

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**



Кафедра ЕОМ

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи №2
з дисципліни

" МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ "
для студентів першого (бакалаврського) рівня вищої освіти спеціальності

123 **"Комп'ютерна інженерія"**

Укладачі:

Юрчук А. Ф. асистент каф. ЕОМ
Цигилик Л.О., ст. викл. каф.ЕОМ

Львів – 2020

Тема роботи:

Структурний опис цифрового автомата. Перевірка роботи автомата за допомогою стенда *Elbert V2 – Spartan 3A FPGA*.

Мета роботи:

На базі стенда *Elbert V2 – Spartan 3A FPGA*, реалізувати цифровий автомат світлових ефектів згідно наступних вимог:

1. Інтерфейс пристрою та функціонал реалізувати згідно отриманого варіанту завдання.
Дивись розділ **Завдання**.
2. Логіку переходів реалізувати з використанням мови опису апаратних засобів *VHDL*.
Заборонено використовуючи оператори *if, switch, for, when*.
3. Логіку формування вихідних сигналів реалізувати з використанням мови опису апаратних засобів *VHDL*. Заборонено використовуючи оператори *if, switch, for, when*.
4. Згенерувати *Schematic* символи для *VHDL* описів логіки переходів та логіки формування вихідних сигналів.
5. Зінтегрувати всі компоненти (логіку переходів, логіку формування вихідних сигналів та пам'ять станів) в єдину систему за допомогою *ISE WebPACK™ Schematic Capture*.
Пам'ять станів реалізувати за допомогою графічних компонентів з бібліотеки.
6. Промодельовати роботу окремих частин автомата та автомата в цілому за допомогою симулятора *ISim*.
7. Інтегрувати створений автомат зі стендом *Elbert V2 – Spartan 3A FPGA* (додати подільник частоти для вхідного тактового сигналу, призначити фізичні виводи на *FPGA*).
8. Згенерувати *BIT* 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 на 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**).

Варіант – 2:

- 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	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

- 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 на 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 НИЖЧОЮ 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**).

Варіант – 3:

- 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	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0
2	0	0	1	1	0	0	0	0
3	0	0	0	1	1	0	0	0
4	0	0	0	0	1	1	0	0
5	0	0	0	0	0	1	1	0
6	0	0	0	0	0	0	1	1
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 на 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 4 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**).

Варіант – 4:

- 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	1
1	0	1	0	0	0	0	1	0
2	0	0	1	0	0	1	0	0
3	0	0	0	1	1	0	0	0
4	0	0	1	1	1	1	0	0
5	0	1	1	1	1	1	1	0
6	1	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0	0

- 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 на 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 4 РАЗИ НИЖЧОЮ 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**).

Варіант – 5:

- 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	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0
4	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0
7	0	0	0	0	1	0	0	0

- 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 на 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 (*TEST*) для подачі логічної «1» на всі виходи одночасно:
 - Якщо *TEST=0* то автомат перемикає сигнали на виходах згідно заданого алгоритму.
 - Якщо *TEST=1* то на всіх виходах повинна бути логічна «1» (всі *LED* увімкнені).
- Для керування сигналом *MODE* використати будь який з 8 *DIP* перемикачів (див. **Додаток – 1**).
- Для керування сигналами *RESET/TEST* використати будь які з *PUSH BUTTON* кнопок (див. **Додаток – 1**).

Варіант – 6:

- 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	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	0	0	0	0	1	0	0	0
5	0	0	0	0	1	1	0	0
6	0	0	0	0	1	1	1	0
7	0	0	0	0	1	1	1	1

