

Хмельницький національний університет  
Факультет програмування  
та комп'ютерних і телекомунікаційних систем  
Кафедра кібербезпеки та комп'ютерних систем і мереж

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Системана програмованій логіці для керування роботою автостоянки  
Назва теми


КВРКІ.170283.17.02.14 ІІЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

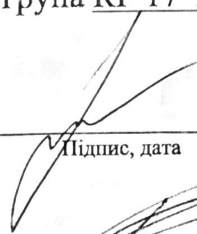
Освітня програма «Комп'ютерна інженерія»  
Назва

Виконав: студент IV курсу, група КІ-17-2

  
Підпис


Д.В. Окрушко  
Ініціали, прізвище

Керівник

  
Підпис, дата

В.Ю. Тітова  
Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

І.В. Муляр  
Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри кібербезпеки та  
комп'ютерних систем і мереж

  
Підпис

Ю.І. Кльоб  
Ініціали, прізвище

« 07 » червня 2021 р.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Завідувач кафедри Ю.П.Кльоц



“ 05 ” 02 2021 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Окрушко Дмитру Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система на програмованій логіці для керування роботою автостоянки

Керівник проекту (роботи) Тітова Вера Юріївна

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

кандидат технічних наук, доцент

Затверджена наказом ректора університету від 05.02.2021 № 11 додаток № 7

2. Строк подання студентом проекту (роботи) на кафедру 28.05.2021

3. Вихідні дані до проекту (роботи) ПЛІС ATMEGA Altera Cyclone 5 DE1-Soc, налаштування – з ПК через USB-порт

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Дослідження предметної області та постановка задачі; обґрунтування базових положень щодо проектування автоматизованої системи керування роботи автостоянки. Опис схем електричних (структурної) проектованої системи; опис алгоритму роботи системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

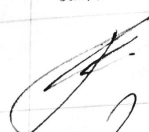
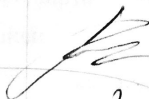


Схема електрична структурна (E1)

Схема стоянки (E8)

Результати роботи (E8)

Алгоритм роботи (E8)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Муляр І.В., доцент кафедри КБКСМ		
Антиплагіат	Муляр І.В., доцент кафедри КБКСМ		

7. Дата видачі завдання « 08 » 02 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1.	Підготовка вступного розділу	Березень - 1 декада	
2.	Огляд існуючих методів, засобів	Березень - 2 декада	
3.	Обґрунтування обраних рішень	Березень - 3 декада	
4.	Підготовка опису електричних схем	Квітень - 1 декада	
5.	Виконання розрахункової частини	Квітень - 1 декада	
6.	Підготовка ескізів креслень	Квітень - 2 декада	
7.	Формулювання висновків	Квітень - 3 декада	
8.	Розробка додатків	Травень - 1 декада	
9.	Погодження розділів з консультантом з нормоконтролю	Травень - 1 декада	
10.	Оформлення графічного матеріалу	Травень - 2 декада	
11.	Оформлення пояснювальної записки	Травень - 2 декада	
12.	Попередній захист кваліфікаційної роботи	Травень - 3 декада	
13.	Доопрацювання кваліфікаційної роботи	Травень - 3 декада	
14.	Подання роботи для перевірки на плагіат	Травень - 3 декада	
15.	Захист кваліфікаційної роботи	Червень - 1 декада	

Студент

Керівник проекту (роботи)

  
Підпис

  
Підпис

Окрушко Д.В.  
Ініціали, прізвище

В.Ю. Тітова  
Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система на програмованій логіці для керування роботою автостоянки».

Автор роботи: Окрушко Дмитро Віталійович.

Керівник роботи: Тітова Вєра Юріївна.

Пояснювальна записка: 72 с., 62 рис., 1 табл., 1 дод., 20 джерел.

Графічна частина: 3 плакати.


ПЛІС, СИСТЕМА УПРАВЛІННЯ, АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ, ДАТЧИКИ РУХУ, СИСТЕМА ПІДРАХУНКУ КІЛЬКОСТІ АВТО, АВТОМАТИЗОВАНА СИСТЕМА ДЛЯ КЕРУВАННЯ РОБОТОЮ АВТОСТОЯНКИ.

Під час роботи над проектом було досліджено процес керування роботою автостоянки на базі програмованих логічних інтегральних схем.

Також було звернено увагу на автоматичну систему підрахунку кількості авто на стоянці і інформування водіїв про кількість вільних місць. Використовував методи дослідження такі як метод синтезу та аналізу.

**Мета кваліфікаційної роботи:** була розробка та моделювання роботи на платі системи на програмованій логіці для керування роботою автостоянки.

Матеріали кваліфікаційної роботи можна використовувати для створення справжньої автоматичної системи керування автостоянкою.

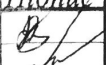



  
Підпис студента

04.06.2021  
Дата



## ЗМІСТ

ВСТУП.....	3
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	5
1.1 Аналіз предметної області і виявлення наявних проблем і завдань.....	5
1.2 Порівняльний аналіз переваг та недоліків наявних рішень.....	9
1.3 Використання ПЛІС в задачах автоматизації. Переваги та недоліки.....	26
1.4 Постановка задачі.....	36
2 ПРОЕКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ.....	38
2.1 Методологічні підходи до вирішення задачі за темою дослідження.....	38
2.2 Вибір апаратної бази.....	45
2.3 Висновки до другого розділу.....	55
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ.....	57
3.1 Алгоритм роботи.....	57
3.2 Отримані результати.....	62
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	69
Додаток А Лістинг програми на мові VHDL.....	71
Додаток Б Копія графічної частини Алгорит роботи системи.....	80
Додаток В Копія графічної частини Структурна схема.....	81
Додаток Г Копія графічної частини Результати роботи.....	82

				<i>КВРКІ.170283.17.02.14 ПЗ</i>			
Зм.	Арк.	№ докум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконала		Окрушко Д.В.				2	70
Перевір.		Тітова В Ю					
Н.контр.		Муляр І.В					
Затвер.		Кльоц Ю.П.					
Система для керування роботою автостоянки Пояснювальна записка					ХНУ КІ-17-2		

## ВСТУП

Сучасні автостоянки не мають єдиної системи для моніторингу вільних місць для паркування. Наявні системи дуже сильно залежать від людини, це призводить до нерівномірного заповнення стоянок, до збільшення часу на пошук вільних місць, а це у свою чергу призводить до більших викидів CO<sub>2</sub> у повітря і великих пробок, що також впливає в цілому на транспортну систему міста. Корпорація IBM провела глобальне дослідження автомобільних стоянок (IBM Global Parking Survey) [1]. Дослідження показало, що водії як в розвинених, так і в країнах, що розвиваються стикаються з однією і тією ж незадовільною ситуацією і проблемами паркування. Було опитано 8042 автомобіліста в 20 містах світу на шести континентах. Результати опитування вказують, що водії щодня докладають великих зусиль для пошуку вільного місця для паркування. У минулому році майже 6 з 10 опитаних водіїв принаймні один раз були змушені відмовитися від пошуку місця для тимчасової стоянки свого автомобіля і поїхати в інше місце, і понад чверть респондентів вступали в суперечку з іншими водіями за місце для паркування.

У дослідженні також відзначається, що поряд зі звичайними пробками більше 30% з них виникають через те, що водії створюють перешкоди на дорозі в процесі пошуку місця для паркування. Неефективні системи організації автостоянок призводять до перевантаженості на дорогах і збільшення викидів вихлопних газів, через них також марнується час водіїв і пасажирів, знижується ефективність праці і губляться економічні можливості.

Проблеми пробок і автостоянок досягли кризового рівня в усьому світі, вважають в IBM, а викиди вихлопних газів автомобілів ще більше посилюють несприятливу екологічну ситуацію. Всі ці фактори негативно впливають на життя людей в усьому світі, де державні органи, приватний сектор і громадськість шукають нові ефективні засоби, в тому числі будівництво

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	<i>Арк.</i>
						3
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

нових доріг, для пом'якшення і усунення негативних наслідків від завантаженості доріг.

Індустрія автостоянок унікальна — немає ніякої іншої галузі, де більше 50% доходів надходить від штрафів (для водіїв, які порушують правила паркування). Причина в тому, що ця зріла індустрія навіть 2-3 роки тому ще використовувала застарілу бізнес-модель: з величезною різницею в ціні між відкритою і підземною автостоянкою, оплатою тільки готівкою, поганими знаками розмітки і т.д. І в міру того, як населення міст з плином часу збільшувалася, проблема автостоянок ставала все більш критичною. Таким чином, поява мобільних платіжних рішень, технологій інтернету речей (Internet of Things) стало ковтком свіжого повітря для галузі, оскільки їх використання дозволяє кардинально поліпшити досвід паркування клієнтів.

Переваги використання рішення для автостоянки на базі інтернет речей:

- Збільшення доходів від послуг паркування для муніципалітетів і бізнесу.
- Краще управління дорожнім рухом (менше пробок).
- Зниження викидів CO<sub>2</sub>.
- Скорочення часу, що витрачається водіями на автостоянку (час на пошук вільного місця і на оплату паркування).

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	<i>Арк.</i>
						4
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Сьогодні автостоянка є невід'ємною складовою транспортної системи. Це відіграє велику і вирішальну роль в управлінні дорожнім рухом та заторами. Автостоянка на вулицях Хмельницького є однією з основних проблем, яка робить ситуацію з дорожнім рухом хаотичною на наших околицях міста. Більшість доріг у місті Хмельницький вузькі і з малою кількістю пішохідних смуг та зебр. Усім відомо, що транспортні засоби широко використовуються для різного типу перевезення. Також, важливо мати ефективну, просту, надійну систему підтримки чистоти транспортних засобів, що також потрібно вирішити.

Проблема місця для паркування автомобілів величезна, що зробить цю роботу актуальною через не правильно організовану систему паркування, недостатню кількість місць для паркування та відсутність кваліфікованих охоронців. Існує загальносвітовий і глобальний перехід до автоматичної системи паркування автомобілів, щоб розрахувати точний доступний простір для автомобілів та для збору доходів як плата за паркування.

Але основним складним завданням створенням єдиної глобальної системи керування це те що, на даний момент у світі не існує якогось загального стандарту в програмних та мережних засобах при побудові спеціалізованих систем.

Автостоянка – це дещо більше, ніж просто місце стоянки автомобілів. Автоматизована система паркування має:

- Дозволяти проводити автоматичний облік кількості вільних та зайнятих місць

					КВРКІ.170283.17.02.14 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

- Забезпечувати контроль доступу автомобілів по талонах або якимось іншим чином.
- Спростувати контроль грошових розрахунків, що фактично виключає ймовірність крадіжок, значно полегшує роботу персоналу та знижує залежність роботи установи від людського фактору, таким чином вирішується складна задача керування автостоянкою.

Автоматизовані системи паркування на даний момент створенні для вирішення задачі автоматичного контролю, ефективного керування, швидкого обслуговування клієнтів та безпечного регулювання постійного руху транспортних потоків транспортних засобів міста Хмельницького.

Також автоматизовані системи паркування дозволять якісно та швидко надати послуги платного керування клієнтами, принести задоволення від паркування та завоювати довіру відвідувачів, забезпечити високу рентабельність власникам. Ще така система дозволить значною мірою зменшити затрати часу і пального під час пошуку час та і бронювати місце на конкретно вказаний час.

Розгортання такої системи на перший погляд може здаватися не вигідним для власників автостоянок або для державних підприємств, бо для цього потрібно кваліфіковані мережні інженери та дорого-вартісне спорядження тощо. Алгоритм для вирішення задачі оптимально використати наявні ресурси зможе значною мірою автоматизувати процес, дозволивши витратити менше часу під час проектування мережі для фахівців, і також зменшити затрати на розгортання і встановлення самої системи.

Ціль роботи є надання повного контролю для забезпечення вільного простору для автостоянок за допомогою сучасних ПЛІС, датчиків руху та інших систем. Також показати що за допомогою системи для автостоянок стане можливим підвищити ефективність і зручність автостоянок не витративши на це багато людських та технічних ресурсів.

					КВРКІ.170283.17.02.14 ПЗ	Арк. 6
Зм.	Арк.	№ докум.	Підпис	Дата		

Потрібно також провести дослідження і розробити алгоритм для вирішення оптимальної задачі розташування сенсорів в системі автостоянки. І також потім провести тестування алгоритму на різних схемах автостоянок з перешкодами для розповсюдження сигналу. Тому будь яка запропонована наявна система з автоматизованою системою автостоянки має відповідати таким вимогам:

- Спрощення самого процесу паркування для автомобілів різного типу
- Надання інформації водію щодо наявних місць на автостоянці.
- Вміти надавати водію всі корисну для нього інформацію(чи вільні сусідні місця для паркування, скільки залишилося вільних місць на інших стоянках тощо )
- Щоб система могла приймати рішення про перенаправлення водія на іншу стоянку, тобто щоб система могла приймати сама рішення
- Здатність по штучно сприймати кількість автомобілів в режимі реального часу.

Оскільки ці проблеми мають бути вирішені на самому початку, щоб забезпечити безперебійну роботу системи потрібно для досягнення поставленої мети необхідно вирішити наступні завдання:

- Вивчити особливості отримання сигналу від датчиків
- Вивчити доступні в продажі ПЛІС та їх аналоги
- Придумати як будуть взаємодіяти елементи системи автостоянки між собою
- Спроекувати архітектуру ПЗ
- Визначитись з фізичним розташуванням датчиків і їх взаємодію з ПЛІС та як і що підключити
- Розробити відповідне програмне забезпечення для аналізу і обробки сигналів
- Дослідити роботу семи сегментних індикаторів або інших дисплеїв

					КВРКІ.170283.17.02.14 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

- Розробити програмне забезпечення для забезпечення місць на автостоянці
- Дослідити особливості програмування на VHDL
- Виконати проект
- Провести належне тестування розробленого програмного забезпечення і всієї системи а також зібрати отримані дані та дослідити справжню а не прогнозовану ефективність розробленої системи стоянки.

Вхідними даними для вирішення цих задач буде інформація про доступні стоянки, системи автостоянок, аналоги таких систем. Також інформація отримана від адміністраторів автостоянок, з якими проблемами вони стикаються і можливі шляхи вирішення на їх погляд. Ще інформація про кількість доступних місць для стоянки яка буде надходити з датчиків які будуть тимчасово розташовані на стоянці.

Вихідними даними буде інформація з датчиків та індикаторів. Тобто це кількість доступних місць на стоянці або ступінь завантаженості стоянки.

А об'єктом дослідження буде комплекс різноманітних інформаційних систем для контролю та моніторингу простору автостоянки. Методи будуть включати в себе:

- Різноманітний аналіз, систематизація та узагальнення отриманих даних про наявні розроблені способи автоматизації роботи стоянок комплексів.
- Аналіз алгоритмів та підходів, які дозволять провести моніторинг за місцями автостоянки
- Та різні експерименти та дослідження з розпізнавання тестових прикладів.

Основним напрямком розвитку будуть smart датчики для паркування. Ці датчики зазвичай вбудовуються в дорогу або полотно де має паркуватися авто і відстежують вільне чи зайняте місце і потім ці дані в реальному часі передаються на ядро системи. Використовуючи цілу мережу таких датчиків

					КВРКІ.170283.17.02.14 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

буде створюватися мапа місць для стоянки і цілих стоянок. Стан яких буде виводитись на великі дисплеї або в додаток для смартфона.

Існує дуже багато способів як можна покращити роботу цих датчиків. Наприклад, для закритих або навісних стоянок є розробка і інтеграція автоматизованих стоянок, де водія майже нічого не потрібно робити. По суті водій буде просто заїжджати на місце для стоянки, паркуватися, і на цьому його дії закінчатся. А система в час як тільки водій заїде буде знати що ще одна машина заїхала на стоянку і одразу виведе інформація про це на загальне табло і повідомить про це систему автостоянки.

Або ще можна використати ПЛІС (електронні компоненти або елементи, які використовують для створення цифрових інтегральних схем). Оскільки кафедра має у власності ПЛІС ATMEGA Cyclone 5 вибір буде в їх сторону.

Отож основною задачею розробленої системи має бути зменшення часу для пошуку вільних місць, а зменшення часу на пошук впливає на загальну завантаженість транспортної системи міста Хмельницький і взагалі міст України (оскільки система має бути максимально гнучкою для інтеграції у будь якого місті будь то СМТ чи мегаполіс) та на рівень забруднення вихлопними газами повітря. Оскільки економія палива і зменшення вихлопних газів це один із плюсів і кроків до збереження навколишнього середовища.

## 1.2 Порівняльний аналіз переваг та недоліків наявних рішень

Із постійним розвитком сучасних технологій, використання комп'ютерів у будь-якій сфері життя людей та все більшому поширенню можливостей вільно, якісно витратити власний час, проблема для контролю або моніторингу кількості транспортних засобів і вільних місць стоянки для них стає все більш актуальнішою.

Якщо подивитись на дані 2001 року середня кількість авто на одну тисячу осіб населення складала близько 50 авто. Проте, на сьогоднішній день,

					КВРКІ.170283.17.02.14 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

кількості авто на одну тисячу населення складає вже приблизно дві сотні автомобілів. Зважаючи на це з'являється необхідність створення нових місць для паркування, а можливо і відкриття нових автостоянок. При такій великій кількості авто і автостоянок є важливим аспектом використовувати єдину систему для обліку, пошуку та бронювання місць на стоянці.

Сьогодні на ринку представлені різноманітні системи і підходи для вирішення проблеми. Але все таки відсутня система де можна було б отримати інформацію про всі наявні і доступні автостоянки міста, чи хоча б певну якусь з них щоб можна було забронювати місце.

Розглянемо такі до прикладу такі системи автостоянок

- “Simple Parking”
- “Vector AP 2000”
- Yandex parking
- “All parking”
- Автостоянка в Борисполі
- Автостоянка 2.5.5
- PARKEON
- AS101 Pro Park
- Аксіома груп “Розумна стоянка”

Усі ці системи були розроблені для контролю та автоматизації процесу в'їзду та виїзду машин з стоянок і з можливістю бронювання місць на автостоянці.

Розглянемо усі ці системи детальніше:

#### 1. “Simple Parking” [2]

Вигляд системи зображено на рисунку 1.1

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	<i>Арк.</i> <i>10</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

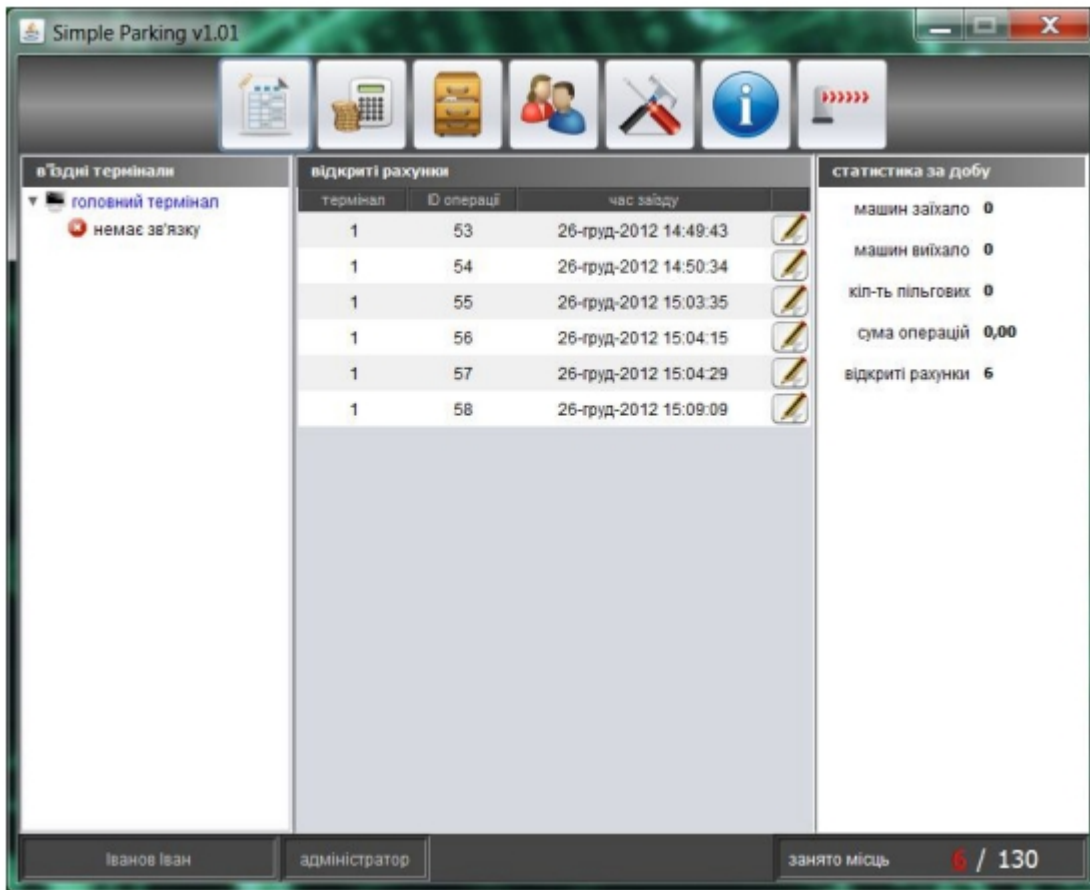


Рисунок 1.1 - Система "Simple Parking"

У цієї системи доступний зручний інтерфейс користувача, що дозволяє легко розібратися як користуватися нею. У системі доступний перегляд схеми автостоянки, зображення самої схеми автостоянки з доступом до вільних місць, Можна переглянути які і де місця вже зайняті. Можна також переглянути розташування конкретного транспортного засобу.

