ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерних систем та технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Програмна реалізація методу побудови трійкових логічних функцій на основі двовходового багатопорогового елемента багатозначної логіки»

(тема кваліфікаційної роботи українською мовою)

«Software implementation of the method of constructing ternary logic functions based on a two-input multi-threshold element of multi-valued logic»

(тема кваліфікаційної роботи англійською мовою)

Виконала: здобувачка денної форми навчання
спеціальності 123 Комп'ютерна інженерія
(код, назва спеціальності)
Освітня програма Комп'ютерна інженерія
(назва)
Будат Катерина Володимирівна
(прізвище, ім'я, по-батькові здобувача)

Керівник ст. викл. Мартинович Л.Я. _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент д.т.н., проф. Гунченко Ю.О.
(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
комп'ютерних систем та технологій
№ ___ від ____.____. 20___ р.

Завідувач кафедри
_____ Юрій ГУНЧЕНКО
(підпис) (прізвище, ім'я)

Захищено на засіданні ЕК № __
протокол № __від ____.____.20___ р.

Оцінка_____/____/_____
(за національною шкалою/шкалою ECTS/ бали)
Голова ЕК
_____ Світлана АНТОЩУК
(підпис) (прізвище, ім'я)

Одеса 2025

АНОТАЦІЯ

У кваліфікаційній роботі розглядається тема «Програмна реалізація методу побудови трійкових логічних функцій на основі двовходового багатопорогового елемента багатозначної логіки».

Актуальність дослідження обумовлена потребою в нових підходах до побудови обчислювальних систем, які б забезпечували високу щільність логіки, енергоефективність і масштабованість. У роботі проаналізовано сучасний стан досліджень у сфері трійкової логіки, вивчено особливості побудови логічних елементів на її основі та визначено обмеження існуючих підходів.

Запропоновано метод побудови трійкових логічних функцій із використанням двовходового чотирьохпорогового багатопорогового елемента багатозначної логіки (БПЕБЛ), що дозволяє досягти високої функціональної гнучкості при збереженні структурної простоти. Розроблено програмний інструмент, який автоматизує синтез трійкових функцій, будує відповідні логічні схеми та візуалізує результати.

Отримані результати можуть бути використані для побудови логічних схем у системах цифрової обробки сигналів, комп’ютерних архітектурах нового покоління, а також у пристроях, де критичними є енергоефективність, компактність і швидкодія.

ABSTRACT

The qualification thesis addresses the topic: "Software Implementation Of The Method Of Constructing Ternary Logic Functions Based On A Two-Input Multi-Threshold Element Of Multi-Valued Logic".

The relevance of the research is driven by the need for new approaches to the design of computational systems that ensure high logic density, energy efficiency, and scalability. The thesis analyzes the current state of research in the field of ternary logic, examines the characteristics of logic element construction based on it, and identifies the limitations of existing approaches.

A method is proposed for constructing ternary logic functions using a two-input, four-threshold multi-threshold element of multi-valued logic (MTEMVL), which provides high functional flexibility while maintaining structural simplicity. A software tool has been developed that automates the synthesis of ternary functions, constructs the corresponding logic circuits, and visualizes the results.

The obtained results can be used for building logic circuits in digital signal processing systems, next-generation computer architectures, as well as in devices where energy efficiency, compactness, and high performance are critical.

ЗМІСТ

ВСТУП

Стрімкий розвиток цифрових технологій супроводжується постійним зростанням вимог до продуктивності, енергоефективності та компактності сучасних обчислювальних систем. З огляду на ці виклики, актуальним стає пошук нових підходів до побудови логічних елементів, схем та архітектур обробки інформації, здатних забезпечити вищу щільність обчислень, кращу масштабованість і зниження енергоспоживання.

Одним із перспективних напрямів у цьому контексті є використання багатозначної логіки, зокрема трійкової логіки, яка передбачає застосування трьох логічних рівнів замість традиційних двох. Такий підхід дозволяє зменшити кількість необхідних логічних елементів, збільшити обсяг інформації, що передається однією логічною одиницею, а також оптимізувати процеси зберігання та обробки даних.

Проте впровадження трійкової логіки стикається з низкою проблем, зокрема — складністю реалізації фізичних компонентів, які здатні стабільно працювати з трьома логічними рівнями, а також забезпеченням сумісності з наявною двійковою елементною базою.

Одним із рішень, що здатне подолати зазначені обмеження, є застосування багатопорогових елементів багатозначної логіки (БПЕБЛ). Такі елементи дозволяють реалізовувати трійкові логічні функції шляхом керування рівнями активації на основі суми вхідних сигналів, що дає змогу досягати значної функціональної гнучкості без надмірної ускладненості схеми.

Незважаючи на перспективність такого підходу, питання алгоритмізації процесу побудови трійкових логічних функцій із використанням БПЕБЛ залишається відкритим. Особливо актуальною є потреба у розробці ефективного програмного інструменту, який би дозволяв автоматизовано здійснювати синтез трійкових функцій, будувати відповідні логічні схеми та забезпечував візуальне представлення отриманих результатів.

З огляду на викладене, актуальність даної дипломної роботи полягає в необхідності створення програмного методу побудови трійкових логічних функцій на основі багатопорогового елемента, що відкриває нові можливості для дослідження, моделювання та впровадження багатозначної логіки у сучасну цифрову техніку.

Метою цієї роботи є розробка та програмна реалізація методу побудови трійкових логічних функцій, що базуються на двовходовому чотирьохпороговому елементі багатозначної логіки.

Для досягнення поставленої мети передбачено розв’язання таких завдань:

– проаналізувати сучасний стан досліджень у галузі трійкової логіки та її застосувань;

– дослідити проблеми і архітектурні підходи до побудови трійкових логічних елементів;

– розробити метод побудови трійкових логічних функцій із використанням чотирьохпорогового БПЕБЛ та проаналізувати його ефективність;

– створити програмне забезпечення для автоматизованого синтезу трійкових функцій із використанням БПЕБЛ;

– протестувати програму на прикладах базових трійкових логічних операцій.

Об’єктом дослідження є трійкові логічні функції, а предметом — методи їх програмної реалізації за допомогою двовходового багатопорогового елемента багатозначної логіки.

Результати дослідження можуть бути використані для побудови багатозначних логічних схем, що мають застосування в сучасних комп’ютерних архітектурах, системах цифрової обробки сигналів, енергоефективних обчислювальних пристроях та інших галузях, де важливими є швидкодія, компактність та логічна гнучкість.

1 ТЕОРЕТИЧНІ ОСНОВИ ТРІЙКОВОЇ ЛОГІКИ

1.1 Основні поняття трійкової логіки

У межах класичної цифрової схемотехніки домінуючим підходом традиційно залишається двійкова логіка, яка оперує двома логічними станами — логічним нулем (0) та логічною одиницею (1). Такий підхід має глибоку історію та ефективно використовується в побудові більшості сучасних електронних обчислювальних машин. Проте в умовах постійного ускладнення мікроелектронних систем, стрімкого зростання потреб в енергоефективності, мініатюризації пристроїв, а також зростаючого попиту на продуктивність та обчислювальну щільність, виникає об’єктивна необхідність пошуку нових логічних парадигм.

Однією з найбільш перспективних альтернатив класичному двійковому підходу є трійкова логіка — логіка, яка передбачає використання трьох дискретних логічних станів замість двох. Аналогічно до біта в двійковій системі, основною одиницею інформації у трійковій системі є трит (від англ. trinary digit). Один трит містить близько 1.58496 біта інформації (тобто $\log_2 3$). У різних системах ці значення можуть інтерпретуватися по-різному, наприклад:

- несиметрична логіка із значеннями {0, 1, 2};
- несиметрична негативна логіка із значеннями {0, -1, -2};
- симетрична логіка із значеннями {−1, 0, +1} тощо [1].

Трійкова логіка є частковим випадком багатозначної логіки (мультивалентної логіки), що розглядає логічні системи з кількістю станів більше ніж два. Теоретичні основи багатозначної логіки були закладені ще у працях Яна Лукашевича (Jan Łukasiewicz) у XX столітті. У подальшому трійкова логіка отримала розвиток у контексті побудови нових типів логічних елементів, систем автоматичного керування, штучного інтелекту та квантових обчислень.

1.2 Переваги та перспективи розвитку трійкової логіки

Однією з фундаментальних переваг є підвищення інформаційної ємності логічного сигналу. Кожен трійковий розряд може кодувати більше інформації, ніж двійковий, що дозволяє зменшити кількість необхідних логічних елементів для зберігання або передавання однакового обсягу даних. Це веде до скорочення загальної кількості компонентів у схемі, зменшення площі на кристалі та потенційного спрощення структури інтегральних мікросхем.

Крім того, трійкова логіка дає змогу знизити апаратну складність цифрових схем завдяки меншій кількості логічних вентилів та з'єднань, необхідних для реалізації тих самих функцій, що й у двійковому варіанті. Як наслідок, підвищується щільність інтеграції, зменшується загальна кількість транзисторів, а отже — підвищується технологічна ефективність проєктування електронних пристроїв.

Окремої уваги заслуговує потенціал до зниження енергоспоживання. Завдяки меншій кількості комутацій і можливості зниження тактової частоти без втрати продуктивності, трійкові системи можуть стати основою для енергоефективних рішень, що є критично важливим у сфері мобільної, портативної та вбудованої електроніки [2].

Також трійкова логіка створює сприятливі умови для моделювання нечіткої логіки, багатозначних процесів та природної невизначеності. Вона дозволяє природно описувати проміжні логічні стани, що складно реалізувати у двійкових системах. Це відкриває широкі можливості для застосування в галузях штучного інтелекту, обробки знань, експертних систем, а також в адаптивному та нечіткому керуванні.

З філософської та когнітивної точки зору, трійкова логіка відображає більш гнучкий підхід до моделювання процесів мислення, де між однозначними відповідями «так» і «ні» можливі проміжні варіанти. У цьому контексті трійкова логіка може вважатися ближчою до реального процесу

людського прийняття рішень, що відкриває перспективи її використання в нейроподібних і когнітивно орієнтованих обчислювальних архітектурах.

Таким чином, трійкова логіка має значний потенціал для подальшого розвитку як у фундаментальних дослідженнях, так і в прикладних технологіях, зокрема в контексті підвищення продуктивності, ефективності та адаптивності цифрових систем нового покоління.

1.3 Огляд сучасного стану досліджень

У сучасній науково-технічній літературі простежується стійка тенденція до активного вивчення трійкової логіки як альтернативи традиційним підходам до цифрового опрацювання інформації. Якщо раніше дослідження у цій галузі мали переважно теоретичний характер, то сьогодні дедалі більше уваги приділяється практичній реалізації трійкових схем і систем на основі новітніх технологій. У фокусі перебувають як розробка фізичних елементів трійкової логіки, так і створення архітектур вищого рівня — від базових логічних блоків до повноцінних обчислювальних пристроїв.

Значна частина досліджень присвячена пошуку ефективних технологій для побудови трійкових логічних схем. Так, у роботі [3] аналізується використання графенових нанострічкових транзисторів (GNRFET) для реалізації базових елементів трійкової логіки. Завдяки можливості керування пороговою напругою, GNRFET відкривають нові перспективи для побудови трійкових інверторів, елементів NAND, NOR, декодерів, мультиплексорів та напівсуматорів.

Важливою віхою розвитку цієї галузі стало створення прототипів обчислювальних пристроїв на трійковій логіці. У роботі [4] запропоновано архітектуру трійкового процесора (TLP), побудованого на основі вуглецевих нанотрубкових транзисторів (CNTFET). Цей процесор виконує всі інструкції за один такт, підтримує різні типи команд і складається з повноцінного набору функціональних блоків — модуля вибірки, регістрів, арифметико-логічного

блоку та пам’яті. Моделювання в середовищі HSPICE показало високу ефективність архітектури за критеріями швидкодії, енергоспоживання та складності реалізації.

Ще одним важливим напрямом є оптимізація логічних блоків на трійковій логіці. У нещодавньому дослідженні запропоновано однобітну арифметико-логічну одиницю (ALU) [5], побудовану на базі CMOS-технології з використанням трійкових логічних елементів (T-Gates). Запропонована архітектура дозволяє зменшити кількість логічних вентилів та забезпечити компактнішу реалізацію порівняно з бінарною логікою. Модель ALU була описана мовою VHDL, що надало змогу здійснити функціональну перевірку, попри обмеження традиційних засобів моделювання трійкових схем. Згідно з результатами, застосування трійкової логіки дозволило скоротити кількість транзисторів на 25% у порівнянні з аналогічною бінарною реалізацією.

Особливу роль у цьому напрямі відіграє багатопороговий елемент багатозначної логіки (БПЕБЛ) [6], що забезпечує уніфікований підхід до побудови трійкових цифрових систем. Його здатність змінювати логічну поведінку залежно від налаштування порогових значень дозволяє реалізовувати широкий спектр функціональностей, що підтверджується створенням різноманітних компонентів: трійкового півсуматорів [7], повного суматора [8], RS-тригера [9], а також універсального пристрою для побудови трійкових унарних операцій [10].

