

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»



Co-funded by the  
Erasmus+ Programme  
of the European Union



МЕТОДИЧНІ ВКАЗІВКИ  
до виконання лабораторних робіт з дисципліни  
**“ Валідація та верифікація інформаційних систем ”**  
для студентів ОР «Магістр»  
спеціальності 122 ”Комп’ютерні науки”  
усіх форм навчання  
**Частина 1. Функціональне тестування**  
**з використанням віддаленої лабораторії GOLDi**

2023 р.

Методичні вказівки до виконання лабораторних робіт з дисципліни ” Валідація та верифікація інформаційних систем ” для студентів ОР «Магістр» спеціальності 122 ”Комп’ютерні науки” усіх форм навчання. Частина 1. Функціональне тестування з використанням віддаленої лабораторії GOLDi / Укл.: Г.В. Табунщик, І.Я. Зеленцова, Т.В. Голуб.- Запоріжжя: НУ«Запорізька політехніка», 2023. - 38с.

Укладачі: Г.В. Табунщик, професор, к.т.н.  
І.Я. Зеленцова, доцент, к.т.н.  
Т.В. Голуб, доцент, к.т.н.

Рецензент А.В. Пархоменко, доцент, к.т.н.

Відповідальний за випуск: І.Я. Зеленцова, доцент, к.т.н.

*Матеріали розроблено за підтримки міжнародного проекту 619034-EPP-1-2020-1-UA-EPPKA2-CBHE-JP “Cross-domain competences for healthy and safe work in the 21st century” [WORK4CE], що фінансується Європейською Комісією. Підтримка Європейською комісією випуску цієї публікації не означає схвалення змісту, який відображає лише думки авторів, і Комісія не може нести відповідальність за будь-яке використання інформації, що міститься в ній.*

Затверджено  
на засіданні кафедри ПЗ  
Протокол № 12 від 09.06.2023

Затверджено  
на засіданні НМК КНТ  
Протокол № 3 від 27.10.2023

**ЗМІСТ**

1 Лабораторна робота №1. Робота з FSM моделями .....	4
1.1 Мета роботи .....	4
1.2 Завдання на лабораторну роботу.....	4
1.3 Постановка задачі .....	4
1.4 Виконання лабораторної роботи .....	5
Лабораторна робота № 2. Функціональне тестування пристроїв з використанням віддаленої лабораторії GOLDI .....	9
2.1 Мета роботи .....	9
2.2 Завдання на лабораторну роботу.....	9
2.3 Виконання лабораторної роботи .....	9
Лабораторна робота № 3. Тестування схеми FSM на платі швидкого прототипування .....	16
3.1 Мета роботи .....	16
3.2 Завдання на лабораторну роботу.....	16
3.3 Постановка задачі .....	16
3.4 Виконання лабораторної роботи .....	17
Перелік джерел посилань.....	35
Додаток А Інструкція з підключення та приклад використання паketу віддаленої лабораторії GOLDI .....	36

# **1 ЛАБОРАТОРНА РОБОТА №1. РОБОТА З FSM МОДЕЛЯМИ**

## **1.1 Мета роботи**

Виконати розробку FSM-моделі керуючого автомату для заданого робочого процесу. Опис процесу сформулювати і надати як постановку задачі. Виконати тестування розробленої моделі. Ознайомитися з інструментами дистанційної лабораторії GOLDI.

## **1.2 Завдання на лабораторну роботу**

1.2.1 Створити умови задачі – технічне завдання на розробку моделі, керуючої автоматом Мура.

1.2.2 Зробити теоретичну частину – синтез графа станів та таблиці переходів.

1.2.3 Виконати тестування отриманої моделі автомата Мура у пакеті Goldi.

1.2.4 Скріншоти графа, таблиці та часової діаграми тестування додати до звіту.

1.2.5 Зробити висновок по результати верифікації моделі керуючого автомата.

## **1.3 Постановка задачі**

По конвеєру переміщуються коробки цукерок. Перевіряється вага кожної коробки датчиком ваги. За нормальну вагу приймається 200 г. У такому випадку коробка переміщується у спеціальний ящик з такими ж коробками цукерок. Після кожного переміщення коробки у ящик, аналізується його вага за допомогою ваг. Якщо вага ящика дорівнює 1 кг, то ящик замінюється новим, порожнім.

## 1.4 Виконання лабораторної роботи

### 1.4.1 Теоретична частина

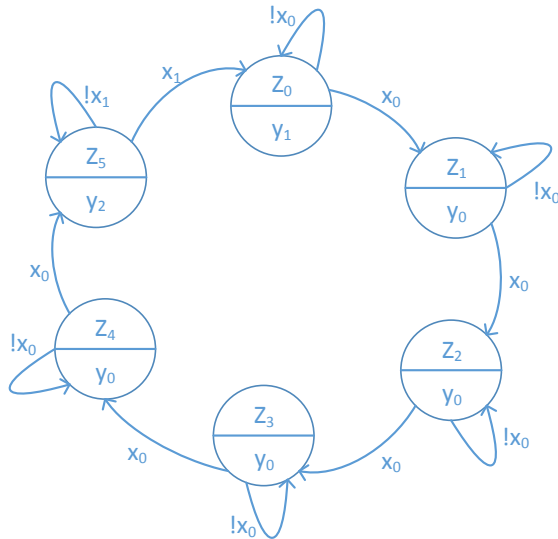


Рисунок 1.1 - Підготовка графа станів автомата

$$\begin{cases} Y = f(Z) \\ \Phi = f(x, Z) \end{cases} \text{ – Система функцій керуючого автомата Мура}$$

Вхідні сигнали:

$x_0$  – датчик ваги встановив, що вага коробки 200г.

$x_1$  – датчик ваги встановив, що ящик важить 1кг.

$X = \{x_0, x_1\}$

Вихідні сигнали:

$y_0$  – поточна коробка переміщується у ящик

$y_1$  – ящик замінюється новим (порожнім) ящиком

$y_2$  – почати завантаження нового ящика

$Y = \{y_0, y_1, y_2\}$

## Граф станів автомата Мура.

Стани:

z0 – 0 г.

z1 – 200 г.

z2 – 400 г.

z3 – 600 г.

z4 – 800 г.

z5 – 1 кг.

Формули переходів.

$$Z0 = Z0 \& !x0 \# Z1 \& x0$$

$$Z1 = Z1 \& !x0 \# Z2 \& x0$$

$$Z2 = Z2 \& !x0 \# Z3 \& x0$$

$$Z3 = Z3 \& !x0 \# Z4 \& x0$$

$$Z4 = Z4 \& !x0 \# Z5 \& x0$$

$$Z5 = Z0 \& x1$$

Формули виходів.

$$y1 = Z0$$

$$y0 = Z1$$

$$y0 = Z2$$

$$y0 = Z3$$

$$y0 = Z4$$

$$y2 = Z5$$

Таблиця переходів автомата Мура

	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>	Z <sub>4</sub>	Z <sub>5</sub>
Z <sub>0</sub>	!x <sub>0</sub>	x <sub>0</sub>				
Z <sub>1</sub>		!x <sub>0</sub>	x <sub>0</sub>			
Z <sub>2</sub>			!x <sub>0</sub>	x <sub>0</sub>		
Z <sub>3</sub>				!x <sub>0</sub>	x <sub>0</sub>	
Z <sub>4</sub>					!x <sub>0</sub>	x <sub>0</sub>
Z <sub>5</sub>	x <sub>1</sub>					!x <sub>1</sub>

Таблиця виходів автомата Мура

	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>
Z <sub>0</sub>		1	
Z <sub>1</sub>	1		
Z <sub>2</sub>	1		
Z <sub>3</sub>	1		
Z <sub>4</sub>	1		
Z <sub>5</sub>			1

