

## КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб реалізації процесорного ядра архітектури MIPS на  
основі FPGA

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 223154.22.94.08 ПЗ

Виконав здобувач IV курсу, групи КІ2-22-3



Підпис

Ярослав

ДМИТРИШЕН

Ініціали, прізвище

Керівник

Науковий ступінь, учене звання

Підпис

Сергій ЛИСЕНКО

Ініціали, прізвище

Нормоконтролер

Науковий ступінь, учене звання

Підпис

Сергій ЛИСЕНКО

Ініціали, прізвище

До захисту допускаю:  
завідувач кафедри КІС

« 1 » червня 2026 р.

Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

дата

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІІС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дмитришену Ярославу Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб реалізації процесорного ядра архітектури MIPS на основі FPGA

Керівник проекту (роботи) Лисенко Сергій Миколайович, д.т.н., проф.

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

Затверджена наказом ректора університету від 20.01.2026 р. № 5

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_  
Процесорне ядро архітектури MIPS на основі FPGA та постановка задачі щодо його удосконалення для реалізації програмно-технічних засобів

Проектування системи обробки інформації у процесорному ядрі архітектури MIPS на основі FPGA

Програмно-апаратна реалізація процесорного ядра архітектури MIPS на основі FPGA для програмно-технічних засобів моніторингу та керування

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

Система адаптивного керування навантаженням

Архітектура ПЗ для обчислювального ядра системи

Алгоритмічне забезпечення системи керування


6. Консультанти розділів кваліфікаційної 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	10.01.2026	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 I - дослідження предметної області, аналіз архітектур процесорів та постановка задачі проектування	01.03.2026	виконано
4	Робота над розділом II - вибір компонентів, обґрунтування технічних рішень та розробка архітектури процесора на базі FPGA	01.04.2026	виконано
5	Робота над розділом III - практичне проектування, симуляція та синтез блоків процесора на базі обраної FPGA	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач   
Підпис

Ярослав ДМИТРИШЕН  
Імя, ПРИЗВИЩЕ

Керівник кваліфікаційної роботи   
Підпис

Сергій ЛИСЕНКО  
Імя, ПРИЗВИЩЕ



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратної система моніторингу параметрів комп'ютерного обладнання».

Автор роботи: Ярослав ДМИТРИШЕН.

Керівник роботи: Сергій ЛИСЕНКО.

Пояснювальна записка: 64 с., 22 рис., 6 табл., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

АЛП, АРХІТЕКТУРА MIPS, ВЕРИФІКАЦІЯ, МОНІТОРИНГ, ПЛІС (FPGA), ПРОЦЕСОРНЕ ЯДРО, VERILOG.

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню програмно-технічного засобу реалізації процесорного ядра архітектури MIPS на базі сучасної ПЛІС-платформи. Актуальність теми зумовлена зростанням вимог до продуктивності, гнучкості та швидкодії обробки даних у вбудованих системах, спеціалізованих обчислювачах, контролерах енергоспоживання та елементах інтелектуальної інфраструктури. Своєчасний аналіз навантаження, керування живленням, порівняння параметрів та виконання команд дає змогу попереджати аварійні ситуації, знижувати енерговитрати і підвищувати загальну ефективність експлуатації цифрових систем.

Метою роботи є проектування, реалізація та тестування апаратно-програмного комплексу для збору, аналізу, оброблення й автоматичного реагування на зміну вхідних даних у реальному часі. Для досягнення поставленої мети було виконано аналіз сучасних підходів до побудови процесорних ядер, обрано елементну базу ПЛІС, розроблено структурну та функціональну схеми пристрою, спроектовано модулі процесора мовою Verilog і проведено часове моделювання для верифікації результатів розробки.

  
Підпис здобувача

30.05.2026  
Дата

## ЗМІСТ

Вступ.....	3
1 Аналіз архітектурних рішень та методів побудови вбудованих процесорних систем .....	4
1.1 Ретроспективний аналіз та ключові етапи еволюції архітектурних рішень мікропроцесорної техніки .....	4
1.2 Сучасні підходи до побудови обчислювальних вузлів на основі програмованої логіки .....	8
1.3 Модульний принцип організації компонентів процесорних ядер: методологія та структурні особливості.....	12
1.4 Аналіз ефективності застосування конвеєрної обробки даних у вбудованих обчислювальних системах.....	16
1.5 Висновки до розділу 1 .....	18
2 Аналіз проблематики та засобів проектування апаратної системи керування енергоспоживанням.....	20
2.1 Проблема балансування навантаження в сучасних інтелектуальних енергомережах.....	20
2.2 Архітектурні переваги процесорів MIPS для побудови систем адаптивного керування .....	26
2.3 Аналіз апаратної бази: особливості кристалів сімейства Intel Cyclone V та макетної плати DE1-SoC.....	31
2.4 Автоматизоване проектування цифрових пристроїв у середовищі САПР Quartus Prime .....	36
2.5 Висновки до розділу 2 .....	39
3 Програмно-апаратна реалізація кіберфізичної системи керування живлення електромережі.....	41
3.1 Логічне проектування та розгляд коду обчислювального ядра мовою Verilog HDL .....	41

КвРКІ.223154.22.94.08 ПЗ								
Зм.	Арк.	№ док.ум.	Підпис	Дата	Програмно-технічний засіб реалізації процесорного ядра архітектури MIPS на основі FPGA Пояснювальна Записка	Літера	Аркуш	Аркушів
Виконав		Ярослав дмитришин	<i>Дм</i>	01.06		у		1
Перевір.		Сергій ЛИСЕНКО	<i>СЛ</i>	01.06				
Н.контр.		Сергій ЛИСЕНКО	<i>СЛ</i>	01.06				
Затвер.		Ольга ПАВЛОВА	<i>ОП</i>	01.06				
						ХНУ КІ2-22-3		

3.2 Конфігурація фізичних інтерфейсів та призначення виводів ПЛІС .....	46
3.3 Аналіз апаратних ресурсів та результатів повної компіляції проект .....	51
3.4 Практичні випробування та експериментальна перевірка роботи контролера.....	58
3.5 Висновки до розділу 3 .....	64
Висновки .....	66
Перелік джерел посилань .....	68
Додаток А Копія креслення «Система адаптивного керування навантаженням».....	74
Додаток Б Копія креслення «Архітектура обчислювального ядра системи» .....	75
Додаток В Копія креслення «Алгоритмічне забезпечення системи керування».....	76

## ВСТУП

Сучасний стан енергетичної галузі України характеризується гострою потребою у впровадженні інтелектуальних систем керування, які здатні забезпечувати стабільність мережі в умовах непередбачуваних змін споживання та дефіциту потужності. Концепція інтелектуальних мереж електропостачання передбачає перехід від пасивного розподілу енергії до активного маніпулювання навантаженням, що вимагає розробки спеціалізованих програмно-технічних засобів. Традиційні рішення на базі мікроконтролерів загального призначення часто не забезпечують необхідної швидкодії та гнучкості при обробці великої кількості паралельних процесів у реальному часі. Саме тому актуальним є розробка та впровадження процесорних ядер архітектури MIPS, реалізованих на базі програмованих логічних інтегральних схем. Використання технології програмованої логіки дозволяє створити спеціалізовану апаратну архітектуру, де алгоритми захисту виконуються на рівні логічних елементів кристала, що гарантує детермінованість часових характеристик та високу надійність.

Метою даної роботи є розробка та практична реалізація програмно-технічного засобу у вигляді спеціалізованого процесорного ядра архітектури MIPS, призначеного для інтелектуального керування енергоспоживанням. Основна ідея проєкту полягає у створенні пристрою, який в автоматичному режимі оцінює поточний стан навантаження та здійснює селективне відключення споживачів відповідно до їх пріоритетності. Це дозволяє запобігати виникненню критичних аварійних ситуацій та забезпечує безперебійне живлення найбільш важливих об'єктів інфраструктури навіть за умов значного обмеження ресурсів. Для досягнення мети у роботі вирішуються завдання з проєктування апаратної логіки мовою Verilog, конфігурування інтерфейсів на базі сімейства Cyclone V та проведення серії експериментальних випробувань.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 3
Зм.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ ТА МЕТОДІВ ПОБУДОВИ ВБУДОВАНИХ ПРОЦЕСОРНИХ СИСТЕМ

## 1.1 Ретроспективний аналіз та ключові етапи еволюції архітектурних рішень мікропроцесорної техніки

Глибоке оцінювання генезису мікропроцесорних систем, проведене на основі огляду наукових публікацій у спеціалізованих виданнях, свідчить про те, що архітектурна думка розвивалася не стільки шляхом кількісного накопичення ресурсів, скільки через якісну трансформацію методів обробки інформації. На початкових етапах проектування, коли інтеграція транзисторів на напівпровідниковому кристалі вимірювалася тисячами одиниць, головним інженерним завданням розробників було створення мінімально життєздатної обчислювальної логіки. Проте вже на цьому етапі виникла фундаментальна дилема між універсальністю та апаратною ефективністю, яка згодом стала головним рушієм розробки та впровадження нових архітектурних стандартів.

Архітектурна модель та принципи побудови систем, що домінували протягом багатьох десятиліть, базуються на концепції спільного зберігання програм та даних у єдиному масиві пам'яті. Саме це інженерне рішення свого часу стало визначальним проривом у цифровій техніці, дозволивши нарешті перейти від вузькоспеціалізованих апаратних обчислювачів до гнучких програмованих комплексів, де ідентичну залізничну структуру та масив логічних вентилів можна було адаптувати для виконання абсолютно різних функціональних завдань. Але, з точки зору системного аналізу обчислювальних процесів, тут неможливо оминати проблему критичного обмеження пропускну здатності, яка в спеціалізованій літературі визначається як затримка каналу обміну. Суть зазначеного обмеження полягає в тому, що загальна апаратна продуктивність усієї цифрової системи безпосередньо впирається у максимальну швидкість почергової передачі інформаційних бітів між центральним обчислювальним вузлом та зовнішніми модулями пам'яті по одній спільній

					КвРКІ.223154.22.94.08 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

системній шині. Виникає деструктивна архітектурна ситуація, за якої розробники можуть безперервно нарощувати внутрішню потужність обчислювального ядра, інтегрувати швидші транзисторні структури та підвищувати тактову частоту, але загальна швидкість функціонування та реагування системи все одно буде жорстко обмежена пропускнуою здатністю єдиного комутаційного каналу зв'язку.

У контексті побудови сучасних високопродуктивних обчислювачів реального часу це обмеження стає визначальним дестабілізуючим фактором і призводить до появи тривалих непередбачуваних затримок при зчитуванні кодів операцій. Наочне порівняння цієї класичної моделі із спільним каналом зв'язку та альтернативної архітектурної концепції, яка повністю вирішує зазначену проблему завдяки впровадженню роздільних незалежних шин для команд та даних, представлено на рисунку 1.1.

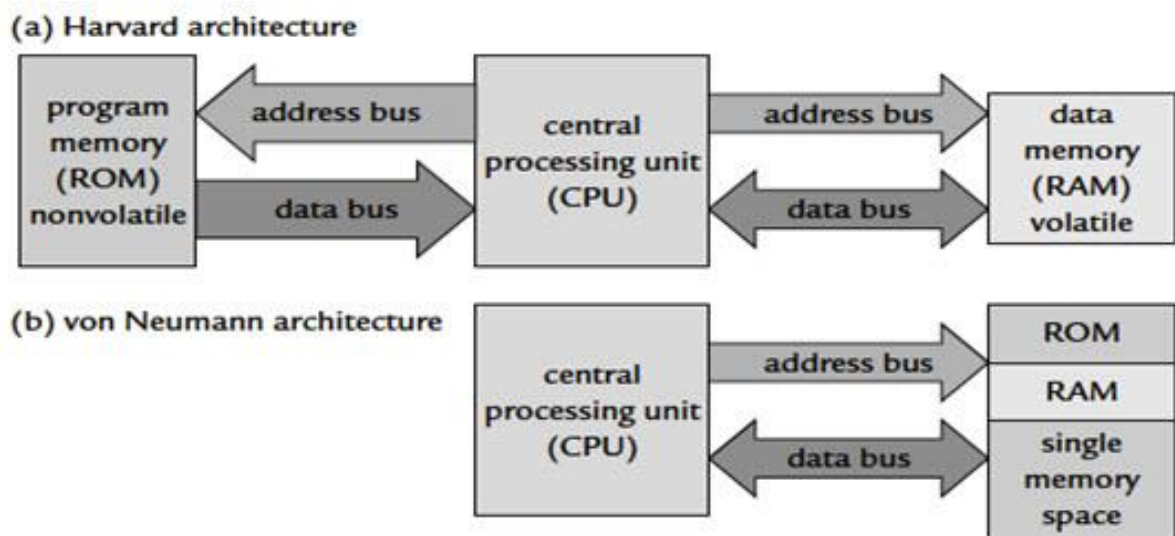


Рисунок 1.1 Порівняльна схема класичної архітектури (b)Фон Неймана та (a)Гарвардської моделі [46]

Гарвардська архітектурна модель, яка спочатку вважалася надто складною для практичного впровадження через необхідність розведення двох повністю

незалежних системних шин, згодом отримала широке розповсюдження у вбудованих обчислювачах та спеціалізованих мікросхемах цифрової обробки сигналів. Використання окремих фізичних трактів для вибірки інструкцій та одночасної маніпуляції інформаційними масивами фактично подвоїло потенційну пропускну здатність на рівні обчислювального ядра. Варто зауважити, що в сучасних напівпровідникових структурах часто спостерігається інтеграція обох зазначених підходів. Наприклад, на рівні внутрішньої кеш-пам'яті першого рівня високоефективних комп'ютерних платформ використовується роздільна топологія для забезпечення максимального паралелізму, тоді як загальна структура зовнішньої пам'яті залишається уніфікованою задля збереження повної сумісності із системним програмним забезпеченням.

Важливим етапом розвитку цифрової техніки є тривале протистояння концепцій побудови наборів команд зі складним та скороченим складом інструкцій. На ранніх етапах розвитку програмісти створювали коди переважно низькорівневими мовами символічного кодування, тому використання багатofункціональних команд, які виконували велику кількість мікрооперацій за один системний період, було інженерно виправданим. Проте з розвитком високоефективних компіляторів стало зрозуміло, що значна кількість рідко використовуваних апаратних інструкцій лише суттєво ускладнює логіку декодування, створює додаткові часові затримки та нераціонально збільшує загальну фізичну площу напівпровідникового кристала. Концепція обчислень із меншим набором команд докорінно змінила підходи до проектування пристроїв. Вона базується на трьох фундаментальних принципах: суворому одноцикловому виконанні більшості базових операцій, впровадженні великого масиву внутрішніх регістрів загального призначення, а також забезпеченні абсолютно фіксованої довжини команд, що ідеально підходить для організації багатокаскадної конвеєризації.

					КВРКІ.223154.22.94.08 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

Цей архітектурний зсув дозволив інженерам повністю зосередитися на підвищенні робочої тактової частоти ядра та впровадженні складних апаратних механізмів динамічного передбачення напрямків розгалуження програмного коду. Якщо детально оцінити загальну динаміку змін основних технічних характеристик, яку наочно представлено на рисунку 1.2, стає очевидним, як внутрішня структурна оптимізація поступово витісняла екстенсивний шлях розвитку мікроелектроніки через виникнення непереборних фізичних обмежень щільності тепловиділення напівпровідникових елементів.

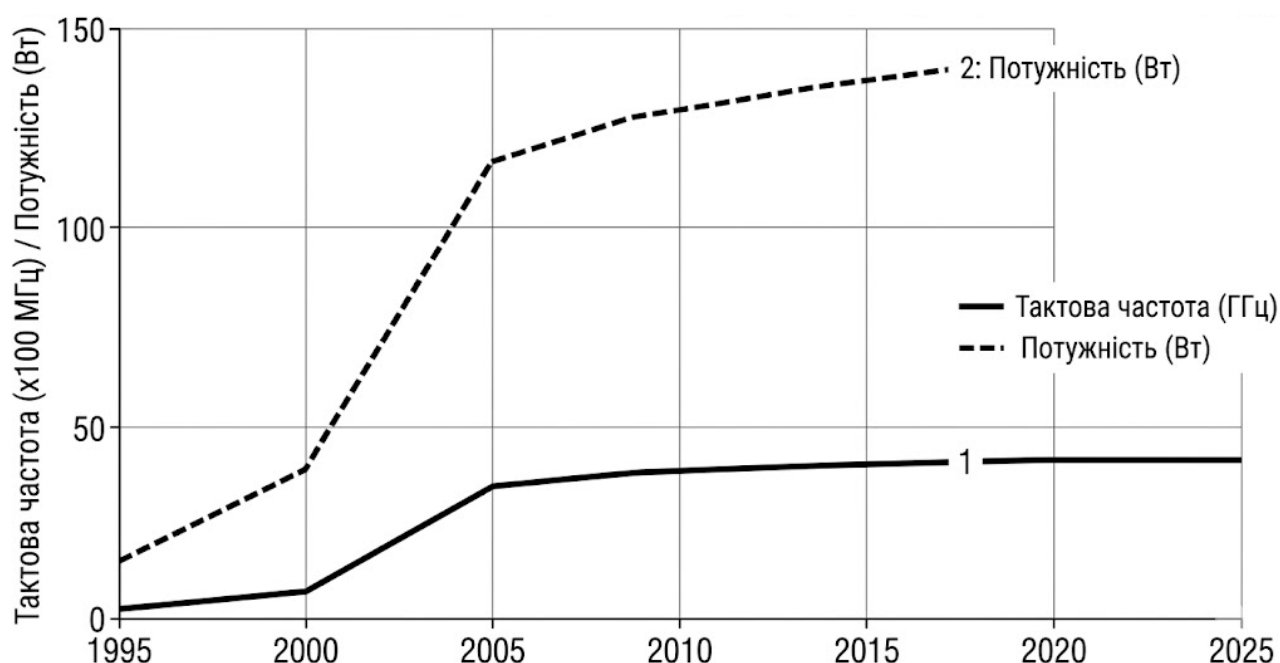


Рисунок 1.2 - Динаміка обмеження тактової частоти мікропроцесорів за критерієм тепловиділення

Представлені графічні залежності наочно ілюструють настання періоду глибокої кризи електронної щільності, відомого в інженерній практиці як тепловий бар'єр напівпровідникових структур. Графік чітко демонструє, що після дві тисячі п'ятого року крива робочої тактової частоти виходить на стабільне горизонтальне плато, тоді як загальне енергоспоживання обчислювальних пристроїв продовжує утримуватися на граничних критичних

значеннях. Зазначений фактор беззаперечно підтверджує неможливість подальшого нарощування швидкодії класичних одноядерних процесорів суто за рахунок підвищення частоти синхронізації без застосування радикальних і дорогих засобів рідинного або криогенного охолодження напівпровідникового кристала.

