Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до 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 №14

Виконав:

ст. гр. КІ-201

Когут Д.А.

Прийняв:

ст. викладач   
каф. ЕОМ  
Козак Н. Б.

Львів 2024

**Мета роботи:** На базі стенда реалізувати цифровий автомат світлових ефектів.

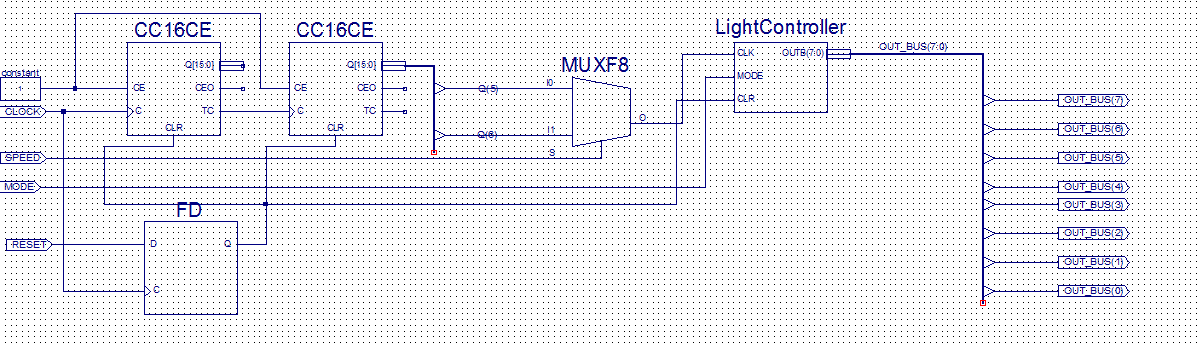
Згідно мого варіанту(2) потрібно реалізувати такі комбінації:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

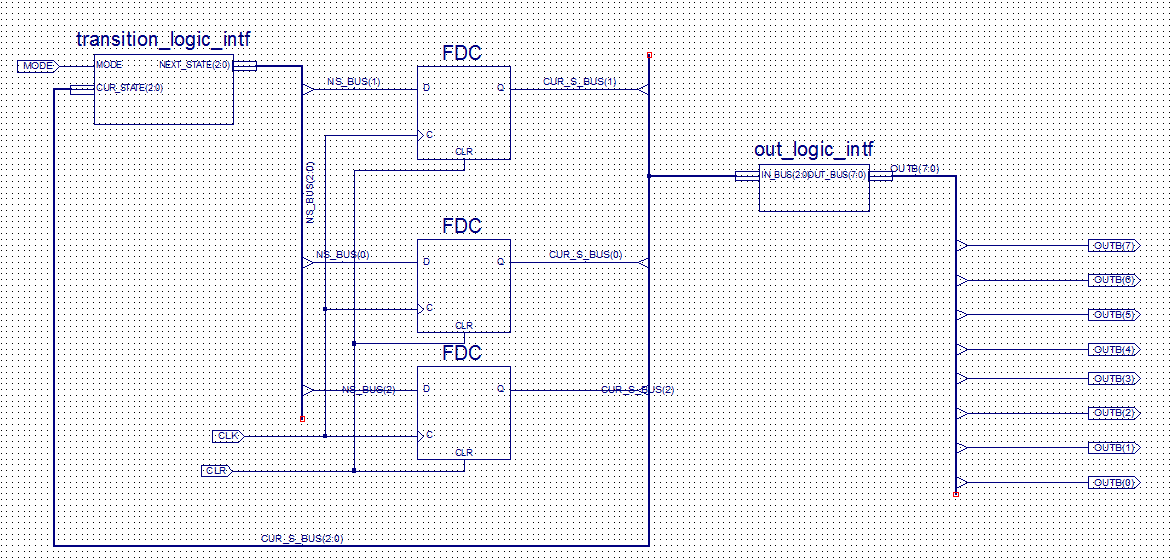
Та дотримуватися таких вимог:

* Пристрій повинен використовувати тактовий сигнал від мікроконтролера і знижувати частоту за допомогою внутрішнього подільника. Мікроконтролер є частиною стенда Elbert V2 – Spartan 3A FPGA. Тактовий сигнал заведено на вхід LOC P129 FPGA
* 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.
  + Якщо MODE = 1 то стан пристрою декрементується по зростаючому фронту тактового сигналу пам’яті станів.
* 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 перемикачів
* Для керування сигналами RESET/SPEED використати будь які з кнопок