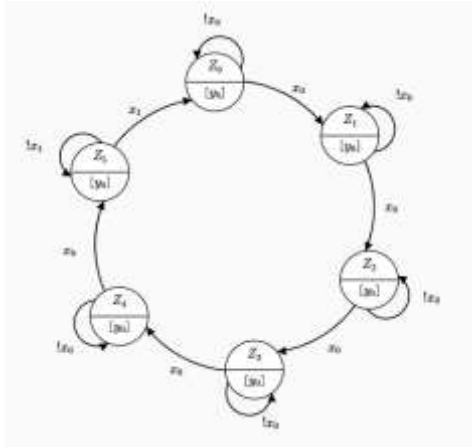


Рисунок 1.2 – Граф станів автомата Мура в системі GOLDi

Transition matrix

Input Variables: 2  
 $X = \{x_1, x_2\}$

	Transitions								Output Variables		
	$x_1 x_2$	$x_1 \bar{x}_2$	$\bar{x}_1 x_2$	$\bar{x}_1 \bar{x}_2$	$x_1 x_2$	$x_1 \bar{x}_2$	$\bar{x}_1 x_2$	$\bar{x}_1 \bar{x}_2$	$z_0$	$z_1$	$z_2$
$z_0$	00	01							0	1	
$z_1$		00	01						1		
$z_2$			00	01					1		
$z_3$				00	01				0		
$z_4$					00	01			0		
$z_5$	01					00	01				1
$z_6$											
$z_7$											

Рисунок 1.3 – Таблиця переходів автомата

z-Equations unminimized

$$z_0 = !z^2!z^1!z^0*x_0 + !z^2!z^1z^0*x_0 + !z^2z^1!z^0*x_0 + !z^2z^1z^0*x_0 + z^2!z^1z^0*x_0 + z^2!z^1z^0*x_1$$

$$z_1 = !z^2!z^1z^0*x_0 + !z^2z^1!z^0*x_0 + !z^2z^1z^0*x_0 + !z^2z^1z^0*x_0$$

$$z_2 = !z^2z^1z^0*x_0 + z^2!z^1!z^0*x_0 + z^2!z^1z^0*x_0 + z^2!z^1z^0*x_1$$

Рисунок 1.4 – Функції переходів

y-Equations

$$y_0 = lz^2lz^1z^0 + lz^2z^1lz^0 + lz^2z^1z^0 + z^2lz^1lz^0$$

$$y_1 = lz^2lz^1lz^0$$

$$y_2 = z^2lz^1z^0$$

Рисунок 1.5 – Функції виходів

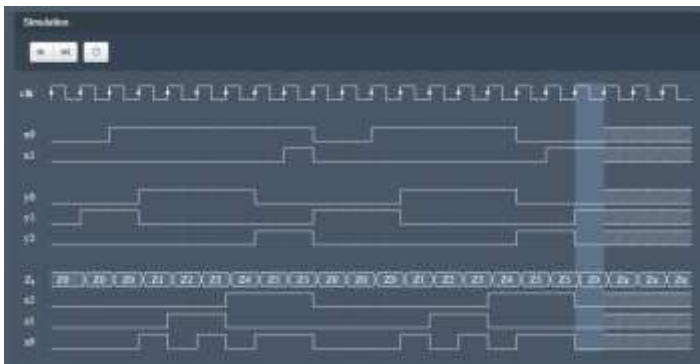


Рисунок 1.6 – Часова діаграма, результати тестування роботи автомата Мура в системі GOLDi

## ВИСНОВКИ

Виконано розробку FSM-моделі керуючого автомату для заданого робочого процесу.

Виконано тестування розробленої моделі за допомогою дистанційної лабораторії GOLDi. Результати тестування показали, що модель успішно пройшла верифікацію

**Інструкція з підключення та приклад використання пакету віддаленої лабораторії GOLDi наведено в додатку А.**



## ЛАБОРАТОРНА РОБОТА № 2. ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ ПРИСТРОЇВ З ВИКОРИСТАННЯМ ВІДДАЛЕНОЇ ЛАБОРАТОРІЇ GOLDI

### 2.1 Мета роботи

Навчитись використовувати кінцеві автомати для функціонального тестування, керувати фізичними моделями пристроїв віддаленої лабораторії GOLDi.

### 2.2 Завдання на лабораторну роботу

2.2.1 Скласти граф керуючого автомату для керування ліфтом на 3 поверхи.

2.2.2 Побудувати математичну модель.

2.2.3 Виконати верифікацію математичної моделі.

2.2.4 Підключитися к лабораторії Goldi для керуванням ліфтом.

2.2.5 Перевірити роботу на реальному об'єкті.

Розробити 2 моделі. У першій моделі ліфт рухається циклічно вгору та вниз. У другій моделі рух ліфта залежить від натискання кнопки, а також додати керування led – індикатором.

### 2.3 Виконання лабораторної роботи

#### 2.3.1 Теоретична частина

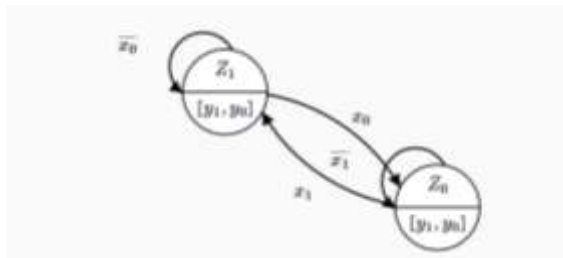


Рисунок 2.1 – Граф станів ліфту, що рухається вгору або вниз

$x_0$  – кнопка ВГОРУ  
 $x_1$  – кнопка ВНИЗ  
 $z_1$  – рух вниз  
 $z_0$  – рух вгору

Input Variables: 2		Transitions		Output Variables	
$X = \{x_0, x_1\}$		$z_0$	$z_1$	$y_0$	$y_1$
$z_0$	$x_0$	$z_0$	$z_1$	1	0
$z_1$	$x_1$	$z_1$	$z_0$	0	1

Рисунок 2.2 – Таблиця переходів для руху вгору та вниз

Формули для керуючого автомата Мура

$$z_0 = \neg z_0 * x_1 + z_0 * \neg x_0$$

$$y_0 = z_0$$

$$y_1 = \neg z_0$$

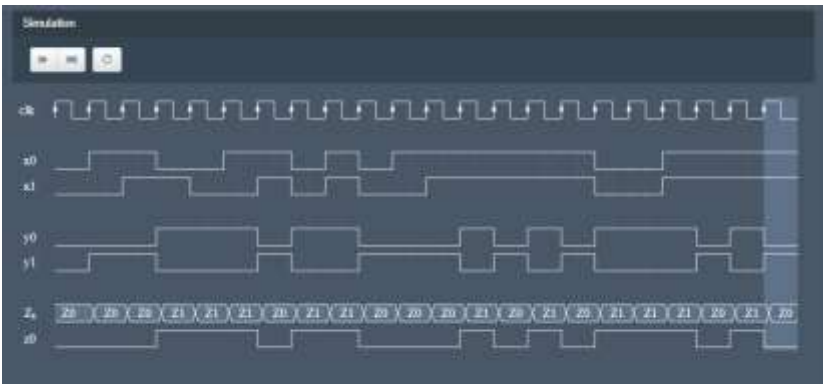


Рисунок 2.3 – Симуляція руху

Граф на рисунку 3.4 відображає модель керування ліфта таким чином, що є 4 стани. У першому стані кнопка не натиснута і ліфт

нікуди не їде. У трьох інших станах в залежності від натиснутої кнопки ліфт поїде на відповідний поверх.

На відміну від рис. 3.4, рис 3.1 відображає циклічний рух ліфта вгору та вниз. Це обумовлено двома станами, в яких може перебувати ліфт (або рух вгору, або рух вниз).

