

КВАЛІФІКАЦІЙНА РОБОТА

Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ. 22103.22.02.24 ПЗ

Виконав здобувач IV курсу, група KI2-22-2

Керівник

ДФ

Науковий ступінь, учене звання

Нормоконтролер канд. фіз.-мат. наук, доц.

Науковий ступінь, учене звання

До захисту допускаю:
завідувач кафедри КІС
«01» червня 2026 р.

дата

Підпис

Назар УКРАЇНЕЦЬ

Ініціали, прізвище

Підпис

Юрій ВОЙЧУР

Ініціали, прізвище

Підпис

Тетяна КИСІЛЬ

Ініціали, прізвище

підпис

Ольга ПАВЛОВА

Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Українцеві Назару Богдановичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32

Керівник проекту (роботи) Войчур Ю.О., ДФ, старший викладач

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

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2026 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз відомих систем керування світлофорами

Проектування розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

Реалізація та симуляція розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

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

Структура вузла розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

Часові діаграми роботи системи керування

Формат команди у протоколі передачі даних

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – Аналіз відомих систем керування світлофорами	01.03.2026	виконано
4	Робота над розділом 2 – Проектування розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32	01.04.2026	виконано
5	Робота над розділом 3 – Реалізація розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32	28.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2026	виконано
7	Попередній захист ВКР	25.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач

Підпис

Назар УКРАЇНЕЦЬ

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

Підпис

Юрій ВОЙЧУР

Імя, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32».

Автор роботи *Українець Назар Богданович*.

Керівник роботи: *Войчур Юрій Олексійович*.

Пояснювальна записка: 60 с., 22 рис., 8 табл., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

КЕРУВАННЯ СВІТЛОФОРАМИ, МАШИНА СТАНІВ, СПЕЦІАЛІЗОВАНА КОМП'ЮТЕРНА СИСТЕМА.

Мета кваліфікаційної роботи: розробка та верифікація розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та мікроконтролера ESP32, в якій логіка керування кожним перехрестям реалізована у вигляді скінченного автомата на програмованій логічній інтегральній схемі.

У сучасних містах проблема заторів на дорогах набула критичного характеру, перетворившись на одну з ключових перешкод для сталого розвитку урбанізованих територій. Щоденні затори спричиняють величезні економічні втрати через витрату пального, зниження продуктивності праці, зростання викидів вуглекислого газу та інших шкідливих речовин, а також суттєво погіршують якість життя населення через постійний стрес, втрату часу та підвищений ризик аварій. Традиційні системи керування світлофорами, які працюють за жорстко фіксованими циклами та заздалегідь заданими таймінгами, вже давно не відповідають реальним умовам руху, особливо в години пік, під час нештатних ситуацій чи при нерівномірному розподілі транспортних потоків. Саме тому актуальність переходу до розподілених інтелектуальних систем керування дорожнім рухом стає дедалі очевиднішою: такі системи здатні в реальному часі аналізувати поточну ситуацію на перехрестях, адаптивно змінювати тривалість фаз світлофорів, координувати роботу сусідніх вузлів і навіть отримувати пріоритет для громадського транспорту, екстрених служб чи пішоходів.


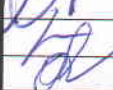
Підпис студента



Дата

ЗМІСТ

Скорочення та умовні позначки.....	4
Вступ.....	5
1 Аналіз відомих систем керування світлофорами.....	7
1.1 Аналіз відомих систем керування світлофорами.....	7
1.2 Аналіз відомих стратегій для систем керування світлофорами.....	9
1.3 Теоретичні основи керування світлофорами.....	12
1.3.1 Математичні моделі перехрестя.....	13
1.3.2 Алгоритми керування світлофорами.....	16
1.3.3 Критерії ефективності системи керування.....	19
1.4 Висновки. Постановка задачі.....	20
2 Проектування розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32.....	22
2.1 Постановка вимог до розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32.....	22
2.2 Архітектура розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32.....	25
2.3 Опис протоколу передачі даних, формат команди.....	30
2.4 Аналіз та вибір апаратних засобів.....	35
2.5 Висновки.....	40
3 Реалізація та симуляція розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32.....	41

КвРКІ. 22103.22.02.24 ПЗ					
Зм.	Арк.	№докум.	Підпис	Дата	
Виконав		Українець Н.Б.			
Перевір.		Войчур Ю.О.			
Н.контр.		Кисіль Т.М.	Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32		
Затвер.		Павлова О.О.			
			Літера	Аркуш	Аркушів
				2	60
ХНУ, КІ2-22-2					

3.1 Реалізація розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 у середовищі Quartus II.....	41
3.2 Симуляція розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 в ModelSim Altera.....	46
3.2.1 Симуляція керування одним світлофором за допомогою машини станів	46
3.2.2 Симуляція розподіленої спеціалізованої комп'ютерної системи керування світлофорами.....	51
3.3 Висновки	58
Висновки	59
Перелік джерел посилань	61
Додаток А Копія креслення «Структура вузла розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32».....	71
Додаток Б Копія креслення «Часові діаграми роботи системи керування»	72
Додаток В Копія креслення «Формат команди у протоколі передачі даних»	73

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ВМ – Вбудований мікропроцесор

ПЛІС – Програмована логічна інтегральна схема

МК – Мікроконтролер

СКС – Спеціалізована комп'ютерна система

ІТС – Інтелектуальна транспортна система

РСК – Розподілена система керування

ЦОД – Цикл обслуговування дороги

ШІМ – Широтно-імпульсна модуляція

UART – Універсальний асинхронний приймач-передавач (Universal Asynchronous Receiver-Transmitter)

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						4
Зм.	Арк.	№докум.	Підпис	Дата		

ВСТУП

У сучасних містах проблема заторів на дорогах набула критичного характеру, перетворившись на одну з ключових перешкод для сталого розвитку урбанізованих територій. Щоденні затори спричиняють величезні економічні втрати через витрату пального, зниження продуктивності праці, зростання викидів вуглекислого газу та інших шкідливих речовин, а також суттєво погіршують якість життя населення через постійний стрес, втрату часу та підвищений ризик аварій. Традиційні системи керування світлофорами, які працюють за жорстко фіксованими циклами та заздалегідь заданими таймінгами, вже давно не відповідають реальним умовам руху, особливо в години пік, під час нештатних ситуацій чи при нерівномірному розподілі транспортних потоків. Саме тому актуальність переходу до розподілених інтелектуальних систем керування дорожнім рухом стає дедалі очевиднішою: такі системи здатні в реальному часі аналізувати поточну ситуацію на перехрестях, адаптивно змінювати тривалість фаз світлофорів, координувати роботу сусідніх вузлів і навіть отримувати пріоритет для громадського транспорту, екстрених служб чи пішоходів.

Одним із перспективних напрямків реалізації таких рішень є використання FPGA (програмованих логічних матриць) у поєднанні з недорогими та енергоефективними мікроконтролерами на кшталт ESP32. FPGA забезпечують надзвичайно високу швидкість обробки паралельних сигналів, надійну реалізацію складних кінцевих автоматів, детерміноване реагування в мікросекундному діапазоні та можливість інтеграції апаратних модулів для обробки даних від великої кількості датчиків – усе це критично важливо для безпечного та надійного керування світлофорами в реальному часі. Водночас ESP32 виступає ідеальним комунікаційним та обчислювальним вузлом нижчого рівня: завдяки вбудованому Wi-Fi, Bluetooth, підтримці протоколів IoT (MQTT, HTTP, WebSocket), низькому енергоспоживанню та достатній обчислювальній потужності він дозволяє створювати розподілену мережу перехресть, обмінюватися даними про щільність

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 5
Зм.	Арк.	№докум.	Підпис	Дата		

трафіку, синхронізувати фази між сусідніми вузлами, передавати статистику в хмару чи диспетчерський центр, а також швидко реагувати на локальні події.

Такий гібридний підхід, за якого FPGA забезпечує детерміновану апаратну реалізацію критичної логіки керування на рівні окремого перехрестя, а мікроконтролер ESP32 відповідає за мережеву інтеграцію, передачу конфігураційних даних та координацію між вузлами, відкриває перспективи для побудови розподілених спеціалізованих комп'ютерних систем керування світлофорами нового покоління. Такі системи здатні функціонувати автономно на локальному рівні з гарантованими часовими характеристиками, не залежачи від стану мережі чи центрального сервера, і водночас інтегруватися в єдину міську інфраструктуру інтелектуального транспорту для координованого управління трафіком у масштабах міста.

Метою роботи є розробка та верифікація розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та мікроконтролера ESP32, в якій логіка керування кожним перехрестям реалізована у вигляді скінченного автомата на програмованій логічній інтегральній схемі.

Об'єктом дослідження є процеси автоматизованого керування виконавчими пристроями світлофорів на основі розподілених обчислювальних вузлів з використанням гібридної апаратної архітектури FPGA та ESP32.

Предметом дослідження є методи реалізації детермінованого керування світлофорними фазами на FPGA, а також програмно-технічні засоби організації розподіленої мережевої інфраструктури для обміну даними.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						6
Зм.	Арк.	№докум.	Підпис	Дата		

1 АНАЛІЗ ВІДОМИХ СИСТЕМ КЕРУВАННЯ СВІТЛОФОРАМИ

1.1 Аналіз відомих систем керування світлофорами

Аналіз відомих систем керування світлофорами розкриває еволюцію технологій від простих механічних пристроїв до складних інтелектуальних мереж, які інтегрують штучний інтелект, машинне навчання та датчики реального часу для оптимізації трафіку в урбанізованих середовищах. Початок цієї еволюції сягає кінця XIX століття, коли перші світлофори з'явилися для регулювання руху коней і пішоходів, а вже на початку XX століття, зокрема в 1912 році в Солт-Лейк-Сіті, були впроваджені електричні системи з червоним і зеленим сигналами, доповнені звуковим попередженням про зміну фаз, що заклало основу для автоматизованого контролю. У 1920-х роках у Детройті з'явилися триколірні системи з чотирьохстороннім розміщенням, які дозволяли координувати рух на перехрестях, але вони все ще працювали за фіксованими циклами, не враховуючи реальну інтенсивність трафіку, що призводило до неефективності в пікові години та збільшення заторів.

З середини XX століття почався перехід до адаптивних систем, які динамічно змінювали тривалість сигналів на основі даних від датчиків, таких як індукційні петлі чи камери, дозволяючи реагувати на поточні умови руху. Однією з найвідоміших таких систем є SCOOT (Split Cycle Offset Optimisation Technique), розроблена у Великобританії в 1970-х роках, яка оптимізує поділ циклу, зсув фаз і загальну тривалість сигналів для мережі перехресть, аналізуючи дані в реальному часі для зменшення затримок і підвищення пропускної здатності, що особливо ефективно в густонаселених містах, хоча вимагає значних інвестицій у інфраструктуру та централізоване керування, що може бути вразливим до збоїв у комунікаціях. Аналогічно, австралійська SCATS (Sydney Coordinated Adaptive Traffic System), впроваджена в 1960-х і постійно удосконалювана, координує сигнали на основі локальних і глобальних даних, дозволяючи адаптацію до змін трафіку, і показала себе ефективною в зменшенні середнього часу очікування на 20-30% у великих мегаполісах, але її недоліком є

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 7
Зм.	Арк.	№докум.	Підпис	Дата		

залежність від точності датчиків і потенційна перевантаженість системи в умовах екстремальних подій, як аварії чи масові заходи.

У США поширеними є системи на кшталт RHODES (Real-time Hierarchical Optimized Distributed Effective System), яка застосовує ієрархічний підхід для розподіленого керування, оптимізуючи трафік на рівні окремих перехресть і цілих мереж за допомогою алгоритмів реального часу, що дозволяє інтегрувати пріоритети для громадського транспорту чи екстрених служб, зменшуючи загальні затримки, хоча її складність вимагає висококваліфікованого обслуговування.

Інша система, ОРАС (Optimized Policies for Adaptive Control), фокусується на віртуальних фіксованих циклах з динамічною корекцією, забезпечуючи гнучкість у відповідь на флуктуації трафіку, і часто поєднується з ACS Lite для спрощених версій, що робить її доступною для менших міст, але з меншою точністю в порівнянні з повноцінними адаптивними мережами. Ці системи загалом покращують ефективність на 10-20% порівняно з фіксованими, зменшуючи витрати палива та викиди, однак їхня вразливість до кібератак і залежність від стабільного живлення стають критичними проблемами в сучасних умовах.

З появою штучного інтелекту та машинного навчання в 2010-2020-х роках системи керування світлофорами перейшли до інтелектуального рівня, де алгоритми, такі як нечітка логіка типу Mamdani, дозволяють моделювати невизначеність трафіку, динамічно регулюючи зелене світло на основі вхідних параметрів, як щільність руху чи черги транспортних засобів, що демонструє кращі результати в симуляціях для трисигнальних перехресть, зменшуючи затримки на 15-25% порівняно з традиційними методами. Наприклад, системи з підтримкою векторної регресії та випадковими лісами для прогнозування трафіку аналізують історичні дані для передбачення піків, оптимізуючи сигнали заздалегідь, що особливо корисно в динамічних урбаністичних середовищах, хоча вимагає значних обчислювальних ресурсів і якісних даних для навчання моделей. Новітні розробки, як Passable, інтегрують глибоке навчання для

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 8
Зм.	Арк.	№докум.	Підпис	Дата		

реального моніторингу з виявленням інцидентів і сповіщенням транспортних засобів, поєднуючи камери з AI для адаптації сигналів, що не тільки зменшує затори, але й підвищує безпеку, попереджаючи про потенційні аварії.

У реальних застосуваннях ці технології впроваджуються в містах по всьому світу: Нью-Йорк і Токіо використовують інтелектуальні системи з датчиками та AI для мінімізації ризиків аварій і оптимізації потоку, тоді як Копенгаген акцентує на пріоритеті для велосипедистів і пішоходів, інтегруючи екологічні аспекти. Загалом, аналіз показує, що відомі системи еволюціонували від статичних до динамічних і прогнозуючих, значно покращуючи ефективність трафіку, але стикаються з викликами, як висока вартість, вразливість до зовнішніх факторів і потреба в інтеграції з IoT та V2X для повної автономності, що відкриває шлях для гібридних рішень на базі FPGA та мікроконтролерів, здатних поєднувати швидкість обробки з розподіленою архітектурою для подолання цих обмежень.

1.2 Аналіз відомих стратегій для систем керування світлофорами

Сучасні стратегії керування дорожнім рухом, що ґрунтуються на методах оптимізації, будуються на фундаментальному припущенні, що динаміку транспортних потоків можна математично описати, а майбутні стани системи – спрогнозувати з достатньою точністю. Основна концепція цих підходів полягає у формулюванні складних математичних задач, розв’язання яких дозволяє знайти ідеальні плани перемикання сигналів світлофорів. Головними критеріями ефективності в таких моделях виступають мінімізація сумарних затримок транспорту та скорочення довжини черг на підходах до перехресть. Проте практична реалізація таких методів у середовищі підключених транспортних засобів стикається з низкою критичних викликів. Зокрема, складність моделей різко зростає при спробі врахувати конфліктні потоки на кількох перехрестях одночасно, що потребує значних обчислювальних ресурсів та надточної прогностичної аналітики. Залежно від способу подолання цих труднощів,

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 9
Зм.	Арк.	№докум.	Підпис	Дата		

дослідники класифікують оптимізаційні методи на централізовані, децентралізовані та ієрархічні.

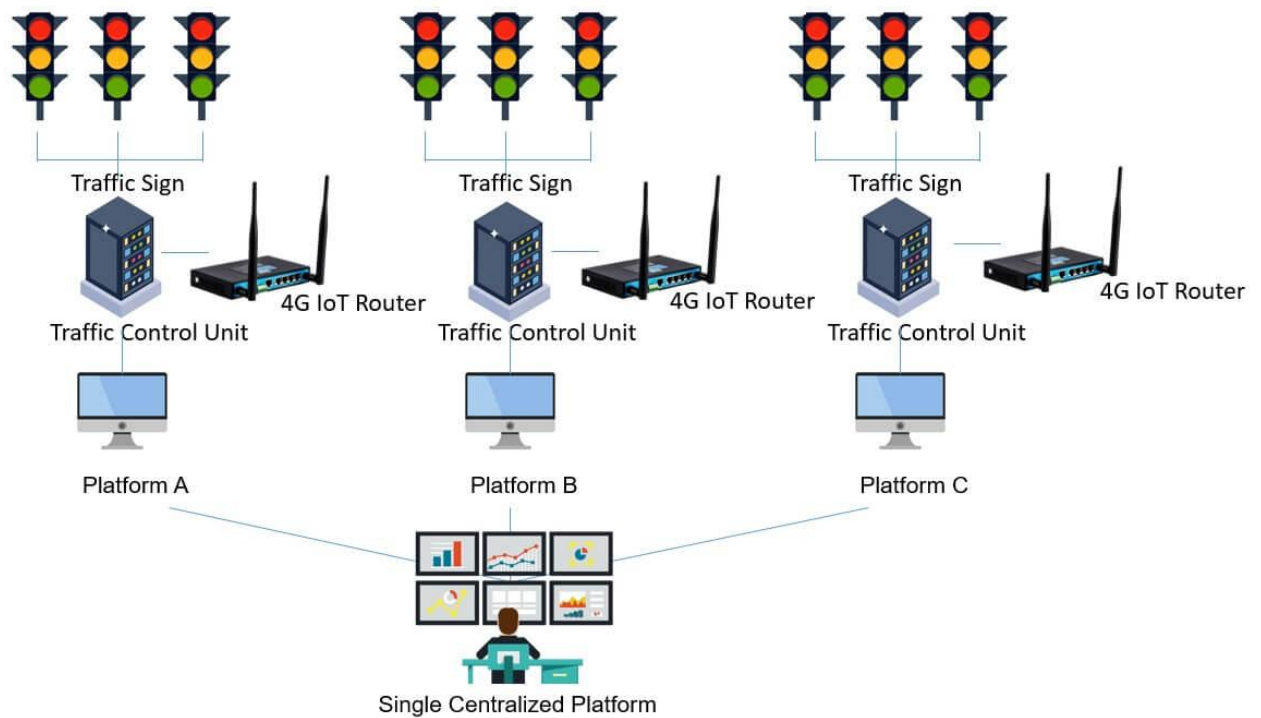


Рисунок 1.1 – Централізована ієрархічна система керування світлофорами

Централізовані підходи намагаються охопити транспортну мережу як єдину систему, прагнучи до глобального оптимуму (рис. 1.2). Оскільки пряме моделювання кожного окремого автомобіля створює надмірну кількість змінних, розробники використовують стратегії спрощення, наприклад, об'єднання транспортних засобів у «псевдо-платони» на основі часових інтервалів між ними. Це дозволяє застосовувати апарат змішано-цілочисельного лінійного програмування для розрахунку оптимальних послідовностей фаз у реальному часі. Окрім традиційного часового регулювання, сучасні централізовані моделі дедалі частіше інтегрують екологічні показники, такі як сумарне споживання палива, та використовують складні алгоритми – від динамічного програмування до вдосконалених генетичних методів та лінійно-квадратичних регуляторів. Такі системи здатні динамічно оновлювати модель трафіку, адаптуючись до результатів

Зм.	Арк.	№докум.	Підпис	Дата

попередніх рішень, що забезпечує високу точність керування у великих масштабах, попри високу обчислювальну вартість.