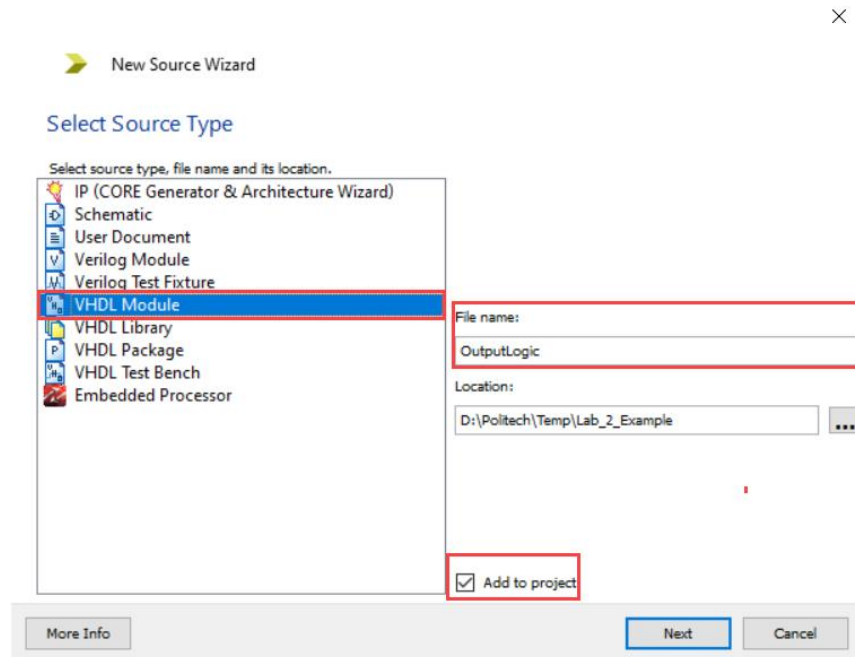
- Пристрій повинен використовувати $12MHz$ тактовий сигнал від мікроконтролера *IC1* і знижувати частоту за допомогою внутрішнього подільника. Мікроконтролер *IC1* є частиною стенда *Elbert V2 – Spartan 3A FPGA*. Тактовий сигнал заведено на вхід $LOC = P129\text{ FPGA}$ (див. **Додаток – 1**).
- Інтерфейс 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 (*TEST*) для подачі логічної «1» на всі непарні виходи одночасно:
 - Якщо $TEST=0$ то автомат перемикає сигнали на виходах згідно заданого алгоритму.
 - Якщо $TEST=1$ то на непарних виходах (7, 5, 3, 1) повинна бути логічна «1» (непарні *LED* увімкнені).
- Для керування сигналом *MODE* використати будь який з 8 *DIP* перемикачів (див. **Додаток – 1**).
- Для керування сигналами *RESET/TEST* використати будь які з *PUSH BUTTON* кнопок (див. **Додаток – 1**).

МЕТОДИЧНІ ВКАЗІВКИ:

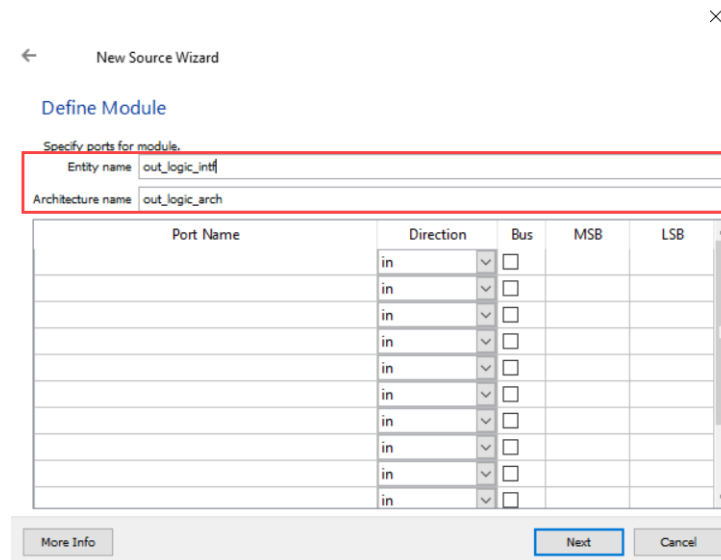
- Даний розділ надає покрокові інструкції реалізації Варіанту №1 завдання з наступними обмеженнями:
- Пристрій реалізує 4 стани замість 8.
- Пристрій реалізує лише 4 сигнали керування для *LED_7..LED_4*.
- Інтерфейс пристрою не містить входів *SPEED/TEST/RESET*.

Покрокові інструкції:

1. Створити новий проект, користуючись методичними вказівками до лабораторної роботи №1.
2. Додати до проекту новий *VHDL* файл, в якому буде реалізовано логіку формування вихідних сигналів.
 - Виконати команду *Project->New Source*.
 - У *New Project Wizard* обрати тип файла *VHDL Module*. В полі *File Name* вказати назву файла. Увімкнути опцію *Add to Project*.