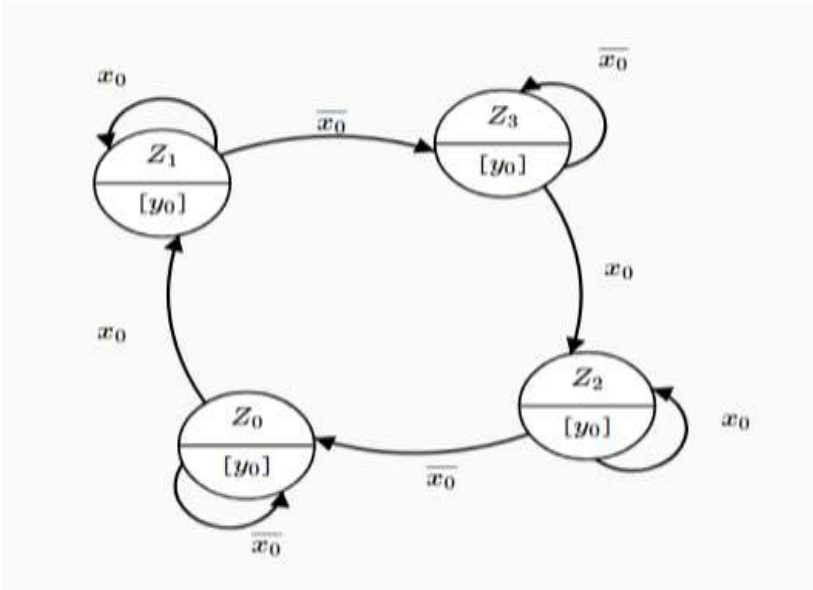


Рисунок 2.4 – Граф станів ліфту, що керується кнопкою

- $z_0$  – кнопка не натиснута та індикатор не горить
- $z_1$  – натиснення кнопки, загоряється індикатор
- $z_2$  – кнопка не натиснута, індикатор горить
- $z_3$  – натиснення кнопки, потухає індикатор

Input Variables: 2  
 $X = [x_1, x_0]$

Transitions					Output Variables		
	$z_0$	$z_1$	$z_2$	$z_3$	$y_0$	$y_1$	$y_2$
$z_0$	00	00					
$z_1$		00	00		1		
$z_2$			00	00	1		
$z_3$	00			00			

Рисунок 2.5 – Таблиця переходів автомата Мура

z-Equations

$$z_0 = !z_1!z_0*x_0 + z_1!z_0*x_0 + !z_1*z_0*x_0 + z_1*z_0*x_0$$

$$z_1 = z_1!z_0*x_0 + !z_1*z_0*x_0 + !z_1*z_0*x_0 + z_1*z_0*x_0$$


---

y-Equations

$$y_0 = z_1!z_0 + !z_1*z_0$$

Рисунок 2.6 – Функції переходів та виходів

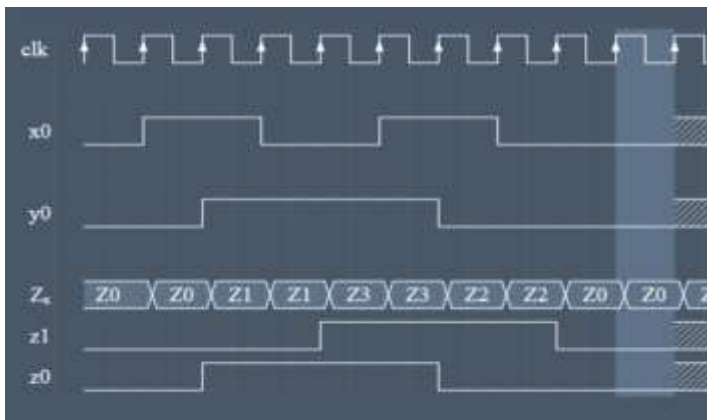


Рисунок 2.7 – Часова діаграма тестування автомата Мура

Після успішного виконання симуляції переходимо до функціонального тестування автомату Мура в системі GOLDI

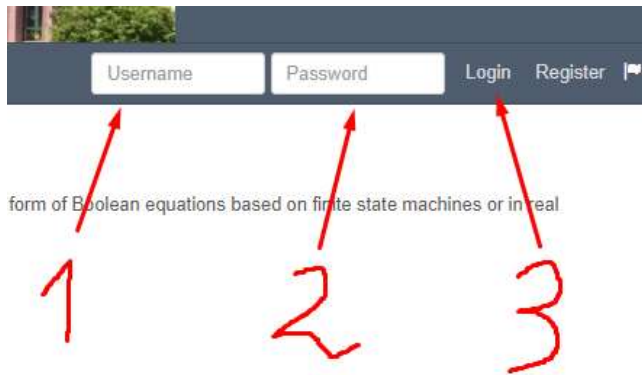


Рисунок 2.8 – Вхід до системи GOLDI, авторизація



Рисунок 2.9 – Обрання лабораторії та компонентів для подальшої роботи з ними

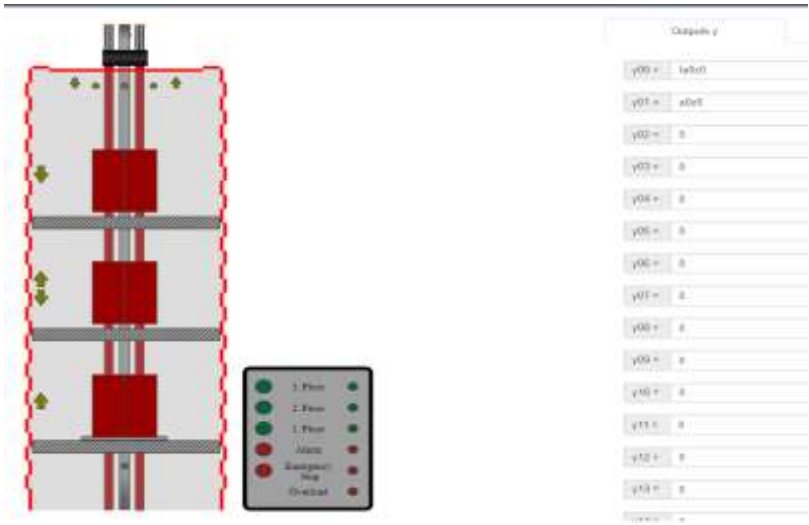


Рисунок 2.10 – Видяг установки до запуску та опис її станів

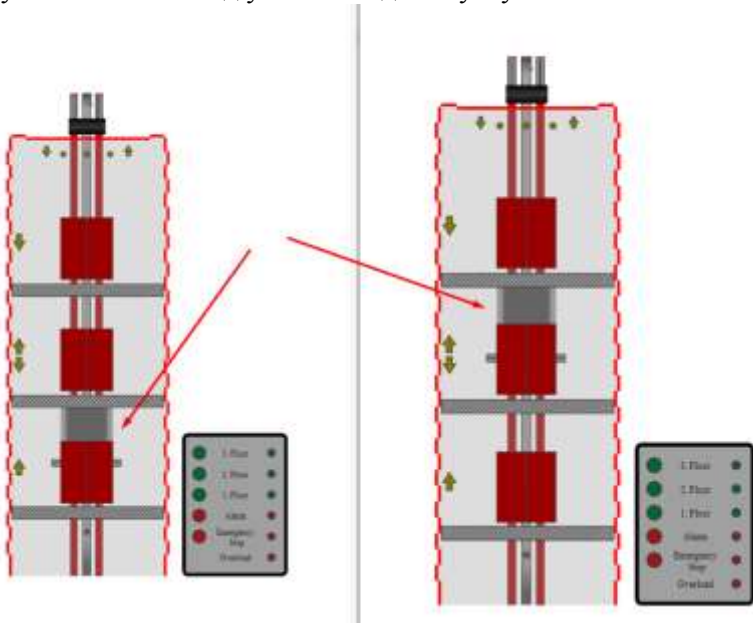


Рисунок 2.11 – Видяг установки після запуску

Циклічний рух:

y00= !a0z0

y10= a0z0



Рисунок 2.12 – Вигляд запусненої установки # 2 (використання кнопки та лампочки)

На рисунку 2.12 відображено запуснену установку. Рівняння, записані в y00 та y01 були взяті з файлу ECP.

## ВИСНОВКИ

В ході лабораторної роботи розробили керуючий автомат для керування ліфта.

У першій моделі ліфт рухається циклічно ввєрх та вниєз. У дрїгїй моделї рух ліфта залежить від натискання кнопки, та також додано керування led – індикатором.

Окрім цього було розроблено графи відповідних моделей та протестовано їх праєездатність за допомогою вбудованого інструменту симуляції.

## ЛАБОРАТОРНА РОБОТА № 3. ТЕСТУВАННЯ СХЕМИ FSM НА ПЛАТІ ШВИДКОГО ПРОТОТИПУВАННЯ

### 3.1 Мета роботи

Ознайомитися з основними етапами структурного синтезу FSM, виконати моделювання і тестування в онлайн лабораторії Goldi - опис схеми на мові VHDL в Quartus II. Виконати тестування схеми керуючого автомата Мура на платі Altera\Intel з використанням перемикачів та 7-сегментного індикатора.

