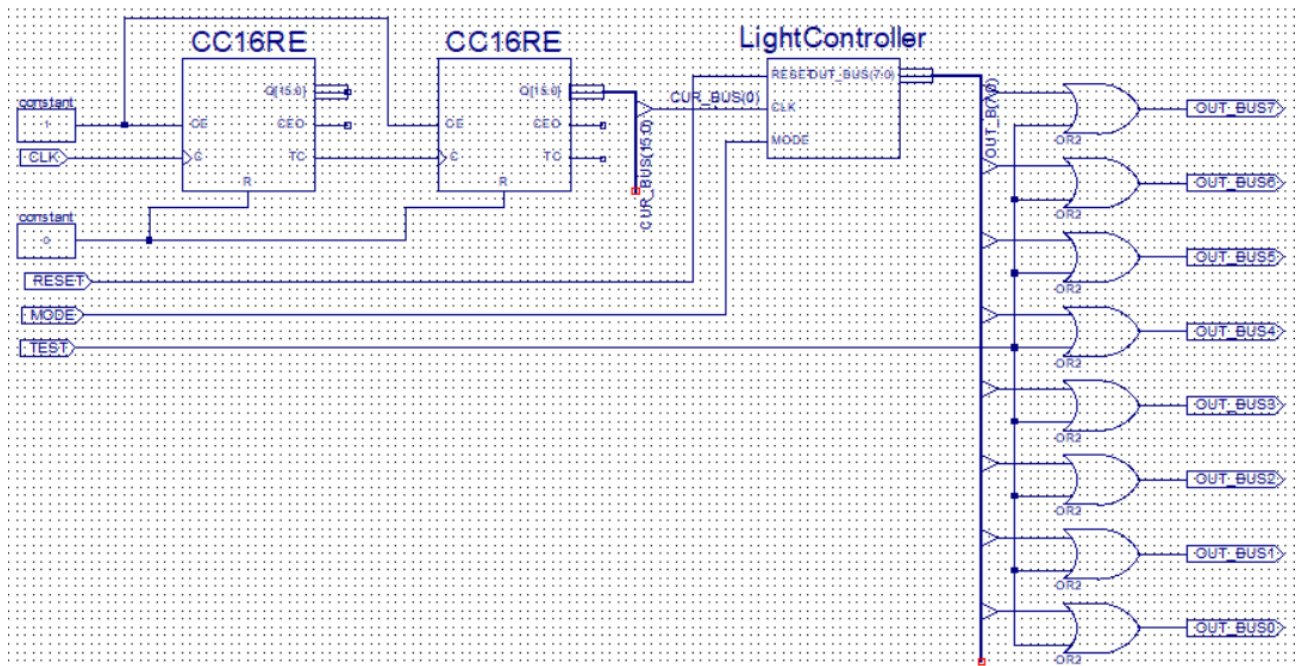
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity transition_logic_intf is
5  Port (MODE : in  std_logic;
6        CUR_STATE : in  std_logic_vector(2 downto 0);
7        NEXT_STATE : out std_logic_vector(2 downto 0)
8        );
9  end transition_logic_intf;
10
11 architecture transition_logic_arch of transition_logic_intf is
12
13 begin
14
15     NEXT_STATE(0) <= not CUR_STATE(0);
16
17     NEXT_STATE(1) <= (MODE and not(CUR_STATE(1)) and not(CUR_STATE(0))) or
18                     (MODE and CUR_STATE(1) and CUR_STATE(0)) or
19                     (not(MODE) and CUR_STATE(1) and not(CUR_STATE(0))) or
20                     (not(MODE) and not(CUR_STATE(1)) and CUR_STATE(0));
21
22     NEXT_STATE(2) <= (CUR_STATE(2) and CUR_STATE(1) and not(CUR_STATE(0))) or
23                     (MODE and CUR_STATE(2) and CUR_STATE(0)) or
24                     (CUR_STATE(2) and not(CUR_STATE(1)) and CUR_STATE(0)) or
25                     (not(MODE) and not(CUR_STATE(2)) and CUR_STATE(1) and CUR_STATE(0)) or
26                     (MODE and not(CUR_STATE(2)) and not(CUR_STATE(1)) and not(CUR_STATE(0))) or
27                     (not(MODE) and CUR_STATE(2) and not(CUR_STATE(0)));
28
29 end transition_logic_arch;

```

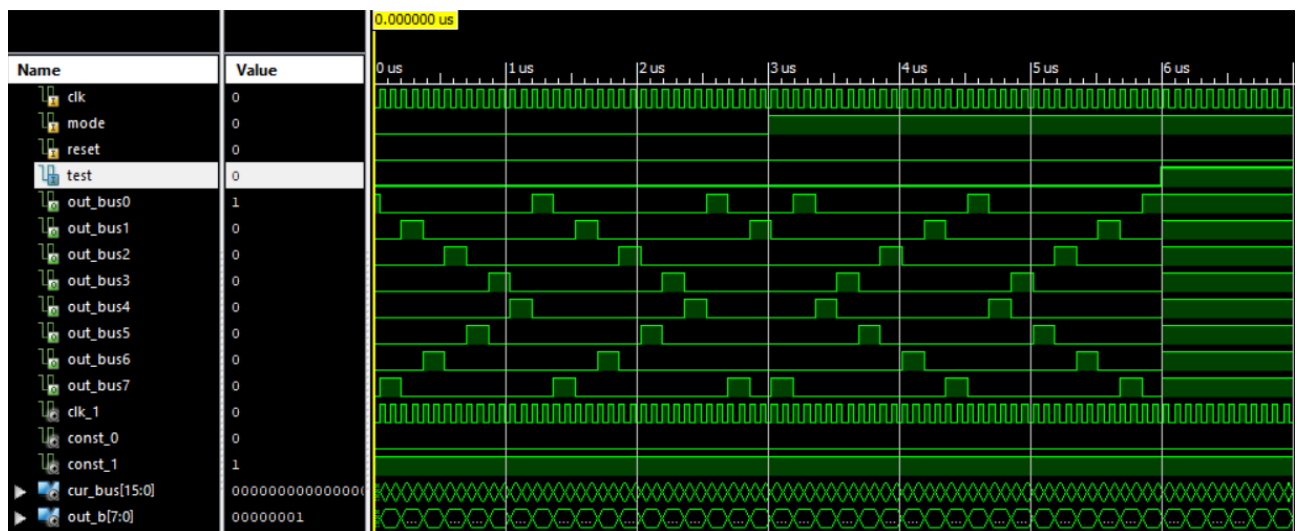
5. За допомогою симулятора Isim провів моделювання роботи схеми при всіх можливих комбінаціях сигналів на входах.



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



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





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


```

11 CONFIG VCCAUX = "3.3" ;
12
13 # Clock 12 MHz
14 NET "Clk" LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
15
16 #####
17 # LED
18 #####
19
20 NET "OUT_BUS0" LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
21 NET "OUT_BUS1" LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
22 NET "OUT_BUS2" LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
23 NET "OUT_BUS3" LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
24 NET "OUT_BUS4" LOC = P50 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
25 NET "OUT_BUS5" LOC = P51 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
26 NET "OUT_BUS6" LOC = P54 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
27 NET "OUT_BUS7" LOC = P55 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
28
29 #####
30 # DP Switches
31 #####
32
33 NET "MODE" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
34 NET "RESET" LOC = P69 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
35 NET "TEST" LOC = P68 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;

```

11. Відкрив директорію проекту у файловому менеджері та переконалася, що файли прошиття успішно згенерувалися.

 toplevel.bin	09.05.2023 9:46	Файл BIN	54 КБ
 toplevel.bit	09.05.2023 9:46	Файл BIT	54 КБ

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