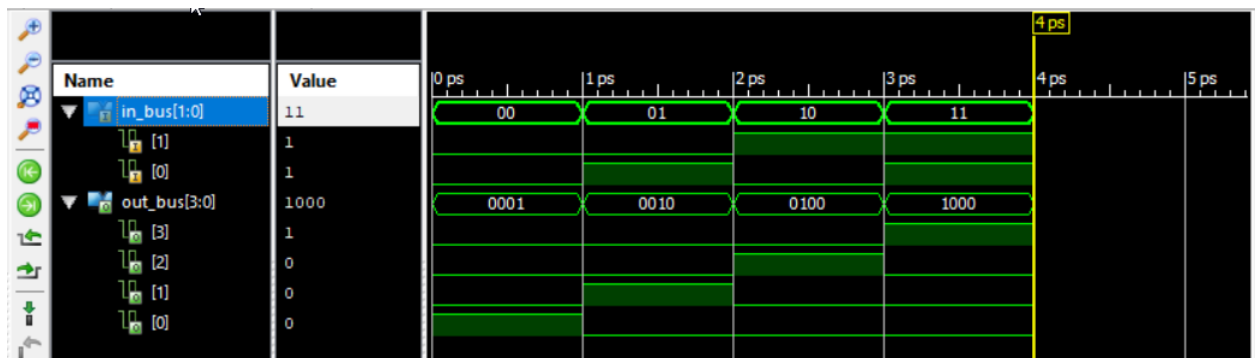


- У наступному вікні вказати ім'я інтерфейса і архітектури. Решту полів можна залишити порожніми.

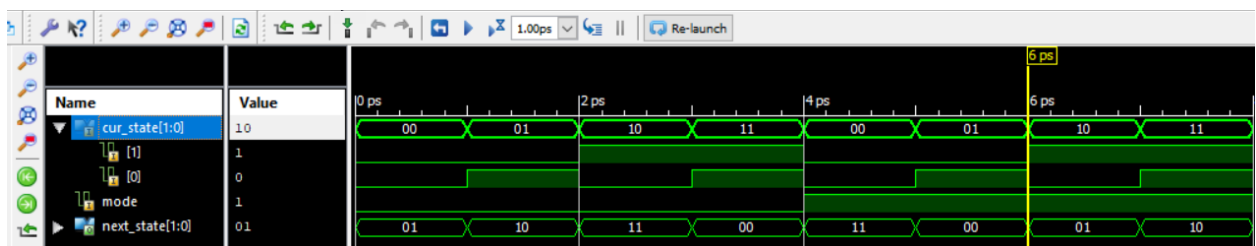


Натиснути *Next*. Завершити створення проекту, слідуючи підказкам.

- У створеному файлі (файл *OutputLogic.vhd* в проекті *Lab_2_Example*) імплементувати інтерфейс логіки формування вихідних сигналів, а також – логічні вирази для формування кожного вихідного сигналу, залежно від поточного стану автомата.
- Промодельовати роботу схеми формування вихідних сигналів з усіма можливими наборами вхідних сигналів (див. процес симуляції описано в методичних вказівках до лабораторної роботи №1)

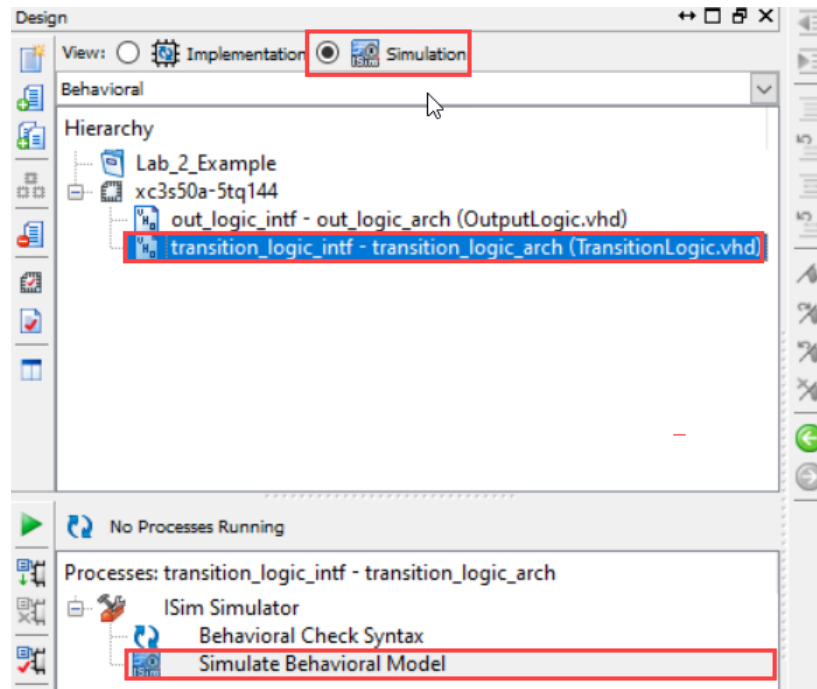


- Аналогічним чином описати і промодельовати логіку формування переходів (файл *TransitionLogic.vhd* в проекті *Lab_2_Example*).