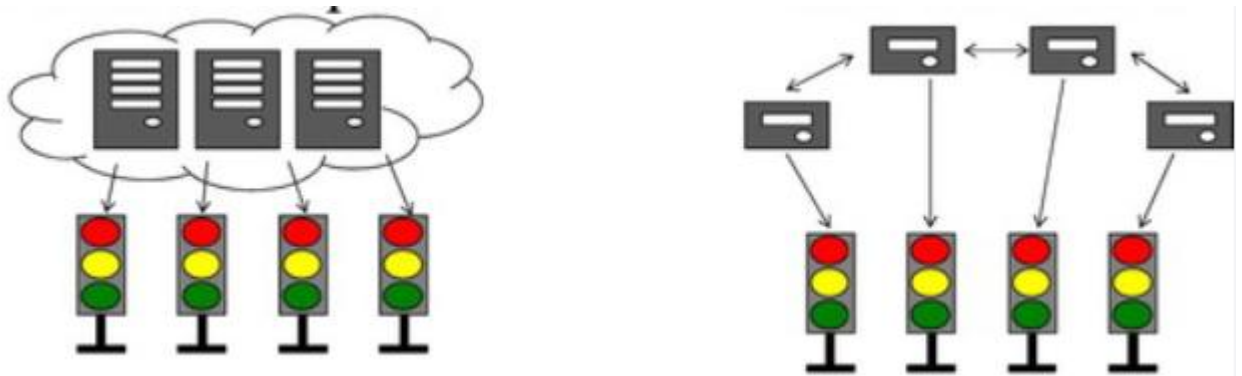


Рисунок 1.2 – Централізована та розподілена архітектура системи керування світлофорами

На противагу їм, децентралізовані підходи спрямовані на зниження обчислювальної складності шляхом розподілу задач між локальними контролерами кожного перехрестя. Такі системи фокусуються на індивідуальній оптимізації об'єктів, часто ігноруючи безпосередню координацію з сусідами, що робить їх швидшими, але потенційно менш ефективними на глобальному рівні. Типовим рішенням тут є використання концепції «ковзного горизонту» прогнозування, де система оцінює ситуацію на найближчі 15-30 секунд. Особлива увага в децентралізованих методах приділяється запобіганню заторам через максимізацію пропускну здатності кожного виїзду з перехрестя окремо. Важливим аспектом є також робота в умовах змішаного потоку, де лише частина авто є підключеними; у таких випадках локальні контролери використовують дані з детекторів та алгоритми слідування за лідером для оцінки стану непідключених транспортних засобів, формуючи оптимальні черги відправлення платонів.

Ієрархічні підходи пропонують структуру, що поєднує переваги обох попередніх методів, розділяючи задачу на кілька рівнів – зазвичай макроскопічний та мікроскопічний. На верхньому рівні (наприклад, рівні транспортного коридору або району) вирішуються стратегічні задачі, такі як оптимізація зсувів фаз для створення «зеленої хвилі», що стає обмеженням для нижнього рівня. На рівні

Зм.	Арк.	№докум.	Підпис	Дата

окремих перехресть агенти приймають тактичні рішення щодо розподілу фаз, орієнтуючись на рух конкретних автомобілів. Сучасні трирівневі системи залучають локальних, регіональних та центральних агентів, де кожен рівень має власну цільову функцію – від мінімізації індивідуальних затримок до пошуку оптимального глобального циклу роботи мережі. Використання метаевристичних алгоритмів, таких як алгоритм феєрверків, дозволяє таким ієрархічним структурам ефективно балансувати між локальною гнучкістю та загальномережевою продуктивністю, створюючи стійку та адаптивну систему керування міським трафіком.

Таким чином еволюція систем керування трафіком сьогодні спрямована на перехід від жорстких алгоритмів до гнучких оптимізаційних та інтелектуальних моделей, що здатні в реальному часі опрацьовувати величезні масиви даних від підключених транспортних засобів. Оскільки сучасні підходи реалізують ідею переходу від багаторівневої ієрархічної оптимізації до глибокого навчання з підкріпленням вимагають великих обчислювальних ресурсів для мінімізації затримок, тому критичного значення набуває апаратна реалізація таких систем. У цьому контексті впровадження програмованих логічних інтегральних схем (ПЛІС) стає ключовим фактором актуалізації нових систем керування. ПЛІС дозволяють реалізувати паралельну обробку даних на апаратному рівні, що забезпечує наднизьку затримку при прийнятті рішень, яка є недосяжною для традиційних процесорів. Саме використання ПЛІС відкриває шлях до створення високоефективних локальних контролерів, здатних одночасно виконувати складні нейромережеві обчислення та розв'язувати оптимізаційні задачі безпосередньо на перехрестях, що є необхідною умовою для розгортання повноцінної інфраструктури «розумного міста».

1.3 Теоретичні основи керування світлофорами

Керування світлофорами є прикладною задачею теорії управління транспортними потоками, яка поєднує методи математичного моделювання, теорії

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 12
Зм.	Арк.	№докум.	Підпис	Дата		

систем масового обслуговування та алгоритмічної оптимізації. Ефективність роботи регульованого перехрестя залежить від коректності математичної моделі, що описує поведінку транспортного потоку, та від алгоритму, який на основі цієї моделі приймає рішення про перемикання фаз. Розуміння теоретичних основ є необхідною передумовою для обґрунтованого вибору архітектурних рішень при проектуванні апаратно-програмної системи керування.

1.3.1 Математичні моделі перехрестя

Перехрестя як об'єкт керування може бути описане за допомогою кількох класів математичних моделей, кожен з яких відображає різні аспекти транспортного процесу та має різну обчислювальну складність.

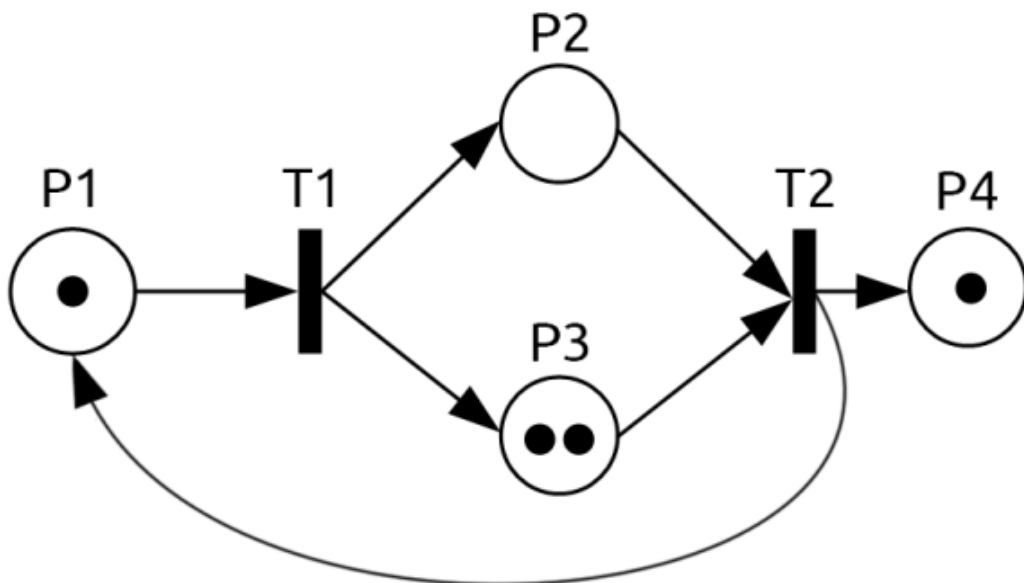
Графова модель перехрестя є найбільш природним способом формального опису топології транспортної мережі. У цій моделі перехрестя представляється як вузол орієнтованого графа $G = (V, E)$, де множина вершин V відповідає зонам накопичення транспортних засобів (смугам очікування), а множина дуг E – можливим напрямкам руху через перехрестя. Кожній дузі приписується пропускна здатність – максимальна кількість транспортних засобів, що можуть проїхати за одиницю часу при зеленому сигналі. Конфліктні напрямки руху, тобто ті, що перетинаються у просторі і не можуть бути активними одночасно, виявляються через аналіз графа конфліктів: два напрямки вважаються конфліктними, якщо відповідні дуги мають спільну точку перетину у просторі перехрестя. Задача планування фаз у термінах графової моделі зводиться до знаходження максимального незалежного розбиття множини дуг на групи неконфліктних напрямків – кожна така група утворює одну фазу циклу.

Теорія черг (або теорія масового обслуговування) надає аналітичний апарат для опису динаміки накопичення і розсіювання черг транспортних засобів на підходах до перехрестя. Класична модель черги $M/D/1$ описує систему з пуассонівським вхідним потоком транспортних засобів (інтенсивність λ автомобілів за секунду) та детермінованим часом обслуговування (час, необхідний

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 13
Зм.	Арк.	№докум.	Підпис	Дата		

одному автомобілю для проїзду через перехрестя). Середня довжина черги у стаціонарному режимі описується формулою Поллачека–Хінчина: $L = \rho^2 / (2(1 - \rho)) + \rho$, де $\rho = \lambda / \mu$ є коефіцієнтом завантаження системи, а μ – інтенсивність обслуговування. Середній час очікування транспортного засобу у черзі виражається через формулу Вебстера: $W = C(1 - \lambda_g / C)^2 / (2(1 - \rho)) + x^2 / (2\lambda(1 - x))$, де C – тривалість циклу, λ_g – частка зеленого часу, $x = \lambda / s$ – ступінь насиченості смуги, s – потік насичення. Ця формула є фундаментальним інструментом для оптимізації тривалості циклу та розподілу зеленого часу між напрямками.

Мережі Петрі забезпечують формальну мову для моделювання паралельних, асинхронних та конкурентних процесів, що робить їх природним засобом опису роботи системи керування світлофорами з кількома перехрестями (рис. 1.3).



Рисунко 1.3 – Модель мережі Петрі

Мережа Петрі являє собою орієнтований двочастковий граф $PN = (P, T, F, M_0)$, де P – множина позицій, T – множина переходів, $F \subseteq (P \times T) \cup (T \times P)$ – відношення інцидентності, $M_0: P \rightarrow \mathbb{N}$ – початкове маркування. У контексті керування світлофором позиції відповідають станам (фазам) кожного перехрестя, переходи – умовам зміни фаз (закінчення таймера, наявність

транспортного засобу на детекторі), а маркування відображає поточний стан усієї мережі. Властивості безпечності мережі (відсутність стану, в якому два конфліктних переходи активні одночасно) можуть бути формально доведені методами аналізу досяжності, що є важливою перевагою перед неформальними специфікаціями. Дерево досяжності мережі Петрі дозволяє перевірити відсутність тупикових станів та живість системи – гарантію того, що кожна фаза циклу рано чи пізно буде активована.

Кінцевий автомат (скінченний автомат, FSM) є найбільш широко застосовуваною моделлю для практичної реалізації логіки керування світлофором в апаратних системах.

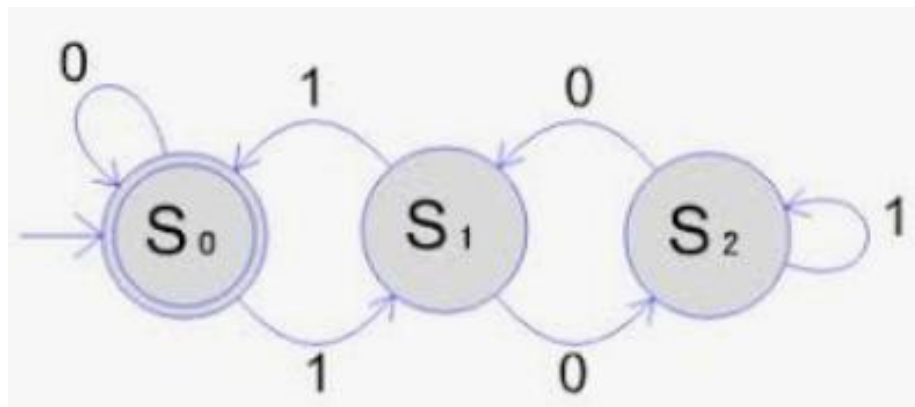


Рисунок 1.4 – Скінченний автомат

Формально FSM визначається як кортеж $M = (Q, \Sigma, \Delta, q_0, F)$, де Q – скінченна множина станів, Σ – вхідний алфавіт, $\Delta: Q \times \Sigma \rightarrow Q$ – функція переходів, $q_0 \in Q$ – початковий стан, $F \subseteq Q$ – множина допустимих станів. У контексті керування перехрестям множина станів Q відповідає можливим комбінаціям сигналів (PHASE_NS_GREEN, PHASE_NS_YELLOW, PHASE_EW_GREEN тощо), вхідний алфавіт Σ включає події закінчення таймерів та команди від системи керування вищого рівня, а функція переходів Δ детерміновано визначає наступний стан для кожної пари (поточний стан, вхідна подія). Перевагою FSM перед мережами Петрі є значно менша обчислювальна складність реалізації в апаратурі: функція переходів безпосередньо транслюється в

комбінаційну логіку на FPGA, а кожен стан відповідає одному набору значень регістра стану. Саме FSM обрано як основну модель для реалізації вузлів керування в даному проекті завдяки її детермінованості, придатності до синтезу в VHDL та прозорій верифікації через функціональну симуляцію.

1.3.2 Алгоритми керування світлофорами

Алгоритми керування світлофорами поділяються на два принципових класи: алгоритми з фіксованим циклом (preimed control) та адаптивні алгоритми, що реагують на поточний стан транспортного потоку в реальному часі.

Алгоритм фіксованого циклу є найдавнішим і найпростішим у реалізації. Тривалість кожної фази та порядок їх чергування задаються наперед і не змінюються в процесі роботи. Оптимальна тривалість циклу C^* розраховується за формулою Вебстера: $C^* = (1.5L + 5) / (1 - Y)$, де L – загальний втрачений час за цикл (сума міжфазових інтервалів), $Y = \sum y_i$ – сума критичних коефіцієнтів завантаження по кожній фазі, $y_i = q_i/s_i$ – відношення потоку q_i до потоку насичення s_i . Зелений час для кожної фази розподіляється пропорційно до відповідного критичного коефіцієнта: $g_i = y_i / (\sum Y_i) \times (C^* - L)$. Алгоритм фіксованого циклу забезпечує передбачувану роботу та просту апаратну реалізацію, але є неоптимальним при нестационарному трафіку, оскільки не враховує реальну завантаженість підходів у момент прийняття рішення. Саме цей алгоритм реалізований у даному проекті як основний режим роботи вузлів, оскільки він відповідає вимогам до детермінованої апаратної логіки і дозволяє повністю специфікувати поведінку системи без сенсорної інфраструктури.

Алгоритм динамічного керування з пріоритетами розширює базовий цикл механізмом реакції на зовнішні події. Найпростішою формою є алгоритм транспортно-залежного керування, при якому тривалість зеленої фази подовжується до тих пір, поки на індукційних петлях виявляються транспортні засоби, але не перевищує максимально допустимого значення g_{max} . Формально умова завершення фази записується як: завершити поточну фазу, якщо $t_{gap} \geq \tau$

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 16
Зм.	Арк.	№докум.	Підпис	Дата		

або $g \geq g_{\max}$, де t_{gap} – час від останнього виявлення автомобіля на детекторі, τ – порогове значення розриву (зазвичай 3-5 секунд). Алгоритм пріоритетного проїзду забезпечує продовження або передчасне завершення поточної фази світлофора при наближенні маршрутного транспортного засобу, що дозволяє скоротити його затримку на перехресті без суттєвого погіршення умов для решти учасників дорожнього руху.

Нечітка логіка застосовується для керування світлофором в умовах неточної або неповної інформації про стан транспортного потоку. Система нечіткого керування складається з трьох компонентів: фазифікатора, бази правил та дефазифікатора (рис. 1.5).

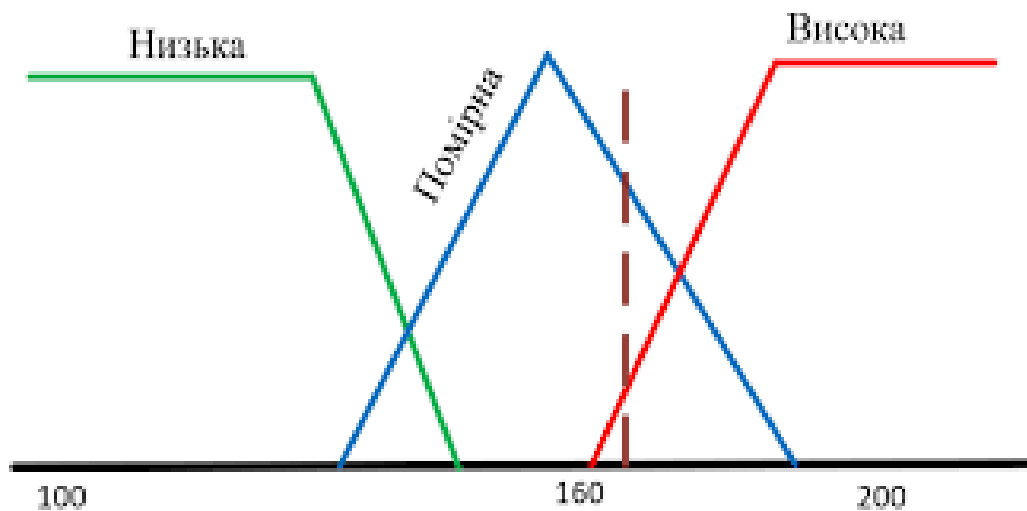


Рисунок 1.5 – Функції належності у системі нечіткого логічного висновку

Вхідними змінними зазвичай є довжина черги на поточній зеленій фазі (short, medium, long) та довжина черги на наступній фазі (short, medium, long), а вихідною змінною – рекомендована зміна тривалості поточної зеленої фази (large decrease, small decrease, no change, small increase, large increase). База правил містить продукційні правила типу «ЯКЩО поточна черга коротка ТА наступна черга довга, ТО значно зменшити тривалість». Нечітке виведення за методом Мамдані дозволяє плавно адаптувати тривалість фаз без різких стрибків, характерних для порогових алгоритмів. Перевагою нечіткого керування є можливість формалізації досвіду

Зм.	Арк.	№докум.	Підпис	Дата

оператора у вигляді лінгвістичних правил, однак складність налаштування бази правил та відсутність гарантій оптимальності є суттєвими недоліками.

Нейромережеві алгоритми керування базуються на навчанні апроксиматора функції оптимального керування за історичними даними або шляхом взаємодії з імітаційною моделлю трафіку. Рекурентні нейронні мережі (LSTM) використовуються для прогнозування майбутніх потоків транспортних засобів на основі поточних і минулих вимірювань, а отриманий прогноз передається оптимізатору тривалості циклу.

