**Міністерство освіти і науки України**

**Національний університет “Запорізька політехніка”**

*кафедра програмних засобів*

**ЗВІТ**

З лабораторної роботи № 6

з дисципліни “**ВЕРИФІКАЦІЯ ЦИФРОВИХ СИСТЕМ**”

на тему: “ РОБОТА З ПЛАТОЮ ШВИДКОГО ПРОТОТИПУВАННЯ”

Варіант № 22

Виконав:

Студент групи КНТ-217 О.І. Шавалда

Прийняв:

к.т.н, доцент Т. І. Каплієнко

м. Запоріжжя

2019

**1 Хід роботи**

**Завдання 1:**

Розробити тест-кейс для плати для швидкого прототипування, заснованої на MAX® V - 5M1270Z CPLD Altera, і модифікувати проект test1 відповідно до завдання: змінити номер функціонального перемикача і кількість світлодіодів, що світяться, відповідно до номеру варіанту.

**Тест-кейс test1:**

Унікальний ідентифікатор варіанти тестування – test1. Короткий опис варіанта тестування – самий правий перемикач sw[0] буде вмикати / вимикати самий правий світлодіод ld[0].

Порядок виконання - ввімкнути плату, ввімкнути правий перемикач.

Вимоги - тест завантажений на плату, плата підключена до комп'ютера.

Критерій завершеності - при включеному правом перемикачі світиться правий світлодіод, при вимкненому - не світиться.

**Код тест-кейсу test1:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port( pSW: in std\_logic\_vector(7 downto 0);

pLED: out std\_logic\_vector(7 downto 0);

pLED1: out std\_logic\_vector(7 downto 0)

);

end ent;

architecture firstTask of ent is

begin

PLED(0) <= pSW(0);

pLED(1) <= pSW(0);

end architecture;

**Код завдання:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port( pSW: in std\_logic\_vector(7 downto 0);

pLED: out std\_logic\_vector(7 downto 0);

pLED1: out std\_logic\_vector(7 downto 0)

);

end ent;

architecture firstTask of ent is

begin

pLED(4) <= pSW(5);

pLED(5) <= pSW(5);

pLED(6) <= pSW(5);

pLED(7) <= pSW(5);

pLED1(0) <= pSW(5);

pLED1(1) <= pSW(5);

pLED1(2) <= pSW(5);

pLED1(3) <= pSW(5);

pLED1(4) <= pSW(5);

pLED1(5) <= pSW(5);

pLED1(6) <= pSW(5);

pLED1(7) <= pSW(5);

end architecture;