Розробка програмного забезпечення для побудови двовходових трійкових функцій, яку буде представлено в межах цієї роботи, базується саме на використанні багатопорогового елементу як структурної основи.

Єдність технологічної основи на базі БПЕБЛ сприяє стандартизації та масштабуванню трійкових систем [11].

Таким чином, сучасні дослідження демонструють сталість розвитку трійкових обчислювальних технологій, у яких поєднуються новітні матеріали, ефективні архітектурні рішення та універсальні логічні елементи.

1.4 Висновок

У цьому розділі було розглянуто фундаментальні аспекти трійкової логіки як перспективної альтернативи класичній двійковій логіці. Встановлено, що трійкова логіка, яка оперує трьома логічними станами, забезпечує підвищену інформаційну щільність, зменшення апаратної складності та потенціал до зниження енергоспоживання, що особливо актуально в умовах зростання вимог до продуктивності та мініатюризації електронних систем.

Проаналізовано сучасний стан досліджень у цій галузі, що свідчить про зростаючий інтерес наукової спільноти до практичної реалізації трійкових схем на основі новітніх матеріалів, таких як графенові та вуглецеві нанотранзистори.

Особливу увагу приділено багатопорогового елементу багатозначної логіки, який відкриває шлях до уніфікованої побудови трійкових цифрових пристроїв.

Таким чином, трійкова логіка має суттєвий потенціал для подальшого розвитку, як у теоретичному, так і в прикладному напрямі, що обґрунтовує доцільність створення програмного інструментарію для реалізації трійкових логічних функцій.

2 ПРОБЛЕМИ ТА ПІДХОДИ ДО РЕАЛІЗАЦІЇ ТРІЙКОВИХ ЕЛЕМЕНТІВ

2.1 Труднощі реалізації трійкових елементів на практиці

Незважаючи на численні переваги трійкової логіки з теоретичної точки зору, її практичне впровадження стикається з низкою суттєвих проблем, які уповільнюють перехід від концептуальних рішень до реальних технічних застосувань. Однією з ключових перешкод є несумісність із існуючою цифровою інфраструктурою, яка протягом десятиліть розвивалася на основі двійкової логіки. Архітектури процесорів, системи передачі даних, цифрові інтерфейси та більшість апаратних і програмних засобів побудовані з розрахунку на два логічні стани. У таких умовах інтеграція трійкових елементів вимагає впровадження додаткових блоків перетворення між двійковим і трійковим представленням, що ускладнює схемотехнічну реалізацію, збільшує затримки, погіршує енергетичні показники і знижує загальну ефективність системи.

Ще однією серйозною проблемою є відсутність стандартизованої підтримки трійкової логіки з боку сучасних інструментів автоматизованого проєктування. У більшості систем САПР немає готових моделей поведінки трьохзначної логіки, а симулятори не підтримують повноцінну перевірку трійкових схем. Це змушує розробників користуватися модифікованими або неформальними засобами, що ускладнює розробку, знижує надійність і обмежує масштабованість таких рішень у промисловості.

Рівень технологічної зрілості трійкової логіки також залишається низьким. Попри те, що ідеї багатозначної логіки виникли ще в середині XX століття, їхнє впровадження й досі не набуло системного характеру. Більшість реалізацій мають експериментальний статус, а серійні продукти практично відсутні. Це пояснюється як браком економічної доцільності для масового

виробництва, так і складністю у виготовленні відповідної елементної бази з достатнім рівнем надійності та стабільності роботи.

Окрему увагу слід звернути на проблеми масштабованості. Із зростанням складності цифрової системи, побудованої на трійковій логіці, виникають додаткові виклики, пов’язані з точністю розрізнення трьох логічних рівнів, стійкістю до шумів, температурними впливами та електромагнітними перешкодами. Для забезпечення стабільної роботи необхідно застосовувати спеціальні методи компенсації та корекції, що, у свою чергу, ускладнює архітектуру системи та підвищує її вартість [12].

У сукупності ці чинники пояснюють, чому попри активні наукові дослідження трійкова логіка ще не отримала широкого практичного застосування й залишається здебільшого предметом лабораторних експериментів та перспективних дослідницьких розробок.

2.2 Архітектурні підходи до побудови трійкових схем

В умовах домінування двійкових технологій у цифровій електроніці особливої актуальності набувають підходи, які дозволяють реалізовувати трійкову логіку із застосуванням наявної двійкової елементної бази. Така стратегія дає змогу скористатися перевагами багатозначної логіки — зокрема, зменшенням глибини логічних обчислень та потенційним скороченням кількості логічних елементів — без необхідності повної перебудови технологічної інфраструктури.

Одним із класичних напрямів є побудова трійкових функцій шляхом каскадування двійкових логічних елементів. Подібна імітація трьох логічних станів досягається через комбінацію стандартних вентилів, що дозволяє реалізувати базові операції трійкової логіки, зокрема AND, OR та NOT. Перевагою такого підходу є його повна сумісність із CMOS-технологіями, що робить його привабливим для практичних реалізацій [2]. Водночас, складність таких схем зростає нелінійно зі збільшенням логічної функціональності, що

призводить до збільшення затримок, енергоспоживання та зменшення масштабованості.

Іншим перспективним напрямом є інтеграція трійкової логіки безпосередньо в структуру пам’яті, зокрема шляхом використання логіки-в-пам’яті на основі нанолистових транзисторів із потрійним затвором. Цей підхід дозволяє одночасно реалізовувати логічну та запам’ятовуючу функції в одному елементі, що знижує витрати часу та енергії на передачу даних між пам’яттю і процесором. Однак така технологія наразі перебуває на ранньому етапі розвитку, потребує складних процесів виробництва та характеризується високою чутливістю до параметричних варіацій, що ускладнює її промислову реалізацію [13].

Ще один архітектурний підхід ґрунтується на використанні незбалансованих трійкових вентилів, у яких три логічні стани кодуються за допомогою двобітних комбінацій. Це дозволяє зменшити апаратну складність реалізації порівняно з повноцінною трійковою логікою, забезпечуючи при цьому базову тривалість логічних обчислень. Подібні рішення добре адаптуються до традиційної двійкової інфраструктури та забезпечують задовільну сумісність з існуючими цифровими платформами. Водночас двобітне подання ускладнює логічну прозорість і може призвести до зростання обсягу інформаційного представлення при реалізації складніших функцій [14].

Особливої уваги заслуговує мемристивна реалізація трійкових логічних елементів. Мемристори, здатні фіксувати три стабільні стани опору, природним чином підходять для побудови трійкових вентилів. Вони забезпечують високу щільність інтеграції, низьке енергоспоживання та здатність тривалого зберігання станів, що особливо важливо в енергообмежених системах. Разом із тим складність точного контролю рівнів опору, обмежений життєвий цикл пристроїв і труднощі інтеграції з CMOS-архітектурами залишаються чинниками, що стримують широке впровадження таких рішень [15].

Таким чином, існує низка підходів до реалізації трійкових схем на базі двійкових компонентів, кожен з яких має свої переваги й обмеження. Вибір конкретної архітектури значною мірою залежить від цільових характеристик системи, доступних технологій та вимог до енергоефективності, щільності й швидкодії.

2.3 Висновок

Реалізація трійкових логічних елементів залишається складною інженерною задачею через низку технічних, економічних та технологічних бар’єрів. Попри теоретичні переваги трійкової логіки, її практичне впровадження гальмується несумісністю з усталеними двійковими стандартами, відсутністю підтримки в САПР, низькою технологічною зрілістю та складністю масштабування.

Архітектурні підходи на основі двійкових компонентів дають змогу частково долати ці обмеження та відкривають можливості для експериментальних і спеціалізованих застосувань. Проте кожен з них має власні обмеження, що ускладнює вибір оптимального рішення для широкого використання.

У зв’язку з цим особливої актуальності набуває розробка уніфікованого методу побудови трійкових логічних функцій, який дозволяв би стандартизувати підходи до реалізації багатозначної логіки, забезпечити її сумісність із сучасною інфраструктурою та полегшити інтеграцію трійкових елементів у існуючі системи. Такий метод має враховувати як фізичні обмеження, так і особливості логічного подання даних, і може стати ключовим кроком до практичного поширення трійкової логіки в цифровій електроніці.

3 ПОБУДОВА ТРІЙКОВИХ ЛОГІЧНИХ ФУНКЦІЙ НА ОСНОВІ ДВОВХОДОВОГО БПЕБЛ

3.1 Чотирьохпорогова реалізація БПЕБЛ

У даній роботі використано багатопороговий елемент багатозначної логіки (БПЕБЛ) з чотирма порогами (рисунок 3.1) [6], який забезпечує п’ять рівнів розрізнення вхідного сигналу.

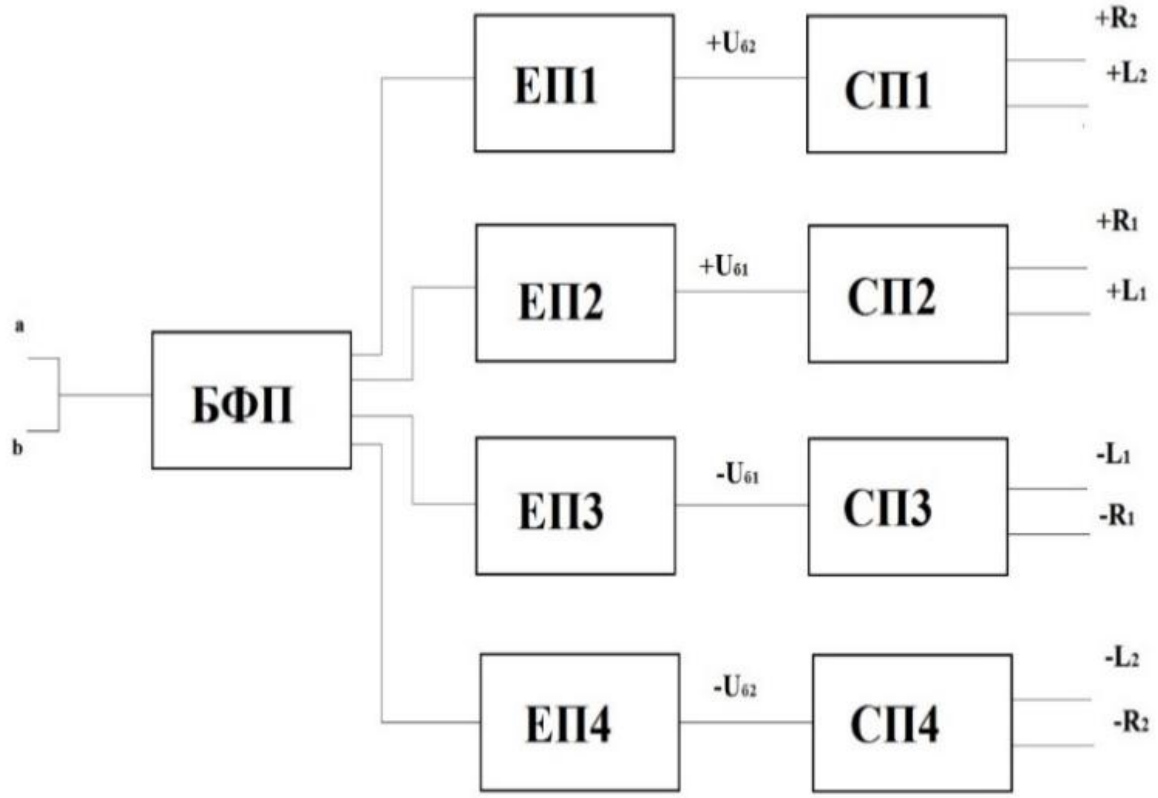

Рисунок 3.1 – Структурна схема чотирьохпорогового БПЕБЛ

Реалізація БПЕБЛ базується на таких компонентах:

- блок формування порогів (БФП), який забезпечує визначення рівнів порогової чутливості;
- емітерні повторювачі (ЕП1–ЕП4), кожен з яких активується при досягненні певного рівня вхідного сигналу;

- струмові перемикачі (СП1–СП4), які забезпечують комутацію струму залежно від вихідного сигналу відповідного емітерного повторювача.

На вхід блоку формування порогів надходять k дискретних струмових сигналів Ij від попередніх елементів. Ці сигнали можуть мати одне з кількох стандартних значень (наприклад, для двійкової логіки це будуть два значення: Ij = +1 і Ij = 0; для трійкової симетричної системи — три значення: Ij = +1, Ij = 0, Ij = –1). Загальне число сигналів можна виразити так:

$$k = k_{+1} + k_{-1} + k_0,$$

Відповідно:

– k_{+1} кількість вхідних сигналів зі значенням +1,

– k_0 зі значенням 0,

– k_{-1} зі значенням -1.