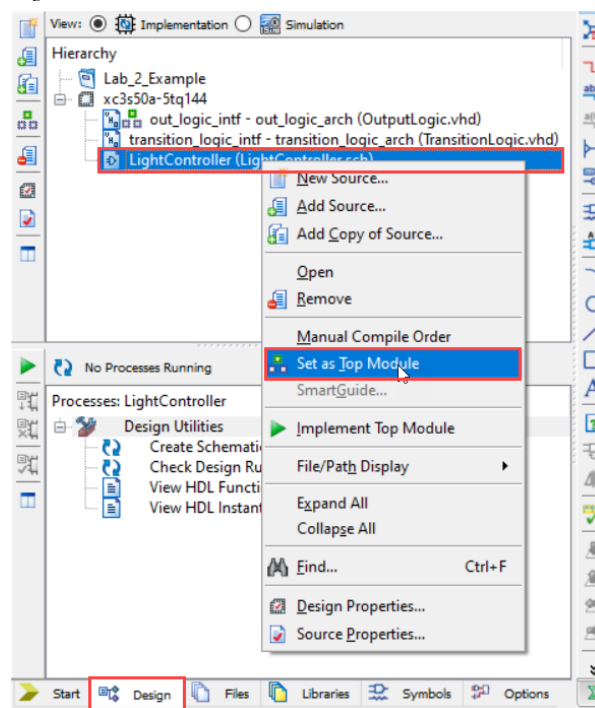


Примітки:

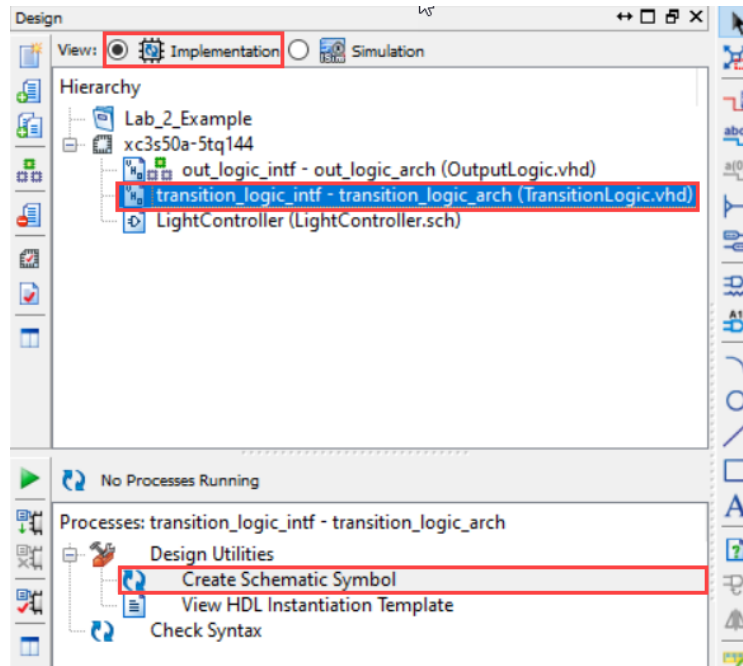
- Перед запуском симулятора переконайтесь що вибрано саме той файл, для якого збирається проводити симуляцію.



