Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ

Звіт

З лабораторної роботи № 2, варіант 1

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

Абросімов А.С.

Перевірив: викладач

Козак Н.Б.

Львів – 2023

Мета роботи: На базі стенду Elbert V2 – Spartan 3A FPGA, реалізувати цифровий автомат світлових ефектів відповідно до наступних вимог:

1. Інтерфейс пристрою та функціонал реалізувати згідно отриманого варіанту завдання.

2. Логіку переходів реалізувати з використанням мови опису апаратних засобів.

3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів.

4. Згенерувати символи для описів логіки переходів та логіки формування вихідних сигналів.

5. Інтегрувати всі компоненти, логіку переходів, логіку формування вихідних сигналів та пам'ять станів в єдину систему за допомогою ISE WebPack. Пам'ять станів реалізувати за допомогою графічних компонентів з бібліотеки

6. Промоделювати роботу окремих частин автомата та автомата в цілому за допомогою симулятора iSim.

7. Інтегрувати створений автомат зі стендом Elbert V2 – Spartan 3A FPGA.

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 нв 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 виконання

1. Створюю новий проект та додаю до нього новий VHDL файл, в якому буду реалізовувати логіку формування вихідних сигналів.

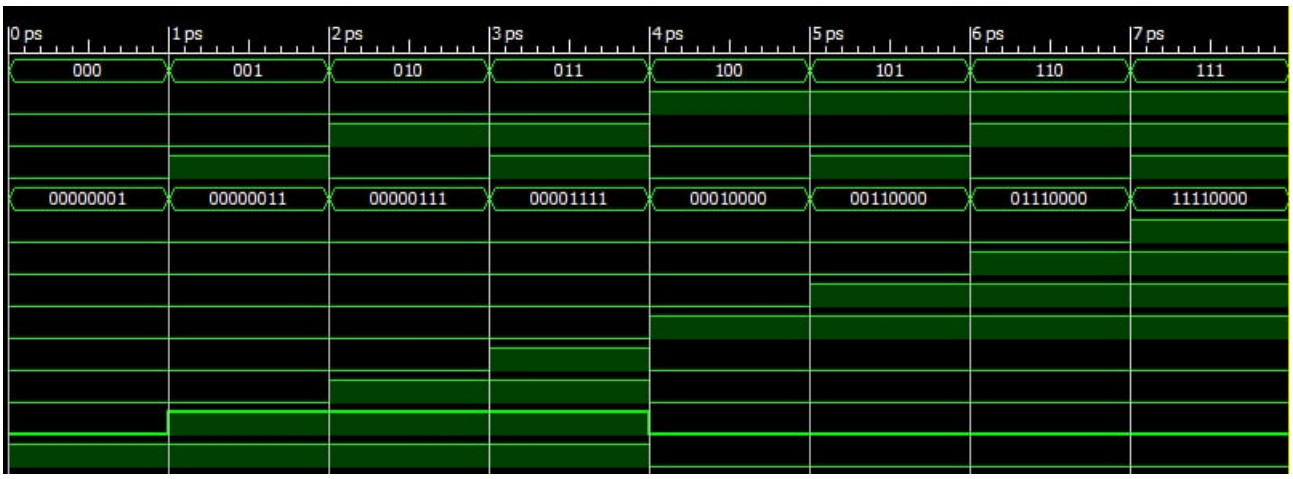
2. У створеному файлі із назвою OutputLogic.vhd імплементую інтерфейс логіки кожного вихідного сигналу, а також логічні вирази для формування кожного вихідного сигналу відповідно до варіанту.

Лістинг файлу OutputLogic.vhd:

```
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity out_logic_intf is
33     Port( IN_BUS : in std_logic_vector(2 downto 0);
34           OUT_BUS : out std_logic_vector(7 downto 0)
35           );
36  end out_logic_intf;
37
38  architecture out_logic_arch of out_logic_intf is
39
40  begin
41
42     OUT_BUS(0) <= not IN_BUS(2);
43
44     OUT_BUS(1) <= not IN_BUS(2) and (IN_BUS(1) or IN_BUS(0));
45
46     OUT_BUS(2) <= not IN_BUS(2) and IN_BUS(1);
47
48     OUT_BUS(3) <= not IN_BUS(2) and IN_BUS(1) and IN_BUS(0);
49
50     OUT_BUS(4) <= IN_BUS(2);
51
52     OUT_BUS(5) <= IN_BUS(2) and (IN_BUS(1) or IN_BUS(0));
53
54     OUT_BUS(6) <= IN_BUS(2) and IN_BUS(1);
55
56     OUT_BUS(7) <= IN_BUS(2) and IN_BUS(1) and IN_BUS(0);
57
58  end out_logic_arch;
59
```

OutputLogic.vhd

3. Проводжу моделювання роботи схеми формування вихідних сигналів.