БФП здійснює сумування струмів та формує чотири порогові рівні, які через емітерні повторювачі передаються до струмових перемикачів. Активація певного емітерного повторювача залежить від результату суми струмів [6]. Відповідність між сумою вхідних сигналів та активністю емітерних повторювачів наведена в таблиці 3.1:

Таблиця 3.1 – Значення напруг на виходах ЕП1 - ЕП4

Сума вхідних струмів	ЕП1(+$U_{б2}$)	ЕП2 (+$U_{б1}$)	ЕП3 (–$U_{б1}$)	ЕП4 (–$U_{б2}$)
– –	1	1	0	0
–	0	1	0	0
0	0	0	0	0
+	0	0	1	0
++	0	0	1	1

Кожен з струмових перемикачів має два виходи (R та L), на яких формується стандартний струм залежно від активності сигналів з емітерних

повторювачів. Поведінка СП описується функцією terlev, яка дозволяє реалізувати логіку з урахуванням усіх п’яти можливих рівнів сумарного вхідного сигналу (таблиця 3.2):

Таблиця 3.2 – Можливі сигнали на виходах двоходового БПЕБЛ

Сума вхідних струмів	Вихідні сигнали струмових перемикачів							
	СП1		СП2		СП3		СП4	
terlev	+R2	+L2	+R1	+L1	–R1	–L1	–R2	–L2
– –	0	+	0	+	–	0	–	0
–	+	0	0	+	–	0	–	0
0	+	0	+	0	–	0	–	0
+	+	0	+	0	0	–	–	0
++	+	0	+	0	0	–	0	–

Виходи формуються шляхом комбінування певної кількості заданих вихідних сигналів, які арифметично додаються з урахуванням їх знакових значень [15].

3.2 Реалізація функцій за допомогою БПЕБЛ

БПЕБЛ дозволяє реалізовувати широкий клас трійкових функцій завдяки комбінації виходів структурних перетворювачів та використанню чотирьох порогових рівнів. Його архітектура забезпечує достатню гнучкість для побудови як базових, так і складних операцій, зокрема трійкових аналогів кон’юнкції, диз’юнкції, інверсії тощо.

Розглянемо приклад реалізації сильної кон’юнкції (таблиця 3.3) в трійковій логіці за допомогою БПЕБЛ [16]. Ця операція визначається як мінімальне з трьох можливих значень аргументів і є прямим узагальненням логічного AND для багатозначної системи.

Таблиця 3.3 – Таблиця істинності сильної кон’юнкції

a	b	terlev	$a\&_L b$	-R2	+R1	-R1	+L1
–	–	– –	–	–	0	–	+
–	0	–	–	–	0	–	+
–	+	0	–	–	+	–	0
0	–	–	–	–	0	–	+
0	0	0	–	–	+	–	0
0	+	+	0	–	+	0	0
+	–	0	–	–	+	–	0
+	0	+	0	–	+	0	0
+	+	++	+	0	+	0	0

Як показано в табл. 3.3, для побудови цієї функції необхідно об’єднати сигнали з виходів: -R2, +R1, -R1, +L1. Тобто:

$$a\&_L b = -R2; +R1; -R1; +L1;$$

3.3 Висновок

Завдяки використанню чотирьох порогів та структурованому підходу до обробки вхідних струмових сигналів, БПЕБЛ дозволяє формувати широкий набір логічних функцій у компактному вигляді. Однак для системного впровадження трійкової логіки в практичні цифрові пристрої необхідне створення уніфікованого методу побудови трійкових функцій, який би охоплював не лише окремі приклади реалізації, а й забезпечував формальну основу для автоматизованого проєктування багатозначних схем. Такий метод має враховувати узагальнені правила комбінації вихідних сигналів БПЕБЛ, дозволяти формувати довільні логічні функції трьохзначної логіки та бути адаптованим до сучасних САПР. Це забезпечить подальший розвиток трійкових технологій і підвищить їхню придатність до використання в масштабних інтегральних схемах.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПОБУДОВИ ТРІЙКОВИХ ЛОГІЧНИХ ФУНКЦІЙ

4.1 Алгоритм побудови трійкових логічних функцій

Розроблений у межах цієї роботи алгоритм дозволяє здійснювати побудову трійкових логічних функцій двох змінних шляхом формування необхідних вихідних сигналів багатопорогового елемента багатозначної логіки (БПЕБЛ), який має чотири симетричні пороги і, відповідно, п’ять рівнів активації.

Алгоритм побудови функції можна формалізувати у такій послідовності:

1. Обчислення суми вхідних сигналів (terlev).

Для кожної можливої комбінації значень вхідних змінних $(a,\ b)$, де $a,b \in \{-1,\ 0,\ +1\}$, визначається сума вхідних струмів, яка відповідає певному пороговому рівню активації. В результаті формується величина terlev, що може набувати одного з п’яти можливих рівнів: «−−», «−», «0», «+», «++».

2. Формування підмножин значень функції.

Усі вхідні комбінації $(a,\ b)$ розподіляються на три підмножини залежно від значення логічної функції $f(a,b)$:

- ті, для яких $f(a,\ b) = -1$
- ті, де $f(a,\ b) = 0$
- та ті, де $f(a,\ b) = +1$

3. Відбір відповідних виходів перемикачів.

Згідно з таблицею відповідності між рівнями terlev та активними виходами струмових перемикачів (СП), визначається набір виходів, які слід активувати для генерації бажаного логічного рівня:

- для рівня «+» активуються виходи: +R1, +R2, +L2, +L1;
- для рівня «−» — виходи: −R1, −R2, −L2, −L1;
- для рівня «0» — відсутня активація, тобто виходи не генерують струм, або утворюється баланс між протилежними рівнями.

4. Усунення конфліктів і небажаних активацій.

Якщо виявляється, що вибраний вихід, який формує правильне значення функції для певної комбінації вхідних змінних, водночас спричиняє хибну активацію або деформацію функції в іншій комбінації, виконується компенсація. Для цього додається додатковий вихід струмового перемикача, який активується на значенні terlev, що відповідає проблемній комбінації, та генерує протилежний сигнал. Це дозволяє «нейтралізувати» небажаний ефект та зберегти правильне функціонування для всіх можливих комбінацій.

5. Комбінація виходів і перевірка результату.

Після завершення відбору та компенсації виходів здійснюється об’єднання всіх активних виходів у логічну структуру. Якщо при перевірці виявляється, що деяка комбінація $(a,\ b)$ все ще дає неправильне значення функції $f(a, b)$, процедура повертається до попереднього етапу для повторного внесення корекційних виходів. Цей процес повторюється ітеративно до досягнення повної відповідності між функціональним описом і поведінкою схеми.

Таким чином, запропонований алгоритм дозволяє гнучко і точно формувати трійкові логічні функції довільної форми, використовуючи мінімально необхідний набір виходів багатопорогового елемента. За рахунок використання чотирьох порогів та відповідної системи термінальних рівнів terlev досягається висока точність і компактність логічної реалізації, що підтверджує ефективність підходу для реалізації багатозначних цифрових систем.

4.2 Вибір програмних інструментів і бібліотек для реалізації

Для реалізації програмної моделі побудови трійкових логічних функцій на основі двовходового багатопорогового елемента багатозначної логіки було обрано мову програмування Python, яка забезпечує широкий спектр засобів

для візуалізації, обробки даних та створення графічного інтерфейсу користувача.

У процесі розробки програмного забезпечення використовувалися такі програмні бібліотеки:

– tkinter — стандартна бібліотека Python для створення графічного інтерфейсу користувача. Застосовується для побудови основної структури віконного додатка, організації елементів управління (кнопок, полів вводу, меню тощо).

– ttkbootstrap — надбудова над tkinter, що забезпечує сучасне стилістичне оформлення елементів інтерфейсу згідно з концепцією Bootstrap. Дозволяє реалізувати привабливий, зрозумілий та інтуїтивно зручний інтерфейс користувача.

– Pillow (PIL) — бібліотека для роботи з растровими зображеннями. Застосовується для відкриття, обробки та збереження зображень, які можуть використовуватись для візуалізації логічних схем, результатів моделювання тощо.

– matplotlib — потужний інструмент для побудови графіків та діаграм. У роботі використовується для графічного представлення логічних функцій, таблиць істинності, а також результатів обчислень.

– threading —модуль для багатопотоковості, який використовується як для забезпечення плавності роботи інтерфейсу під час виконання тривалих операцій, так і для паралельного виконання обчислень, що підвищує продуктивність.

Вибір зазначених бібліотек обумовлений їх функціональністю, зручністю інтеграції, широким спектром можливостей для графічного представлення та високою сумісністю між собою. Це дозволило реалізувати зручне у використанні, стабільне та функціональне програмне забезпечення для моделювання трійкових логічних функцій відповідно до поставленої задачі.

4.3 Опис роботи розробленого програмного забезпечення

Розроблена програма здатна синтезувати трійкові двовходові функції на основі чотирьохпорогового БПЕБЛ. Програма забезпечує користувача зручним інтерфейсом для введення цільової функції, автоматичного виконання синтезу та генерації схеми її реалізації.

Основне завдання програми полягає у визначенні такої комбінації сигналів виходів струмових перемикачів (RL-сигналів), яка дозволить коректно реалізувати задану логічну функцію за допомогою структури БПЕБЛ. Результатом роботи програми є не лише текстове представлення формули, а й автоматично побудована структурна схема, таблиця відповідності сигналів, а також можливість збереження отриманих результатів.

Алгоритм роботи програми включає кілька ключових етапів:

1. Введення логічної функції користувачем: користувач вводить значення логічної функції, для цього в інтерфейсі використано зручні випадаючі списки (комбобокси), в яких кожному варіанту комбінації відповідає окреме поле для вибору значення функції (−1, 0, +1).

2. Синтез функції: після введення всіх значень функції програма проводить аналіз та здійснює синтез: визначає, які саме виходи БПЕБЛ необхідно активувати для забезпечення правильного формування значення функції для кожної з комбінацій.

3. Генерація структурної схеми: на основі результатів синтезу автоматично будується структурна схема, яка відображає активні виходи струмових перемикачів, що формують задану функцію. Схема включає умовні позначення елементів (емітерні повторювачі, струмові перемикачі) та підключення між ними згідно з логікою дії БПЕБЛ.

4. Виведення результаті: отримані результати відображаються у кількох формах:

– таблиця з комбінаціями значень rl-виходів і їх сумами.

– побудована схема функції.

– комбінація активних виходів струмових перемикачів, необхідна для реалізації функції.

На рисунку 4.1 подано загальний вигляд головного вікна програми:

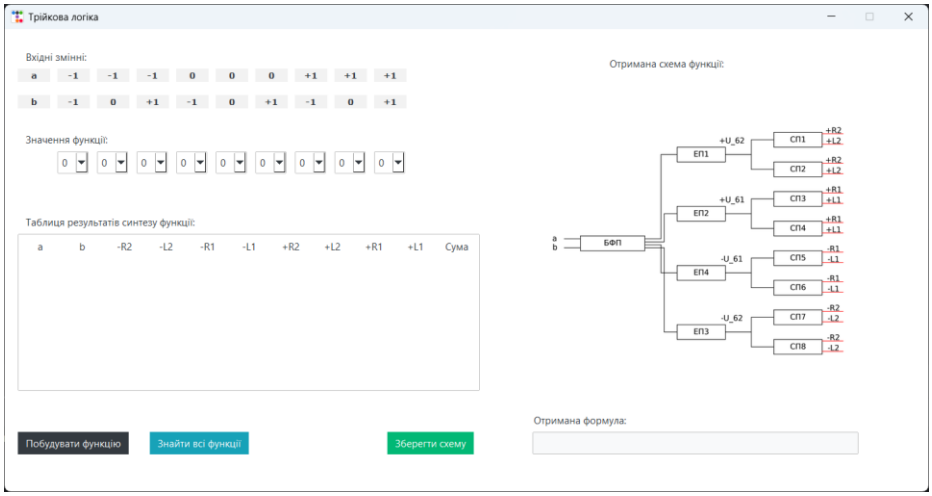

Рисунок 4.1 – Головне вікно програми

На рисунку 4.2 продемонстровано поля-комбобокси, за допомогою яких користувач задає значення функції для всіх дев’яти комбінацій вхідних змінних.