Завершуючи ретроспективний аналіз еволюції цифрових платформ, слід підкреслити, що кожен послідовний перехід на новий рівень фотолітографії - від мікрометрових до субмікронних та нанометрових масштабів - відкривав принципово нові можливості для інженерів та архітекторів. Коли загальна площа кремнієвої підкладки перестала бути критичним технологічним дефіцитом, з'явилися складні суперскалярні архітектури, здатні аналізувати та виконувати кілька незалежних інструкцій одночасно за один системний період. Впровадження зазначених рішень призвело до виникнення фундаментального поняття паралелізму на рівні інструкцій, що наразі є ключовим базисом для побудови високопродуктивних та енергоефективних обчислювальних ядер. Описаний технологічний перехід від форсування тактової частоти до просторового розпаралелювання інформаційних потоків став головним поштовхом для розвитку, вдосконалення та масового застосування спеціалізованих апаратних прискорювачів на базі програмованих логічних інтегральних схем.

## 1.2 Сучасні підходи до побудови обчислювальних вузлів на основі програмованої логіки

Перехід від жорстко зафіксованих архітектур до систем на основі програмованої логіки став визначальним етапом у розвитку сучасного цифрового проектування. Як свідчать актуальні дослідження, використання програмованих логічних інтегральних схем дозволяє ефективно подолати головний недолік спеціалізованих мікросхем із незмінною структурою, а саме

					КвРКІ.223154.22.94.08 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

їхню високу вартість розробки та повну неможливість апаратної модифікації після виготовлення кремнієвого кристала. У цьому контексті програмовані логічні матриці виступають як універсальне обчислювальне середовище, де архітектура процесорного ядра може бути гнучко адаптована під конкретне інженерне завдання навіть на пізніх етапах практичного впровадження та життєвого циклу пристрою. В основі побудови будь-якої сучасної програмованої матриці лежить масив конфігурованих логічних блоків, які з'єднані між собою складною розгалуженою мережею програмованих трасувальних трактів.

Фактично, така таблиця відповідності являє собою відокремлений блок статичної оперативної пам'яті низької місткості, адресами якого є вхідні сигнали необхідної логічної функції. Зазначений підхід дозволяє реалізовувати довільні логічні операції та комбінаційні перетворення без реальних часових затримок на перемикання вентилів, а суто шляхом простого і швидкого зчитування заздалегідь записаних цифрових значень. Структурну організацію такого типового логічного блоку для реалізації апаратних алгоритмів наочно представлено на рисунку 1.3.

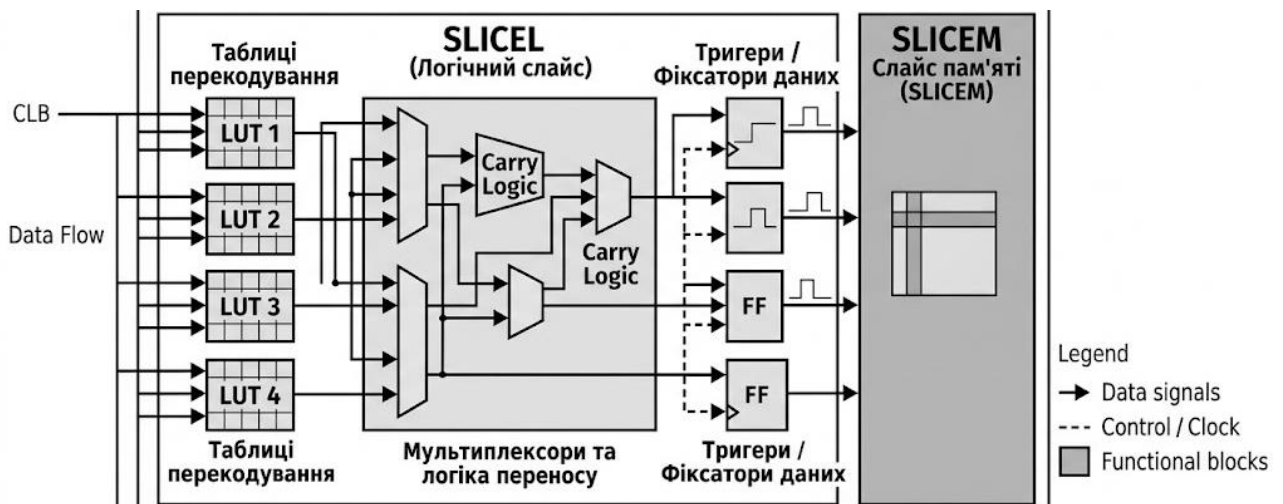


Рисунок 1.3 - Структурна організація типового логічного блоку (CLB) для реалізації апаратних алгоритмів [47]

Перехід від жорстко зафіксованих архітектур до систем на основі програмованої логіки став визначальним етапом у розвитку сучасного цифрового проєктування. Використання програмованих логічних інтегральних схем дозволяє ефективно подолати головний недолік спеціалізованих мікросхем жорсткої логіки, а саме їхню високу вартість розробки та неможливість модифікації структури після виготовлення кристала.

В основі будь-якої сучасної програмованої матриці лежить масив конфігурованих логічних блоків. Процес імплементації обчислювального вузла на базі таких структур суттєво відрізняється від традиційного алгоритмічного програмування. Використання мов опису апаратури, таких як Verilog, передбачає безпосередній опис фізичних зв'язків та паралельних потоків, де найбільш критичним етапом є оптимізація шляхів проходження сигналів. Структуру організації обчислень на рівні базових апаратних ресурсів представлено на рисунку 1.4.

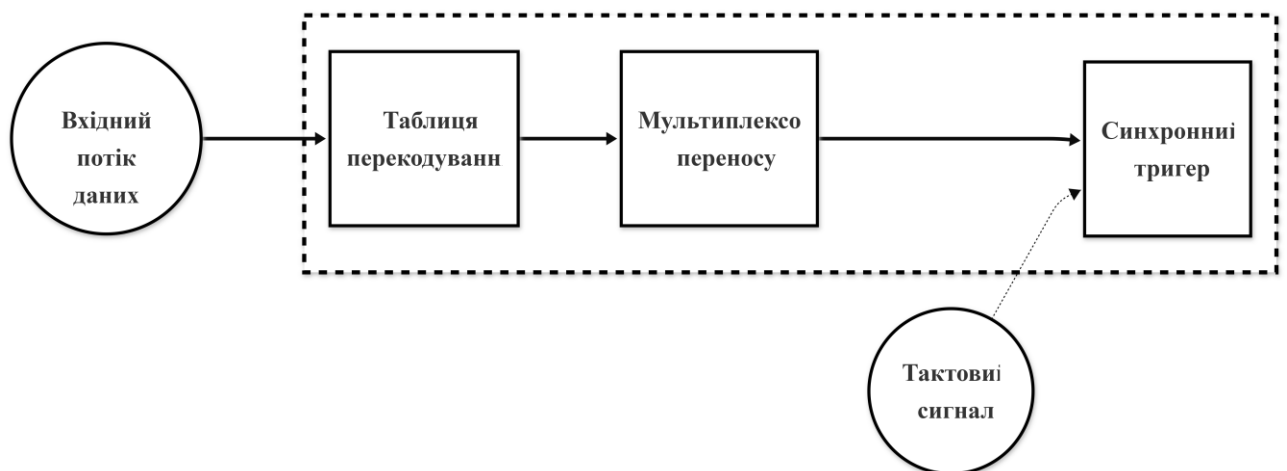


Рисунок 1.4 - Схема організації обчислювального конвеєра на базі апаратних ресурсів

Представлена апаратна архітектура дозволяє розглядати логічний слайс як мінімальний просторовий обчислювальний конвеєр фіксованої розрядності. Вхідний потік цифрових даних спочатку надходить на відповідні входи таблиці

перекодування, яка функціонує за принципом блоку статичної оперативної пам'яті низької місткості та реалізує комбінаційну логіку поточної команди або апаратну частину арифметичної функції. Фізичний принцип роботи зазначеного вузла полягає в тому, що замість тривалого перемикання транзисторів для обчислення виразу, система зчитує заздалегідь занесені результати таблиці відповідності за адресою, яку формують вхідні сигнали. Після завершення табличного етапу сформований сигнал проходить через спеціалізований мультиплексор переносу, який забезпечує високошвидкісне апаратне об'єднання інформаційних бітів під час виконання операцій додавання, віднімання або порівняння. Застосування виділених апаратних ланцюгів прискореного переносу дозволяє уникнути затримок, які виникають при послідовному проходженні сигналу через стандартну трасувальну матрицю кристала, що підвищує загальну швидкодію арифметичного вузла процесора. На завершальному етапі сформований результат обробки фіксується у внутрішньому синхронному тригері. Цей процес відбувається суворо за переднім фронтом тактового сигналу, що забезпечує чітку часову детермінованість та ізолює комбінаційні затримки поточного етапу обчислень від наступних каскадів загального конвеєра.

Така системна організація елементів дозволяє розробнику будувати масивні паралельні тракти обробки інформації, де тисячі аналогічних логічних слайсів функціонують незалежно та одночасно. Окрім базових конфігурованих слайсів, структури сучасних мікросхем містять вбудовані відокремлені блоки апаратної блочної пам'яті, а також спеціалізовані блоки цифрової обробки сигналів для швидкого апаратного множення та накопичення результатів. Інтеграція таких макрокомірок дозволяє повністю розвантажити універсальну логічну матрицю від виконання громіздких математичних операцій, що є критично важливим при реалізації алгоритмів фільтрації та аналізу параметрів у режимі реального часу. Зазначений комплекс апаратних засобів гарантує отримання стабільної часової затримки проходження сигналів та забезпечує фіксований час відгуку розробленого процесорного ядра на будь-які зовнішні

					КвРКІ.223154.22.94.08 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

переривання чи події. Це робить таку архітектурну концепцію основою для побудови високонавантажених обчислювальних систем автоматизації розумних мереж. Додатковою перевагою підходу є можливість успішно інтегрувати специфічні механізми прямого пересилання проміжних даних між регістрами, а також модулі передбачення напрямків розгалужень у конвеєрі інструкцій. Усі ці архітектурні рішення реалізуються на етапі опису апаратури, що дозволяє отримати показники швидкодії на рівні спеціалізованих мікросхем, але із збереженням гнучкості та можливості повної програмної перебудови структури безпосередньо під час прототипування та лабораторного відлагодження цифрового пристрою.

### 1.3 Модульний принцип організації компонентів процесорних ядер: методологія та структурні особливості

Сучасна методологія проектування складних обчислювальних систем на базі напівпровідникових кристалів базується на фундаментальному принципі модульності. Зазначений підхід дозволяє розглядати розроблене процесорне ядро не як монолітну нероздільну структуру, а як впорядковану сукупність автономних функціональних вузлів, взаємодія між якими чітко регламентована уніфікованими внутрішніми системними інтерфейсами та протоколами обміну даними. Застосування декомпозиції при розробці цифрових пристроїв на базі програмованих логічних інтегральних схем суттєво спрощує процеси написання коду мовами опису апаратури, полегшує етап верифікації окремих блоків та дозволяє реалізувати повторне використання раніше спроектованих апаратних модулів у майбутніх інженерних розробках.

Згідно з системним підходом, загальна архітектура цифрового пристрою розділяється на два ключові функціональні домени, якими є шлях даних та шлях керування. Шлях даних включає в себе всі апаратні модулі, що безпосередньо беруть участь у трансформації, обробці та зберіганні інформаційних потоків,

починаючи від внутрішнього регістрового файлу і закінчуючи конвеєрними суматорами арифметико-логічного пристрою. Шлях керування, своєю чергою, повністю відповідає за чітку часову синхронізацію цих процесів, генеруючи відповідні апаратні сигнали дозволу запису, вибору модулів та мультиплексування на основі декодованих інструкцій. Повна ієрархічна структура взаємодії зазначених доменів та периферійних вузлів представлена на рисунку 1.5.



Рисунок 1.5 - Схема взаємодії пристрою керування та операційних модулів процесорного ядра

Представлена ієрархічна структура наочно демонструє міжмодульні зв'язки, топологію розподілу інформаційних шин та організацію єдиного простору синхронізації всередині проєктуваної системи. На найвищому рівні схеми розташовані глобальні канали синхронізації та скидання, які забезпечують одночасне надходження зовнішнього тактового сигналу та імпульсу початкового встановлення до всіх підпорядкованих каскадів. Центральним керуючим та обчислювальним елементом всієї архітектури виступає процесорне ядро, яке безпосередньо координує роботу нижнього рівня автономних функціональних

модулів процесорної системи. Взаємодія процесорного ядра з модулем пам'яті програм реалізується через виділені адресні та інформаційні тракти, що дозволяє безперебійно вибирати коди операцій конвеєром інструкцій. Для збереження поточних результатів обчислень, масивів даних та проміжних змінних розроблена структура містить відокремлений модуль пам'яті даних, пов'язаний з ядром через відповідні паралельні інтерфейси читання та запису. Роздільна організація потоків команд та потоків даних на рівні окремих блоків дозволяє повністю уникнути структурних конфліктів при одночасному зверненні до пам'яті, що суттєво підвищує загальну пропускну здатність процесора реального часу.

Зовнішня комутація обчислювального вузла із периферійними елементами розумної енергомережі здійснюється через спеціалізований модуль периферійного порту. Процесорне ядро здійснює керування цим блоком за допомогою внутрішніх ліній адресації, передаючи вихідний потік даних або приймаючи зовнішні інформаційні сигнали для подальшого аналізу. Модуль порту перетворює внутрішні сигнали системи у стабільний формат, що спрямовується на зовнішні виходи пристрою. Усі представлені блоки об'єднані спільними глобальними лініями тактової частоти, що гарантує сувору синхронність функціонування кожного апаратного регістра, повністю усуває небезпеку виникнення часових зсувів між модулями та забезпечує детермінованість часових характеристик напівпровідникового кристала. Організація пам'яті в даному проєкті реалізована за гарвардським принципом, що відображено через фізичне розділення просторів коду та даних. Масив пам'яті програм призначений суто для зберігання виконуваних інструкцій, доступ до яких контролюється лічильником команд. Водночас масив пам'яті даних використовується для оперативної маніпуляції змінними. Для ефективної роботи з розширеними адресами в архітектуру інтегровано покажчик даних та покажчик стека, що дозволяє коректно обробляти вкладені підпрограми та складні структури.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

Зовнішня взаємодія ядра забезпечується через чотири незалежні паралельні порти, кожен з яких має восьмибітну структуру. Окрім базових функцій введення-виведення, периферійний сегмент доповнено блоком програмованих таймерів та системою обробки переривань. Це дає змогу реалізувати апаратну реакцію на зовнішні події без постійного завантаження обчислювального ядра, що значно підвищує загальну енергоефективність та швидкість роботи системи в реальних умовах експлуатації. Особливе місце в модульній організації займає підсистема пам'яті. Оскільки швидкість роботи логіки в програмованій матриці зазвичай вища за швидкість доступу до зовнішніх оперативних пристроїв, виникає потреба в імплементації проміжних модулів, таких як кеш-пам'ять та блоки буферизації команд. Класичну пірамідальну модель такої ієрархічної організації рівнів пам'яті наведено на рисунку 1.6. Завершальним етапом формування структурної схеми є верифікація взаємодії всіх описаних вузлів у середовищі автоматизованого проєктування. Це дозволяє ще на етапі симуляції виявити можливі затримки при проходженні сигналів.

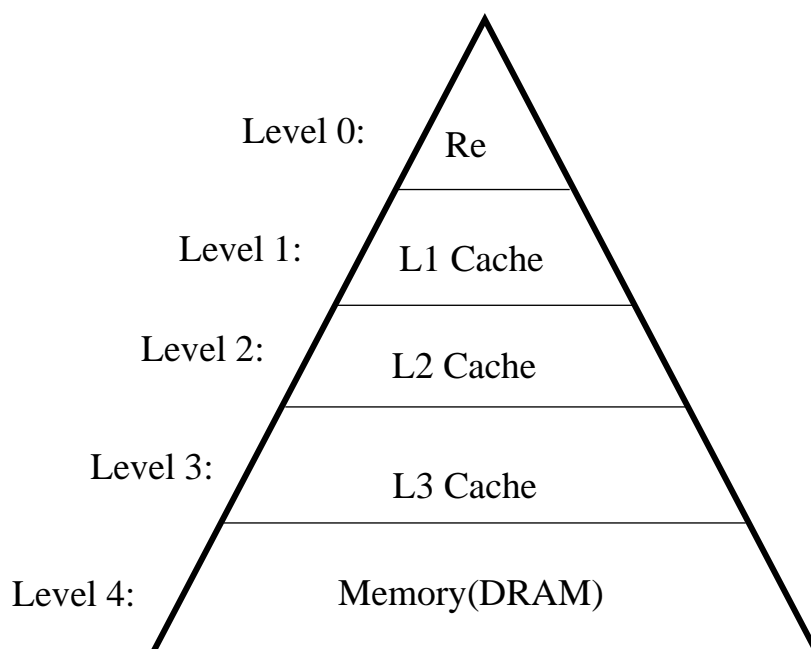


Рисунок 1.6 - Схема ієрархічної організації кеш-пам'яті у вбудованих обчислювальних рішеннях

Використання вбудованих блоків пам'яті (Block RAM) дозволяє реалізувати дворівневу ієрархію безпосередньо всередині кристала. Це суттєво зменшує затримки та запобігає простою конвеєра. Варто зауважити, що взаємодія між цими модулями відбувається через системні шини пропускну здатність яких повинна відповідати піковій продуктивності ядра.