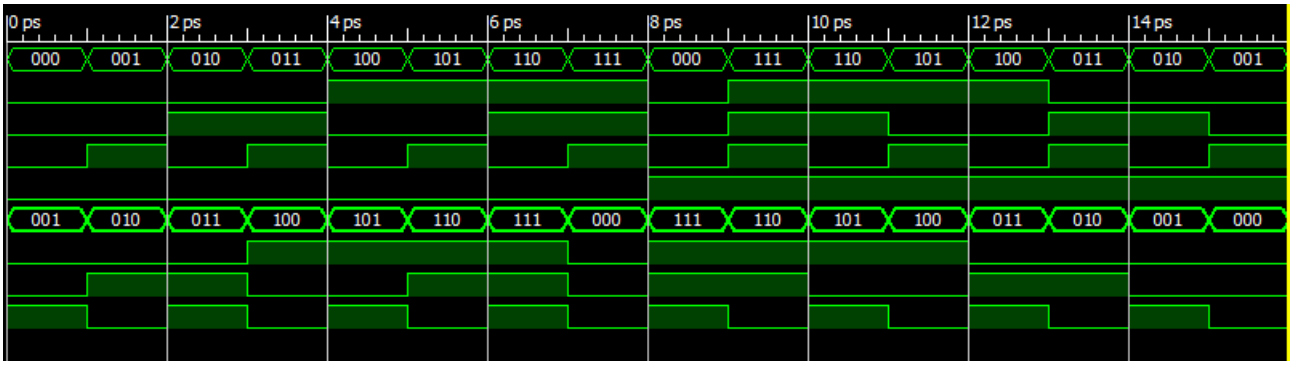
4. Аналогічно до пунктів 2 та 3 проводжу імплементацію та моделювання логіки формування перехідних сигналів (файл Transitionlogic.vhd).

Лістинг файлу TransitionLogic.vhd:

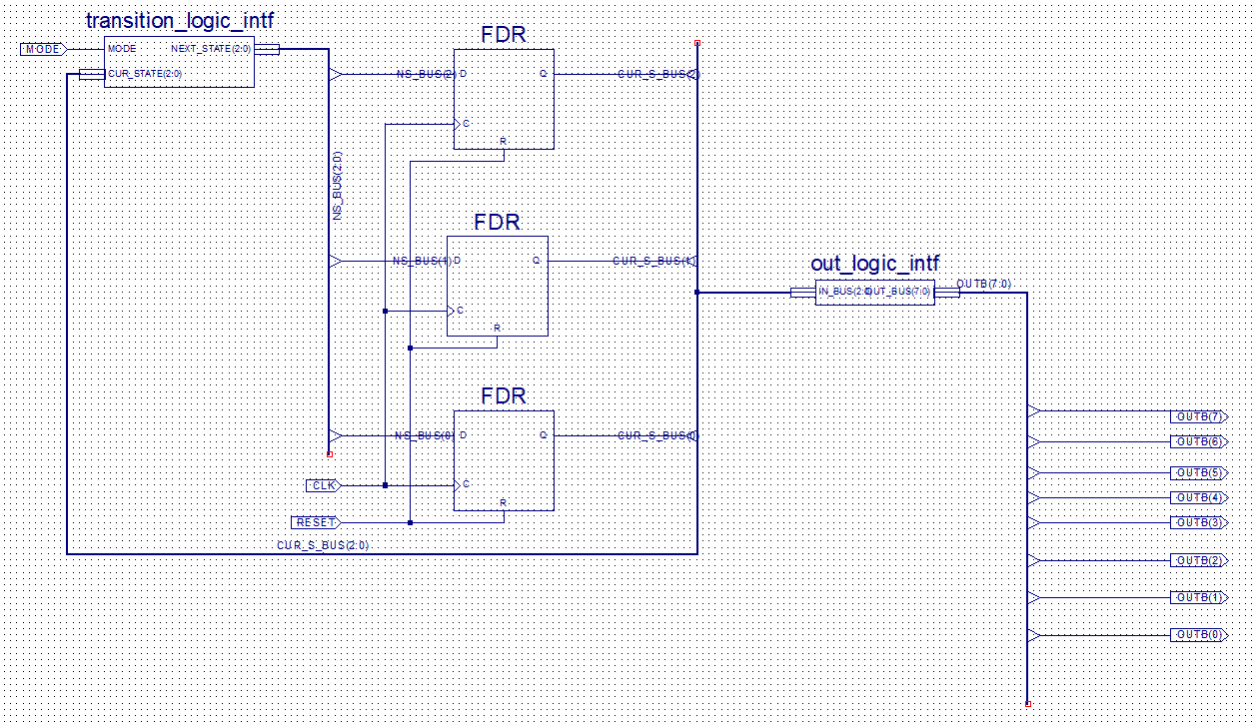
```
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity transition_logic_intf is
33     Port( CUR_STATE : in std_logic_vector (2 downto 0);
34           MODE : in std_logic;
35           NEXT_STATE : out std_logic_vector (2 downto 0)
36           );
37  end transition_logic_intf;
38
39  architecture transition_logic_arch of transition_logic_intf is
40
41  begin
42
43      NEXT_STATE(0) <= (not(MODE) and not(CUR_STATE(1)) and not (CUR_STATE(0))) or
44                       (not(MODE) and CUR_STATE(1) and not(CUR_STATE(0))) or
45                       (MODE and not(CUR_STATE(1)) and not (CUR_STATE(0))) or
46                       (MODE and CUR_STATE(1) and not (CUR_STATE(0)));
47
48      NEXT_STATE(1) <= (not(MODE) and not(CUR_STATE(1)) and CUR_STATE(0)) or
49                       (not(MODE) and CUR_STATE(1) and not (CUR_STATE(0))) or
50                       (MODE and not(CUR_STATE(1)) and not(CUR_STATE(0))) or
51                       (MODE and CUR_STATE(1) and CUR_STATE(0));
52
53      NEXT_STATE(2) <= ((not(MODE) and CUR_STATE(2)) and not (CUR_STATE(1) and CUR_STATE(0))) or
54                       ((not(MODE) and not (CUR_STATE(2))) and (CUR_STATE(1) and CUR_STATE(0))) or
55                       ((MODE and CUR_STATE(2)) and (CUR_STATE(1) or CUR_STATE(0))) or
56                       (MODE and not CUR_STATE(2) and not CUR_STATE(1) and not CUR_STATE(0));
57
58  end transition_logic_arch;
59
60
```

