Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Зображення, що містить текст

Автоматично згенерований опис

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 4

Виконала:

Ст. гр. КІ-202

Замкова К.А

Прийняв:

Козак Н.Б.

Львів – 2023

**ЗАВДАННЯ**

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

2. Додати до проекту новий файл в якому буде реалізовано логіку формування вихідних сигналів.

3. У створеному файлі файл в проекті імплементувати інтерфейс логіки формування вихідних сигналів а також логічні вирази для формування кожного вихідного сихналу залежно від поточного стану автомата.

4. Промоделювати роботу схеми формування вихідних сигналів з усіма можливими наборами вхідних сигналів.

5. Аналогічним чином описати і промоделювати логіку формування переходів.

6. Додати до проекту новий файл в проекті в якому реалізувати пам ять стану автомата та зв язати між яобою всі його частини.

7. У файлі реалізувати пам ять стану автомата а також інтегрувати всі його компоненти між собою.

8. Промоделювати роботу автомата.

9. Створити файл верхнього рівня.

10. В файлі реалізувати подільник вхідної частоти та інтегрувати його зі створеним автоматом.

11. Додати файл та призначити виводам схеми фізичні виводи цільової FPGA.

12. Згенерувати бінарний файл та запрограмувати стенд.

13. Перевірити роботу системи на стенді Змінюючи положення DIP перимикача переконатись що автомат перемикається між інкрементним да декрементним режимами роботи.

**Варіант 4:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Стан |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**Порядок виконання**

1. Додати до проекту новий файл в якому буде реалізовано логіку формування вихідних сигналів.

OutputLogic.vhd

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

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

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

OUT\_BUS**(**3**)** **<=** **(**IN\_BUS**(**2**)** **and** **not(**IN\_BUS**(**1**)))or(**IN\_BUS**(**2**)and** **not(**IN\_BUS**(**0**)))or(not(**IN\_BUS**(**2**))and(**IN\_BUS**(**1**))and** IN\_BUS**(**0**));**

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

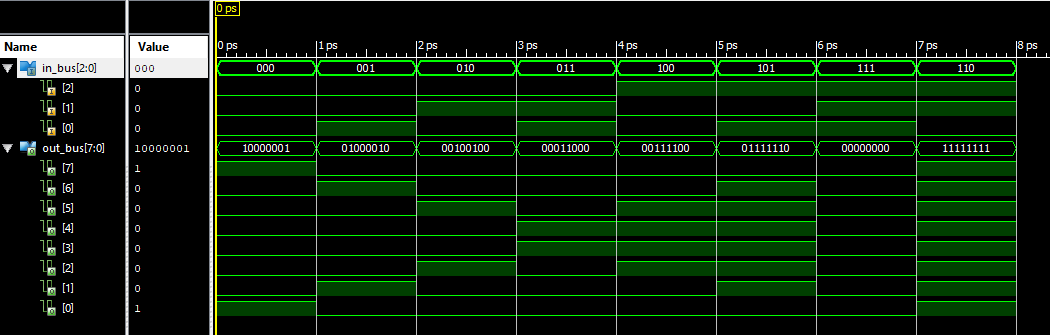
OUT\_BUS**(**5**)** **<=** **(**IN\_BUS**(**1**)** **and** **not(**IN\_BUS**(**0**)))** **or** **(**IN\_BUS**(**2**)and** **not(**IN\_BUS**(**1**)));**

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

OUT\_BUS**(**7**)** **<=** **(not(**IN\_BUS**(**2**))** **and** **not(**IN\_BUS**(**1**))** **and** **not(**IN\_BUS**(**0**)))or(**IN\_BUS**(**2**)** **and** IN\_BUS**(**1**)** **and** **not(**IN\_BUS**(**0**)));**

**end** out\_logic\_arch**;**

1. Промоделювати роботу схеми формування вихідних сигналів з усіма можливими наборами вхідних сигналів.:



1. Аналогічним чином описати і промоделювати логіку формування переходів:

TransitionLogic.vhd

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

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

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

**(**MODE **and** CUR\_STATE**(**1**)** **and** CUR\_STATE**(**0**));**

NEXT\_STATE**(**2**)** **<=** **(not** **(**MODE**)** **and** CUR\_STATE**(**2**)** **and** **not(**CUR\_STATE**(**1**)))or**

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

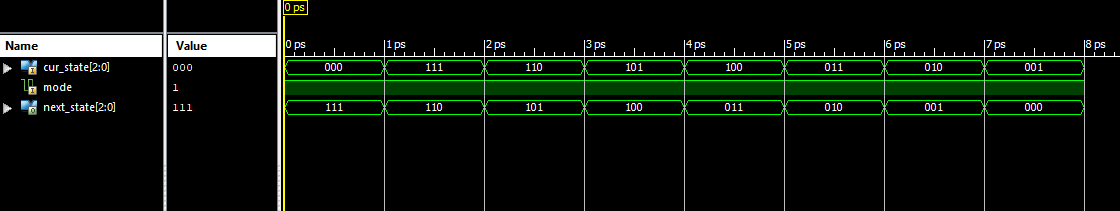
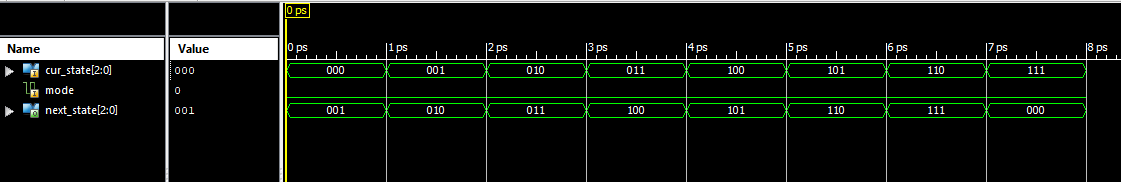
**(**MODE **and** CUR\_STATE**(**2**)** **and** CUR\_STATE**(**0**))or**

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

**(**MODE **and** **not(**CUR\_STATE**(**2**))** **and** **not(**CUR\_STATE**(**1**))** **and** **not(**CUR\_STATE**(**0**)));**