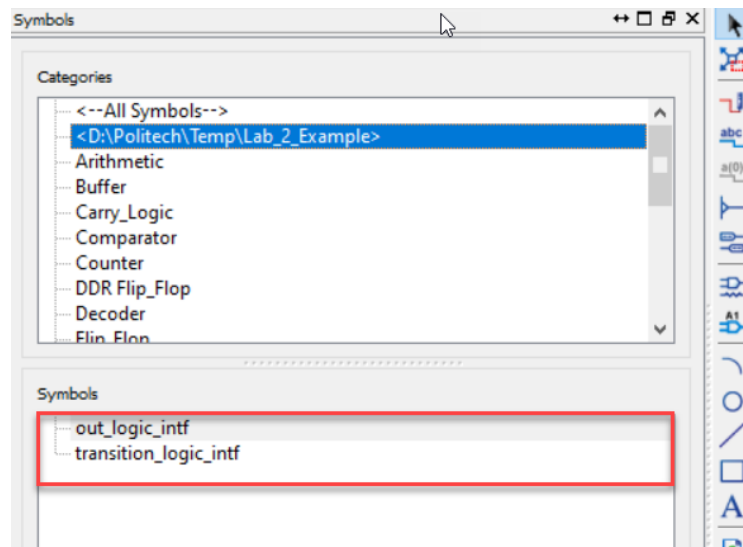
7. Додати до проекту новий *Schematic* (файл *LightController.sch* в проекті *Lab_2_Example*), в якому реалізувати пам'ять стану автомата та зв'язати між ябою всі його частини.
8. На вкладці *Design* менеджера проектів виділити новостворений файл та виконати для нього команду *Set as Top Module* з контекстного меню.



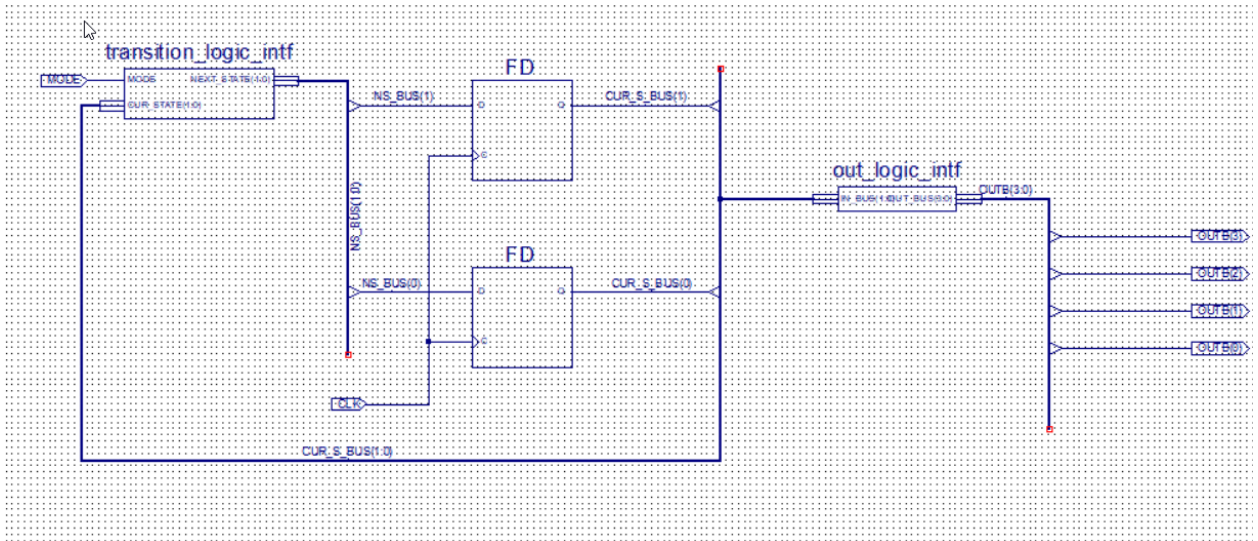
9. Згенерувати *Schematic* символи для файлів *OutputLogic.vhd* та *TransitionLogic.vhd*, виконавши команду *Create Schematic Symbol* для кожного файла:



10. У менеджері проектів відкрити файл *LightController.sch*. Далі перейти до бібліотеки компонентів і переконатись, що символи для файлів *OutputLogic.vhd* та *TransitionLogic.vhd* згенеровано:



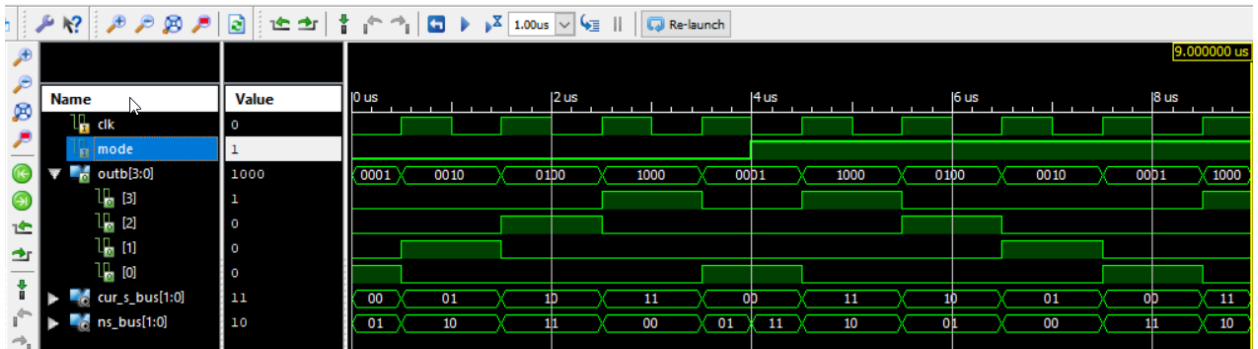
11. У файлі *LightController.sch* реалізувати пам'ять стану автомата, а також – інтегрувати всі його компоненти між собою:



Примітки:

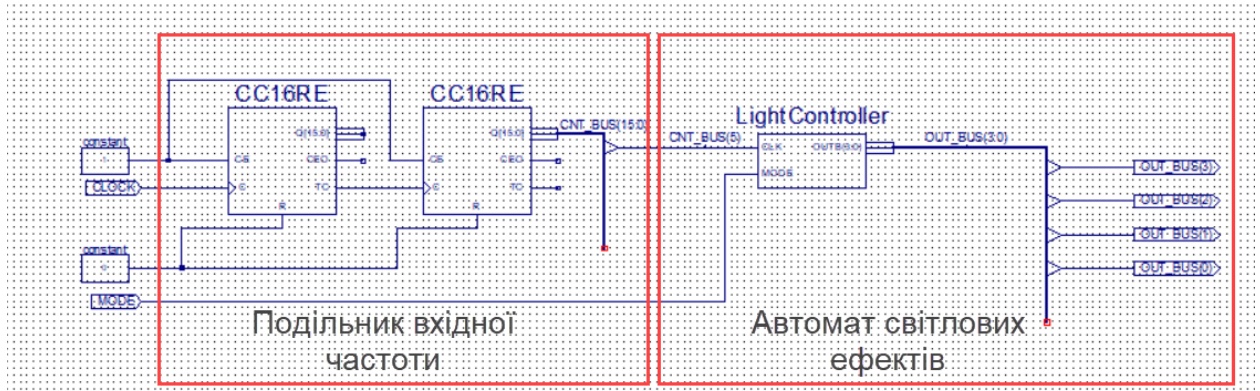
- Всі варіанти завдань мають вимогу реалізувати вхід синхронного скидання (*RESET*). На цьому рівні його легко можна реалізувати.

12. Промодельовати роботу автомата:



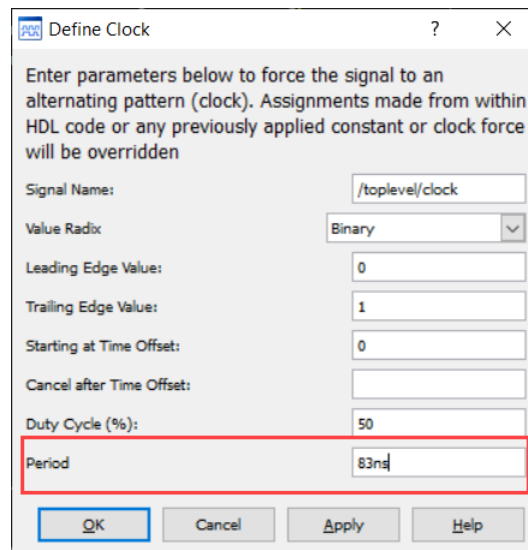
13. Створити *Schematic* файл верхнього рівня (файл *TopLevel.sch* в проекті *Lab_2_Example*).
14. На вкладці *Design* менеджера проектів виділити новостворений файл та виконати для нього команду *Set as Top Module* з контекстного меню.
15. Виконати команду *Create Schematic Symbol* для файлу *LightController.sch* та згенерувати СИМВОЛ.