OutputLogic.vhd TransitionLogic.vhd

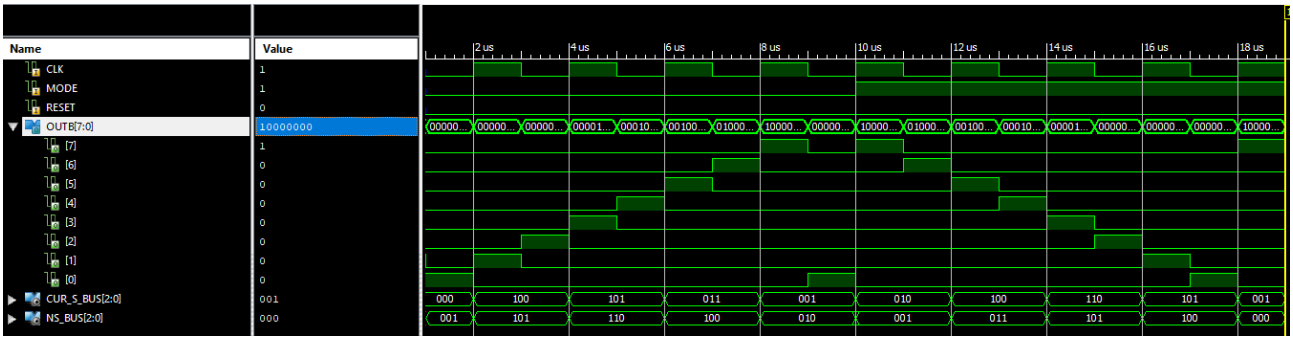
5. Промоделював роботу схеми формування вихідних сигналів з усіма можливими наборами сигналів.



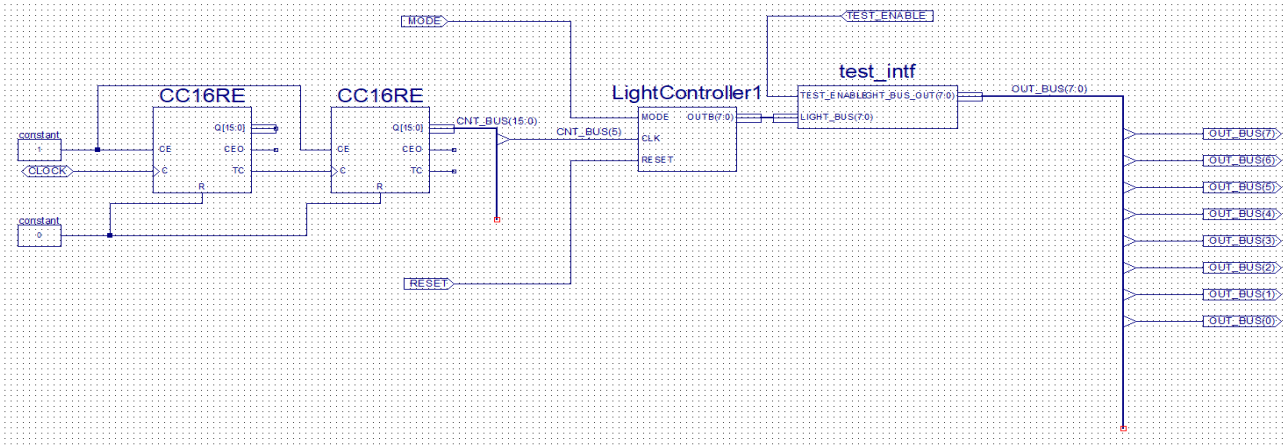
6. Генерую Schematic символи для файлів OutputLogic.vhd та Transition Logic.vhd. За допомогою згенерованих та інших стандартних символів у файлі LightController.sch реалізовую пам’ять стану автомата.



7. Симулюю роботу автомата.