### 3.2 Завдання на лабораторну роботу

3.2.1 Використати індивідуальне завдання з лабораторної роботи № 1, виконати підготовку синтезу FSM Мура по графу алгоритму керування.

3.2.2 Скласти граф-схему алгоритму керування.

3.2.3 Скласти таблицю кодування станів.

3.2.4 Скласти таблицю переходів автомата Мура.

3.2.5 Скласти формули функцій переходів та вихідних сигналів.

3.2.6 Описати автомат Мура на мові VHDL.

3.2.7 Створити проект в Quartus та синтезувати автомат згідно з індивідуальним завданням.

3.2.8 Призначити сигнали X на позиційні перемикачі, а сигнали Y на світлодіоди.

3.2.9 Перевірити алгоритм згідно з граф-схемою.

### 3.3 Постановка задачі

Завдання сформульовано на основі постановки задачі з лабораторної роботи № 1 і призначене для продовження моделювання:

По конвеєру переміщуються коробки цукерок. Перевіряється вага кожної коробки датчиком ваги. За нормальну вагу приймається 200 г. У такому випадку коробка переміщується у спеціальний ящик з такими ж коробками цукерок. Після кожного переміщення коробки у ящик, аналізується його вага за допомогою ваг. Якщо вага ящика дорівнює 1 кг, то ящик замінюється новим, порожнім.



### 3.4 Виконання лабораторної роботи

3.4.1 Теоретична частина відповідно етапам методики проектування

#### Крок 1. Постановка задачі

$$\begin{cases} Y = f(Z) \\ \Phi = f(x, Z) \end{cases} \quad - \text{ Система функцій керуючого автомата}$$

Мура

Вхідні сигнали:

$x_0$  – датчик ваги встановив, що коробка важить 200г.

$x_1$  – датчик ваги встановив, що ящик важить 1 кг.

$X = \{x_0, x_1\}$  – множина вхідних сигналів схеми автомата.

Вихідні сигнали:

$y_0$  – поточна коробка переміщується у ящик

$y_1$  – ящик замінюється новим (порожнім) ящиком

$y_2$  – почати завантаження нового ящика

$Y = \{y_0, y_1, y_2\}$  – множина вихідних сигналів схеми автомата.

#### Крок 2. Розробка граф-схеми алгоритму керування

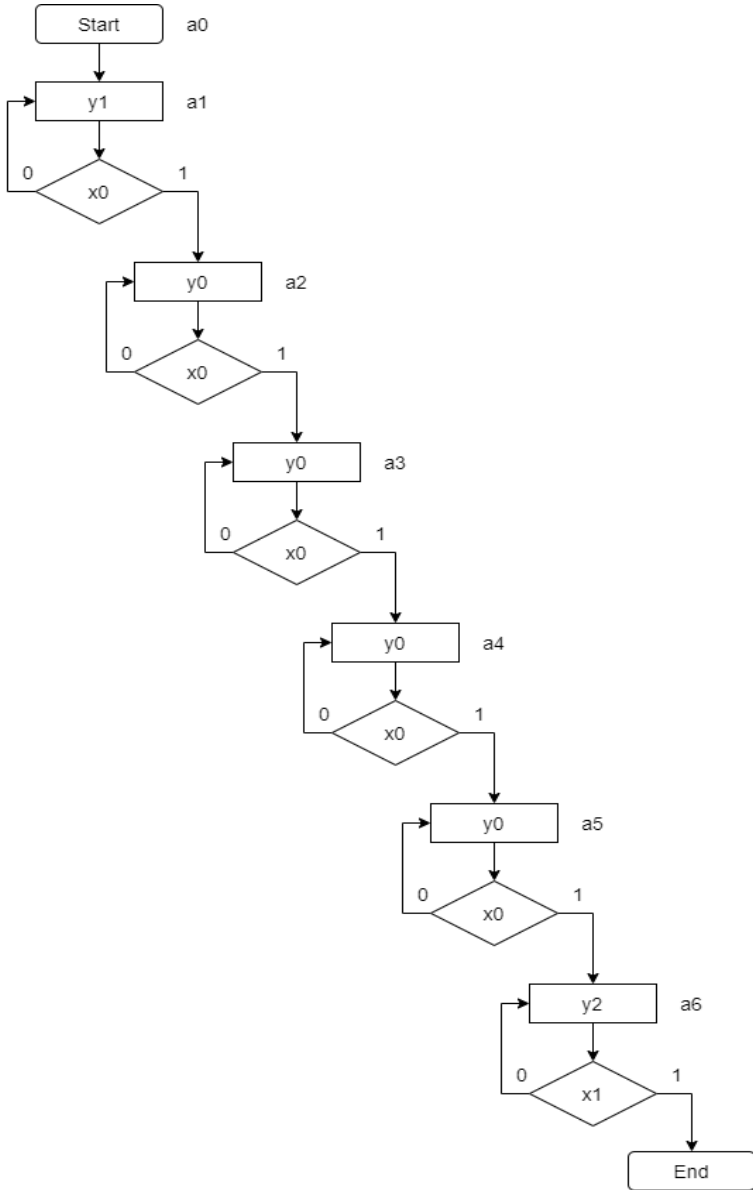


Рисунок 3.1 - Граф станів автомата Мура

Множина станів автомата Мура:

$A = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$

$M = |A| = 7;$

**Крок 3.** Кодування станів:

Розрядність коду

$R = \lceil \log_2 7 \rceil = 3;$

$D = [D_2, D_1, D_0]$  - функції активації D-тригерів у регістрі пам'яті стану.

a0	a1	a2	a3	a4	a5	a6
0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0
-	D0	D1	D1D0	D2	D2 D0	D2D1

**Крок 4.** Таблиця переходів автомата Мура (пряма структурна таблиця)

Вихідний стан	Наступний стан	Вхідний сигнал	Вихідний сигнал	Функція переходів (пам'ять)	Перехід		
am	k(am)	as	k(as)	x	y(am)	D	№
a0	000	a1	001	1	-	D0	1
a1	001	a2	010	x0	y1	D1	2
		a1	001	!x0		D0	3
a2	010	a3	011	x0	y0	D1 D0	4
		a2	010	!x0		D1	5
a3	011	a4	100	x0	y0	D2	6
		a3	011	!x0		D1 D0	7
a4	100	a5	101	x0	y0	D2 D0	8
		a4	100	!x0		D2	9
a5	101	a6	110	x0	y0	D2 D1	10
		a5	101	!x0		D2 D0	11
a6	110	a0	000	x1	y2	-	12
		a6	110	!x1		D2 D1	13

**Крок 5.** Формули функцій переходів (активація тригера D):

$D_0 = a_0 + a_1 * !x_0 + a_2 * x_0 + a_3 * !x_0 + a_4 * x_0 + a_5 * !x_0$

$D_1 = a_1 * x_0 + a_2 + a_3 * !x_0 + a_5 * x_0 + a_6 * !x_1$

$D_2 = a_3 * x_0 + a_4 + a_5 + a_6 * !x_1$

**Крок 6.** Формули функцій вихідних сигналів:

$$y_0 = a_2 + a_3 + a_4 + a_5$$

$$y_1 = a_1$$

$$y_2 = a_6$$

### 3.4.2 Експериментальна частина з використанням плати Altera

**Крок 7.** Опис автомата Мура на мові VHDL

7.1 Переходимо на сайт :

<https://fpgasoftware.intel.com/13.0sp1/?edition=web>

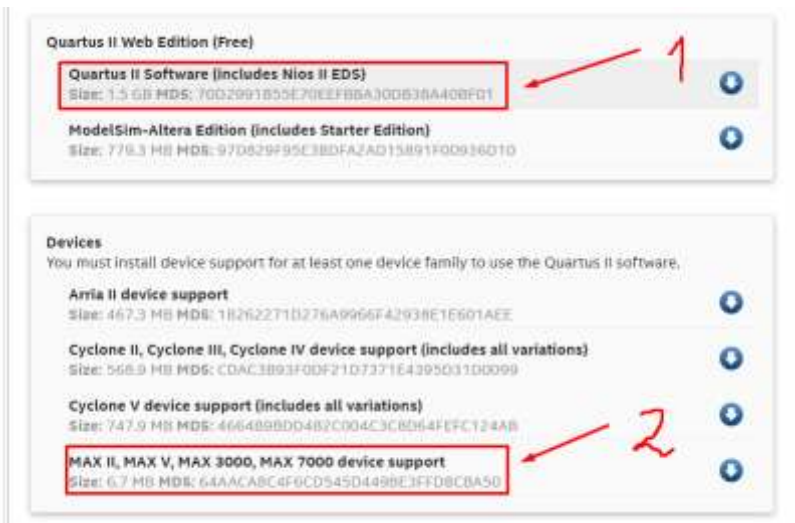


Рисунок 3.2 – Завантаження компонентів

7.2 Завантажуємо компоненти, зазначені на рисунку 3.2.