Алгоритми навчання з підкріпленням, зокрема глибок Q-мережі, розглядають систему керування як агента, що обирає дії (продовжити поточну фазу або перемкнути) з метою мінімізації кумулятивної нагороди, визначеної як загальний час очікування транспортних засобів.

Попри значний потенціал, нейромережеві алгоритми потребують великих обсягів навчальних даних, обчислювально інтенсивних процесорів для інференсу та складної процедури верифікації безпеки, що робить їх застосування в апаратних системах реального часу на базі FPGA нетривіальним завданням.

Генетичні алгоритми застосовуються переважно для офлайн-оптимізації параметрів фіксованих планів циклів. Особина у генетичному алгоритмі кодує вектор параметрів плану: тривалості циклів та зелених фаз для кожного перехрестя мережі. Функція пристосованості обчислюється шляхом запуску мікроскопічного симулятора трафіку (наприклад, SUMO) з даним набором параметрів і вимірювання середнього часу затримки транспортних засобів.

Оператори схрещування та мутації генерують нові плани, а відбір за значенням функції пристосованості поступово зміщує популяцію до оптимуму. Генетичні алгоритми здатні знаходити глобальний оптимум у багатовимірному просторі параметрів, де аналітичні методи (такі як формула Вебстера) дають лише локальний оптимум для ізольованого перехрестя без урахування взаємодій між суміжними вузлами мережі.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 18
Зм.	Арк.	№докум.	Підпис	Дата		

1.3.3 Критерії ефективності системи керування

Оцінка якості роботи системи керування світлофорами здійснюється за кількома групами критеріїв, що відображають різні аспекти транспортної ефективності та безпеки.

Основним показником ефективності є середній час затримки транспортного засобу \bar{d} , що визначається як різниця між фактичним часом проїзду ділянки та часом проїзду без затримок: $\bar{d} = (1/N) \times \sum_i (t_{iactual} - t_{ifree})$, де N – загальна кількість транспортних засобів, $t_{iactual}$ – фактичний час прибуття i -го автомобіля на виходах із перехрестя, t_{ifree} – час прибуття при вільному русі. Середня затримка безпосередньо характеризує рівень обслуговування перехрестя за шкалою від А ($\bar{d} < 10$ с) до F ($\bar{d} > 80$ с).

Пропускна здатність перехрестя Q визначається як максимальна кількість транспортних засобів, що можуть проїхати через усі підходи за одиницю часу при заданому плані сигналів: $Q = \sum_i s_i \times (g_i/C)$, де s_i – потік насичення i -го підходу, g_i – тривалість зеленої фази, C – тривалість циклу. Ступінь насиченості $x = q/Q$, де q – фактичний потік, є безрозмірним показником завантаженості перехрестя, тобто при $x < 0.85$ рух стабільний, при $x > 1.0$ утворюються незгасаючі черги. Пропускна здатність є особливо важливим критерієм для координованих систем керування, оскільки обмежуюча ланка в мережі перехресть визначає пропускну здатність усього транспортного коридору.

Кількість зупинок на один транспортний засіб є критерієм, що корелює як з витратами палива, так і з рівнем викидів шкідливих речовин. Кожна зупинка та подальше прискорення потребує додаткової енергії, яка у разі автомобілів з двигунами внутрішнього згоряння повністю розсіюється у вигляді тепла при гальмуванні. Частка транспортних засобів, що зупиняються, описується формулою: $P_{stop} = 0.9 \times (x \times C) / (2 \times g_{eff})$, де g_{eff} – ефективний зелений час. Мінімізація кількості зупинок є основним критерієм при проектуванні режиму «зеленої хвилі» для координованого керування суміжними перехрестями.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 19
Зм.	Арк.	№докум.	Підпис	Дата		

Паливна ефективність як інтегральний критерій оцінюється через модель витрат палива VSP: $VSP = v \times (a + g \times \sin\theta + \psi) + \varepsilon \times v^3$, де v – швидкість, a – прискорення, g – прискорення вільного падіння, θ – кут нахилу дороги, ψ – коефіцієнт опору кочення, ε – аеродинамічний коефіцієнт. Витрати палива пропорційні до позитивних значень VSP, тому зменшення кількості зупинок та розгонів (що забезпечується оптимізованим керуванням) безпосередньо скорочує споживання пального та викиди CO₂. Дослідження показують, що адаптивне керування світлофорами порівняно з фіксованим циклом здатне скоротити загальне споживання палива на магістральних вулицях на 10–15% при незмінній геометрії вулично-дорожньої мережі.

Для розподілених систем, що керують кількома перехрестями одночасно, важливим додатковим критерієм є синхронність фаз між вузлами, тобто показник того, наскільки точно фактичні моменти перемикання відповідають розрахунковому плану координації. У апаратних реалізаціях на FPGA синхронність визначається точністю дільника частоти та надійністю механізму ширококомовної синхронізації, а не програмними затримками, характерними для мікроконтролерних реалізацій. Саме детермінованість апаратного виконання є головним аргументом на користь використання FPGA як платформи для вузлів керування у розподіленій системі.

1.4 Висновки. Постановка задачі

Таким чином проведений теоретичний огляд показав Розробка розподіленої спеціалізованої комп'ютерної системи керування дорожнім рухом на базі ПЛІС та мікроконтролерів ESP32 є важливим та доцільним завданням, що відповідає сучасним викликам концепції «Розумного міста» та розвитку інтелектуальних транспортних систем. Поєднання апаратної гнучкості ПЛІС із комунікаційними можливостями ESP32 дозволяє створити ефективне рішення для паралельної обробки великих масивів даних від сенсорів та підключених транспортних засобів у режимі реального часу з наднизькою затримкою. Це забезпечує перехід від

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 20
Зм.	Арк.	№докум.	Підпис	Дата		

традиційних циклічних алгоритмів до адаптивних стратегій керування, що суттєво зможе оптимізувати пропускну здатність дорожньої мережі, мінімізувати час очікування і, як наслідок, знизити рівень викидів шкідливих речовин, що є важливим з точки зору збереження екологічного стану планети.

Таким чином для вирішення цього завдання слід виконати:

- 1) визначити технічні та функціональні вимоги до системи керування світлофорами на базі FPGA та ESP32;
- 2) спроектувати структуру розподіленої системи, визначивши ролі FPGA та ESP32 у керуванні трафіком;
- 3) розробити протокол обміну інформацією та визначити формат команд для взаємодії між вузлами;
- 4) підібрати конкретні моделі ПЛІС, мікроконтролерів та периферійних компонентів;
- 5) розробити цифрову логіку системи керування та сконфігурувати її для роботи на FPGA;
- б) протестувати роботу машини станів окремого світлофора та перевірити функціонування всієї розподіленої мережі в режимі симуляції.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						21
Зм.	Арк.	№докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ РОЗПОДІЛЕНОЇ СПЕЦІАЛІЗОВАНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ КЕРУВАННЯ СВІТЛОФОРАМИ НА БАЗІ FPGA ТА ESP32

2.1 Постановка вимог до розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

Проектована розподілена спеціалізована комп'ютерна система призначена для автоматизованого керування дорожніми світлофорами на двох суміжних перехрестях у складі єдиної розподіленої мережі. Основна вимога до системи полягає у забезпеченні безпечного та ефективного регулювання транспортних потоків шляхом точного дотримання заданих часових параметрів фаз світлофорного циклу та гарантованої відсутності конфліктних станів, за яких зелений сигнал одночасно подається для взаємно перпендикулярних напрямків руху.

Досліджувана модель перехрестя представляє собою класичний хрестоподібний перетин двох автомобільних доріг з чотирма напрямками руху (рис. 2.1). Регулювання транспортних потоків здійснюється за допомогою чотирьох світлофорів, кожен з яких має три стандартні сигнали керування: червоний, жовтий та зелений, що забезпечують поетапне та безпечно пропускання транспортних засобів через перехрестя.

З функціональної точки зору система має реалізовувати повний чотирифазний цикл керування перехрестям, що включає фазу зеленого сигналу для напрямку Північ–Південь, фазу жовтого попереджувального сигналу для того ж напрямку, захисну паузу з одночасним червоним сигналом для всіх напрямків, фазу зеленого сигналу для напрямку Схід–Захід, відповідну жовту фазу та повторну захисну паузу перед початком нового циклу. Тривалість кожної з перелічених фаз має бути незалежно програмованою у діапазоні від 1 до 255 секунд без необхідності перепрограмування апаратної логіки.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 22
Зм.	Арк.	№докум.	Підпис	Дата		

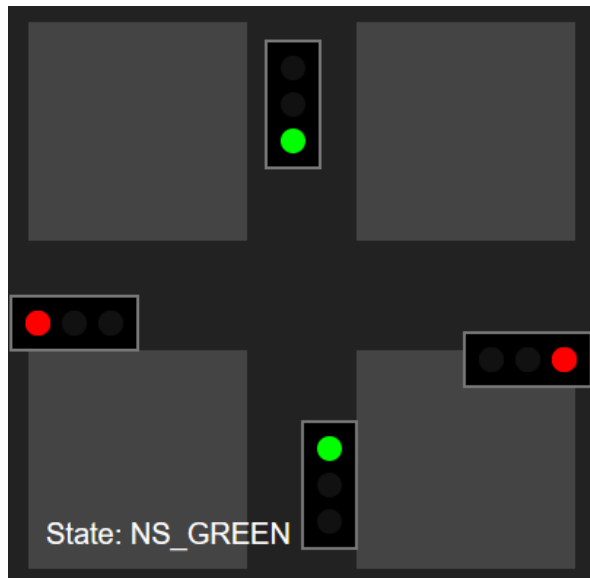


Рисунок 2.1 – Досліджувана модель перехрестя

Проектована система має підтримувати чотири режими роботи, а саме:

- нормальний режим;
- нічний режим;
- аварійний режим;
- ручний режим.

Нічний режим передбачає переведення світлофорів обох перехресть у стан жовтого миготіння з частотою один герц, що відповідає вимогам дорожнього регулювання в умовах зниженої інтенсивності руху. Аварійний режим забезпечує негайну фіксацію червоного сигналу на всіх напрямках усіх керованих перехресть для забезпечення безперешкодного проїзду пріоритетного транспортного засобу. Режим ручного керування дозволяє оператору системи зафіксувати будь-яку довільну фазу на будь-якому вузлі без автоматичного перемикання.

До системи також висувається вимога щодо централізованого керування усіма вузлами від єдиного головного контролера. Головний контролер повинен мати можливість у будь-який момент часу змінювати конфігураційні параметри кожного вузла незалежно, не перериваючи роботи інших вузлів мережі. Передача команд від головного контролера до вузлів має здійснюватися по спільній послідовній шині з підтримкою індивідуальної адресації окремих вузлів та широкомовної адресації для одночасного впливу на всі вузли системи.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						23
Зм.	Арк.	№докум.	Підпис	Дата		

Принципово важливою вимогою є підтримка координованого керування суміжними перехрестями з реалізацією режиму «зеленої хвилі». Таким чином система має забезпечувати можливість задання часового зміщення між початком циклів різних вузлів з точністю до однієї секунди, що дозволяє синхронізувати зелені фази суміжних перехресть відповідно до швидкості транспортного потоку між ними. Механізм синхронізації повинен гарантувати одночасний старт усіх вузлів після отримання ширококомовного сигналу синхронізації від головного контролера незалежно від поточного стану автоматів.

З апаратної точки зору логіка керування кожним вузлом має бути реалізована на програмованій логічній інтегральній схемі FPGA, що забезпечує детерміновані часові характеристики виконання операцій та незалежність від операційної системи. Роль головного контролера та реалізація комунікаційного рівня покладається на мікроконтролер ESP32, який завдяки наявності апаратного UART-контролера здатен забезпечити надійну передачу конфігураційних даних. Інтерфейс між головним контролером та вузлами FPGA має базуватися на стандартному протоколі UART, що унеможливорює залежність від пропрієтарних протоколів і спрощує масштабування системи.

Система має задовольняти вимогам щодо надійності та відновлюваності: при отриманні сигналу скидання від головного контролера кожен вузол повинен негайно переходити у безпечний стан із червоними сигналами на всіх напрямках, після чого відновлювати нормальну роботу у синхронізованому режимі. Перехід між режимами роботи має відбуватися атомарно тобто без проміжних невизначених станів, які могли б призвести до некоректного відображення сигналів на фізичних світлофорах.

Таким чином, висунуті вимоги є обґрунтованими та такими, що відповідають принципам організації дорожнього руху на стандартних регульованих перехрестях. Разом і з тим вони відображають типову логіку роботи реального чотиристороннього перехрестя зі світлофорним регулюванням та можуть бути використані для моделювання реальних умов функціонування транспортної інфраструктури.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 24
Зм.	Арк.	№докум.	Підпис	Дата		

2.2 Архітектура розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

Архітектура системи побудована за дворівневим ієрархічним принципом. Верхній рівень представлений головним контролером на базі мікроконтролера ESP32, який виконує функції централізованого диспетчера і є єдиним джерелом конфігураційних команд у системі. Нижній рівень утворений двома ідентичними вузлами на базі FPGA, кожен з яких відповідає за автономне керування одним перехрестям. Зв'язок між рівнями реалізований через спільну послідовну шину UART, до якої підключені всі учасники мережі за топологією «один передавач – багато приймачів» (рис. 2.1).

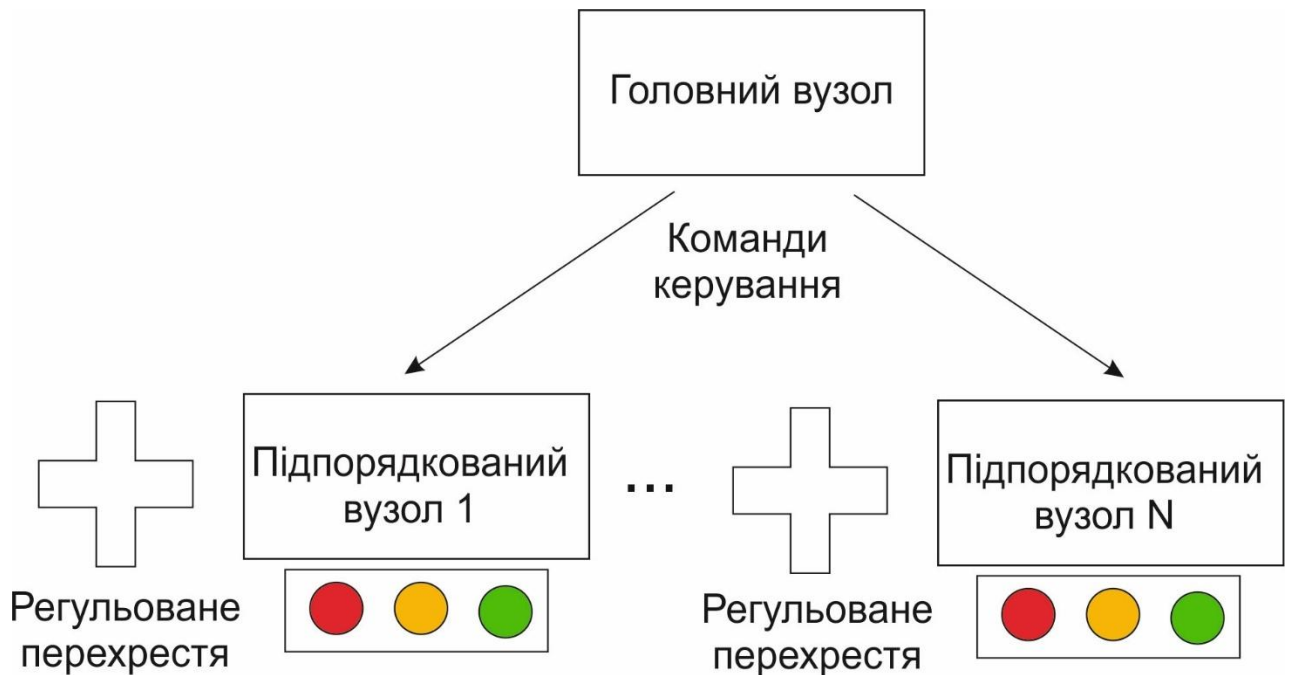


Рисунок 2.1 – Узагальнена архітектура розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

Головний контролер ESP32 виступає виключно у ролі головного пристрою та не виконує жодних функцій безпосереднього керування сигналами світлофорів. Його єдиною відповідальністю є формування та передача конфігураційних пакетів на адресу конкретних вузлів або ширококомовно на всі вузли одночасно. Входами головного контролера є параметри конфігурації вузлів (тривалості фаз, зміщення,

режим роботи) та керуючі строби (`send_config`, `send_sync`, `send_reset`), що ініціюють відповідні транзакції передачі. Єдиним виходом є лінія передачі UART (`uart_txd`), яка підключається паралельно до входів `uart_rxd` обох вузлів програмованої користувачем вентиляційної матриці.

Структура вузла розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 подано на рис. 2.3. Архітектурно кожен вузол керування світлофорами складається із чотирьох модулів, що з'єднані послідовним потоком передачі даних. Перший модуль – дільник частоти `clk_prescaler` отримує на вхід тактовий сигнал частотою 50 МГц та сигнал скидання `rst_n`, а на виході формує єдиний сигнал `tick_1hz`, що визначає послідовність коротких імпульсів з частотою один герц (рис. 2.2). Цей сигнал є спільною часовою базою для всіх часозалежних операцій вузла і подається на вхід модуля скінченної машини станів. Параметри дільника (`CLK_FREQ` та `OUT_FREQ`) є узагальненими, що дозволяє адаптувати модуль до різних тактових частот без зміни архітектури.



Рисунок 2.2 – Схема формування підрахунку модулем скінченної машини станів

Другий модуль UART приймача `uart_rx` є єдиною точкою входу зовнішніх команд у вузол. На його вхід надходить сигнал `uart_rxd` із спільної шини, а також тактовий сигнал та скидання. Модуль здійснює побітове декодування послідовного потоку даних відповідно до протоколу UART та накопичує прийняті байти у внутрішньому тристадійному буфері. Після успішного прийому трьох байтів модуль виставляє на один такт імпульс `pkt_valid` та виводить вміст байтів через

шини pkt_byte0, pkt_byte1 та pkt_byte2. Ці три виходи разом з імпульсом pkt_valid утворюють інтерфейс між модулем прийому та модулем декодування команд.