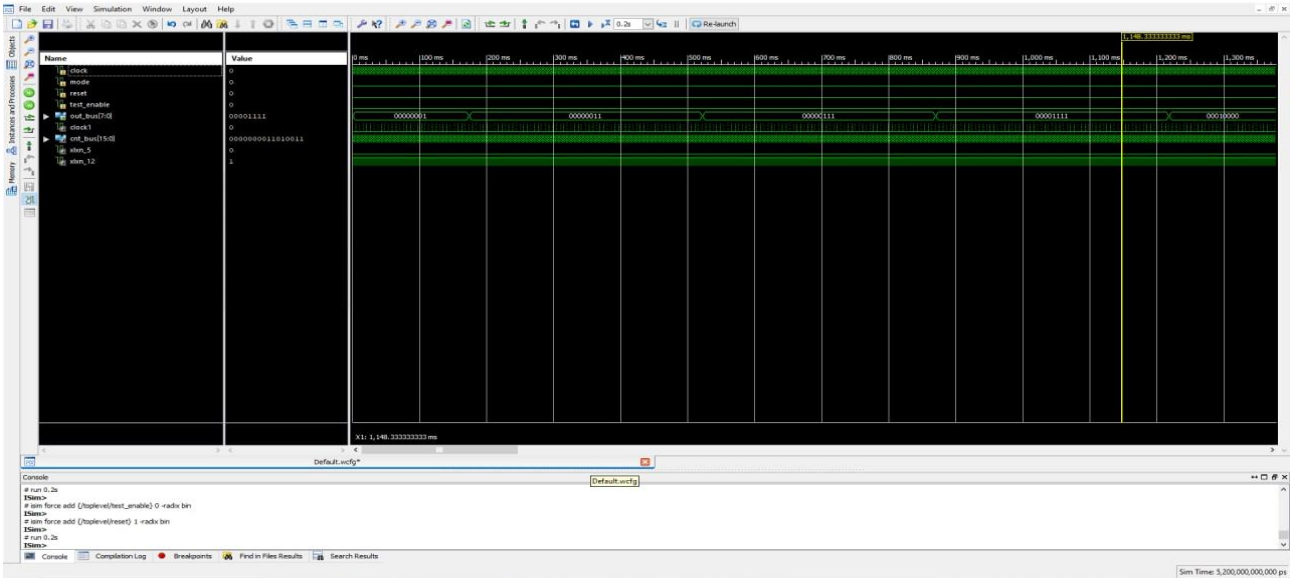


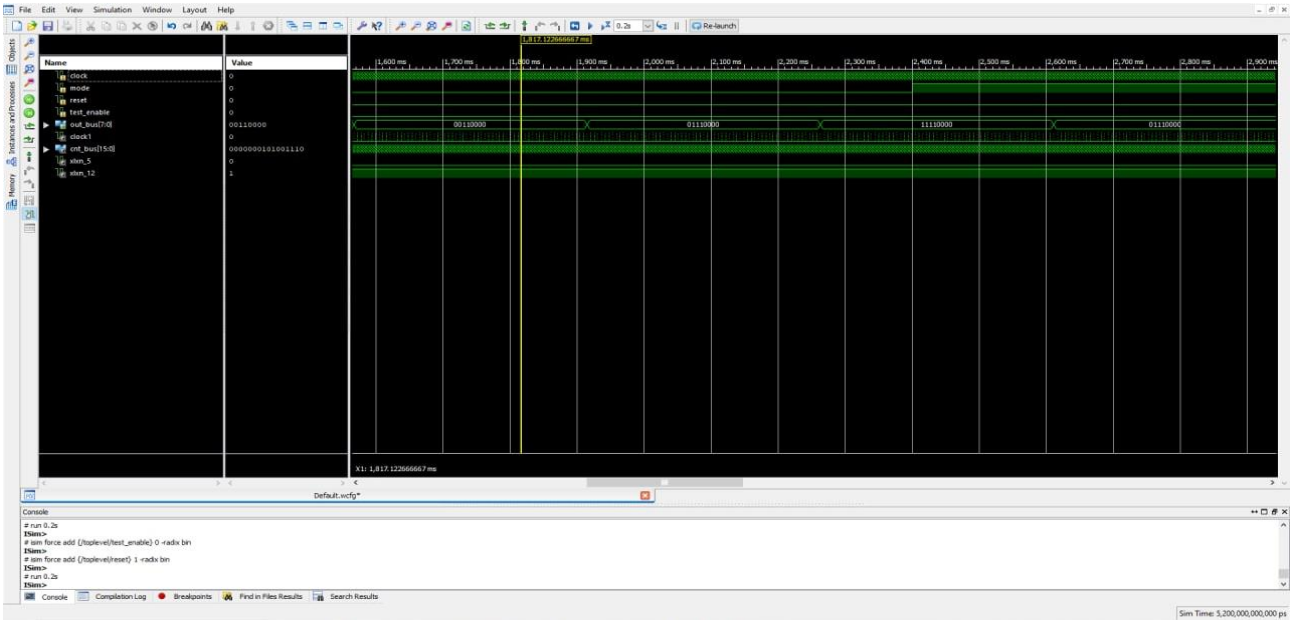
8. Генерую Schematic символи для файлу LightController.sch. За допомогою згенерованих та інших стандартних символів у файлі TopLevel.sch реалізовую подільник вхідної частоти.