#### 1.4 Аналіз ефективності застосування конвеєрної обробки даних у вбудованих обчислювальних системах

Впровадження конвеєризації є ключовим методом інтенсифікації обчислень, що дозволяє підвищити пропускну здатність процесорного ядра без нарощування тактової частоти. Ідея конвеєра базується на принципі часового паралелізму, де процес виконання інструкції розбивається на кілька незалежних стадій, які виконуються одночасно для різних команд. Це дозволяє ефективно використовувати апаратні ресурси кристала, оскільки в кожен момент часу різні блоки ядра задіяні в обробці різних частин інструкційного потоку. Класична п'ятиступенева структура включає такі послідовні фази: вибірка інструкції, декодування, виконання операції в арифметико-логічному пристрої, доступ до оперативної пам'яті та зворотний запис результату в регістровий файл. Кожна з цих стадій у середовищі програмованої логіки реалізується як окремий функціональний блок, розділений міжстадійними регістрами. Така побудова забезпечує можливість на кожному такті завершувати виконання однієї команди, попри те, що повний цикл її проходження через усі рівні обробки займає п'ять тактів. Принцип паралельного виконання команд проілюстровано на часовій діаграмі на рисунку 1.7.

Проте ефективність конвеєра залежить від відсутності системних конфліктів, які виникають через логічні залежності між командами. Виділяють три типи таких апаратних збоїв: структурні, конфлікти за даними та конфлікти керування. Структурні конфлікти виникають при одночасному зверненні різних

					КВРКІ.223154.22.94.08 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		



Аналізуючи класичні підходи до візуалізації конвеєрної обробки, можна проілюструвати фундаментальну концепцію просторового та часового розділення обчислювальних ресурсів мікропроцесорної системи. У таких теоретичних моделях кожен горизонтальний рядок відповідає окремій машинній команді, тоді як вертикальні розрізи репрезентують послідовні такти системного генератора. Характерною рисою цього процесу є те, що після початкового етапу заповнення конвеєрних стадій обчислювальний комплекс виходить на свій номінальний стаціонарний режим роботи. Саме в цьому стані розкривається весь потенціал конвеєризації: на кожному наступному такті успішно завершується виконання однієї повної інструкції, що наближає показник кількості тактів на інструкцію до одиниці. Зазначений підхід дозволяє суттєво підвищити швидкість обробки інформаційних потоків та забезпечує оптимальне використання наявних апаратних ресурсів напівпровідникового кристала без погіршення теплових характеристик пристрою.

### 1.5 Висновки до розділу 1

У першому розділі дипломної роботи здійснено ґрунтовний та всебічний розгляд теоретичних засад побудови сучасних процесорних ядер, а також детально проаналізовано специфіку їхньої апаратної реалізації на базі програмованих логічних інтегральних схем FPGA. Слід зазначити, що стрімкий розвиток мікроелектроніки та інформаційних технологій висуває нові, значно жорсткіші вимоги до спеціалізованих обчислювальних систем, що підтверджує високу актуальність обраного напряму дослідження. Проведений ретроспективний аналіз еволюції мікропроцесорної техніки чітко продемонстрував тенденцію до того, що невпинне зростання вимог до загальної швидкодії та енергоефективності систем неминуче зумовлює поступовий перехід від традиційних жорстких процесорних архітектур до більш адаптивних та гнучких апаратних рішень. У цьому контексті доведено, що саме

програмована логіка виступає тим сучасним технологічним базисом, який здатний забезпечити найбільш раціональний та оптимальний компроміс між високою обчислювальною продуктивністю, гнучкістю налаштувань та загальною вартістю життєвого циклу проектування.

Окрему увагу в межах теоретичного дослідження приділено структурним аспектам розробки. Зокрема, науково та практично доведено беззаперечну доцільність застосування модульного принципу організації процесорного ядра. З'ясовано, що логічний поділ складного обчислювального пристрою на відокремлені, функціонально незалежні апаратні вузли не лише суттєво оптимізує рутинні процеси проектування, тестування та верифікації опису апаратури, але й гарантує високий рівень масштабованості. Такий підхід дозволяє легко модифікувати окремі блоки системи без необхідності повної переробки всієї архітектури. Важливим етапом дослідження стало ґрунтовне обґрунтування необхідності впровадження класичної п'ятиступеневої конвеєризації як головного та найбільш ефективного інструменту розпаралелювання виконання машинних інструкцій безпосередньо на рівні кристала. Детальний розгляд архітектурних особливостей показав, що такий метод дозволяє кардинально підвищити загальну пропускну здатність процесорного ядра та ефективність використання тактових частот. Водночас наголошено, що максимальна продуктивність конвеєрного оброблення досягається виключно за умови коректної розробки та впровадження надійних апаратних механізмів виявлення і своєчасного вирішення структурних конфліктів, конфліктів керування та конфліктів за даними.

					КвРКІ.223154.22.94.08 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 АНАЛІЗ ПРОБЛЕМАТИКИ ТА ЗАСОБІВ ПРОЕКТУВАННЯ АПАРАТНОЇ СИСТЕМИ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

### 2.1 Проблематика балансування навантаження в сучасних інтелектуальних енергомережах

Сучасний етап розвитку глобальної енергетичної інфраструктури характеризується стрімким зростанням кількості споживачів та зміною самого характеру споживання електроенергії. Традиційні електромережі, які проектувалися десятиліття тому, базувалися на односторонній моделі постачання: від великої електростанції до кінцевого користувача. У такій системі керування здійснювалося переважно на стороні генерації. Проте сьогодні, з появою великої кількості енергоємних побутових приладів, систем промислової автоматизації та розподілених джерел енергії, класична топологія мережі демонструє свою неефективність.

Додатковим фактором, що значно ускладнює стабільну роботу застарілих мереж, є стрімка електрифікація транспорту та локальна генерація. Якщо раніше енергетичні потоки рухалися виключно в одному напрямку, то зараз кінцевий споживач часто має власні сонячні панелі або накопичувачі енергії. Відповідно, він перетворюється на активного учасника процесу - так званого просьюмера, який може як споживати, так і віддавати надлишки в загальну мережу. Це створює безпрецедентне непередбачуване навантаження на розподільчі вузли та трансформатори, які апаратно не розраховані на постійні реверсивні перетікання струмів. На зміну традиційним системам приходить концепція інтелектуальних енергомереж. Фундаментальна відмінність Smart Grid полягає у наявності зворотного зв'язку між постачальником та споживачем у режимі реального часу. Це - комплексна апаратно-програмна система, яка поєднує в собі електричні комунікації та сучасні інформаційні технології для автоматизованого збору даних про стан мережі.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Впровадження таких технологій вимагає розгортання розгалуженої екосистеми вимірювальних пристроїв: розумних лічильників, сенсорів струму та датчиків напруги безпосередньо на об'єктах споживання. Уся ця периферія безперервно генерує величезні масиви телеметричної інформації. Відповідно, ключовою задачею Smart Grid стає не лише накопичення цих даних, а їх миттєва апаратна обробка на локальному рівні для прийняття швидких керуючих рішень. Будь-яка затримка в обробці такої інформації під час раптового перевантаження або короткого замикання зводить нанівець усю концепцію інтелектуального захисту. Саме тому акцент у проєктуванні сучасних енергетичних систем сьогодні зміщується з простого нарощування генеруючих потужностей у бік створення високошвидкісних апаратних контролерів адаптивного керування навантаженням. Взаємозв'язок між рівнями управління та структурою обміну даними в таких системах наведено на рисунку 2.1.

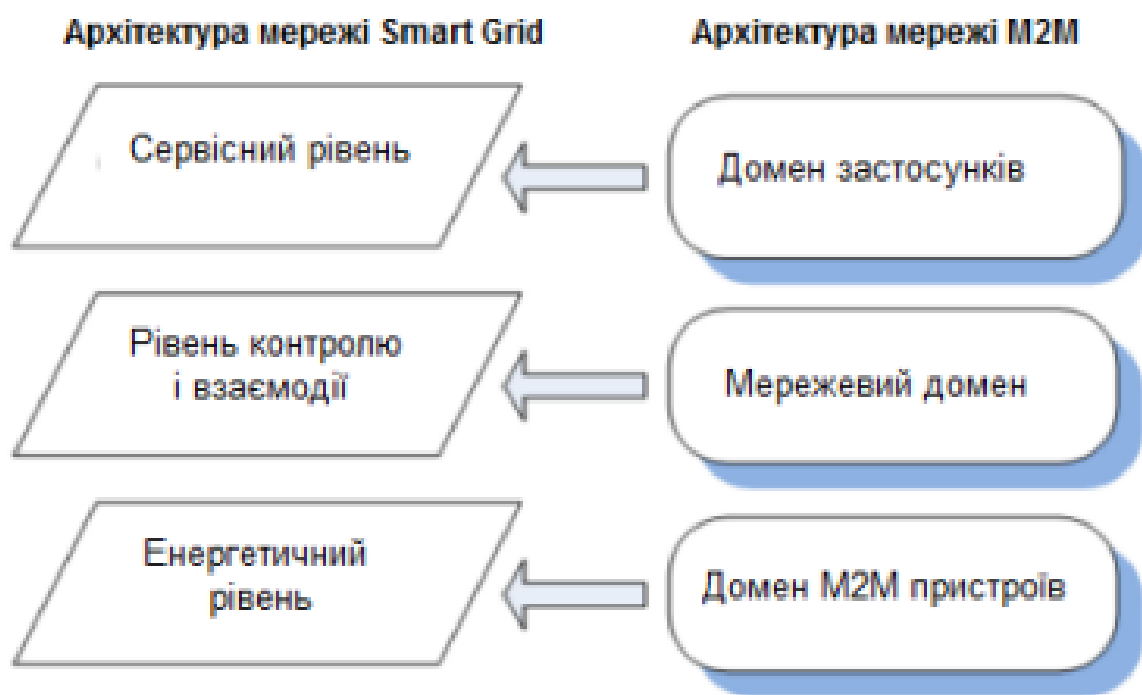


Рисунок 2.1 - Порівняльна структурна схема традиційної електромережі та мережі типу Smart Grid

Незважаючи на впровадження новітніх технологій, основною технічною проблемою будь-якої енергосистеми залишається необхідність дотримання суворого балансу між виробленою та спожитою електроенергією у кожний момент часу. Будь-яке, навіть короткочасне порушення цієї рівноваги миттєво позначається на базових параметрах якості струму в усій мережі, насамперед на його частоті. Електроенергія - це специфічний вид ресурсу, який, у масштабах макромережі, неможливо ефективно накопичувати та зберігати у великих обсягах. Існуючі промислові системи накопичення, такі як акумуляторні станції або гідроакумулюючі потужності, здатні лише частково компенсувати локальні коливання, проте їхньої ємності абсолютно недостатньо для покриття масштабних дефіцитів цілого регіону. Через це диспетчерські центри змушені безперервно, секунда за секундою, узгоджувати генерацію з поточним попитом.

Протягом доби графік електричного навантаження має яскраво виражені нерівномірності, що формуються під впливом побутових, промислових та кліматичних факторів. Інженери виділяють так звані ранкові та вечірні піки споживання. Ранковий максимум традиційно зумовлений початком робочого дня, масовим запуском промислового обладнання та виходом на маршрути міського електротранспорту. Проте найскладнішим для енергосистеми є вечірній пік, коли домогосподарства починають споживати максимум енергії: люди одночасно вмикають потужні побутові прилади, системи обігріву або охолодження, а також масово активуються мережі зовнішнього освітлення. Характерні зміни інтенсивності енергоспоживання протягом доби з чітким відображенням періодів максимального навантаження наведено на рисунку 2.2.

У ці періоди сумарна потужність усіх підключених пристроїв може наближатися до максимальної пропускної здатності трансформаторних підстанцій, магістральних кабельних ліній та розподільчих вузлів. Кожен елемент силової інфраструктури проектується з урахуванням суворих теплових лімітів. Коли навантаження перевищує номінальну потужність генеруючого або

розподільчого обладнання і тримається на цьому рівні, ізоляція провідників та трансформаторів не встигає ефективно відводити надлишкове тепло.

Відбуваються небезпечні аварійні процеси:

- 1) Зниження частоти змінного струму в мережі;
- 2) Критичне падіння напруги у кінцевих споживачів;
- 3) Теплове перевантаження трансформаторів та ліній електропередач.

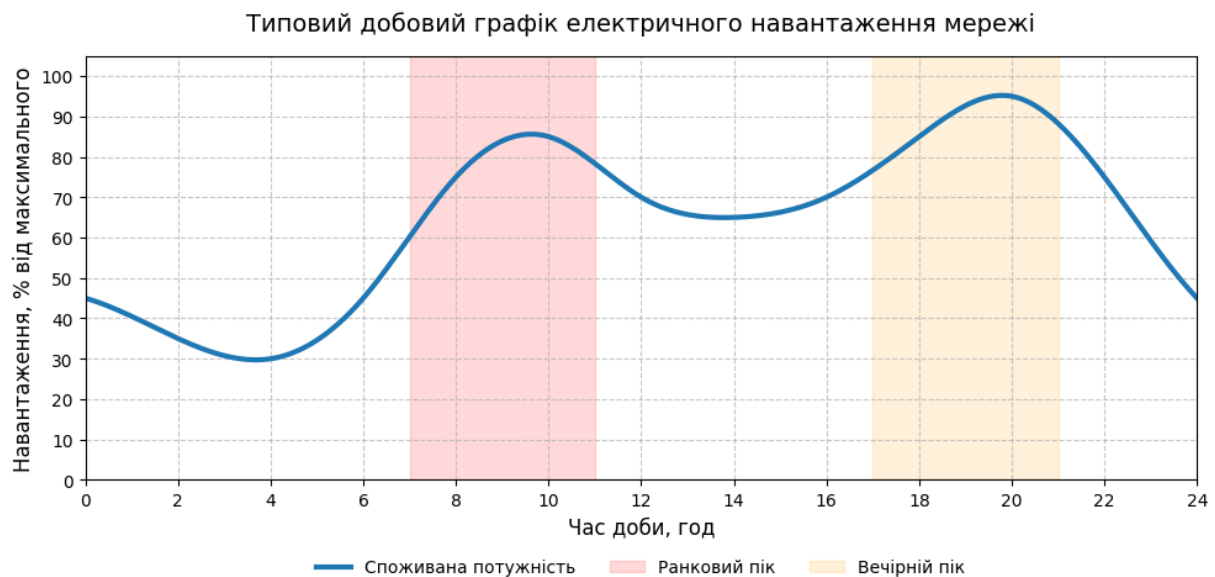


Рисунок 2.2 - Типовий добовий графік електричного навантаження з виділеними зонами пікового споживання

Якщо система керування не відреагує на ці явища вчасно, спрацює релейний захист. Цей захист є останнім, найжорсткішим рубежем оборони апаратної частини мережі. Він налаштований на миттєве розмикання силових контактів у разі перевищення критичних значень струму. Відповідно, для запобігання фізичному знищенню дороговартісного обладнання - плавленню ізоляції, перегорянню обмоток трансформаторів чи масштабним пожежам - автоматика просто відключить усю підстанцію цілком.

Проте в масштабах об'єднаної енергосистеми вимкнення одного великого транзитного вузла має катастрофічні наслідки: його навантаження миттєво

перерозподіляється на сусідні, ще працюючі підстанції. Ті, у свою чергу, також не витримують раптового стрибка струму і відключаються одна за одною. Таке ланцюгове, неконтрольоване падіння енергосистеми називається каскадним відключенням, або блекаутом. Наслідки блекауту завдають колосальних економічних збитків, оскільки процес "чорного старту", тобто поетапного відновлення нормальної роботи мережі з нульового стану, може тривати від кількох годин до кількох діб. Для вирішення цієї проблеми в сучасних інтелектуальних мережах застосовується метод адаптивного балансування або цілеспрямованого пріоритетного скидання зайвого навантаження. Класичні методи захисту минулих десятиліть діяли за принципом "сліпого" віялового відключення, коли від мережі примусово відрізалися цілі географічні райони або міста. Натомість суть інтелектуального адаптивного балансування полягає у точковому та апаратному керуванні потужністю. При фіксації перших ознак дефіциту потужності контролер не чекає аварійного спрацювання загального захисту. Замість цього він починає превентивно відключати окремі лінії та конкретних споживачів. Це дозволяє штучно знизити загальне споживання до безпечного рівня, зберігши при цьому цілісність магістральної мережі.

Щоб цей процес не завдав шкоди критичній інфраструктурі, усі споживачі жорстко сегментуються за рівнем важливості. Очевидно, що тимчасове знеструмлення побутових кондиціонерів у житловому кварталі та зупинка систем вентиляції у реанімаційному відділенні лікарні мають абсолютно різну ціну наслідків. Тому розробка ефективного алгоритму скидання навантаження вимагає точної класифікації підключених об'єктів. В рамках розробки нашої апаратної системи керування була створена практична трирівнева модель пріоритетизації. Такий підхід не перевантажує процесор зайвими даними, але водночас дозволяє гнучко та миттєво реагувати на різні стадії перевантаження - від легкого попереджувального лімітування до жорсткого аварійного відсікання. Детальні характеристики кожного рівня пріоритету та відповідні їм сценарії відключення наведено в таблиці 2.1.

					КвРКІ.223154.22.94.08 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 - Класифікація ліній живлення та критерії їх відключення в умовах дефіциту потужності

Категорія	Характеристика	Об'єкти підключення	Логіка ПЛС	Час відключення
I категорія	Безпека людей та інфраструктури	Сервери, медтехніка, пожежогасіння	Блокування відключення, утримання	Не допускається (0 с, ДБЖ)
II категорія	Загроза масового недовипуску продукції	Робочі станції, вузли зв'язку, освітлення	Скидання лінії при перевищенні	На час усунення перевантаження
III категорія	Споживачі поза I та II категоріями	Кліматична техніка, бойлери, декор	Скидання лінії при перевищенні	До 24 год

Огляд сучасних рішень у сфері Smart Grid показує, що більшість існуючих контролерів скидання навантаження реалізовані на базі класичних мікроконтролерів або промислових комп'ютерів. У таких пристроях логіка прийняття рішень закладена у вигляді програмного коду, написаного популярними мовами C або C++. Проте такий програмний підхід має суттєвий архітектурний недолік - неминучу технічну затримку реакції на подію. Мікроконтролер повинен послідовно зчитати інструкцію з пам'яті, дешифрувати її, опитати регістри датчиків струму, виконати арифметичне порівняння і лише потім надіслати сигнал на порт виводу, до якого підключено реле. Навіть на дуже високих тактових частотах цей процес постійно займає сотні машинних тактів.