7.3 Файл з розширенням .qdz додаємо у директорії, в якій плануємо встановлювати Quartus. Приклад зображено на рисунку 3.3.

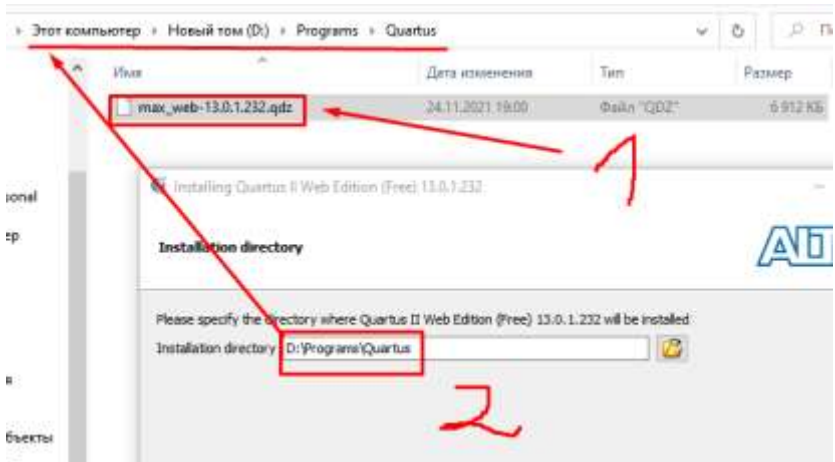


Рисунок 3.3 – Встановлення Quartus. Обрання директорії

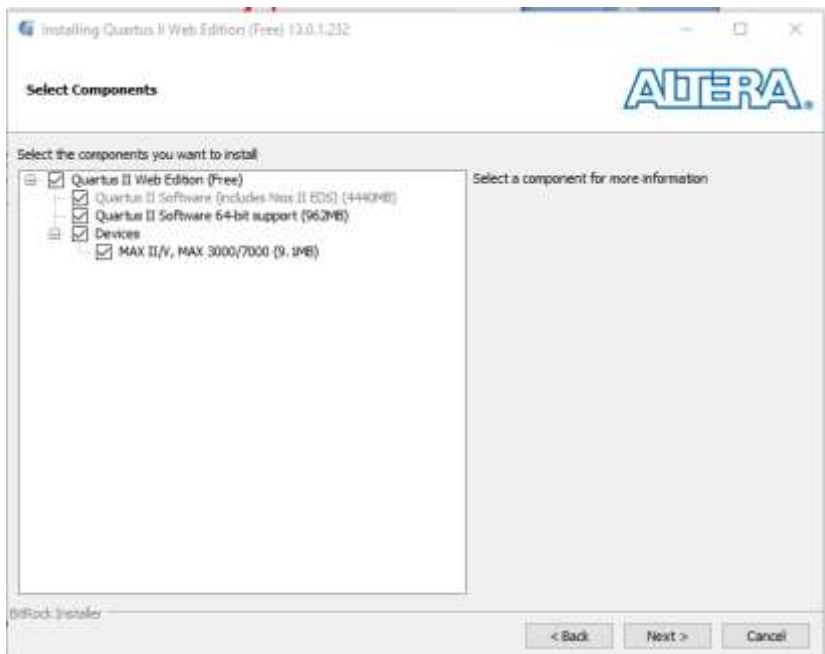


Рисунок 3.4 – Встановлення Quartus. Обрання додаткових компонентів програми

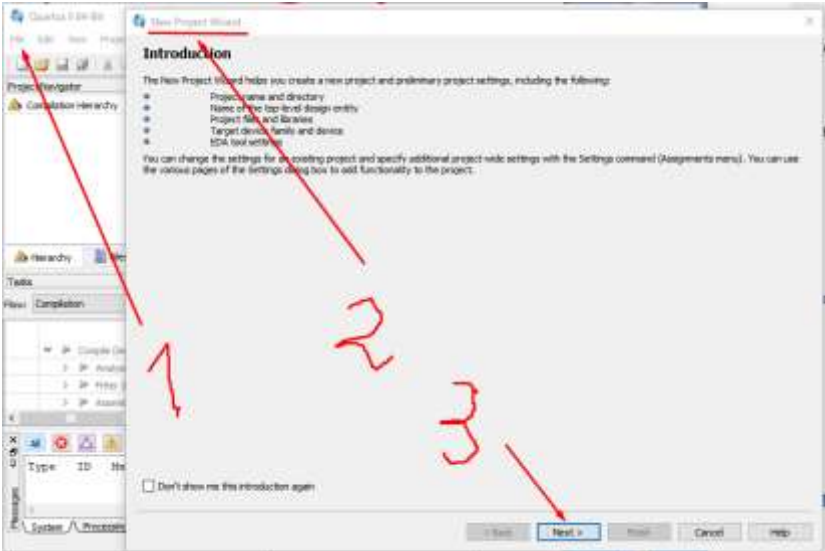


Рисунок 3.5 – Створення нового проекту

7.4 Для створення нового проекту необхідно:

- 1) Натиснути «File»;
- 2) Далі натиснути «New Project Wizard»;
- 3) У новому вікні натискаємо кнопку «Next» (рис 4.5);
- 4) Заповнити поля (рис 4.6), де перше – шлях проекту, друге – назва, третє - Top-level Entity проекту;
- 5) Знову натискаємо «Next»;
- 6) Натискаємо «Next» на «Page 2 of 5» і переходимо на «Page 3 of 5»;
- 7) Вибираємо сімейство MAX V і пристрій 5M1270ZT144C5, як показано на рисунку 4.7;
- 8) Натискаємо «Finish».

**Directory, Name, Top-Level Entity [page 1 of 5]**

What is the starting directory for this project?  
D:\Programs\Quartus\quartus\bin\4

What is the name of this project?  
Lab\_4

What is the name of the top-level design entity for this project? The name is case sensitive and must exactly match the entity name in the design file.  
lab-4

Use Existing Project Settings...

< Back   Next >   Finish   Cancel   Help

Рисунок 3.6 – Заповнення полів шляху проекту

**Family & Device Settings [page 3 of 5]**

Select the family and device you want to target for compilation.  
You can install additional device support with the Install Device command on the Tools menu.

**Device family**

Family: MAX V  
Device: 10K10K10K10

**Target device**

Auto device selected by the Filter  
 Specific device selected in Available device list  
 Other: ...

**Show in Available device list**

Package: Any  
Pin count: Any  
Speed grade: Any  
Name filter:   
 Show advanced devices    HardCopy compatible only

**Available devices:**

Name	Core Voltage	LUTs	LUT blocks
10K10K10K10-1	1.8V	1270	1
10K10K10K10-2	1.8V	1270	1
10K10K10K10-3	1.8V	1270	1
10K10K10K10-4	1.8V	1270	1
10K10K10K10-5	1.8V	1270	1
10K10K10K10-6	1.8V	1270	1
10K10K10K10-7	1.8V	1270	1
10K10K10K10-8	1.8V	1270	1
10K10K10K10-9	1.8V	1270	1
10K10K10K10-10	1.8V	1270	1

Completion device:

HardCopy:   
 List DSP & RAM in HardCopy device resources

< Back   Next >   Finish   Cancel   Help

Рисунок 3.7 – Вибір сімейства мікросхем для проектування

7.5 Для створення файлу програми в проєкті натискаємо «File -> New -> VHDL File», як показано на рисунку 4.8.