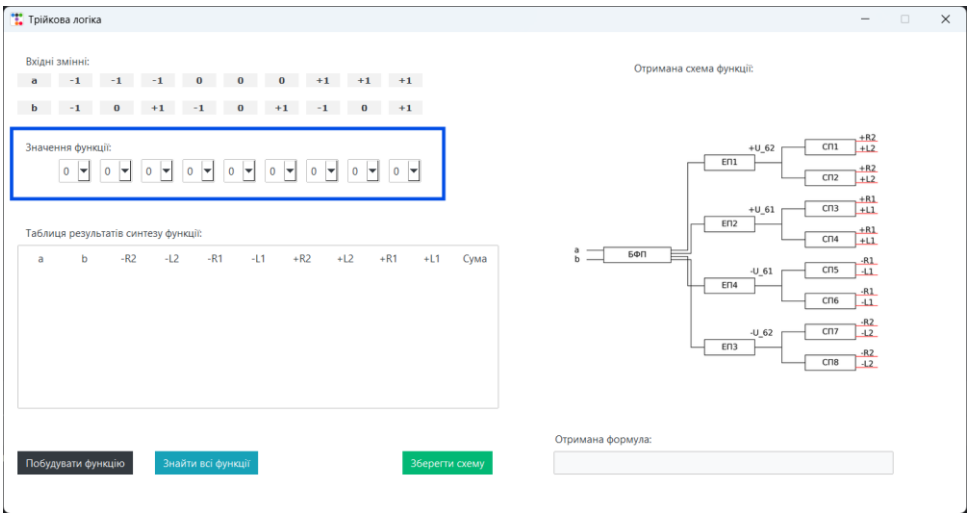

Рисунок 4.2 – Комбобокси для введення функції

Таблиця результатів (рис. 4.3) показує, які саме RL-виходи треба для побудови функції, їх значення на кожній комбінації поєднання двох змінних, а також їх сума:

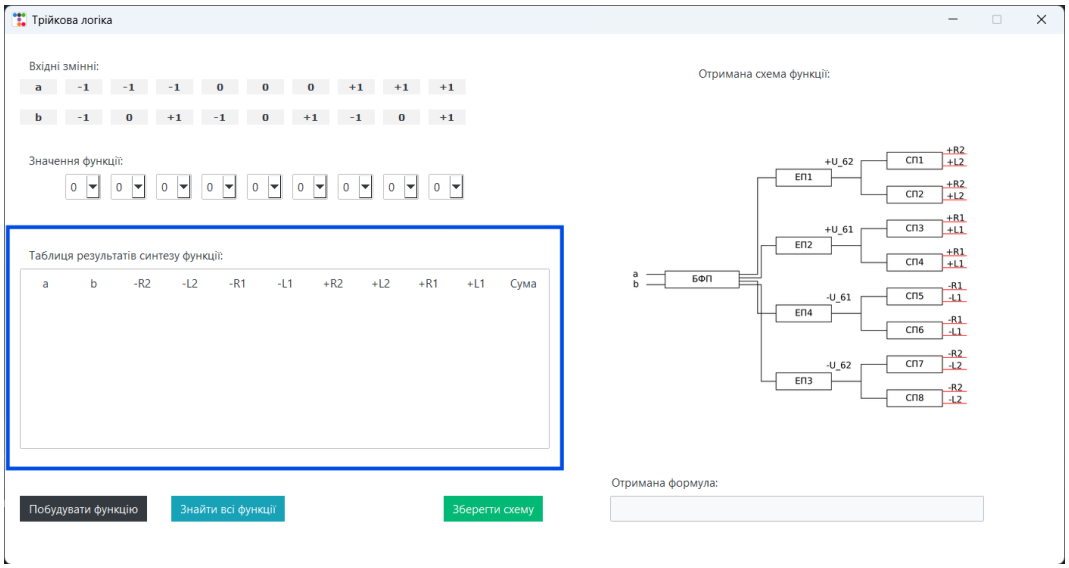

Рисунок 4.3 – Таблиця результатів синтезу функції

Структурна схема відображається одразу після знаходження комбінації в області відміченій на рисунку 4.4:

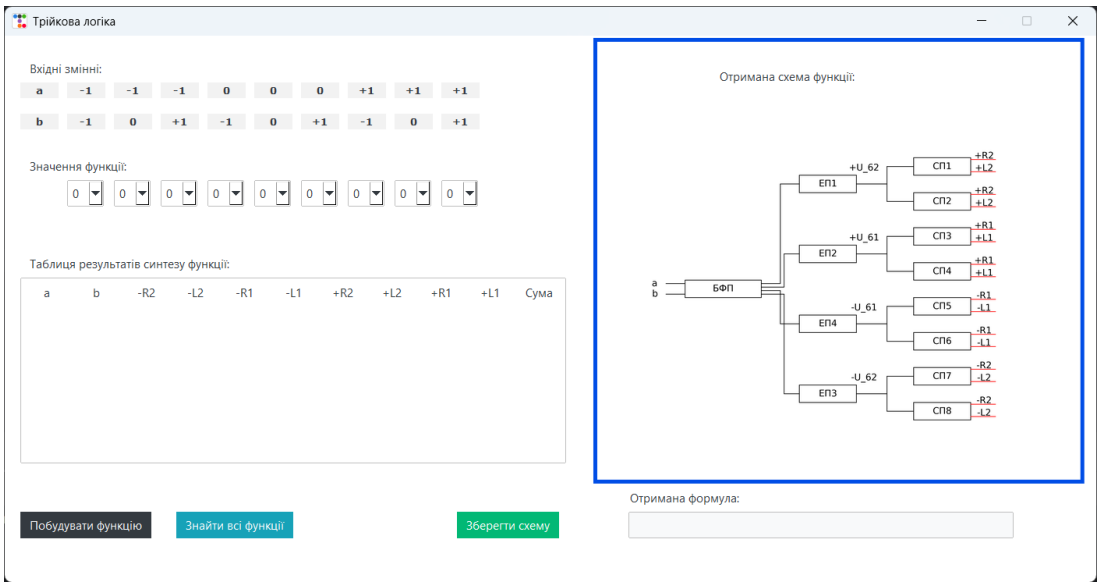

Рисунок 4.4 – Структурна схема функції

Поле для перегляду згенерованої комбінації RL-виходів (формули) наведено на рисунку 4.5:

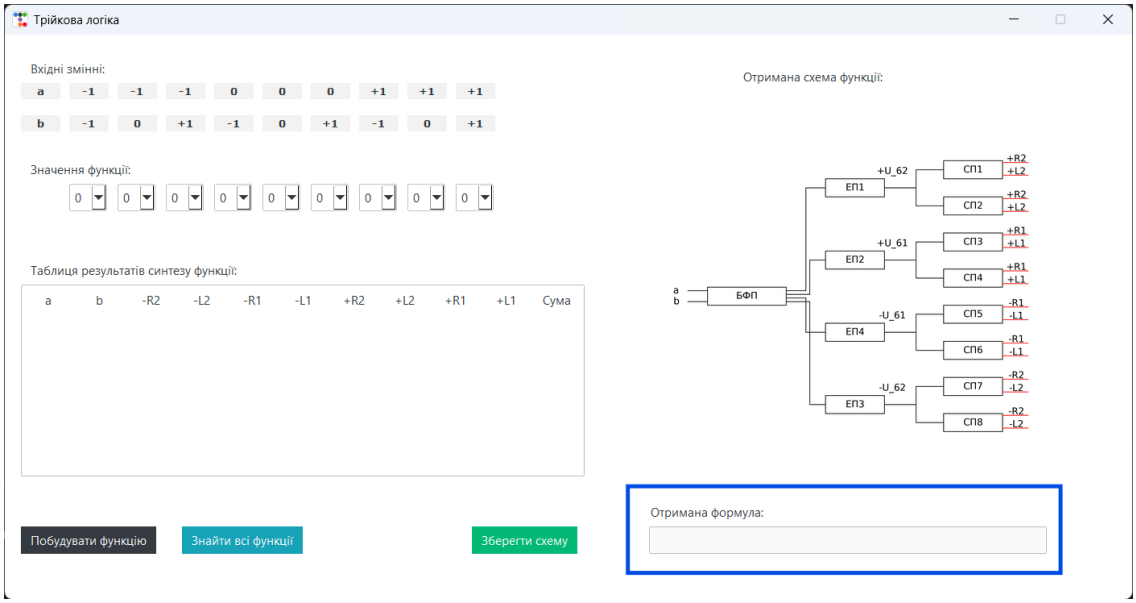

Рисунок 4.5 – Поле для відображення комбінації

На рисунку 4.6 представлено три основні функціональні кнопки інтерфейсу:

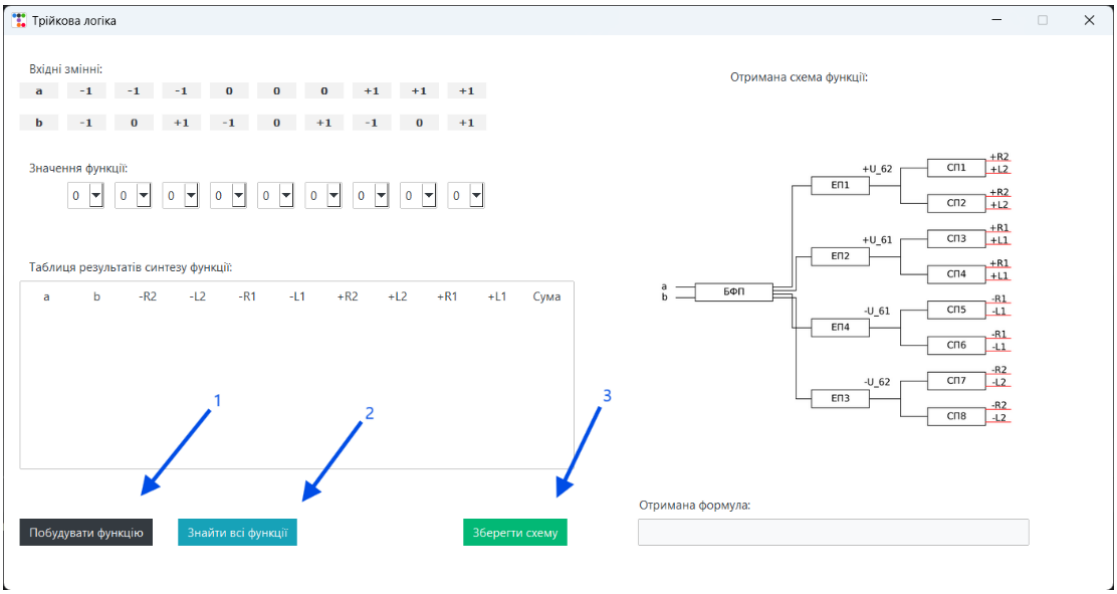

Рисунок 4.6 – Кнопки програми

1. Кнопка «Побудувати функцію» — здійснює аналіз введених користувачем значень функції, синтезує її реалізацію та будує відповідну схему.

2. «Знайти всі функції» — виконує повний перебір можливих трійкових функцій і знаходить RL-комбінації для кожної з них.

3. «Зберегти схему» — дозволяє зберегти згенеровану структурну схему у вигляді графічного файлу для використання у звітах або документації.

4.4 Тестування програмної реалізації та аналіз результатів

Кількість можливих логічних операцій у трійковій логіці значно перевищує аналогічну кількість у двійковій, що пов’язано з більшою кількістю можливих значень кожної змінної. Обчислення кількості таких операцій виконується за формулою 3^{3^n}, де n — це кількість аргументів. Для двійкової логіки кількість таких операцій визначається за допомогою формули 2^{2^n}. Так, наприклад, для двох аргументів у двійковій логіці кількість можливих функцій становить: $2^{2^2} = 16$. Це означає, що існує 16 різних способів реалізації двомісних булевих функцій (таких як AND, OR, XOR тощо).

Натомість у трійковій логіці, де кожен аргумент може набувати одного з трьох значень (−1, 0, +1), кількість можливих комбінацій значень аргументів значно зростає, і для двох змінних вона дорівнює: $3^{3^2} = 19683$ різних функції від двох трійкових змінних.

Така кількість функцій свідчить про значно більшу функціональну повноту трійкових логічних систем. Це відкриває нові можливості в галузі логічного синтезу та цифрової схемотехніки. Зокрема, трійкова логіка дозволяє ефективніше моделювати складні логічні та обчислювальні процеси, особливо в умовах обмеженого апаратного ресурсу або при проєктуванні компактних енергоефективних пристроїв.

У таблиці 4.1 подано приклади кількох типових двомісних трійкових операцій.

Таблиця 4.1 – Таблиця істинності двомісних трійкових операцій

a	b	Сильна кон’юкція	Диз’юнкція	Логічне додавання по модулю три	Приймати все	Консенсус
		$a\&_L b$	$a \vee b$	$a \oplus b$	$a \boxplus b$	$a \boxtimes b$
–	–	–	–	+	–	–
–	0	–	–	–	–	0
–	+	–	0	0	0	0
0	–	–	–	–	–	0
0	0	–	0	0	0	0
0	+	0	+	+	+	0
+	–	–	0	0	0	0
+	0	0	+	+	+	0
+	+	+	+	–	+	+

Для перевірки коректності роботи програми було здійснено синтез кожної з функцій, наведених у таблиці 4.1. На рисунках 4.7–4.11 представлено побудовані схеми для кожної з операцій:

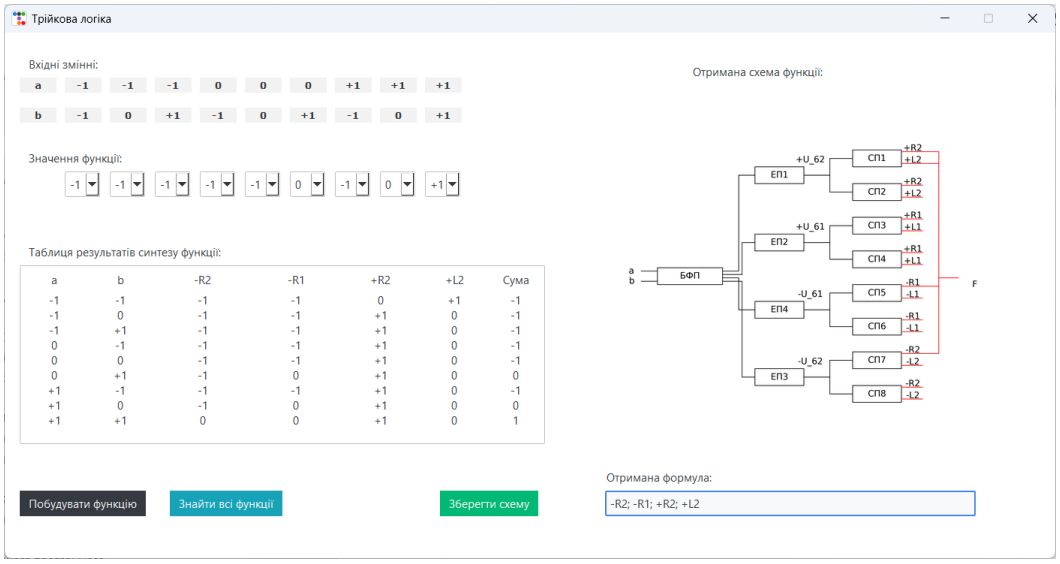

Рисунок 4.7 – Синтез сильної кон’юнкції

Сильна кон’юнкція (a &$_l$ b) виконується як операція мінімуму між двома аргументами, подібно до двійкової операції «і». Вона повертає найменший ступінь істинності з двох вхідних значень.