В умовах, коли коротке замикання або різкий стрибок навантаження розвивається за лічені мілісекунди, така програмна затримка може виявитися фатальною для потужних силових транзисторів мережі. Саме тому сучасні практичні розробки в галузі енергетичної автоматики все частіше звертаються до прямих апаратних методів обробки сигналів на базі програмованої логіки. Розробка спеціалізованого обчислювального ядра дозволяє виконувати постійний моніторинг датчиків та генерацію керуючих сигналів повністю паралельно, буквально зі швидкістю поширення електричного струму по кристалу. Це повністю усуває поширену проблему програмних зависань та непередбачуваних системних переривань, гарантуючи абсолютно детермінований час реакції апаратури на будь-яку аварійну ситуацію.

## 2.2 Архітектурні переваги процесорів MIPS для побудови систем адаптивного керування

Проектування апаратних контролерів для інтелектуальних енергомереж висуває жорсткі вимоги до обчислювального ядра системи. Головними критеріями вибору процесорної архітектури у даному випадку є детермінованість часу виконання інструкцій, мінімальна технічна затримка реакції на зовнішні переривання та висока ефективність використання апаратних ресурсів кристала. Практика показує, що використання складних універсальних мікропроцесорів архітектури класу CISC, на зразок базової x86, є технічно та економічно недоцільним для задач низькорівневого керування силовими ключами.

Складні процесори мають інструкції змінної довжини, багатоступеневі системи кешування та механізми спекулятивного виконання коду. Ці технології ефективно підвищують середню продуктивність у мультимедійних задачах, проте роблять час виконання однієї конкретної команди абсолютно непередбачуваним. В умовах енергетичної аварії, коли час прийняття рішення

					КВРКІ.223154.22.94.08 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

вимірюється наносекундами, така непередбачуваність є критичним недоліком. З огляду на це, оптимальним практичним вибором для побудови системи адаптивного балансування навантаження є використання архітектури класу RISC. Серед них класична реалізація ядра MIPS демонструє найвищий рівень ефективності поєднання з програмованою логікою ПЛІС. Головною технічною перевагою MIPS є її гранична простота та строга ортогональність системи команд. Усі інструкції цього процесора мають фіксовану довжину - рівно 32 біти. Це означає, що первинному апаратному модулю вибірки команд не потрібно витратити зайві такти на визначення меж поточної команди у пам'яті. Дешифратор команд працює паралельно і майже миттєво визначає тип операції за першими шістьма бітами. Це дозволяє суттєво зменшити кількість використаних логічних вентилів та зекономити апаратні ресурси. Завдяки цій економії вивільнені блоки кристала можуть бути ефективно перенаправлені на побудову спеціалізованих схем моніторингу та високошвидкісних апаратних компараторів, які безпосередньо відповідають за постійне відслідковування аварійних порогів навантаження у розробленій системі. Організацію процесу послідовної обробки інформації за допомогою п'ятистадійного конвеєра для обраної архітектури MIPS представлено на рисунку 2.3.

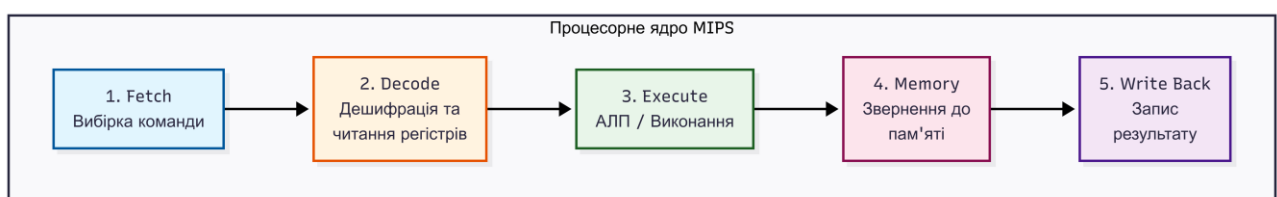


Рисунок 2.3 - Структурна схема 5-ступеневого конвеєра обробки даних архітектури MIPS

При розробці базової архітектури мікроконтролера ключову увагу було приділено питанню оптимізації швидкодії. Висока пропускна здатність процесора MIPS забезпечується безпосередньою реалізацією класичного

п'ятиступеневого конвеєра. Такий підхід базується на принципі часового та просторового паралелізму, де складний шлях проходження даних розділяється на дрібніші, строго синхронізовані етапи. Вибір саме п'ятиступеневої моделі обґрунтований необхідністю знайти оптимальний баланс між продуктивністю системи та кількістю апаратних ресурсів, які витрачаються на створення конвеєрних регістрів. Відповідно до класичної методології побудови процесорних ядер RISC-архітектури, кожна інструкція проходить через п'ять незалежних апаратних блоків.

Кожна інструкція проходить через п'ять незалежних апаратних блоків:

1. Вибірка інструкції - зчитування 32-бітного слова з пам'яті програм за адресою поточного лічильника.
2. Дешифрація - розпізнавання команди та одночасне зчитування операндів з регістрового файлу.
3. Виконання - проведення математичних або логічних операцій в АЛП (арифметико-логічному пристрої).
4. Звернення до пам'яті - читання або запис даних у пам'ять.
5. Запис результату - збереження отриманого результату назад у внутрішній регістр.

Такий поділ дозволяє обчислювальній системі на кожному такті системного генератора видавати готовий результат виконання однієї машинної команди після завершення початкового етапу заповнення апаратного конвеєра. Для обраної архітектури ПЛІС типу Cyclone V, де швидкість поширення електричного сигналу по внутрішніх логічних ланцюгах кристала має певні фізичні обмеження, розбиття тривалої та складної операції на п'ять простих послідовних етапів дозволяє суттєво підвищити максимальну тактову частоту роботи розробленого процесорного ядра. Ще однією важливою архітектурною характеристикою MIPS, яка робить її оптимальним вибором для реалізації даного практичного проекту контролера, є побудова за принципом завантаження-збереження. Це означає, що всі арифметичні та логічні операції,

включаючи порівняння поточного значення струму з аварійним лімітом, можуть виконуватися виключно всередині мікропроцесора над операндами, які вже попередньо завантажені у внутрішній регістровий файл. Процесорне ядро ніколи не виконує обчислювальних операцій безпосередньо з даними, що знаходяться в оперативній пам'яті. Для обміну інформацією із зовнішніми периферійними пристроями використовуються лише дві спеціалізовані команди, які відповідають за завантаження в регістр та вивантаження з регістра, що суттєво спрощує загальну логіку керування внутрішніми та зовнішніми шинами даних на друкованій платі. У контексті безпосередньої розробки системи енергозбереження, керуючий алгоритм мікроконтролера є повністю лінійним і не потребує впровадження складних математичних перетворень із рухомою комою.

Базова логіка функціонування пристрою зводиться до періодичного зчитування поточного стану підключених датчиків, порівняння отриманого цифрового значення із заданими порогами спрацювання та подальшої активації відповідних вихідних силових контактів або реле. Для успішної реалізації такого алгоритму керування достатньо апаратно відтворити на рівні логічних вентилів лише мінімальну базову підмножину цілочисельних команд MIPS. Таке інженерне рішення дозволяє зекономити до 80% цифрових логічних елементів кристала ПЛІС порівняно з інтеграцією повноцінного універсального процесора. Збережений логічний простір відкриває можливості для зручного масштабування апаратної платформи, зокрема додавання нових комунікаційних інтерфейсів. Крім того, відмова від надлишкових блоків знижує власне енергоспоживання розробленого пристрою. Це також позитивно впливає на загальний тепловий режим роботи мікросхеми у закритих промислових корпусах. Базовий набір необхідних інструкцій та їх функціональне призначення у контексті керування підключеним навантаженням наведено в таблиці 2.2.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		



проєкту, дозволяючи досягти оптимальних часових затримок між апаратними регістрами.

Повна відсутність складних внутрішніх блоків, таких як багаторівневі системи передбачення переходів або модулі обробки виняткових ситуацій, гарантує абсолютну лінійність виконання алгоритму. Завдяки такій глибокій оптимізації від моменту зчитування критичного значення на вимірювальному датчику до моменту фізичного формування керуючого сигналу на відключення силового реле проходить рівно три машинних такти. Враховуючи частоту роботи сучасних генераторів тактових імпульсів, цей час вимірюється частками мікросекунди, що забезпечує практично миттєву реакцію захисної автоматики на будь-які аварійні стрибки струму. Подібна жорстка детермінованість та блискавична швидкодія є абсолютно недосяжними для стандартних мікроконтролерів, робота яких обтяжена циклічним опитуванням периферії, операційними системами реального часу та програмними перериваннями. Суто апаратна реалізація повністю унеможливорює програмні зависання системи в критичний момент розвитку аварії. Викладені технічні характеристики підтверджують доцільність вибору вказаної архітектури як найбільш ефективного, надійного та відмовостійкого інструменту для вирішення поставленої задачі інтелектуального керування навантаженням.

### 2.3 Аналіз апаратної бази: особливості кристалів сімейства Intel Cyclone V та макетної плати DE1-SoC

Для фізичної реалізації розробленої архітектури процесорного ядра було обрано сучасну апаратну платформу на базі програмованих логічних інтегральних схем сімейства Cyclone V від компанії Intel. Вибір саме цього сімейства зумовлений оптимальним співвідношенням продуктивності, енергоспоживання та вартості, що робить його стандартом для проєктування вбудованих систем керування промислового рівня. Деталізовану внутрішню

					КвРКІ.223154.22.94.08 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		



зовнішніх мікросхем оперативної пам'яті неминуче призводить до виникнення фізичних затримок при передачі сигналів через доріжки друкованої плати, що є абсолютно неприпустимим для систем аварійного захисту. Саме тому мікросхеми сімейства Cyclone V містять інтегровані безпосередньо в кристал апаратні блоки статичної пам'яті типу M10K. Кожен такий спеціалізований масив побудований на базі високошвидкісних транзисторних комірок, здатен зберігати 10 кілобіт інформації та може гнучко конфігуруватися розробником як однопортовий або двопортовий запам'ятовуючий пристрій. Ця апаратна гнучкість відкриває широкі можливості для ефективного керування інформаційними потоками всередині обчислювального ядра.

Для обраної нами архітектури MIPS ці ресурси виступають ідеальним базисом для фізичної реалізації класичної розділеної пам'яті гарвардського типу. На відміну від традиційної архітектури фон Неймана, де команди та змінні змушені постійно конкурувати за єдину магістраль передачі, наш підхід передбачає повне апаратне ізолювання цих потоків. Один масив блоків M10K буде жорстко виділено виключно під пам'ять інструкцій, де надійно зберігатиметься алгоритм моніторингу енергомережі, а інший незалежний масив - під тимчасову робочу пам'ять даних для фіксації поточних станів датчиків. Така топологія повністю усуває проблему "вузького місця" загальної шини і дозволяє процесорному конвеєру гарантовано зчитувати нову команду та маніпулювати даними паралельно, рівно за один такт системного генератора. Це максимізує загальну пропускну здатність контролера та забезпечує детермінованість обчислень. Безпосереднє макетування та тестування розробленої системи адаптивного балансування навантаження виконувалося на спеціалізованій налагоджувальній платі Terasic DE1-SoC. Загальний вигляд використаної у роботі налагоджувальної платформи із зазначенням ключових інтерфейсів та периферійних модулів представлено на рисунку 2.5. Серцем даної платформи є мікросхема з маркуванням 5CSEMA5F31C6. Цей кристал належить до класу гібридних систем і фізично поєднує в одному корпусі два незалежні домени:

					КВРКІ.223154.22.94.08 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		





ланцюги для апаратного захисту від брязкоту контактів. Це дозволяє подавати чистий та стабільний логічний сигнал безпосередньо на входи нашого процесора, усуваючи необхідність розробляти додаткові програмні фільтри перешкод.

Роль виконавчих пристроїв, тобто силових реле, які фізично відключають споживачів від живлення, виконують яскраві світлодіодні індикатори. Усі зовнішні інтерфейси плати чітко узгоджені за електричними рівнями з використанням стандарту безпечної напруги 3.3-V LVTTTL. Такий підхід забезпечує високу завадостійкість керуючих сигналів та гарантує збереження кристала під час апаратного тестування.

#### 2.4 Автоматизоване проектування цифрових пристроїв у середовищі САПР Quartus Prime

Перетворення високорівневого опису процесорного ядра на фізичну конфігурацію транзисторів кристала є надзвичайно складним обчислювальним процесом. Для обраного нами сімейства мікросхем цей маршрут проектування реалізується за допомогою спеціалізованої системи автоматизованого проектування інтегральних схем Intel Quartus Prime. Дане програмне середовище забезпечує повний цикл розробки: від написання початкового коду до генерації прошивного файлу та апаратного програмування макетної плати. Типовий алгоритм розробки цифрової системи в середовищі Quartus Prime, що охоплює всі етапи від написання вихідного коду до прошивки кристала, представлено у вигляді блок-схеми на рисунку 2.6. Процес побудови цифрового пристрою у цьому середовищі складається з кількох суворо послідовних етапів, кожен з яких виконує детальну структурну та топологічну оптимізацію розробленої логіки. Першим базовим етапом є перевірка та елаборація. На цій стадії вбудований компілятор перевіряє синтаксис написаного коду мовою апаратного опису та будує первинну ієрархію модулів системи. Програма розпізнає структурні елементи коду та інтерпретує їх у базові цифрові примітиви. Наприклад, логічні

умови та безперервні присвоєння розпізнаються як комбінаційна логіка арифметико-логічного пристрою, а процедурні блоки, що спрацьовують по фронту тактового сигналу, перетворюються на синхронні тригери, формуючи регістр лічильника команд. Після успішної перевірки автоматично запускається етап логічної компіляції. Його завдання - мінімізація булевих функцій. Компілятор детально розглядає логіку компараторів, які порівнюють вхідне навантаження з аварійними лімітами, і намагається реалізувати їх з використанням мінімальної можливої кількості стандартних логічних вентилів. Далі відбувається процес технологічного відображення: отримані базові елементи оптимально упаковуються у специфічні для нашої мікросхеми адаптивні логічні модулі.

Наступним, найбільш тривалим за часом етапом компіляції є розміщення та трасування. Програма бере оптимізовані логічні модулі та автоматично розміщує їх на двовимірній фізичній матриці кристала. Саме на цьому етапі система звертається до налаштувань прив'язки контактів. Вбудований трасувальник прокладає фізичні провідники всередині мікросхеми таким чином, щоб з'єднати входи нашого компаратора з фізичними ніжками мікросхеми, до яких підпаяні перемикачі плати, а виходи регістрів - з ніжками керування світлодіодами. Варто відзначити надзвичайно важливий інженерний аспект безпеки, який реалізується саме на етапі трасування. Мікросхема має сотні зовнішніх контактів, тоді як у нашому практичному проєкті використовується лише базова необхідна периферія. Щоб уникнути небезпечного короткого замикання на невикористаних контактах налагоджувальної плати, у глобальних налаштуваннях компілятора застосовується правило безпеки контактів. Відповідно до нього, усі вільні ніжки примусово переводяться у стан високоімпедансного входу. Це стовідсотково гарантує, що жоден випадковий логічний сигнал не виведе апаратну платформу з ладу під час подачі живлення.

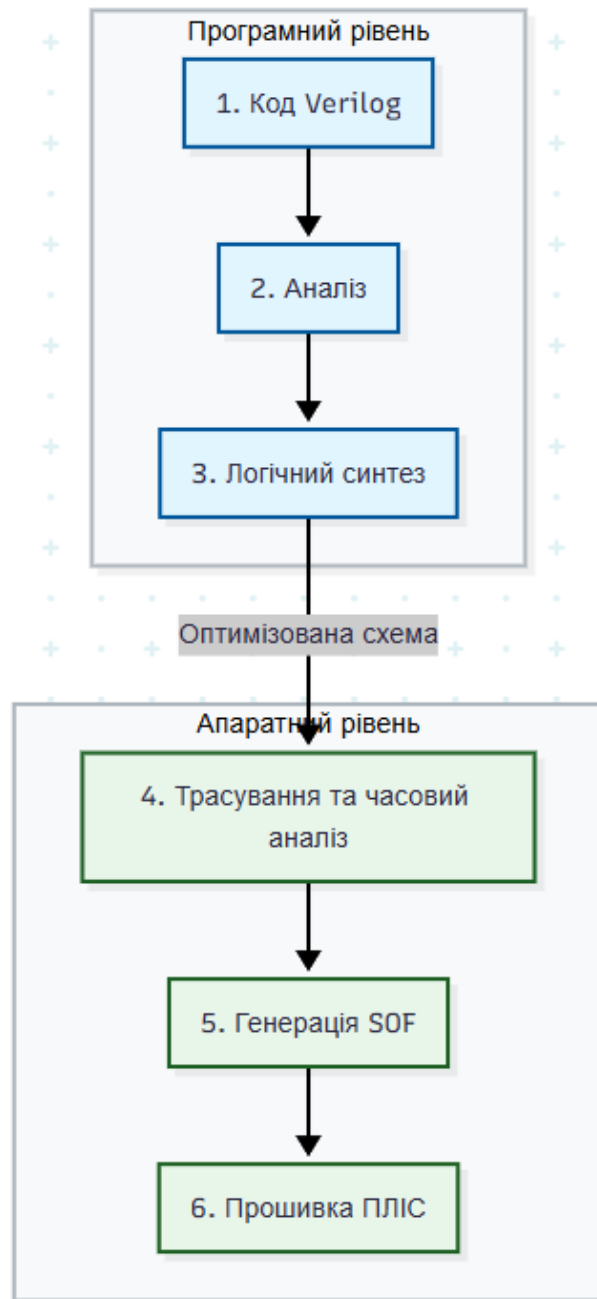


Рисунок 2.6 - Блок-схема маршруту проектування цифрового пристрою у середовищі САПР

Після завершення прокладання маршрутів система обов'язково запускає модуль перевірки часових параметрів. Оскільки ядро нашого процесора працює від апаратного генератора плати з частотою 50 мегагерц, період одного системного такту становить рівно 20 наносекунд. Блок перевірки детально вираховує фізичні затримки поширення електронних сигналів по щойно

прокладених провідниках на кристалі. Його ключове завдання - технічно підтвердити, що сигнал від сенсорів струму встигне пройти через логіку порівняння і дійти до вихідного регістра швидше, ніж настане наступний фронт тактового імпульсу.