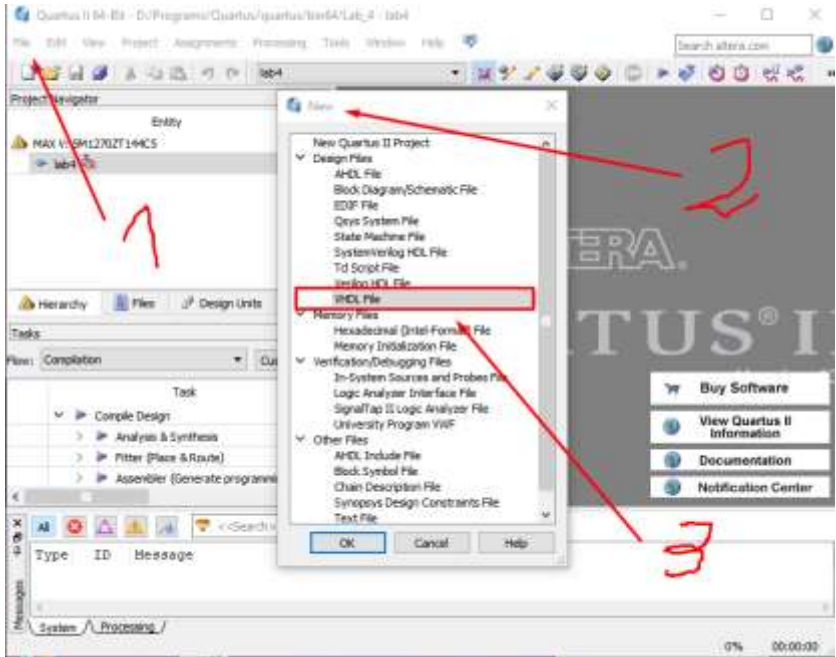


Рисунок 3.8 – Створення VHDL файлу проєкту

Лістинг 3.1

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.all;
```

```
entity lab4 is
```

```
port(
```

```
Reset: in STD_LOGIC; -- перемикач 2
```

```
CLK: in STD_logic; -- генератор частот
```

```
x0: in STD_LOGIC; --перемикач 0
```

```
x1: in STD_LOGIC; -- перемикач 1
```

```
led0: out std_logic_vector(7 downto 0); -- 7-сегментний
```

індикатор



```

        led1: out std_logic_vector(7 downto 0) -- 7-сегментный
индикатор
    );
end lab4;

```

```

architecture lab4 of lab4 is
signal D, Q: std_logic_vector(2 downto 0);
signal a0,a1,a2,a3,a4,a5,a6: std_logic;
begin
    process(CLK, Reset)
    begin
        if (Reset='1') then
            Q <= "000";
        elsif (CLK'event and CLK='1') then
            Q <= D;
        end if;
    end process;

    process(Q)
    begin
        a0 <= '0';
        a1 <= '0';
        a2 <= '0';
        a3 <= '0';
        a4 <= '0';
        a5 <= '0';
        a6 <= '0';
        case Q is
            when "000" => a0 <= '1';
            when "001" => a1 <= '1';
            when "010" => a2 <= '1';
            when "011" => a3 <= '1';
            when "100" => a4 <= '1';
            when "101" => a5 <= '1';
            when "110" => a6 <= '1';
            when others => a0 <= '1';
        end case;
    end process;
end lab4;

```

```

end process;

process(x0,x1,a0,a1,a2,a3,a4,a5,a6)
begin
    D(0) <= a0 or (a1 and not(x0)) or (a2 and x0) or (a3 and
not(x0)) or (a4 and x0) or (a5 and not(x0));
    D(1) <= (a1 and x0) or a2 or (a3 and not(x0)) or (a5 and
x0) or (a6 and not(x1));
    D(2) <= (a3 and x0) or a4 or a5 or (a6 and not(x1));

    if (a2 ='1' or a3 ='1' or a4 ='1' or a5 ='1' or a0 ='1')
then
    інвертований
        led1 <= not "01100011"; -- сигнал
        led0 <= not "00000000";
    elsif (a1 ='1') then
        led0 <= not"01100011";
        led1 <= not"00000000";
    elsif (a6='1') then
        led1 <= not"01011100";
        led0 <= not"00000000";
    end if;
end process;

end lab4;

```

В даному випадку для відображення вихідних сигналів на платі FPGA, використано 7-сегментні індикатори в якості поля для даної задачі.



Рисунок 3.9 – Приклад відображення результатів на індикаторі  
стенду

Навколо вихідного сигналу загораються по 4 світлодіоди для відображення вихідного сигналу, а інші вимкнені.

Далі необхідно виконати компіляцію проекту (рис 4.10).

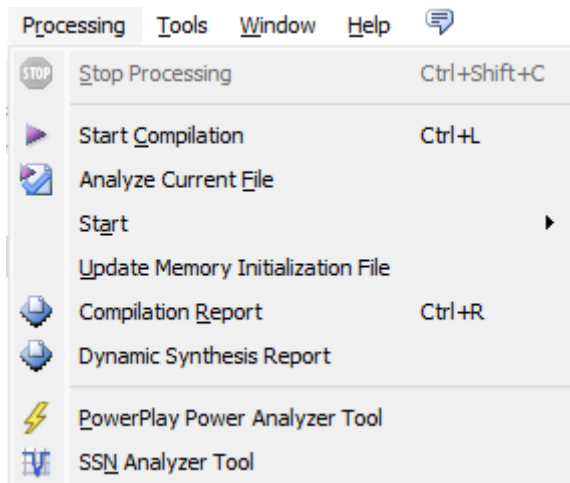


Рисунок 3.10 – Компіляція проекту

Якщо проект скопілювався без помилок, то в консолі буде повідомлення подібне до того, що на рис. 3.11.