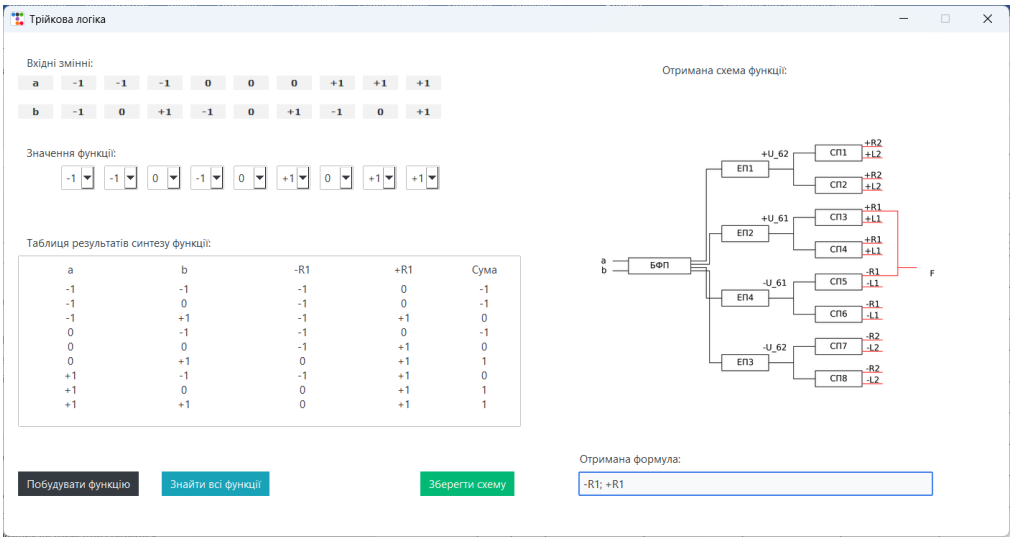

Рисунок 4.8 – Синтез диз’юнкції

Диз’юнкція (a ∨ b) є аналогом операції максимуму і відповідає двійковому «або». Вона визначає найвищий рівень істинності серед вхідних аргументів.

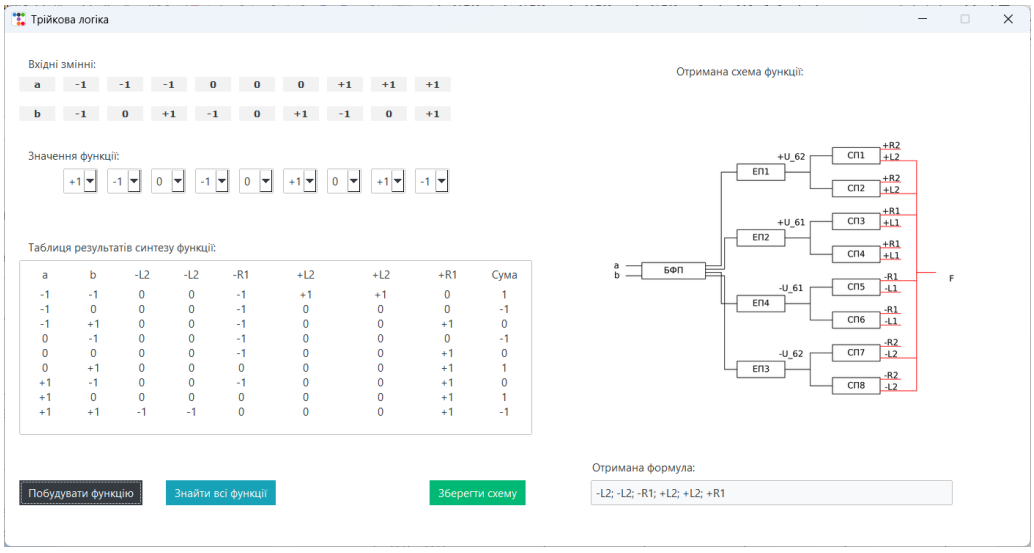

Рисунок 4.9 – Синтез додавання по модулю три

Логічне додавання по модулю три (a ⊕ b) узагальнює двійкову операцію XOR для трійкових значень. Воно виконується шляхом додавання аргументів за модулем 3 без урахування переносу, працюючи на рівні молодшого трита.

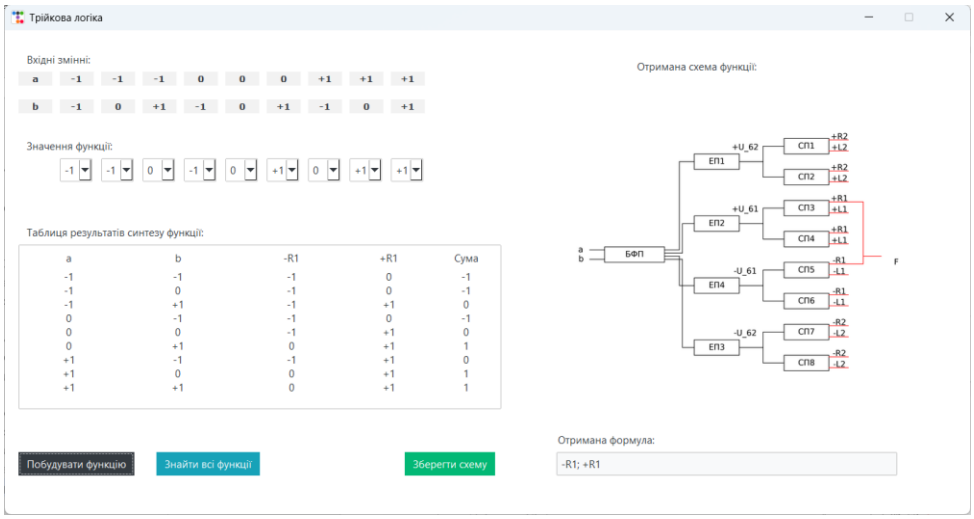

Рисунок 4.10 – Синтез «Приймати все»

Оператор «приймати все» (a ⊞ b) повертає визначене значення, якщо хоча б один із аргументів є визначеним. У випадку протиріччя або повної невизначеності результат також стає невизначеним.

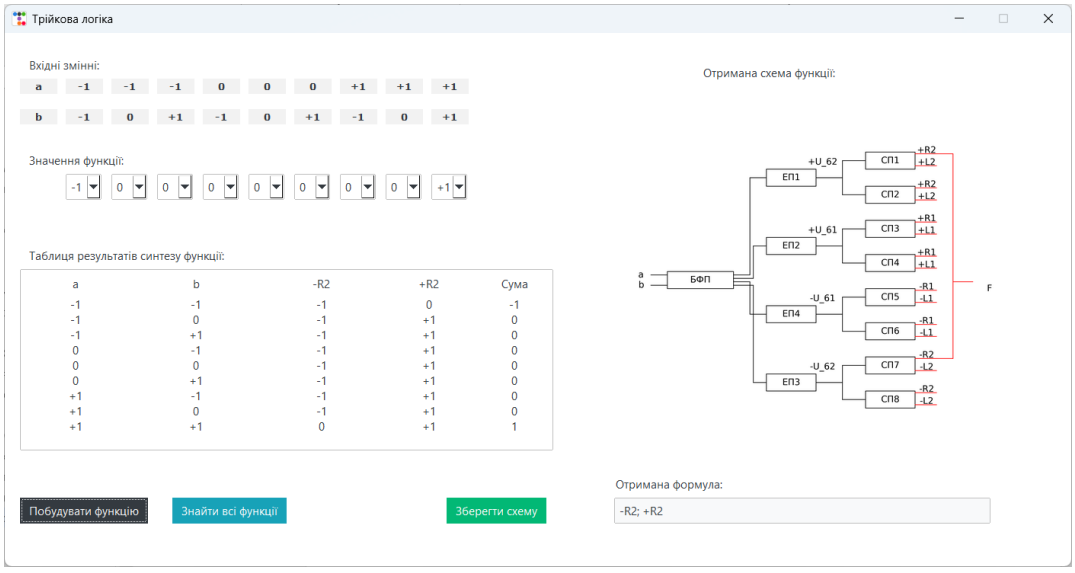

Рисунок 4.11 – Синтез консенсусу

Оператор консенсусу (a ⊠ b) є логічним дуалом до операції «приймати все» і повертає істинність лише у випадку повної узгодженості аргументів. Якщо значення різняться або є невизначеними — результат також невизначений.

І як видно з поданих прикладів, стовпчики «Суми» у таблиці результатів синтезу кожної функції повністю відповідають значенням, наведеним у таблиці 4.1. Це підтверджує правильність реалізації алгоритму та функціональну придатність побудованої програми для синтезу трійкових логічних елементів.

Окрім перевірки конкретних базових функцій, було виконано повний перебір з метою пошуку реалізацій усіх можливих бінарних трійкових логічних функцій, що можливо реалізувати на основі одного чотирьохпорогового БПЕБЛ. Для цього у програмі передбачено кнопку «Знайти всі функції», яка активує автоматизований синтез.

На рисунку 4.12 та 4.13 представлено результат виконання цього синтезу. Встановлено, що на базі одного елемента БПЕБЛ можливо синтезувати 231 унікальні функції.

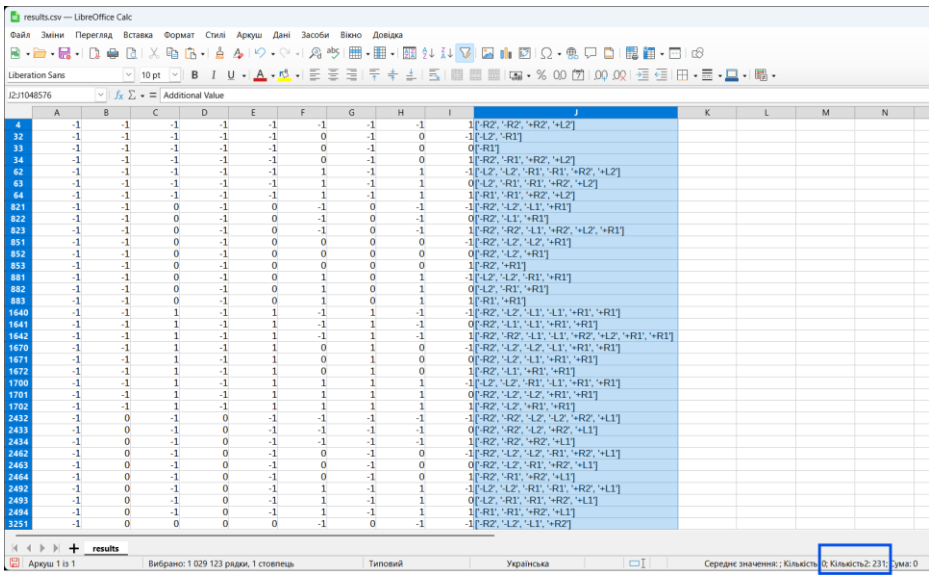

Рисунок 4.12 – Знаходження комбінацій для створення усіх бінарних трійкових логічних функцій (1)

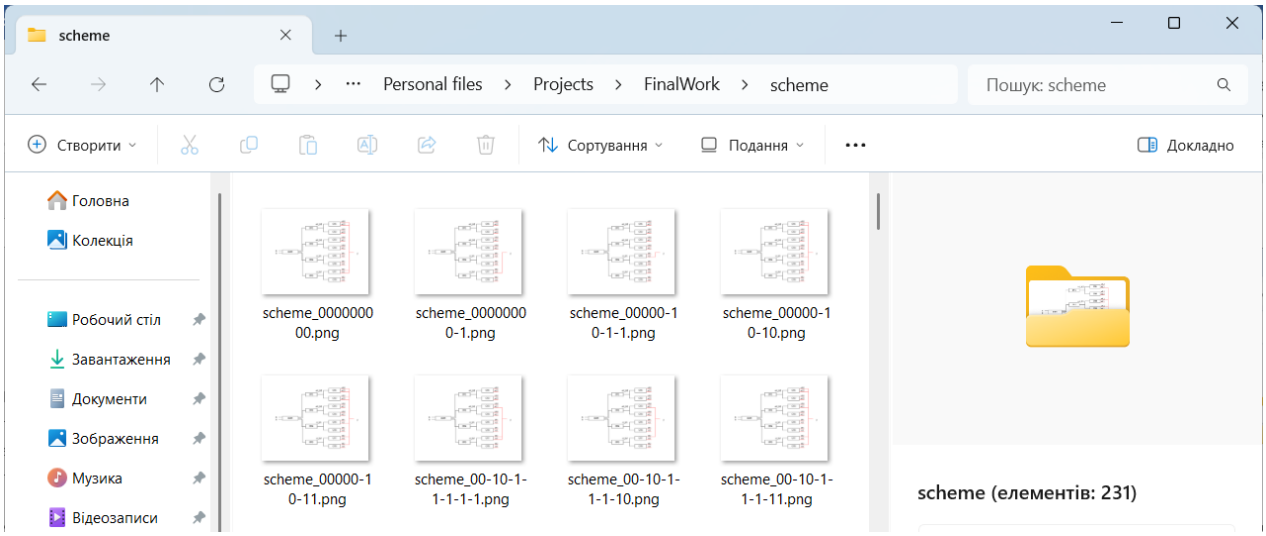

Рисунок 4.13 – Знаходження комбінацій для створення усіх бінарних трійкових логічних функцій (2)

Інформацію про реалізовані функції разом із відповідними RL-комбінаціями збережено у форматі CSV для подальшого аналізу. Всі побудовані схеми також автоматично збережено у форматі PNG у директорії scheme, що забезпечує зручний доступ до їх візуального представлення.