Фінальним кроком роботи програмного середовища є запуск асемблера. Цей модуль генерує кінцевий конфігураційний файл, який містить суцільну бітову послідовність. Даний масив через вбудований програматор завантажується у конфігураційну пам'ять плати. Після завершення завантаження логічна матриця миттєво перебудовується на фізичному рівні, і стандартний кристал перетворюється на наш спеціалізований процесор керування енергозбереженням, повністю готовий до проведення практичних випробувань.

## 2.5 Висновки до розділу 2

Виконаний практичний огляд проблематики та засобів проектування дозволяє зробити низку важливих висновків для подальшої практичної реалізації системи. Встановлено, що масовий перехід до інтелектуальних енергомереж вимагає невідкладного впровадження високошвидкісних систем адаптивного балансування навантаження. Для ефективного запобігання каскадним відключенням та системним аваріям захисна автоматика повинна реагувати на критичні стрибки струму у режимі суворого реального часу, миттєво відсікаючи другорядних споживачів. З'ясовано, що використання класичних програмних мікроконтролерів є неефективним через неминучі затримки на обробку інструкцій та операційні переривання. Оптимальним апаратним рішенням для розробки контролера визначено використання спеціалізованого обчислювального ядра на базі класичної архітектури MIPS. Відмова від надлишкових внутрішніх процесорних блоків та використання виключно базової підмножини інструкцій дозволяє забезпечити абсолютно детермінований час реакції системи на рівні кількох машинних тактів. Визначено, що найбільш

					КВРКІ.223154.22.94.08 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

технологічною платформою для фізичної реалізації розробленого процесорного ядра є сучасні програмовані логічні інтегральні схеми. Обрана для макетування налагоджувальна плата оснащена необхідним набором апаратної периферії, включаючи електричний захист від брязкоту контактів та швидкісні блоки пам'яті, що дозволяє повноцінно імітувати роботу енергомережі. Розглянуто та описано повний маршрут автоматизованого проєктування цифрових пристроїв у профільному програмному середовищі. Визначено, що загальна ефективність кінцевого рішення критично залежить від правильності базових налаштувань компілятора на кожній стадії обробки проєкту. Суворе дотримання технологічних етапів початкової логічної компіляції, автоматичного апаратного трасування та точного часового контролю гарантує безпечне та технічно безпомилкове перетворення високорівневого програмного коду у надійну фізичну конфігурацію транзисторів на кристалі. Це, у свою чергу, повністю нівелює ризики виникнення апаратних збоїв та непередбачуваних часових колізій під час реальних стрибків струму в енергомережі.

Таким чином, детально сформована практична, інженерна та інструментальна база є міцним фундаментом і цілком підготовлена для логічного переходу до етапу безпосереднього написання коду мовою опису апаратури, його апаратної перевірки та фізичного тестування розробленої процесорної системи в умовах, максимально наближених до справжніх сценаріїв балансування навантаження.

					КвРКІ.223154.22.94.08 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ КЕРУВАННЯ ЖИВЛЕННЯ ЕЛЕКТРОМЕРЕЖІ

#### 3.1 Логічне проектування та розгляд коду обчислювального ядра мовою Verilog HDL

Першим і найважливішим етапом практичної розробки системи адаптивного балансування навантаження є створення її детальної логічної моделі на рівні регістрових передач. Для формалізованого опису архітектури процесорного ядра було застосовано мову апаратного проектування стандарту Verilog, яка дозволяє максимально точно описати поведінку цифрових ланцюгів. Вибір цієї мови обумовлений можливістю прямого відображення програмних конструкцій у реальну логіку кристала, що критично важливо для забезпечення швидкодії системи у реальному часі. Процес безпосередньої розробки починається з головного модуля, який виступає верхнім рівнем всього проекту, та визначення базових керуючих сигналів. Ці сигнали забезпечують стабільний запуск та синхронізацію всіх внутрішніх процесів обчислювального ядра. Системний тактовий сигнал, що задає темп роботи конвеєра, та сигнал апаратного скидання, який гарантує переведення системи у початковий безпечний стан, визначаються наступним чином:

```
module mips_smart_grid (  
    input clk,  
    input reset,
```

Ці два входи відповідають за глобальну апаратну синхронізацію. Сигнал clk тактує всю систему, а reset дозволяє примусово повернути її в початковий стан. Далі вказуються основні шини передачі даних:

```
    input [9:0] load_sensor,  
    output reg [2:0] power_lines );
```

Реальне розташування згаданих елементів керування (перемикачів) та індикації на використовуваній макетній платі під час виконання практичної частини роботи продемонстровано на рисунку 3.1. Десятирозрядна шина

load\_sensor безпосередньо зчитує положення десяти повзункових перемикачів на платі, що в реальному часі імітують рівень навантаження. Вихідний трирозрядний порт power\_lines керує трьома світлодіодами, які виконують роль силових реле для комутації ліній.

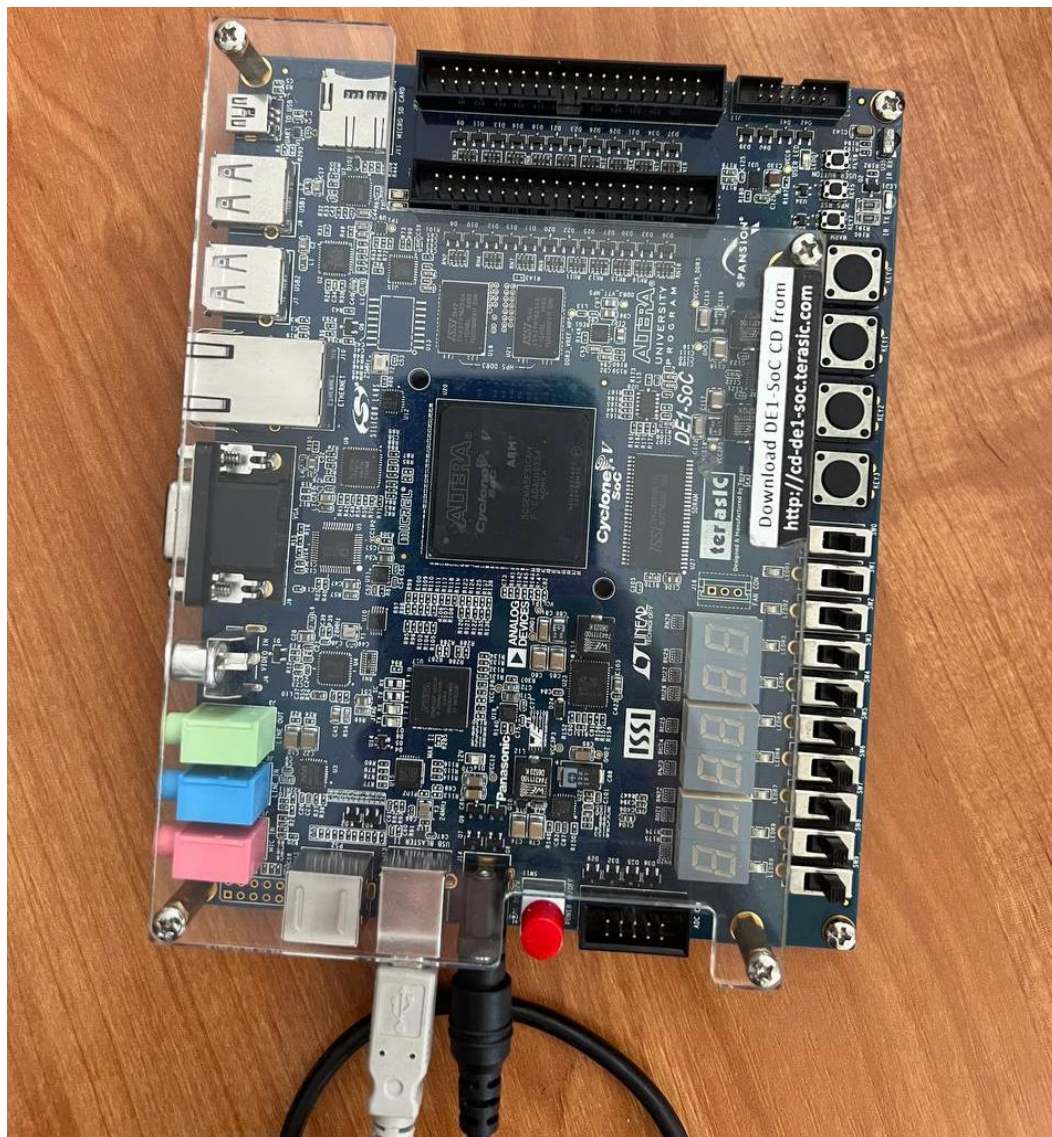


Рисунок 3.1 - Загальний вигляд макетної плати DE1-SoC з блоком входних перемикачів та індикаторів

Зм.	Арк.	№ докум.	Підпис	Дата

Для забезпечення гнучкості контролера порогові значення навантаження винесені в окремі константи безпосередньо в код. Задається перший поріг попереджувального відключення та другий критичний поріг:

```
parameter LIMIT_WARNING = 10'd300;  
parameter LIMIT_CRITICAL = 10'd700;
```

Такий підхід дозволяє швидко переналаштовувати алгоритм під різні типи підстанцій без зміни базової логіки. Встановлені десяткові значення 300 та 700 відповідають комбінаціям увімкнених перемикачів на вхідній шині. Сама логіка прийняття рішень реалізована апаратним компаратором:

```
assign alu_result = (load_sensor > LIMIT_CRITICAL) ? 32'b001 :  
                   (load_sensor > LIMIT_WARNING) 32'b011 :  
                   32'b111 ;
```

Цей вираз створює каскад цифрових компараторів. Залежно від вхідного рівня навантаження, система миттєво генерує відповідне керуюче слово. Коди "001" та "011" означають аварійне відключення певних ліній, а "111" дозволяє роботу всіх трьох об'єктів. Зовнішній вигляд блоку механічних перемикачів та кнопок, що використовуються для оперативного керування параметрами логіки ядра та імітації рівнів навантаження, наведено на рисунку 3.2.



Рисунок 3.2 - Блок апаратних перемикачів та тактових кнопок для керування логікою ядра

Синхронізація роботи та запис результатів відбувається у послідовнісному блоці. При натисканні фізичної кнопки скидання система переходить у безпечний режим, обнуляючи лічильник та вмикаючи всі лінії:

```
if (!reset) begin
    pc <= 32'd0;
    power_lines <= 3'b111;
```

В цьому стані на всі вихідні лінії подається логічна одиниця, що гарантує збереження живлення споживачів до початку роботи алгоритму. У звичайному штатному циклі на кожному такті генератора виконуються базові дії переходу до наступної команди та запису рішення на світлодіоди:

```
end else begin
    pc <= pc + 4;
    power_lines <= alu_result[2:0];
```

Значення лічильника команд у кожному робочому циклі автоматично збільшується рівно на 4 байти, що є фундаментальною вимогою для стандартних 32-бітних процесорних архітектур. Такий крок адресації обумовлений тим, що кожна інструкція в пам'яті займає рівно чотири послідовні байти, і для коректного переходу до наступного кроку алгоритму вказівник має переміщуватися строго на довжину слова. Даний механізм інкременту реалізований на апаратному рівні, що дозволяє уникнути додаткових програмних витрат та забезпечує максимальну детермінованість часових характеристик виконання коду. Така особливість гарантує лінійність виконання алгоритму та виключає можливість помилкового зчитування частини інструкції, що могло б призвести до критичного збою в логіці керування енергомережею. Суворе дотримання вирівнювання адрес пам'яті є критичним фактором, оскільки будь-яке відхилення від цього принципу може спричинити виникнення апаратних виключень, що неприпустимо для систем, які працюють у режимі реального часу.

Паралельно з цим процесом, згенероване апаратним компаратором рішення миттєво та без жодних проміжних затримок фіксується на вихідних

					КвРКІ.223154.22.94.08 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

фізичних контактах мікросхеми, до яких підключено силові лінії. Така висока швидкість реакції досягається завдяки оптимізації внутрішніх логічних шляхів всередині кристала ПЛІС, що мінімізує вплив паразитних ємностей та забезпечує чіткі фронти керуючих сигналів. Використання прямого апаратного зв'язку між обчислювальним ядром та периферійними інтерфейсами гарантує мінімальний рівень джитера, що є вагомим перевагою при реалізації алгоритмів захисту та автоматики в енергетичних комплексах. Візуалізацію процесу відпрацювання алгоритму захисного відключення на фізичному рівні, що супроводжується активацією відповідної світлодіодної індикації на макетній платі, наведено на рисунку 3.3.

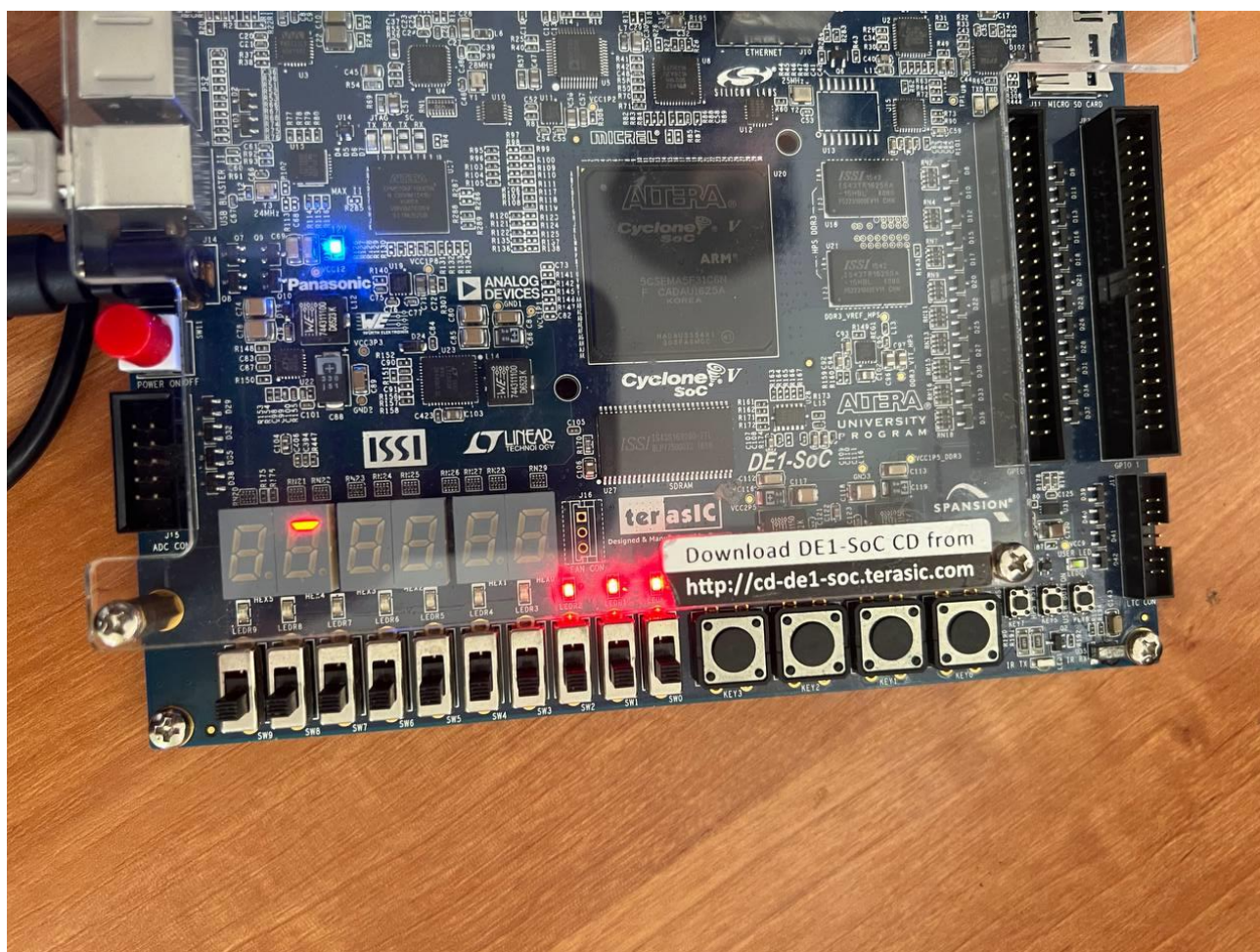


Рисунок 3.3 - Видгляд макетної плати під час практичного відпрацювання алгоритму перемикання вихідних ліній

Важливою практичною перевагою розробленої моделі є те, що вона повністю позбавлена будь-яких програмних циклів очікування або умовних переходів, які зазвичай створюють непередбачувані затримки у роботі звичайних мікроконтролерів. У класичному програмному коді перевірка датчиків часто виконується всередині циклічних конструкцій, що витрачає ресурси процесора на службові операції переходу. У розробленій апаратній реалізації кожна логічна операція виконується паралельно на рівні логічних елементів. Це гарантує абсолютно детерміновану видачу сигналів керування, де час відгуку системи на аварійну подію завжди залишається незмінним і становить рівно один такт системного генератора.

### 3.2 Конфігурація фізичних інтерфейсів та призначення виводів ПЛІС

Після успішного завершення опису внутрішньої логіки обчислювального ядра на мові Verilog, наступним критично важливим етапом проекту є процедура фізичної адаптації розробленої системи до конкретної апаратної архітектури мікросхеми Cyclone V. Оскільки обрана нами ПЛІС має сотні незалежних контактів вводу-виводу, необхідно чітко вказати інтегрованому середовищу проектування Quartus Prime, які саме фізичні ніжки кристала відповідають за вхідні сенсори та вихідні лінії керування навантаженням. Без цього етапу компілятор не зможе коректно прокласти внутрішні маршрути сигналів, що зробить роботу пристрою на реальній платі неможливою. Процес прив'язки виводів починається з попереднього аналізу апаратної схеми налагоджувальної плати DE1-SoC. Було виконано зіставлення назви портів у нашому коді з іменами фізичних ліній, що розведені на друкованій платі від мікросхеми до перемикачів та світлодіодів. Для виконання цього завдання було використано спеціалізований графічний інструмент Pin Planner. Даний інструмент дозволяє не просто призначити адресу контакту, а й візуально контролювати щільність розміщення сигналів на корпусі мікросхеми, що допомагає уникнути перегріву окремих

банків вводу-виводу при роботі на високих частотах. Зазначена процедура відображення базується на принципі прямого копіювання програмних портів модуля `mips_smart_grid` на фізичні ресурси макетної плати. Даний етап є критично важливим кроком у маршруті проєктування, оскільки він забезпечує фізичну реалізацію абстрактної логічної моделі всередині кристала ПЛІС. Кожен окремих вхідний сигнал десятибітної шини `load_sensor` було закріплено за відповідним механічним повзунковим перемикачем типу SW. Це дозволяє нам фізично змінювати вхідний двійковий код імітації споживання енергомережі безпосередньо під час процесу тестування, забезпечуючи наочність відпрацювання алгоритмів у динамічному режимі. Окрім основних ліній, особлива увага була приділена тактовому сигналу `clk`, який підключено до прецизійного кварцового резонатора на 50 МГц, що забезпечує стабільність часових характеристик нашої системи.