**Результати виконання:**

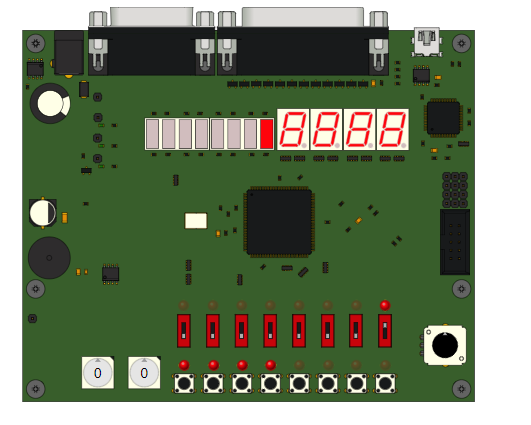


Рисунок 1.1 – Результат виконання тест-кейсу test1

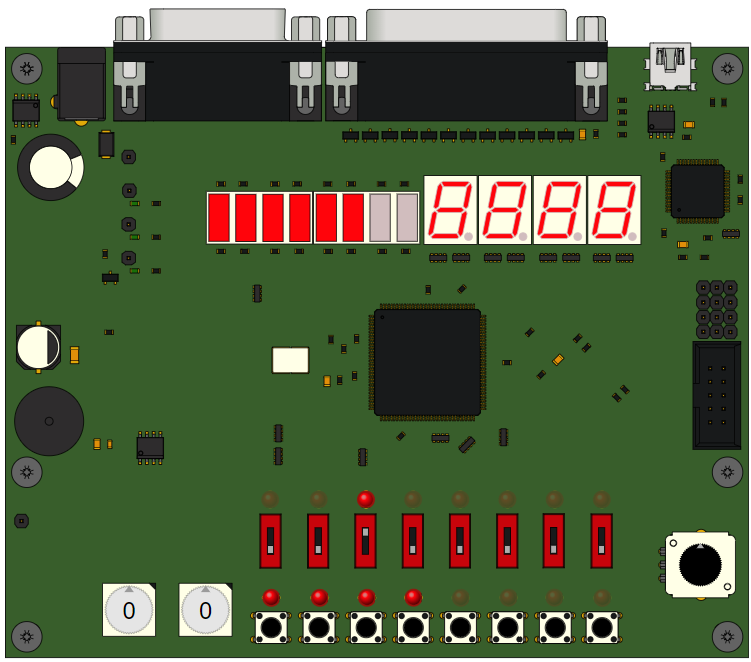


Рисунок 1.2 – Результат виконання завдання 1

**Завдання 2:**

Розробити тест-кейс для плати для швидкого прототипування, заснованої на MAX® V - 5M1270Z CPLD Altera, і модифікувати проект test2 відповідно до завдання: змінити вхідні і вихідні параметри відповідно до таблиці.

Таблиця вхідних і вихідних параметрів:



**Тест-кейс test2:**

Унікальний ідентифікатор варіанти тестування – test2.

Короткий опис варіанта тестування - значення важелів sw[0] и sw[1] в якості значень вхідних параметрів модуля basic\_boolean a і b, а світлодіоди ld [0] ... ld [4] будуть показувати значення результатів роботи модуля basic\_boolean not\_a, not\_b, a\_and\_b, a\_or\_b, a\_nand\_b (0 = FALSE = викл або 1 = TRUE = вкл).

Порядок виконання - включити плату, поперемінно включати і вимикати в різних комбінаціях два правих перемикача.

Вимоги - тест завантажений на плату, плата підключена до комп'ютера.

Критерій завершеності - при включеному перемикачі sw [0] і вимкненому sw [1] світиться перший, третій і четвертий світлодіод; при вимкненому перемикачі sw [0] і включеному sw [1] світиться нульовий, третій і четвертий світлодіод; при вимкненому перемикачі sw [0] і вимкненому sw [1] світиться нульовий, перший і четвертий світлодіод; при включеному перемикачі sw [0] і включеному sw [1] світиться другий і третій світлодіод.

**Код тест-кейсу test2:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port(

pSW: in std\_logic\_vector(7 downto 0);

pLED: out std\_logic\_vector(7 downto 0);

pLED1: out std\_logic\_vector(7 downto 0)

);

end ent;

architecture firstTask of ent is

procedure basic\_boolean (

a: in std\_logic; b: in std\_logic

) is

variable notA: std\_logic := NOT a;

variable notB: std\_logic := NOT b;

variable aAndB: std\_logic := a AND b;

variable aOrB: std\_logic := a OR b;

variable aNandB: std\_logic := a NAND b;

begin

pLED(0) <= notA;

pLED(1) <= notA;

pLED(2) <= notB;

pLED(3) <= notB;

pLED(4) <= aAndB;

pLED(5) <= aAndB;

pLED(6) <= aOrB;

pLED(7) <= aOrB

pLED1(0) <= aNandB;

pLED1(1) <= aNandB;

end basic\_boolean;

begin

basic\_boolean(pSW(0), pSW(1));

end architecture;

**Код завдання:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port( pSW: in std\_logic\_vector(7 downto 0);

pLED: out std\_logic\_vector(7 downto 0);

pLED1: out std\_logic\_vector(7 downto 0)

);

end ent;

architecture firstTask of ent is

procedure basic\_boolean (

switcher: in std\_logic\_vector(7 downto 0)

