

Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи № 2

з дисципліни: «Моделювання комп'ютерних систем»

«Структурний опис цифрового автомата Перевірка роботи автомата за  
допомогою стенда. Elbert V2 – Spartan 3A FPGA»  
Варіант 13

Виконав:

ст. гр. КІ-201

Костюк І. В.

Прийняв:

Козак Н. Б.

Львів 2023

**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 Дивись розділ Завдання
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів Заборонено використовуючи оператори
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів Заборонено використовуючи оператори
4. Згенерувати символи для описів логіки переходів та логіки формування вихідних сигналів
5. Згрупувати всі компоненти логіки переходів логіку формування вихідних сигналів та пам'ять станів в єдину систему. 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 додати подільник частоти для вхідного тактового сигналу призначити фізичні виводи.
8. Згенерувати файл та перевірити роботу за допомогою стенда Elbert V2 – Spartan 3A FPGA.
9. Підготувати і захистити звіт

## 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:

Створив новий проект користуючись методичними вказівками до лабораторної роботи №1.

Додав до проекту новий VHDL файл Output\_Logic, в якому буде реалізовано логіку формування вихідних сигналів.

## Лістинг:

```
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)) and not (IN_BUS(1)) and not (IN_BUS(0))); -- 0
000

    OUT_BUS(1) <= (not (IN_BUS(2)) and not (IN_BUS(1)) and (IN_BUS(0)));      -- 1
001

    OUT_BUS(2) <= (not(IN_BUS(2)) and (IN_BUS(1)) and not(IN_BUS(0)));        -- 2
010

    OUT_BUS(3) <= (not(IN_BUS(2)) and (IN_BUS(1)) and (IN_BUS(0)));           -- 3
011

    OUT_BUS(4) <= ((IN_BUS(2)) and not(IN_BUS(1)) and not(IN_BUS(0)));        -- 4
100

    OUT_BUS(5) <= ((IN_BUS(2)) and not(IN_BUS(1)) and (IN_BUS(0)));           -- 5
101

    OUT_BUS(6) <= ((IN_BUS(2)) and (IN_BUS(1)) and not(IN_BUS(0)));          -- 6
110

    OUT_BUS(7) <= ((IN_BUS(2)) and (IN_BUS(1)) and (IN_BUS(0)));             -- 7
111

end out_logic_arch;
```

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

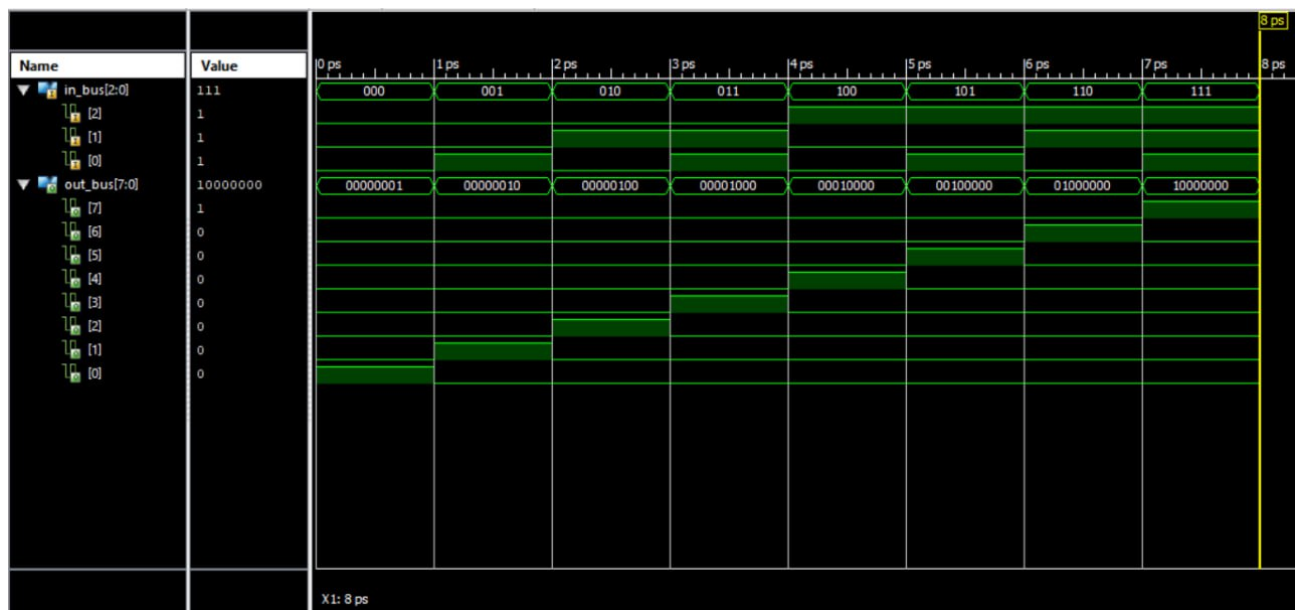


Рис. 1. Симуляція Output\_Logic

Додав до проекту VHDL файл Transition\_Logic, в якому реалізував логіку формування переходів:

### Лістинг:

```
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(2))and not(CUR_STATE(1))
and not(CUR_STATE(0))) or    --0

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

                                (not(MODE) and (CUR_STATE(2))and not(CUR_STATE(1)) and
not(CUR_STATE(0))) or    --4

                                (not(MODE) and (CUR_STATE(2))and  (CUR_STATE(1)) and not
(CUR_STATE(0))) or    --6

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

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

                                ((MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and not
(CUR_STATE(0))) or    --4

                                ((MODE) and (CUR_STATE(2))and  (CUR_STATE(1)) and not
(CUR_STATE(0))) ;    --6


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

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

                                (not(MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and
(CUR_STATE(0))) or    --5

                                (not(MODE) and (CUR_STATE(2))and  (CUR_STATE(1)) and
not(CUR_STATE(0))) or    --6

                                ((MODE) and not(CUR_STATE(2))and  not(CUR_STATE(1)) and
not (CUR_STATE(0))) or    --4

                                ((MODE) and not(CUR_STATE(2))and  (CUR_STATE(1)) and
(CUR_STATE(0))) or    --7

                                ((MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and
not(CUR_STATE(0))) or    --3

                                ((MODE) and (CUR_STATE(2))and  (CUR_STATE(1)) and
(CUR_STATE(0))) ;    --0


    NEXT_STATE(2) <=      (not(MODE) and not(CUR_STATE(2))and  (CUR_STATE(1)) and
(CUR_STATE(0))) or    --3

                                (not(MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and
not (CUR_STATE(0))) or    --4

                                (not(MODE) and (CUR_STATE(2))and not (CUR_STATE(1)) and
(CUR_STATE(0))) or    --5

```

```

        (not(MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not
(CUR_STATE(0))) or      --6

        ((MODE) and not(CUR_STATE(2))and not(CUR_STATE(1)) and not
(CUR_STATE(0))) or      --7

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

        ((MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and not
(CUR_STATE(0))) or      --5

        ((MODE) and (CUR_STATE(2))and (CUR_STATE(1)) and
(CUR_STATE(0))) ;      --6

end transition_logic_arch;

```

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



Рис. 2. Симуляція Transition\_Logic 0 -> 7



Рис. 3. Симуляція Transition\_Logic 7 -> 0

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.