**Виконання роботи:**

****

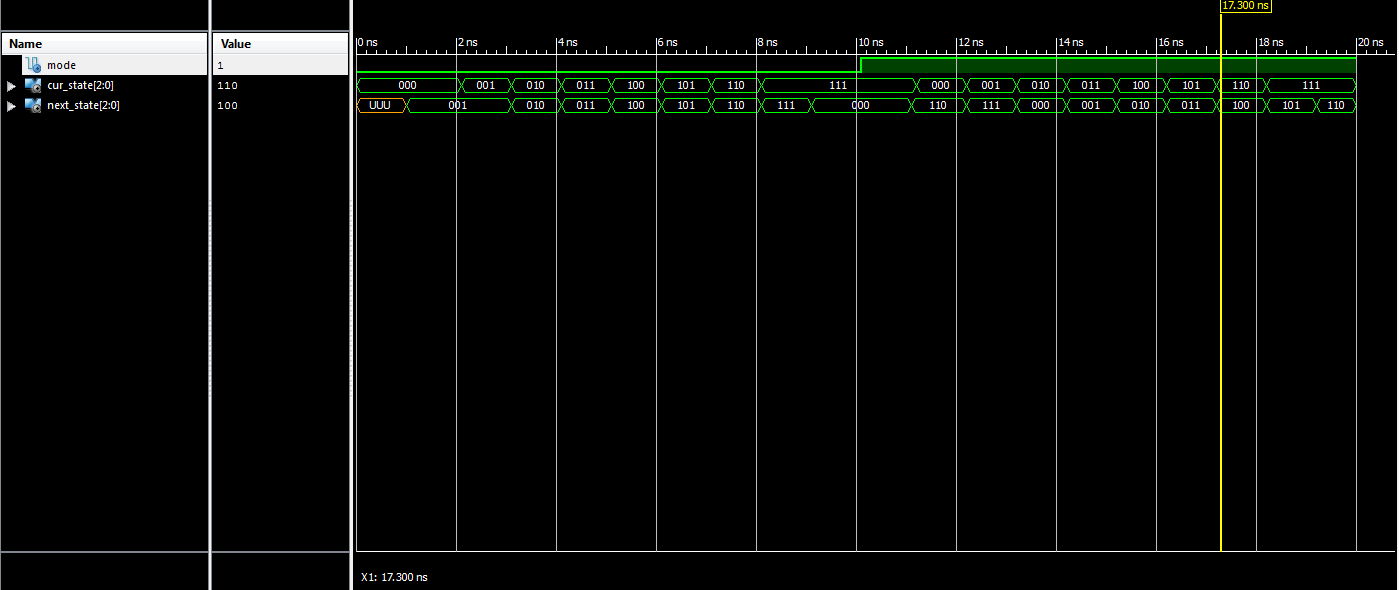
*Рис. 1 – Top Level*

**

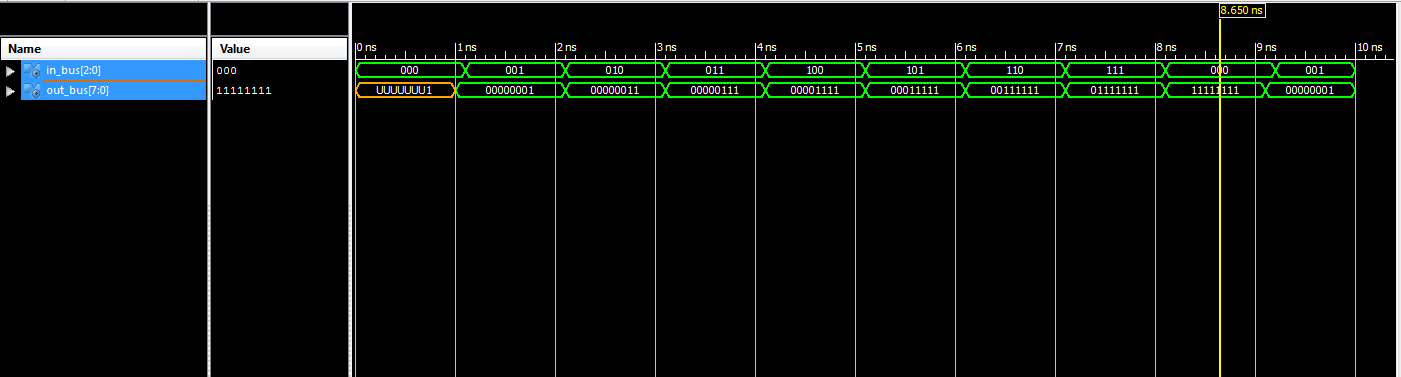
*Рис. 2 – Light Controller*

|  |
| --- |
| Файл TransitionLogic.vhd  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);  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 CUR\_STATE(2) and not CUR\_STATE(1) and not CUR\_STATE(0)) or  (not CUR\_STATE(2) and CUR\_STATE(1) and not CUR\_STATE(0)) or  (CUR\_STATE(2) and not CUR\_STATE(1) and not CUR\_STATE(0)) or  (CUR\_STATE(2) and CUR\_STATE(1) and not CUR\_STATE(0))) after 1ns;    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 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 CUR\_STATE(2) and not CUR\_STATE(1) and not CUR\_STATE(0)) or  (MODE and not CUR\_STATE(2) and CUR\_STATE(1) and CUR\_STATE(0))) after 1ns;    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 not 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 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))) after 1ns;  end transition\_logic\_arch; |