) is

variable firstAndThird: std\_logic := switcher(1) AND switcher(3);

variable notFirst: std\_logic := NOT switcher(1);

variable notThird: std\_logic := NOT switcher(3);

begin

pLED(0) <= firstAndThird;

pLED(1) <= firstAndThird;

pLED(2) <= notFirst;

pLED(3) <= notFirst;

pLED(4) <= notFirst;

pLED(5) <= notFirst;

pLED(6) <= notFirst;

pLED(7) <= notFirst;

pLED1(0) <= notThird;

pLED1(1) <= notThird;

pLED1(2) <= firstAndThird;

pLED1(3) <= firstAndThird;

pLED1(4) <= notThird;

pLED1(5) <= notThird;

pLED1(6) <= firstAndThird;

pLED1(7) <= firstAndThird;

end basic\_boolean;

begin

basic\_boolean(pSW);

end architecture;

**Результати виконання:**

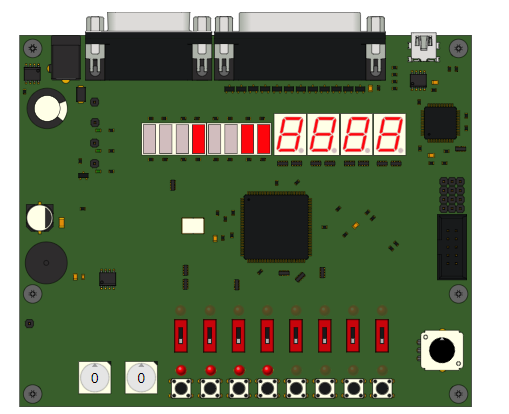


Рисунок 1.3 – Результат виконання тест-кейсу test2

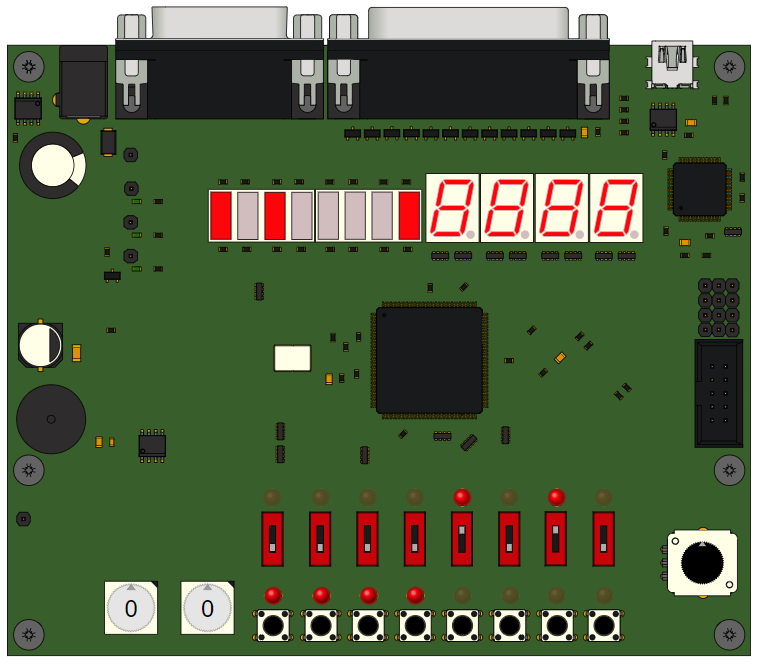


Рисунок 1.4 – Результат виконання завдання 2

**Завдання 3:**

Розробити тест-кейс для плати для швидкого прототипування, заснованої на MAX® V - 5M1270Z CPLD Altera, і модифікувати проект test3 відповідно до завдання: змінити першу виведену цифру на номер варіанту, другу - на число, що дорівнює номеру варіанту плюс 1.

**Тест-кейс test3:**

Унікальний ідентифікатор варіанти тестування – test3.