4.5 Висновок

Створене програмне забезпечення охоплює повний цикл — від введення даних до побудови аналітичного та візуального подання трійкової логічної функції, синтезованої на основі чотирьохпорогового елемента багатозначної логіки. Воно є універсальним інструментом для моделювання, тестування та дослідження трійкових функцій.

Хоча всього існує 19 683 бінарних трійкових функції, вони всі можуть бути реалізовані шляхом комбінації базових.

Отримані результати можуть бути застосовані у створенні логічних вузлів, цифрових підсистем або навіть в архітектурі обчислювальних систем на основі багатозначної логіки.

ВИСНОВОК

У роботі проаналізовано сучасний стан розробок у сфері трійкової логіки, виявлено її потенційні переваги порівняно з традиційною двійковою логікою, а також розглянуто ключові архітектурні підходи до побудови трійкових логічних елементів.

Було розроблено метод побудови трійкових логічних функцій з використанням двовходового чотирьохпорогового БПЕБЛ. Запропонований метод дозволяє реалізовувати широкий клас трійкових логічних операцій, зберігаючи при цьому структурну простоту та функціональну гнучкість.

Результатом роботи стало створення програмного засобу, що автоматизує процес синтезу трійкових логічних функцій, будує відповідні логічні схеми та надає їх візуалізацію. Проведене тестування на прикладах базових логічних операцій засвідчило коректність реалізації та практичну придатність розробленого інструменту.

Таким чином, поставлені в роботі завдання були повністю виконані. Запропонований підхід може слугувати основою для подальших досліджень у галузі багатозначної логіки, а також мати практичне застосування при розробці інноваційних обчислювальних архітектур, де критично важливими є зменшення енергоспоживання, збільшення щільності логічних елементів та забезпечення гнучкого управління логікою систем.

За результатами виконаної кваліфікаційної роботи опубліковано тези доповіді на всеукраїнській науковій конференції [17], статтю у Центральноукраїнському науковому віснику [18], а також підготовлено та представлено наукову конкурсну роботу. Відповідні матеріали наведено в додатках А, Б, В.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wang X.-Y., Dong C.-T., Wu Z.-R. A review on the design of Ternary Logic Circuits. Chinese Physics B. 2021. DOI: https://doi.org/10.1088/1674-1056/ac248b (дата звернення: 01.03.2025).
2. Zahoor F., Ahmad I., Kazim R. et al. Design implementations of ternary logic systems: A critical review // Results in Engineering. 2024. Article ID: 102761. DOI: https://doi.org/10.1016/j.rineng.2024.102761 (дата звернення: 15.04.2025).
3. Sandhie Z. T., Ahmed F. U., Chowdhury M. H. Design of ternary logic and arithmetic circuits using GNRFET // IEEE Open Journal of Nanotechnology. 2020. Т. 1. С. 77–87. DOI: https://doi.org/10.1109/ojnano.2020.3020567 (дата звернення: 18.04.2025).
4. Gadgil S., Sandesh G. N., Vudadha C. Design of a Ternary Logic Processor Using CNTFET Technology. Circuits, Systems, and Signal Processing. 2024. DOI: https://doi.org/10.1007/s00034-024-02726-x (дата звернення: 22.04.2025).
5. Priya A., Kumar A. Optimization of 1-Bit ALU using Ternary Logic // International Research Journal of Engineering Science Technology and Innovation. — 2024. — Vol. 6. — P. 1605–1610. — URL: https://www.researchgate.net/publication/384011130_Optimization_of_1-Bit_ALU_using_Ternary_Logic (дата звернення: 01.05.2025).
6. Багатопороговий елемент багатозначної логіки : пат. 118735 Україна : H03K19/00. № u201701717 ; заявл. 23.03.2017 ; опубл. 28.08.2017, Бюл. № 16.
7. Трійковий півсуматор на основі багатопорогового елемента багатозначної логіки : пат. 122996 Україна : H03K19/00 G06F7/00. № u201706115 ; заявл. 16.06.2017 ; опубл. 12.02.2018, Бюл. № 3.

8. Трійковий повний однорозрядний суматор : пат. 139770 Україна : H03K19/00. № u201905060 ; заявл. 13.05.2019 ; опубл. 27.01.2020, Бюл. № 2.
9. Трійковий RS-тригер : пат. 149386 Україна : H03K19/00. № u202104077 ; заявл. 13.07.2021 ; опубл. 11.11.2021, Бюл. № 45.
10. Універсальний пристрій для побудови трійкових унарних операцій : пат. 130182 Україна : H03K19/00. № u201806401 ; заявл. 08.06.2018 ; опубл. 26.11.2018, Бюл. № 22.
11. Левчук В., Гунченко Ю., Кузніченко С. та ін. Концепція побудови логічних і арифметичних пристроїв для багатозначних логік // Інформаційні технології та комп'ютерне моделювання : матеріали Міжнар. наук.-практ. конф. (Івано-Франківськ, 14–18 трав. 2018 р.). Івано-Франківськ, 2018. С. 216–219. URL: https://journal.comp-sc.if.ua/test/index.php/ITCM/article/view/428/220 (дата звернення: 15.04.2025).
12. Gunchenko Y., Shugailo Y., Bercov Y., Martynovych L. Analysis of the current state of the elements of ternary logic // Collection of Scientific Papers of Military Institute of Taras Shevchenko National University of Kyiv. 2022. № 76. С. 88–101. DOI: https://doi.org/10.17721/2519-481x/2022/76-08 (дата звернення: 15.04.2025).
13. Han J., Son J., Ryu S., Cho K., Kim S. Binary and ternary logic-in-memory using nanosheet feedback field-effect transistors with triple-gated structure // Scientific Reports. – 2024. – Т. 14, № 1. – DOI: https://doi.org/10.1038/s41598-024-57290-w. – (дата звернення: 01.05.2025).
14. Vijay V., Pittala C. S., Koteshwaramma K. C., Shaik A. S., Chaitanya K., Birru S. G., Medapalli S. R., Thoranala V. R. Design of Unbalanced Ternary Logic Gates and Arithmetic Circuits // Journal of VLSI Circuits and Systems. — 2022. — Vol. 4, No. 1. — P. 20–26. — DOI: https://doi.org/10.31838/jvcs/04.01.04 (дата звернення: 07.05.2025).

15. Li X.-J., Wang X.-Y., Li P., Iu H. H. C., Cheng Z.-Q. Ternary combinational logic gate design based on tri-valued memristors // Frontiers in Physics. – 2023. – Т. 11. – DOI: https://doi.org/10.3389/fphy.2023.1292336. – дата звернення: 13.05.2025).
16. Martynovych L., Gunchenko Y., Shugailo Y., Bercov Y. Design and synthesis of ternary logic elements // Computer Systems and Information Technologies. 2022. № 4. С. 52–60. DOI: https://doi.org/10.31891/csit-2022-4-8 (дата звернення: 15.04.2025).
17. Будат К. В., Мартинович Л. Я. Програмна побудова трійкових логічних елементів // Інформатика, інформаційні системи та технології: тези доповідей XXII Всеукраїнської конференції студентів і молодих науковців (Одеса, 25 квітня 2025 р.). – Одеса: Університет Ушинського, 2025. – С. 183.
18. Будат К. В., Мартинович Л. Я. Синтез трійкових двовходових логічних функцій із використанням багатопорогового логічного елементу // Центральноукраїнський науковий вісник. Технічні науки. – 2025. – Вип. 11(42), ч. II. – С. 45–51. – URL: https://mapiea.kntu.kr.ua/pdf/11(42)_II/11(42)_II_2025.pdf (дата звернення: 04.06.2025).

ДОДАТОК А

Тези доповіді на Всеукраїнській науковій конференції

Державний заклад
«ПІВДЕННОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
імені К. Д. УШИНСЬКОГО»

КПМ & І

ОНУ, кафедра МОКС

ОДЕСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ імені І. І. МЕЧНИКОВА

ДВАДЦЯТЬ ДРУГА ВСЕУКРАЇНСЬКА КОНФЕРЕНЦІЯ
СТУДЕНТІВ І МОЛОДИХ НАУКОВЦІВ

ІНФОРМАТИКА, ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

25 квітня 2025 р.

Одеса – 2025

Рисунок А.1 – Титульна сторінка збірника тез

Рис. 1.3 – Інтерфейс системи Crisis360

Висновки. Аналіз показав, що ефективна інформаційна система для координації рятувальних операцій повинна включати такі компоненти: моніторинг та аналітика в реальному часі, геоінформаційні системи (GIS) для оптимізації маршрутів, інтеграцію з іншими службами, а також гнучкість і масштабованість для роботи в різних умовах. Використання цих підходів у розробці власного рішення дозволить підвищити ефективність реагування на надзвичайні ситуації, зменшити час прийняття рішень та мінімізувати втрати.

У подальших дослідженнях планується детальніше вивчити можливості інтеграції штучного інтелекту для прогнозування розвитку кризових ситуацій та підвищення адаптивності систем до локальних умов.

Література

1. Система WebEOC [Електронний ресурс]. – Режим доступу: https://www.juvare.com/webeoc/.
2. Система Guardian IMS [Електронний ресурс]. – Режим доступу: https://www.qitplus.com/guardian-ims/.
3. Система Crisis360 [Електронний ресурс]. – Режим доступу: https://download.cnet.com/crisis360-emergency-management-communications-for-business-continuity-and-situational-awareness-for-iphone/3000-2064_4-75916673.html.

ПРОГРАМНА ПОБУДОВА ТРІЙКОВИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ

Будат К. В., Мартинович Л. Я.
ОНУ імені І.І. Мечникова

Ключові слова: трійкова логіка, багатопороговий елемент багатозначної логіки, трійковий логічний елемент, багатозначна логіка

Рисунок А.2 – Текст тези (1)

З розвитком інформаційних технологій зростає потреба в нових підходах до проєктування цифрових систем, що поєднують швидкодію, енергоефективність і компактність. Перспективною альтернативою традиційній бінарній логіці є трійкова логіка, яка завдяки трьом станам дозволяє підвищити обчислювальну щільність і зменшити складність з'єднань. Вона забезпечує значно ширші можливості для побудови логічних функцій: для двох входів у трійковій системі існує 19 683 варіанти функцій проти 16 у бінарній. Це вимагає нових ефективних методів синтезу. Побудова трійкових функцій є ключовою для створення енергоефективних архітектур, актуальних для мобільних, вбудованих і високопродуктивних систем.

Метою роботи є розробка програмного забезпечення для автоматизованої побудови двовходових трійкових логічних функцій.

У межах дослідження розроблено програмний модуль (рис. 1) для пошуку конфігурації виходів, необхідної для реалізації трійкової логічної функції відповідно до заданих параметрів. Побудова логічних елементів базується на багатопороговому елементі багатозначної логіки (БПЕБЛ) [1], а саме на його чотирьохпороговій версії. Процес передбачає підбір оптимального поєднання виходів цього елемента для формування потрібної логічної функції.