## 2. "Vector AP 2000" [3]

Вигляд системи на Рисунку 1.2





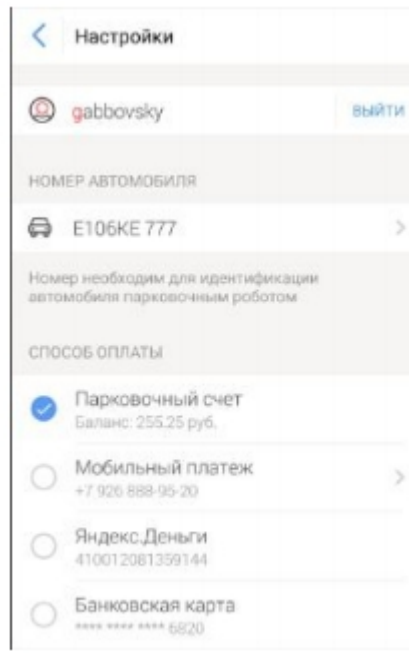


Рисунок 1.5 - Разные способы оплаты за место на стоянке

Ще для зручності користувача є функція для оплати місця на стоянці онлайн, що в наш час дуже зручно і вважається безпечно. Також є функція яка допомагає дивитися і аналізувати фактичний час перебування авто на стоянці і також показати скільки ще залишилося часу перебування на стоянці. Також якщо водій десь відійшов на прогулянку на карті буде зображено де знаходиться його авто.

					КВРКІ.170283.17.02.14 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

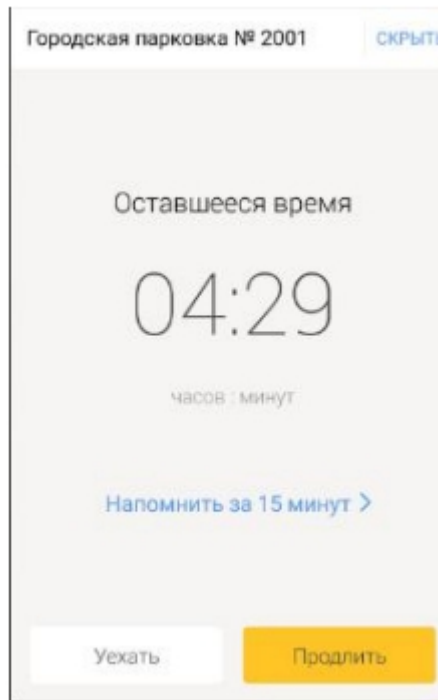


Рисунок 1.6 - Секундомір у системі

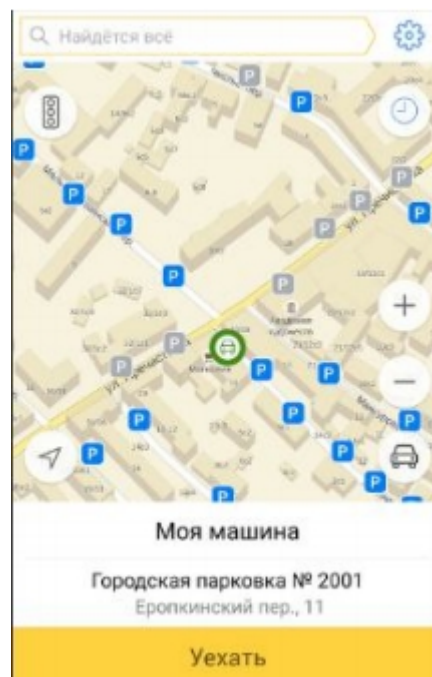


Рисунок 1.7 - Пошук власного авто на карт

#### 4. "All parking" [3]

Як виглядає система зображено на рисунку 1.8

					КВРКІ.170283.17.02.14 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		





Рисунок 1.9 - Програма PARKEON

PARKEON це один із найбільш популярних сервісів. Він є в більш ніж 4000 містах і вже в 60 мегаполісах світу, і цей сервіс пропонує нові розумні транспортні рішення і рішення для сучасних стоянок.

Отже як працює ця система:

1) В'їзд

- Машина зупиняється перед шлагбаумом
- Датчик фіксує що присутня машина, а відповідна камера фіксує і автоматично розпізнає і записує номер авто, дату, і фактичний час в'їзду до системи автостоянки.
- Система управління отримує дані, і додає їх до власної бази даних
- Якщо клієнт вже є у базі, то також додається інформація про дату резервування або членство.
- Потім шлагбаум підіймається, дозволяючи водієві потрапити до стоянки. І потім він автоматично опускається.

2) Виїзд

- Машина зупиняється перед виїздом де стоїть шлагбаум
- Датчик фіксує що під'їхала машина, а камера фіксує номер машини і звіряє інформацію про платіж у власній базі даних.

					КВРКІ.170283.17.02.14 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

- Система перевіряє статус і чи коректно пройшов платіж або дозвіл на безоплатну стоянку якщо такий був
- Шлагбаум підіймається, цим самим він дозволяє водієві покинути стоянку.
- І потім автоматично закривається як тільки машина покидає зону датчика.

### 6. AS 101 Pro Park [12]

На рисунку 1.10 і 1.11 зображено сучасну систему “Pro Park”



Рисунок 1.10 - Приклад реалізованої вже системи

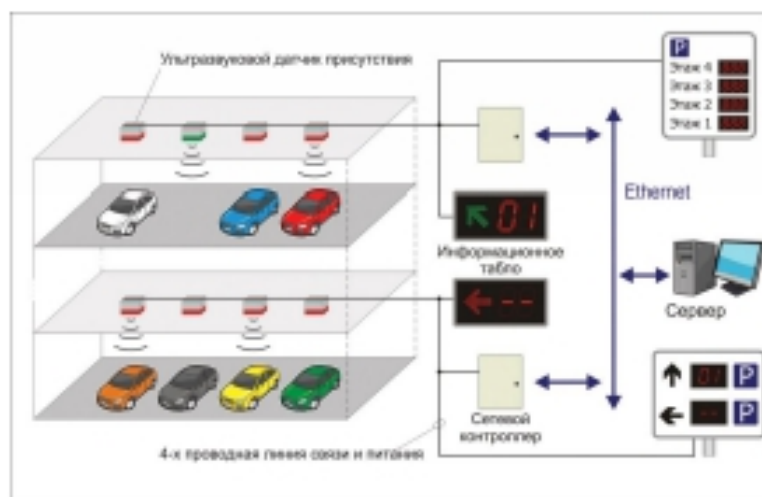


Рисунок 1.11- схема зображення Pro Park

Принцип роботи цієї крутої системи полягає в тому що система точно визначає розташування вільних і зайнятих місць і також підрахунок кількості

машин що в'їхали або виїхали. Світлові індикатори кожного місця стоянки і табло показує водію усі доступні місця і оптимальний маршрут до зазначеного місця. Система в ріал-тайм здійснює контроль завантаженості з наданням адміністрації усієї інформації. Система виконана на базі AsS101 Pro і забезпечує високу якість та надійність, зручність і простотою у використанні. Під час розробки цього проекту було розроблено нові пристрої.

### 7. Стоянка в Борисполі

На рисунку 1.12-1.14 і зображено систему “Стоянка в Борисполі”

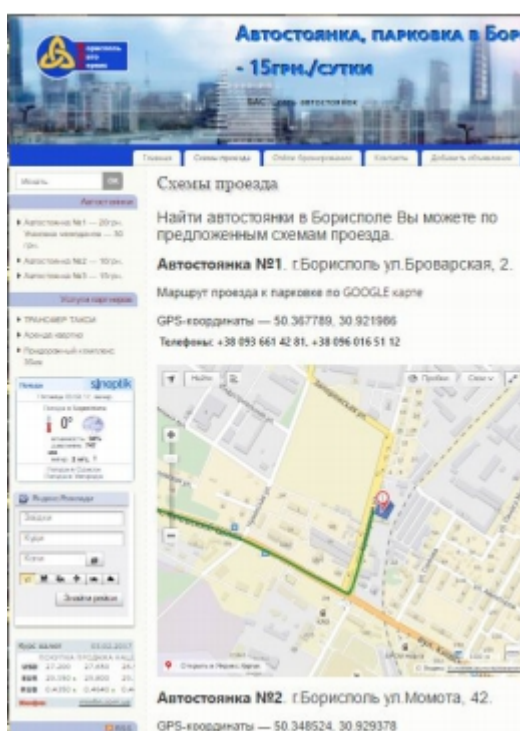


Рисунок 1.12 - Загальний вигляд

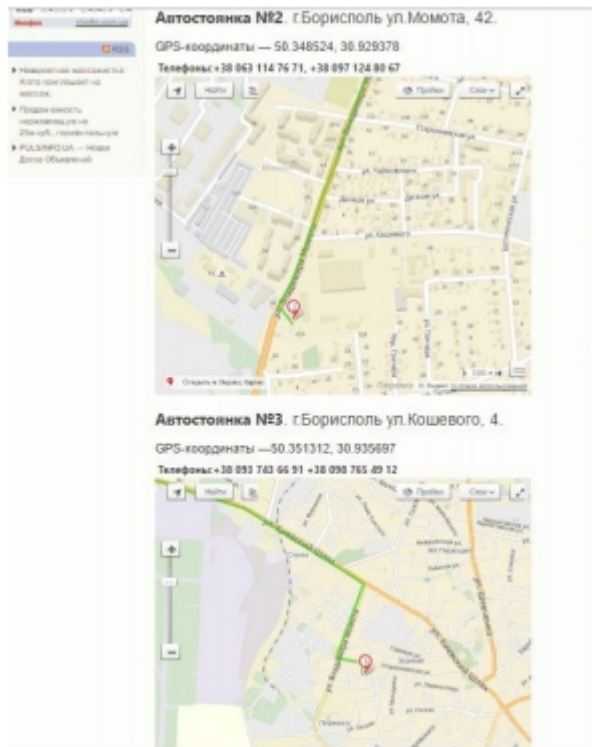


Рисунок 1.13 - Схема проїзду

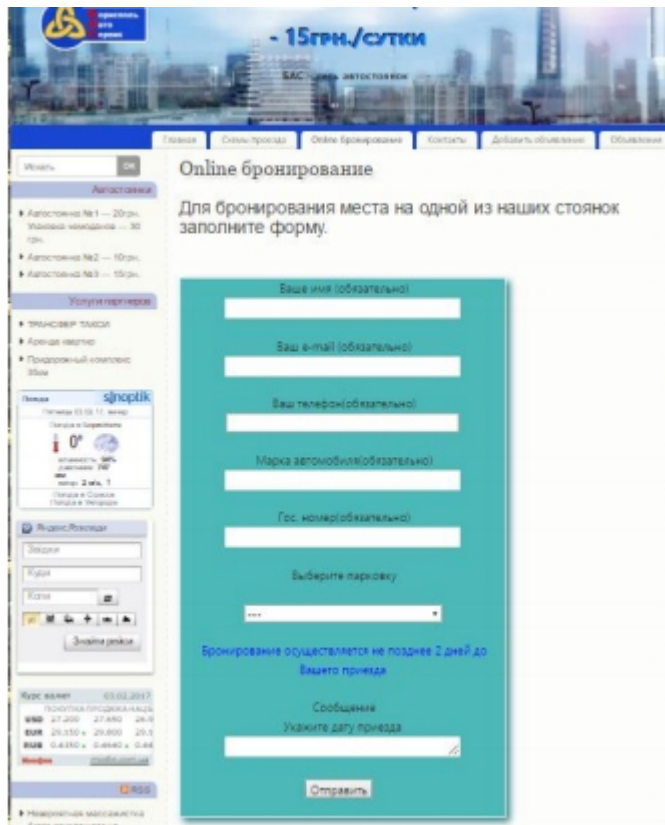


Рисунок 1.14 - Можливість онлайн бронювання

Ця система онлайн (сайт) має простий та зручний інтерфейс. Веб-програма проста, та немає ніяких особливих функцій. В загальному містить інформаційний характер про послуги на стоянці. На рисунку 1.12-1.13 зображено варіанти шляху до різних автостоянок. В середині системи використовуються Google maps також є GPS і звісно ж адреса і номер телефону щоб можна було зв'язатися з адміністрацією.

На рисунку 1.14 зображено вкладку з бронюванням місця. Потрібно як завжди заповнити форму з даними водія, дані про авто.

### 8.Автостоянка 2.5.5 [13]

На рисунку 1.15 зображено систему “Автостоянка”

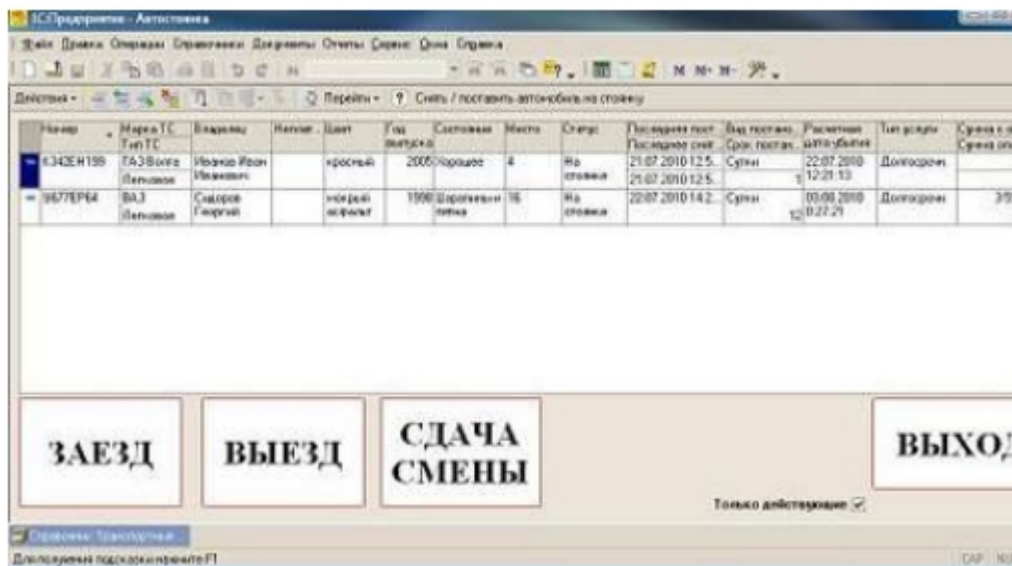


Рисунок 1.15 - Загальний вигляд системи

Це програма для ПК. В ній користувач може:

- Вести облік онлайн в базі
- Отримати звітність та інформацію про платежі
- Власноруч вибрати схему автостоянки де є вільне місце
- Показати на схемі де саме його транспортний засіб
- Оформляти операцію заїзду і виїзду авто зі стоянки
- Приймати кошти онлайн

- Розраховувати вартість послуг наданою системою
- Має власний облік знижок
- Також може нараховувати пеню за прострочення
- Вести контроль над днями і годинами прострочення

#### 9. Аксіома груп “Розумна стоянка” [4]

На рисунку 1.16 зображено систему “Розумна стоянка”



Рисунок 1.16 - Як виглядає табло

Основна задача проекту це інтеграція цілого апаратного комплексу, що дозволить ріал-тайм слідкувати стан кожного вільного місця для стоянки. Рішення використовує технології компанії Nedap AVI яка розробила для цього SENSIT. Фішка в чому, в кожне місце стоянки підключається датчик який за допомогою інфрачервоного світла і електромагнітного сенсору визначає чи зайняте місце чи ні. Далі датчики передають інформацію до командного центру, тобто серверу. Розроблене програмне забезпечення цією компанією дозволяє сортувати та обробити отриману інформацію і потім вже будувати різного типу звітності, і також виводити отриману інформацію на цифрове

					КВРКІ.170283.17.02.14 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

табло. Все обладнання сучасне, яке проходило тестування у вуличних умовах і адаптоване під різні кліматичні умови. Усі побажання міської служби були виконані. На сьогоднішній день ця система введена в експлуатацію і користувачі вже можуть отримувати повну інформацію на рахунок стоянки, рівень завантаженості та доступних місць.

Таблиця 1.1 – Порівняльна характеристика програмних продуктів:

Назва продукту	Доступний функціонал	Інтерфейс
Simple Parking	-Облік -Реєстрація -Перевірка -Зображення авто -Контроль прострочення	Desktop
Vector AP 2000	-Зображення авто -Реєстрація -Знижки -Онлайн оплата -Зображення вільних місць	Desktop
Yandex.parking	-Онлайн бронювання місця -Схема в'їзду і виїзду -Зображення авто -Онлайн оплата -Наявний розрахунок вартості послуг -Контроль прострочення -Контроль часу перебування авто на стоянці	Mobile

Продовження таблиці 1.1 – Порівняльна характеристика програмних продуктів

All parking	<ul style="list-style-type: none"> <li>-Облік</li> <li>-Доступна схема стоянки</li> <li>-Онлайн оплата</li> <li>-Знижки</li> <li>-Контроль прострочення</li> <li>-Реєстрація</li> <li>-Нарахування пені</li> </ul>	Desktop
Автостоянка в Борисполі	<ul style="list-style-type: none"> <li>-Схема в'їзду і виїзду</li> <li>-Координати на GPS</li> <li>-Онлайн бронювання місця</li> <li>-Доступна інформація про інші послуги</li> </ul>	Web
Автостоянка 2.5.5	<ul style="list-style-type: none"> <li>-Облік</li> <li>-Зображення авто</li> <li>-Контроль прострочення</li> </ul>	Desktop
PARKEON	<ul style="list-style-type: none"> <li>-Схема в'їзду і виїзду</li> <li>-Зображення авто</li> <li>-Облік</li> <li>-Контроль прострочення</li> <li>-Онлайн бронювання місця</li> <li>-Інформація про інші послуги</li> <li>-Наявний розрахунок вартості послуг</li> <li>-Знижки</li> <li>-Зображення вільних місць</li> </ul>	Real system

Кінець таблиці 1.1 – Порівняльна характеристика програмних продуктів

AS101 Pro Park	<ul style="list-style-type: none"> <li>-Схема в'їзду і виїзду</li> <li>-Зображення авто</li> <li>-Онлайн бронювання місця</li> <li>-Доступна інформація про інші послуги</li> <li>-Наявний розрахунок вартості послуг</li> <li>-Знижки</li> <li>-Контроль часу перебування авто на стоянці</li> <li>-Зображення вільних місць</li> <li>-Реєстрація</li> <li>-Зображення вільних місць</li> </ul>	Real system
----------------	--	-------------

Розглянувши аналоги системи можна виділити серед них основні переваги та недоліки. Багато з програмного забезпечення має зручний інтерфейс, більшість додатків онлайн а справжні системи вже введені в експлуатацію.

Одна частина розглянутих мною аналогів є системи які реалізовані на базі мікроконтролерів з використанням інших сучасних технологій, таких як комп'ютерний зір чи інфрачервоні системи аналізу тощо. Але не зважаючи те що вони доволі дешеві в порівнянні з ПЛІС та доволі популярні серед розробників, мікроконтролери все ж таки мають свої недоліки. Один із них це безпека. Безпека це одна із головних проблем мікроконтролерів, оскільки по суті це лише запрограмований фізичний пристрій, в порівняні з ПЛІС де є не лише код а й своя всередині запрограмована архітектура. А отже у ПЛІС вищий рівень безпеки в порівняні з мікроконтролерами. А інші розглянуті рішення були лише інформаційними додатками для різного типу користувачів.

І окрім якоїсь інформаційної цінності не несли в собі, або навіть взагалі не несли в собі якусь користь. Що зі своєї сторони взагалі не вирішало проблеми виникнення заторів на дорогах.

Коли буду розробляти власну систему обов'язково врахую усі зроблені мною дослідження і врахую всі переваги та недоліки досліджених мною систем та інших факторів впливу.

### 1.3 Використання ПЛІС в задачах автоматизації. Переваги та недоліки

Програмована логічна інтегральна схема - або ще ПЛІС [5] , це електронний компонент який зазвичай використовують для створення цифрових інтегральних схем. Головна особливість ПЛІС полягає в тому що її логіку не визначають при виготовлені, а встановлюють її за допомогою програмування ПЛІС. Для цього існують спеціальні програматори або спеціальна середовища для налагоджування, які дозволяють задати бажану архітектуру для пристрою, у вигляді звичайної принципової схеми або також можливо задати архітектуру за допомогою спеціальних мов (VHDL, Verilog тощо).