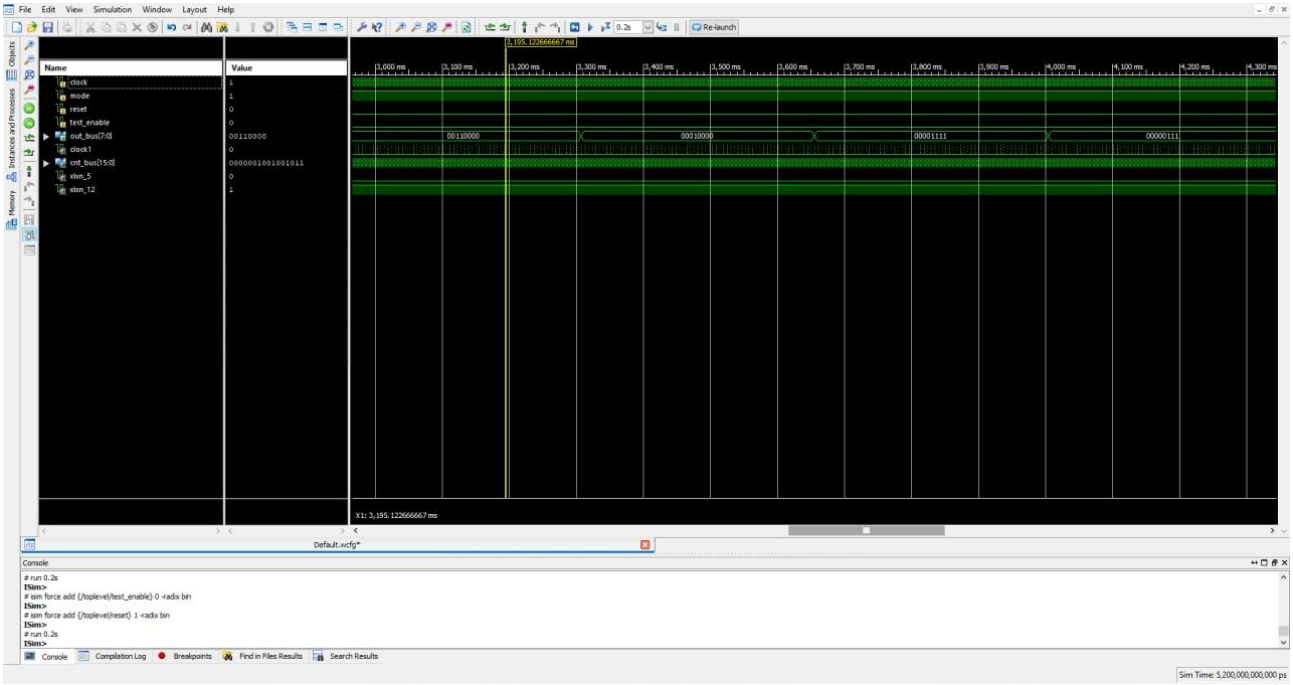


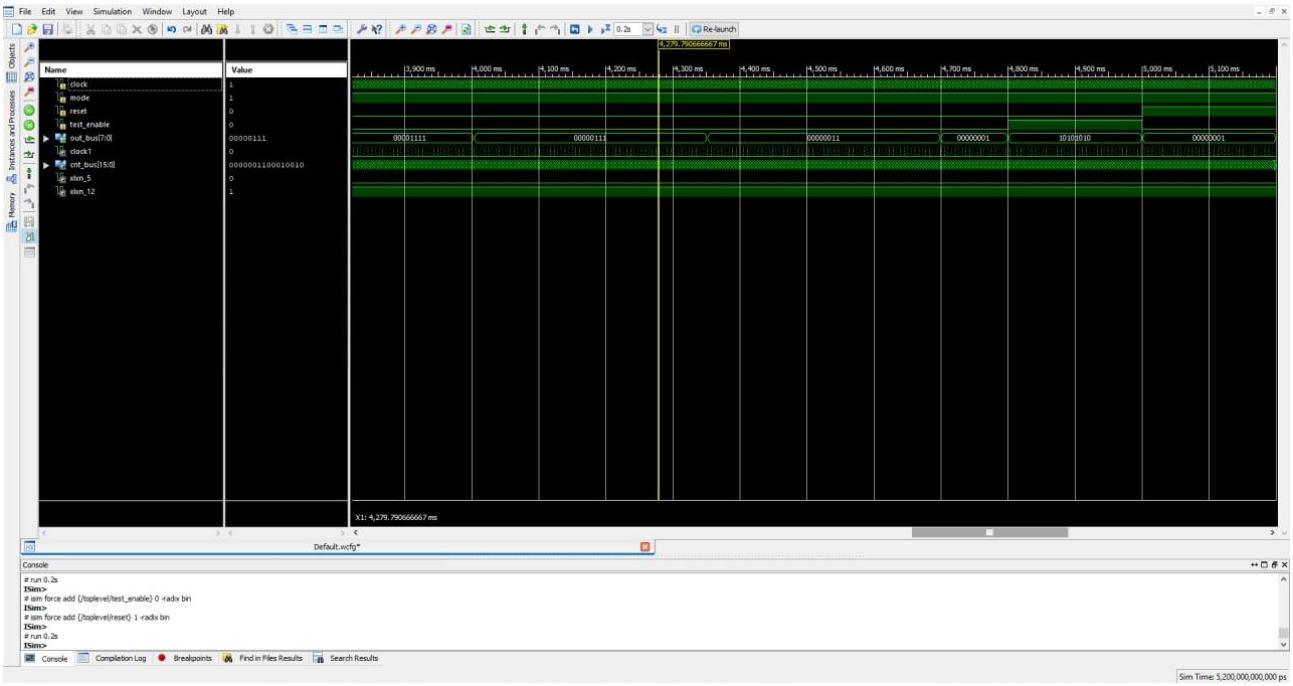
9. На вкладці Design менеджера проектів виконав команду Set as Top Module.

10. Змоделював роботу кінцевої схеми.







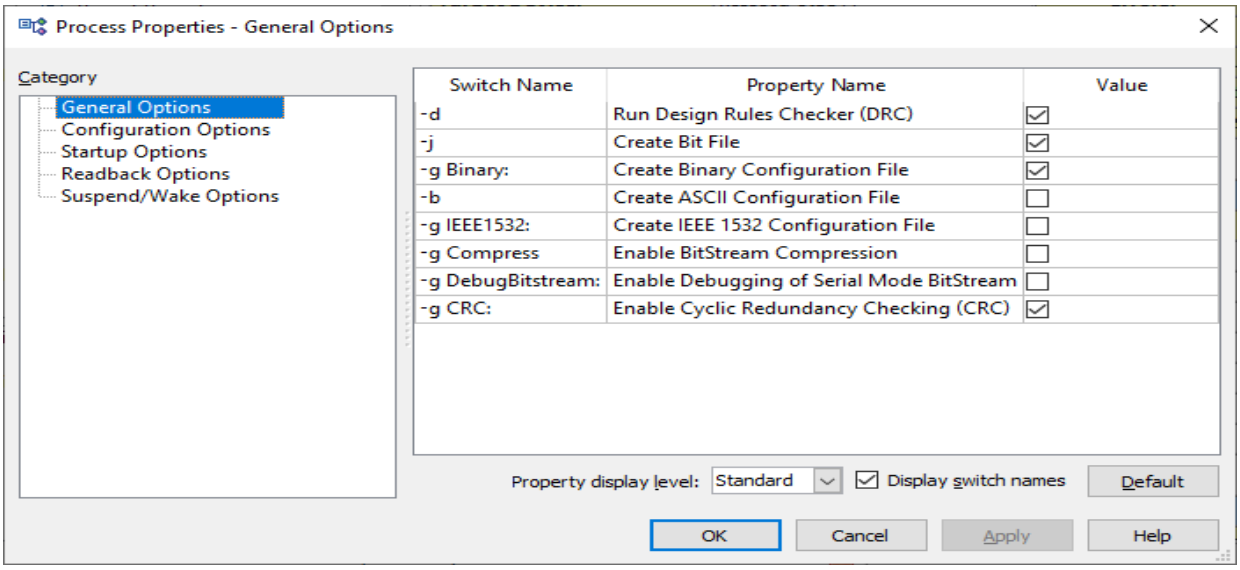


11. Додаю файл scheme.ucf та призначаю виводам схеми фізичні виводи цільової FPGA.