16. В файлі *TopLevel.sch* реалізувати подільник вхідної частоти та інтегрувати його зі створеним автоматом (використовуючи символ, створений зі схеми автомата):

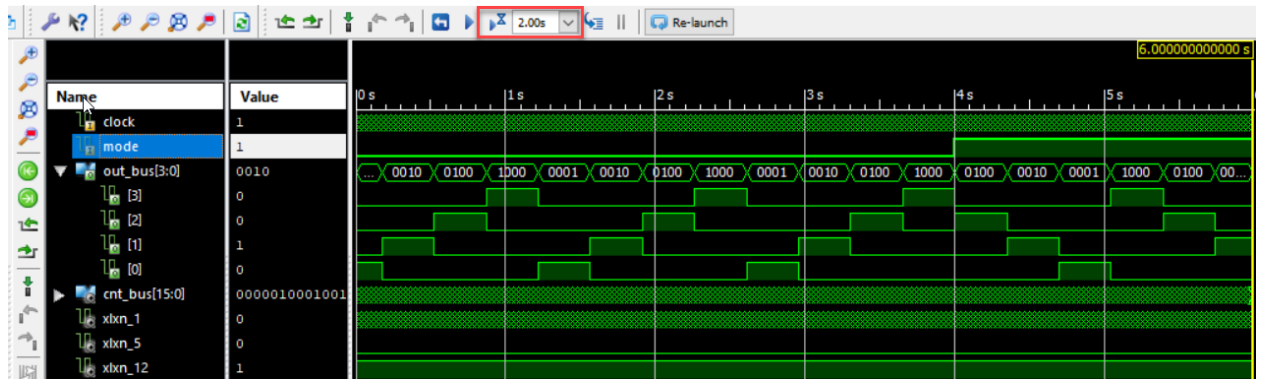


Примітки:

- Всі варіанти завдань мають вимогу реалізувати вхід скидання *SPEED* або вхід *TEST*. На цьому рівні їх легко можна реалізувати.
17. Запустіть симулятор для файлу *TopLevel.sch*. Встановіть період 83ns (12MHz) для сигналу *CLOCK*:



18. Встановіть значення 2s для часу симуляції та виконайте симуляцію системи в цілому з різними значеннями сигналів *MODE/RESET/SPEED/TEST*:



Примітки:

- Якщо симуляція триває занадто довго то можна тимчасово(на час симуляції) зменшити значення подільника вхідної частоти і час симуляції.
19. Додати файл *Constraints.ucf* та призначити виводам схеми фізичні виводи цільової FPGA. **Додаток-1** містить таблиці відповідності між позначеннями в оригінальному UCF файлі ([elbertv2.ucf](#)) і елементами на стенді *Elbert V2 – Spartan 3A FPGA*, а також – малюнок з розміщенням елементів на платі.

- Перейти на вкладку *Design*, відкрити створений файл і скопіювати до нього вміст [elbertv2.ucf](#).

```

8  #*****
9  #                               UCF for ElbertV2 Development Board
10 #*****
11 CONFIG VCCAUX = "3.3" ;
12
13 # Clock 12 MHz
14 NET "CLOCK"                      LOC = P129 | IOSTANDARD = LVCMOS33 | PERIOD = 12MHz;
15
16 #####
17 #                               LED
18 #####
19
20 NET "OUT_BUS(0)"                  LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
21 NET "OUT_BUS(1)"                  LOC = P47 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
22 NET "OUT_BUS(2)"                  LOC = P48 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
23 NET "OUT_BUS(3)"                  LOC = P49 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
24
25 #####
26 #                               DP Switches
27 #####
28
29 NET "MODE"                        LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
30

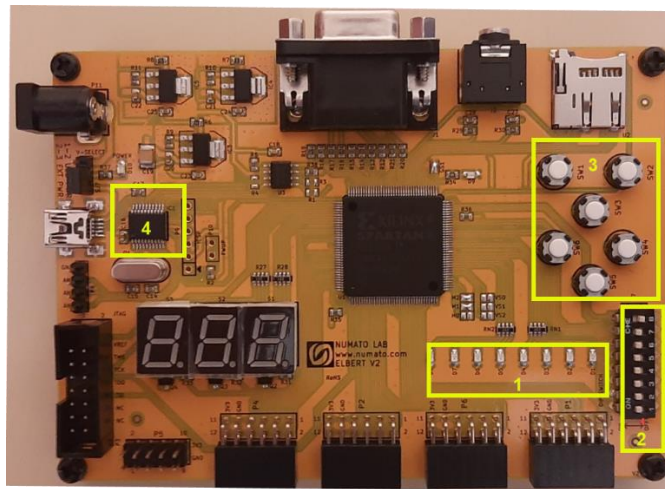
```

