Міністерство освіти і науки України

Національний університет „Львівська політехніка”



**ЛАБОРАТОРНА РОБОТА**

з дисципліни

**МОДЕЛЮВАННЯ КОМП’ЮТЕРНИХ СИСТЕМ**

**Звіт з лабораторної роботи №2**

на тему:

«Структурний опис цифрового автомата Перевірка роботи автомата за допомогою стенда Elbert V2 - Spartan 3A FPGA»

**Виконав:**

**Прийняв:**

студент групи KI-202

Порубайміх О.Є.

Козак Н.Б.

Львів – 2023

**Мета:** на базі стенда Elbert V2 - Spartan 3A FPGA реалізувати цифровий автомат світлових ефектів згідно наступних вимог

**Завдання**

A screenshot of a computer

Description automatically generated with low confidence

**Виконання роботи**

Код OutputLogic

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(0)) and not(IN\_BUS(1)) and not(IN\_BUS(2)));

  OUT\_BUS(1) <= (not(IN\_BUS(1)) and not(IN\_BUS(2)));

  OUT\_BUS(2) <= (not(IN\_BUS(2)) and ((not(IN\_BUS(1)) and IN\_BUS(0)) or (not(IN\_BUS(0)) and IN\_BUS(1))));

  OUT\_BUS(3) <= (not(IN\_BUS(2)) and IN\_BUS(1));

  OUT\_BUS(4) <= ((not(IN\_BUS(2)) and IN\_BUS(1) and IN\_BUS(0)) or (IN\_BUS(2) and not(IN\_BUS(1)) and not(IN\_BUS(0))));

  OUT\_BUS(5) <= (IN\_BUS(2) and not(IN\_BUS(1)));

  OUT\_BUS(6) <= (IN\_BUS(2) and ((IN\_BUS(1) and not(IN\_BUS(0))) or (IN\_BUS(0) and not(IN\_BUS(1)))));

  OUT\_BUS(7) <= (IN\_BUS(2) and IN\_BUS(1));

end out\_logic\_arch;

Код TransitionLogic

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(CUR\_STATE(0)));

  NEXT\_STATE(1) <= ((not(MODE) and ((CUR\_STATE(0) and not(CUR\_STATE(1))) or

                                     (not(CUR\_STATE(0)) and CUR\_STATE(1)))) or

                     (MODE and ((CUR\_STATE(0) and CUR\_STATE(1)) or

                                (not(CUR\_STATE(0)) and not(CUR\_STATE(1))))));

   NEXT\_STATE(2) <= (

     (not(MODE) and (

       (not(CUR\_STATE(2)) and CUR\_STATE(1) and CUR\_STATE(0)) or

       (CUR\_STATE(2) and (not(CUR\_STATE(1)) or not(CUR\_STATE(0))))

     )) or

     (MODE and (

       (not(CUR\_STATE(2)) and not(CUR\_STATE(1)) and not(CUR\_STATE(0))) or

       (CUR\_STATE(2) and (CUR\_STATE(1) or (CUR\_STATE(0))))

     )));

end transition\_logic\_arch;

A diagram of a computer

Description automatically generated with low confidence

Рис. 1. LightController

A diagram of a computer

Description automatically generated with low confidence

Рис. 2. TopLevel

Код testbench

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( CLOCK     :  IN  STD\_LOGIC;

          MODE     :  IN  STD\_LOGIC;

          SPEED : IN  STD\_LOGIC;

          RESET : IN  STD\_LOGIC;

          OUT\_BUS : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0));

   END COMPONENT;

   SIGNAL CLOCK : STD\_LOGIC;

   SIGNAL MODE     :  STD\_LOGIC;

   SIGNAL SPEED : STD\_LOGIC;

   SIGNAL RESET : STD\_LOGIC;

   SIGNAL OUT\_BUS : STD\_LOGIC\_VECTOR (7 DOWNTO 0);

  CONSTANT PERIOD : TIME := 83 NS;           -- CLOCK period

  CONSTANT TEST\_DURATION : TIME := 5000 MS;  -- Common TEST duration

  CONSTANT TEST\_DELAY : TIME := 1000 MS;     -- RESET duration

BEGIN

   UUT: TopLevel PORT MAP(

    CLOCK => CLOCK,

    MODE => MODE,

    SPEED => SPEED,

    RESET => RESET,

    OUT\_BUS => OUT\_BUS

   );

  CLOCK\_PROCESS: PROCESS

  BEGIN

    CLOCK <= '0';

    WAIT FOR PERIOD / 2;

    CLOCK <= '1';

    WAIT FOR PERIOD / 2;

  END PROCESS;

   MAIN\_PROCESS : PROCESS

   BEGIN

    -- Reset beofre start

    RESET <= '1';

    WAIT FOR 100 ms;

    -- TEST 1: Direct MODE, x1 SPEED

    MODE <= '0';

    SPEED <= '0';

    RESET <= '0';

    WAIT FOR TEST\_DURATION;

    RESET <= '1';

    WAIT FOR TEST\_DELAY;

    -- TEST 2: Reverse MODE, x1 SPEED

    RESET <= '0';

    MODE <= '1';

    WAIT FOR TEST\_DURATION;

    RESET <= '1';

    MODE <= '0';

    WAIT FOR TEST\_DELAY;

    -- TEST 3: Direct MODE, x4 SPEED

    RESET <= '0';

    SPEED <= '1';

    WAIT FOR TEST\_DURATION;

    RESET <= '1';

    WAIT FOR TEST\_DELAY;

    -- TEST 4: Reverse MODE, x4 SPEED

    RESET <= '0';

    MODE <= '1';

    WAIT FOR TEST\_DURATION;

   END PROCESS;

END;

A screen shot of a computer

Description automatically generated with low confidence

Рис. 3. Результат запуску testbench

Як видно з результатів, при Mode = 0 сигнали йдуть з ліва на право, а при Mode = 1 – з права на ліво.

Також, при Speed = 1 сигнали переключаються в 4 рази швидше, ніж при Speed = 0.

**Висновок:** я реалізував цифровий автомат світлових ефектів.