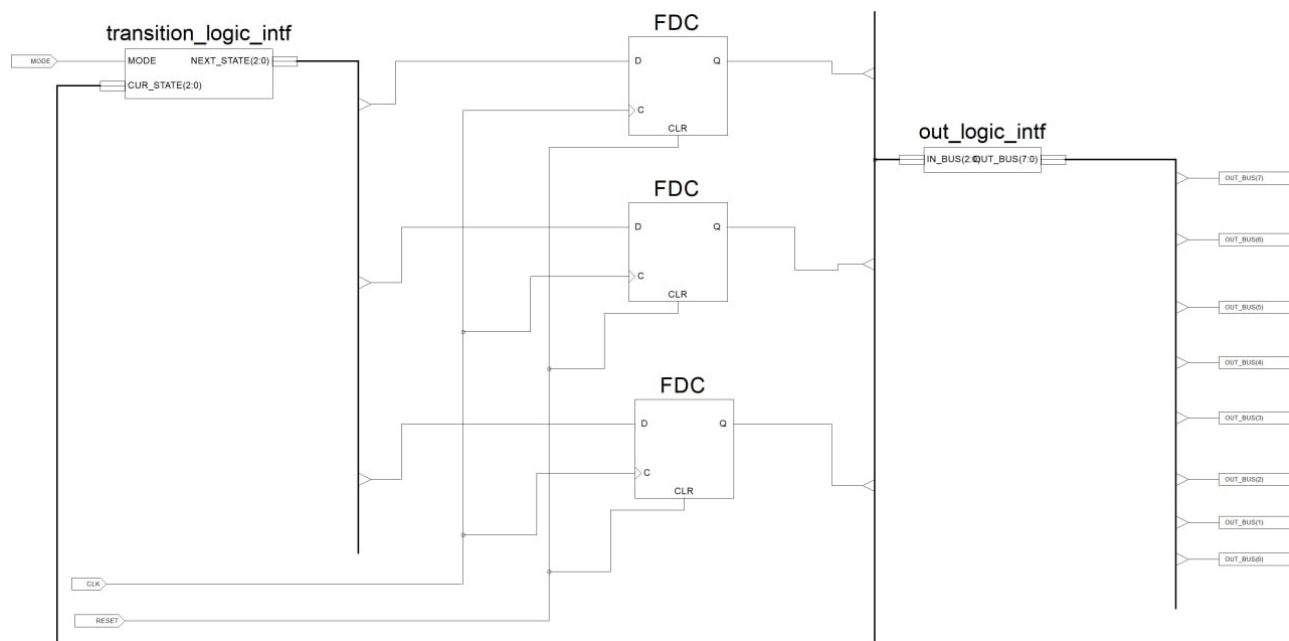


Рис. 4. Schematic Light\_Controller

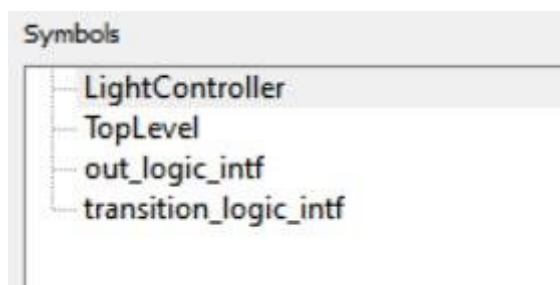


Рис. 5. Згенеровані Schematic символи для файлів Output\_Logic і Transition\_Logic

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

При MODE 0:

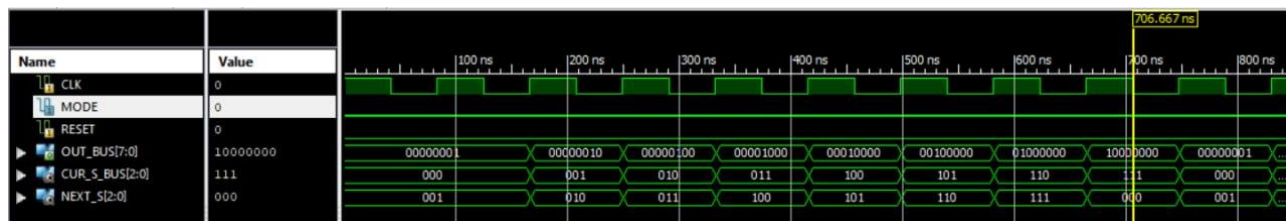


Рис. 6. MODE 0



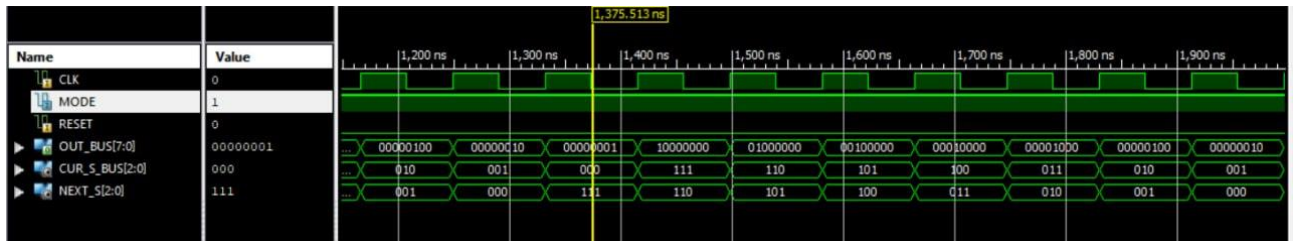


Рис. 7. MODE 1



Рис. 8. RESET

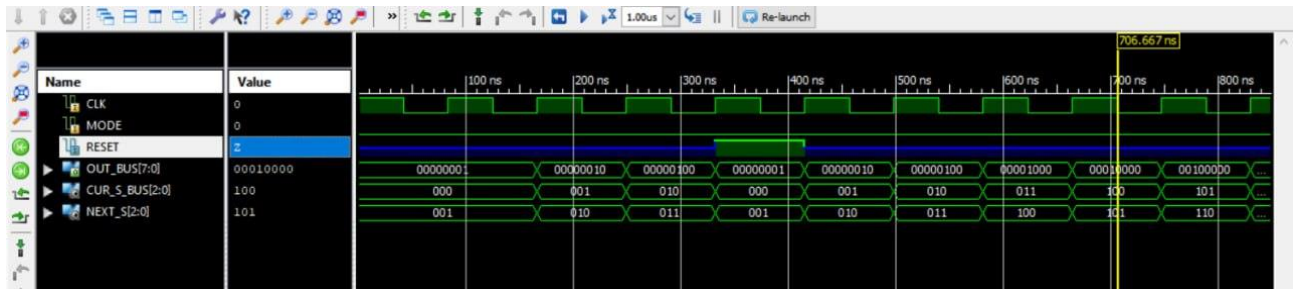


Рис. 9. RESET

Додав до проекту *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*.

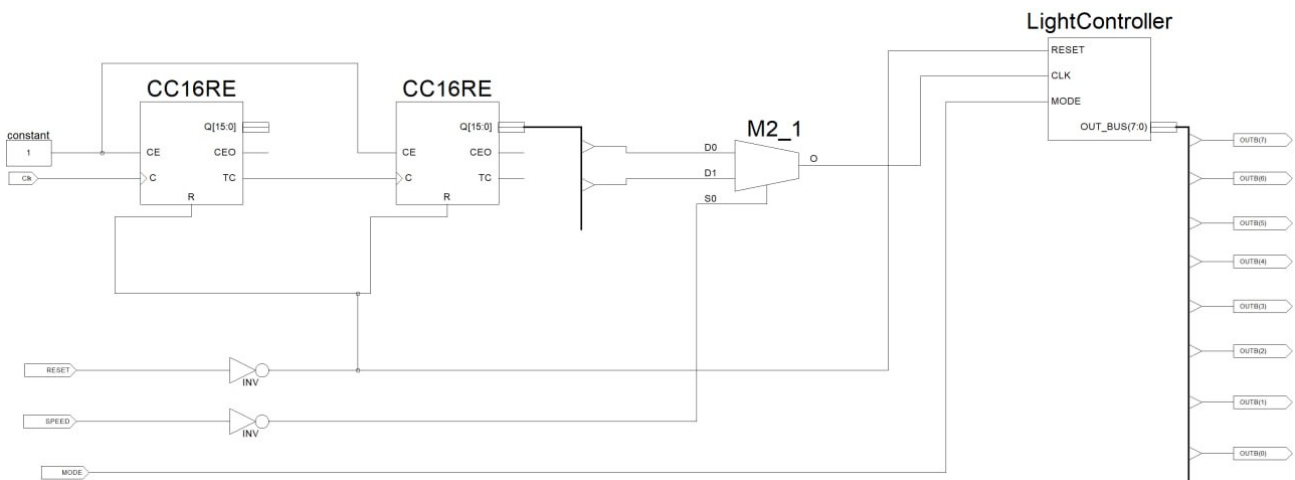


Рис. 10. Top\_Level

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

SPEED =1, MODE = 0 (до лінії)

SPEED =0 MODE = 1 (після лінії).