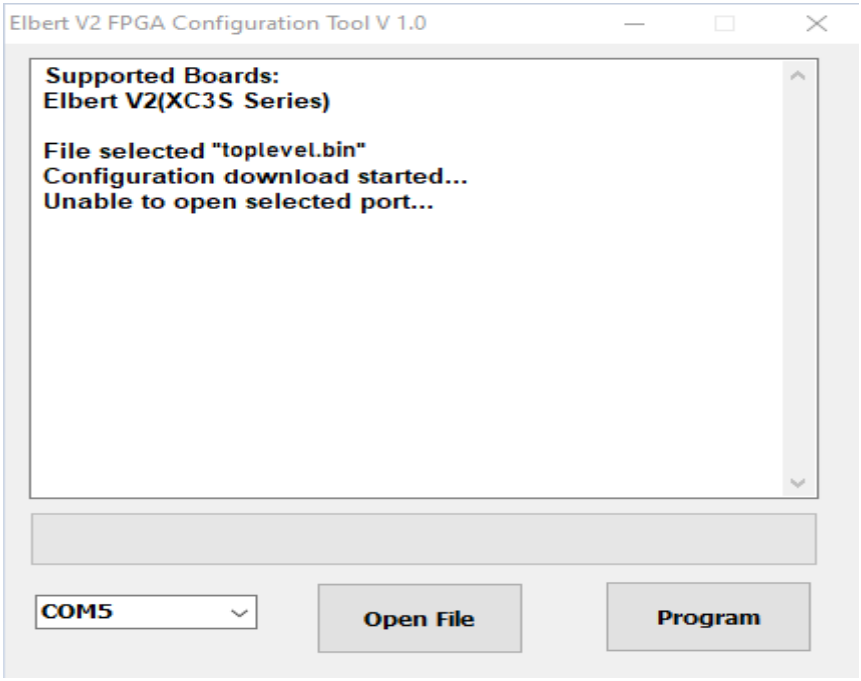
```
##############################################################################################

                                        LED

##############################################################################################

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

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

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

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

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

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

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

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


##############################################################################################

                              DP Switches

##############################################################################################


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

    NET "SPEED"             LOC = P69   | PULLUP  | IOSTANDARD = LVCMOS33 | SLEW =
SLOW | DRIVE = 12;

    NET "RESET"             LOC = P68   | PULLUP  | IOSTANDARD = LVCMOS33 | SLEW =
SLOW | DRIVE = 12;

    NET "CLK"                               LOC = P67   | PULLUP  | IOSTANDARD =
LVCMOS33 | SLEW = SLOW | DRIVE = 12;
```

12. Генерую BIT файл, який названий *toplevel.bin* для цільової FPGA. Для цього послідовно запускаю процеси Synthesize – XST, Implement Design та Generate Programming File.



13. Перевіряю роботу на стенді. Програмування стенду згенерованим *toplevel.bin* файлом.



Додаткове завдання.

Побудувати схему реалізації восьмирозрядного лічильника імпульсів за модулем ділення числа М. Модуль числа М задається у відповідності до вказаного варіанту завдання.

Модуль числа імпульсів М задається як десяткове число, що утворюється наступним чином за формулою:

$$\mathbf{M_{10} = k*N+128,}$$

де, N – порядковий номер у журнальному списку, а k – коефіцієнт пропорційності, який приймає наступні значення:

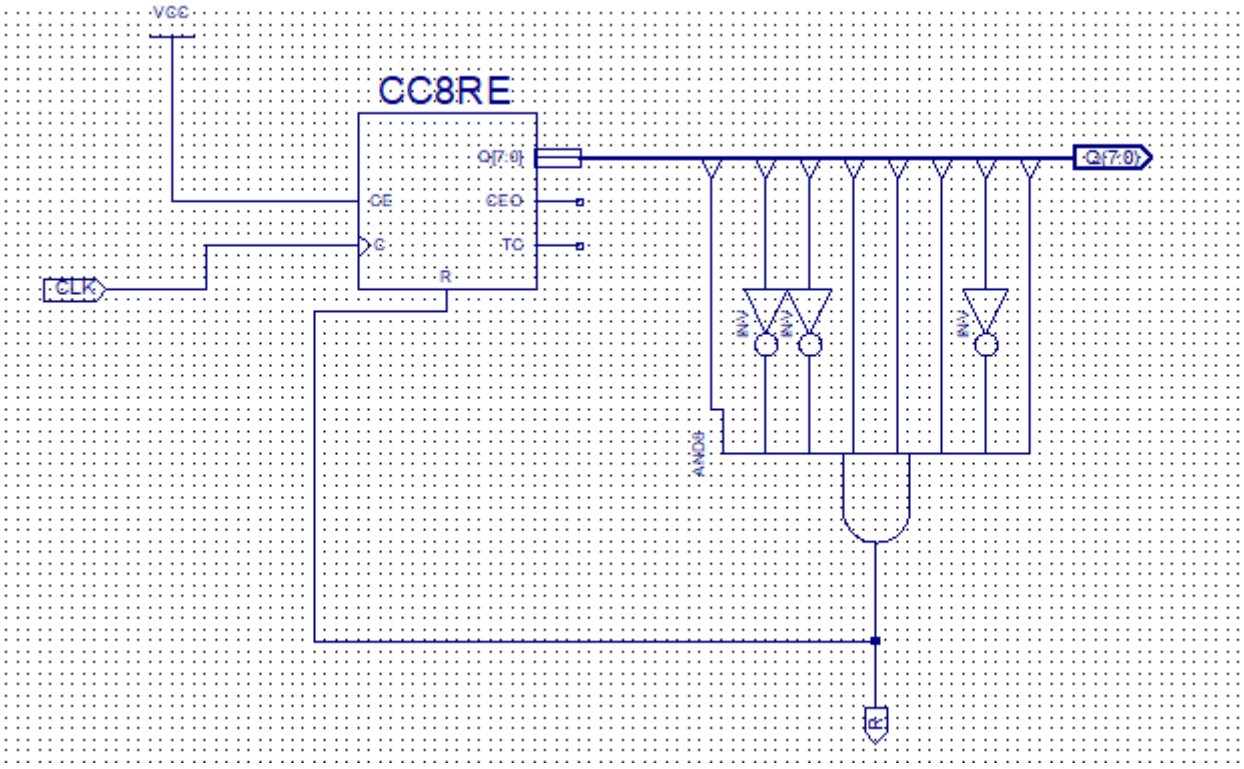
k = 5, якщо N = від 1 до 9.