```

Type   ID   Message
-----
3 332146 Worst-case setup slack is -2.354
3 332146 Worst-case hold slack is 1.375
3 332140 No Removable paths to report
3 332140 No Removal paths to report
3 332146 Worst-case minimum pulse width slack is -2.365
3 332001 The selected device family is not supported by the report_robustability command.
3 332102 Design is not fully constrained for setup requirements
3 332102 Design is not fully constrained for hold requirements
3 298000 Quartus II 64-Bit TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
3 298000 Quartus II Full Compilation was successful. 0 errors, 19 warnings

```

Рисунок 3.11 – Приклад результатів компіляції

Для плати MAX V – 5M1270Z CPLD Prototyping Board (табл.

4.1)

Name	Input/Output	Pin
Bargraph0[7]	Output	PIN_139
Bargraph0[6]	Output	PIN_138
Bargraph0[5]	Output	PIN_137
Bargraph0[4]	Output	PIN_134
Bargraph0[3]	Output	PIN_133
Bargraph0[2]	Output	PIN_132
Bargraph0[1]	Output	PIN_131
Bargraph0[0]	Output	PIN_130
Bargraph1[7]	Output	PIN_3
Bargraph1[6]	Output	PIN_2
Bargraph1[5]	Output	PIN_1
Bargraph1[4]	Output	PIN_144
Bargraph1[3]	Output	PIN_143
Bargraph1[2]	Output	PIN_142
Bargraph1[1]	Output	PIN_141
Bargraph1[0]	Output	PIN_140
Button[7]	Input	PIN_53
Button[6]	Input	PIN_57
Button[5]	Input	PIN_59
Button[4]	Input	PIN_61
Button[3]	Input	PIN_63
Button[2]	Input	PIN_67

Button[1]	Input	PIN_69
Button[0]	Input	PIN_71
Buzzer	Output	PIN_4
Clock	Input	PIN_18
Frequency	Input	PIN_20
FTDI_nCTS	Output	PIN_5
FTDI_nRTS	Input	PIN_6
FTDI_RXD	Output	PIN_7
FTDI_TXD	Input	PIN_8
GPIO[11]	Output	PIN_127
GPIO[10]	Output	PIN_73
GPIO[9]	Output	PIN_74
GPIO[8]	Output	PIN_75
GPIO[7]	Output	PIN_76
GPIO[6]	Output	PIN_77
GPIO[5]	Output	PIN_79
GPIO[4]	Output	PIN_94
GPIO[3]	Output	PIN_95
GPIO[2]	Output	PIN_106
GPIO[1]	Output	PIN_107
GPIO[0]	Output	PIN_129
Hex0[7]	Output	PIN_84
Hex0[6]	Output	PIN_88
Hex0[5]	Output	PIN_89
Hex0[4]	Output	PIN_91
Hex0[3]	Output	PIN_93
Hex0[2]	Output	PIN_85
Hex0[1]	Output	PIN_86
Hex0[0]	Output	PIN_87
Hex1[7]	Output	PIN_96
Hex1[6]	Output	PIN_102
Hex1[5]	Output	PIN_103
Hex1[4]	Output	PIN_104

Hex1[3]	Output	PIN_105
Hex1[2]	Output	PIN_97
Hex1[1]	Output	PIN_98
Hex1[0]	Output	PIN_101
Hex2[7]	Output	PIN_108
Hex2[6]	Output	PIN_112
Hex2[5]	Output	PIN_113
Hex2[4]	Output	PIN_114
Hex2[3]	Output	PIN_117
Hex2[2]	Output	PIN_109
Hex2[1]	Output	PIN_110
Hex2[0]	Output	PIN_111
Hex3[7]	Output	PIN_118
Hex3[6]	Output	PIN_122
Hex3[5]	Output	PIN_123
Hex3[4]	Output	PIN_124
Hex3[3]	Output	PIN_125
Hex3[2]	Output	PIN_119
Hex3[1]	Output	PIN_120
Hex3[0]	Output	PIN_121
HexCode0[3]	Input	PIN_52
HexCode0[2]	Input	PIN_50
HexCode0[1]	Input	PIN_51
HexCode0[0]	Input	PIN_49
HexCode1[3]	Input	PIN_48
HexCode1[2]	Input	PIN_44
HexCode1[1]	Input	PIN_45
HexCode1[0]	Input	PIN_43
Incremental[1]	Input	PIN_80
Incremental[0]	Input	PIN_81
Switch[7]	Input	PIN_55
Switch[6]	Input	PIN_58
Switch[5]	Input	PIN_60

Switch[4]	Input	PIN_62
Switch[3]	Input	PIN_66
Switch[2]	Input	PIN_68
Switch[1]	Input	PIN_70
Switch[0]	Input	PIN_72

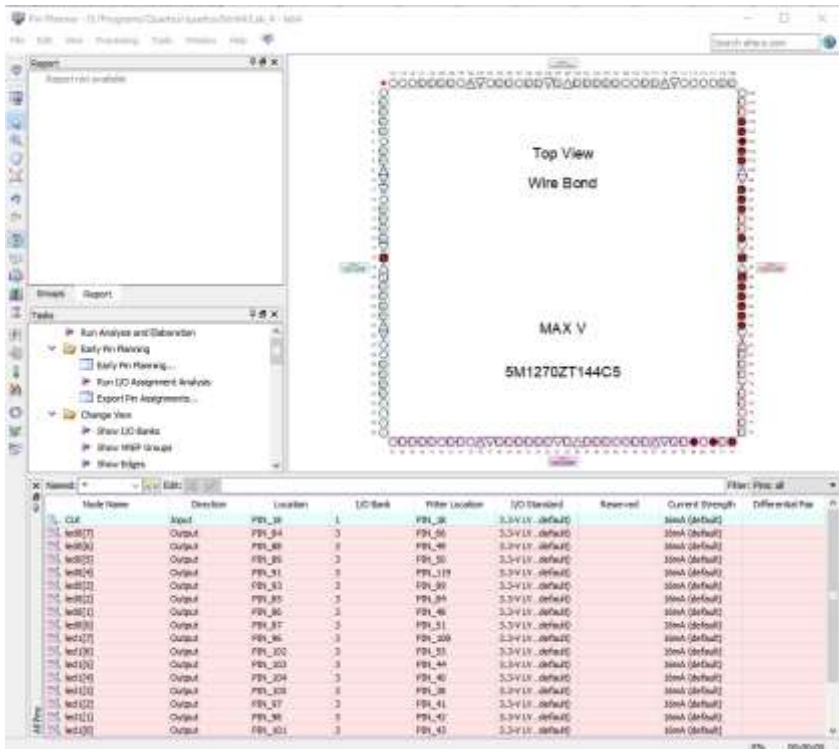


Рисунок 3.12 – Призначення контактів сигналів мікросхеми (пінів)

Зеленим кольором в таблиці виділені використані пін.

Вхідні сигнали - за допомогою перемикачів 0 та 1, а Reset – перемикач – 2.

Clock – до 18 порту, який являється генератором частоти на платі.

Вихідні сигнали – два 7-сигментні світлодіоди Hex0 та Hex1

Після ініціалізації портів та компіляції в каталозі проєкту з'являється каталог `output_files` в якому є файл з розширенням `.prof` - двійковий файл що містить дані для програмування пристрою MAX® II або MAX® V, або пристрою конфігурації, який може конфігурувати інші пристрої (рис. 4.13).