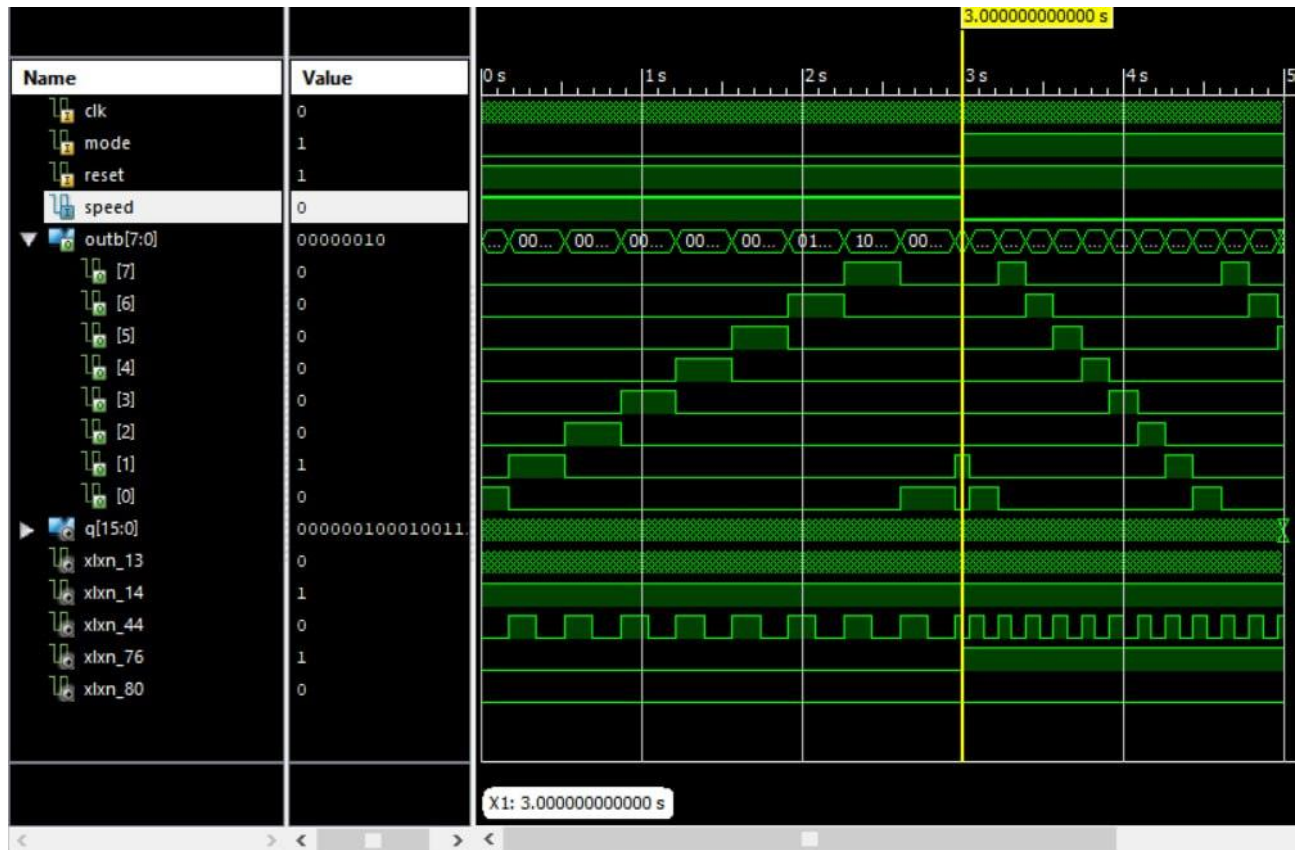


Рис. 11. Симуляція Top\_Level

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

### Лістинг:

```
#+++++
+++++
#

# This file is a .ucf for ElbertV2 Development Board
#

# To use it in your project :
#

# * Remove or comment the lines corresponding to unused pins in the project
#
```

```

# * Rename the used signals according to the your project
#

#++++++
+++++

*****
*****#

#
#                               UCF for ElbertV2 Development Board
#

*****
*****#

CONFIG VCCAUX = "3.3" ;

#Clock 12 MHz

NET "Clk"                      LOC = P129   | IOSTANDARD = LVCMOS33 | PERIOD =
12MHz;

#####
#####

#                               LED

#####
#####

NET "OUTB(0)"                  LOC = P46    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(1)"                  LOC = P47    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(2)"                  LOC = P48    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(3)"                  LOC = P49    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(4)"                  LOC = P50    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(5)"                  LOC = P51    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(6)"                  LOC = P54    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

NET "OUTB(7)"                  LOC = P55    | IOSTANDARD = LVCMOS33 | SLEW = SLOW
| DRIVE = 12;

#####
#####

#                               DP Switches

```

```
#####
#####

    NET "MODE"          LOC = P70      | PULLUP    | IOSTANDARD = LVCMOS33 | SLEW =
SLOW | DRIVE = 12;

#####
#####



#                               Switches

#####
#####

    NET "RESET"          LOC = P76      | PULLUP    | IOSTANDARD = LVCMOS33 | SLEW =
SLOW | DRIVE = 12;

    NET "SPEED"          LOC = P75      | PULLUP    | IOSTANDARD = LVCMOS33 | SLEW =
SLOW | DRIVE = 12;
```

Згенерував файли прошиття.

 toplevel.bin	06.05.2023 18:06	Файл BIN	54 КБ
 toplevel.bit	06.05.2023 18:06	Файл BIT	54 КБ

*Рис. 12. Файли прошиття*

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