Хід виконання

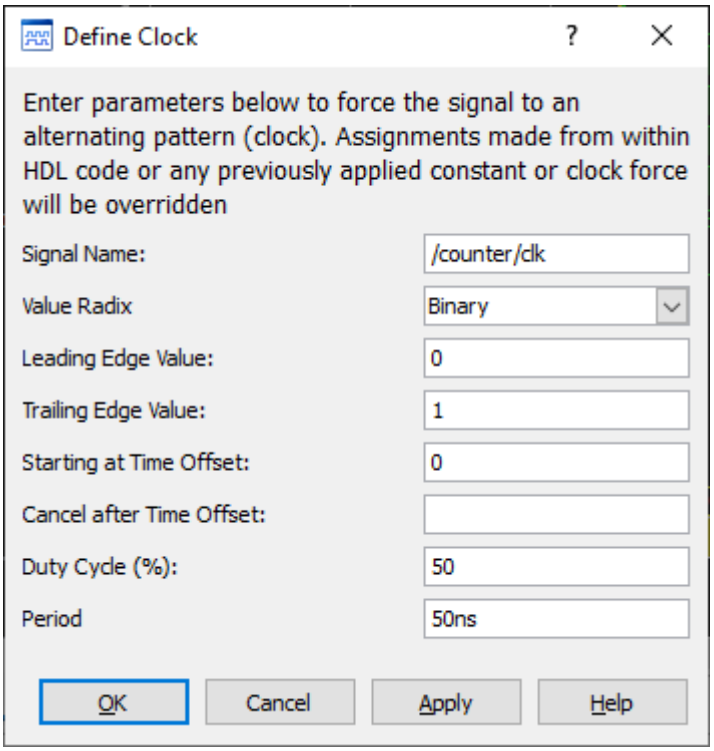
N = 6, k = 5

$$\mathbf{M_{10} = 5*6+128 = 158_{10} = 1001\ 1110_2}$$

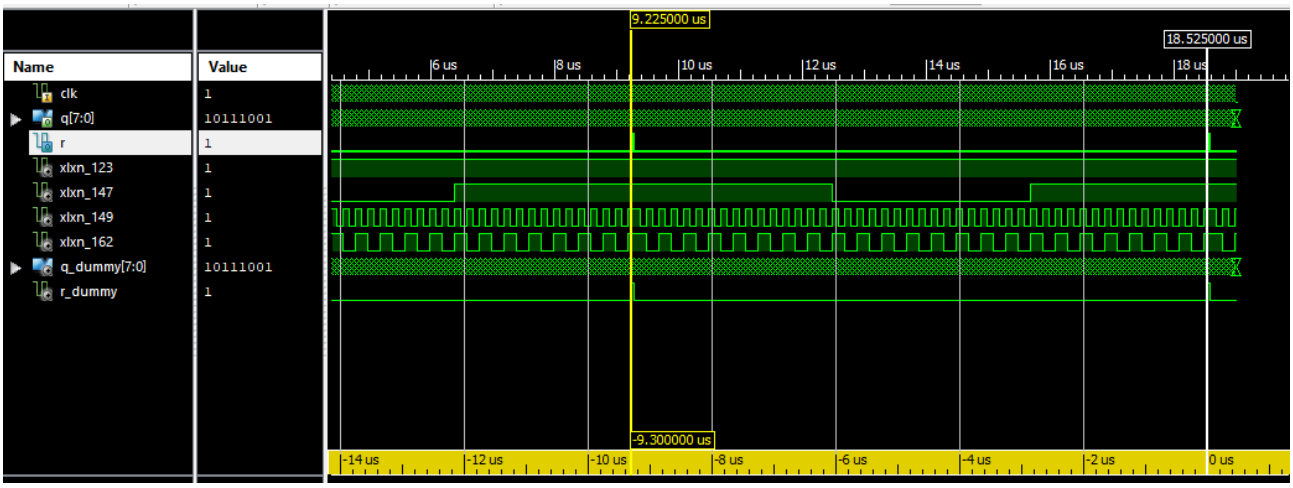
1. Схемна реалізація завдання у Xilinx ISE:



2. Встановив період CLK в 50 ns = 0.05 us.



3. Проводжу симуляцію роботи схеми.