Короткий опис варіанта тестування - при включеному стані самого правого перемикача sw [0] буде світитися другий і четвертий цифровий дисплей з цифрою «7», при вимкненому - буде світитися перший і третій цифровий дисплей з цифрою «5».

Порядок виконання - включити плату, включити правий перемикач.

Вимоги - тест завантажений на плату, плата підключена до комп'ютера.

Критерій завершеності - при включеному стані самого правого перемикача sw [0] світиться другий і четвертий цифровий дисплей з цифрою «7», при вимкненому - перший і третій цифровий дисплей з цифрою «5».

**Код тест-кейсу test3:**

library IEEE;

entity secondTask is

port( pSW: in std\_logic\_vector(7 downto 0);

pLCD0: out std\_logic\_vector(7 downto 0);

pLCD1: out std\_logic\_vector(7 downto 0);

pLCD2: out std\_logic\_vector(7 downto 0);

pLCD3: out std\_logic\_vector(7 downto 0)

);

end secondTask;

architecture numbers of secondTask is

begin

nums: process(pSW) IS

variable offCode: std\_logic\_vector(7 downto 0) := "11111111";

variable fiveCode: std\_logic\_vector(7 downto 0) := "10010010";

variable sevenCode: std\_logic\_vector(7 downto 0) := "11111000";

begin

if (pSW(0) = '1') then

pLCD0 <= offCode; pLCD1 <= sevenCode;

pLCD2 <= offCode; pLCD3 <= sevenCode;

else

pLCD0 <= fiveCode; pLCD1 <= offCode;

pLCD2 <= fiveCode; pLCD3 <= offCode;

end if;

end process nums;

end numbers;

**Код завдання:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port(

pSW: in std\_logic\_vector(7 downto 0);

pLCD0: out std\_logic\_vector(7 downto 0);

pLCD1: out std\_logic\_vector(7 downto 0);

pLCD2: out std\_logic\_vector(7 downto 0);

pLCD3: out std\_logic\_vector(7 downto 0);

pLCD4: out std\_logic\_vector(7 downto 0)

);

end ent;

architecture numbers of ent is

begin

nums: process(pSW) IS

variable offCode: std\_logic\_vector(7 downto 0) := "11111111";

variable twoCode: std\_logic\_vector(7 downto 0) := "10100100";

variable threeCode: std\_logic\_vector(7 downto 0) := "10110000";

begin

if (pSW(0) = '1') then

pLCD1 <= offCode;

pLCD2 <= offCode;

pLCD3 <= twoCode;

pLCD4 <= twoCode;

else

pLCD1 <= threeCode;

pLCD2 <= twoCode;

pLCD3 <= offCode;

pLCD4 <= offCode;

end if;

end process nums;

end numbers;