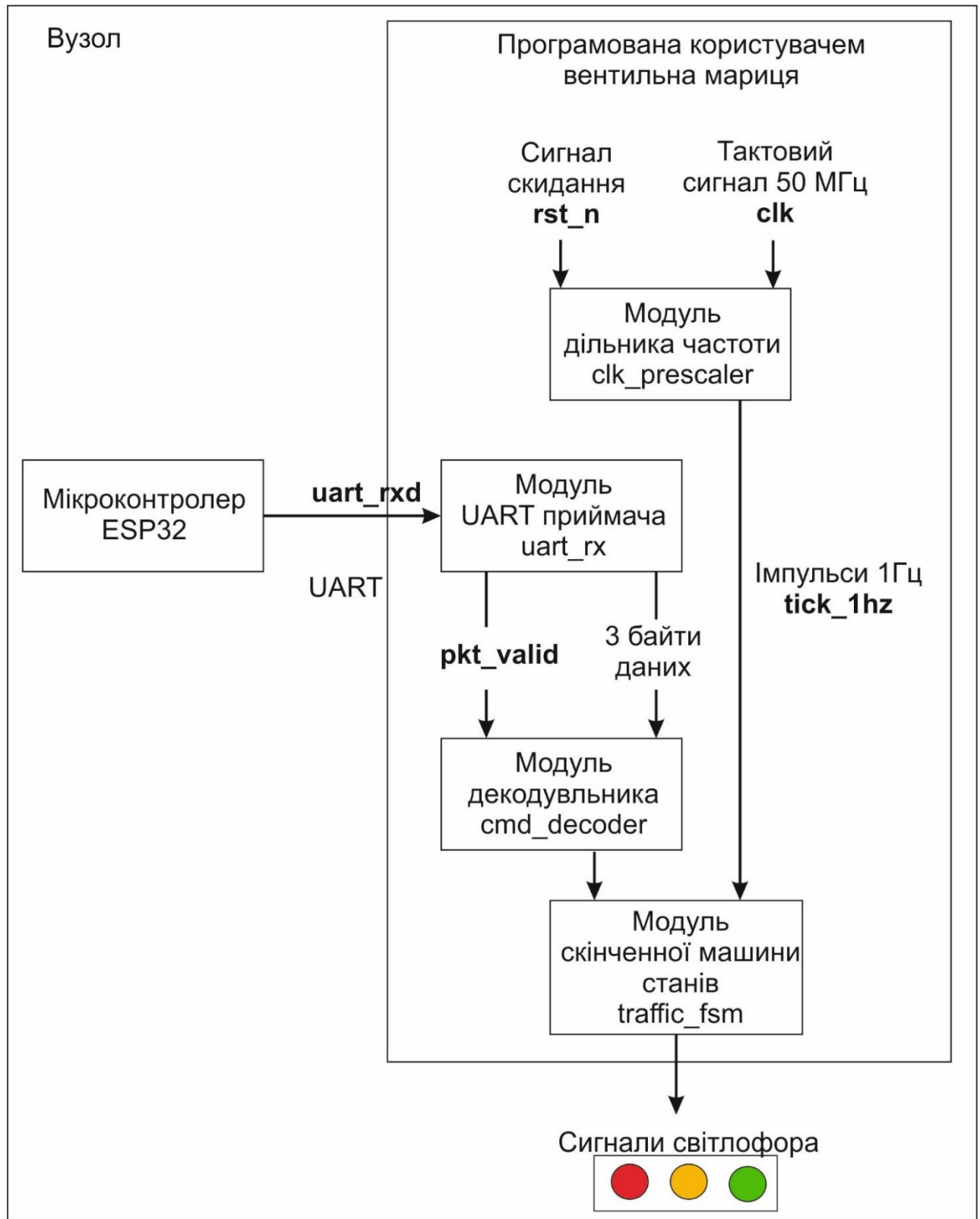


Рисунок 2.3 – Структура вузла розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32

Зм.	Арк.	№докум.	Підпис	Дата

Третій модуль модуль декоування `cmd_decoder` є центральним елементом комунікаційного рівня вузла. На його входи надходять чотири сигнали від `uart_rx` (`pkt_valid` та три байти пакету), а також тактовий сигнал і скидання. Модуль аналізує поле адресата у старших двох бітах першого байта: якщо адреса збігається із власним ідентифікатором `NODE_ID` або містить ширококомовне значення `00`, пакет обробляється; в іншому випадку ігнорується. Поле коду команди у молодших чотирьох бітах першого байта визначає, який саме параметр конфігурації оновлюється: тривалість зеленої NS-фази, зеленої EW-фази, жовтої фази, захисного інтервалу, зміщення або код ручної фази. Усі конфігураційні параметри зберігаються у внутрішніх регістрах модуля і виводяться назовні через відповідні вихідні порти: `cfg_mode`, `cfg_green_ns`, `cfg_green_ew`, `cfg_yellow`, `cfg_allred`, `cfg_offset` та `cfg_manual_phase`. Окремими виходами є стробуючі сигнали `do_sync` та `do_reset`, що формуються при отриманні відповідних ширококомовних команд та подаються безпосередньо на керуючі входи модуля машини станів.

Четвертий модуль `traffic_fsm` є модулем машини станів та представляє собою апаратне ядро вузла й реалізує скінченний автомат керування перехрестям. На його входи надходять: тактовий сигнал `clk`, сигнал скидання `rst_n`, секундний тик `tick_1hz` від дільника, весь набір конфігураційних сигналів від `cmd_decoder`, а також стробуючі сигнали `do_sync` та `do_reset`. Автомат функціонує на основі лічильника фаз, який інкрементується при кожному імпульсі `tick_1hz` і порівнюється з поточним цільовим значенням тривалості фази. При досягненні граничного значення відбувається перехід у наступний стан відповідно до заданої послідовності циклу. Виходами модуля є двовимірні тривітові вектори `ns_light` та `ew_light`, де кожен біт відповідає одному кольору сигналу – червоному, жовтому або зеленому. Додатково модуль виводить діагностичні сигнали: `current_phase` (поточний стан автомата у вигляді перелічуваного типу), `phase_timer` (поточне значення лічильника фази), `cycle_count` (кількість завершених повних циклів) та `sync_ok` (прапор успішної синхронізації).

Структурний модуль `traffic_node` об'єднує всі чотири описані компоненти в єдиний ієрархічний блок, що представляє один вузол перехрестя. Зовнішній

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 28
Зм.	Арк.	№докум.	Підпис	Дата		

інтерфейс вузла включає три вхідні сигнали (clk, rst_n, uart_rxd) та дванадцять вихідних сигналів: шість бітів стану світлофорів (ns_red, ns_yellow, ns_green, ew_red, ew_yellow, ew_green) та шість діагностичних сигналів (dbg_phase, dbg_timer, dbg_cycle, dbg_mode, dbg_sync_ok). Ідентифікатор вузла NODE_ID передається як узагальнений параметр під час інстанціювання, що дозволяє використовувати один і той самий RTL-код для обох вузлів системи з різними адресами без будь-яких змін у вихідному коді.

Верхній структурний модуль traffic_system_top інстанціює екземпляри traffic_node та з'єднує їх спільним вхідним портом master_uart_rx. Для даної роботи створюється два вузла, таким чином перший екземпляр отримує значення NODE_ID = "01", другий – NODE_ID = "10". Вихідні сигнали кожного вузла виводяться на окремі порти верхнього рівня з префіксами n1_ та n2_, формуючи повний зовнішній інтерфейс системи. На рівні верхнього модуля відсутня будь-яка додаткова логіка: вся функціональність зосереджена у модулях нижніх рівнів ієрархії, а верхній рівень виконує виключно комутаційну роль.

Потік даних у системі має чітко виражений односпрямований характер на рівні конфігурування: команди рухаються від головного мікроконтролера ESP32 через UART-шину до uart_rx кожного вузла, далі через cmd_decoder до регістрів конфігурації, і нарешті до входів модуля скінченої машини станів, яка формує вихідні сигнали світлофорів. Зворотній потік, тобто передача статусних даних від вузлів до головного контролера, реалізується через діагностичні виходи dbg_* і в рамках даного проекту призначений для підключення до логічного аналізатора або осцилографа, а не для автоматичної обробки контролером. Така асиметрична архітектура суттєво спрощує протокол взаємодії та виключає можливість конфліктів на спільній шині передачі.

Часова координація між вузлами забезпечується на рівні протоколу, а не апаратних зв'язків між вузлами. Кожен вузол має незалежний дільник частоти та незалежний лічильник фази, що виключає апаратні залежності між вузлами і дозволяє додавати нові вузли до системи без модифікації існуючих. Синхронізація досягається через ширококомовну команду CMD_SYNC, після отримання якої кожен

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 29
Зм.	Арк.	№докум.	Підпис	Дата		

нерозривного блоку передачі. Фіксована довжина пакету спрощує логіку синхронізації на стороні приймача: після детектування першого байта приймач завжди очікує рівно два додаткові байти, не потребуючи аналізу вмісту для визначення меж пакету.

Перший байт пакету є керуючим і містить три поля. Старші два біти (біти 7–6) утворюють поле `NODE_ID` – ідентифікатор вузла-адресата. Біти 5–4 утворюють поле `MODE` – код режиму роботи, який має бути встановлений на вузлі. Молодші чотири біти (біти 3–0) містять поле `CMD` – код конкретної команди, що визначає дію, яку вузол має виконати. Структуру першого байта наведено у таблиці 2.2.

Таблиця 2.2 – Структура керуючого байта (Байт 0)

Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
<code>NODE_ID[1]</code>	<code>NODE_ID[0]</code>	<code>MODE[1]</code>	<code>MODE[0]</code>	<code>CMD[3]</code>	<code>CMD[2]</code>	<code>CMD[1]</code>	<code>CMD[0]</code>
Адреса вузла (2 біти)		Режим (2 біти)		Код команди (4 біти)			

Поле `NODE_ID` визначає адресата команди. Значення 00 є широкомовною адресою і означає, що команда адресована одночасно всім вузлам мережі незалежно від їх індивідуальних ідентифікаторів. Значення 01 адресує перший вузол (перехрестя 1), значення 10 – другий вузол (перехрестя 2). Значення 11 зарезервовано для можливого розширення системи. Кодування поля `NODE_ID` наведено у таблиці 2.3.

Таблиця 2.3 – Кодування поля `NODE_ID`

Код <code>NODE_ID</code>	Призначення
00	Широкомовна адресація (всі вузли)
01	Вузол 1 (перехрестя 1)
10	Вузол 2 (перехрестя 2)
11	Зарезервовано

Поле MODE передається разом із кожною командою і дозволяє змінити режим роботи вузла в рамках тієї ж транзакції, що й конфігураційний параметр, скорочуючи загальну кількість пакетів при реконфігурації. Якщо зміна режиму не є метою поточної команди, поле MODE заповнюється поточним значенням або нулем без змін. Кодування поля MODE наведено у таблиці 2.4.

Таблиця 2.4 – Кодування поля MODE

Код MODE	Режим роботи
00	MODE_NORMAL – нормальний цикл
01	MODE_NIGHT – жовте миготіння
10	MODE_EMERGENCY – примусовий червоний
11	MODE_MANUAL – ручна фаза

Другий та третій байти пакету є параметричними і їх інтерпретація визначається значенням поля CMD. Для більшості команд другий байт містить числовий параметр у діапазоні 0–255, що безпосередньо відповідає тривалості фази у секундах або величині зміщення. Третій байт використовується лише командою SET_ALL для одночасного задання тривалості двох зелених фаз; в усіх інших командах третій байт ігнорується і заповнюється нулем. Повний перелік підтримуваних команд наведено у таблиці 2.5.

Таблиця 2.5 – Кодування поля CMD та інтерпретація параметрів

Код CMD	Байт 1	Байт 2	Опис команди
0x0	–	–	NOP – порожня команда, ігнорується
0x1	Тривалість (с)	–	SET_NS_DUR – тривалість зеленої фази NS

Продовження таблиці 2.5

0x2	Тривалість (с)	–	SET_EW_DUR – тривалість зеленої фази EW
0x3	Тривалість (с)	–	SET_YELLOW – тривалість жовтої фази
0x4	Тривалість (с)	–	SET_ALLRED – тривалість захисного інтервалу
0x5	Зміщення (с)	–	SET_OFFSET – затримка старту циклу
0x6	Код фази (0–7)	–	SET_MANUAL_PHASE – фіксація конкретної фази
0x7	–	–	SYNC – старт/синхронізація циклу (зазвичай broadcast)
0x8	–	–	RESET – скидання FSM у стан ALL_RED
0x9	NS тривалість	EW тривалість	SET_ALL – одночасне задання обох зелених фаз

Процедура конфігурування системи після увімкнення або при зміні параметрів руху складається з фіксованої послідовності транзакцій. Спочатку головний контролер надсилає широкомовну команду RESET (NODE_ID = 00, CMD = 0x8), що переводить усі вузли у безпечний стан з червоними сигналами на всіх напрямках. Далі для кожного вузла послідовно надсилаються команди SET_NS_DUR, SET_EW_DUR, SET_YELLOW, SET_ALLRED та SET_OFFSET з індивідуальними значеннями параметрів. Завершальним кроком є широкомовна команда SYNC (NODE_ID = 00, CMD = 0x7), після отримання якої всі вузли одночасно починають відлік зміщення та запускають нормальний цикл.

Між окремими байтами одного пакету витримується мінімальний міжбайтовий інтервал тривалістю чверть бітового інтервалу, що відповідає

приблизно 2 мкс при швидкості 115 200 бод. Між послідовними пакетами витримується міжпакетний інтервал тривалістю не менше двох бітових інтервалів. Ці значення обрані достатніми для гарантованого завершення обробки поточного пакету модулем `cmd_decoder` до початку прийому наступного, але мінімальними для скорочення загального часу конфігурування системи. При передачі повного набору з одинадцяти пакетів (по п'ять для кожного вузла плюс один SYNC) загальний час конфігурування не перевищує 15 мс при швидкості 115 200 бод, що є незначним відносно мінімальної тривалості фази у одну секунду.

Для наочного розуміння кодування пакетів у таблиці 2.6 наведено приклади реальних команд з поясненням кожного байта у шістнадцятковому форматі.

Таблиця 2.6 – Приклади пакетів команд

Байт 0	Байт 1	Байт 2	Пояснення
0x41	0x1E	0x00	Вузол 1, режим NORMAL, SET_NS_DUR = 30 секунд
0x42	0x04	0x00	Вузол 1, режим NORMAL, SET_EW_DUR = 4 секунди
0x55	0x08	0x00	Вузол 1, режим NORMAL, SET_OFFSET = 8 секунд
0x97	0x05	0x04	Вузол 2, режим NORMAL, SET_ALL: NS=5с, EW=4с
0x07	0x00	0x00	Broadcast SYNC – старт усіх вузлів одночасно
0x08	0x00	0x00	Broadcast RESET – всі вузли у стан ALL_RED
0x90	0x00	0x00	Вузол 2, MODE_NIGHT (NOP) – перехід у нічний режим
0xA0	0x00	0x00	Broadcast, MODE_EMERGENCY – примусовий червоний

Відсутність механізму підтвердження прийому є свідомим проектним рішенням, що обумовлене детерміністичним характером апаратного середовища. У реальній системі всі вузли фізично підключені до одного джерела сигналу, і ймовірність втрати пакету через апаратний збій є незначною. У разі необхідності

підвищення надійності протокол може бути розширений введенням контрольної суми у четвертий байт пакету та організацією зворотного каналу від вузлів до головного контролера без зміни існуючого формату команд – шляхом збільшення довжини пакету з трьох до чотирьох байтів та додавання окремої UART-лінії для відповідей.

2.4 Аналіз та вибір апаратних засобів

Вибір апаратної платформи для реалізації розподіленої системи керування світлофорами визначається сукупністю технічних вимог до кожного рівня ієрархії системи. На рівні вузлів перехресть ключовими критеріями є детермінованість часових характеристик, паралельність виконання апаратної логіки та наявність достатньої кількості виводів для підключення сигнальних ліній. На рівні головного контролера пріоритетними є гнучкість програмування, наявність апаратного інтерфейсу UART та можливість подальшого розширення системи засобами бездротового зв'язку. Окремим критерієм є доступність безкоштовних інструментів розробки та симуляції, оскільки проект виконується в академічному середовищі.

Для реалізації апаратної логіки вузлів перехресть розглядалися програмовані логічні інтегральні схеми трьох платформ: Intel Cyclone V (кристал 5CSEMA5F31C6), Xilinx Artix-7 (плата Basys 3) та Intel MAX 10 (плата DE10-Lite). Порівняльний аналіз цих платформ наведено в таблиці 2.1.

Кристал Intel Cyclone V 5CSEMA5F31C6 обраний як апаратна платформа вузлів системи з кількох вагомих причин. По-перше, він містить 85 000 логічних елементів, що з великим запасом перекриває потреби проекту і залишає ресурс для масштабування системи до більшої кількості вузлів або додавання нових функцій без заміни кристала. По-друге, сімейство Cyclone V у варіанті SX містить вбудоване двоядерне ядро ARM Cortex-A9 (HPS – Hard Processor System), яке в рамках даного проекту не використовується, але є суттєвою перевагою для можливого розширення системи на рівні операційної системи Linux без зовнішнього мікроконтролера. По-третє, кристал повністю підтримується

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 35
Зм.	Арк.	№докум.	Підпис	Дата		

середовищем Quartus Prime у версіях Lite та Pro, що забезпечує сумісність з безкоштовним інструментарієм розробки та інтегрованим логічним аналізатором SignalTap II. Тактова частота вбудованого генератора становить 50 МГц, що є стандартною базовою частотою для більшості навчальних та промислових FPGA-проектів і достатньою для реалізації всіх часових алгоритмів системи. Слід зазначити, що кристал не має вбудованої Flash-пам'яті для зберігання конфігурації і використовує зовнішню мікросхему EPCQ, проте це є стандартною практикою для більшості FPGA сімейства Cyclone і не становить технічної складності при розгортанні системи (таблиця 2.7).

Таблиця 2.7 – Порівняльний аналіз FPGA-платформ

Параметр	Intel Cyclone V 5CSEMA5F31C6	Xilinx Artix-7 (Basys 3)	Intel MAX 10 (DE10-Lite)
Логічних елементів	85 000	33 280	50 000
Тактова частота	50 МГц	100 МГц	50 МГц
Вбудована Flash	Ні (EPCQ зовн.)	Ні	Так
Блоків DSP	87	90	144
GPIO виводів	72	48	72
Підтримка Quartus	Так (Lite/Pro)	Ні (Vivado)	Так (Lite)
SignalTap II	Так	Так (ILA)	Так
Hard ARM ядро	Так (Cortex-A9)	Ні	Ні

Платформа Xilinx Artix-7 на Basys 3 має вищу тактову частоту та розвинену екосистему IP-ядер, однак вимагає використання середовища Vivado, яке є несумісним із потоком верифікації ModelSim-Altera та файлами проекту формату QSF/QPF. Платформа Intel MAX 10 є технічно сумісною з Quartus і має перевагу у вигляді вбудованої Flash-пам'яті, однак поступається Cyclone V за кількістю логічних елементів та не має вбудованого процесорного ядра (рис.2.4).