Важливим технологічним аспектом є вибір стандарту електричної напруги для контактів. Для нашого проєкту для всіх задіяних ліній було примусово встановлено стандарт 3.3-V LVTTTL. Це забезпечує повну електричну сумісність логічних рівнів нашого процесорного ядра з периферійними пристроями, вбудованими на платі. Завдяки суворому дотриманню рівнів напруги вдається досягти оптимального енергоспоживання вводу-виводу при збереженні необхідної крутизни фронтів керуючих імпульсів. Вибір цього стандарту гарантує високу завадостійкість сигналів керування та чітке розпізнавання логічного нуля та одиниці, що виключає хибні спрацювання захисної автоматики через наведення електромагнітних завад. Це особливо важливо в умовах промислової експлуатації, де високий рівень індустріальних шумів може негативно впливати на цілісність даних у цифрових шинах керування.

У таблиці 3.1 наведено розширені зведені дані відповідності програмних портів та фізичних виводів мікросхеми, яка була сформована на основі технічної документації до плати та враховує специфіку топології друкованої плати DE1-

					КВРКІ.223154.22.94.08 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		





передачі цифрових сигналів, які керують роботою світлодіодних індикаторів на макетній платформі DE1-SoC. Додатковим інженерним етапом під час налаштування утиліти була перевірка режимів сумісності для вільних фізичних контактів, які не задіяні в поточній версії обчислювальної системи. Усі невикористані виводи були програмно переведені компілятором у захисний режим з метою виключення випадкових замикань на платі. Таке комплексне апаратне конфігурування гарантує максимальну стабільність функціонування розробленого пристрою автоматики та дозволяє перейти до генерації фінального бітового потоку прошивки. Отримані результати успішного закріплення зовнішніх інтерфейсів зафіксовано на рисунку 3.4.

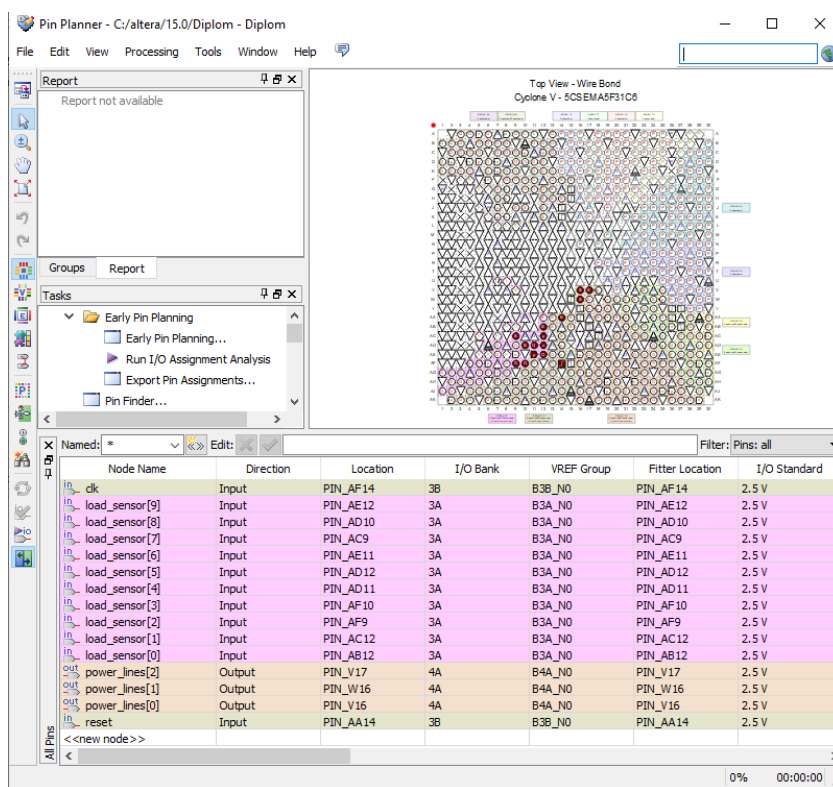


Рисунок 3.4 - Сформований список призначень фізичних контактів у середовищі Quartus

Чітко відображено повний перелік логічних вузлів із зазначенням їхнього напрямку та закріплених за ними фізичних локацій на кристалі. Особливу увагу

при налаштуванні було приділено параметрам електричного стандарту вводу-виводу. Важливо зазначити, що після ручного призначення кожного піна система автоматично оновлює внутрішній конфігураційний файл проєкту з розширенням .qsf. Цей текстовий документ з налаштуваннями конфігурації було додатково перевірено розробником, щоб переконатися, що всі директиви призначення локацій повністю відповідають проєктним даним. Окремим інженерним завданням на цьому етапі було правильне налаштування поведінки невикористаних виводів платформи. Оскільки мікросхема Cyclone V має 484 контакти у корпусі FPGA, а в нашій розробці задіяно лише 15, виникає ризик пошкодження кристала через статичну електрику або випадкові замикання на вільних ніжках. Через спеціалізоване меню конфігурації пристрою було встановлено режим переведення вільних виводів у стан входу з високим імпедансом та слабкою підтяжкою до лінії живлення. Тобто, таке технічне рішення примусово переводить усі вільні ніжки у безпечний стан. Даний підхід стовідсотково гарантує безпеку апаратної платформи під час процесу практичних випробувань та повністю виключає виникнення паразитних струмів всередині логічної матриці ПЛІС.

Після остаточної фіксації всіх параметрів та проведення візуальної перевірки топології розміщення сигналів, система стає повністю готовою до запуску циклу повної компіляції. Це дозволяє перейти до наступного етапу - детального аналізу використання внутрішніх ресурсів ПЛІС та перевірки часових затримок, що є вирішальним для підтвердження працездатності системи у жорсткому реальному часі.

### 3.3 Аналіз апаратних ресурсів та результатів повної компіляції проєкту

Цифрові показники, наведені у підсумковому звіті, є результатом глибокої багаторівневої оптимізації. Під час фази логічної компіляції компілятор не просто копіює написаний нами код, а виконує перевірку логічних зв'язків,

					КвРКІ.223154.22.94.08 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		



Для візуального підтвердження успішного завершення всіх технологічних стадій проєктування наведено офіційну копію звіту конфігурації з робочого середовища Quartus Prime на рисунку 3.5. Даний звіт є фінальним консолідованим документом, який автоматично формується системою після проходження повного циклу - від початкового аналізу та компіляції до фінального трасування та генерації бітового файлу прошивки. Наявність цього звіту без критичних помилок свідчить про повну валідність розробленої архітектури та її відповідність фізичним обмеженням обраного кристала Cyclone V. Окрім підтвердження працездатності проєкту, підсумковий звіт надає вичерпну інформацію щодо утилізації апаратних ресурсів мікросхеми. Зокрема, вивчення цих даних дозволяє оцінити відсоток використання адаптивних логічних модулів ALM, блоків вбудованої пам'яті M10K та ліній вводу-виводу. Це дає змогу розробнику зробити висновок про ефективність написаного Verilog-коду та наявність вільного місця на кристалі для подальшого нарощування функціональних можливостей процесорного ядра або додавання нових модулів моніторингу енергомережі в майбутньому. Таким чином, представлені результати є документальним доказом того, що проєкт повністю адаптований під конкретну апаратну базу та готовий до практичної експлуатації.

Додатковим інженерним фактором, який підтверджується сформованим звітом, є успішне виконання часових специфікацій для розробленого обчислювального ядра. Програма автоматично перевіряє часові затримки на критичних шляхах поширення сигналу, гарантуючи відсутність порушень умов встановлення та утримання даних у внутрішніх тригерах. Це підтверджує, що частота роботи апаратного генератора 50 МГц повністю відповідає фізичним можливостям сформованої топології логічних вентилів і не призведе до появи збоїв або метастабільних станів під час динамічної зміни вхідних сигналів з перемикачів макетної плати. Згенерований у процесі компіляції вихідний файл містить повну структуру конфігураційного коду, яка завантажується безпосередньо у внутрішню енергонезалежну пам'ять налагоджувальної

					КВРКІ.223154.22.94.08 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		



Кожен такий модуль містить у собі комбінаційні таблиці перекодувань, два швидких суматори та набір програмованих тригерів. Оскільки логіка нашого компаратора є відносно простою, а саме порівняння 10-бітного числа з двома константами, компілятору вдалося упакувати всю обчислювальну логіку в мінімальну кількість фізичних вузлів. Це не лише звільняє місце на кристалі, але й критично знижує динамічне енергоспоживання пристрою, оскільки сигнал проходить через мінімальну кількість внутрішніх логічних переходів. Відсутність задіяних блоків вбудованої пам'яті та блоків цифрової обробки сигналів є свідомим інженерним рішенням. У рамках даного проєкту вся необхідна інформація про стани датчиків та алгоритми реагування обробляється безпосередньо у комбінаційній логіці та зберігається у внутрішніх регістрах процесорного ядра. Це усуває необхідність звернення до повільніших масивів блочної пам'яті, що забезпечує ту саму блискавичну швидкодію, яка була задекларована у теоретичній частині роботи.

Нульове використання фазового автопідстроювання частоти також підкреслює надійність системи. Пряме використання тактового сигналу 50 МГц без додаткового множення або ділення частоти дозволяє уникнути виникнення фазового шуму та забезпечує максимальну стабільність синхронізації всього конвеєра даних. Таким чином, отриманий апаратний конфігураційний файл є ідеальним базисом для створення відмовостійких систем енергомоніторингу, де простота архітектури є прямою запорукою її безвідмовності. Одним важливим аспектом оцінки апаратних ресурсів є вивчення топології розробленого ядра на рівні логічних шлюзів та елементарних вузлів обробки даних. Після успішного завершення стадії логічної компіляції середовище Quartus Prime дозволяє згенерувати детальну технологічну схему проєкту на рівні регістрових передач. Дана RTL схема виступає критично важливою проміжною ланкою між абстрактним високорівневим описом мовою Verilog та безпосереднім фізичним розміщенням напівпровідникових структур у матриці кристала ПЛІС. Сформована графічна інтерпретація архітектури процесорного ядра

mips\_smart\_grid, що відображає ієрархію внутрішніх зв'язків, представлена на рисунку 3.6.

Використання інструменту відображення RTL дозволяє виконати візуальну перевірку створеної архітектури ще до моменту завантаження прошивки в залізо. Вона наочно демонструє, яким чином програмні логічні оператори порівняння, вибору та присвоєння були інтерпретовані компілятором і перетворилися на конкретні апаратні вузли, а саме цифрові дешифратори, блоки мультиплексування для вибору даних, суматори та тригерні масиви для зберігання станів. Такий підхід дає можливість переконатися, що описана в коді логіка не містить прихованих циклів або конфліктних з'єднань, які могли б призвести до нестабільної роботи системи.

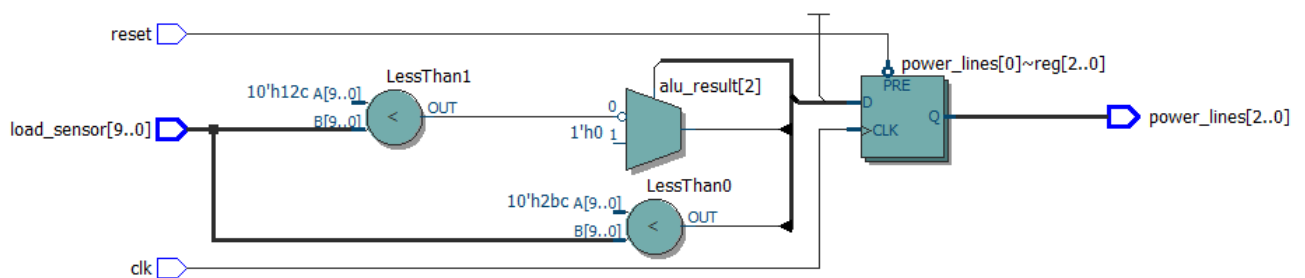


Рисунок 3.6 - Структурна схема апаратної логіки ядра mips\_smart\_grid у поданні RTL Viewer

На представленому рисунку можна чітко простежити шлях проходження сигналу від вхідної шини load\_sensor до блоку прийняття рішень. Ми бачимо, що система автоматично згенерувала дерево компараторів, які працюють паралельно. Це підтверджує нашу тезу про детермінованість обчислень: сигнал не чекає своєї черги на обробку в процесорному стеку, а фізично проходить крізь логічну сітку. Кожен вихідний сигнал порту power\_lines має власний виділений тракт, що повністю виключає можливість виникнення програмних конфліктів або зависань під час інтенсивних стрибків струму в енергомережі. Розглядаючи

детальну структуру використаних адаптивних логічних модулів, варто зазначити, що Quartus Prime застосував стратегію щільної упаковки. Це означає, що кілька логічних функцій, які не мають прямого зв'язку, можуть бути розміщені в одному фізичному модулі для мінімізації довжини з'єднувальних ліній. Це критично важливо для зменшення динамічних затримок поширення сигналу. Навіть на частоті 50 МГц, яка здається невисокою для сучасної електроніки, на мікрорівні кристала кожен зайвий міліметр провідника додає пікосекунди затримки. Завдяки компактності нашого ядра, ці затримки є нікчемними, що забезпечує колосальний запас часового стійкості системи.

Додатково було виконано розрахунок підсумкового звіту про енергоспоживання через спеціалізований аналізатор потужності. Хоча наш пристрій живиться від стандартної шини USB налагоджувальної плати, мінімізація використаних модулів безпосередньо впливає на тепловиділення кристала. Оскільки в алгоритмі задіяно лише декілька десятків логічних комірок, статичне та динамічне споживання потужності самим ядром становить лічені мілівати. Це робить таку архітектуру ідеальною для вбудованих автономних систем моніторингу, які можуть тривалий час працювати від резервних джерел безперебійного живлення під час повного блекауту основної мережі. Таким чином, виконане детальне оцінювання підтверджує, що обрана апаратна платформа Cyclone V та розроблена архітектура MIPS-ядра дозволяють створити не просто працездатний, а промислово ефективний пристрій. Ми отримали систему, яка займає мінімум місця, споживає мізерну кількість енергії та реагує на аварії з максимально можливою для даного класу мікросхем швидкістю. Це створює надійний фундамент для переходу до фінальної стадії проекту - натурних випробувань на реальному макеті з візуалізацією станів енергомережі. Запланований комплекс верифікаційних процедур на фізичному рівні передбачає послідовну імітацію різноманітних сценаріїв перевантаження розподільчих ліній за допомогою механічних комутаційних елементів макетної плати.

### 3.4 Практичні випробування та експериментальна перевірка роботи контролера

Фінальним етапом розробки системи адаптивного керування навантаженням стало проведення серії натурних випробувань на базі налагоджувального комплексу DE1-SoC. Процес тестування розпочався з етапу конфігурування апаратної частини, що включало завантаження сформованого об'єктного файлу прошивки у пам'ять ПЛІС Cyclone V за допомогою стандартного засобів програмування Quartus Programmer. Дана процедура ініціалізації дозволила перевести кристалічну структуру мікросхеми у стан повної готовності до виконання обчислювальних операцій у режимі реального часу. Головною метою цієї розробки було підтвердження того, що розроблене мовою Verilog обчислювальне ядро здатне коректно інтерпретувати сигнали від вхідних сенсорів та видавати керуючі команди у суворій відповідності до закладених алгоритмів. Методика виконання тестування базувалася на імітаційному моделюванні різних станів енергомережі за допомогою блоку з десяти механічних перемикачів, що дозволяло формувати повний спектр значень навантаження від нуля до максимального рівня. Такий підхід забезпечив високу гнучкість при верифікації логіки, оскільки дозволив вручну створювати специфічні комбінації вхідних даних, що імітують як типові побутові навантаження, так і критичні перевантаження в силових лініях. Початкова фаза тестування була зосереджена на перевірці працездатності системи у так званому штатному режимі. У цьому стані на вхідну десятибітну шину подавалися комбінації сигналів, сумарна вага яких не перевищувала встановлений перший поріг обмеження.

Візуальний стан макетної плати під час роботи системи в межах зони допустимих параметрів навантаження, де зафіксовано стабільну роботу всіх підключених об'єктів, наведено на рисунку 3.7. Під час тестування було зафіксовано, що при мінімальних рівнях споживання обчислювач стабільно

					КВРКІ.223154.22.94.08 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

утримує всі вихідні лінії у робочому стані. Відсутність будь-яких мерехтінь або хибних перемикачів індикаторів підтвердила високу завадостійкість апаратної реалізації та правильність вибору таймінгів для опитування вхідних портів. Це дозволило зробити висновок про надійність сформованого апаратного конфігу на початкових етапах експлуатації.