**Результати виконання:**

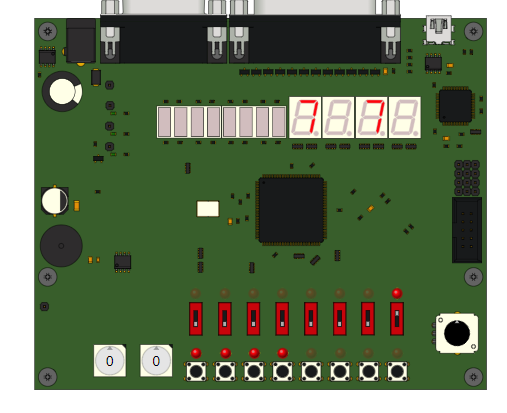
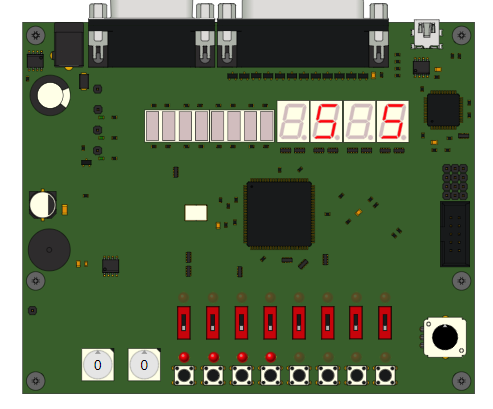
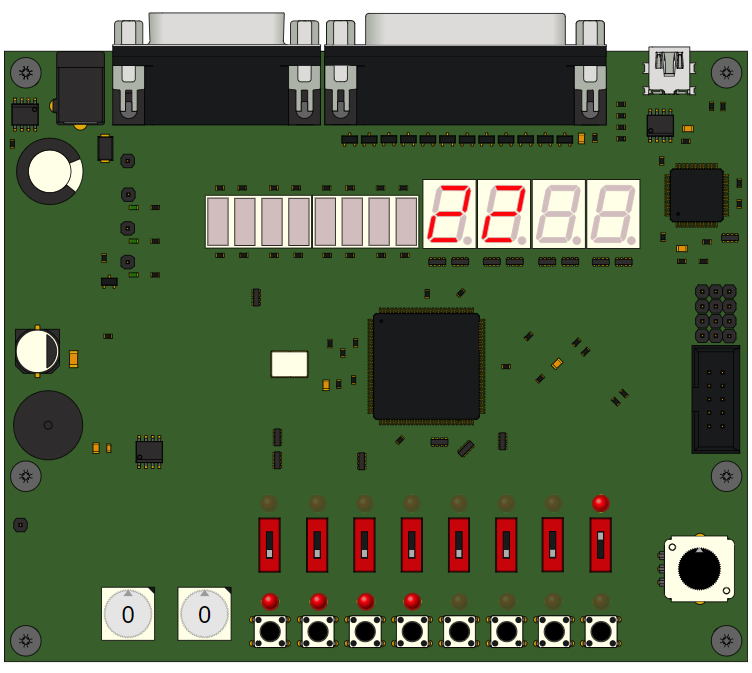
 

Рисунок 1.5 – Результат виконання тест-кейсу test3



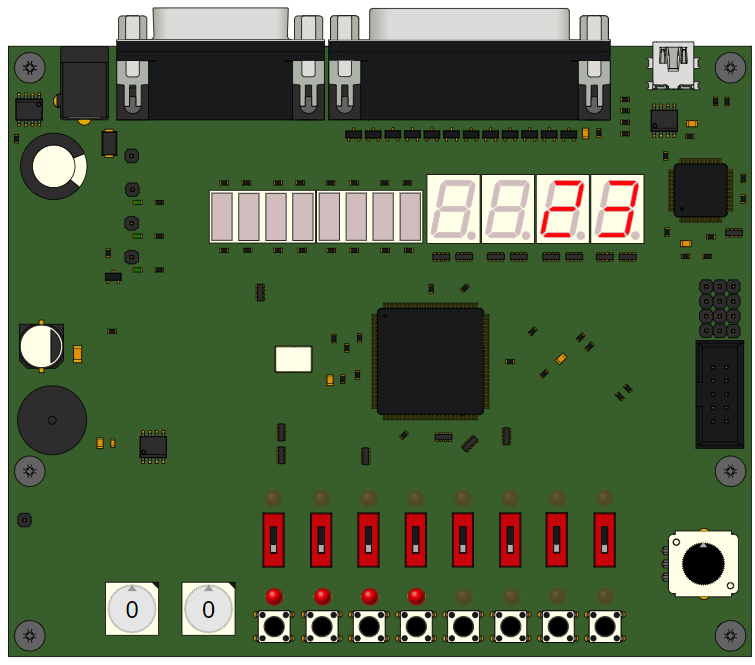


Рисунок 1.6 – Результат виконання завдання 3

**Завдання 4:**

Розробити тест-кейс для плати для швидкого прототипування, заснованої на MAX® V - 5M1270Z CPLD Altera, і модифікувати проект test4 відповідно до завдання: змінити номер миготливого світлодіода відповідно до варіанта, і частоту миготіння на 1 / варіант сек.