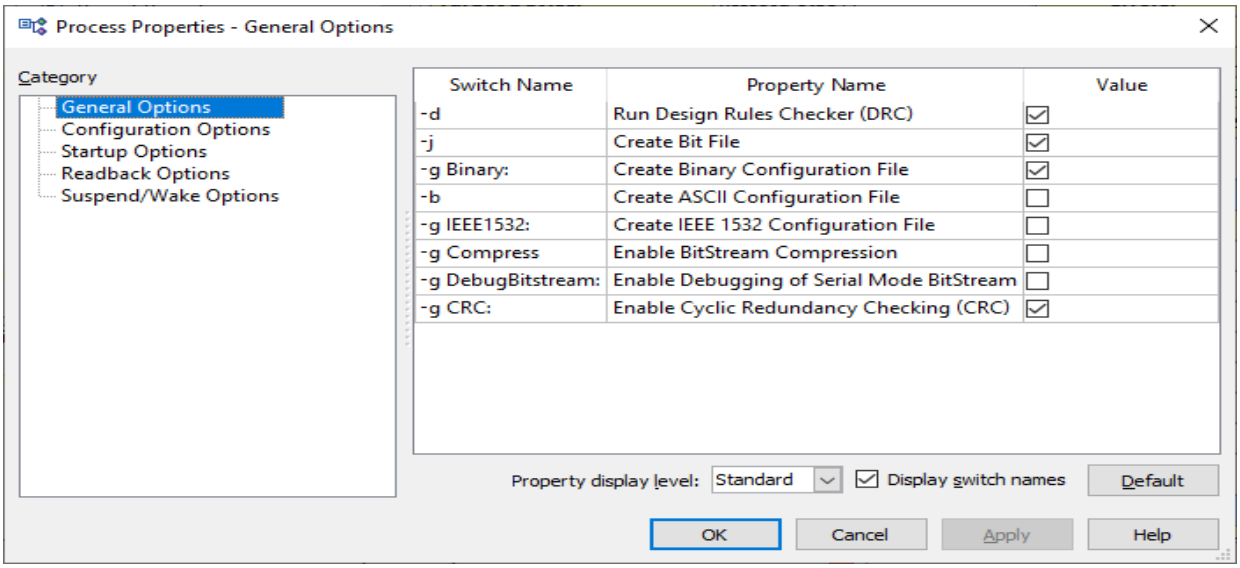


$T_{вих}$ = 18.525000 мкс – 9.225000 мкс = 9.3 мкс

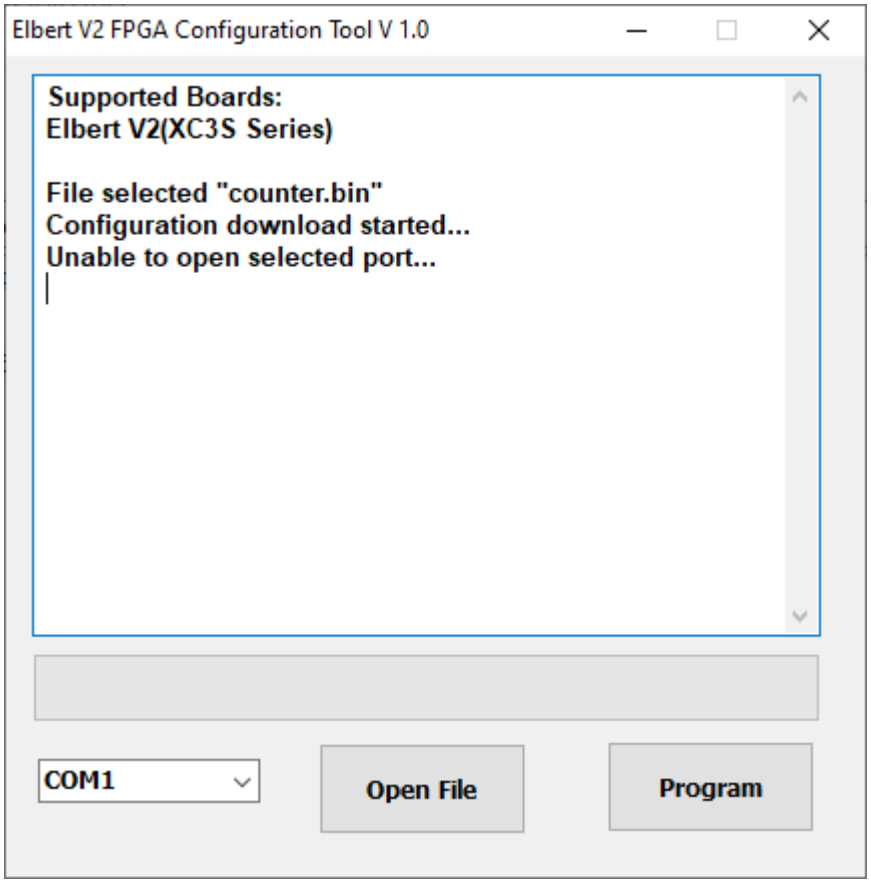
$T_{вх}$ = 50 нс = 0.05 мкс

К = $T_{вих}$/ $T_{вх}$ = 186 мкс.

4. Генерую BIT файл, який названий *counter.bin* для цільової FPGA. Для цього послідовно запускаю процеси Synthesize – XST, Implement Design та Generate Programming File.



5. Програмування лабораторного стенду отриманим *counter.bin* файлом.



Висновок: На даній лабораторній роботі я на базі стенда Elbert V2- Spartan 3A FPGA реалізував цифровий автомат світлових ефектів. Навчився створювати нові елементи і описувати логіку їх роботи засобами VHDL.