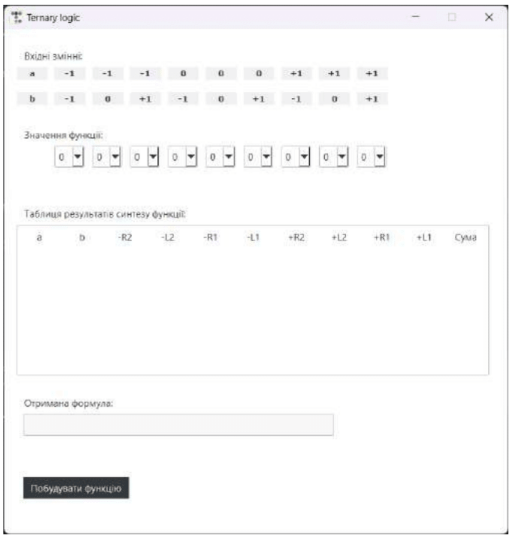

Рис. 1 – Головне вікно програми

Введення трійкової логічної функції в програмному модулі здійснюється за допомогою комбобоксів, що не лише забезпечує зручність задання значень для кожної комбінації вхідних змінних, але й зменшує ймовірність введення некоректних даних. Результати побудови відображаються у вигляді таблиці, в

Рисунок А.3 – Текст тези (2)

якій подано підібрані виходи R та L, їхні значення для відповідної комбінації входів, а також загальна сума після їх об'єднання. Окрім того, генерується формула, яка описує реалізовану логічну функцію у вигляді комбінації сигналів R та L. Приклад побудови функції представлено на рисунку 2.

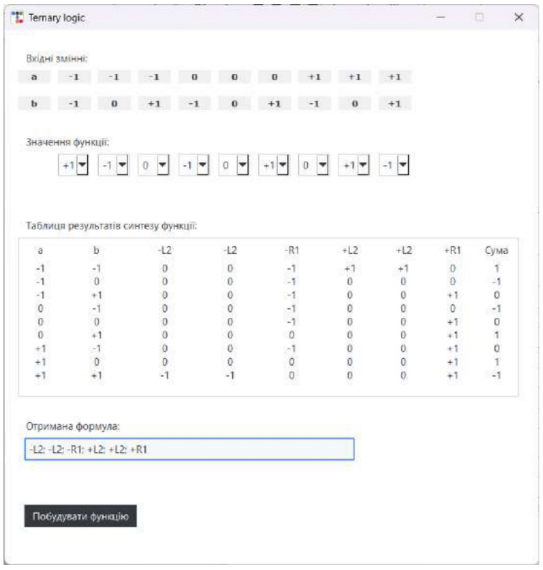

Рис. 2 – Побудова функції «Логічне додавання по модулю три»

Розроблений програмний модуль автоматизує процес побудови двовходових трійкових логічних функцій, що дозволяє суттєво зменшити трудомісткість синтезу та знизити ймовірність помилок. Завдяки цьому інструменту підвищується ефективність проєктування трійкових цифрових систем, що є важливим кроком у напрямі створення більш продуктивних і енергоефективних обчислювальних архітектур.

Література

1. Багатопороговий елемент багатозначної логіки : пат. 118735 Україна : H03K19/00. № u201701717 ; заявл. 23.03.2017 ; опубл. 28.08.2017, Бюл. № 16.

МІЖДИСЦИПЛІНАРНИЙ ЗВ'ЯЗОК ОСВІТНІХ КОМПОНЕНТ ПРИ РОЗРОБЦІ ШКАЛИ НОНІУСА

Волянський В. В., Волянський С. В., Подоструєць Л. О.

Державний університет інтелектуальних технологій і зв’язку

Ключові слова: вимірювання геометричних величин, комп'ютерна графіка, шкала ноніуса, autocad, autolisp.

У сфері інформаційно-вимірювальних технологій точність вимірювань відіграє головну роль у забезпеченні якості продукції та процесів. Для

Рисунок А.4 – Текст тези (3)

ДОДАТОК Б

Стаття у фаховому науковому журналі

ISSN 2664-262X Центральноукраїнський науковий вісник. Технічні науки. 2025. Вип. 11(42), ч. II

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 004.421.2:519.7 https://doi.org/10.32515/2664-262X.2025.11(42).2.45-51

К. В. Будат, Л. Я. Мартинович
Одеський національний університет імені І.І.Мечникова, м. Одеса, Україна
e-mail: budat.kateryna@stud.onu.edu.ua, larysa.yaroslavna@onu.edu.ua

Синтез трійкових двовходових логічних функцій із використанням багатопорогового логічного елементу

У статті розглянуто метод синтезу трійкових бінарних логічних функцій із використанням багатопорогового елемента. Проведено порівняльний аналіз трійкової та двійкової логіки, розроблено алгоритм синтезу трійкових функцій і реалізовано програмний модуль для автоматизації цього процесу. Результати дослідження відкривають нові можливості для розвитку трійкової логіки в енергоефективних та швидкодійних обчисленнях.

трійкова логіка, багатопороговий елемент багатозначної логіки, пороговий елемент трійкової логіки, трійковий логічний елемент, багатозначна логіка

Постановка проблеми. У сучасних умовах розвитку обчислювальної техніки та зростання вимог до швидкодії, енергоефективності та надійності цифрових систем виникає необхідність дослідження альтернативних логічних основ, зокрема багатозначної логіки. Одним із найбільш перспективних напрямів є трійкова логіка, яка передбачає використання трьох логічних станів замість двох у класичній бінарній логіці. Такий підхід дозволяє підвищити щільність представлення інформації, зменшити кількість необхідних логічних елементів та потенційно скоротити час обробки даних. Особливу увагу до цього напряму привертає те, що класичні бінарні апаратні засоби, які домінували протягом останніх десятиліть, поступово досягають своїх фізичних меж. У зв'язку з цим виникає потреба пошуку нових архітектур і логічних підходів, здатних забезпечити подальший розвиток обчислювальної техніки.

Незважаючи на потенційні переваги, синтез трійкових логічних функцій є значно складнішим у порівнянні з двійковими, з огляду на збільшену кількість можливих комбінацій входів і виходів. Зокрема, проблема автоматичного синтезу трійкових бінарних функцій потребує розробки спеціалізованих алгоритмів і програмних засобів, здатних забезпечити ефективний аналіз і побудову відповідних логічних структур.

Актуальність цієї задачі обумовлюється декількома чинниками. По-перше, трійкова логіка відкриває нові можливості для створення більш компактних і гнучких цифрових пристроїв. По-друге, вона дозволяє реалізувати нові принципи обробки інформації, які краще відповідають сучасним викликам — як з точки зору продуктивності, так і з точки зору енергоспоживання. По-третє, з наукової точки зору, дослідження в цій галузі сприяють поглибленню розуміння принципів побудови обчислювальних систем і створюють підґрунтя для майбутніх проривів у галузі електроніки та інформатики [1].

Аналіз останніх досліджень і публікацій. У зв'язку з посиленим інтересом наукової спільноти до альтернативних логічних основ, зокрема до трійкової логіки, останніми роками було опубліковано низку праць, присвячених розробці як теоретичних,

Рисунок Б.1 – Текст статті (1)

ISSN 2664-262X Central Ukrainian Scientific Bulletin. Technical Sciences. 2025. Issue 11(42), Part II

так і прикладних аспектів побудови багатозначних логічних схем. Вони охоплюють різні підходи до реалізації трійкових елементів, включаючи нанотехнології, а також спеціалізовані електронні компоненти, серед яких особливе місце займає багатопороговий логічний елемент.

У статті [2] розглядається використання Graphene Nano Ribbon Field Effect Transistor (GNRFET) для реалізації трійкових логічних та арифметичних схем. Автори підкреслюють, що GNRFET має унікальні властивості, які дають змогу контролювати порогову напругу, що є ключовим для багатозначної логіки (MVL). Вони запропонували новий підхід до проєктування трійкових логічних воріт та схем, таких як інвертори, NAND, NOR, трійковий декодер, мультиплексор і напівсуматор.

Особливо важливими є дослідження, що грунтуються на багатопороговому елементі багатозначної логіки (БПЕБЛ) [3]. Цей елемент, завдяки своїй гнучкості та можливості реалізовувати різноманітні логічні функції в залежності від порогових значень і комбінацій виходів, став основою для створення більш складних компонентів, таких як трійковий півсуматор [4], суматор [5], тригер [6] та універсальний пристрій для побудови унарних трійкових операцій [7]. В [8] показано, що такий підхід дозволяє простіше створювати різні трійкові логічні елементи, використовуючи один універсальний принцип.

Розробка програмного забезпечення для синтезу двовходових трійкових функцій, яку буде представлено в межах цієї роботи, базується саме на використанні багатопорогового елементу як структурної основи.

Загалом, аналіз сучасних наукових і технічних джерел дозволяє зробити висновок про наявність стійкої тенденції до розвитку трійкових логічних систем, де багатопороговий елемент виступає не лише як функціональна одиниця, а як ключова технологічна платформа для подальших досліджень і розробок.

Постановка завдання. Мета наукової статті полягає в розробці методу синтезу трійкових бінарних логічних функцій із використанням багатопорогового елементу. Для цього необхідно вирішити такі задачі: 1) провести порівняльний аналіз функцій трійкової і двійкової логіки; 2) розробити алгоритм синтезу трійкових бінарних логічних функцій з використанням багатопорогового елементу; 3) реалізувати програмний модуль синтезу та провести тестування роботи створеного програмного забезпечення.

Виклад основного матеріалу. Трійкові системи досліджуються ще з 1950-х років. Проте розвиток двійкових комп'ютерів, що стали домінуючими впродовж наступних десятиліть, витіснив інтерес до трійкової логіки. Останнім часом інтерес до трійкових систем знову зріс, зокрема через їхній потенціал у галузях, що потребують високої щільності обчислень і зниження енергоспоживання.

Трійкова логіка має низку переваг над двійковою. Вона потребує приблизно на 36.9% менше операцій, що підвищує загальну ефективність обчислень. Завдяки здатності кодувати більше інформації в одному логічному елементі, трійкові системи зменшують обсяг необхідної пам’яті. Також вони дозволяють створювати компактніші логічні схеми з меншою кількістю елементів і мають потенціал до зниження енергоспоживання за рахунок меншої кількості переходів між логічними станами [9]. Все це робить трійкову логіку перспективною альтернативою традиційній двійковій.

У цій роботі реалізація трійкових логічних елементів базується на використанні БПЕБЛ у його чотирьохпороговій версії з 8 струмовими перемикачами. Якщо на вході струмового перемикача (СП) присутній активний сигнал, струм формується на виході L цього СП. У протилежному випадку струм генерується на виході R. Таким чином, через

Рисунок Б.2 – Текст статті (2)

ISSN 2664-262X Центральноукраїнський науковий вісник. Технічні науки. 2025. Вип. 11(42), ч. II

комбінацію сигналів на виходах RL можливо реалізувати необхідну логічну поведінку. Ці сигнали додаються з урахуванням їхніх знакових значень [10].

Розроблений алгоритм побудови на основі чотирьохпорогового БПЕБЛ можна формалізувати у наступному вигляді:

1. Для кожної можливої комбінації вхідних змінних (a,b) визначається значення суми вхідних сигналів terlev [**Ошибка! Источник ссылки не найден.**]. Значення terlev може набувати одного з п’яти рівнів: «−−», «−», «0», «+», «++».

2. Значення функції f розбивається на три підмножини: ті комбінації (a,b), для яких f(a,b)=−; ті, де f(a,b)=0; та ті, де f(a,b)=+.

3. Згідно з таблицею відповідності між значеннями terlev та активними вихідними сигналами перемикачів, для кожного рівня функції («−», «0», «+») відбираються відповідні виходи:

- для рівня «+» — виходи, що формують позитивний сигнал при потрібних значеннях terlev (+R1 ,+R2, +L2, +L1),
- для рівня «−» — виходи, які генерують від’ємний сигнал при відповідних terlev (–R1 ,–R2, –L2, –L1),
- для рівня «0» — виходи формують нульовий рівень або відсутність активного сигналу.

4. Якщо виявляється, що обраний вихідний сигнал, який забезпечує правильне значення функції для певної комбінації вхідних змінних, водночас спотворює значення функції для іншої комбінації, вживаються заходи компенсації. У такому випадку додається додатковий вихід струмового перемикача, який активується при значенні terlev, що відповідає спотвореній комбінації, і формує протилежний логічний рівень. Таким чином забезпечується нейтралізація небажаного впливу та коректне відображення функції для всіх вхідних комбінацій.

5. Обрані виходи комбінуються, якщо на якійсь комбнації вхідних змінних вихідне значення не дорівнює значенню шуканої функції, повертаємось до пункту 4.

Для прискорення процесу синтезу трійкових логічних функцій та зниження ймовірності помилок на етапі підбору перемикачів було розроблено програмне забезпечення мовою програмування Python. головне вікно програми зображено на рисунку 1.