|  |
| --- |
| Файл OutputLogic.vhd  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 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) <= '1';  OUT\_BUS(1) <= not (not IN\_BUS(2) and not IN\_BUS(1) and not IN\_BUS(0)) after 1ns;  OUT\_BUS(2) <= not (not IN\_BUS(2) and not IN\_BUS(1)) after 1ns;  OUT\_BUS(3) <= not ((not IN\_BUS(2) and not IN\_BUS(1)) or (not IN\_BUS(2) and IN\_BUS(1) and not IN\_BUS(0))) after 1ns;  OUT\_BUS(4) <= IN\_BUS(2) after 1ns;  OUT\_BUS(5) <= ((IN\_BUS(2) and IN\_BUS(1)) or (IN\_BUS(2) and not IN\_BUS(1) and IN\_BUS(0))) after 1ns;  OUT\_BUS(6) <= (IN\_BUS(2) and IN\_BUS(1)) after 1ns;  OUT\_BUS(7) <= (IN\_BUS(2) and IN\_BUS(1) and IN\_BUS(0)) after 1ns;  end out\_logic\_arch; |



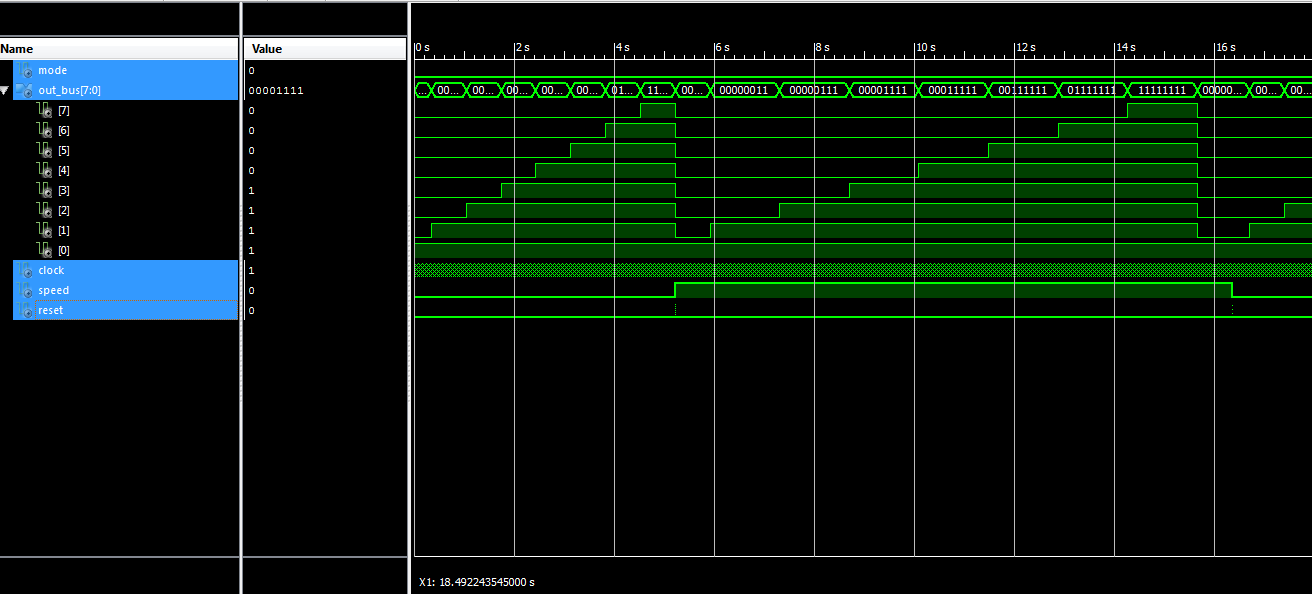
*Рис. 3 – Часова діаграма TransitionLogic*