Альтернатива ПЛІС:

- Програмний контролер
- Базові матричні кристали(вимагають втручання ще під час виробництва)
- ASIC (дорогі в одиничних екземплярах)
- Спеціалізовані процесори
- Мікроконтролери (ПЛІС працює швидше в порівнянні з мікроконтролерами)

Іноді буває що процесори ще на виробництві модифікують, а вже потім вставляють в ПЛІС. Це допомагає спростити проектування ПЛІС. Сьогодні

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	Арк. 26
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЛІС є найбільш поширеними і найбільш використовуваними елементами бази для пристроїв будь-якого призначення. У ПЛІС є переваги і ось які [6]:

- Широкомасштабне виробництво
- Доступна ціна
- Надійність
- Швидкодія
- Універсальність
- Велика різноманітність параметрів для сигналів(вхідних та вихідних)
- Легко та швидко проектується
- Підтримка різних зручних та високоефективних засобів автоматизованого проектування.
- Можливість легко та зручно модифікувати проекти

Також, ПЛІС як і інші апаратні засоби мають свою класифікацію.

Наприклад:

- Тип пам'яті
- Кількість етапів програмування та проектування
- Архітектурні
- Складність інтеграції
- Затримка сигналів
- Системні характеристики
- Як виготовляють

Ще ПЛІС мають такі різновиди (Рисунок 1.24)

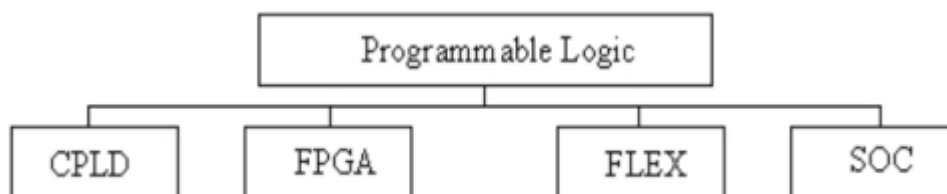


Рисунок 1.24 - Загальна класифікація ПЛІС



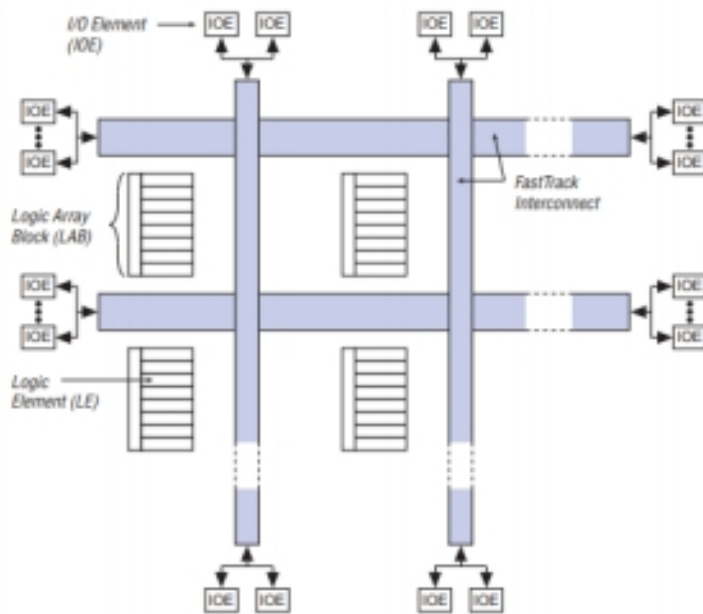


Рисунок 1.25 - Архітектура FLEX

На основі FLEX архітектури вже будують комбіновані архітектури. Де поєднують різні реалізації алгоритмів та моделі пам'яті. І від них вже будуються системи на кристалі (SOC).

#### CPLD (Complex programmable logic device)

CPLD - це комплексний програмований пристрій (Рисунок 1.27), а отже це комбінація програмованої множини АБО/І елементів та комірок. Елементи АБО/І можуть бути перепрограмовані безліч раз та виконувати різноманітні функції. Комірки - це функціональні блоки, які виконують, або послідовну логіку, або комбінаторну, це все залежить від поставленої задачі. Вони також мають гнучкість для визначення істинності або доповнення, і різні шляхи зворотнього зв'язку. CPLD архітектура для підвищення своєї продуктивності використовує аналогові підсилювачі. Це підвищення продуктивності було реалізовано через високі вимоги. Це дозволяє розробникам використовувати повторно одну й ту саму архітектуру CPLD при проектуванні високоякісних, малопотужних та високопродуктивних пристроїв.

На сьогоднішній день CPLD використовують структуру логіки без живлення та є легко перепрограмованими. Це зручна архітектура для розробки різноманітних цифрових пристроїв. Тому що можна легко реалізувати функції які задані через диз'юнктивну форму. Це дуже важливо, оскільки це має велике значення при оптимізації схем, які складені з простих логічних елементів.

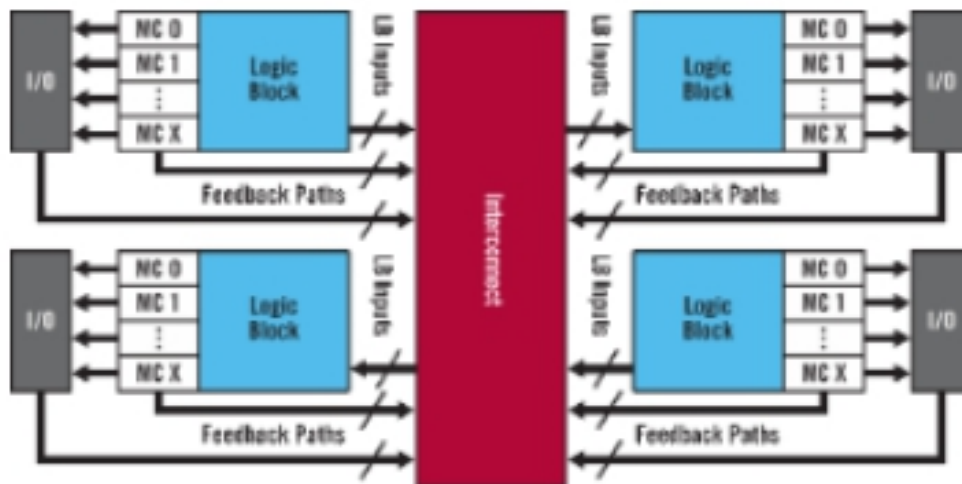


Рисунок 1.26 - Архітектура CPLD

- FPGA (Field programmable gate array)

Сама архітектура FPGA з'явилася дуже давно. Але з того часу набирає великих обертів. Прогрес цієї технології значно розширив можливості, і зробив аналогом для складних реалізацій. Можливість замінити програмне забезпечення модулю значно впливає на результат розробленого пристрою. А особливо на потужність та швидкодію пристрою.

FPGA будується з матриці логічних блоків, які з'єднані між собою за допомогою мережних з'єднань. Маршрутизація, та програмована логіка під час використання FPGA робить її дуже зручною та гнучкою у використанні. Але є і мінуси. Це сповільнює роботу пристрою. Також збільшується споживання електроенергії. Проте її продовжують вдосконалювати.

FPGA є вигідним та дешевим рішенням для багатьох малих та середніх задач. Це дозволяє пристрою швидко вийти на ринок в порівнянні з різними інтегральними схемами, які в свою чергу потребують більше грошей та часу. Ще є плюсів є те, що якщо потрібно замінити частину логіки відповідно до вирішення поставленої задачі, то частина логіки ПЛІС може бути частково або повністю переписана, а решта буде працювати так само. Також зручно ставити нові оновлення, варто лише завантажити новий бітовий потік програми. Але сильна сторона, тобто гнучкість та можливість швидко щось змінити, є причиною у програші перед ASIC. Через гнучкість нової системи, вона збільшується у розмірах. Втрачають швидкість і використовують більше енергії в порівнянні з аналогом від ASIC. Зазвичай ці недоліки через зв'язок маршрутизації. А це майже 90% від усієї площі FPGA. Але не зважаючи на це FPGA це альтернатива цифрової системи, оскільки це значно дешевше і менший час щоб вийти на ринок з новим розробленим пристроєм.

FPGA складається з таких елементів:

- Логічні блоки (блоки, що реалізують логічні функції)
- Блоки вводу-виводу (вони підключені до логічних блоків через маршрутизацію і з'єднують мікросхеми)
- Програмована маршрутизація (з'єднує логічні функції)

Загальний приклад FPGA (Рисунок 1.26). Configurable Logic Block (CLB) зазвичай розміщуються у вигляді двовимірної сітки і вони мають бути пов'язані з ресурсами маршрутизації. Блоки вводу та виводу інформації розташовані на сітці периферії. Вони також мають бути підключені до програмного з'єднання маршрутизації. Програмування FPGA починається з апаратного опису, потім синтезу, потім об'єднання та складання. Після цього цей опис має бути розміщений на кристалі FPGA для коректної роботи.

					КВРКІ.170283.17.02.14 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

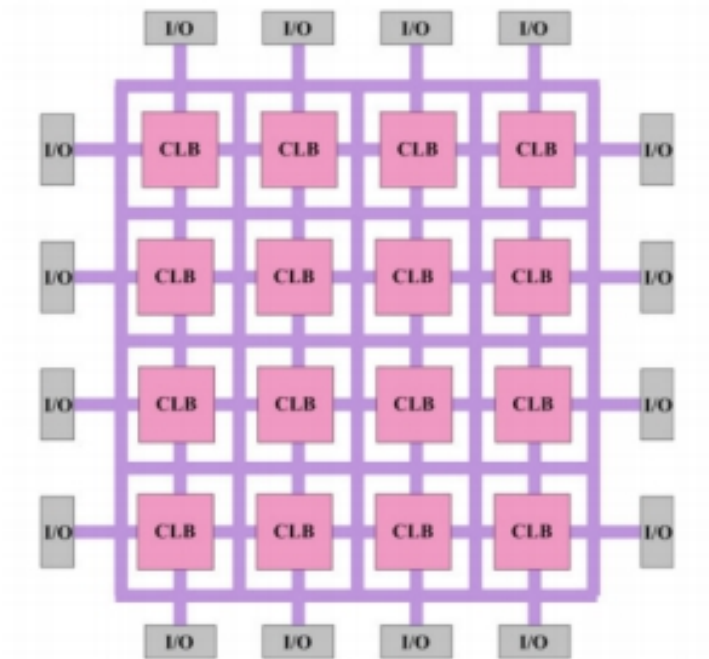


Рисунок 1.26 - Архітектура FPGA

Сьогодні FPGA (фірми ALTERA) пропонує великий вибір вбудованих блоків пам'яті (SRAM - Static random Access memory), високошвидкісні трансфери, логічні блоки, блоки маршрутизації та швидкісні блоки входу-виводу інформації. Вбудована логіка у поєднанні з програмним забезпеченням значно зменшує час розробки спеціалізованих цифрових пристроїв.

ALTERA пропонує такі FPGA ПЛІС [9]:

1. FPGA Cyclone (саме це сімейство я буду використовувати у реалізації дипломної роботи, а саме автоматизовану систему автостоянки) була побудована щоб вирішувати різного роду проблеми.

Вона задовольняє потреби низького споживання електроенергії. Має високу гнучкість та ефективність. Ці характеристики дозволяють вийти на ринок в короткі терміни.

Кожне нове покоління вирішує такі проблеми:

- Гнучкість інтеграції
- Підвищена продуктивність

- Швидкий вихід на ринок
- Менша потужність споживання

Приклади FPGA ALTERA (Рисунок 1.27 - 1.28): Cyclone 5( наявна на кафедрі, тому я буду працювати саме з цією ПЛІС), Cyclone 4, Cyclone 3, Cyclone 10.

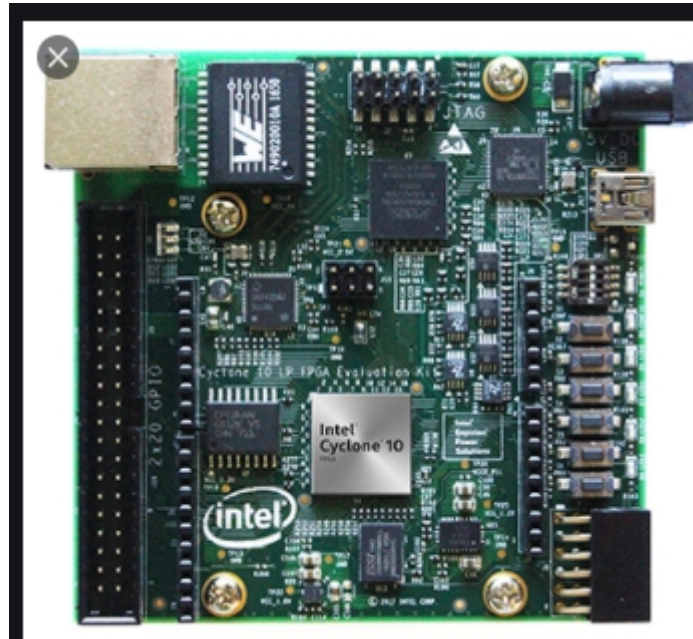


Рисунок 1.27 - Cyclone 10



Рисунок 1.28 - Cyclone 5







- Дослідити та вивчити теоретичний матеріал
- Дослідити шляхи вирішення поставленої задачі
- Дослідити наявні аналоги
- На основі проведених досліджень визначити основні функції системи.
- Створити продукт який би міг включати усі необхідні функції і автоматизувати процес пошуку автостоянки
- Під час реалізація використовувати ПЛІС та мову програмування VHDL

Оскільки усі розглянуті аналогічні системи насправді являють собою лише інформаційний додаток для усіх користувачів будь-то адміністрація стоянки чи звичайні водії. Тому в даному випадку усі рішення залишаються насамперед за користувачем системи автоматизованої стоянки. Тобто усе це взагалі не виключає людський фактор. А отже головна проблема, така як, затори на дорогах та час який витрачений на пошуки місця для паркування, можуть так і залишитися не вирішеними.

А інші системи які були розроблені з автоматичним керуванням реалізовані на базі мікроконтролерів. З економічної точки зору це вигідно, оскільки мікроконтролери дешеві, та досить популярні, що також має свої плюси, оскільки їх завжди можна знайти та замовити. Але вони мають один великий мінус перед ПЛІС. Мікроконтролери програють ПЛІС з точки зору безпеки. Їх можна легко зламати, бо по факту це лише запрограмований пристрій. І рівень хакера має бути доволі низький, щоб здійснити атаку. А от ПЛІС заламати вже набагато важче, оскільки це не запрограмований пристрій, а її робота визначається насамперед не просто набором команд а прописаної в середині неї архітектурою, а отже рівень хакера має бути вже набагато вищим щоб здійснити атаку пристрій розроблений на базі ПЛІС. Отже, використовувати ПЛІС під час розробки цього проекту буде більш доцільніше і безпечніше, тому я обрав ПЛІС. Під час розробки своєї власної системи потрібно врахувати усі дослідження, тобто переваги та недоліки розглянутих систем.

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	<i>Арк.</i> 37
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## 2 ПРОЕКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

### 2.1 Методологічні підходи до вирішення задачі за темою дослідження

Нас сьогоднішній день існує доволі багато різних підходів як технічних так і архітектурних для вирішення поставленої задачі. Серед них є речі які мають щось спільне, але також і те що їх відрізняє. Взагалі існують класифікації систем для сучасних стоянок. Наприклад [8]:

- Agent model
- Vehicle to Infrastructure
- GPS system
- System based on Computer CV
- System based on RFID
- System based on wireless sensors network
- Системи на основі нечіткої логіки

Якщо всі ці системи розглянути більш детально і знайти сильні сторони та всі плюси які система може дати користувачам то отримаємо такі результати:

#### 1. Agent model

Цей тип моделі (Рисунок 2.1) був представлений академікам і їхній спільноті загалом.

Ця система має два шляхи розвитку. Вона може бути як той чи інший суб'єкт, який може помічати усі зміни через датчики чи сенсори, або ж може існувати як незалежна система, яка реагує на зміни навколишнього середовища шляхом обміну інформацією та взаємодією. Її сильні сторони це майже повна автономність, швидкість реагування на зміни та адаптація до нових умов.

По іншому, моделювання з багатьма агентами це -- метод, який був розроблений для опису систем, які мають сутності, взаємодію між ними та автономністю.

Також, в основі методу багатофазної навігації що базується на основі дворівневої карти графіку, лежить модель з багатьма агентами. Підхід розподілу полягає в побудові розумної системи з рекомендаціями. Її ще називають APGIS.

По суті, APGIS складається з менших дочірніх елементів (сервісні центри, стоянки, самих машин, автопарків). І він має чотири основні функції:

- Пошук місця для паркування
- Розрахунок вартості паркування
- Бронювання місць
- Створення маршрутів паркування та їх узгодження

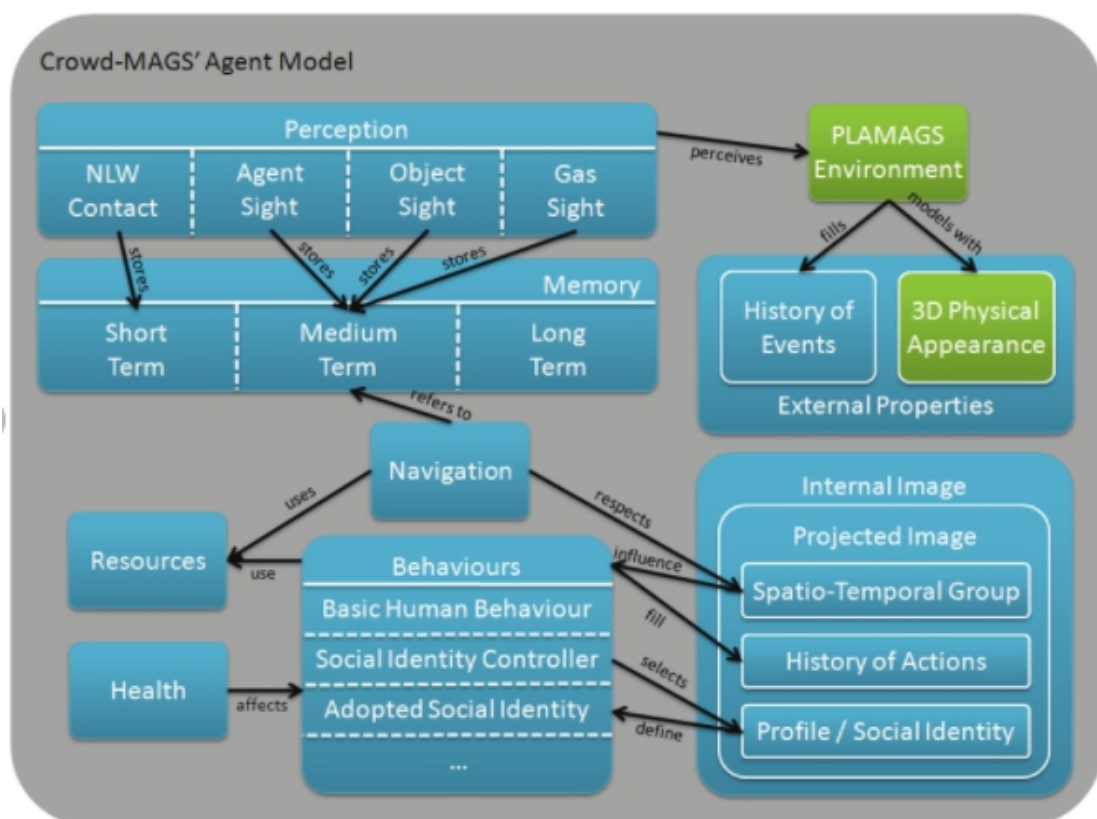


Рисунок 2.1 - Agent model





ситуації, обладнання може аналізувати кілька кадрів в секунду, а потім може посилати дані на сервер, звідки їх отримає водій і буде проінформований про зміну на стоянці. Зазвичай “оком” машини є відеокамери. Їх встановлюють таким чином, щоб було видно усю автостоянку. Далі система по пікселям зчитує отримане зображення і так може розпізнати вільне місце чи ні. Правда, іноді вони не правильно виявляють припарковані автомобілі. Ця система, зазвичай, націлена на ті випадки, коли потрібно дізнатися на скільки заповнена стоянка.



Рисунок 2.4 - Computer CV detect system

## 5. System based on RFID

Цю систему (Рисунок 2.5) поки що використовують на етапі прототипу. За допомогою цієї системи також легко проводити моніторинг стоянки. Основна ідея полягає у використанні RFID технології яка залежить від електромагнітного поля для того щоб визначення та переглядання міток, які були додані до об’єктів раніше.