**Тест-кейс test4:**

Унікальний ідентифікатор варіанти тестування – test4.

Короткий опис варіанта тестування - необхідно вмикати / вимикати самий правий світлодіод ld [0] з частотою раз в секунду.

Порядок виконання - включити плату.

Вимоги - тест завантажений на плату, плата підключена до комп'ютера.

Критерій завершеності - при включенні плати самий правий світлодіод ld [0] загоряється з частотою раз в секунду.

**Код завдання:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port(

pSW: in std\_logic\_vector(7 downto 0);

pLED: out std\_logic\_vector(7 downto 0);

pLED1: out std\_logic\_vector(7 downto 0);

clk: in std\_logic

);

end ent;

architecture firstTask of ent is

signal ticks: integer := 0;

signal currentSignal: std\_logic := '0';

begin

process is

variable freq: integer := 21e6 / 22;

begin

wait until rising\_edge(clk);

ticks <= ticks + 1;

if ticks > freq then

ticks <= 0;

currentSignal <= not currentSignal;

pLED1(2) <= currentSignal;

pLED1(3) <= currentSignal;

end if;

end process;

end architecture;

**Результати виконання:**

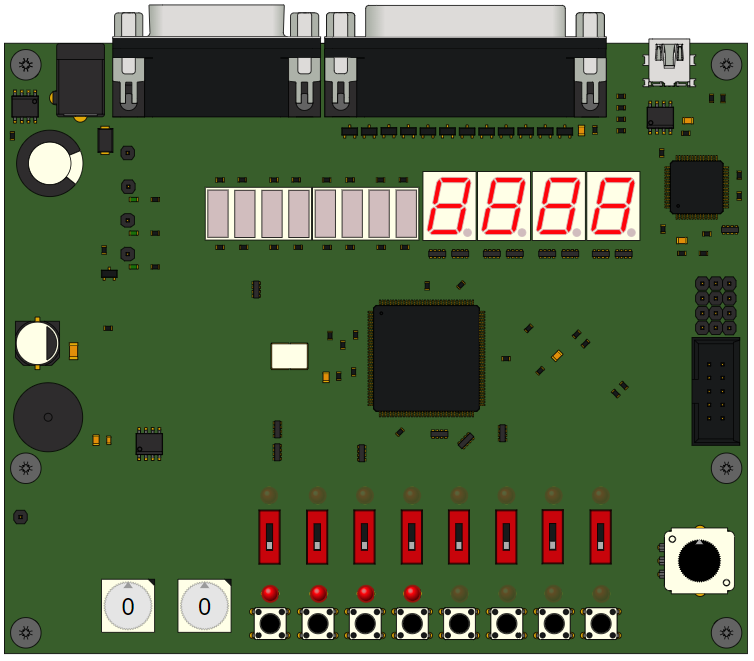
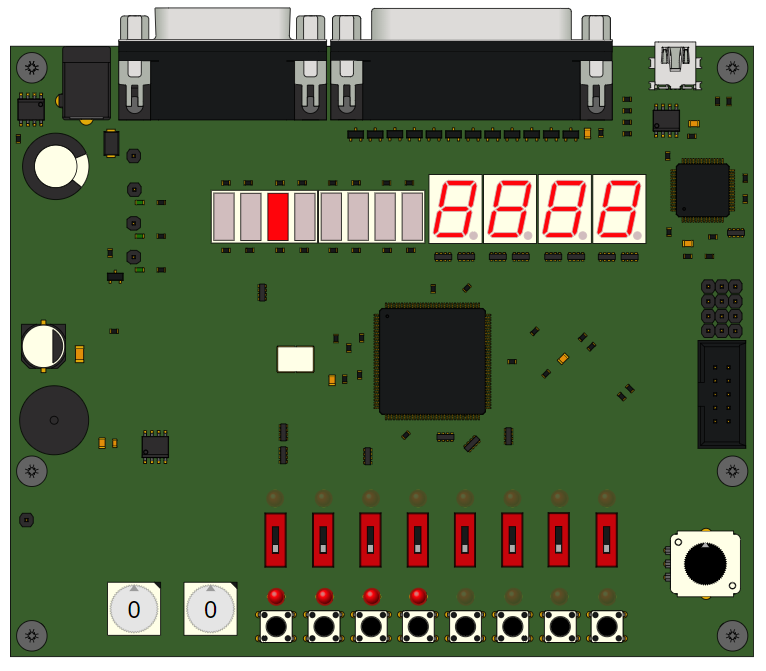
 