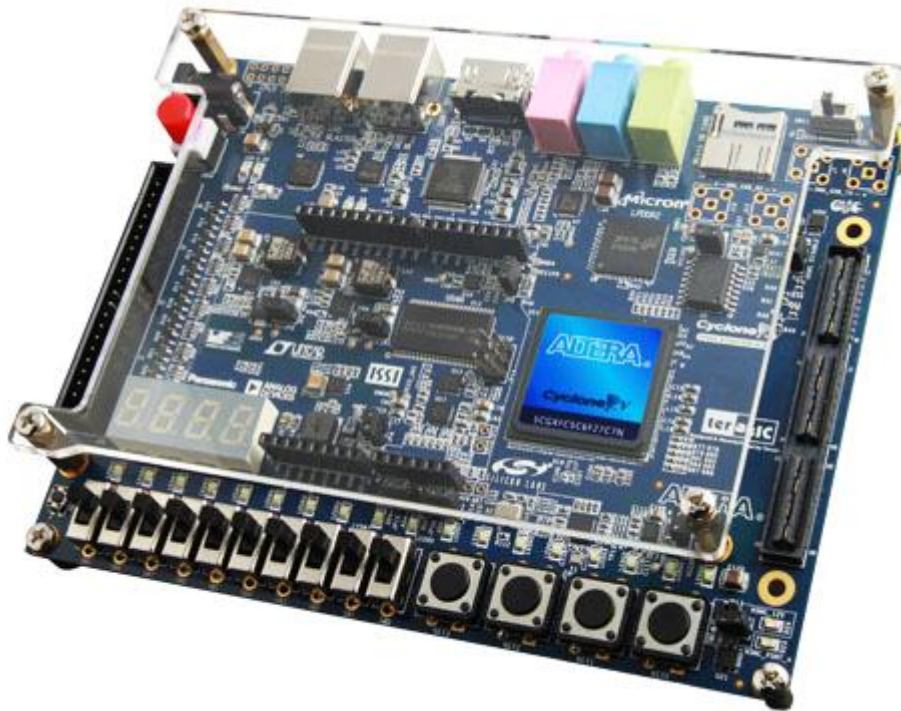


Рисунок 2.4 – ПЛІС Altera Cyclone V

Для реалізації головного контролера розподіленої системи розглядалися три мікроконтролерні платформи: ESP32, STM32F103 та Arduino Mega. Порівняльний аналіз наведено в таблиці 2.8.

Таблиця 2.8 – Порівняльний аналіз мікроконтролерних платформ

Параметр	ESP32	STM32F103	Arduino Mega
Тактова частота	240 МГц	72 МГц	16 МГц
Ядра	2 (Xtensa LX6)	1 (Cortex-M3)	1 (AVR)
Апаратних UART	3	3	4
Wi-Fi / Bluetooth	Так / Так	Ні / Ні	Ні / Ні
Рівень сигналів GPIO	3.3 В	3.3 В	5 В
FreeRTOS	Вбудована	Так (порт)	Обмежено
Flash-пам'ять	4 МБ	64–512 КБ	256 КБ

Мікроконтролер ESP32 виробництва Espressif Systems обраний як головний контролер системи завдяки оптимальному поєднанню обчислювальної потужності, комунікаційних можливостей та вартості (рис. 2.5). Двоядерний процесор Xtensa LX6 з тактовою частотою 240 МГц забезпечує достатню продуктивність для одночасного обслуговування черги UART-пакетів та обробки команд оператора. Наявність трьох апаратних UART-контролерів дозволяє виділити окремий порт для шини керування вузлами FPGA без конфліктів з відлагоджувальним інтерфейсом. Принциповою перевагою ESP32 є наявність вбудованих модулів Wi-Fi 802.11 b/g/n та Bluetooth 4.2, що забезпечує можливість бездротового дистанційного керування системою у подальших розширеннях проекту без будь-яких апаратних змін.

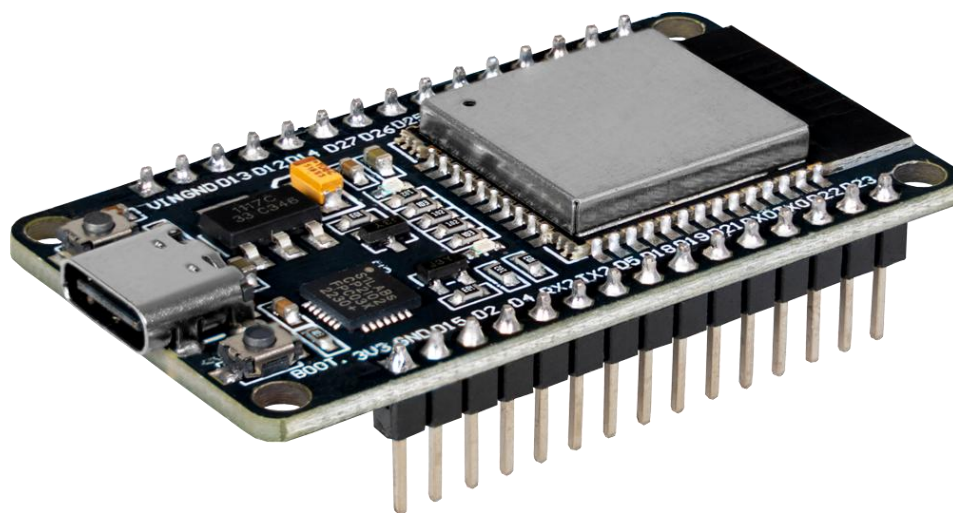


Рисунок 2.5 – Мікроконтролер ESP32

Рівень логічних сигналів ESP32 становить 3.3 В, що збігається зі стандартом сигналів GPIO кристала Cyclone V (3.3-V LVTTL), і тому підключення між пристроями виконується прямим з'єднанням без необхідності узгоджувачих схем перетворення рівнів. Це суттєво спрощує апаратну частину проекту порівняно зі схемою на базі Arduino Mega, яка використовує рівні 5 В і потребувала б дільника напруги або спеціалізованого перетворювача рівнів на кожній лінії зв'язку з FPGA.

Мікроконтролер STM32F103, незважаючи на достатню тактову частоту та наявність апаратних UART, позбавлений вбудованих засобів бездротового зв'язку, що обмежує можливості розширення системи та вимагає додаткових модулів.

Arduino Mega має найнижчу тактову частоту серед розглянутих варіантів (16 МГц), обмежений обсяг Flash-пам'яті (256 КБ) та відсутність підтримки FreeRTOS, що є суттєвим недоліком при необхідності одночасного обслуговування кількох комунікаційних потоків.

Вибір протоколу зв'язку між головним контролером та вузлами FPGA обґрунтований такими міркуваннями. Протокол UART є апаратно підтримуваним як на ESP32, так і реалізованим засобами VHDL на кристалі Cyclone V без залучення складних IP-ядер. Альтернативний протокол SPI забезпечує вищу швидкість передачі, але потребує окремої лінії вибірки для кожного веденого пристрою, що при масштабуванні системи до більшої кількості вузлів призводить до зростання кількості з'єднань. Протокол I2C підтримує адресацію на рівні протоколу, але його максимальна швидкість 400 кбод є надлишково повільною для систем з великою кількістю конфігураційних параметрів. Протокол UART зі швидкістю 115 200 бод дозволяє передати повний конфігураційний пакет для одного вузла менш ніж за 300 мікросекунд, що є цілком достатнім для систем із секундними часовими циклами і при цьому забезпечує просту апаратну реалізацію шини «один передавач – багато приймачів».

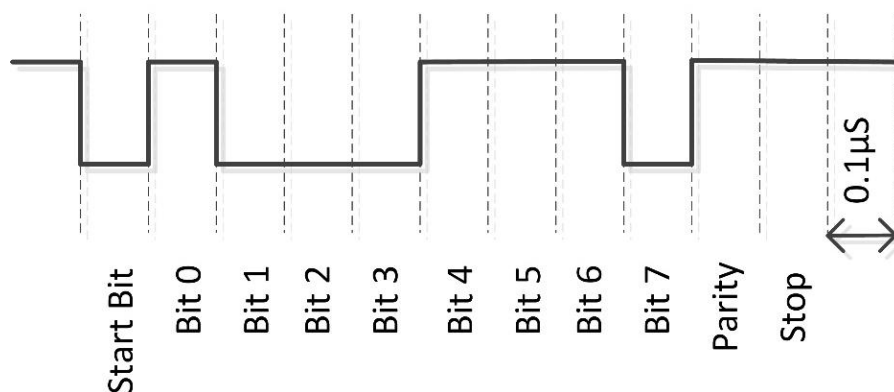


Рисунок 2.6 – Приклад діаграми кодування сигналів за протоколом UART

Таким чином, обрана апаратна конфігурація – Intel Cyclone V 5CSEMA5F31C6 у ролі програмованої логіки вузлів перехрестя та ESP32 у ролі головного контролера з UART-інтерфейсом – оптимально відповідає вимогам проекту за критеріями обчислювальної потужності, сумісності рівнів сигналів,

доступності інструментів розробки та потенціалу для подальшого розширення функціональності системи.

2.5 Висновки

В рамках даного розділа було розглянуто основні теоретичні та проектні засади створення розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі мікроконтролера ESP32 та FPGA Cyclone V. Визначено вимоги до такої системи, акцентуючи увагу на централізованому управлінні та використанні апаратно-орієнтованих рішень для критичних задач реального часу. Далі описано архітектуру системи, яка побудована за дворівневим ієрархічним принципом, де ESP32 виступає єдиним головним контролером і джерелом конфігураційних команд, а два ідентичні вузли на базі FPGA Cyclone V забезпечують незалежне керування окремими перехрестями. Особливу увагу приділено способу організації зв'язку між рівнями – через послідовну шину UART, що працює за топологією «один передавач – багато приймачів», що забезпечує простоту реалізації, достатню пропускну здатність і надійність у реальних умовах. Також було представлено опис протоколу обміну даними та формату команд, а також проведено аналіз та обґрунтовано вибір апаратних засобів.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						40
Зм.	Арк.	№докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ РОЗПОДІЛЕНОЇ СПЕЦІАЛІЗОВАНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ КЕРУВАННЯ СВІТЛОФОРАМИ НА БАЗІ FPGA ТА ESP32

3.1 Реалізація розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 у середовищі Quartus II

Реалізація розподіленої системи керування світлофорами виконана у середовищі Quartus Prime Lite Edition з використанням мови опису апаратури VHDL стандарту 2008. Проєкт орієнтований на програмовані логічні інтегральні схеми сімейства MAX 10 виробництва Intel (Altera), зокрема на кристал 10M50DAF484C7G, що встановлений на навчальній платі DE10-Lite. Вибір даної платформи обумовлений наявністю достатньої кількості логічних елементів для розміщення двох незалежних вузлів керування, вбудованою підтримкою протоколу UART на рівні GPIO-інтерфейсу та сумісністю з інструментарієм функціональної симуляції ModelSim-Altera.

Проєкт створювався за допомогою майстра New Project Wizard у середовищі Quartus II. На етапі створення було визначено верхній рівень ієрархії – модуль traffic_system_top, який виступає кореневим елементом усієї проєктної ієрархії та інстанціює обидва вузли перехрестя (рис. 3.1). Усі файли вихідного коду на мові VHDL додавались до проєкту через діалог Add/Remove Files in Project у визначеному порядку, що відповідає ієрархії залежностей між модулями: першим компілюється пакет спільних типів traffic_pkg.vhd, після чого – базові компоненти, потім складніші модулі верхнього рівня (рис. 3.2). Порядок компіляції є критично важливим, оскільки кожен модуль посилається на типи та константи, оголошені у пакеті traffic_pkg, і його відсутність у бібліотеці work на момент компіляції залежних модулів призведе до помилки розбору.

Ключовим структурним елементом проєкту є пакет traffic_pkg.vhd, в якому зосереджено всі спільні визначення системи. У пакеті оголошено перелічуваний тип t_phase, що описує вісім можливих станів FSM світлофора: PHASE_IDLE, PHASE_ALL_RED, PHASE_NS_GREEN, PHASE_NS_YELLOW,

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 41
Зм.	Арк.	№докум.	Підпис	Дата		

Модуль `clk_prescaler.vhd` реалізовував параметризований дільник частоти, що перетворює вхідний тактовий сигнал частотою 50 МГц у послідовність одиничних імпульсів з частотою 1 Гц. Дільник побудований на основі лічильника з максимальним значенням $CLK_FREQ / OUT_FREQ - 1$, що для номінальних параметрів становить 49 999 999. При досягненні цього значення лічильник скидається до нуля, а вихідний сигнал `tick_1hz` формує імпульс тривалістю один такт. Обидва параметри `CLK_FREQ` та `OUT_FREQ` є узагальненими (`generic`), що дозволяє без зміни вихідного коду використовувати модуль як у реальній схемі на частоті 50 МГц, так і у прискореній симуляції на частоті 1 МГц.

Модуль `uart_rx.vhd` реалізував приймач послідовного інтерфейсу UART зі швидкістю передачі 115 200 бод у форматі 8N1. Приймач побудований як дворівнева машина станів: на першому рівні окремий процес здійснює побітове декодування вхідного сигналу `rx` з семплюванням у середині бітового інтервалу, на другому рівні – збирає три послідовно прийняті байти у пакет і формує строб `pkt_valid` разом із вмістом регістрів `pkt_byte0`, `pkt_byte1` та `pkt_byte2`. Такий поділ обов'язків дозволяє незалежно налаштовувати швидкість UART через параметр `BAUD_RATE` та розмір пакету без зміни логіки нижнього рівня. Параметри `CLK_FREQ` та `BAUD_RATE` є узагальненими, що забезпечує повну портованість модуля між різними тактовими частотами.

Модуль `cmd_decoder.vhd` декодував прийняті UART-пакети та зберігав параметри конфігурації у внутрішніх регістрах. Кожен вузол системи мав власний ідентифікатор `NODE_ID`, що передавався як узагальнений параметр під час інстанціювання. Декодер перевіряє поле адреси у старших двох бітах першого байта пакету та обробляє команду лише у тому випадку, якщо адреса збігається з ідентифікатором вузла або містить широкомовне значення. У проекті реалізовано такі типи команд:

- встановлення тривалості зеленої фази NS та EW;
- тривалості жовтого сигналу;
- тривалості захисного інтервалу ALL_RED;
- встановлення зміщення фази;

- встановлення ручної фази;
- ширококомовна синхронізація;
- скидання.

При отриманні команди оновлюється відповідний регістр конфігурації, а сигнал `cfg_updated` формує одиничний імпульс, сповіщаючи FSM про зміну параметрів.

Центральним модулем кожного вузла є `traffic_fsm.vhd` – синхронний скінченний автомат Мілі, що реалізує повний цикл керування перехрестям. Скінченний автомат побудований на двох процесах: синхронний процес `p_fsm` обробляє переходи між станами та керує лічильником таймера, комбінаційний процес `p_outputs` формує вихідні значення сигналів `ns_light` та `ew_light` залежно від поточного стану. Така двопроесна архітектура відповідає канонічному стилю опису FSM у VHDL і гарантує відсутність комбінаційних петель. Даний автомат підтримує також пріоритетну обробку керуючих сигналів: `do_reset` має найвищий пріоритет і переводить машину станів у стан `PHASE_ALL_RED` незалежно від поточного стану, сигнал `do_sync` – середній пріоритет із переведенням у стан `PHASE_IDLE` для відліку зміщення, зміна сигналу `mode` – обробляється на кожному такті в межах нормального циклу.

Модуль `traffic_node.vhd` об'єднує чотири описані компоненти у єдиний структурний модуль за допомогою явного інстанціювання через ключове слово `entity work`. Кожному екземпляру компонента призначаються відповідні внутрішні сигнали через `port map`, що забезпечує чітке розділення інтерфейсів між рівнями ієрархії. Вихідні триколірні сигнали розподіляються по окремих портах `ns_red`, `ns_yellow`, `ns_green`, `ew_red`, `ew_yellow` та `ew_green` через пряме звернення до бітів вектора `ns_light` та `ew_light`. Діагностичні виходи `dbg_phase`, `dbg_timer`, `dbg_mode` та `dbg_sync_ok` формуються шляхом виклику функцій перетворення з пакету `traffic_pkg` і призначені для використання з логічним аналізатором SignalTap II або при підключенні до зовнішнього мікроконтролера.

Верхній рівень проекту `traffic_system_top.vhd` інстанціює два екземпляри модуля `traffic_node` з різними значеннями узагальненого параметра `NODE_ID`:

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 44
Зм.	Арк.	№докум.	Підпис	Дата		

перший вузол отримує ідентифікатор 01, другий – 10. Обидва вузли підключаються до єдиного вхідного порту `master_uart_rx`, що відповідає фізичній топології розподіленої системи, в якій усі вузли приєднані до спільної шини даних від центрального контролера. Вихідні сигнали кожного вузла виведені на окремі порти верхнього рівня з префіксами `n1_` та `n2_` відповідно, що спрощує призначення виводів у файлі обмежень QSF.

Файл обмежень `traffic_ctrl.qsf` містить призначення виводів кристала відповідно до розводки плати DE10-Lite. Тактовий сигнал підключається до виводу `PIN_P11`, що відповідає генератору 50 МГц на платі. Сигнал скидання `rst_n` призначено на `PIN_B8`, що відповідає кнопці `KEY[0]` з активним низьким рівнем. UART-вхід від `master ESP32` підключається до виводу `PIN_W10` на роз'ємі `GPIO_1`. Вихідні сигнали світлофорів призначено на виводи світлодіодів `LEDR[0]–LEDR[9]` для демонстрації роботи на платі без додаткового обладнання. У файлі QSF також встановлено цільовий пристрій `10M50DAF484C7G`, стандарт вводу-виводу 3.3-V LVTTL для всіх зовнішніх сигналів та параметри синтезу в режимі збалансованої оптимізації.

Для функціональної симуляції проекту у середовищі ModelSim-Altera налаштований окремий потік верифікації через параметри EDA Tool Settings у Quartus. Testbench-модуль `tb_traffic_system.vhd` реєструється як іменованій стенд перевірки з верхнім модулем `tb_traffic_system`, що дозволяє запускати симуляцію безпосередньо з меню Tools – Run Simulation Tool – RTL Simulation. Альтернативно, симуляція запускається вручну через TCL-скрипт `sim_run.do`, який виконує компіляцію всіх модулів у бібліотеку `work` у правильному порядку залежностей, запускає симулятор та автоматично налаштовує вікно часових діаграм з іменованими та кольоровими сигналами для зручного аналізу результатів.

Загальна ієрархія проекту складається з дев'яти VHDL-файлів: одного пакету, п'яти RTL-модулів, одного структурного модуля верхнього рівня, одного модуля імітації `master`-контролера та одного `testbench`. Така організація проекту забезпечила чітке розмежування між синтезованою частиною (перші сім файлів) та верифікаційним оточенням (останні два файли), що відповідає промисловим

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 45
Зм.	Арк.	№докум.	Підпис	Дата		

практикам розробки цифрових систем на FPGA та дозволила спростити потенційне масштабування системи в майбутньому.

3.2 Симуляція розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 в ModelSim Altera

З метою перевірки працездатності спроектованої розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 було виконано її симуляцію та тестування в середовищі ModelSim Altera.

Всього було реалізовано два глобальні тестбенчі для симуляції. Наявність двох тестбенчів пояснювалась складністю одночасної перевірки як системи керування самим світлофором, так і комунікацією між вузлами у проєктованій спеціалізованій комп'ютерній системі. Тому перший тестбенч передбачав перевірку роботи керування саме одним світлофором через машину станів, тоді як другий перевіряв розсилку команд головним вузлом до підпорядкованих.