					КВРКІ.170283.17.02.14 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

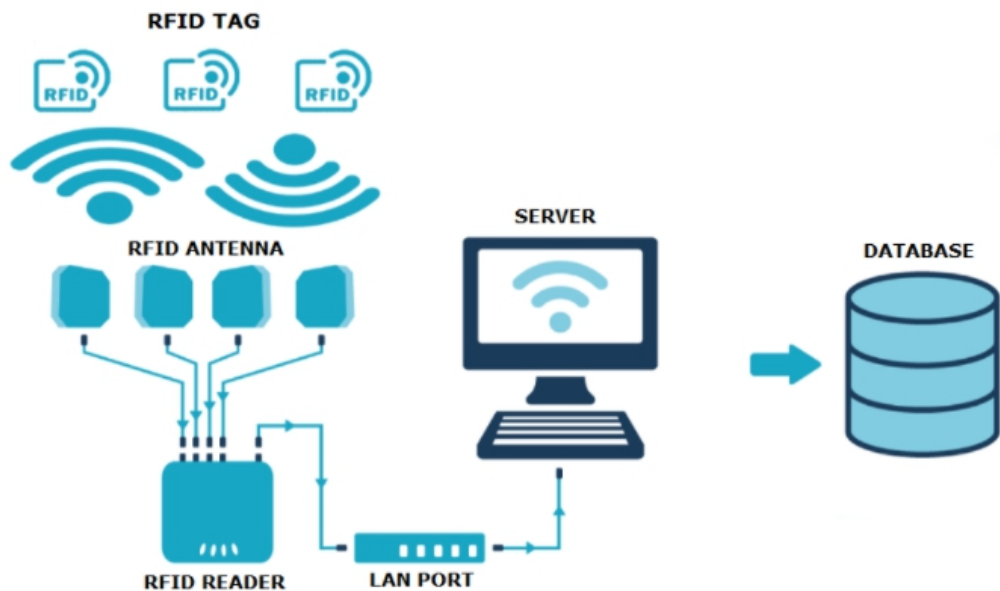


Рисунок 2.5 - RFID structure system

#### 6. System based on wireless sensors network

Це одна з найпопулярніших систем (Рисунок 2.6) за останні 10 років. Тому що ці системи мають багато різних переваг:

- Гнучкість
- Штучний інтелект
- Дешеве обладнання
- Легко та швидко розгорнути систему
- Проглядання.

Зазвичай, це тип системи, яка використовує різноманітні датчики та сенсори для моніторингу умов довкілля, широко розповсюджені, особливо у спеціалізованих навчальних закладах, через те що дешеві та їх легко налаштувати.

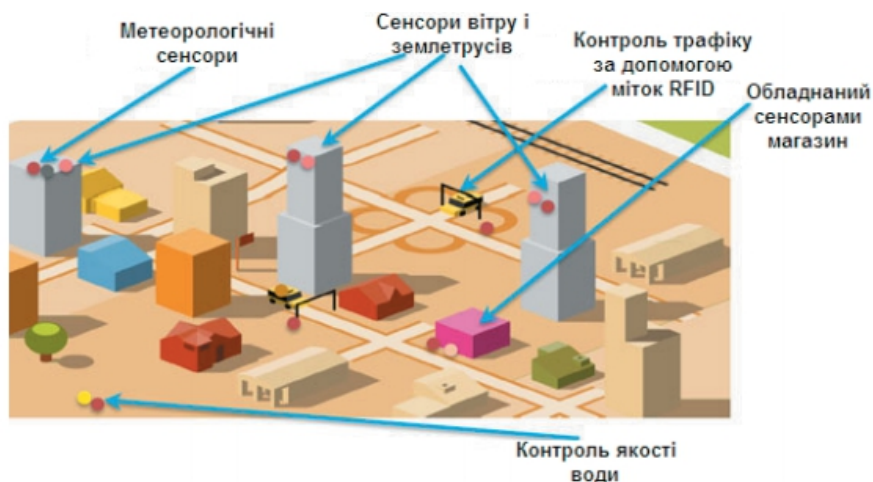


Рисунок 2.6 - System based on wireless sensors network

## 7. Системи на основі нечіткої логіки

Ще з 60-х років коли вперше почали використовувати нечітку логіку, вона зіграли свою роль у розробці та промисловості. Нечіткі системи управління (Рисунок 2.7), насправді є системами керування на основі нечітких елементів, які аналізують аналогові вхідні дані (які бувають від 0 до 1), тоді як цифрові елементи діють на значеннях 1 або 0. Сьогодні вважають що нечітка логіка це стандарт, який використовують при аналізі сигналів або даних. Нечіткий код який був розроблений щоб контролювати щось механічне. Отже, з часом була винайдена система, яка залежала від FLC (нечіткий логічний контролер). Перевага FLC з ПЛІС над FLC з програмним забезпеченням полягає в тому що буде витрачено менше часу на обробку вхідних даних. А все тому, що це система нечіткого керування і в ній використовується реалізація правил, які засновані на архітектурі нейронної мережі. Це дає великий прогрес цій системі, оскільки система може вчитись ріал-тайм та адаптуватися до нових умов. А передавач і антена будуть надавати інформацію про вільні місця на стоянці передавати інформацію користувачам.

						КВРКІ.170283.17.02.14 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата			

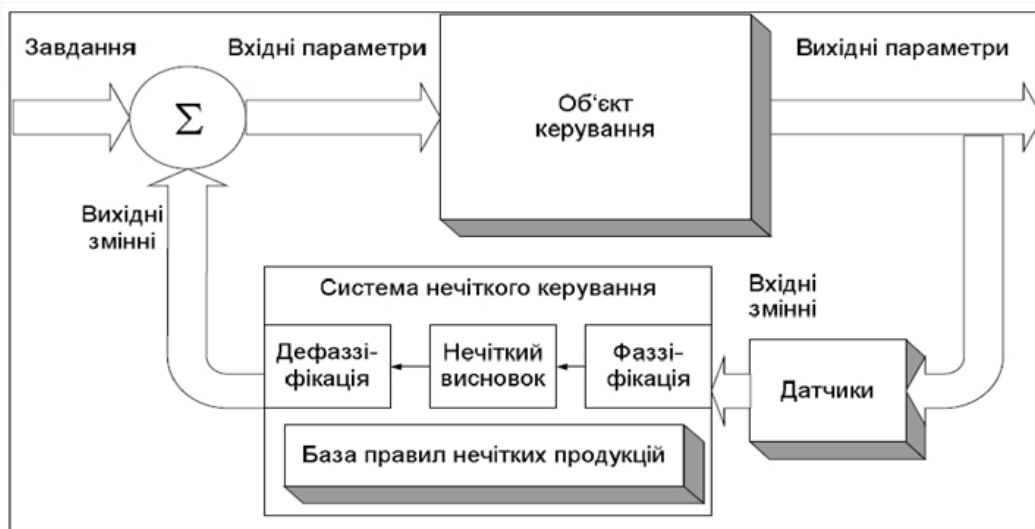


Рисунок 2.7 - Загальна схема нечіткого керування

## 2.2 Вибір апаратної бази

### ATMEGA Altera Cyclone 5

Я вирішив обрати Cyclone 5 DE1-SoC (Рисунок 2.8) для розробки автоматизованої системи паркування, оскільки це надійна апаратна платформа для проектування та реалізації різних проектів, до того ж вона побудована навколо FPGA Altera Soc, яка об'єднує в собі двох-ядерні вбудовані ядра Cortex-A9, з програмованою логікою (підтримка VHDL), що робить її дуже зручною у використанні під час проектування архітектури та легко перепрограмованою у випадку, якщо потрібно внести якісь зміни у розробку проекту. І це все в поєднанні з високопродуктивною та малопотужною системою (Рисунок 2.9). Ця плата інтегрує систему жорстких дисків на основі ARM, яка включає в себе процесор, різноманітні периферійні пристрої та інтерфейси пам'яті, яка пов'язані з FPGA використовуючи при цьому шину даних з високою пропускнуною здатністю.

Плата DE1-SoC включає в себе:

- Високошвидкісна пам'ять DDR3

- Можливість використовувати аудіо та відео-зв'язку
- Можливість працювати з Ethernet

Плата DE1-SoC [10] містить в собі усі потрібні компоненти, необхідні для того щоб можна було використовувати плату разом з комп'ютером з Windows (починаючи з версії XP і пізніше). Також для компіляції проектів для плати DE1-SoC обов'язково потрібна 64-розрядна ОС та 64-розрядна версія Quartus 2.

Плата включає в себе такі компоненти (Рисунок 2.10):

- One Cyclone 5 Soc in 896-pin FBGA
- FPGA з конфігураціями безпеки
- Таймер(SMA input тощо)
- Різні види пам'яті (DDR3, QSPI, CFI, micro SD)
- Комунікаційні порти (PCI Express, HSMC port, USB 2.0, Gigabit Ethernet port, SDI port, CAN, real-time clock)
- Input/Output (LED displays, push buttons, DIP switches, power supply, Mechanical)

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	<i>Арк.</i> 46
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

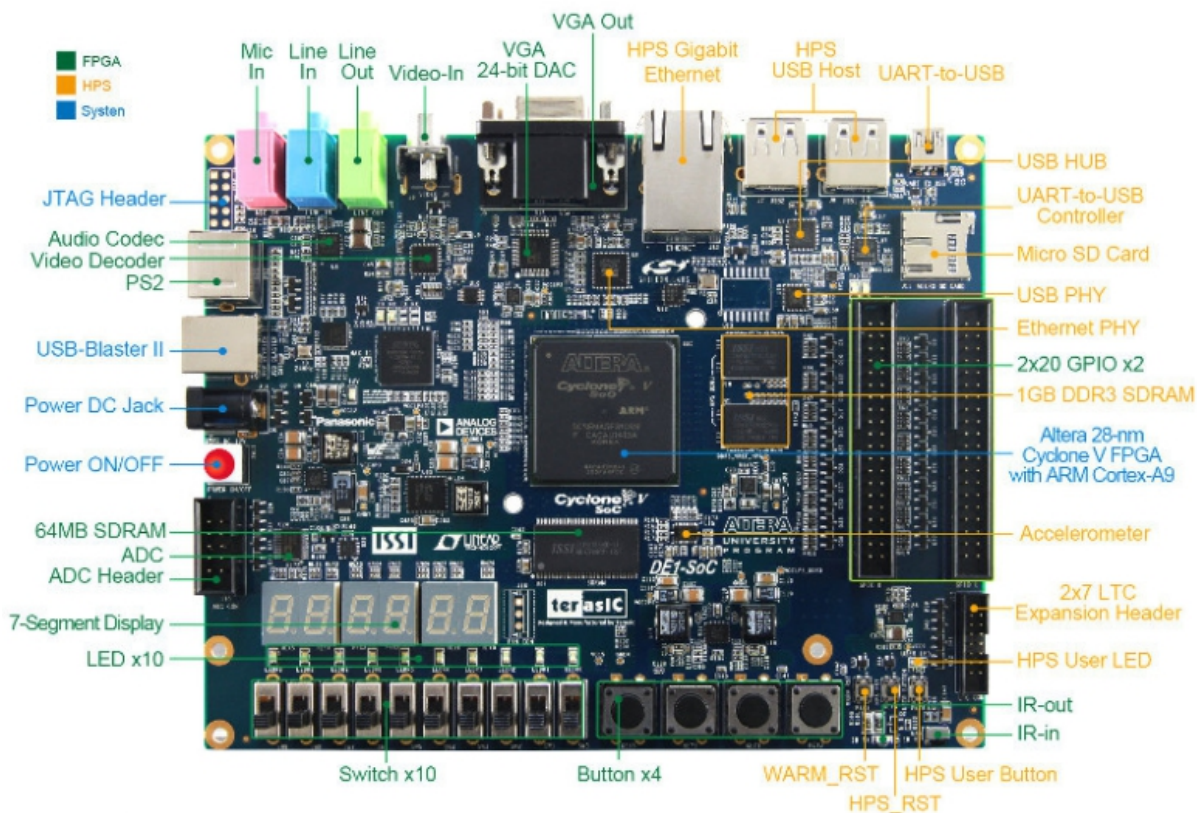


Рисунок 2.8 - Фізична плата

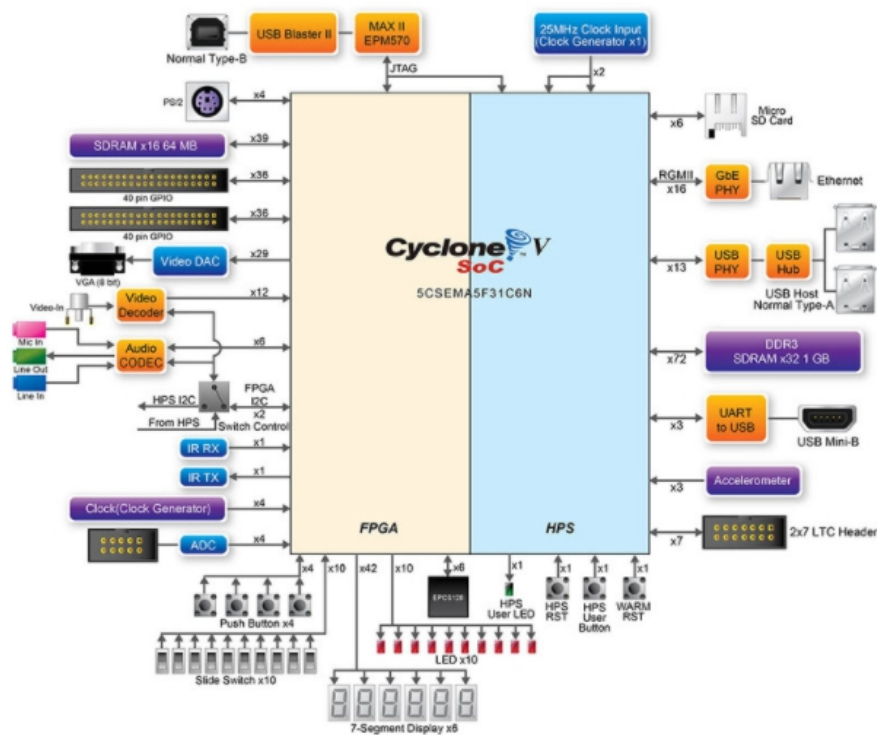


Рисунок 2.9 - Структурна схема плати

Зм.	Арк.	№ докум.	Підпис Дата

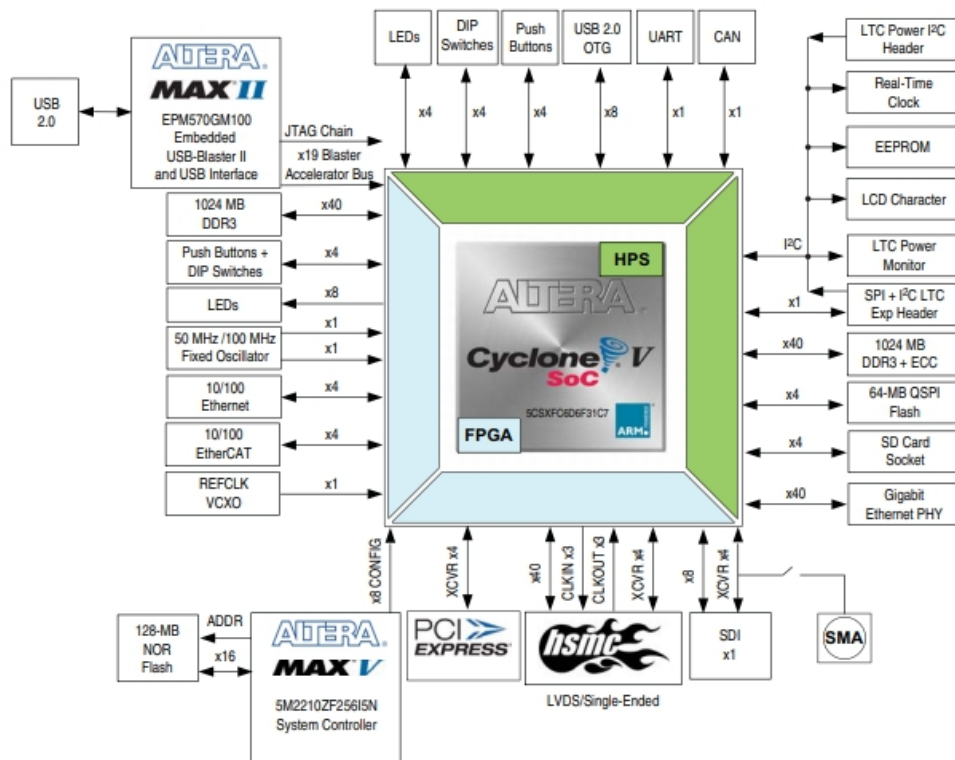


Рисунок 2.10 - Блок-діаграма компонентів та їх розміщення на платі VHDL

VHDL (Рисунок 2.11 - Приклади коду) - це мова для опису апаратури інтегральних схем. Мова проектування VHDL на сьогоднішній день є базовою мовою для розробки повного циклу апаратури сучасних систем для обрахунків.

Написання моделі на VHDL дозволяє реалізувати автоматичний синтез схеми яка функційно рівна початковій моделі. За допомогою VHDL можна робити проектування на різних рівнях абстракції. Також існує можливість ієрархічного проектування, що є максимально зручних та гнучким під час роботи у великих проектах у великому колі розробників. У VHDL є три основних його частини опису:

- Проблемно-орієнтовна (яка і робить VHDL мовою опису апаратури)
- Об'єктно - орієнтоване

- Алгоритмічну (комбінація мова програмування Pascal та Ada)  
Для VHDL існує три загальноприйняті стандарти:
- ГОСТ 50754-95 (мова опису апаратури цифрових систем VHDL)
- IEEE std 1076-2008 (Standard language reference manual)
- IEEE std 1076-2002 (Standard language reference manual) [20]

На мові VHDL були створені описи відкритих мікропроцесорів ERC32 (SPARC V7) і LEON (SPARC V8).

Також на основі стандарту IEEE std1076-2008 [19] розроблена Open Source VHDL Verification Methodology. Вона дозволяє реалізувати функційне покриття та генерацію тестів, які використовують для верифікації функційних блоків. В рамках OS-VVM написано декілька пакетів VHDL які генерують псевдо-випадкові тести. Отже виходячи зі всіх цих описаних сторін плати, саме на мові VHDL я буду описувати проект для розумної системи автостоянки на платі Cyclone 5 De1-Soc.

```

-- (this is a VHDL comment)
/*
   this is a block comment (VHDL-2008)
*/
-- import std_logic from the IEEE library
library IEEE;
use IEEE.std_logic_1164.all;

-- this is the entity
entity ANDGATE is
  port (
    I1 : in std_logic;
    I2 : in std_logic;
    O  : out std_logic);
end entity ANDGATE;

-- this is the architecture
architecture RTL of ANDGATE is
begin
  O <= I1 and I2;
end architecture RTL;

```

Рисунок 2.11 - Лістинг VHDL (до прикладу реалізація AND gate)

IDE Quartus 2

					КВРКІ.170283.17.02.14 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		





1. Як створити проект (Рисунок 2.13)
2. Додаткові налаштування (Рисунок 2.14)
3. Додати потрібні файли для конфігурації (Рисунок 2.15)
4. Обираємо плату (Рисунок 2.16)
5. Останній крок конфігурації (Рисунок 2.17)
6. Подивитись конфігурацію (Рисунок 2.18)
7. Робота з файлами (Рисунок 2.19)
8. Компіляція проекту (Рисунок 2.20)