Рисунок 1.7 – Результат виконання завдання 4

**Завдання 5:**

Розробити тест-кейс для плати для швидкого прототипування, заснованої на MAX® V - 5M1270Z CPLD Altera, (на ваш розсуд).

**Тест-кейс test5:**

Унікальний ідентифікатор варіанти тестування – test5.

Короткий опис варіанта тестування – увімкнений стан перемикача sw[0] починає роботу циклічного перемикання світлодіодів починаючи з тих, що по краях зходячись до центру. Якщо перемикач у вимкненому стані – робота зупиняється. Кількість ввімкнених світлодіодів показує поточне число на першому індикатор (від 1 до 8).

Порядок виконання - включити плату, включити перший перемикач.

Вимоги - тест завантажений на плату, плата підключена до комп'ютера.

Критерій завершеності – при включеному першому перемикачу запускається зміна числа на індикаторах (від 0), при вимкненому – призупиняється; згідно цифрі на першому індикаторі запалюється відповідна кількість світлодіодів.

**Код завдання:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ent is

port(

pSW: in std\_logic\_vector(7 downto 0); clk: in std\_logic;

pLED: out std\_logic\_vector(7 downto 0); pLED1: out std\_logic\_vector(7 downto 0);

pLCD1: out std\_logic\_vector(7 downto 0) := "11000000";

pLCD2: out std\_logic\_vector(7 downto 0) := "11000000";

pLCD3: out std\_logic\_vector(7 downto 0) := "11000000";

pLCD4: out std\_logic\_vector(7 downto 0) := "11000000"

);

end ent;

architecture firstTask of ent is

type num is array(0 to 9) of std\_logic\_vector(7 downto 0);

type led is array(0 to 4) of std\_logic\_vector(7 downto 0);

type lcd is array(0 to 3) of integer;

signal ticks: integer := 0;

begin

process is

variable freq: integer := 21e6; -- 1 sec

variable state: integer := 0;

variable currentNumber: integer := 0;

variable lcdValue: lcd := (0, 0, 0, 0);

variable numbers: num := ("11000000", "11111001", "10100100", "10110000", "10011001", "10010010", "10000010", "11111000", "10000000", "10010000");

begin

wait until rising\_edge(clk);

if pSW(0) = '1' then

ticks <= ticks + 1; -- pause

end if;

if ticks > freq then

ticks <= 0;

case state is

when 0 => pLED(0) <= '1';pLED(1) <= '1';pLED1(6) <= '1';pLED1(7) <= '1';

when 1 => pLED(2) <= '1';pLED(3) <= '1';pLED1(4) <= '1';pLED1(5) <= '1';

when 2 => pLED(4) <= '1';pLED(5) <= '1';pLED1(2) <= '1';pLED1(3) <= '1';

when 3 => pLED(6) <= '1';pLED(7) <= '1';pLED1(0) <= '1';pLED1(1) <= '1';

when others => pLED <= "00000000"; pLED1 <= "00000000";

end case;

if (state = 4) then

state := 0;

currentNumber := currentNumber + 1;

if currentNumber > 9 then

lcdValue(1) := lcdValue(1) + 1; currentNumber := 0;

end if;

if lcdValue(1) > 9 then

lcdValue(1) := 0; lcdValue(2) := lcdValue(2) + 1;

end if;

if lcdValue(2) > 9 then

lcdValue(2) := 0; lcdValue(3) := lcdValue(3) + 1;

end if;

if lcdValue(3) > 9 then

lcdValue(3) := 0;

end if;

pLCD1 <= numbers(currentNumber);

pLCD2 <= numbers(lcdValue(1));

pLCD3 <= numbers(lcdValue(2));

pLCD4 <= numbers(lcdValue(3));

else

state := state + 1;

end if;

end if;

end process;

end architecture;