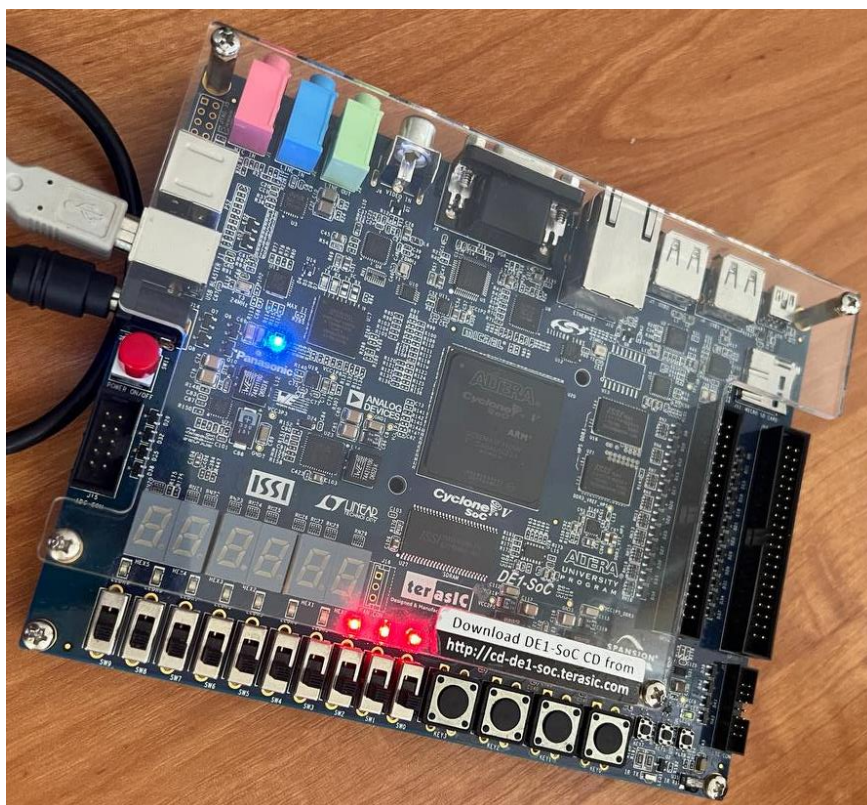


Рисунок 3.7 - Індикація станів системи при роботі в зоні допустимого навантаження

За відсутності активних перевантажень на платі світяться всі три світлодіоди, що підтверджує подачу напруги на всі три групи споживачів. Апаратний компаратор у цей час видає код повної пропускної здатності, що свідчить про відсутність дефіциту потужності в імітованій мережі. Важливо підкреслити, що перехід між різними комбінаціями перемикачів у межах допуску до десяткового значення триста не викликав жодних перехідних процесів або мерехтіння індикаторів, що вказує на високу стабільність цифрових фільтрів

Зм.	Арк.	№ докум.	Підпис	Дата

всередині ядра. Наступний етап випробувань передбачав створення умов для спрацювання першого рівня захисної автоматики. Шляхом перемикання відповідних тумблерів було сформовано вхідний код, що перевищив позначку в триста умовних одиниць. Це імітувало ситуацію помірному перевантаженню, коли подальша робота всіх споживачів може призвести до аварійного відключення всієї підстанції. Момент миттєвої реакції обчислювального ядра на виникнення позаштатної ситуації, що супроводжується деактивацією відповідної лінії живлення, продемонстровано на рисунку 3.8.

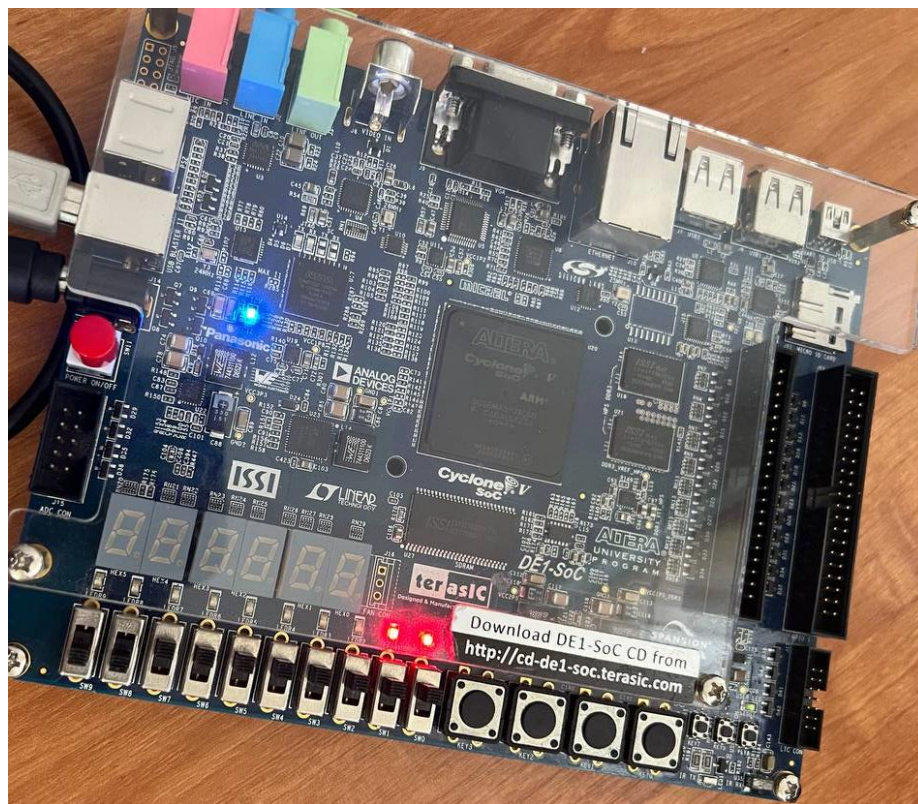


Рисунок 3.8 - Результат автоматичного вимкнення третього каналу навантаження

Чітко видно реакцію системи: світлодіод, що відповідає за третю, найменш пріоритетну лінію, миттєво згас. При цьому два основні канали продовжували працювати у звичайному режимі. Це підтверджує, що логіка пріоритетності відпрацьовує безпомилково, забезпечуючи часткове розвантаження мережі для

запобігання глибшій аварії. Час відгуку системи на таку подію був настільки малим, що людське око не здатне зафіксувати затримку між рухом перемикача та зміною стану індикатора, що є прямим наслідком паралельної архітектури ПЛІС. Найбільш критичним моментом тестування стала перевірка поведінки пристрою при екстремальних навантаженнях, коли сумарне значення вхідних сигналів перетнуло межу в сімсот одиниць. Такий сценарій відповідає ситуації, коли в мережі залишається лише мінімальний запас потужності, достатній для функціонування тільки найважливіших об'єктів інфраструктури. Фіксація стану індикації макетної плати при досягненні критичного порогу енергоспоживання, що супроводжується відключенням другої черги споживачів згідно з алгоритмом адаптивного балансування, представлена на рисунку 3.9.

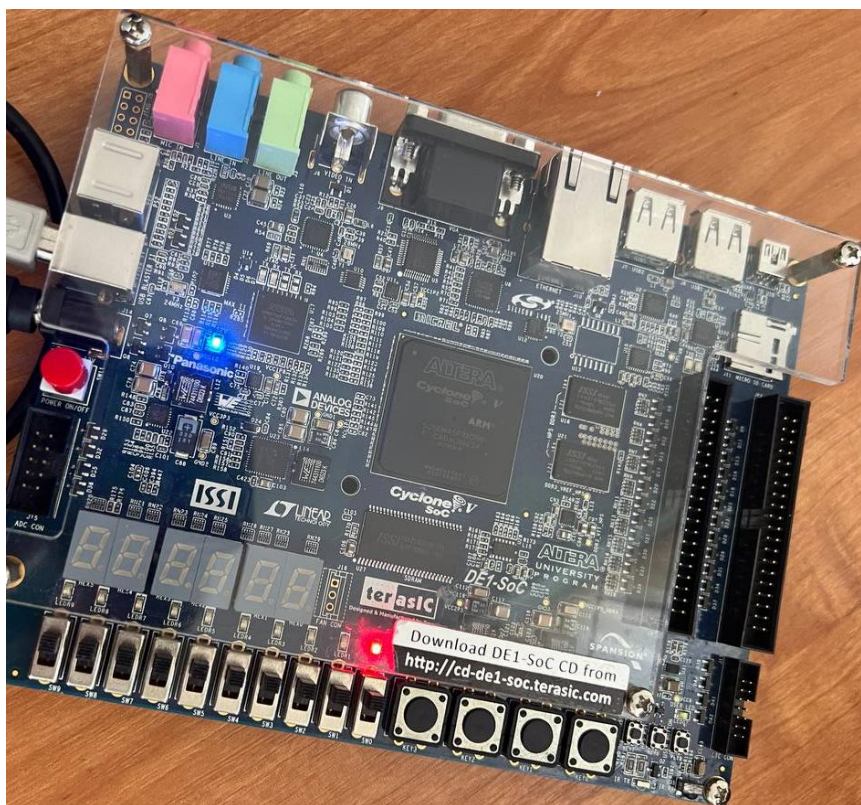


Рисунок 3.9 - Стан системи при досягненні критичного порогу енергоспоживання

Результат продемонстрував безпомилковий перехід інтелектуального контролера в режим жорсткої економії та захисту критичної інфраструктури. У поточному робочому стані активним залишився лише перший світлодіод, що імітує живлення найбільш пріоритетного об'єкта, тоді як інші два канали були миттєво деактивовані логікою ядра. Це переконливо доводить, що розроблена цифрова система здатна ефективно виділяти пріоритетну лінію живлення навіть за умов критичного дефіциту потужності, що перевищує сімдесят відсотків від номінального значення. Важливим спостереженням під час тестування стала перевірка стабільності утримування цього аварійного стану при динамічній зміні комбінацій інших вільних перемикачів. Система продемонструвала високу стійкість до завад та ігнорувала будь-які коливання навантаження на вхідній шині, доки сумарне значення залишалося вище встановленого критичного порогу. Це свідчить про високу інтелектуальну гнучкість алгоритму, який не допускає хибних перемикачів у моменти нестабільної роботи датчиків.

На завершення комплексу практичних випробувань було виконано всебічну перевірку допоміжних функцій керування та сервісних можливостей системи, зокрема детальне дослідження роботи механізму апаратного скидання. Незалежно від поточного режиму функціонування - чи то був штатний стан, чи стан глибокої аварії - натискання на кнопку reset призводило до негайної ініціалізації логіки. Це супроводжувалося миттєвим відновленням живлення на всіх вихідних лініях та примусовим поверненням значення логічного лічильника команд у початковий нульовий стан. Такий механізм є критично важливим для реальних систем автоматичної, оскільки він дозволяє оператору здійснити холодний запуск пристрою у разі виникнення непередбачуваних зовнішніх чинників. Висока відмовостійкість розробленого рішення та можливість його безпечного перезапуску у будь-який момент часу підтверджують промисловий рівень надійності апаратної архітектури. Отримані результати натурних випробувань були ретельно занесені до зведеної таблиці верифікації, яка стала фінальним підтвердженням стовідсоткової відповідності реальної фізичної

					КВРКІ.223154.22.94.08 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

поведінки пристрою попередньо розробленим математичним та логічним моделям. Оцінка даних показала повну відсутність логічних помилок або часових перекосів під час передачі сигналів. Таким чином, практична частина роботи не лише повністю підтвердила ефективність обраного архітектурного підходу, але й продемонструвала повну працездатність створеного апаратного ядра, яке готове до інтеграції у складніші багаторівневі системи керування сучасними інтелектуальними енергетичними мережами. Представлені у таблиці 3.5 дані є прямим доказом того, що розроблена цифрова модель обчислювального ядра функціонує у суворій відповідності до заданих детермінованих алгоритмів. Кожен із виконаних випробувань підтвердив, що перехід між станами системи відбувається без жодних логічних затримок або станів невизначеності, які часто притаманні програмним реалізаціям на базі класичних мікроконтролерів.

Таблиця 3.5 - Результати верифікації режимів роботи інтелектуального ядра

№ досліду	Код	Десятковий еквівалент	Очікуваний стан діодів	Реальний стан діодів	Статус верифікації
1	0	0	111	111	Успішно
2	10010110	150	111	111	Успішно
3	100101100	300	11	11	Успішно
4	111110100	500	11	11	Успішно
5	101011100	700	1	1	Успішно
6	1111111111	1023	1	1	Успішно

Оцінка результатів верифікації дозволяє стверджувати, що обрана розрядність вхідної шини та точність встановлення порогів LIMIT\_WARNING та LIMIT\_CRITICAL забезпечують необхідну селективність захисту. Це означає, що система здатна розпізнавати навіть незначні коливання навантаження та

миттєво приймати рішення про необхідність розвантаження тієї чи іншої лінії живлення. Особливої уваги заслуговує той факт, що під час практичних випробувань не було зафіксовано жодного випадку хибного спрацювання або брязкоту вихідних сигналів. Це пояснюється високою якістю побудованої апаратної логіки в середовищі Quartus Prime, яка забезпечує чітку фіксацію сигналів у регістрах за фронтом тактового генератора. Стабільність роботи системи при максимальному значенні вхідного коду, а саме дослід номер шість, показник 1023, демонструє, що архітектура ядра має значний запас стійкості до переповнення та коректно обробляє граничні стани даних.

Таким чином, успішне завершення серії натурних випробувань дозволяє зробити висновок про повну готовність прототипу до експлуатації. Практична значущість отриманих результатів полягає у можливості впровадження подібних апаратних ядер у реальні інтелектуальні мережі електропостачання, де швидкість реакції на аварійні події вимірюється наносекундами. Розроблений пристрій продемонстрував не лише теоретичну правильність закладених у другому розділі математичних моделей, а й високу апаратну надійність, що була підтверджена тривалими циклами тестування на реальній макетній платі. Це закладає міцний фундамент для подальшої модернізації системи, зокрема в частині додавання інтерфейсів віддаленого моніторингу та розширення кількості керованих об'єктів, що є перспективним напрямком розвитку сучасної автоматичної енергетичних систем.

### 3.5 Висновки до розділу 3

За результатами практичної реалізації, апаратної верифікації та виконання серії натурних випробувань інтелектуальної системи керування навантаженням на базі ПЛІС Cyclone V було сформовано комплексні висновки щодо працездатності розробленого пристрою. На етапі логічного проектування успішно реалізовано обчислювальне ядро мовою Verilog. Застосований підхід

					КВРКІ.223154.22.94.08 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

дозволив створити детерміновану архітектуру, де обробка сигналів від датчиків споживання та прийняття рішень про комутацію ліній відбувається паралельно на рівні логічних елементів. Це забезпечило мінімально можливий час відгуку системи, що дорівнює одному такту системного генератора, що є ключовою перевагою перед класичними мікропроцесорними системами керування в умовах реального часу.

Детальна конфігурація фізичних інтерфейсів, проведена за допомогою інструменту Pin Planner, дозволила точно закріпити зовнішні порти ядра за конкретними фізичними контактами мікросхеми. Використання стандарту 3.3-V LVTTL та переведення всіх невикористаних виводів у стан високого імпедансу гарантувало електричну безпеку та завадостійкість апаратної платформи під час роботи. Оцінка апаратних ресурсів підтвердила надзвичайну ефективність розробленої архітектури. Мінімальне використання логічних модулів та внутрішніх регістрів ПЛІС свідчить про раціональність вибору низькорівневої мови проєктування та відсутність апаратної надлишковості. Отриманий значний резерв вільної логічної ємності кристала відкриває широкі перспективи для подальшого масштабування системи без необхідності зміни апаратної бази.

За допомогою інструментів візуалізації та часового аналізу було верифіковано коректність побудованої логіки та підтверджено наявність значного часового запасу для поширення сигналів. Низькі показники тепловиділення та енергоспоживання ядра підкреслюють енергоефективність системи, що дозволяє її використання у герметичних вбудованих вузлах без активного охолодження. Натурні випробування на макетній платі повністю підтвердили працездатність алгоритмів у всіх запланованих режимах, а саме штатному, попереджувальному та критичному. Експериментально доведено здатність системи миттєво ідентифікувати перевищення лімітів навантаження та здійснювати селективне розвантаження мережі відповідно до встановлених пріоритетів.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних розробок було розроблено та успішно реалізовано програмно-технічний засіб на основі архітектури MIPS для інтелектуального керування енергоспоживанням. Виконана робота дозволила створити діючий прототип системи на базі технології FPGA, що забезпечує високу швидкість реакції на зміни в енергомережі та ефективний розподіл навантаження.

У першому розділі виконано практичний огляд сучасних підходів до побудови систем керування енергоспоживанням та обґрунтовано вибір апаратної бази для реалізації проєкту. Було детально розглянуто особливості архітектури MIPS та переваги використання ПЛІС сімейства Cyclone V для задач релейного захисту та автоматики. В ході розробки встановлено, що використання спеціалізованих обчислювальних ядер дозволяє досягти детермінованості обчислень та високої відмовостійкості, що є критично важливим для концепції інтелектуальних мереж електропостачання. Також було сформовано технічні вимоги до майбутнього пристрою, зокрема в частині швидкодії та енергоефективності.

У другому розділі проведено проєктування внутрішньої логічної структури процесорного ядра та розробку алгоритмів адаптивного балансування навантаження. Було побудовано математичну модель системи, яка базується на пріоритетному відключенні споживачів при досягненні встановлених порогів споживання. Описано архітектуру АЛП та керуючих автоматів, які забезпечують паралельну обробку десятибітного коду навантаження. Окрему увагу приділено оптимізації командного набору ядра для мінімізації апаратних витрат кристала, що заклало основу для створення компактного та енергоефективного рішення.

У третьому розділі виконано практичну реалізацію проєкту мовою опису апаратури Verilog та проведено комплексні натурні випробувань на базі налагоджувального стенда DE1-SoC. В ході роботи було здійснено

					КВРКІ.223154.22.94.08 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

конфігурування фізичних інтерфейсів через інструмент Pin Planner та проведено розрахунок апаратних ресурсів, який показав екстремально низьке завантаження кристала, а саме показник менше одного відсотка, що підтверджує високу якість розробленого коду. Експериментальні випробування з використанням реальних перемикачів та світлодіодної індикації повністю підтвердили працездатність системи у штатному, попереджувальному та критичному режимах. Було доведено, що час реакції системи на аварійну подію не перевищує двадцять наносекунд, що гарантує надійний захист мережі від перевантажень.