- У файлі закоментувати або видалити все, крім наступних розділів:
 - Розділ глобальних налаштувань (Clk, VCCAUX).
 - LED.
 - DP Switches.
- Перейти до розділу *LED*. Переіменувати *LED[0]*, *LED[1]*, *LED[2]* і *LED[3]* на *OUT_BUS(0)*, *OUT_BUS(1)*, *OUT_BUS(2)* і *OUT_BUS(3)*, відповідно до назв виходів на схемі. Решту виводів закоментувати.
- Перейти до розділу *DP Switches*. Переіменувати *DPSwitch[0]* на *MODE*, відповідно до назв входів на схемі. Решту виводів закоментувати.

20. Згенерувати бінарний файл та запрограмувати стенд.
21. Перевірити роботу системи на стенді. Змінюючи положення *DIP* перекача #8 переконатись що автомат перемикається між інкрементним да декрементним режимами роботи.

Додаток – 1.

Розміщення елементів на стенді:



1. Світлодіоди *D1-D8*. Конфігурацію виводів *FPGA*, до яких приєднано світлодіоди, описано в розділі *LED* оригінального *UCF* файла ([elbertv2.ucf](#)). Світлодіоди вмикаються подачею логічної «1» на відповідний вивід *FPGA*.
2. Регістр з 8 *DIP* перемикачів. В положенні *ON* перемикач приєднує вивід *FPGA* до *GND*. В положенні *OFF* – коло розірване і вивід нікуди не підключений. Конфігурацію виводів *FPGA*, до яких приєднано світлодіоди, описано в розділі *DP Switches* оригінального *UCF* файла ([elbertv2.ucf](#)).
Оскільки *UCF* конфігурує виводи як *PULL UP* то положення *ON* (вивід приєднано до *GND*) буде сприйматись як логічний «0», а положення *OFF* (коло розірване, проте вивід підтягнутий до *VDD* за допомогою внутрішнього резистора), буде сприйматись як логічна «1».
3. 5 кнопок *PUSH BUTTON*. В положенні *ON* перемикач приєднує вивід *FPGA* до *GND*. В положенні *OFF* – коло розірване і вивід нікуди не підключений. Конфігурацію виводів *FPGA*, до яких приєднано світлодіоди, описано в розділі *Switches* оригінального *UCF* файла ([elbertv2.ucf](#)).
Оскільки *UCF* конфігурує виводи як *PULL UP* то положення *ON* (вивід приєднано до *GND*) буде сприйматись як логічний «0», а положення *OFF* (коло розірване, проте вивід підтягнутий до *VDD* за допомогою внутрішнього резистора), буде сприйматись як логічна «1».
4. Мікроконтролер, який призначений для програмування стенда і генерування тактового сигналу для *FPGA*. Частота генерованого тактового сигналу становить 12MHz. Конфігурацію виводу *FPGA*, до якого приєднано сигнал *Clk* від мікроконтролера, описано в розділі *UCF for ElbertV2 Development Board* оригінального *UCF* файла ([elbertv2.ucf](#)).

Більше інформації та схему електричну принципову можна знайти за наступним посиланням:

<https://numato.com/product/elbert-v2-spartan-3a-fpga-development-board>

Таблиці відповідності між позначеннями в оригінальному UCF файлі ([elbertv2.ucf](#)) і елементами на стенді *Elbert V2 – Spartan 3A FPGA*.

Розділ *LED*

<i>Позначення в UCF файлі</i>	<i>Маркування на стенді Elbert V2</i>
LED[0]	D8
LED[1]	D7
LED[2]	D6
LED[3]	D5
LED[4]	D4
LED[5]	D3
LED[6]	D2
LED[7]	D1

Розділ *DP Switches*

<i>Позначення в UCF файлі</i>	<i>Маркування на стенді Elbert V2</i>
DPSwitch[0]	P7[8]
DPSwitch[1]	P7[7]
DPSwitch[2]	P7[6]
DPSwitch[3]	P7[5]
DPSwitch[4]	P7[4]
DPSwitch[5]	P7[3]
DPSwitch[6]	P7[2]
DPSwitch[7]	P7[1]