**Результати виконання:**

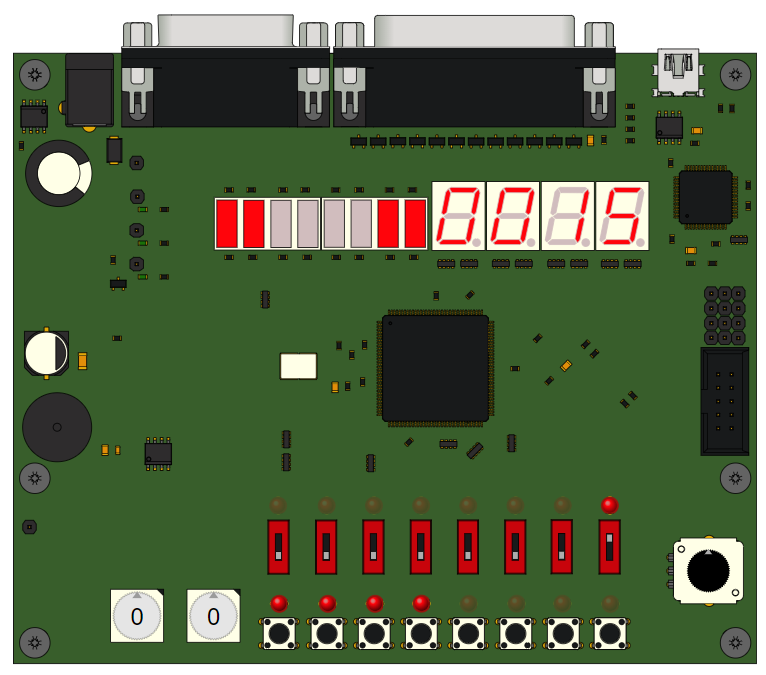
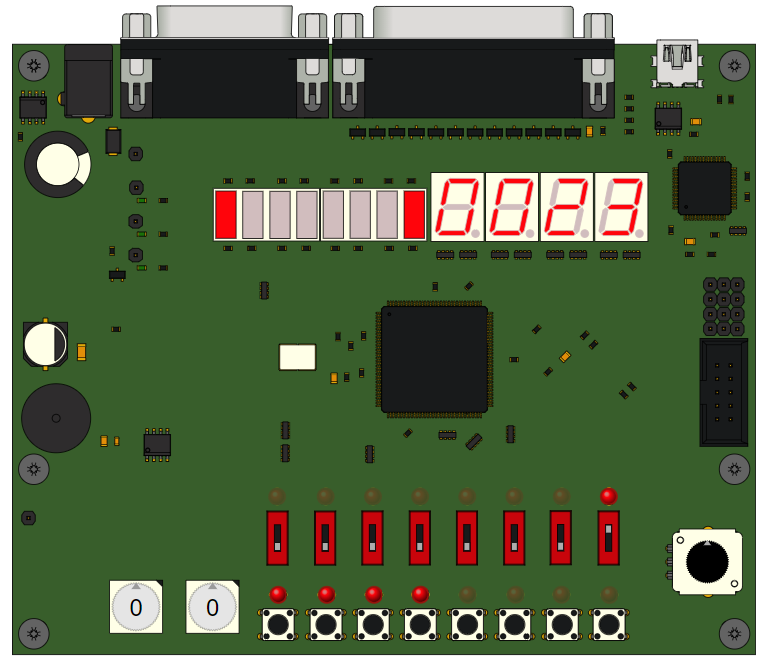
 

Рисунок 1.8 – Результат виконання завдання 5