3.2.1 Симуляція керування одним світлофором за допомогою машини станів

Для верифікації коректності роботи скінченного автомата керування одним перехрестям було розроблено спрощений тестбенч `tb_simple.vhd`, який безпосередньо підключається до модуля `traffic_fsm` без проміжних рівнів UART-інтерфейсу та декодера команд. Такий підхід дозволив ізолювати логіку машини станів від комунікаційного рівня та чітко спостерігати саме за часовою поведінкою автомата в умовах симуляції.

З метою прискорення симуляції тактова частота була встановлена на рівні 1 МГц, а генератор сигналу `tick_1hz` формував один імпульс тривалістю один такт кожні 1000 тактів, що відповідало одній умовній секунді реального часу. Таким чином, часові параметри фаз задавались безпосередньо у секундах, а весь цикл роботи світлофора можна тоді спостерігати у межах декількох мілісекунд симуляційного часу.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 46
Зм..	Арк.	№докум.	Підпис	Дата		

Вхідні параметри конфігурації подавались безпосередньо на порти модуля traffic_fsm у вигляді сигналів типу unsigned. Тривалість зеленої фази для напрямку Північ–Південь (green_ns_dur) встановлена рівною 5 секундам, для напрямку Схід–Захід (green_ew_dur) – 4 секундам, тривалість жовтого сигналу (yellow_dur) – 2 секундам, а захисний інтервал повного червоного (allred_dur) – 1 секунді. Режим роботи задавався сигналом mode типу t_mode, початкове значення якого відповідає нормальному циклічному режиму MODE_NORMAL.

Симуляція розпочиналась із подачі активного низького сигналу скидання rst_n, після чого автомат переходив у початковий стан PHASE_IDLE та очікував синхросигналу (рис. 3.3). Сигнал do_sync, сформований тестбенчем через два такти після зняття скидання, ініціював відлік зміщення фази (phase_offset = 0), і автомат негайно переходив у стан PHASE_ALL_RED. Вихідні сигнали ns_light та ew_light набували значення LIGHT_RED для обох напрямків, що підтверджує коректну реалізацію захисної паузи між фазами.

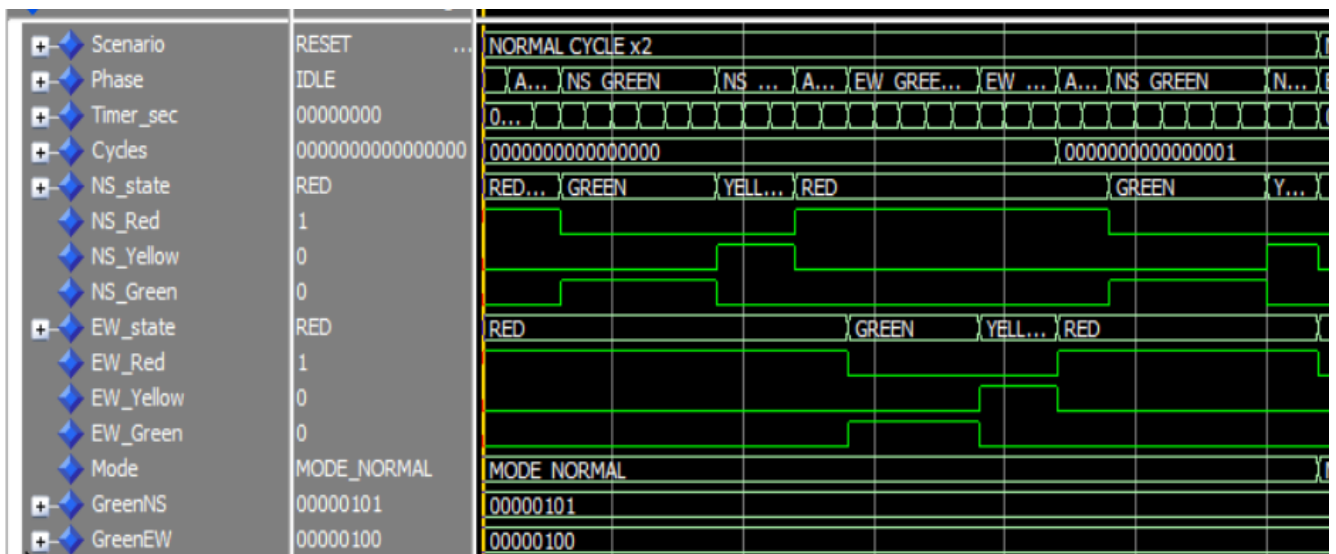


Рисунок 3.3 – Результати симуляції для стану NORMAL CYCLE (звичайний режим)

Після завершення одnoseкундного інтервалу ALL_RED автомат переходив у стан PHASE_NS_GREEN, тобто пріоритетним був напрямком був Північ–Південь. Вихідний сигнал ns_light набуває значення LIGHT_GREEN (бінарне 001), тоді як

ew_light залишається LIGHT_RED (100). Сигнал phase_timer починав відлік від нуля і збільшувався на одиницю із кожним імпульсом tick_1hz. Після досягнення значення 5 таймер скидався і автомат переходив у стан PHASE_NS_YELLOW: ns_light отримував значення LIGHT_YELLOW (010), тобто ew_light залишався червоним. Тривалість жовтої фази складала 2 секунди, після чого відбувався черговий перехід через стан ALL_RED.

У стані PHASE_EW_GREEN відбувається інверсія активного напрямку: ew_light набуває значення LIGHT_GREEN, ns_light – LIGHT_RED. Тривалість зеленої фази для напрямку Схід–Захід становить 4 секунди, після чого слідує PHASE_EW_YELLOW тривалістю 2 секунди та захисний інтервал ALL_RED. Після завершення цього інтервалу автомат знову переходить у PHASE_NS_GREEN, збільшуючи лічильник завершених циклів cycle_count на одиницю. Таким чином, повний цикл світлофора становить 15 умовних секунд, тобто $5 + 2 + 1 + 4 + 2 + 1$.

При проведенні перевірки важливою метою було впевнитись у відсутності колізії зелених сигналів, тобто у жодному стані автомата сигнали ns_light(0) та ew_light(0) не можуть бути одночасно рівними одиниці. Ця властивість забезпечується виключно через логіку переходів скінченного автомата без додаткових апаратних блокувань, що є перевагою архітектури скінченного автомата порівняно з комбінаційною логікою. Під час проведення симуляції подібної ситуації не виявлено, тому можна зробити висновок, що автомат працював коректно.

На другому етапі симуляції тестбенч змінював сигнал mode на значення MODE_NIGHT, що відповідало нічному режиму роботи (рис. 3.2). Як наслідок автомат негайно переходив у стан PHASE_BLINK незалежно від поточної фази циклу. У цьому стані обидва виходи ns_light та ew_light по чергово набувають значень LIGHT_YELLOW та LIGHT_OFF з частотою один герц, реалізуючи режим жовтого миготіння. Перехід між значеннями керувався внутрішнім сигналом blink_toggle, який інвертується при кожному імпульсі tick_1hz.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 48
Зм.	Арк.	№докум.	Підпис	Дата		

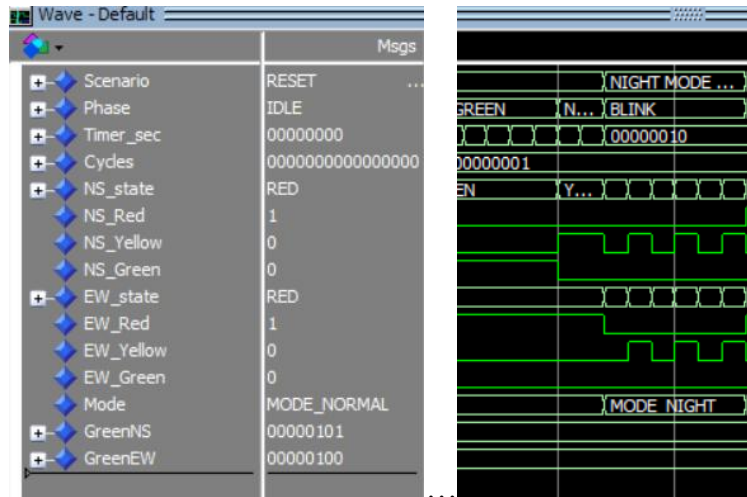


Рисунок 3.4 – Результати симуляції для стану MODE_NIGHT (нічний режим – миготіння жовтим)

Третій сценарій демонстрував роботу аварійного режиму: при встановленні mode = MODE_EMERGENCY автомат переходив у стан PHASE_FORCE_RED, в якому обидва виходи фіксуються у значенні LIGHT_RED незалежно від будь-яких таймерів (рис. 3.5). Цей режим моделює ситуацію проїзду пріоритетного транспортного засобу, коли всі напрямки руху мають бути заблоковані. Після повернення mode до значення MODE_NORMAL та подачі сигналів do_reset і do_sync автомат відновлював нормальний цикл з початкового стану, що підтверджує коректність обробки керуючих сигналів від вищого рівня системи.

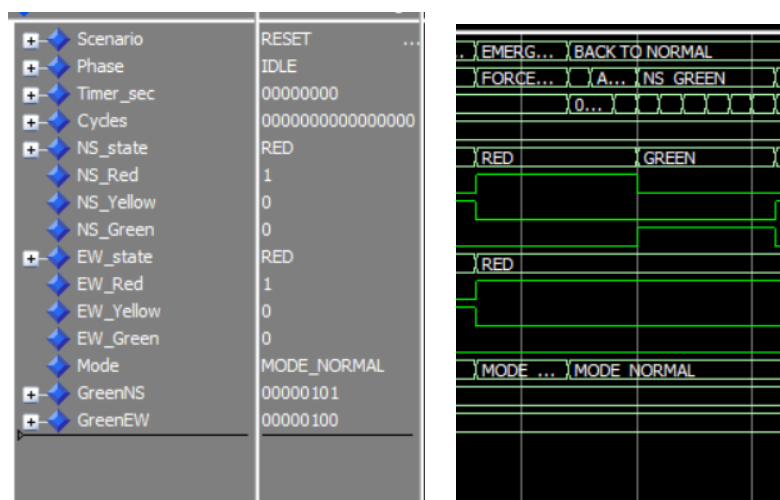


Рисунок 3.5 – Результати симуляції для стану MODE_EMERGENCY (екстремний режим – всі напрямки червоні)

Результати симуляції першого тестбенчу (керування одним світлофором через машину станів) для всіх режимів наведено на рис. 3.б.

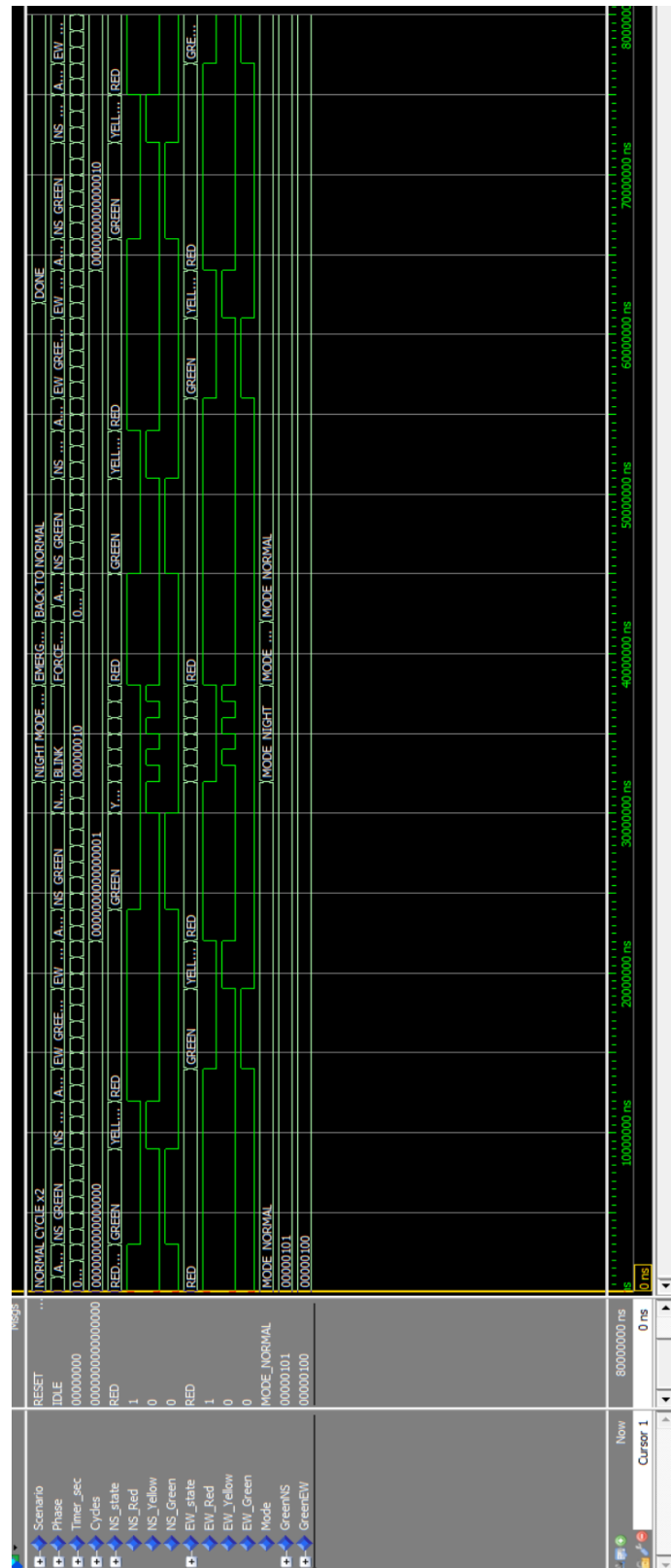


Рисунок 3.б – Результати симуляції першого тестбенчу (керування одним світлофором через машину станів)

Зм..	Арк.	№докум.	Підпис	Дата
------	------	---------	--------	------

Таким чином проведена симуляція і результати симуляції, відображені у вікні часових діаграм ModelSim, підтверджують повну відповідність реалізованого скінченного автомата заданим вимогам. Всі переходи між станами відбуваються точно після закінчення заданого інтервалу, вихідні сигнали змінюються синхронно з тактовим сигналом та не містять неконтрольованих глітчів, а пріоритетні режими (нічний та аварійний) активуються миттєво при зміні вхідного сигналу mode. Також не зафіксовано колізій коли обидва напрямки були б одночасно зеленими.

3.2.2 Симуляція розподіленої спеціалізованої комп'ютерної системи керування світлофорами

Симуляція повної розподіленої системи була реалізована у тестбенчі `tb_traffic_system.vhd`, який інтегрував всі компоненти проєкту, зокрема два вузли перехрестя (`traffic_node`), модуль імітації головного контролера (`master_controller`) та спільну UART-шину передачі даних. На відміну від спрощеного тестбенчу, у даному варіанті взаємодія між master-контролером та вузлами відбувається виключно через послідовний інтерфейс, що повністю відтворює реальну архітектуру системи.

Модуль `master_controller` реалізує скінчений автомат із двома групами станів: станами формування черги пакетів (`S_ENQ_*`) та станами передачі по UART (`S_TX_*`). При активації вхідного сигналу `send_config` контролер послідовно формує десять конфігураційних пакетів – по п'ять для кожного вузла – та додає до кінця черги широкомовний пакет синхронізації. Кожен пакет складався із трьох байтів: перший байт містив ідентифікатор вузла-адресата у бітах [7:6], код режиму роботи у бітах [5:4] та код команди у бітах [3:0]; другий та третій байти містять числові параметри відповідної команди.

Передача пакетів здійснювалась за протоколом UART зі швидкістю 9600 бод у режимі 8N1. Для прискорення симуляції тактова частота була знижена до 1 МГц, що дає значення `BIT_PERIOD` рівне 104 тактам. Стан `S_TX_START` формував стартовий біт низького рівня тривалістю один бітовий інтервал, стан `S_TX_DATA`

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 51
Зм.	Арк.	№докум.	Підпис	Дата		

попередньо передавав вісім біт даних молодшим бітом вперед, стан S_TX_STOP формував стоповий біт та короткий міжбайтовий інтервал. Між окремими пакетами витримувалась пауза тривалістю два бітових інтервали, що забезпечувало надійне розпізнавання меж пакетів приймачем.

Кожен вузол містив модуль uart_gx, який збирав вхідні байти в буфер та після отримання третього байта формував імпульс pkt_valid разом із вмістом трибайтового пакету на виходах pkt_byte0, pkt_byte1 та pkt_byte2. Модуль cmd_decoder перевіряв поле ідентифікатора адресата: пакет оброблявся лише у тому випадку, якщо поле node_id відповідало власному ідентифікатору вузла або містило широкомовне значення 00. Така архітектура дозволяла підключити обидва вузли до єдиної фізичної шини без додаткової комутаційної логіки.

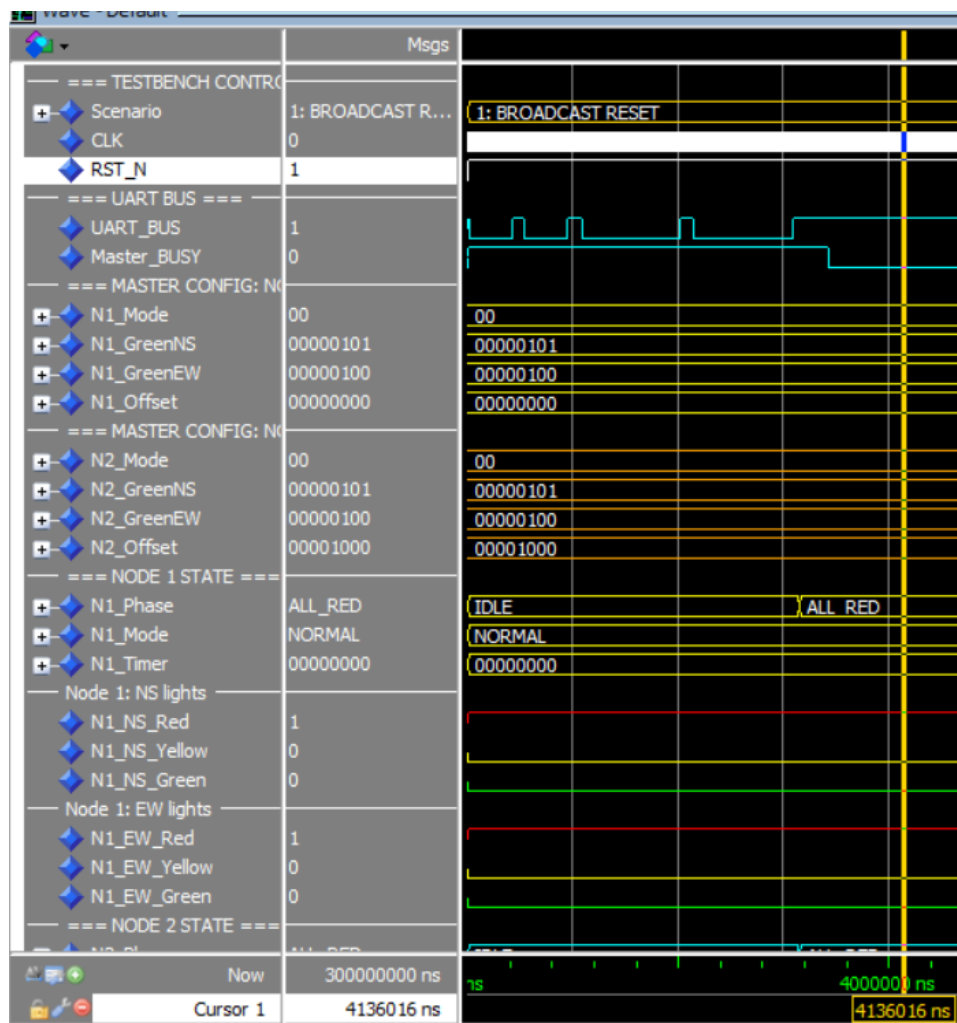


Рисунок 3.7 – Результати симуляції для стану BROADCAST_RESET

Перший сценарій симуляції демонстрував процедуру ініціалізації системи. Після зняття сигналу скидання `rst_n` testbench активує сигнал `send_reset`, що призводить до передачі ширококомовного пакету з кодом команди `CMD_RESET` (0x8). Обидва вузли одночасно отримують цей пакет та переходять у стан `PHASE_ALL_RED`, синхронізуючи свої FSM-автомати до єдиної точки відліку. Сигнал `dbg_sync_ok` на виходах обох вузлів встановлюється у нульове значення до моменту отримання наступного пакету синхронізації.