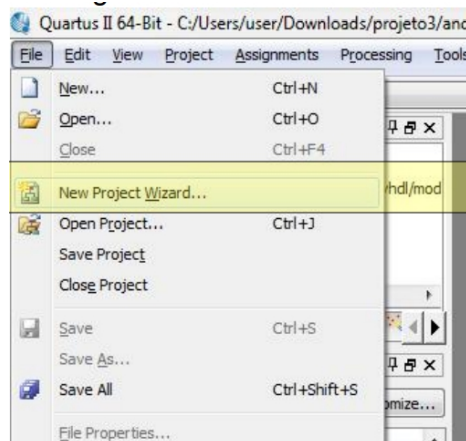


Рисунок 2.13 - Як створити проект

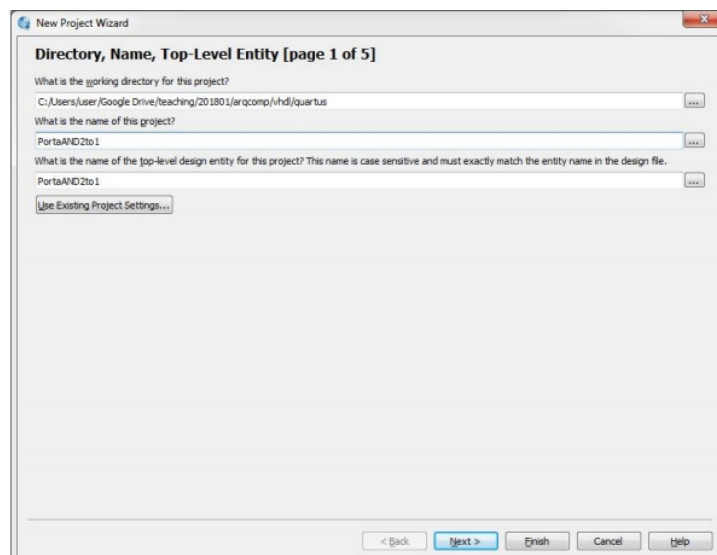


Рисунок 2.14 - Додаткові налаштування

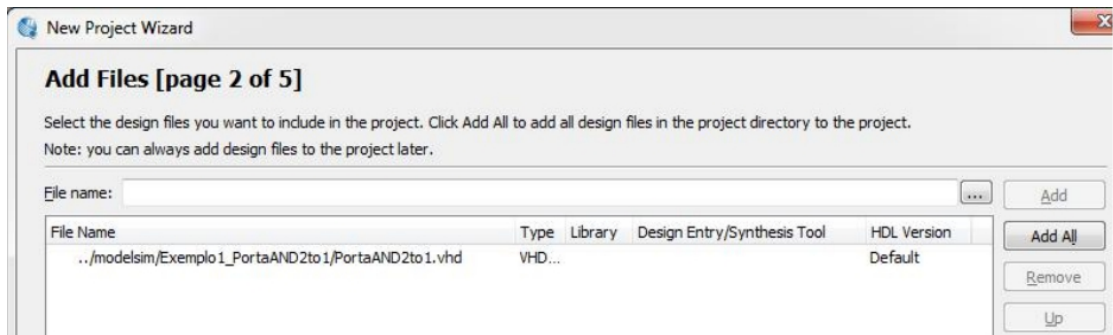


Рисунок 2.15 - Додати додаткові файли

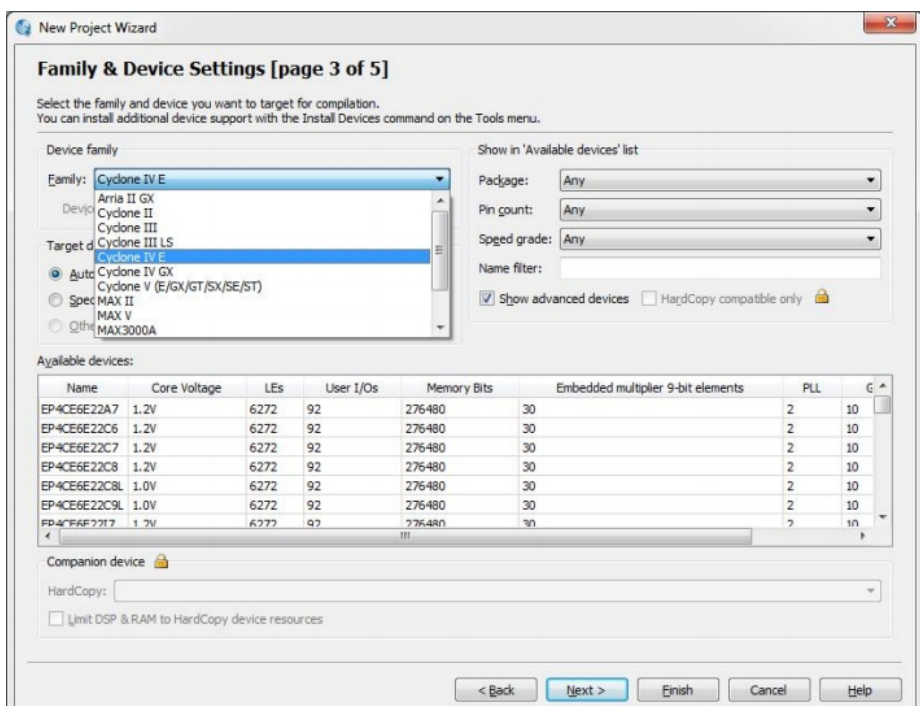


Рисунок 2.16 - Вікно вибору плати

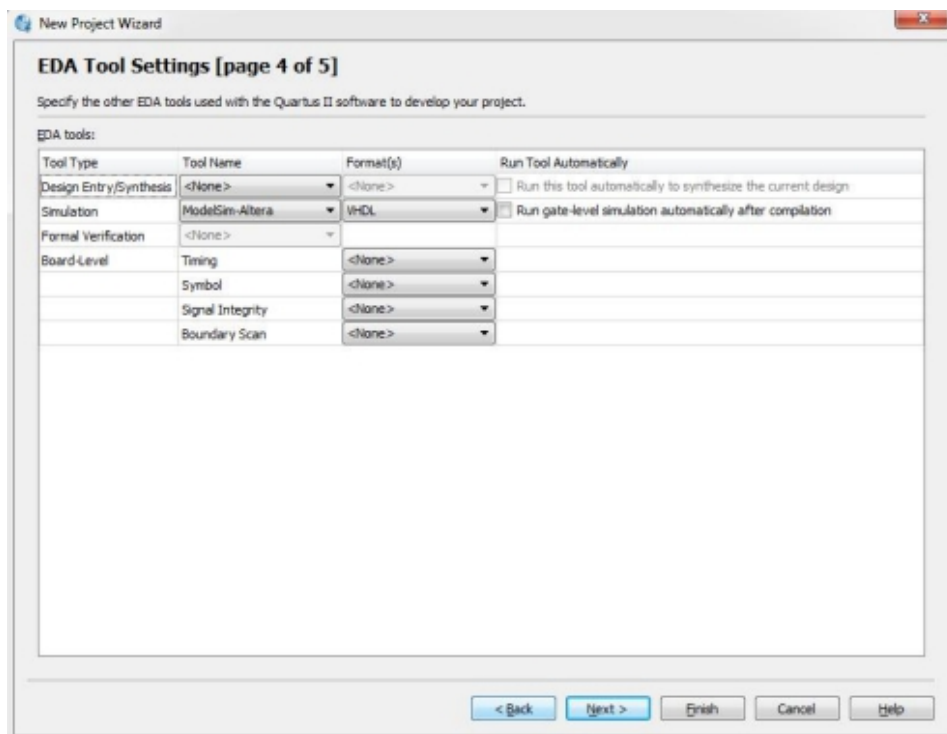


Рисунок 2.17 - Вікно з останнім кроком конфігурації

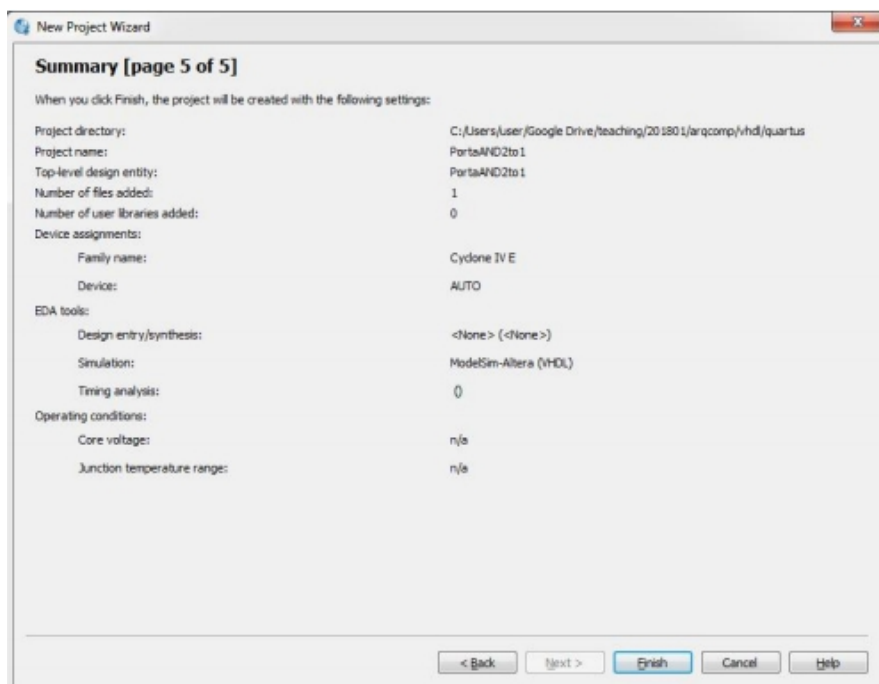


Рисунок 2.18 - Вікно з фінальною конфігурацією проекту

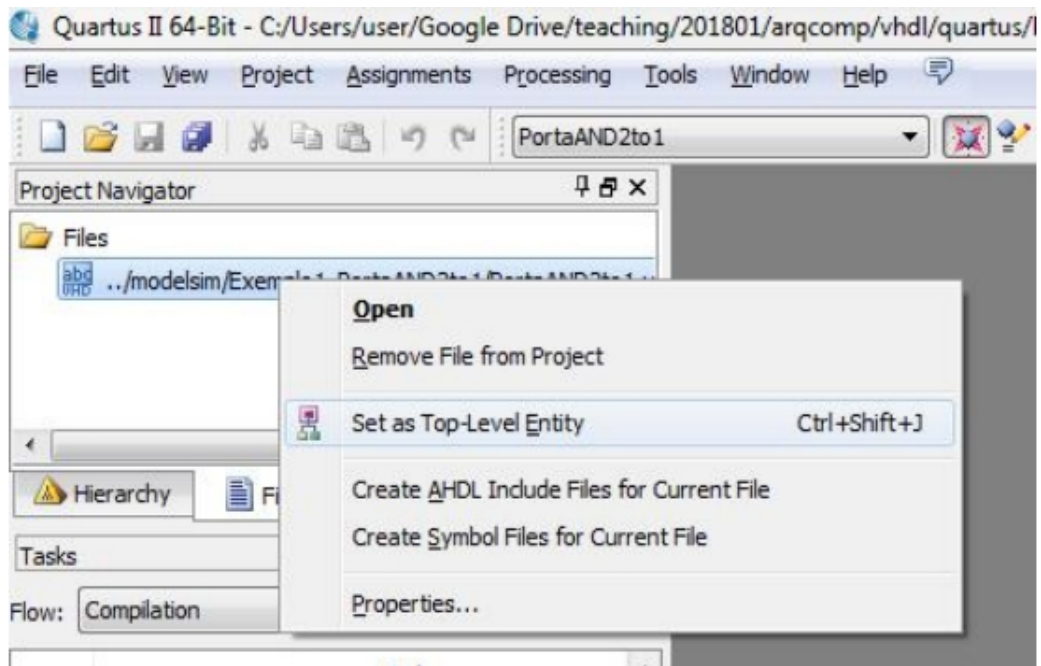


Рисунок 2.19 - Робота з файлами



Рисунок 2.20 - Як почати компіляцію проекту

### 2.3 Висновки до другого розділу

					КВРКІ.170283.17.02.14 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

В другому розділі було проаналізовано які існують наявні способи вирішення поставленої задачі. І було сформовано основні можливості та загальна архітектура розроблювальної системи автоматизованої стоянки. Оскільки на сьогоднішній день ПЛІС є однією з найбільш ефективних та широко поширених елементних баз для побудови специфічних або спеціалізованих цифрових пристроїв та систем різного призначення, я вирішив обрати саме ПЛІС а не мікроконтролери для реалізації автоматизованої системи для паркування. Також ПЛІС має такі переваги:

- Універсальність
- Високий попит на ринку
- Вищий рівень безпеки ніж в мікроконтролерах, що на сьогоднішній день не менш важливо.
- Доволі низька вартість.
- Надійність
- Швидкодійність

Оскільки ПЛІС можна запрограмувати на мові опису архітектури, до прикладу VHDL, то це робить її зручною на різних етапах доступною для модифікацій.

Якщо обирати між CPLD або FPGA, то я зроблю вибір в сторону FPGA оскільки вона краще підходить для виконання поставленої задачі, тобто реалізації спеціалізованого пристрою. Також вона відповідає великому спектру вимог і добре підходить для створення автоматизованої системи стоянки. До якого потім можна буде розробити не лише систему на ПЛІС а й графічний додаток для різних платформ.

					КВРКІ.170283.17.02.14 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

#### 3.1 Алгоритм роботи

Провівши відповідне дослідження та проаналізувавши отриманні результати а також оцінивши свої можливості я дійшов висновку що буду розробляти систему сучасного паркування виконуючи чотири основні правила:

- Система має бути максимально простою
- Використовувати під час проектування ПЛІС (з підходом опису схеми на VHDL), оскільки це значно допоможе автоматизації
- Розширення логіки та алгоритму роботи має бути максимально зручним
- Система має бути дешевою

Я знайшов оптимальне вирішення проблеми. В'їзд до автостоянки буде лише один. Тобто машини зможуть заїжджати і виїжджати з автостоянки лише через одні ворота. Це зумовлено тим що це зекономить час, місце та людський ресурс під час встановлення автоматизованої системи у відведеному для цього місці. На в'їзді до автостоянки будуть стояти два датчика руху і в залежності від того як вони будуть спрацьовувати і буде виводитись загальна кількість машин на загальному дисплеї (на платі це семи-сегментні дисплеї, де кожна цифра описується в програмі, а отже це означає що її можна швидко, зручно та динамічно змінити або додати ). Якщо спрацьовує перший датчик (Рисунок 3.1) а потім другий, то лічильник збільшиться на 1 (програма на Рисунку 3.2, блок-схема алгоритму роботи на Рисунку 3.3 ). І так до 50. Якщо ж навпаки, спочатку другий потім перший то лічильник зменшується на 1 (Рисунок 3.4). Програма на Рисунку 3.5, блок-схема алгоритму роботи на Рисунку 3.6 Як тільки на одна із стоянок буде

					КВРКІ.170283.17.02.14 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

повністю заповнена, тобто досягне відмітки у 50, то потім інші машини вже будуть заїжджати вже на іншу стоянку. В своїй системі я зробив 3 стоянки по 50 місць, що в цілому дозволить вмістити до 150 автомобілів на одній стоянці. Я вирішив обмежити кількість доступних місць на одній автостоянці до 50, і всього зробити 150 тому що я вважаю це оптимальним рішенням, але так як ця система пишеться з використанням підходу на VHDL, цю цифру можна оптимально задати для кожної стоянки окремо. Роль датчиків на платі відіграло дві перемикачі за допомогою яких я імітував їхню роботу спрацьовування по черзі. А роль активування датчика, тобто авто, відіграла кнопка на самій платі. Загальний алгоритм роботи схеми на Рисунку 3.7



Рисунок 3.1 - Якщо спрацьовує перший датчик

```

--Якщо машини заїждють
if (clk='1' and clk'event) then

if (clk_in='1' and clk_out='0') then

    if(count_first < "110010") then
    count_first:= count_first + '1';

    elsif (count_second < "110010") then
    count_second:= count_second + '1';

    elsif (count_third < "110010") then
    count_third:= count_third + '1';

    else
    count_first:= "110010";
    count_second:= "110010";
    count_third:= "110010";
    end if;

```

Рисунок 3.2 - Програми якщо машини заїжджають

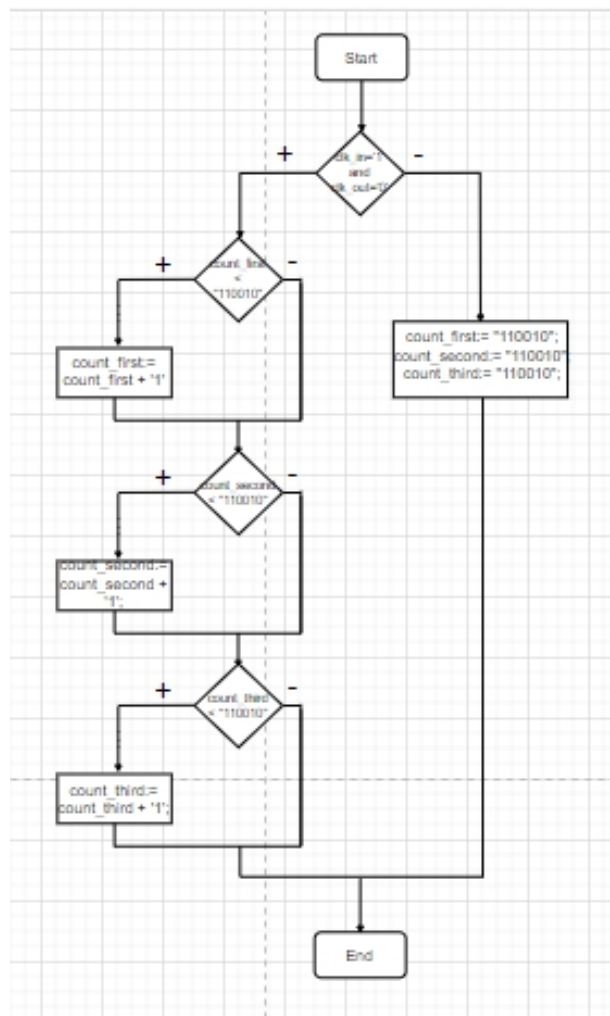


Рисунок 3.3 - Алгоритм роботи логічного блоку заїзду машин

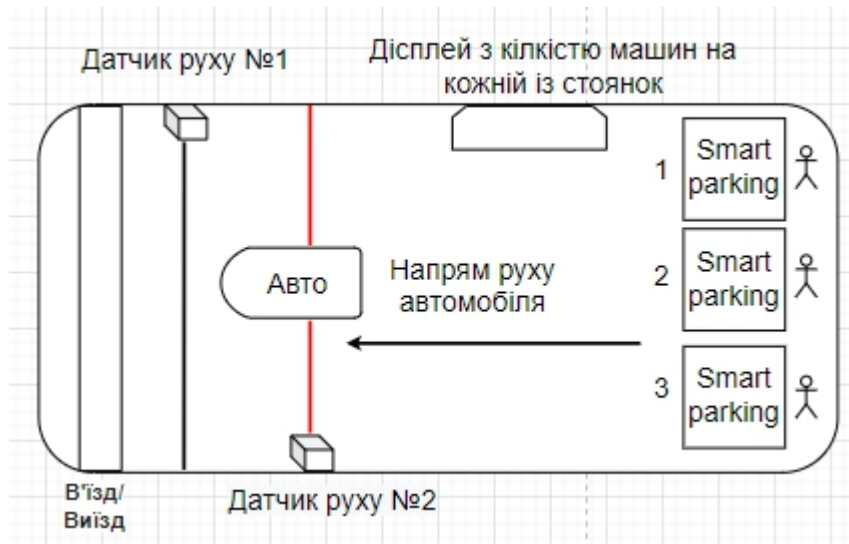


Рисунок 3.4 - Якщо спрацьовує другий датчик

```

--Якщо машини виїжають
elsif (clk_out='1' and clk_in='0') then
  if (count_third > "000000" and count_third<"110011") then
    count_third:= count_third - '1';

    elsif (count_second > "000000" and count_second<"110011") then
      count_second:= count_second - '1';

    elsif(count_first > "000000" and count_first<"110011") then
      count_first:= count_first - '1';

  end if;

elsif (clk_in='0' and clk_out='0') then NULL;
elsif (clk_in='1' and clk_out='1') then NULL;
end if;

```

Рисунок 3.5 - Програма якщо машини виїжджають



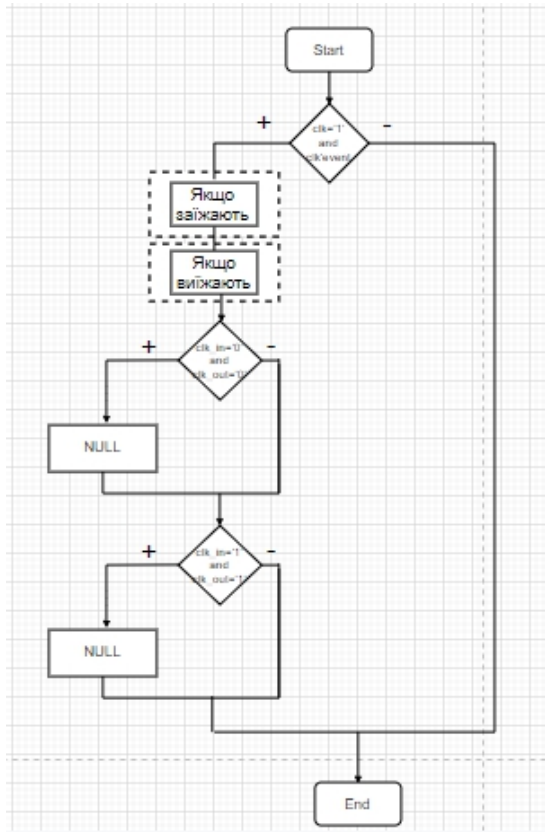


Рисунок 3.7 - Загальний алгоритм роботи системи

### 3.2 Отримані результати

Отож, зробивши програму враховуючи усі нюанси які мені були потрібні, я спочатку змодельював розроблену програму у середовищі Quartus 2 і провів тести. Після того я залив програму на плату і провів ще одні додаткові тестування щоб переконатися що система працює справно. На даний момент система автоматизованої стоянки працює справно. Можна переконатися як вона працює на Рисунку 3.8 - 3.11. Також загальна схема розташування може виглядати по різному.