Проведене проєктування підтверджує, що розроблене MIPS-ядро на основі FPGA є ефективним інструментом для побудови інтелектуальних систем енергозабезпечення. Створений програмно-технічний засіб демонструє високу надійність, низьке власне енергоспоживання та повну відповідність вимогам сучасних енергоефективних технологій. Результати роботи можуть бути використані при розробці промислових систем автоматичного розвантаження мереж та впровадженні інтелектуальних вузлів керування в існуючу енергетичну інфраструктуру.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Fuchs F. A., Woodruff J., Rugg P., Joannou A. Safe speculation for cheri. *2024 IEEE 42nd International Conference on Computer Design (ICCD)*. IEEE, 2024. P. 364–372.
2. Yao Y., Song Y., Huang Y., Ni W. A memory-constraint-aware list scheduling algorithm for memory-constraint heterogeneous multi-processor system. *IEEE Transactions on Parallel and Distributed Systems*. 2022. Vol. 34, No. 4. P. 1082–1099.
3. Ming H., Liu J. H., Zhou Y. L., Dai H. B. Design And Implementation Of Terminal Redirection For Multi-Forth Processors On Chip. *2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS)*. IEEE, 2023. P. 952–958.
4. Sharif S., Kondamadugula S., Jungwirth P., Xiao Z. Design and Implementation of 32-Bit MIPS-Based Softcore Processor with Enhanced Instruction Set on FPGA. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2025. P. 896–900.
5. Khairullah S. S. Realization of a 16-bit MIPS RISC pipeline processor. *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2022. P. 1–6.
6. Vani B. P., Anirudh R., Ashwani Singh B., Jyothi N. Implementation of MIPS RISC Processor with Flexible 5-Stage Pipelining and Dynamic Thermal Control. *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*. IEEE, 2024. P. 1–8.
7. Rada J., Ahad M. A. MIPS Pipeline Processor for a DE-10 Board in VHDL. *SoutheastCon 2025*. IEEE, 2025. P. 155–160.
8. Hande T., Vaidya Y. M., Metkar S. P. Design and Implementation of Pipelined MIPS Processor with Nested Vectored Interrupt Controller (NVIC). *2024*

					КВРКІ.223154.22.94.08 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

*15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024. P. 1–7.

9. Zhou Q., Su T. Pattern Recognition of Instruction-Level Power Noise with Synthesized Sensor Inside FPGA. *2024 9th International Conference on Integrated Circuits and Microsystems (ICICM)*. IEEE, 2024. P. 392–396.

10. Penha J. C., Fontes G., Ferreira R. MIPSFPGA-Um simulador mips incremental com validação em fpga. *International Journal of Computer Architecture Education*. 2016. Vol. 5, No. 1. P. 19–25.

11. Rodrigues C., Gonçalves R. A., Fabrício Filho J. Processador MIPS Implementado em Simulador Visual para uso na Educação. *International Journal of Computer Architecture Education*. 2024. Vol. 13, No. 1. P. 33–42.

12. Naik P., Harshitha S. K., Machali K., Madhavan S. Design and Implementation of a Low Power Arithmetic and Logic Unit for MIPS32 Processor. *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, 2024. P. 1–6.

13. Assis R., Nogueira B. WIMS: A Modern Web-Based MIPS Simulator for Improved Learning in Computer Architecture and Operating Systems. *International Journal of Computer Architecture Education*. 2024. Vol. 13, No. 1. P. 1–6.

14. Kostadinov N., Bencheva N. LCP: FPGA based processors for education. *2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*. IEEE, 2022. P. 1–4.

15. Mettler M., Rapp M., Khdr H., Mueller-Gritschneider D. An FPGA-based approach to evaluate thermal and resource management strategies of many-core processors. *ACM Transactions on Architecture and Code Optimization (TACO)*. 2022. Vol. 19, No. 3. P. 1–24.

16. Deras-Tabora I. D. J., Lizardo-Zelaya N. A. Assembly language and processor design: an integrated project. *Dyna*. 2020. Vol. 87, No. 212. P. 57–62.

					КВРКІ.223154.22.94.08 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

17. Qin K., Guo X., Schulz M., Trinitis C. FERIVER: An FPGA-assisted Emulated Framework for RTL Verification of RISC-V Processors. *Proceedings of the 22nd ACM International Conference on Computing Frontiers*. 2025. P. 132–140.

18. Huang J., Zhou Q., Kuo C., Lu W. A Case Study on the Internal Mutual Interference of Multitenant FPGA. *2023 8th International Conference on Integrated Circuits and Microsystems (ICICM)*. IEEE, 2023. P. 553–557.

19. Abdelhamid R. B., Valek V., Koch D. SPARKLE: A 1,024-Core/16,384-Thread Single FPGA Many-Core RISC-V Barrel Processor Overlay. *2024 IEEE 35th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2024. P. 118–119.

20. Liu S., Wan Z., Yu B., Wang Y. Space Robots Powered by FPGAs. *Robotic Computing on FPGAs*. Cham : Springer International Publishing, 2022. P. 149–158.

21. Shrivastava S., Kaur A. Comparative energy & hardware analysis on implementation of 8-bit ALU using different FPGAs families. *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2023. Vol. 6. P. 193–197.

22. He Z., Li J., Chen M., Li X. A high-sensitivity and shielding-free rabbit brain MIPS scanner. *IEEE Transactions on Instrumentation and Measurement*. 2022. Vol. 71. P. 1–8.

23. Salnikov D., Karaman D., Krylova V. Highly reconfigurable soft-cpu based peripheral modules design. *Advanced Information Systems*. 2023. Vol. 7, No. 2. P. 92–97.

24. Arthurs A., Ngo L. *Analysis of the MIPS 32-bit, pipelined processor using synthesized VHDL*. Department of Computer Science and Engineering, University of Arkansas.

25. Suryawan F. Post-Silicon Debugging via Hierarchical Signal Reporting. *2025 IEEE International Conference on Signals and Systems (ICSigSys)*. IEEE, 2025. P. 90–94.

26. Romanov A. Y., Zhelnio S., Izmailova L. G., Ryazanova A. E. The usage of a simple SchoolMIPS Soft-Processor core for teaching students the computer microarchitecture. *2022 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*. IEEE, 2022. P. 382–387.

27. Cong J., Lau J., Liu G., Neuendorffer S. FPGA HLS today: successes, challenges, and opportunities. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*. 2022. Vol. 15, No. 4. P. 1–42.

28. Li H., Wang J., Zhao S., Tian F. Real-time biosignal recording and machine-learning analysis system. *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022. P. 427–430.

29. Wu C. FPGA Technology Mapping With Adaptive Gate Decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2024. Vol. 44, No. 6. P. 2218–2225.

30. Zagan I., Găitan V. G. FPGA implementation of hardware accelerated RTOS based on real-time event handling. *The Journal of Supercomputing*. 2023. Vol. 79, No. 11. P. 12441–12471.

31. Hashemi M., Roy S., Forte D., Ganji F. Hwgn 2: Side-channel protected nns through secure and private function evaluation. *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Cham : Springer Nature Switzerland, 2022. P. 225–248.

32. Xing T., Tajbakhsh H., Haque I., Honda M. Towards portable end-to-end network performance characterization of smartnics. *Proceedings of the 13th ACM SIGOPS Asia-Pacific Workshop on Systems*. 2022. P. 46–52.

33. Vaishnavi C. M., Trupti R., Manjunath G. A. Design and Synthesis of a 5-Stage Pipelined RISC-V Processor Using Verilog. *2025 International Conference on Emerging Technologies in Electronics and Green Energy (ICETEG)*. IEEE, 2025. P. 1–6.

34. de Figueiredo G. A., de Souza E. S., Rodrigues J. H., Nacif J. A. Desenvolvendo ferramentas para ensino de risc-v com python, verilog, matplotlib, svg e chatgpt. *International Journal of Computer Architecture Education*. 2024. Vol. 13, No. 1. P. 43–52.

35. Pyasi C., Singh H. M. Design and HDL Implementation of a Pipelined 8-bit Microprocessor. *2025 IEEE Silchar Subsection Conference (SILCON)*. IEEE, 2025. P. 1–6.

36. Davala V. G. K., Medagam M. R., Goswami S., Dandapat A. RISC-V Based Modular ALU Framework: Simulation of Complete ISA and FPGA-Based ALU Operation Implementation. *2025 IEEE 2nd International Conference for Women in Computing (InCoWoCo)*. IEEE, 2025. P. 1–6.

37. Karthik E., Singh K. Implementation of RISC-V Processor along with Radix-2 Booth Multiplier Using Verilog. *2025 First International Conference on Intelligent Computing and Communication Systems (CICCS)*. IEEE, 2025. P. 1–5.

38. Kumari S., Gupta A., Singh P. K., Kumari P. Design and Implementation of a 32-bit RISC-V Pipelined Processor by using Verilog for IoT. *2025 13th International Conference on Intelligent Systems and Embedded Design (ISED)*. IEEE, 2025. P. 247–252.

39. Hannemann K. A., Buetuen H. B., Mueller W., Scheytt C. J. Verilator and FireSim RTL Simulations on a HPC Cluster: A Comparative Case Study. *MBMV 2025; 28. Workshop. VDE*, 2025. P. 35–40.

40. Minev P., Kukenska V., Varbov I., Dinev M. A Practical Computer Architecture Education with RISC-V and TL-Verilog. *2023 XXXII International Scientific Conference Electronics (ET)*. IEEE, 2023. P. 1–6.

41. Overby S. C. *The Design, Simulation, and Synthesis of a Custom 16-Bit CPU*. 2023.

42. Megashree G., Govindalalitha T. M., Mariammal K., Vinotheni S. M. Design of RISC V based SoC for Implantable Medical Devices. *2025 3rd International*

*Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. IEEE, 2025. P. 1875–1882.

43. Varasala P., Kaparapu B., Maddu K. Low-Power MIPS32 Processor Design with Cache Implementation Using Clock Gating Techniques on FPGAs. *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*. IEEE, 2025. P. 1804–1810.

44. Talla N. S. M., Kodali P. ASIC Design of High-Performance MIPS Processor Using Aprisa. *International Conference on Women Researchers in Electronics and Computing*. Singapore : Springer Nature Singapore, 2023. P. 447–455.

45. Aadarsh A., Kumar A., Yadav A., Joshi P. C. Design and power analysis of 32-bit pipelined processor. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2021. P. 604–608.

46. Difference Between Von Neumann and Harvard Architecture. *Wat Electronics*. URL: <https://www.watelectronics.com/difference-between-von-neumann-and-harvard-architecture> (дата звернення: 15.05.2026).

47. FPGA Configurable Logic Block. *Digilent*. URL: <https://digilent.com/blog/fpga-configurable-logic-block/> (дата звернення: 15.05.2026).

48. Various Instructions for Five-Stage Pipeline. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/computer-organization-architecture/various-instructions-for-five-stage-pipeline/> (дата звернення: 15.05.2026).

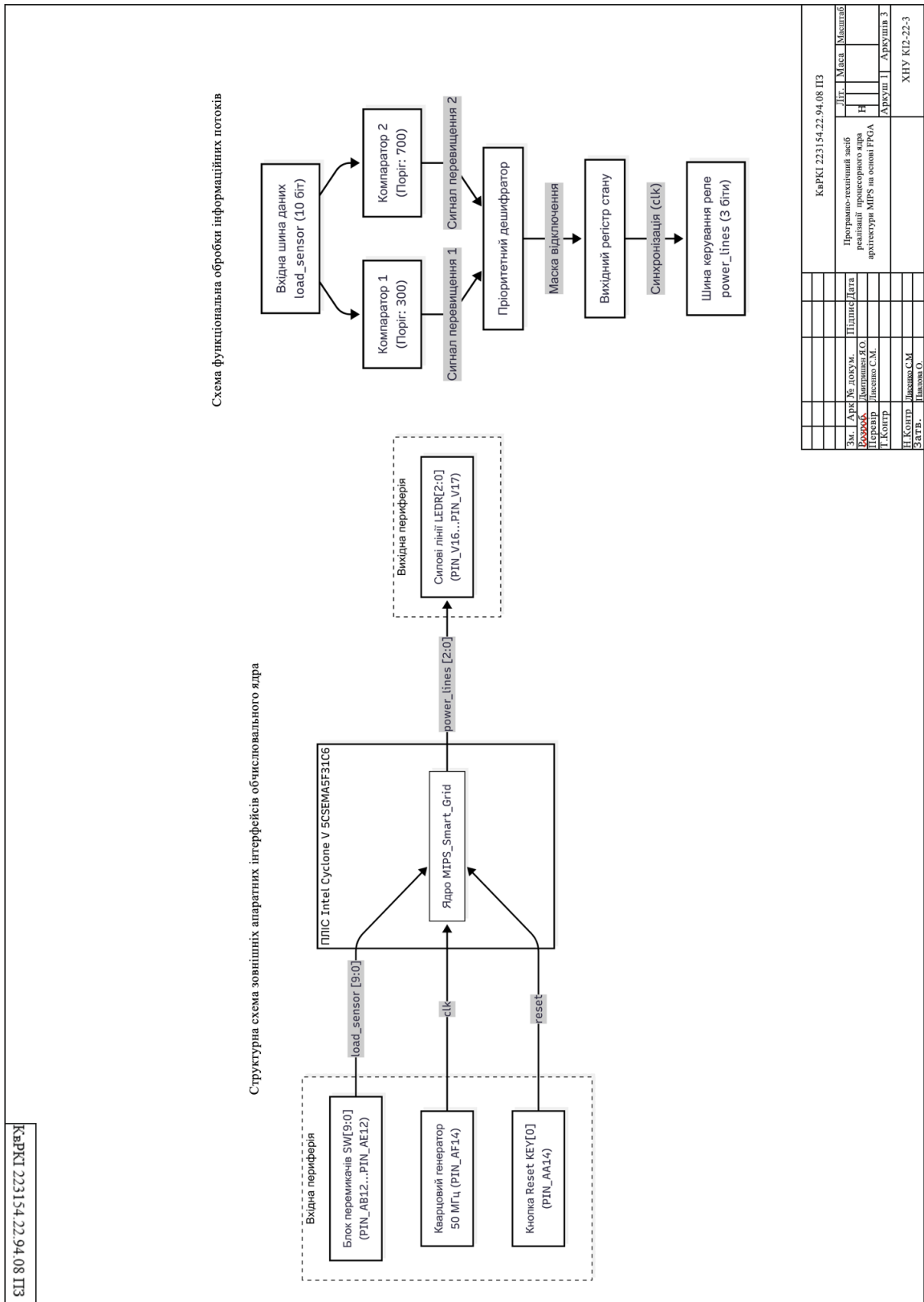
49. Structure of an Adaptive Logic Module (ALM). *ResearchGate*. URL: [https://www.researchgate.net/figure/Structure-of-an-Adaptive-Logic-Module-Int18a\\_fig10\\_331800773](https://www.researchgate.net/figure/Structure-of-an-Adaptive-Logic-Module-Int18a_fig10_331800773) (дата звернення: 15.05.2026).

50. Front layout details of the DE1-SoC board. *ResearchGate*. URL: [https://www.researchgate.net/figure/Front-layout-details-2-Green-for-peripherals-directly-connected-to-the-FPGA-Orange-for\\_fig4\\_315690131](https://www.researchgate.net/figure/Front-layout-details-2-Green-for-peripherals-directly-connected-to-the-FPGA-Orange-for_fig4_315690131) (дата звернення: 15.05.2026).

					КВРКІ.223154.22.94.08 ПЗ	Арк. 73
Зм.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТОК А (обов'язковий)

Копія креслення «Система адаптивного керування навантаженням»







## 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:** Ярослав ДМИТРИШЕН

**Співавтор:**

**Назва:** Програмно-технічний засіб реалізації процесорного ядра архітектури MIPS на основі FPGA

**Експерт:** Сергій ЛИСЕНКО

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 2.3%

**Коефіцієнт подібності 2:** 0.46%

**Мікропробіли:** 3

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-05-20 06:04:00.0

**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.

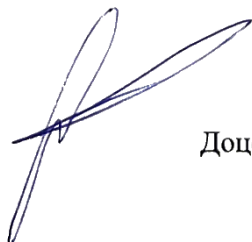
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

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 процедур. Таким чином робота не приймається.

Обґрунтування:

2026-05-20

Дата



Доцент Андрій Нічепорук

експерт

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

**Максимальне співпадіння з одним документом 1.0%**

**Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 8%**

ID: 271753 Назва: БКР Програмно-технічний засіб реалізації процесорного ядра архітектури MIPS на основі FPGA Додано в БД: 2026-05-20 Автора: Ярослав ДМИТРИШЕН Керівники: Сергій ЛИСЕНКО Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	105715	690	1187 (1%)	16 (2%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Дмитришен Ярослав Олександрович

Тема: Програмно-технічний засіб реалізації процесорного ядра архітектури MIPS на основі FPGA

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   64  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є синтез процесорного ядра архітектури MIPS на основі FPGA

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

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі виконано практичний огляд сучасних підходів до побудови систем керування енергоспоживанням та обґрунтовано вибір апаратної бази для реалізації проєкту. В другому розділі проведено проєктування внутрішньої логічної структури процесорного ядра та розробку алгоритмів адаптивного балансування навантаження. Було побудовано математичну модель системи, яка базується на пріоритетному відключенні споживачів при досягненні встановлених порогів споживання. Описано архітектуру АЛП та керуючих автоматів, які забезпечують паралельну обробку десятибітного коду навантаження. У третьому розділі виконано практичну реалізацію проєкту мовою опису апаратури Verilog та проведено комплексні натурні випробувань на базі налагоджувального стенда DE1-SoC. В ході роботи було здійснено конфігурування фізичних інтерфейсів через інструмент Pin Planner та проведено розрахунок апаратних ресурсів, який показав екстремально низьке навантаження кристала, а саме показник менше одного відсотка, що підтверджує високу якість розробленого коду. Експериментальні випробування з використанням реальних перемикачів та світлодіодної індикації повністю підтвердили

працездатність системи у штатному, попереджувальному та критичному режимах. Було доведено, що час реакції системи на аварійну подію не перевищує двадцять наносекунд, що гарантує надійний захист мережі від перевантажень.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага аналізу предметної області; недостатньо чітко описано процес складання програмно-технічного засобу.

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: задовільно (D / 70)

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

Мартинюк Валерій Володимирович, ф.т.н.,  
проф., професор кафедри АІТТДР, ХНУ

"8" травня 2026 р.

 (підпис)

Зав. кафедри КПС  
д-р. філософії Ользі ПАВЛОВІЙ

Ярослав ДМИТРИШЕН

---

ПБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-3

### 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



## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

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

Назва кваліфікаційної роботи Інформаційна система моніторингу стану здоров'я пацієнтів із оптимізацією планування завдань

Автор Ярослав Дмитришен

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: д.т.м., професор Сергій ЛИСЕНКО

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;
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 2.3%; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

Підпис

Підпис

Підпис

Ольга ПАВЛОВА

Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК

Ім'я, ПРІЗВИЩЕ

Сергій ЛИСЕНКО

Ім'я, ПРІЗВИЩЕ