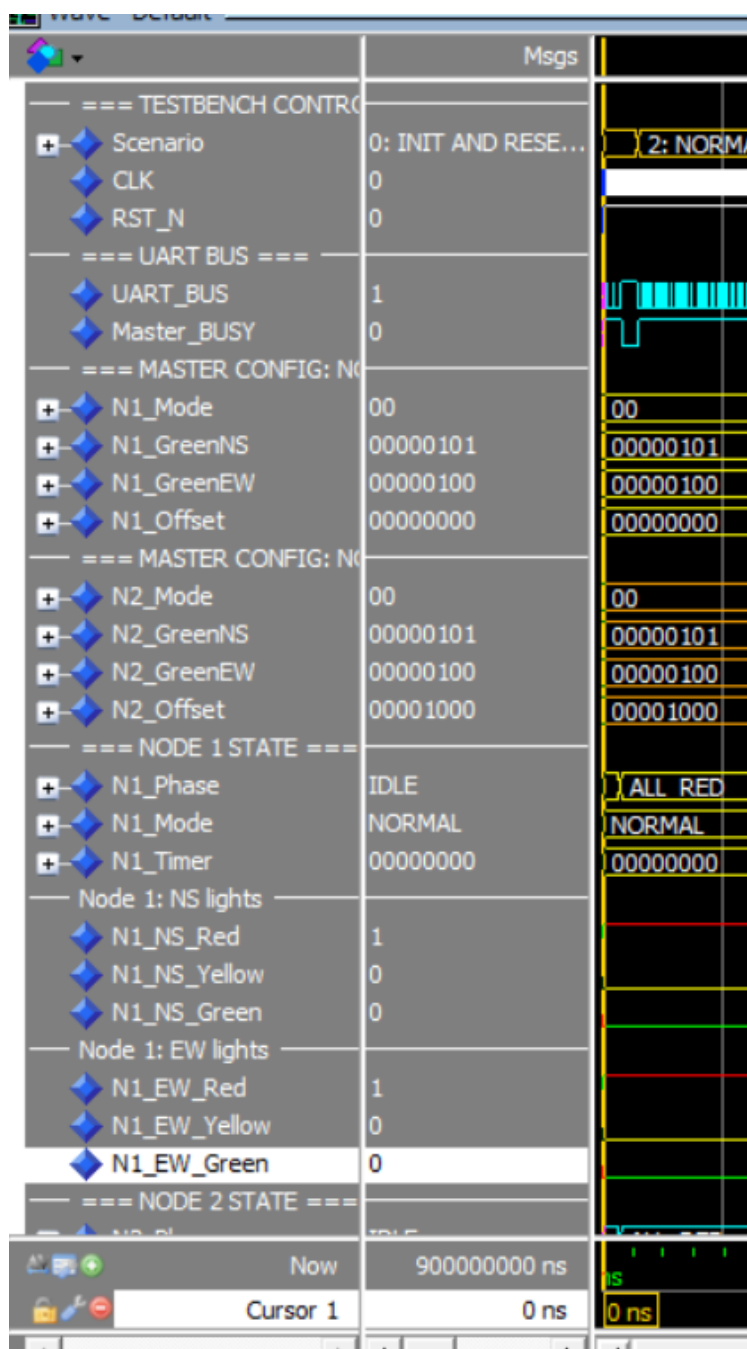


Рисунок 3.8 – N1_Offset (0) та N2_Offset (8)

У другому сценарії master-контролер передавав індивідуальні конфігураційні пакети для кожного вузла. Для вузла з ідентифікатором 01 встановлювались такі параметри: тривалість зеленої фази NS – 5 секунд, тривалість зеленої фази EW – 4 секунди, зміщення (phase_offset) – 0 секунд (рис. 3.9). Для вузла з ідентифікатором 10 встановлювались аналогічні тривалості фаз, однак зміщення становило 8 секунд. Після завершення конфігурування master-контролер автоматично формував ширококомовний пакет CMD_SYNC (0x7), який ініціював одночасний запуск обох вузлів.

Механізм зміщення фаз реалізовував принцип координованого керування, відомого як «зелена хвиля». Вузол 1 одразу після отримання сигналу синхронізації переходить у стан PHASE_ALL_RED та розпочинає нормальний цикл. Вузол 2 залишається у стані PHASE_IDLE протягом восьми умовних секунд, відраховуючи тіки сигналу tick_1hz за допомогою лічильника offset_cnt. Після досягнення заданого зміщення вузол 2 також переходить у PHASE_ALL_RED та розпочинає власний цикл. Внаслідок такої затримки транспортний засіб, що рухається з розрахунковою швидкістю між перехрестями, потрапляє на зелений сигнал на обох перехрестях послідовно, без зупинок.

Третій сценарій перевіряв динамічне перенастроювання параметрів у реальному часі. Тестбенч змінював значення сигналів m_n1_green_ns та m_n1_green_ew та активував send_config, після чого master-контролер передавав оновлені пакети CMD_SET_NS_DUR та CMD_SET_EW_DUR. Модуль cmd_decoder вузла 1 оновлював внутрішні регістри cfg_green_ns та cfg_green_ew, а FSM починав використовувати нові значення таймера вже з наступного переходу у стан PHASE_NS_GREEN або PHASE_EW_GREEN відповідно, не перериваючи поточної фази.

Четвертий та п'ятий сценарії демонстрували глобальне перемикання режимів роботи. При встановленні поля mode у значення 01 у ширококомовному пакеті обидва вузли переходили у режим MODE_NIGHT та стан PHASE_BLINK. На часових діаграмах сигнали ns_light та ew_light обох вузлів по чергово змінювались між значеннями LIGHT_YELLOW та LIGHT_OFF з однаковою фазою, оскільки вузли

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 54
Зм.	Арк.	№докум.	Підпис	Дата		

отримують ідентичний пакет синхронно. При встановленні mode у значення 10 обидва вузли фіксуються у стані PHASE_FORCE_RED, що моделює аварійну зупинку руху на всіх керованих перехрестях одночасно.

Шостий сценарій відтворював штатне відновлення роботи системи після аварійного режиму. Master-контролер передає пакети з mode = 00 (MODE_NORMAL) для обох вузлів, після чого надсилає широкомовний CMD_SYNC. Обидва вузли повертаються до нормального циклу з урахуванням збереженого значення зміщення, і на часових діаграмах знову спостерігається характерна восьмисекундна затримка між фазами вузла 1 та вузла 2. Сигнал dbg_sync_ok встановлюється у значення логічної одиниці в обох вузлах після успішного отримання пакету синхронізації, що підтверджує відновлення координованої роботи розподіленої системи.

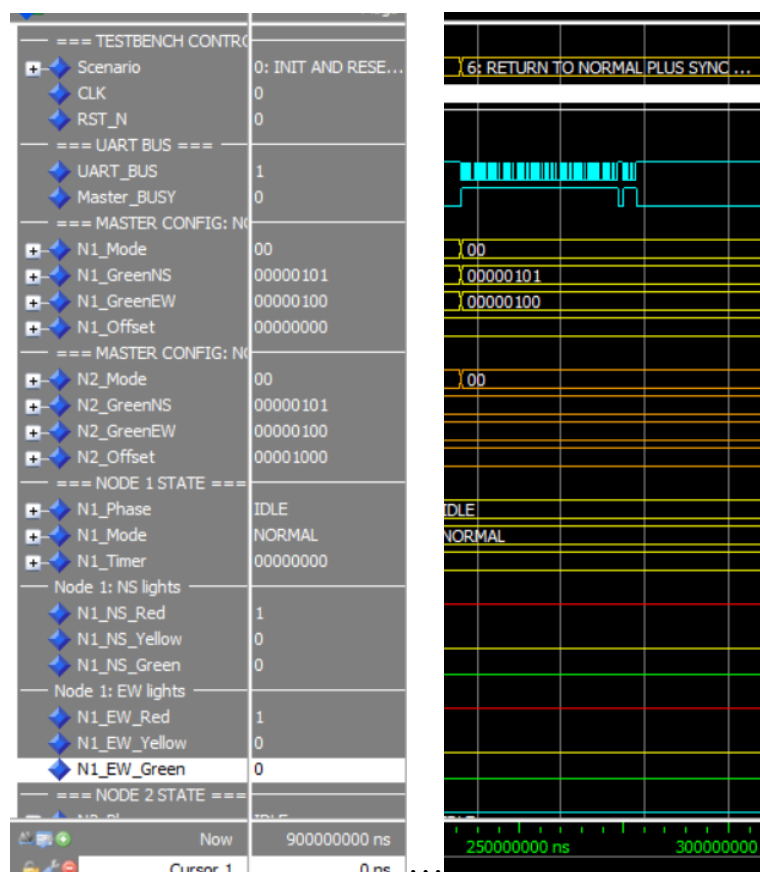


Рисунок 3.9 – Результати симуляції, що демонструють повернення до звичайного режиму роботи (mode=00)

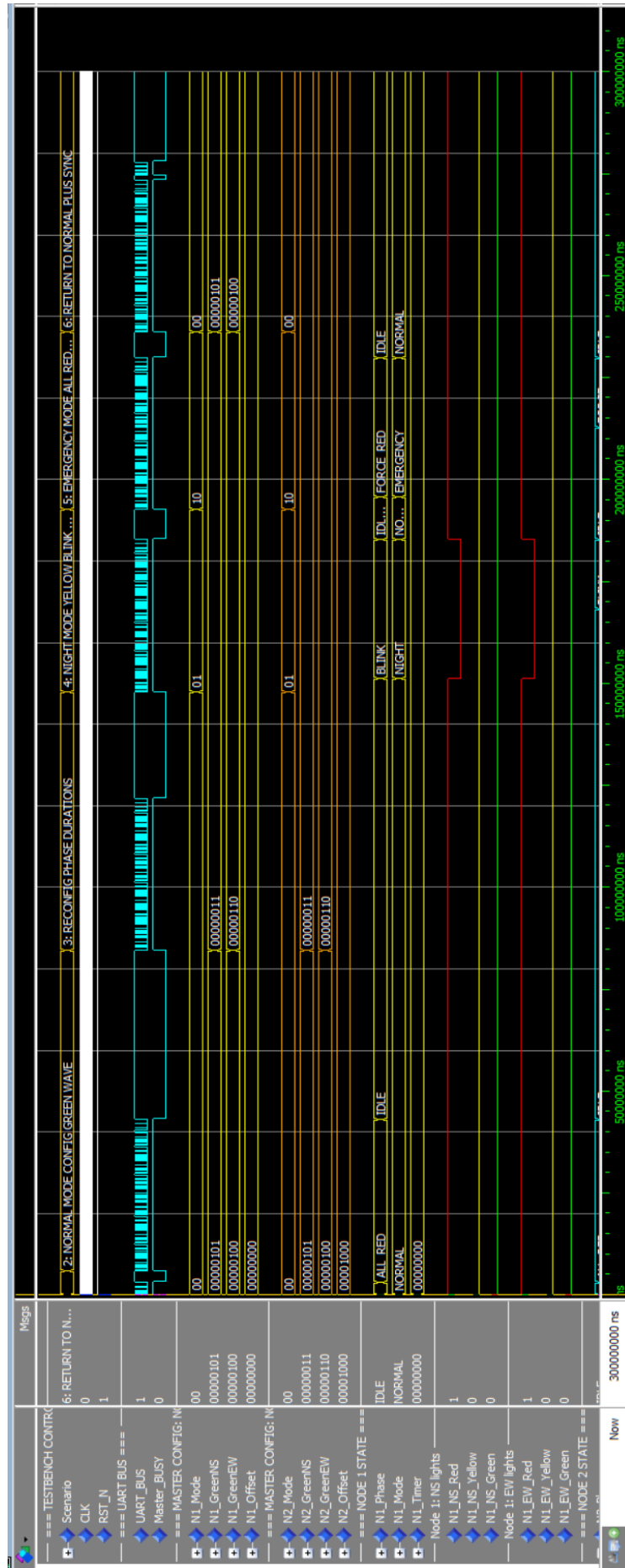
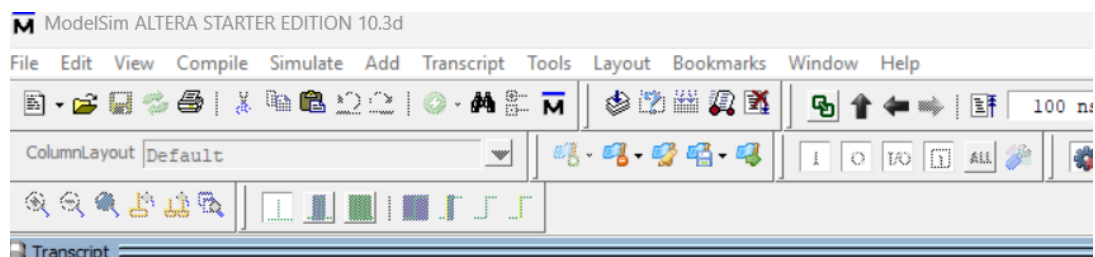


Рисунок 3.10 – Результати симуляції другого тестбенчу

Зм.	Арк.	№докум.	Підпис	Дата
-----	------	---------	--------	------

Загалом результати симуляції підтверджують, що запропонована архітектура розподіленої системи керування світлофорами забезпечувала досить надійну передачу конфігураційних команд по послідовному інтерфейсу, коректну адресацію вузлів у межах спільної шини, синхронізацію циклів між перехрестями за допомогою ширококомовних пакетів та гнучке динамічне перенастроювання параметрів без переривання роботи системи. Використання машини станів як основи керуючої логіки вузла забезпечувало детерміновану часову поведінку та відсутність критичних станів при будь-якій послідовності вхідних команд від master-контролера, що можна було побачити на формах результатів симуляції.

Вікно Transcript після закінчення симуляції наведено на рис. 3.11.



```

ModelSim ALTERA STARTER EDITION 10.3d
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
ColumnLayout Default
Transcript
# -- Loading package NUMERIC_STD
# -- Loading package traffic_pkg
# -- Compiling entity tb_traffic_system
# -- Compiling architecture sim of tb_traffic_system
# -- Loading entity traffic_system_top
# -- Loading entity master_controller
# End time: 11:26:06 on Mar 07,2026, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
# vsim
# Start time: 11:26:06 on Mar 07,2026
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading ieee.numeric_std(body)
# Loading work.traffic_pkg(body)
# Loading work.tb_traffic_system(sim)
# Loading work.traffic_system_top(structural)
# Loading work.traffic_node(structural)
# Loading work.clk_prescaler(rtl)
# Loading work.uart_rx(rtl)
# Loading work.cmd_decoder(rtl)
# Loading work.traffic_fsm(rtl)
# Loading work.master_controller(rtl)
# ** Warning: NUMERIC_STD.">=": metavalue detected, returning FALSE
# Time: 0 ns Iteration: 0 Instance: /tb_traffic_system/u_system/u_node2/u_traffic_fsm
# ** Warning: NUMERIC_STD.">=": metavalue detected, returning FALSE
# Time: 0 ns Iteration: 0 Instance: /tb_traffic_system/u_system/u_node1/u_traffic_fsm
# ** Note: === SIMULATION COMPLETE ===
# Time: 400817500 ns Iteration: 0 Instance: /tb_traffic_system
# ** Note: Check waveform: n1_phase_name, n2_phase_name
# Time: 400817500 ns Iteration: 0 Instance: /tb_traffic_system
# ** Note: Check: n1_ns_green, n1_ew_green, n1_dbg_timer
# Time: 400817500 ns Iteration: 0 Instance: /tb_traffic_system
# 0 ns
# 945 ms
# === Simulation finished ===

```

Рисунок 3.11 – Вікно Transcript після закінчення симуляції

3.3 Висновки

Таким чином було виконано практичну реалізацію та верифікацію спроектованої розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 у середовищі Quartus II. Після цього подальша функціональна симуляція була проведена у середовищі ModelSim Altera.

У процесі реалізації проекту у Quartus II було розроблено ієрархічну структуру з дев'яти VHDL-модулів, де пакет спільних типів утворює основу для всіх компонентів системи, а компонента верхнього рівня інтегрує два незалежні вузли перехресть через спільну UART-шину.

Кожен вузол у розподіленій спеціалізованій комп'ютерній системі керування світлофорами реалізований як структурний модуль, що об'єднує дільник частоти, UART-приймач, декодер команд та скінченний автомат керування. Симуляція керування одним світлофором за допомогою машини станів, виконана у спрощеному тестбенчі з прямим підключенням до модуля скінченного автомата, підтвердила коректність реалізації всіх восьми станів скінченного автомата та правильність часових переходів між ними. Симуляція повної розподіленої системи у іншому тестбенчі охопила сім сценаріїв, що відтворювали повний спектр експлуатаційних режимів.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						58
Зм.	Арк.	№докум.	Підпис	Дата		

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи бакалавра було спроектовано та розроблено розподілену спеціалізовану комп'ютерну систему керування світлофорами на базі FPGA та ESP32.

Розроблена система забезпечує автономне детерміноване керування світлофорними циклами на рівні окремого перехрестя засобами скінченного автомата, реалізованого на FPGA, та централізоване конфігурування параметрів роботи через мікроконтролер ESP32 по протоколу UART. Спроектована система підтримує програмовані тривалості фаз, часове зміщення між вузлами для реалізації режиму «зеленої хвилі», а також спеціальні режими роботи – нічний, аварійний та ручний – з можливістю перемикання у реальному часі без переривання роботи системи.