Як ідея можна додати після того як машина попадає в середину системи, тобто стоянки, перед кожної з стоянок буде стояти шлагбаум (Рисунок 3.12 - 3.13). І біля нього буде умовний світлодіод або інший пристрій який буде сповіщати про заповненість однієї з трьох стоянок. Якщо на стоянці ще менше







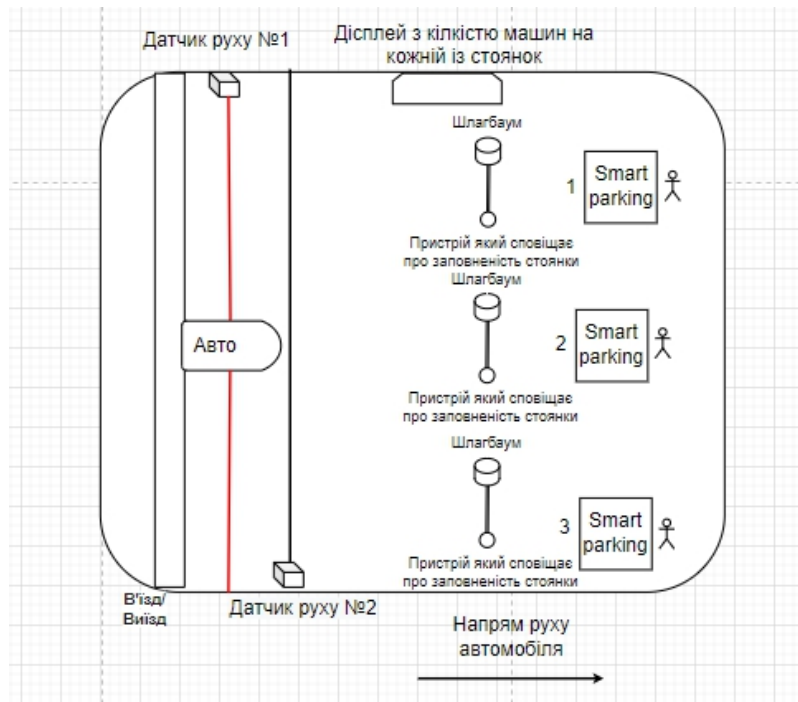


Рисунок 3.13 - Варіант стоянки зі шлагбаумами коли машина заїжджає

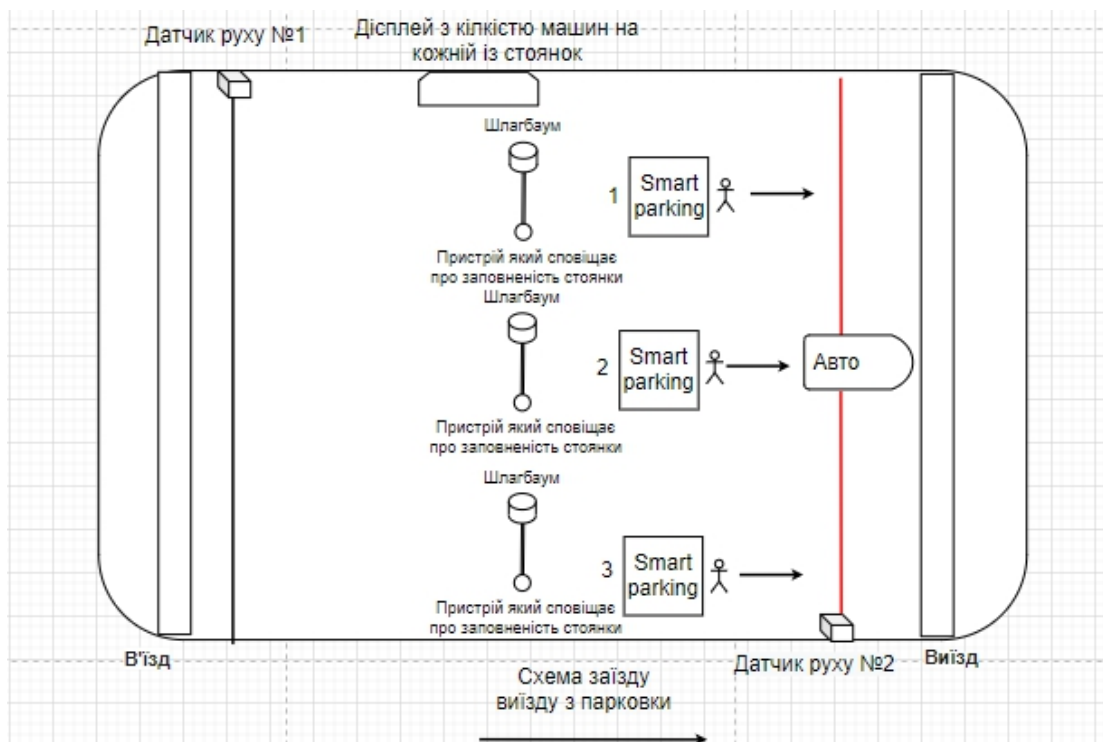


Рисунок 3.14 - Варіант стоянки зі окремих в'їздом та виїздом

Зм.	Арк.	№ докум.	Підпис Дата

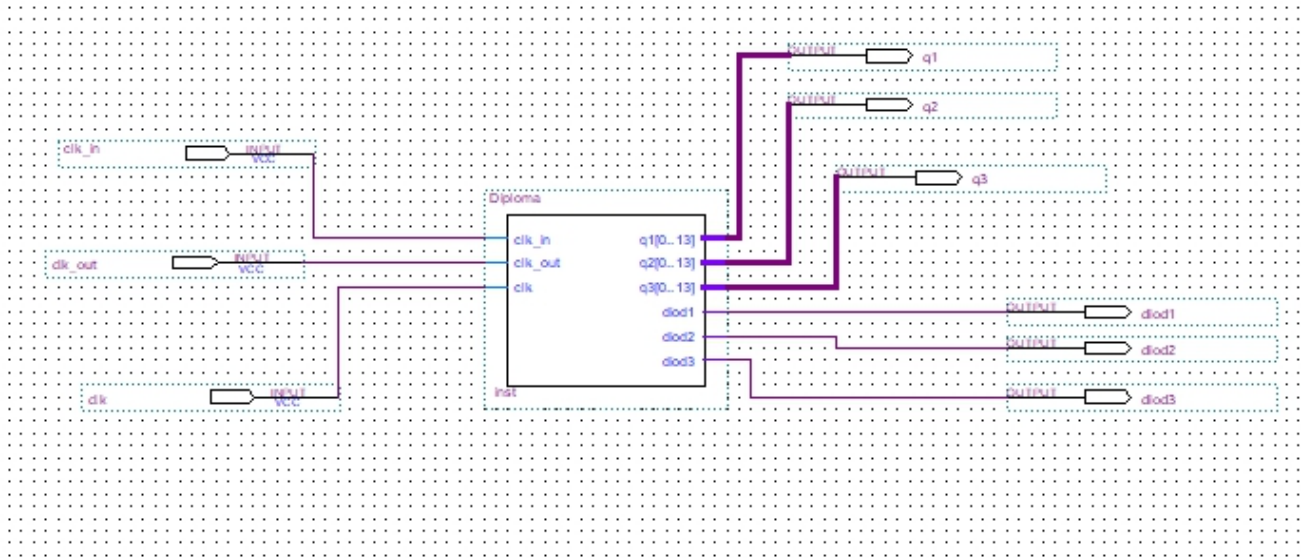


Рисунок 3.15 - Структурна схема ПЛІС у середовищі Quartus

## ВИСНОВКИ

Під час роботи над завданням був ознайомлений з проблемою автоматизованої системи для паркування. Були розглянуті актуальність та проблеми створення таких систем на базі ПЛІС. Було проаналізовано системи аналоги та багато існуючі рішень на даний момент. Також, було коротко проаналізовано особливості кожної з систем. Всі вони були з використанням новітніх технології та дорогими.

Тому було прийнято рішення розробити систему автоматизованої стоянки на базі ПЛІС з використанням підходу програмування на мові VHDL та в майбутньому в комбінації з датчиками руху. Яка б мала малу собівартість та була легка в експлуатації.

Розроблена системи стоянки надає змогу водіям бачити яка кількість місць доступна на даний момент для паркування і на котрій із стоянок є вільні місця.

					КВРКІ.170283.17.02.14 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. IBM Global Parking Survey: Drivers Share Worldwide Parking Woes URL: <https://www-03.ibm.com/press/us/en/pressrelease/35515.wss> (дата звернення: 03.04.2021)
2. Smart Parking Systems URL : <https://smartparkingsystems.com/en/> (дата звернення: 22.03.2021)
3. Офіційний сайт компанії Микком URL: <http://miccom.ru/products/pm-about/pm1> (дата звернення: 22.03.2021)
4. «Розумні парковки» - новий підхід до вирішення проблеми паркінгу в містах. Офіційний сайт компанії Аксіома URL: <http://www.aksioma-group.ru/proekty/sistema-monitoringa-parkovochnykh-mest-vpredelakh-bulvarnogo-koltsa-moskvy/> (дата звернення: 22.03.2021)
5. Все про мікроконтролери. URL: <http://www.maria-online.com/electronics/article.php?lg=uk&q=Мікроконтролер> (дата звернення: 01.05.2021)
6. Мікропроцесори. Структура мікропроцесорів URL: <http://www.yaklass.ru/materiali?chtid=459&mode=cht> (дата звернення: 08.05.2021)
7. Однокристалні мікрокомп'ютери URL: <http://sh.instone.com.ua/tema2.6.php> (дата звернення: 09.05.2021)
8. Характеристики ПЛІС. URL: [https://msn.khnu.km.ua/pluginfile.php/147595/mod\\_resource/content/2/Lekz10.pdf](https://msn.khnu.km.ua/pluginfile.php/147595/mod_resource/content/2/Lekz10.pdf) (дата звернення: 10.05.2021)
9. Altera FLEX 8000. URL: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ds/archives/dsf8k.pdf> (дата звернення: 11.05.2021)

					КВРКІ.170283.17.02.14 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

10. SOC FPGA. URL: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/ab/ab1\\_soc\\_fpga.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ab/ab1_soc_fpga.pdf) (дата звернення: 11.05.2021)
11. Li, X.; Ranga, U.K. Design and implementation of a digital parking lot management system. Technol. Interface J. 2009, 10.
12. Jian, M.-S.; Yang, K. S.; Lee, C.-L. Modular RFID parking management system based on existed gate system integration. WSEAS Trans. Syst. 2008, 7, 706–716.
13. Pala, Z.; Inanç, N. Smart parking applications using RFID technology. In Proceedings of the 1 st RFID Eurasia Conference, Istanbul, Turkey, pp. 121– 123.
14. Yass, A.A.; Yasin, N.M.; Zaidan, B.B.; Zeiden, A.A. New design for intelligent parking system using the principles of management information system and image detection system. In Proceedings of the 2009 International Conference on Computer Engineering and Applications, Manila, Philippines, 6–8 June 2011; Volume 2, pp. 360–364.
15. Bong, D.B.L.; Ting, K.C.; Lai, K.C. Integrated approach in the design of car park occupancy information system (COINS). IAENG Int. J. Comput. Sci. 2008, 35, 7–14.
16. Geng, Y.; Cassandras, C.G. A new ‘smart parking’ system infrastructure and implementation. Procedia–Soc. Behav. Sci. 2012, 54, 1278–1287.
17. Tang, V.W.S.; Zheng, Y.; Cao, J. An intelligent car park management system based on wireless sensor networks.
18. Bi, Y.; Sun, L.; Zhu, H.; Yan, T.; Luo, Z. A parking management system based on wireless sensor network. Acta Autom. 968–977.
19. Benson, J.P.; O’Donovan, T.; O’Sullivan, P.; Roedig, U.; Sreenan, C.; Barton, J.; Murphy, A.; O’Flynn, B. Car-park management using wireless sensor networks. In Proceedings of the 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA pp. 588–595.

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	Арк. 70
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

20. Gu, J.; Zhang, Z.; Yu, F.; Liu, Q. Design and implementation of a street parking system using wireless sensor networks. In Proceedings of the 10th IEEE International Conference on Industrial Informatics (INDIN), Beijing, China, pp. 1212–112

					<i>КВРКІ.170283.17.02.14 ПЗ</i>	<i>Арк.</i>
						<i>71</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## Додаток А

(обов'язковий)

### Лістенинг програми на мові VHDL

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity Project_D
is port
(clk_in: in std_logic;
clk_out, clk: in std_logic;
q1: out std_logic_vector (0 to 13);
q2: out std_logic_vector (0 to 13);
q3: out std_logic_vector (0 to 13));
end entity;

architecture counter_on_max of Project_D
is
begin
process (clk_in,clk_out, clk)
variable count_first, count_first_2, count_first_3 : std_logic_vector(5
downto 0);
variable count_second, count_second_2, count_second_3 :
std_logic_vector(5 downto 0);
variable count_third, count_third_2, count_third_3:std_logic_vector(5
downto 0);
```

```
variable k:std_logic;
begin

--count_first := "000000";
--count_second := "000000";
--count_third := "000000";

--Якщо машини заїжають
if (clk='1' and clk'event) then

if (clk_in='1' and clk_out='0') then

    if(count_first < "110010") then
        count_first:= count_first + '1';

    elsif (count_second < "110010") then
        count_second:= count_second + '1';

    elsif (count_third < "110010") then

        count_third:= count_third + '1';
    else
        count_first:= "110010";
        count_second:= "110010";
        count_third:= "110010";
    end if;

--Якщо машини виїжають
elsif (clk_out='1' and clk_in='0') then
```

```

if (count_third > "000000" and count_third < "110011") then
count_third := count_third - '1';

elsif (count_second > "000000" and count_second < "110011") then
count_second := count_second - '1';

elsif (count_first > "000000" and count_first < "110011") then
count_first := count_first - '1';
end if;

elsif (clk_in='0' and clk_out='0') then NULL;
elsif (clk_in='1' and clk_out='1') then NULL;
end if;

--count_first := "000000";
--count_second := "000000";
--count_third := "000000";
end if;

case count_first is

when "000000" => q1 <= "00000010000001";--00
when "000001" => q1 <= "00000011001111";--01
when "000010" => q1 <= "00000010010010";--02
when "000011" => q1 <= "00000010000110";--03
when "000100" => q1 <= "00000011001100";--04
when "000101" => q1 <= "00000010100100";--05
when "000110" => q1 <= "00000010100000";--06
when "000111" => q1 <= "00000010001111";--07
when "001000" => q1 <= "00000010000000";--08
when "001001" => q1 <= "00000010000100";--09

```

when "001010" => q1<="10011110000001";--10  
when "001011" => q1<="10011111001111";--11  
when "001100" => q1<="10011110010010";--12  
when "001101" => q1<="10011110000110";--13  
when "001110" => q1<="10011111001100";--14  
when "001111" => q1<="10011110100100";--15  
when "010000" => q1<="10011110100000";--16  
when "010001" => q1<="10011110001111";--17  
when "010010" => q1<="10011110000000";--18  
when "010011" => q1<="10011110000100";--19

when "010100" => q1<="00100100000001";--20  
when "010101" => q1<="00100101001111";--21  
when "010110" => q1<="00100100010010";--22  
when "010111" => q1<="00100100000110";--23  
when "011000" => q1<="00100101001100";--24  
when "011001" => q1<="00100100100100";--25  
when "011010" => q1<="00100100100000";--26  
when "011011" => q1<="00100100001111";--27  
when "011100" => q1<="00100100000000";--28  
when "011101" => q1<="00100100000100";--29

when "011110" => q1<="00001100000001";--30  
when "011111" => q1<="00001101001111";--31  
when "100000" => q1<="00001100000001";--32  
when "100001" => q1<="00001100000110";--33  
when "100010" => q1<="00001101001100";--34  
when "100011" => q1<="00001100100100";--35

```
when "100100" => q1<="00001100100000";--36
when "100101" => q1<="00001100001111";--37
when "100110" => q1<="00001100000000";--38
when "100111" => q1<="00001100000100";--39
```

```
when "101000" => q1<="10011000000001";--40
when "101001" => q1<="10011001001111";--41
when "101010" => q1<="10011000010010";--42
when "101011" => q1<="10011000000110";--43
when "101100" => q1<="10011001001100";--44
when "101101" => q1<="10011000100100";--45
when "101110" => q1<="10011000100000";--46
when "101111" => q1<="10011000001111";--47
when "110000" => q1<="10011000000000";--48
when "110001" => q1<="10011000000100";--49
```

```
when "110010" => q1<="01001000000001";--50
```

```
when others => NULL;
end case;
```

```
case count_second is
```

```
when "000000" => q2<="00000010000001";--00
when "000001" => q2<="00000011001111";--01
when "000010" => q2<="00000010010010";--02
when "000011" => q2<="00000010000110";--03
when "000100" => q2<="00000011001100";--04
when "000101" => q2<="00000010100100";--05
```

when "000110" => q2<="00000010100000";--06  
when "000111" => q2<="00000010001111";--07  
when "001000" => q2<="00000010000000";--08  
when "001001" => q2<="00000010000100";--09

when "001010" => q2<="10011110000001";--10  
when "001011" => q2<="10011111001111";--11  
when "001100" => q2<="10011110010010";--12  
when "001101" => q2<="10011110000110";--13  
when "001110" => q2<="10011111001100";--14  
when "001111" => q2<="10011110100100";--15  
when "010000" => q2<="10011110100000";--16  
when "010001" => q2<="10011110001111";--17  
when "010010" => q2<="10011110000000";--18  
when "010011" => q2<="10011110000100";--19

when "010100" => q2<="00100100000001";--20  
when "010101" => q2<="00100101001111";--21  
when "010110" => q2<="00100100010010";--22  
when "010111" => q2<="00100100000110";--23  
when "011000" => q2<="00100101001100";--24  
when "011001" => q2<="00100100100100";--25  
when "011010" => q2<="00100100100000";--26  
when "011011" => q2<="00100100001111";--27  
when "011100" => q2<="00100100000000";--28  
when "011101" => q2<="00100100000100";--29

when "011110" => q2<="00001100000001";--30  
when "011111" => q2<="00001101001111";--31

```
when "100000" => q2<="00001100000001";--32
when "100001" => q2<="00001100000110";--33
when "100010" => q2<="00001101001100";--34
when "100011" => q2<="00001100100100";--35
when "100100" => q2<="00001100100000";--36
when "100101" => q2<="00001100001111";--37
when "100110" => q2<="00001100000000";--38
when "100111" => q2<="00001100000100";--39
```

```
when "101000" => q2<="10011000000001";--40
when "101001" => q2<="10011001001111";--41
when "101010" => q2<="10011000010010";--42
when "101011" => q2<="10011000000110";--43
when "101100" => q2<="10011001001100";--44
when "101101" => q2<="10011000100100";--45
when "101110" => q2<="10011000100000";--46
when "101111" => q2<="10011000001111";--47
when "110000" => q2<="10011000000000";--48
when "110001" => q2<="10011000000100";--49
```