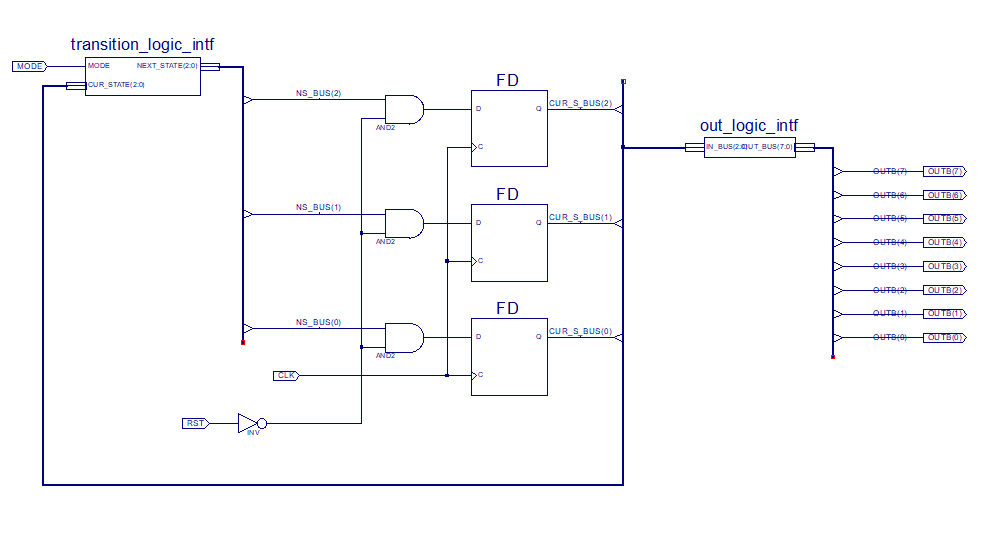
**end** transition\_logic\_arch**;**

**Результат:**

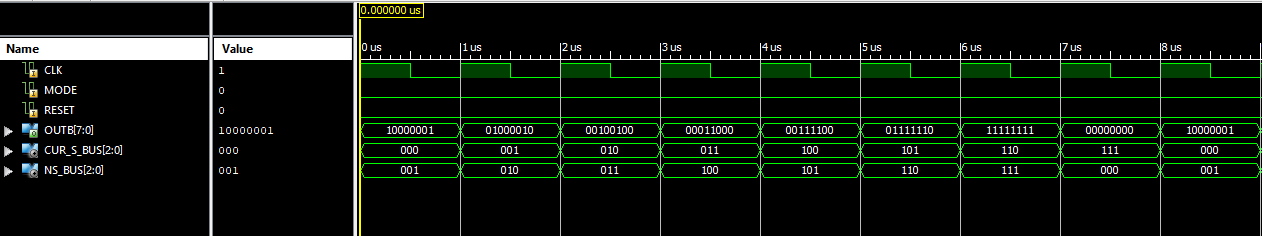


1. Додати до проекту новий файл в проекті в якому реалізувати пам ять стану автомата та зв язати між яобою всі його частини.

LightController.sch

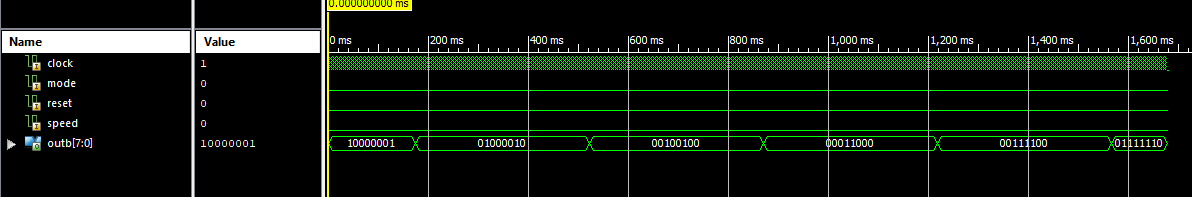
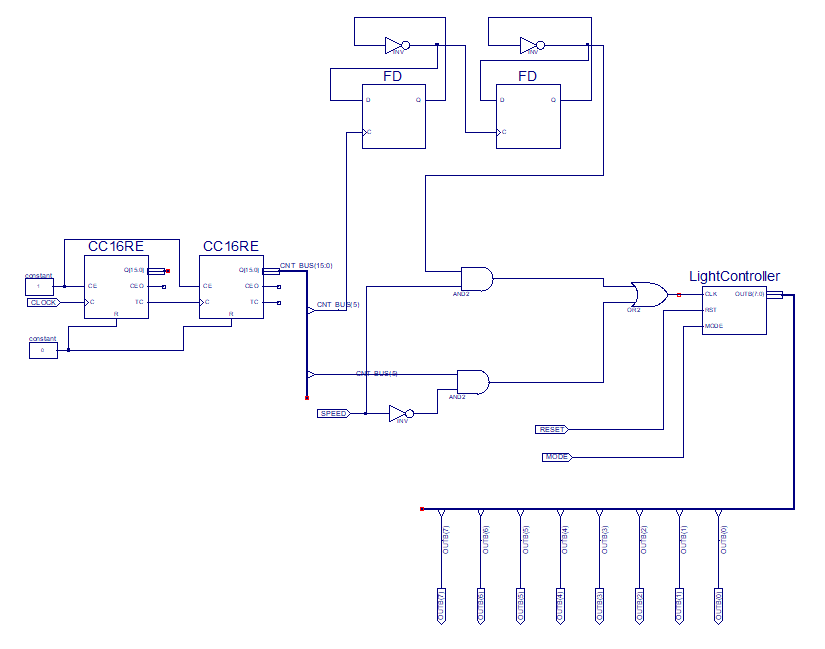


**Результат:**



1. Створити файл верхнього рівня і реалізувати подільник вхідної частоти та інтегрувати його зі створеним автоматом:

TopLevel.sch



1. Створити віртуальний тестовий стенд для перевірки роботи TopLevel

TESTBENCH.VHD

**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**;**

RESET **:** **IN** STD\_LOGIC**;**

CLOCK **:** **IN** STD\_LOGIC**;**

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

SPEED **:** **IN** STD\_LOGIC**);**

**END** **COMPONENT;**

**SIGNAL** MODE **:** STD\_LOGIC**;**

**SIGNAL** RESET **:** STD\_LOGIC**;**

**SIGNAL** CLOCK **:** STD\_LOGIC**;**

**SIGNAL** OUTB **:** STD\_LOGIC\_VECTOR **(**7 **DOWNTO** 0**);**

**SIGNAL** SPEED **:** STD\_LOGIC**;**

**BEGIN**

UUT**:** TopLevel **PORT** **MAP(**

MODE **=>** MODE**,**

RESET **=>** RESET**,**

CLOCK **=>** CLOCK**,**

OUTB **=>** OUTB**,**

SPEED **=>** SPEED

**);**

-- \*\*\* Test Bench - User Defined Section \*\*\*

init **:** **PROCESS**

**BEGIN**

MODE **<=** '0'**;**RESET**<=**'0'**;**SPEED**<=**'0'**;**

**wait** **for** 2000 ms**;**

MODE **<=** '1'**;**RESET**<=**'0'**;**SPEED**<=**'0'**;**

**wait** **for** 2000 ms**;**

MODE **<=** '1'**;**RESET**<=**'0'**;**SPEED**<=**'1'**;**

**wait** **for** 2000 ms**;**

MODE **<=** '1'**;**RESET**<=**'1'**;**SPEED**<=**'1'**;**

**wait** **for** 2000 ms**;**

**wait;**

**END** **PROCESS;**

clk **:** **PROCESS**

**BEGIN**

CLOCK **<=** '0'**;**

**wait** **for** **(**83 ns **/** 2**);**

CLOCK **<=** '1'**;**

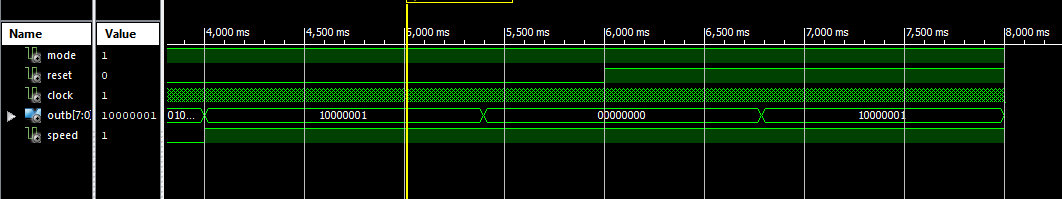
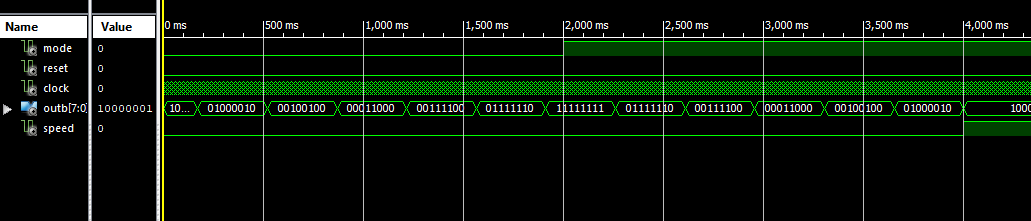
**wait** **for** **(**83 ns **/** 2**);**

**END** **PROCESS;**

-- \*\*\* End Test Bench - User Defined Section \*\*\*

**END;**

Результат



1. Перевірити роботу системи на стенді Змінюючи положення DIP перимикача переконатись що автомат перемикається між інкрементним да декрементним режимами роботи.

**Висновок**: У лабораторній роботі був розглянутий структурний опис цифрового автомата та проведена перевірка його роботи за допомогою стенда Elbert V2 - Spartan 3A FPGA. Аналіз структурного опису дозволив отримати більш глибоке розуміння принципу дії та взаємодії з іншими елементами системи.