У першому розділі виконано аналіз існуючих підходів до побудови систем керування світлофорами, розглянуто архітектурні рішення на основі програмованих логічних інтегральних схем та мікроконтролерів, досліджено принципи побудови розподілених спеціалізованих комп'ютерних систем реального часу та обґрунтовано доцільність використання гібридної архітектури FPGA та ESP32 для реалізації детермінованого керування на рівні вузла з одночасною підтримкою мережевої координації між перехрестями. Досліджено теоретичні основи керування світлофорами.

У другому розділі розглянуто основні теоретичні та проектні засади створення розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі мікроконтролера ESP32 та FPGA Cyclone V. Визначено вимоги до такої системи. Приділено увагу способу організації зв'язку між рівнями – через послідовну шину UART, що працює за топологією «один передавач – багато приймачів», що забезпечує простоту реалізації, достатню пропускну здатність і надійність у реальних умовах. Також було представлено опис протоколу обміну даними та формату команд, а також проведено аналіз та обґрунтовано вибір апаратних засобів.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						59
Зм.	Арк.	№докум.	Підпис	Дата		

У третьому розділі було виконано практичну реалізацію та верифікацію розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32 у середовищі Quartus II з подальшою функціональною симуляцією у середовищі ModelSim Altera. У процесі реалізації проекту у Quartus II було розроблено ієрархічну структуру з дев'яти VHDL-модулів, де пакет спільних типів утворює основу для всіх компонентів системи, а компонента верхнього рівня інтегрує два незалежні вузли перехресть через спільну UART-шину. Кожен вузол у розподіленій спеціалізованій комп'ютерній системі керування світлофорами реалізований як структурний модуль, що об'єднує дільник частоти, UART-приймач, декодер команд та скінченний автомат керування. Симуляція керування одним світлофором за допомогою машини станів, виконана у спрощеному тестбенчі з прямим підключенням до модуля скінченного автомата, підтвердила коректність реалізації всіх восьми станів скінченного автомата та правильність часових переходів між ними. Симуляція повної розподіленої системи у іншому тестбенчі охопила сім сценаріїв, що відтворювали повний спектр експлуатаційних режимів.

					КВРКІ. 22103.22.02.24 ПЗ	Арк.
						60
Зм.	Арк.	№докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Traffic Control in Cities: All-Inclusive Introduction. URL: <https://medium.com/@nasimullah/traffic-control-in-cities-all-inclusive-introduction-ab8bd796bf6e> (дата звернення: 17.03.2026).
2. Smart Cities That Get Traffic Control Right. URL: <https://blog.goodvisionlive.com/6-smart-cities-that-get-traffic-control-right> (дата звернення: 17.03.2026).
3. Traffic Management Solutions. URL: <https://www.digi.com/solutions/by-application/traffic-management> (дата звернення: 17.03.2026).
4. Evolution of USA Traffic Lights. URL: <https://optraffic.com/blog/evolution-of-usa-traffic-lights/> (дата звернення: 17.03.2026).
5. Choosing Traffic Signal Light Manufacturers. URL: <https://optraffic.com/blog/choosing-traffic-signal-light-manufacturers/> (дата звернення: 17.03.2026).
6. Akopov A.S., Beklaryan L.A. Traffic improvement in Manhattan road networks with the use of parallel hybrid biobjective genetic algorithm. *IEEE Access*. 2024. № 12. P. 19532–19552.
7. Yang S.K., Li J.C., Shi H.B. Mix-attention approximation for homogeneous large-scale multi-agent reinforcement learning. *Neural Computing and Applications*. 2023. № 35. P. 3143–3154.
8. Zhu C., Dastani M., Wang S. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*. 2024. № 38. P. 4.
9. Bokade R., Jin X., Amato C. Multi-agent reinforcement learning based on representational communication for large-scale traffic signal control. *IEEE Access*. 2023. № 11. P. 47646–47658.
10. He Y., Wang Y.H., Yu F.R., Lin Q.Z., Li J.Q., Leung V.C.M. Efficient resource allocation for multi-beam satellite-terrestrial vehicular networks: A multi-agent

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 61
Зм.	Арк.	№докум.	Підпис	Дата		

actor-critic method with attention mechanism. *IEEE Transactions on Intelligent Transportation Systems*. 2022. № 23. P. 2727–2738.

11. Nawaz T., Seminara M., Caputo S., Mucchi L., Cataliotti F., Catani J. IEEE 802.15.7-compliant ultra-low latency relaying VLC system for safety-critical ITS. *arXiv*. 2019. arXiv:1906.08773.

12. How IoT Simplifies the Smart Traffic Light System. URL: <https://www.bivocom.com/blog/how-iot-simplifies-the-smart-traffic-light-system> (дата звернення: 17.03.2026).

13. Traffic Light Control System. URL: <https://jackwinsafety.com/traffic-light-control-system/> (дата звернення: 17.03.2026).

14. Dynamic Traffic Light Management System. URL: <https://lightingequipmentsales.com/dynamic-traffic-light-management-system.html> (дата звернення: 17.03.2026).

15. Traffic Lights – Yunex Traffic. URL: <https://www.yunextraffic.com/newsroom/traffic-lights/> (дата звернення: 17.03.2026).

16. Shen W.-H., Tsai H.-M. Testing vehicle-to-vehicle visible light communications in real-world driving scenarios. In: *Proceedings of the 2017 IEEE Vehicular Networking Conference (VNC)*, Torino, Italy, 27–29 November 2017. P. 187–194.

17. Liang X., Du X., Wang G., Han Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*. 2019. № 68. P. 1243–1253.

18. Alvarez Lopez J., Behrisch M., Bieker-Walz L., Erdmann J., Flötteröd Y.P., Hilbrich R. Microscopic traffic simulation using SUMO. In: *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 4–7 November 2018. P. 2575–2582.

19. Taylor & Francis Online. Article page. URL: <https://www.tandfonline.com/doi/full/10.1080/03081060.2019.1609219> (дата звернення: 17.03.2026).

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 62
Зм.	Арк.	№докум.	Підпис	Дата		

20. Who Controls Traffic Lights? URL: <https://kutuotraffic.com/blog/who-controls-traffic-lights/> (дата звернення: 17.03.2026).
21. Smart Traffic Lights News. URL: <https://starisajt.podgorica.me/en/news/145> (дата звернення: 17.03.2026).
22. A New Era at Traffic Lights. URL: <https://en.haberler.com/a-new-era-at-traffic-lights-see-what-it-means-19637276/> (дата звернення: 17.03.2026).
23. Traffic Light Technology. URL: <https://www.kmeel.com/technology/traffic-light/> (дата звернення: 17.03.2026).
24. Introduction to Centralized Traffic Control. URL: <https://www.railwaysignallingconcepts.in/introduction-to-centralized-traffic-control/> (дата звернення: 17.03.2026).
25. Vieira M.A., Vieira M., Louro P., Vieira P. Cooperative vehicular communication systems based on visible light communication. *Optical Engineering*. 2018. № 57. P. 076101.
26. Bibri S.E. A foundational framework for smart sustainable city development: Theoretical, disciplinary, and discursive dimensions and their synergies. *Sustainable Cities and Society*. 2018. № 38. P. 758–794.
27. Bilal J.M., Jacob D. Intelligent traffic control system. In: *Proceedings of the 2007 IEEE International Conference on Signal Processing and Communications*, Dubai, United Arab Emirates, 24–27 November 2007. P. 496–499.
28. Yousefi S., Altman E., El-Azouzi R., Fathy M. Analytical model for connectivity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*. 2008. № 57. P. 3341–3356.
29. Shen W.-H., Tsai H.-M. Testing vehicle-to-vehicle visible light communications in real-world driving scenarios. In: *Proceedings of the 2017 IEEE Vehicular Networking Conference (VNC)*, Torino, Italy, 27–29 November 2017. P. 187–194.
30. Liang X., Du X., Wang G., Han Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*. 2019. № 68. P. 1243–1253.

31. Alvarez Lopez J., Behrisch M., Bieker-Walz L., Erdmann J., Flötteröd Y.P., Hilbrich R. Microscopic traffic simulation using SUMO. In: *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 4–7 November 2018. P. 2575–2582.
32. Caputo S., Mucchi L., Cataliotti F., Seminara M., Nawaz T., Catani J. Measurement-based VLC channel characterization for I2V communications in a real urban scenario. *Vehicular Communications*. 2021. № 28. P. 100305.
33. Miranda R.F., Barriquello C.H., Reguera V.A., Denardin G.W., Thomas D.H., Loose F., Amaral L.S. A review of cognitive hybrid radio frequency/visible light communication systems for wireless sensor networks. *Sensors*. 2023. № 23. P. 7815.
34. Mohebbi S., Zhang Q., Wells E.C., Zhao T., Nguyen H., Li M., Abdel-Mottaleb N., Uddin S., Lu Q., Wakhungu M.J. et al. Cyber-physical-social interdependencies and organizational resilience: A review of water, transportation, and cyber infrastructure systems and processes. *Sustainable Cities and Society*. 2020. № 62. P. 102327.
35. Alahi M.E.E., Sukkuea A., Tina F.W., Nag A., Kurdthongmee W., Suwannarat K., Mukhopadhyay S.C. Integration of IoT-enabled technologies and artificial intelligence for smart city scenario: Recent advancements and future trends. *Sensors*. 2023. № 23. P. 5206.
36. Curtis C., Tiwari R. Transitioning urban arterial roads to activity corridors. *Urban Design International*. 2008. № 13. P. 105–120.
37. Falcocchio J.C., Levinson H.S. *Road Traffic Congestion: A Concise Guide*. New York: Springer, 2015.
38. Barnett J. *Design of Arterial Routes in Urban Areas*. Hong Kong: Public Roads Administration, 1948.
39. Kaluarachchi Y. Implementing data-driven smart city applications for future cities. *Smart Cities*. 2022. № 5. P. 455–474.
40. Mondal M.A., Rehena Z. An IoT-based congestion control framework for intelligent traffic management system. In: *Proceedings of the International Conference*

on *Artificial Intelligence and Data Engineering*, Udupi, India, 23–24 May 2019. P. 1287–1297.

41. Aleko D.R., Djahel S. An IoT enabled traffic light controllers synchronization method for road traffic congestion mitigation. In: *Proceedings of the 2019 IEEE International Smart Cities Conference (ISC2)*, Casablanca, Morocco, 14–17 October 2019. P. 709–715.

42. Rochet C., Belemlih A. Social emergence, cornerstone of smart city governance as a complex citizen-centric system. In: *Handbook of Smart Cities*. Cham: Springer International Publishing, 2020. P. 1–26.

43. Caputo S., Mucchi L., Cataliotti F., Seminara M., Nawaz T., Catani J. Measurement-based VLC channel characterization for I2V communications in a real urban scenario. *Vehicular Communications*. 2021. № 28. P. 100305.

44. Perle Systems. Perle Systems News and Events. URL: https://www.perle.com/news_events/pr-2022-apr-12.shtml (дата звернення: 17.03.2026).

45. Appio F.P., Lima M., Paroutis S. Understanding smart cities: Innovation ecosystems, technological advancements, and societal challenges. *Technological Forecasting and Social Change*. 2019. № 142. P. 1–14.

46. Black J. *Urban Transport Planning: Theory and Practice*. London: Routledge, 2018.

47. Vuchic V. *Transportation for Livable Cities*. London: Routledge, 2017.

48. Weiner E. *Urban Transportation Planning in the United States: History, Policy, and Practice*. New York: Springer, 2016.

49. De Souza A.M., et al Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*. 2017. № 13.

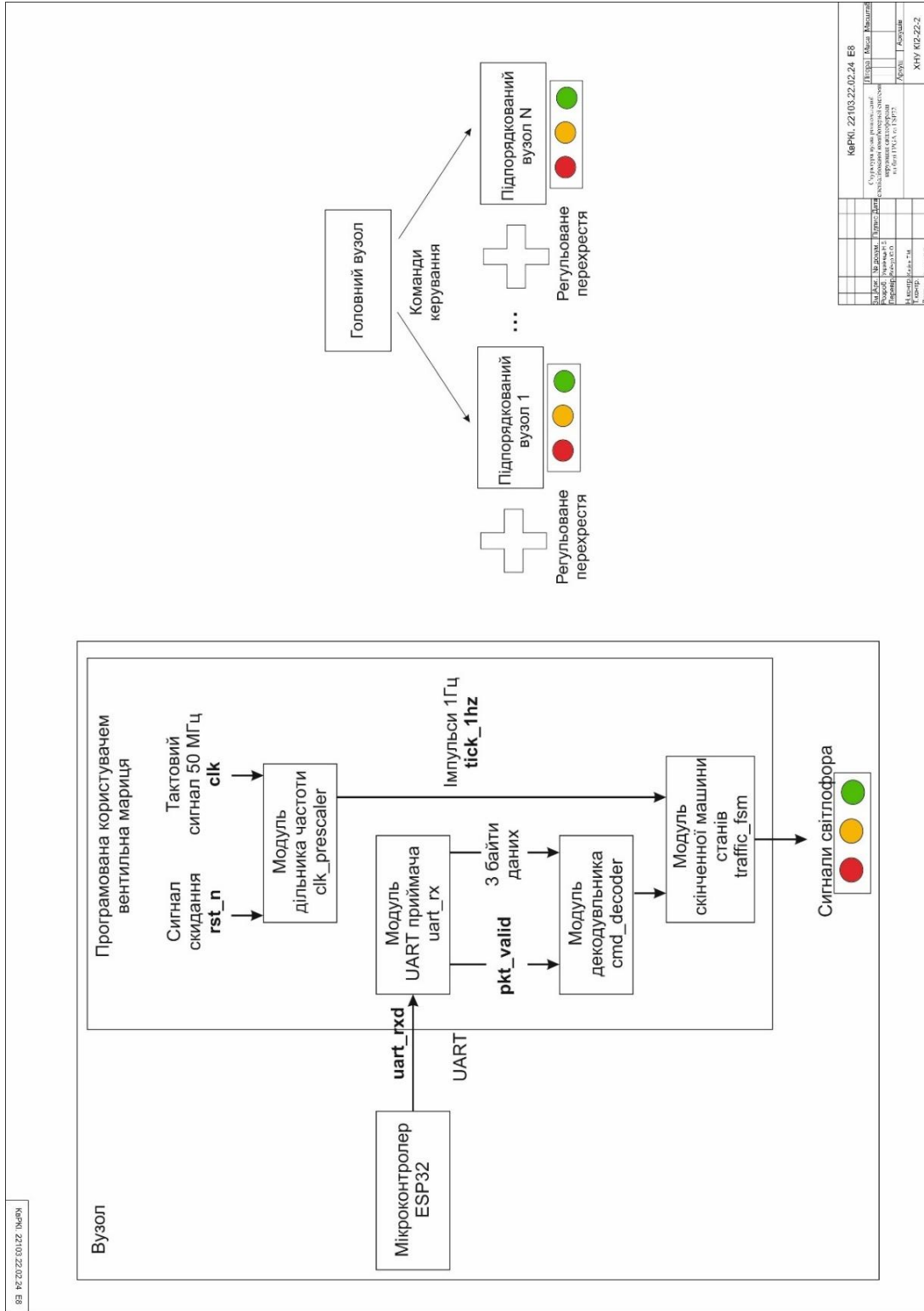
50. Jagarlamudi L. Examining the Effectiveness of Park-and-Ride Facilities in German Metropolitan Areas: Promoting Modal Shift and Mitigating Traffic Congestion. Master's Thesis. Zwickau: West Saxon University of Applied Sciences, 2024.

					КВРКІ. 22103.22.02.24 ПЗ	Арк. 65
Зм.	Арк.	№докум.	Підпис	Дата		

ДОДАТОК А

(обов'язковий)

Копія креслення «Структура вузла розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32»



Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Назар УКРАЇНЕЦЬ

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-2

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

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

1 травня 2026 року



Tue May 26 09:03:39 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилоч в документах: 10%**

ID: 272271 Назва: БКР Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32 Додано в БД: 2026-05-26 Автора: Назар УКРАЇНЕЦЬ Керівники: Юрій ВОЙЧУР Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	87393	542	1412 (2%)	22 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Назар УКРАЇНЕЦЬ

Співавтор:

Назва: Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32

Експерт: Юрій ВОЙЧУР

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 5.34%

Коефіцієнт подібності 2: 2.73%

Мікропробіли: 3

Заміна букв: 5

Інтервали: 0

Білі знаки: 6

Дата створення звіту: 2026-05-26 00:39:31.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-05-26



Доцент Андрій Нічепорук

Дата

експерт

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Назар УКРАЇНЕЦЬ

Тема: Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 60

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка та верифікація розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та мікроконтролера ESP32, в якій логіка керування кожним перехрестям реалізована у вигляді скінченного автомата на програмованій логічній інтегральній схемі.

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: У першому розділі кваліфікаційної роботи проведено аналіз предметної області систем керування світлофорами, а саме: виконано аналіз відомих систем керування світлофорами; досліджено сучасні стратегії керування світлофорними об'єктами; розглянуто теоретичні основи керування світлофорами; проаналізовано математичні моделі перехрестя, алгоритми керування світлофорами та критерії ефективності систем керування; сформовано висновки та постановку задачі дослідження.

У другому розділі кваліфікаційної роботи виконано проектування розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та Espressif Systems ESP32, а саме: визначено вимоги до системи; розроблено архітектуру розподіленої спеціалізованої комп'ютерної системи; описано протокол передачі даних та формат команд; проведено аналіз і вибір апаратних засобів для реалізації системи; сформовано висновки щодо результатів проектування.

У третьому розділі кваліфікаційної роботи виконано реалізацію та симуляцію розподіленої спеціалізованої комп'ютерної системи керування світлофорами на базі FPGA та ESP32, а саме: реалізовано систему у середовищі Quartus II; виконано симуляцію системи у середовищі ModelSim Altera; проведено симуляцію керування одним світлофором за допомогою машини станів; виконано симуляцію функціонування розподіленої спеціалізованої комп'ютерної системи керування світлофорами та проаналізовано результати її роботи.

5. Негативні сторони роботи: До недоліків роботи можна віднести відсутність тестування системи в реальних дорожніх умовах та обмежене врахування динамічних змін транспортного потоку.

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 достатньому технічному рівні.

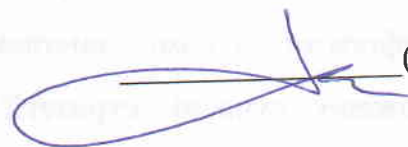
8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре С (75)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Дасігубе Олександр Анатолійович
доцент кафедри КН, ХНУ

"28" 05 2026 р.

 (підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Розподілена спеціалізована комп'ютерна система керування світлофорами на базі FPGA та ESP32

Автор Назар УКРАЇНЕЦЬ

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: ДФ, Юрій ВОЙЧУР

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

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

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) 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 StrikePlagiarism, складає 5,34% і адресується до 27 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

27.05.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

Підпис

Підпис

Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Юрій ВОЙЧУР
Ім'я, ПРІЗВИЩЕ