```
when "110010" => q2<="01001000000001";--50
```

```
when others => NULL;
```

```
end case;
```

```
case count_third is
```

```
when "000000" => q3<="00000010000001";--00
```

```
when "000001" => q3<="00000011001111";--01
```

when "000010" => q3<="00000010010010";--02  
when "000011" => q3<="00000010000110";--03  
when "000100" => q3<="00000011001100";--04  
when "000101" => q3<="00000010100100";--05  
when "000110" => q3<="00000010100000";--06  
when "000111" => q3<="00000010001111";--07  
when "001000" => q3<="00000010000000";--08  
when "001001" => q3<="00000010000100";--09

when "001010" => q3<="10011110000001";--10  
when "001011" => q3<="10011111001111";--11  
when "001100" => q3<="10011110010010";--12  
when "001101" => q3<="10011110000110";--13  
when "001110" => q3<="10011111001100";--14  
when "001111" => q3<="10011110100100";--15  
when "010000" => q3<="10011110100000";--16  
when "010001" => q3<="10011110001111";--17  
when "010010" => q3<="10011110000000";--18  
when "010011" => q3<="10011110000100";--19

when "010100" => q3<="00100100000001";--20  
when "010101" => q3<="00100101001111";--21  
when "010110" => q3<="00100100010010";--22  
when "010111" => q3<="00100100000110";--23  
when "011000" => q3<="00100101001100";--24  
when "011001" => q3<="00100100100100";--25  
when "011010" => q3<="00100100100000";--26  
when "011011" => q3<="00100100001111";--27  
when "011100" => q3<="00100100000000";--28

when "011101" => q3<="00100100000100";--29

when "011110" => q3<="00001100000001";--30

when "011111" => q3<="00001101001111";--31

when "100000" => q3<="00001100000001";--32

when "100001" => q3<="00001100000110";--33

when "100010" => q3<="00001101001100";--34

when "100011" => q3<="00001100100100";--35

when "100100" => q3<="00001100100000";--36

when "100101" => q3<="00001100001111";--37

when "100110" => q3<="00001100000000";--38

when "100111" => q3<="00001100000100";--39

when "101000" => q3<="10011000000001";--40

when "101001" => q3<="10011001001111";--41

when "101010" => q3<="10011000010010";--42

when "101011" => q3<="10011000000110";--43

when "101100" => q3<="10011001001100";--44

when "101101" => q3<="10011000100100";--45

when "101110" => q3<="10011000100000";--46

when "101111" => q3<="10011000001111";--47

when "110000" => q3<="10011000000000";--48

when "110001" => q3<="10011000000100";--49

when "110010" => q3<="01001000000001";--50

when others => NULL;

end case;

end process;

end counter\_on\_max;



# Додаток В

(обов'язковий)

## Копія графічної частини Структурна схема

КвРКІ 170283.17.02.14

### Структурна схема

The structural diagram shows a central control unit with the following connections:

- Inputs:**
  - ck\_in (connected to ck1, ck2, ck3)
  - ck\_out (connected to ck1, ck2, ck3)
  - ck (connected to ck1, ck2, ck3)
- Outputs:**
  - q1, q2, q3 (connected to q1, q2, q3)
  - dm1, dm2, dm3 (connected to dm1, dm2, dm3)

### Схема роботи автостоянки

The operational scheme illustrates the parking process:

- Entrance:** Датчик руху №1 (Motion sensor #1) and Шлагбаум (Barrier).
- Interior:** Дюплет з кількості машин на кожній із стоянок (Duplex with number of cars in each bay). Three Smart parking devices are shown, each with the text "Пристрій мей сповіщає про заповненість стоянки шлагбауму" (Device notifies barrier of bay fullness).
- Exit:** Датчик руху №2 (Motion sensor #2) and Шлагбаум (Barrier).
- Vehicle:** Авто (Car) is shown moving through the lot.
- View:** Вгляд (View) is indicated at both ends of the lot.
- Direction:** Стрілка вказує на схемі заїзду вилізу з парковки (Arrow indicates entry/exit from parking).


КвРКІ 170283.17.02.14

Структурна схема і схема автостоянки

Кресло 2 Сторінка 2

КІ-17-2



A stylized illustration of a desk setup. In the center is a laptop with a teal screen and keyboard. To the left is a stack of three books. Above the laptop is a pen holder containing three pens. To the right is a potted plant with green leaves. In the background, there is a framed map or diagram with orange lines on a teal grid. At the bottom right, there is a small window icon representing a software interface.

ПРЕЗЕНТАЦІЯ НА ТЕМУ:

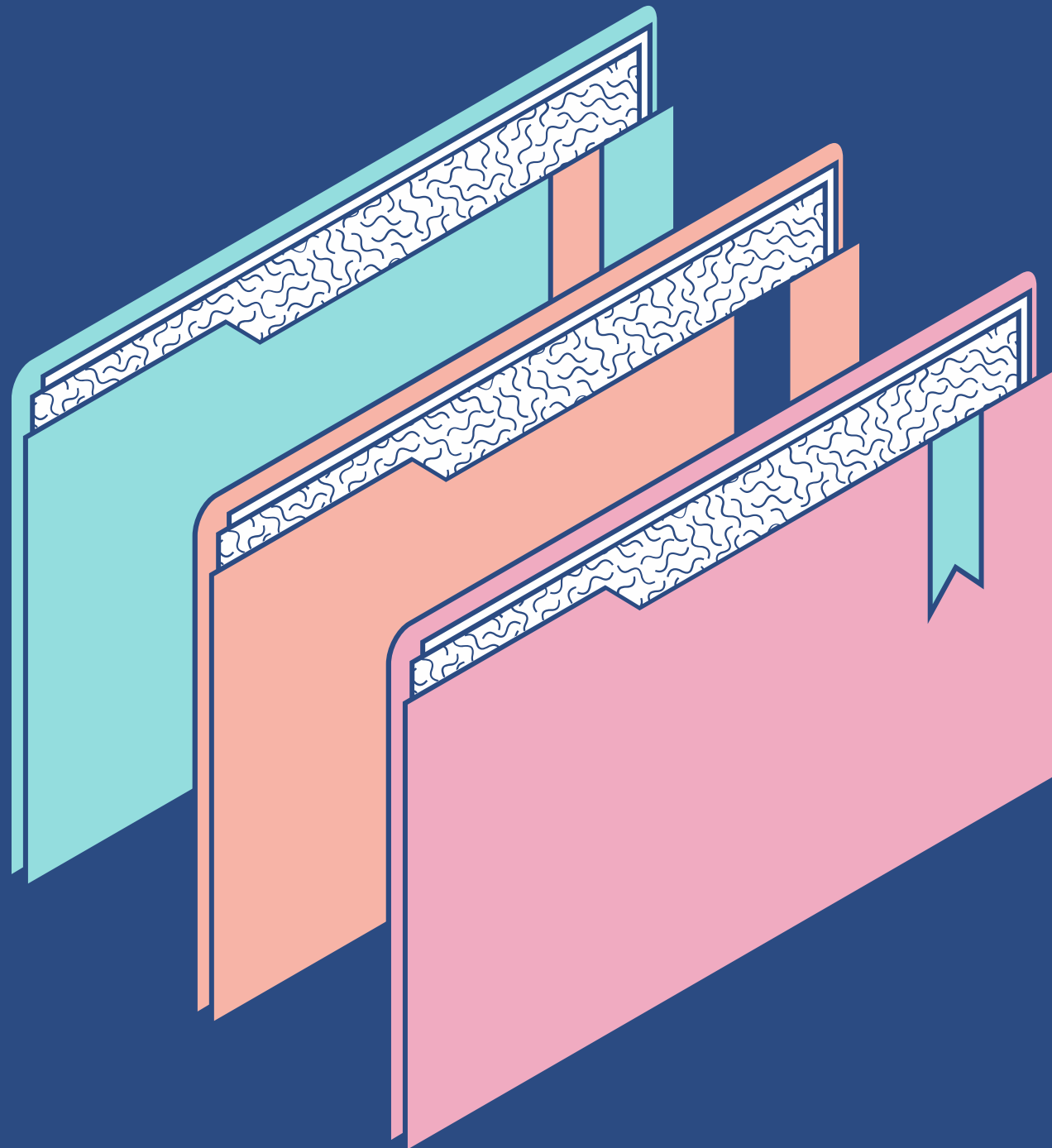
Система на  
програмованій логіці  
для керування  
роботою автостоянки

Студент групи КІ-17-2 Окрушко Д.В.  
Керівник Тітова В.Ю.

# Постановка задачі

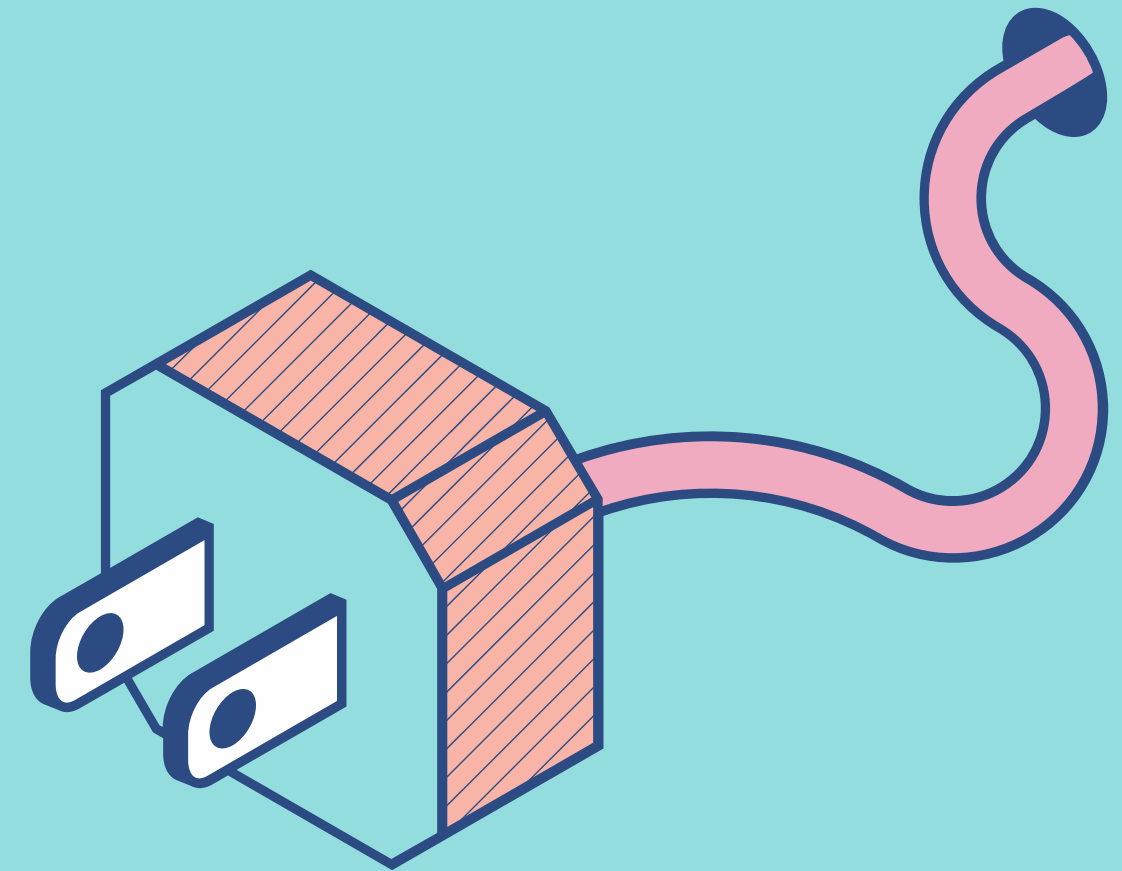
АВТОМАТИЗОВАНА СИСТЕМА ПАРКУВАННЯ МАЄ:

- Дозволяти проводити автоматичний облік кількості вільних та зайнятих місць
- Забезпечувати контроль доступу автомобілів по талонах або якимось іншим чином.
- Спрощення самого процесу паркування для автомобілів різного типу
- Спрощувати контроль грошових розрахунків



# Актуальність проблеми

Автоматизована стоянка - це сучасне рішення, яке дозволить жителям міста почуватись комфортніше, спокійніше та простіше. З кожним роком кількість людей, які купують транспортний засіб зростає, кожен повинен десь її зберігати в безпеці. Для цього автоводії користуються автостоянками з цілодобовою охороною, яка слідкує за порядком на автостоянках.