**

*Рис. 4 – Часова діаграма OutputLogic*

**

*Рис. 5 – Часова діаграма LightController*

**

*Рис 6. – Часова діграма TopLevel*

|  |
| --- |
| *Файл TestBechTopLevel.vsd LIBRARY ieee;*  *USE ieee.std\_logic\_1164.ALL;*  *USE ieee.numeric\_std.ALL;*  *LIBRARY UNISIM;*  *USE UNISIM.Vcomponents.ALL;*  *ENTITY TopLevel\_TopLevel\_sch\_tb IS*  *END TopLevel\_TopLevel\_sch\_tb;*  *ARCHITECTURE behavioral OF TopLevel\_TopLevel\_sch\_tb IS*  *COMPONENT TopLevel*  *PORT( MODE : IN STD\_LOGIC;*  *OUT\_BUS : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);*  *CLOCK : IN STD\_LOGIC;*  *SPEED : IN STD\_LOGIC;*  *RESET : IN STD\_LOGIC);*  *END COMPONENT;*  *SIGNAL MODE : STD\_LOGIC;*  *SIGNAL OUT\_BUS : STD\_LOGIC\_VECTOR (7 DOWNTO 0);*  *SIGNAL CLOCK : STD\_LOGIC := '0';*  *SIGNAL SPEED : STD\_LOGIC;*  *SIGNAL RESET : STD\_LOGIC;*  *BEGIN*  *CLOCK <= not CLOCK after 83ns;*  *UUT: TopLevel PORT MAP(*  *MODE => MODE,*  *OUT\_BUS => OUT\_BUS,*  *CLOCK => CLOCK,*  *SPEED => SPEED,*  *RESET => RESET*  *);*  *-- \*\*\* Test Bench - User Defined Section \*\*\**  *tb : PROCESS*  *BEGIN*  *MODE <= '0';*  *SPEED <= '0';*  *RESET <= '1', '0' after 1ms;*  *wait until RESET = '0';*  *wait for 2ns;*  *assert OUT\_BUS = "00000001";*  *wait for 348128us;*  *assert OUT\_BUS = "00000011";*  *wait for 696255us;*  *assert OUT\_BUS = "00000111";*  *wait for 696255us;*  *assert OUT\_BUS = "00001111";*  *wait for 696255us;*  *assert OUT\_BUS = "00011111";*  *wait for 696255us;*  *assert OUT\_BUS = "00111111";*  *wait for 696255us;*  *assert OUT\_BUS = "01111111";*  *wait for 696255us;*  *assert OUT\_BUS = "11111111";*  *wait for 696255us;*  *SPEED <= '1';*  *RESET <= '1', '0' after 1ms;*  *wait until RESET = '0';*  *wait for 2ms;*  *assert OUT\_BUS = "00000001";*  *wait for 1392509us;*  *assert OUT\_BUS = "00000011";*  *wait for 1392509us;*  *assert OUT\_BUS = "00000111";*  *wait for 1392509us;*  *assert OUT\_BUS = "00001111";*  *wait for 1392509us;*  *assert OUT\_BUS = "00011111";*  *wait for 1392509us;*  *assert OUT\_BUS = "00111111";*  *wait for 1392509us;*  *assert OUT\_BUS = "01111111";*  *wait for 1392509us;*  *assert OUT\_BUS = "11111111";*  *wait for 1392509us;*  *SPEED <= '0';*  *RESET <= '1', '0' after 167ns;*  *wait until RESET = '0';*  *END PROCESS;*  *-- \*\*\* End Test Bench - User Defined Section \*\*\**  *END;* |

**Висновок:** Виконуючи дану лабораторну роботу я навчився реалізовувати цифровий автомат світлових ефектів використовуючи засоби VHDL.