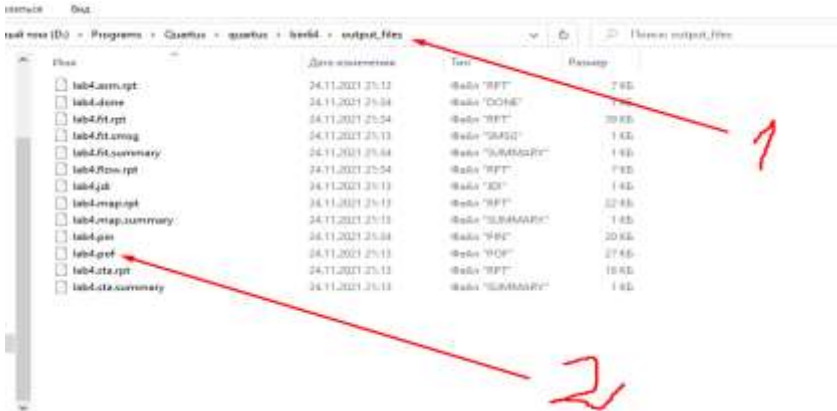


Рисунок 3.13 – Файл конфігурації

Завантажимо файл з розширенням `.prof` в онлайн лабораторію Goldi для симуляції роботи плати швидкого прототипування MAX V – 5M1270Z CPLD.



Рисунок 3.14 – Завантаження файлу



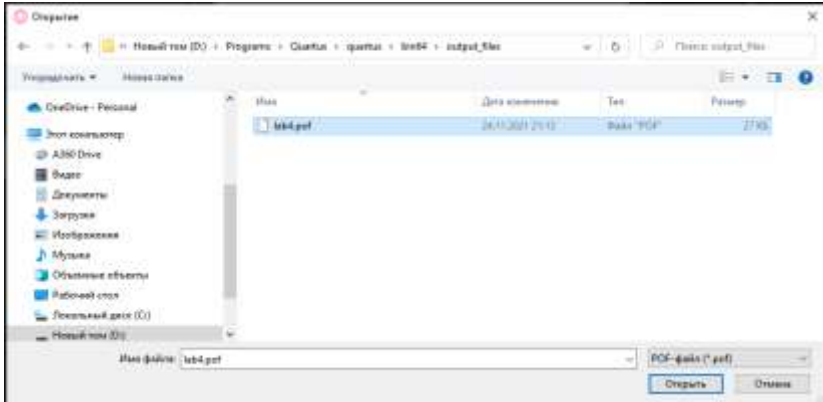


Рисунок 3.15 – Вибір файлу

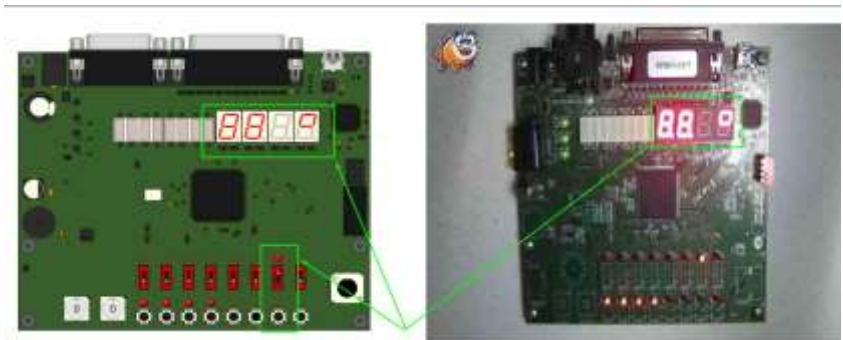


Рисунок 3.16 – Результат тестування 1



Рисунок 3.17 – Результат тестування 2



Рисунок 3.18 – Результат тестування 3

Те, що зображено на рисунках 3.16 – 3.18 повністю відповідає графу на рис. 3.1 та рис 3.9. Адже спочатку модель приймає значення  $y1$  (права верхня комірка) при  $a1$  (рис 3.16). При  $a2$ ,  $a3$ ,  $a4$  або  $a5$  модель матиме значення  $y0$ , тобто верхня ліва комірка (рис 3.17). А при  $a6$  модель матиме значення вихідного сигналу  $y2$ , тобто нижня ліва комірка (рис 3.18).

### ВИСНОВОК

У результаті віддаленого тестування плати в лабораторії Goldi перевірена робота FSM моделі з описом її апаратної реалізації на VHDL.

В ході роботи розроблений опис станів автомата Мура, визначена розрядність коду, закодовані стани. Побудовано граф-схему алгоритму та заповнено таблицю переходів автомата Мура із цієї граф-схеми. Отримані формули для функцій переходів з таблиці. Після цього створений проект в середовищі Quartus II та описаний автомат на мові VHDL, виконано ініціалізацію портів. Далі виконано скомпіляцію проекту, і переконавшись, що немає помилок, протестувано автомат Мура на реальній платі швидкого прототипування фірми Altera\Intel.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Основи надійності цифрових систем : підручник / В.С. Харченко, В.Я. Жихарев, В.М. Ілюшко, В.А. Краснобаєв, П.М. Куліков, І.В. Лисенко, М.В. Нечипорук, Г.М. Тимонькін. – Харків: Нац. аерокосм. ун-т "Харк. авіац. інт", 2017. – 573 с.
2. Віддалений та віртуальний інструментарій в інжинірингу : монографія / А.В. Пархоменко, Г.В. Табунщик, М.О. Поляков, О.М. Гладкова, Т.І. Каплієнко, Т.Ю. Ларіонова; за заг. ред. Карстена Хенке. – Запоріжжя: Дике Поле, 2015. – 250 с.
3. Верифікація та валідація цифрових систем управління / Г.В. Табунщик, Т.І. Каплієнко, О.О. Каплієнко, Д.Ван Мероде – Запоріжжя : Дике Поле. - 2017. – 150 с.

## ДОДАТОК А

### ІНСТРУКЦІЯ З ПІДКЛЮЧЕННЯ ТА ПРИКЛАД ВИКОРИСТАННЯ ПАКЕТУ ВІДДАЛЕНОЇ ЛАБОРАТОРІЇ GOLDI

Для доступу до віддаленої лабораторії GOLDi, переходимо на сайт <https://goldi-labs.net/>. На рисунку А.1 можна побачити головну сторінку сайту.



Рисунок А.1 – Головна сторінка віддаленої лабораторії GOLDi у браузері

Далі треба створити новий обліковий запис користувача. Для цього потрібно натиснути кнопку «Register». Після чого відкриється форма реєстрації облікового запису (рис. А.2 – А.3).



Рисунок А.2 – Кнопка реєстрації облікового запису

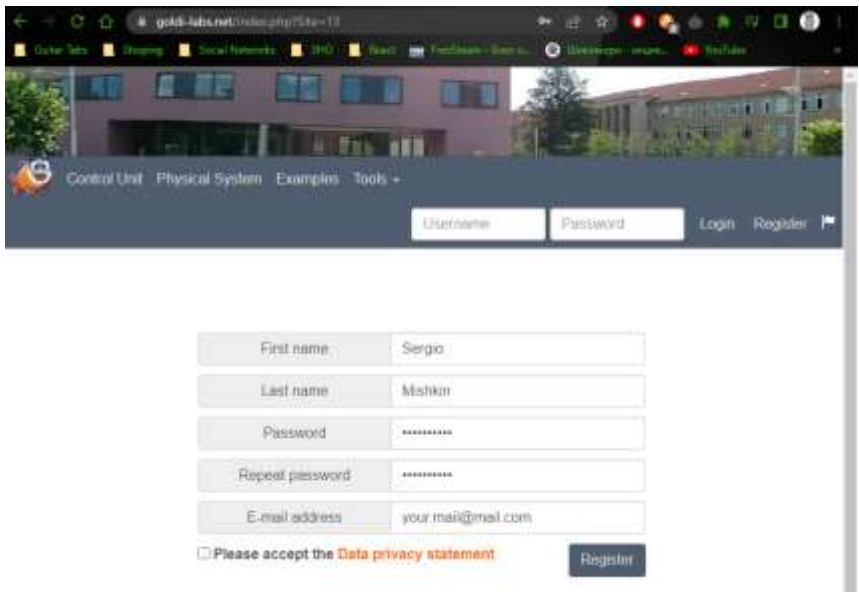


Рисунок А.3 – Форма реєстрації облікового запису

Після реєстрації облікового запису, треба увійти в нього та перейти до вкладки Experiment. Після чого відкриється сторінка «Experiment Locations» де потрібно обрати одну з віддалених лабораторій. Обираємо доступну на даний момент. У моєму випадку це «IUT (Germany)» (рис. А.4).

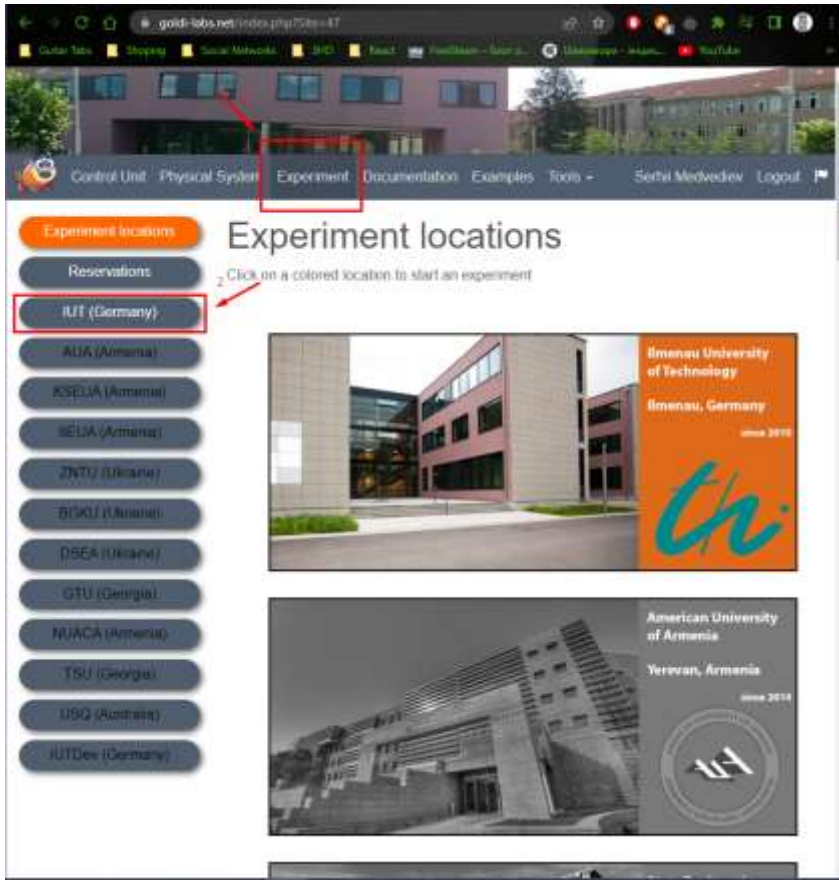


Рисунок А.4 – Вибір віддаленої лабораторії для подальшої роботи

Після цих дій, відкриється список доступних пристроїв керування та фізичних/віртуальних систем. Обираємо пристрій керування «Finite State Machine» та одну з віртуальних систем.