# Аналіз існуючих рішень

## Smart стоянки

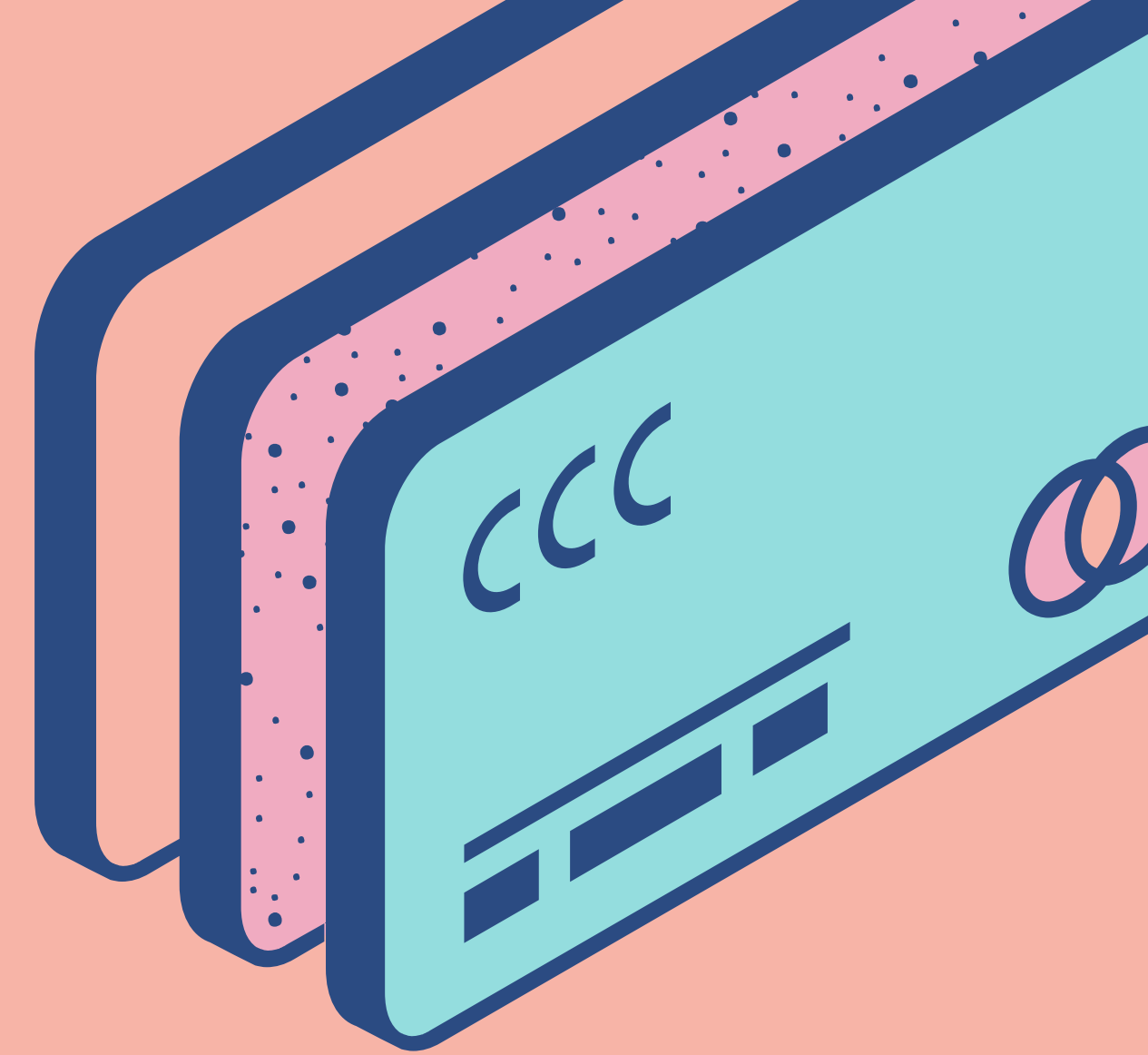
- PARKEON
- ASI01 Pro Park

## Application

- Simple Parking
- Vector AP 2000
- Yandex.parking
- All parking
- Автостоянка 2.5.5
- Автостоянка в Борисполі

# Недоліки систем

SMART & APPLICATION



## Безпека

Smart системи розроблені на базі мікроконтролеїв, тому мають більшу вразливість в порівнянні з ПЛІС.

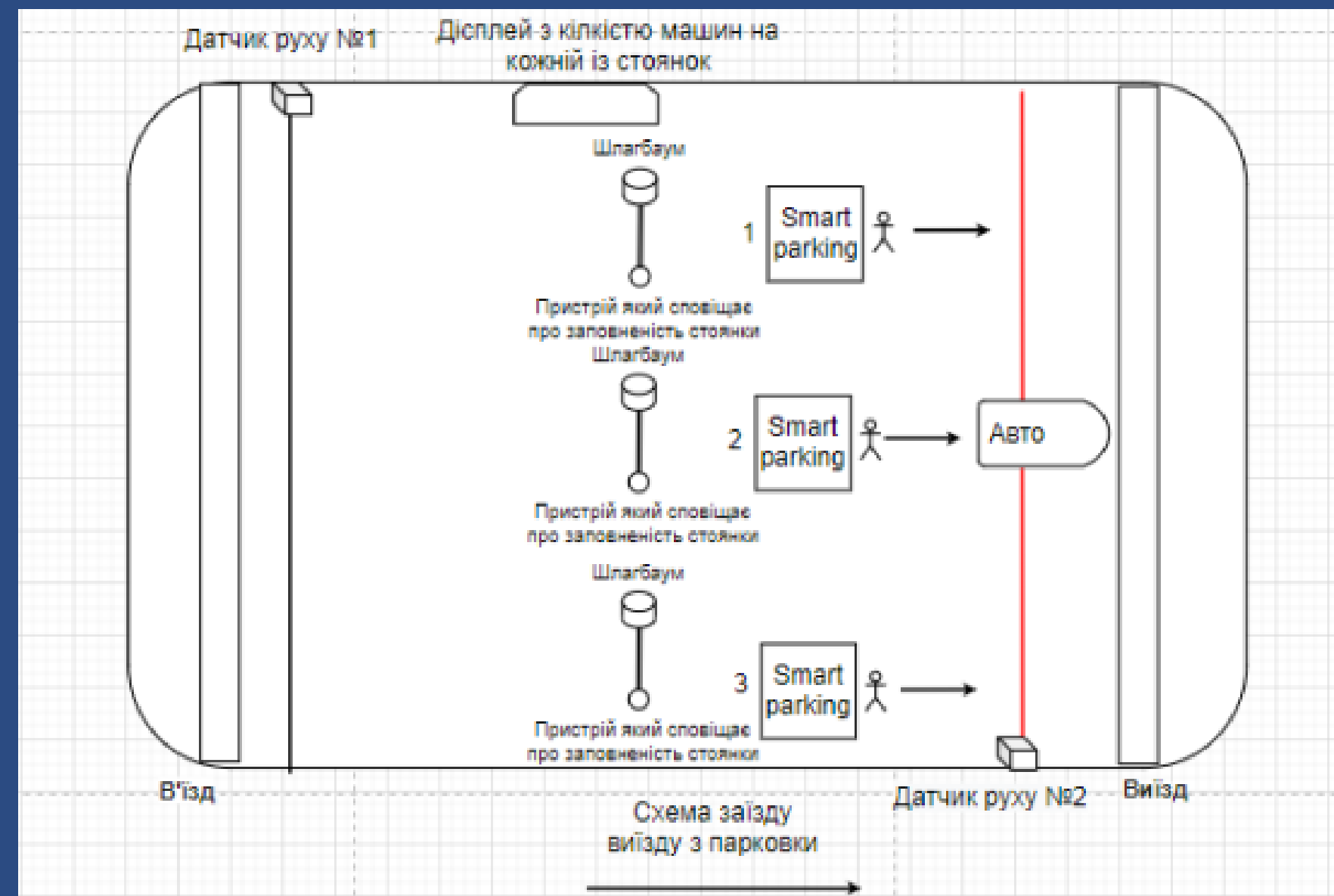
## Користь

Деякі розглянуті рішення були лише інформаційними додатками для різного типу користувачів. І окрім якоїсь інформаційної цінності не несли в собі

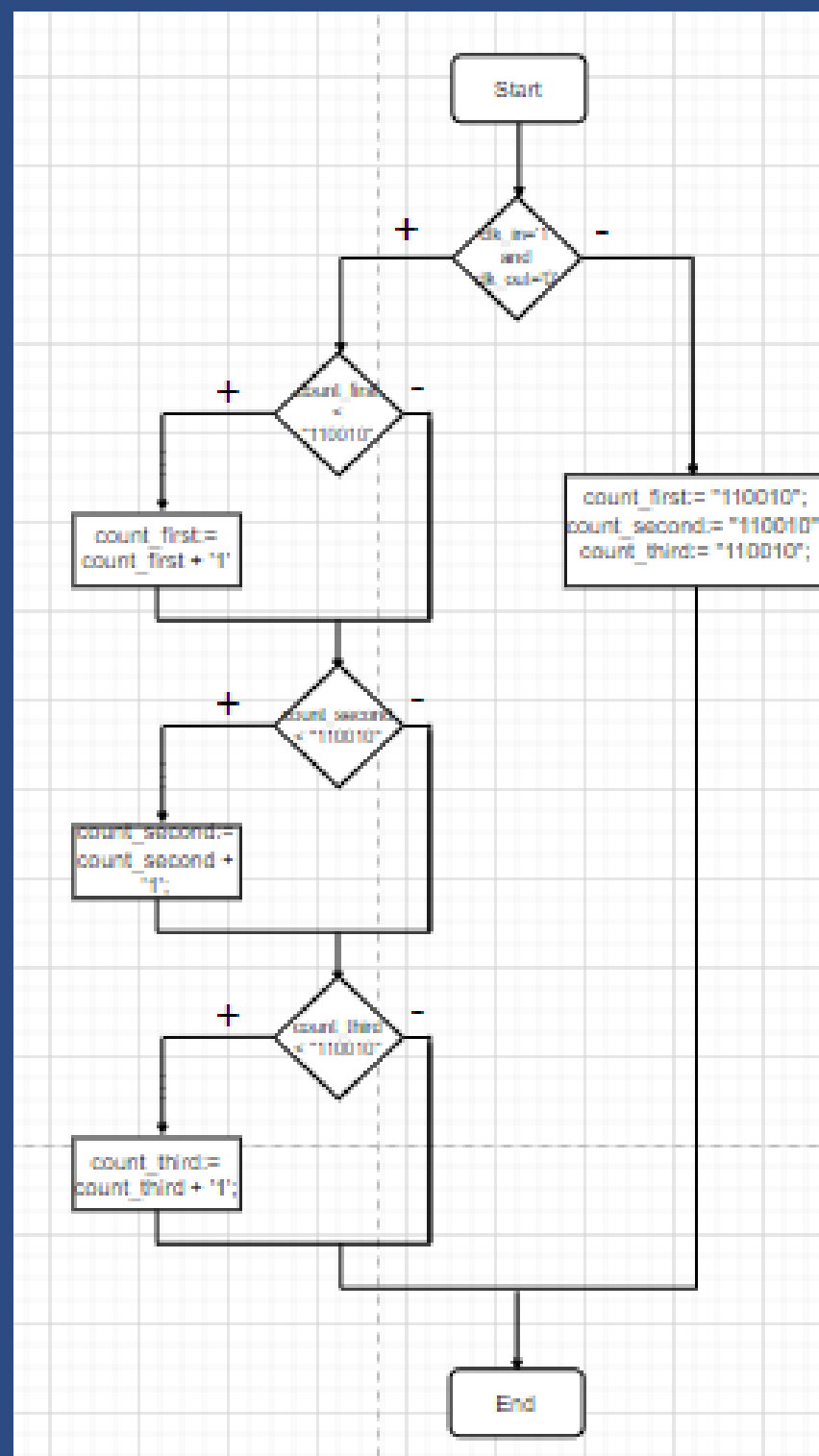
## Ціна

Обидва варіанти були дорогими в розробці

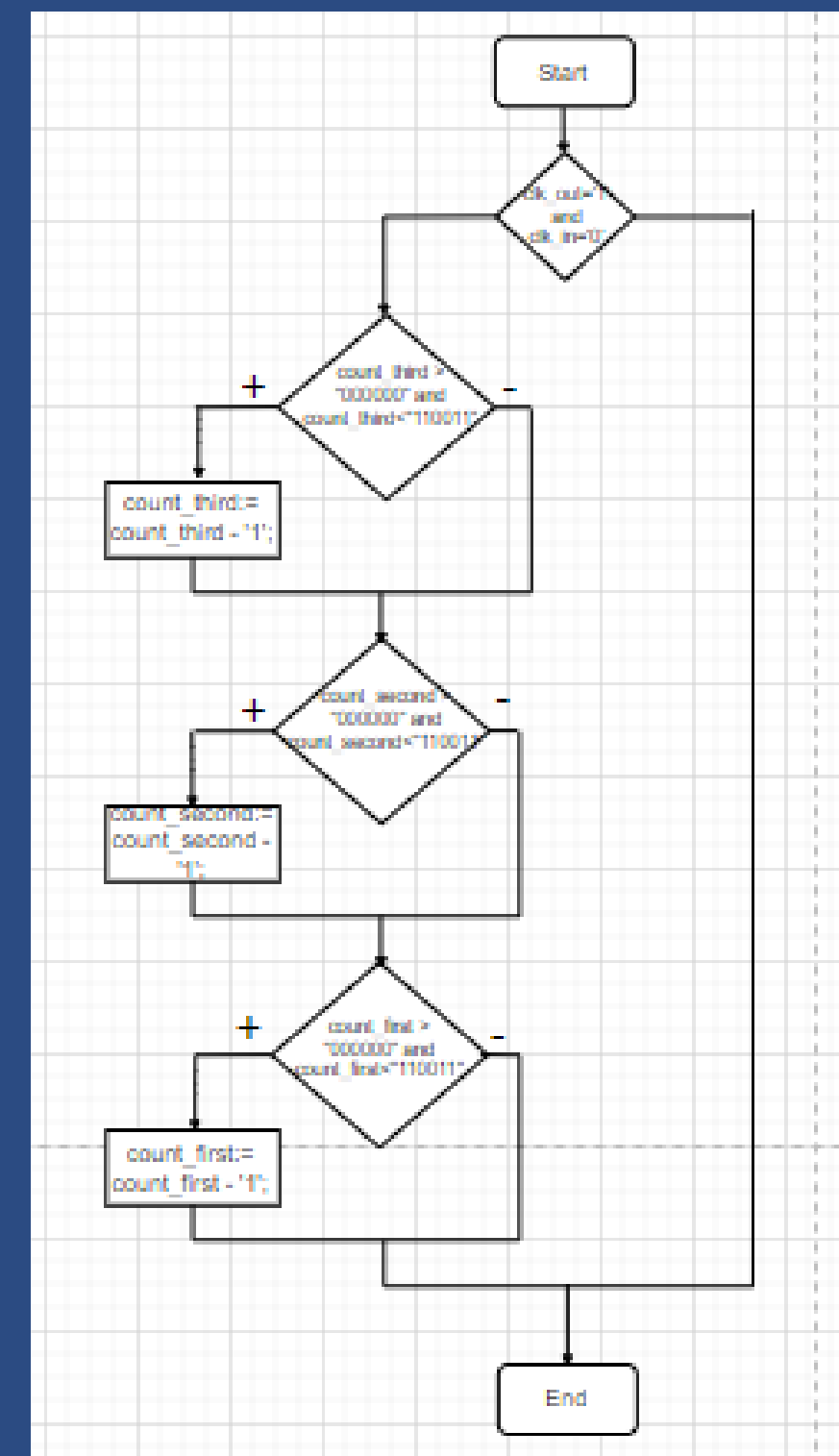
# Схема стоянки



# Алгоритм роботи

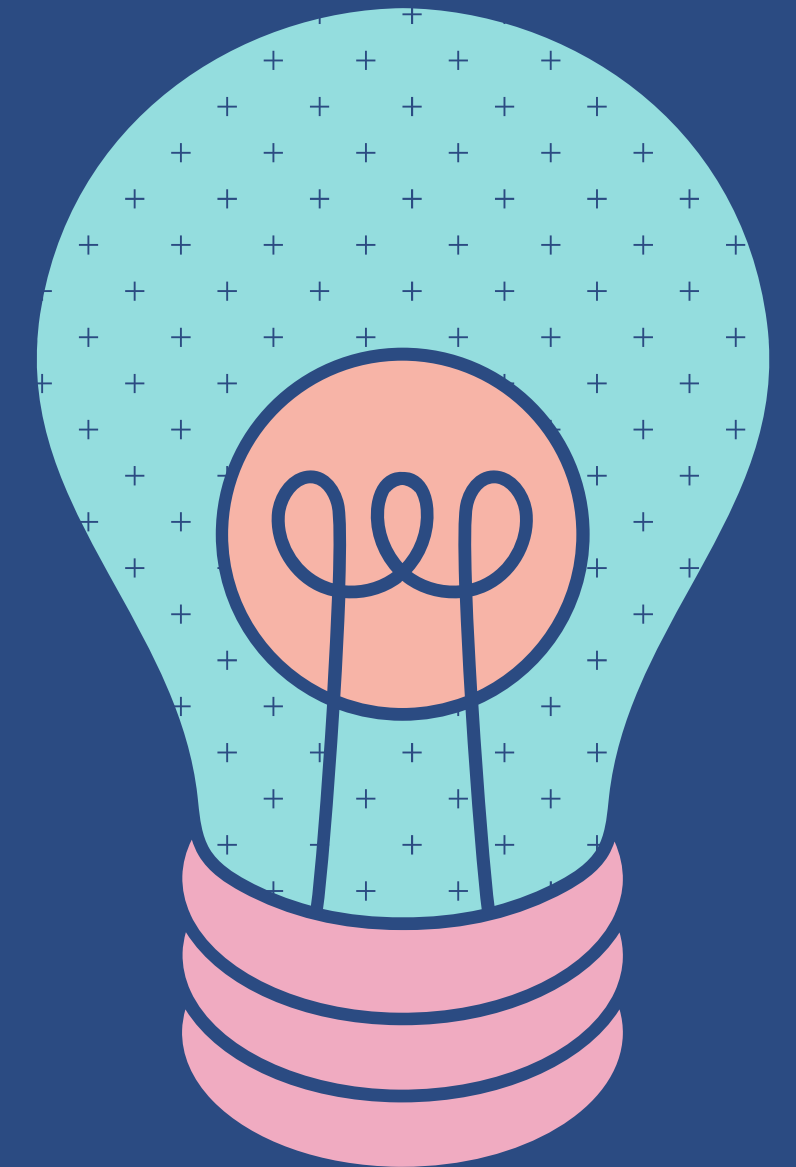
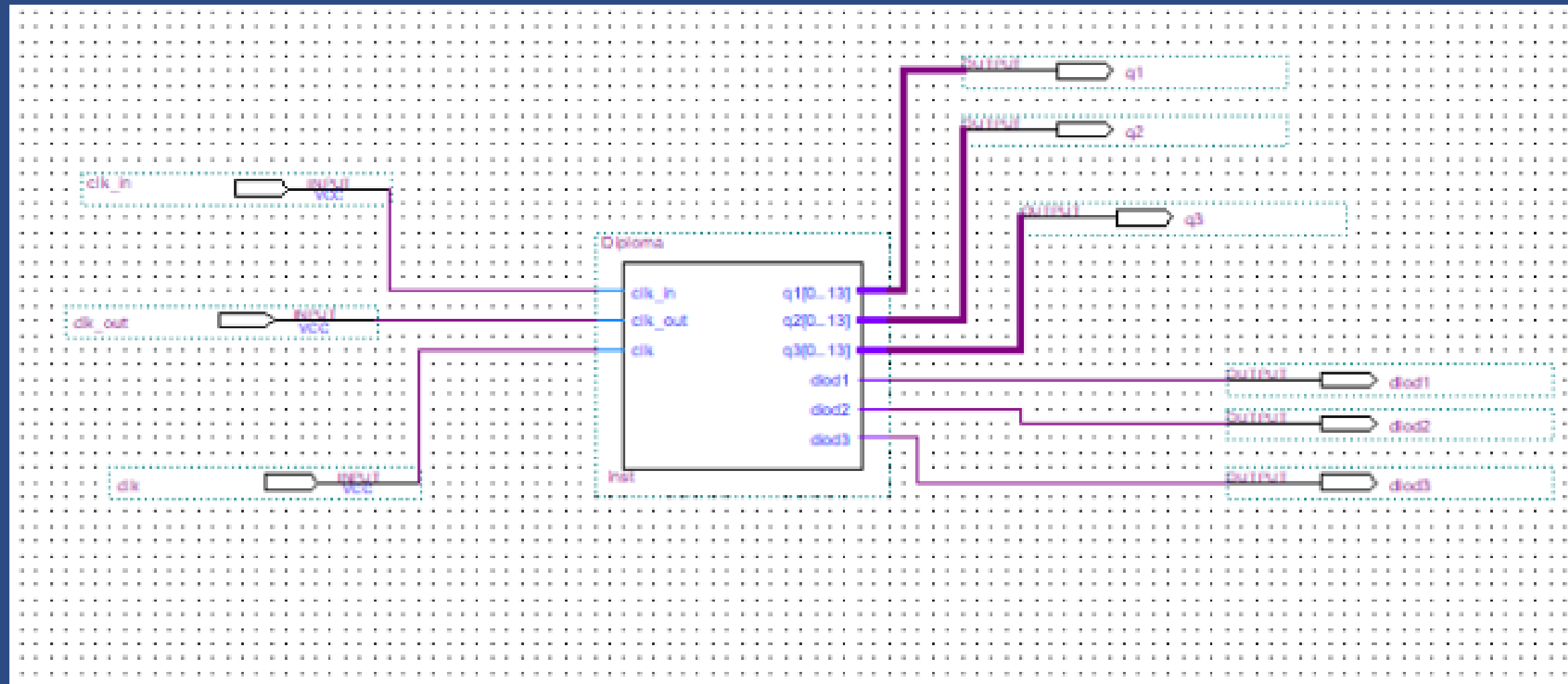


Алгоритм заїзду машин

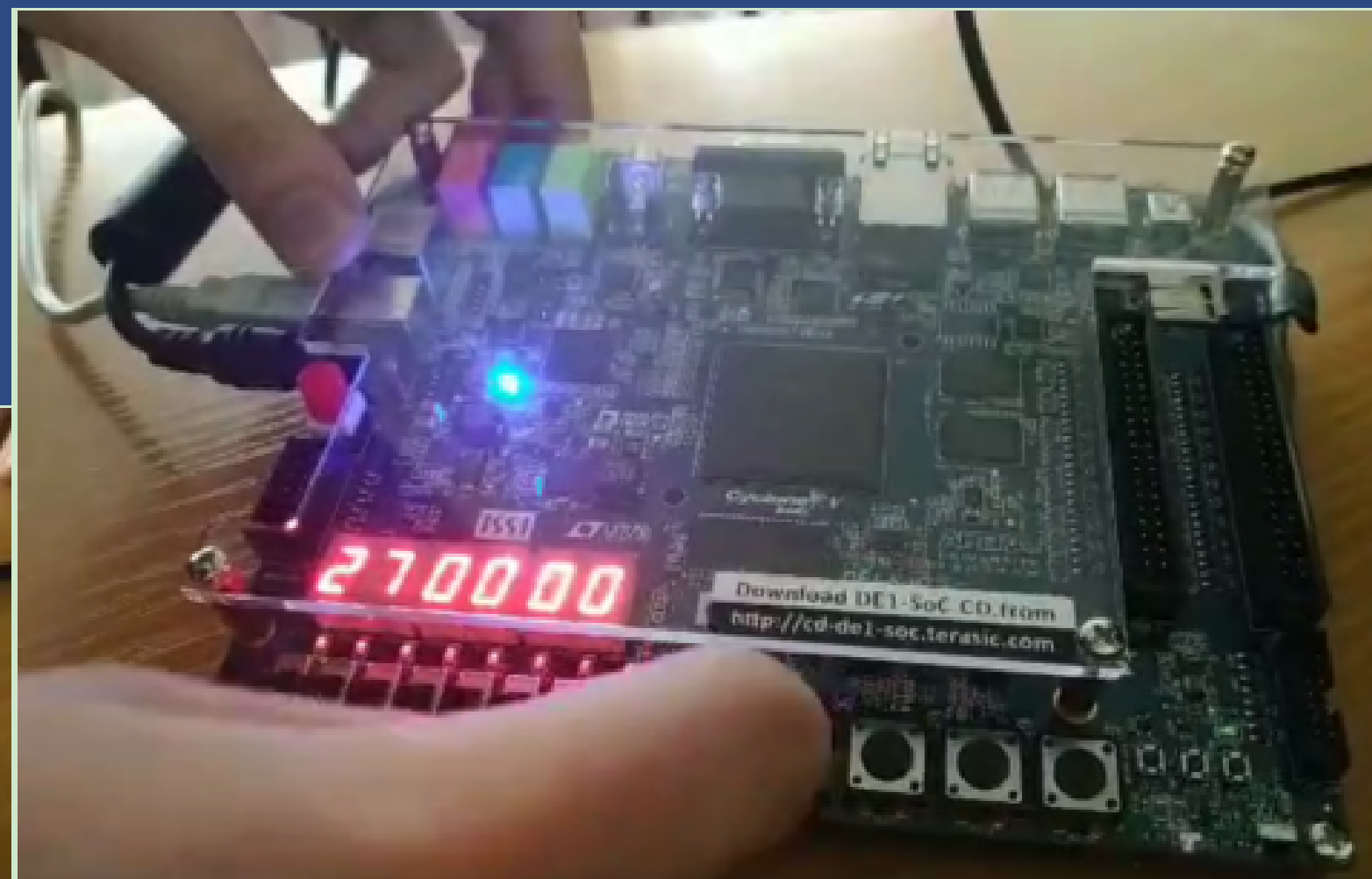
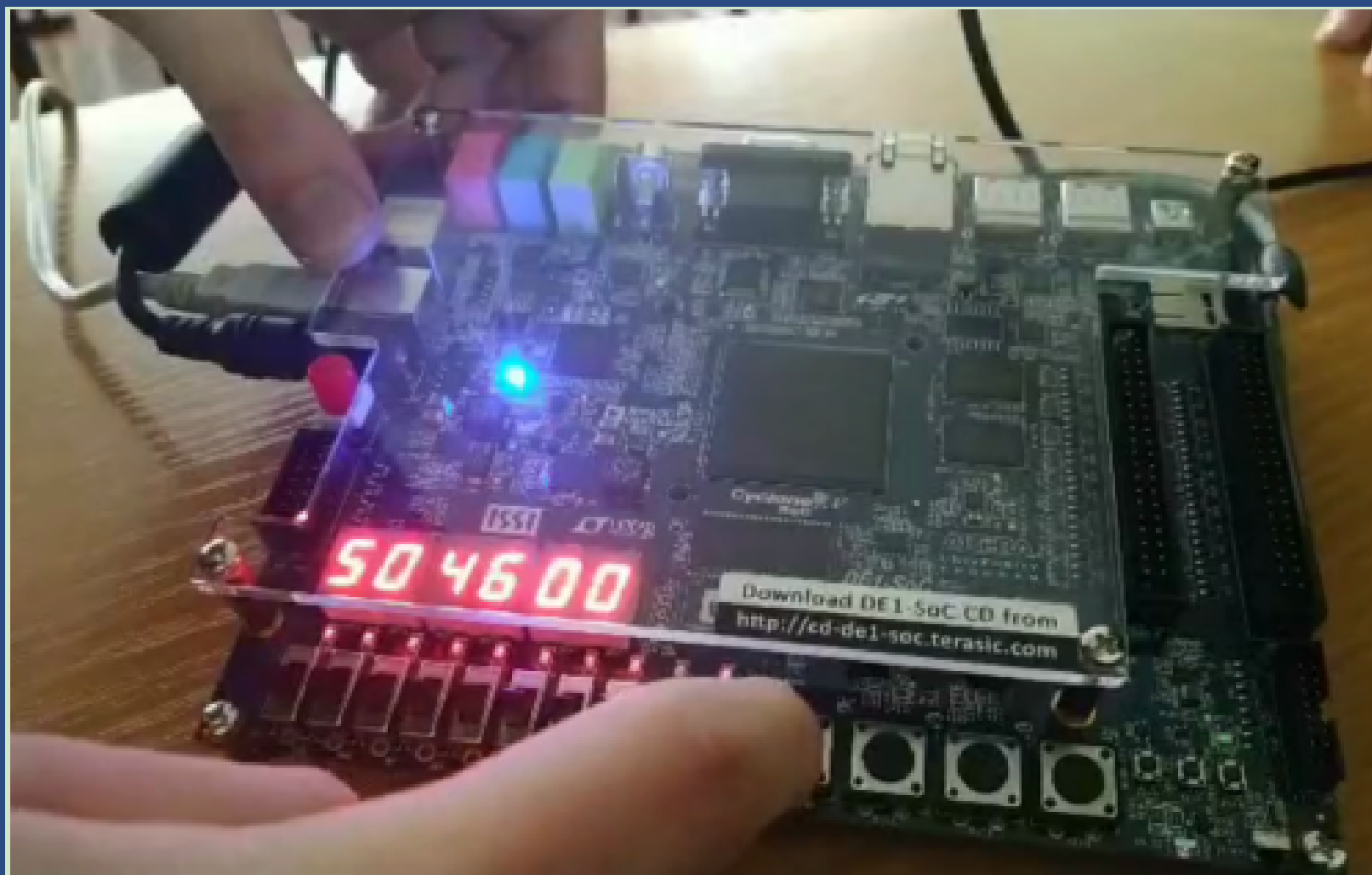


Алгоритм виїзду машин

# Структурна схема



# Результати роботи





**Дякую за увагу**

User name:  
**Кафедра кибербезпеки**

Check ID:  
**1008161677**

Check date:  
**03.06.2021 17:06:57 EEST**

Check type:  
**Doc vs Internet**

Report date:  
**03.06.2021 17:10:12 EEST**

User ID:  
**100005590**

---

File name: **записка**

Page count: **71** Word count: **10052** Character count: **71833** File size: **8.86 MB** File ID: **1008240477**

---

Text modifications detected (similarity score might be affected)

## 15.7% Matches

Highest match: **10.6%** with Internet source ([https://studopedia.net/1\\_48108\\_rozrahunok-obsyagu-vikonuvanih-robot.html](https://studopedia.net/1_48108_rozrahunok-obsyagu-vikonuvanih-robot.html))

15.7% Internet sources 48

Page 73

No Library search was conducted

## 0% Quotes

Exclusion of quotes is off

Exclusion of references is off

## 0% Exclusions

No exclusions

## Modifind

Text modifications detected. Find more details in the online report.

Suspicious formatting 26 Pages

## Anti-Plagiarism v-15.257

**Максимальное совпадение с одним документом 0.0%**

**Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 8%**

ID: 92216 Название: Система на програмованій логіці для керування роботою автостоянки Добавлено в БД: 2021-06-03 Авторы: Д.В. Окрушко Руководители: В.Ю. Тітова Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	48975	492	583 (1%)	7 (1%)

### Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Окрушко Дмитро Віталійович

Тема: Система на програмованій логіці для керування роботою автостоянки

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   72  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є проектування автоматизованої системи на базі ПЛІС для керування роботою автостоянки.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано існуючі системи керування автостоянкою) та виконано постановку задачі роботи. В другому розділі кваліфікаційної роботи проведено аналіз засобів проектування автоматизованих систем керування на ПЛІС. В третьому розділі кваліфікаційної роботи виконано реалізацію системи керування автостоянкою, зокрема спроектовано структурну схему систему, алгоритми роботи її складових, проведено тестування проєктованої системи на засобах ПЛІС.

4. Позитивні сторони роботи: висока практична цінність проведеної роботи, наявність результатів тестування проєктованої системи

---

---

---

---

---

---

---

---

5. Негативні сторони роботи: немає

6. Оцінка графічного оформлення та пояснювальної записки роботи:  
Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення  
документації.

7. Відгук про роботу в цілому: Робота виконана на високому науково-  
технічному рівні

8. Інші зауваження: -

9. Оцінка дипломної роботи: відмінно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) доцент  
кафедри інженерії програмного забезпечення Хмельницького національного  
університету, к.т.н., доцент, Гурман І.В.

" 4 " червня 2021 р.

 (підпис)

Завідувачу кафедри КБСМ  
к-т.техн.наук, доцент. Кльоц Ю.П.

Окрушко Дмитро Віталійович

ПІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ-17-2

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.03.2021

дата

підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система на програмованій логіці для керування роботою автостоянки \_\_\_\_\_

Автор: \_\_\_\_\_ Окрушко Дмитро Віталійович \_\_\_\_\_

Спеціальність: \_\_\_\_\_ 123 – Комп'ютерна інженерія та програмування \_\_\_\_\_

Освітня програма: \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Науковий керівник: \_\_\_\_\_ Тітова Вера Юріївна \_\_\_\_\_

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та депрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 15.7% і адресується до \_\_ першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІСП

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Тітова В.Ю.

С. М. Лисенко

Ю.П. Кльоц