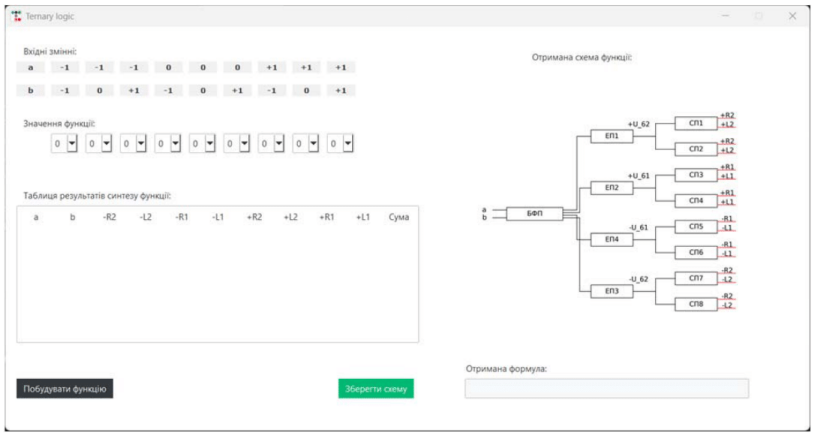

Рисунок 1 – Головне вікно програми

Джерело: розроблено авторами

Рисунок Б.3 – Текст статті (3)

ISSN 2664-262X Central Ukrainian Scientific Bulletin. Technical Sciences. 2025. Issue 11(42), Part II

Весь процес побудови функції відбувається в інтерактивному режимі та максимально спрощений для зручності користувача. На початковому етапі користувач вводить бажану функцію за допомогою інтерфейсу, де значення вибираються зі списків, що обмежують варіанти лише допустимими (–1, 0, +1). Це дозволяє уникнути помилок і забезпечити коректність введених даних.

Після введення функції програма автоматично переходить до наступного етапу — синтезу. На цьому етапі реалізується аналіз заданих значень і виконується обчислення можливих комбінацій сигналів типу R та L, які у відповідному поєднанні дозволяють реалізувати бажану логіку.

Коли синтез завершено, система автоматично формує графічну схему, яка відображає, яким чином сигнали об’єднуються для реалізації функції. Також формується таблиця синтезу, в якій детально показано, які саме сигнали були використані та яким чином на них впливає кожна комбінацію вхідних змінних. Особливу увагу в таблиці приділено сумі цих сигналів, яка повинна співпадати з початково заданими значеннями функції. Це дозволяє не лише побачити кінцевий результат, але й проаналізувати проміжні етапи синтезу та переконатись у правильності виконаних обчислень.

Остаточним етапом є виведення результатів у зручній для користувача формі. У спеціальному полі відображається аналітична формула, яка описує побудовану функцію через комбінації виходів типу R та L. Крім того, користувач має можливість зберегти побудовану структурну схему, що відкриває перспективи для подальшого використання результатів як у навчальних, так і в дослідницьких цілях. Такий підхід дозволяє значно пришвидшити процес проєктування логічних функцій і мінімізувати ймовірність помилок, що зазвичай виникають при ручному моделюванні.

Для перевірки коректності роботи програмного забезпечення було проведено тестування шляхом синтезу наступних функцій:

1. Кон’юнкція - виконується як операція мінімуму між двома змінними, аналогічно до двійкової операції «І». Вона визначає результат як найменший ступінь істинності вхідних аргументів.

2. «Приймати все» - формує визначений результат у випадку, якщо хоча б один із аргументів має відоме значення, і залишає результат невідомим лише при суперечності або повній відсутності інформації.

3. Консенсус - вважає результат істинним лише за умови повної узгодженості всіх вхідних значень. У разі будь-яких розбіжностей результат визначається як невідомий.

Кожна з функцій була реалізована за допомогою запропонованого інтерфейсу. Значення отриманих функції повністю відповідають таблицям істинності. Це підтверджує правильність роботи алгоритму і логіку побудови структурних схем. Результати синтезу подані на рисунках 4–6, де зображено результат роботи програми, а саме: побудовані схеми для кожної з функцій, таблиці синтезу, а також наведено відповідні аналітичні формули, що описують їх реалізацію через комбінації R та L сигналів.

Висновки. У ході дослідження запропоновано алгоритм синтезу трійкових бінарних логічних функцій із використанням чотирьохпорогового багатозначного елемента. Такий підхід дозволяє значно спростити побудову логічних схем, зменшити кількість необхідних апаратних ресурсів і підвищити ефективність обробки даних.

Рисунок Б.4 – Текст статті (4)

ISSN 2664-262X Центральноукраїнський науковий вісник. Технічні науки. 2025. Вип. 11(42), ч. II

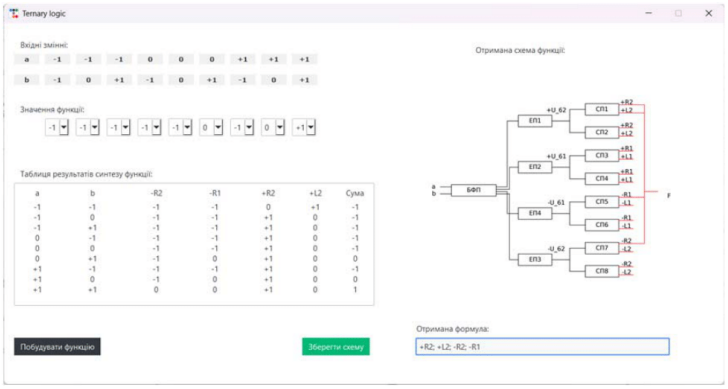

Рисунок 4 – Побудова сильної кон'юнкції

Джерело: розроблено авторами

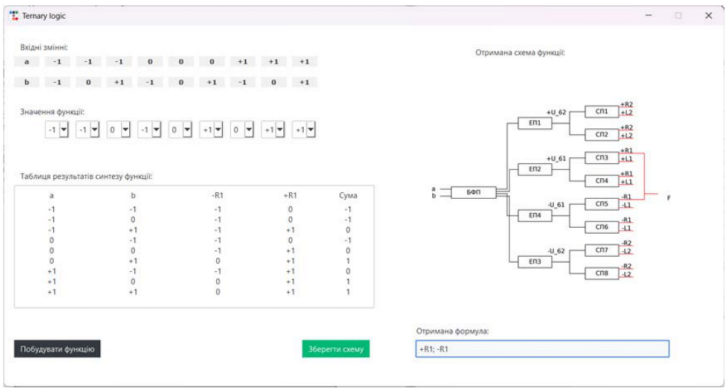

Рисунок 5 – Побудова функції «Приймати все»

Джерело: розроблено авторами

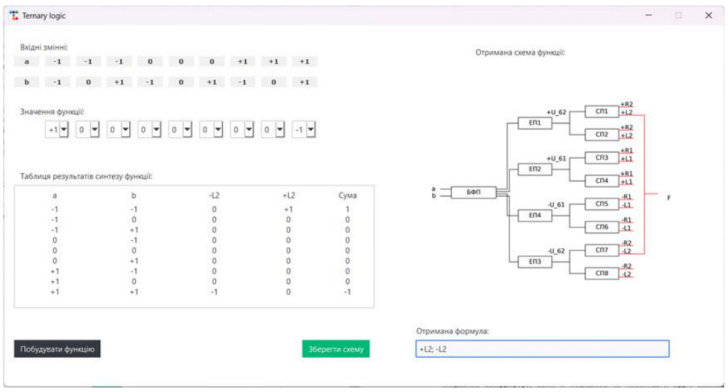

Рисунок 6 – Побудова консенсусу

Джерело: розроблено авторами

Рисунок Б.5 – Текст статті (5)

Застосування багатопорогових елементів відкриває перспективи для створення компактніших і швидкодіючих логічних систем.

Окремою перевагою є створене програмне забезпечення, яке автоматизує процес синтезу логічних функцій. Воно дозволяє користувачеві отримати як графічне зображення схеми, так і відповідну аналітичну формулу, що полегшує подальше теоретичне дослідження або практичне впровадження. Тестування програми підтвердило її функціональність і правильність реалізованих алгоритмів.

Проведений порівняльний аналіз трійкової та двійкової логіки показав, що трійкова логіка має низку важливих переваг, зокрема — потенціал до зменшення складності схем і зростання швидкодії.

Отримані результати мають як теоретичне, так і прикладне значення. Вони можуть бути використані як основа для подальших досліджень у галузі багатозначної логіки та розроблення нових обчислювальних архітектур. Запропоновані рішення сприятимуть розвитку цифрових технологій, орієнтованих на енергоефективність та підвищену швидкодію.

Список літератури

1. Gunchenko, Y., Shugailo, Y., Bercov, Y., Martynovych, L. Analysis of the current state of the elements of ternary logic. *Collection of Scientific Papers of Military Institute of Taras Shevchenko National University of Kyiv*. 2022. № 76. С. 88–101. DOI: https://doi.org/10.17721/2519-481x/2022/76-08 (дата звернення: 15.04.2025).
2. Sandhie, Z. T., Ahmed, F. U., Chowdhury, M. H. Design of ternary logic and arithmetic circuits using GNRFET. *IEEE Open Journal of Nanotechnology*. 2020. Т. 1. С. 77–87. DOI: https://doi.org/10.1109/ojnano.2020.3020567 (дата звернення: 15.04.2025).
3. Багатопороговий елемент багатозначної логіки : пат. 118735 Україна : МПК H03K19/00. № u201701717 ; заявл. 23.03.2017 ; опубл. 28.08.2017, Бюл. № 16.
4. Трійковий півсуматор на основі багатопорогового елемента багатозначної логіки : пат. 122996 Україна : МПК H03K19/00, G06F7/00. № u201706115 ; заявл. 16.06.2017 ; опубл. 12.02.2018, Бюл. № 3.
5. Трійковий повний однорозрядний суматор : пат. 139770 Україна : МПК H03K19/00. № u201905060 ; заявл. 13.05.2019 ; опубл. 27.01.2020, Бюл. № 2.
6. Трійковий RS-тригер : пат. 149386 Україна : МПК H03K19/00. № u202104077 ; заявл. 13.07.2021 ; опубл. 11.11.2021, Бюл. № 45.
7. Універсальний пристрій для побудови трійкових унарних операцій : пат. 130182 Україна : МПК H03K19/00. № u201806401 ; заявл. 08.06.2018 ; опубл. 26.11.2018, Бюл. № 22.
8. Левчук, В., Гунченко, Ю., Кузніченко, С. та ін. Концепція побудови логічних і арифметичних пристроїв для багатозначних логік. *Інформаційні технології та комп'ютерне моделювання : матеріали Міжнар. наук.-практ. конф.* (Івано-Франківськ, 14–18 трав. 2018 р.). Івано-Франківськ, 2018. С. 216–219. URL: https://journal.comp-sc.if.ua/test/index.php/ITCM/article/view/428/220 (дата звернення: 15.04.2025).
9. Zahoor, F., Ahmad, I., Kazim, R. et al. Design implementations of ternary logic systems: A critical review. *Results in Engineering*. 2024. Article ID: 102761. DOI: https://doi.org/10.1016/j.rineng.2024.102761 (дата звернення: 15.04.2025).
10. Martynovych, L., Gunchenko, Y., Shugailo, Y., Bercov, Y. Design and synthesis of ternary logic elements. *Computer Systems and Information Technologies*. 2022. № 4. С. 52–60. DOI: https://doi.org/10.31891/csit-2022-4-8 (дата звернення: 15.04.2025).

References

1. Gunchenko, Y., Shugailo, Y., Bercov, Y., & Martynovych, L. (2022). Analysis of the current state of the elements of ternary logic. *Zbirnyk naukovykh prats' Viys'kovoho instytutu Kyyivs'koho natsional'noho universytetu imeni Tarasa Shevchenka*, (76), 88–101. https://doi.org/10.17721/2519-481x/2022/76-08
2. Sandhie, Z. T., Ahmed, F. U., & Chowdhury, M. H. (2020). Design of ternary logic and arithmetic circuits using GNRFET. *IEEE Open Journal of Nanotechnology*, *1*, 77–87. https://doi.org/10.1109/ojnano.2020.3020567
3. Gunchenko, Y. O. (2017). Multiplet threshold element of multivalued logic (Ukrainian Patent No. 118735). Odesa I. I. Mechnikov National University.

Рисунок Б.6 – Текст статті (6)

4. Gunchenko, Y. O., Lenkov, S. V., Malakhov, Y. V., Shvorov, S. A., Ustymchuk, V. V., Lukin, V. Y., Mezhuev, V. I., Lenkov, Y. S., & Levchuk, V. V. (2018). Ternary half-adder based on a multiplet threshold element of multivalued logic (Ukrainian Patent No. 122996). Odesa I. I. Mechnikov National University.
5. Gunchenko, Y. O., Lenkov, S. V., Shvorov, S. A., Mezhuev, V. I., Levchenko, A. O., Kuznichenko, S. D., Lenkov, Y. S., Nikolaenko, O. Y., Shvorov, A. S., Berkov, Y. M., & Romanenko, K. Y. (2020). Ternary full single-bit adder (Ukrainian Patent No. 139770). Odesa I. I. Mechnikov National University.
6. Gunchenko, Y. O., Glauberman, M. A., Martynovych, L. Y., Romanenko, K. Y., Mezhuev, V. I., Masliy, N. D., Shugailo, Y. B., Berkov, Y. M., & Fastykovskyi, P. P. (2021). Ternary RS flip-flop (Ukrainian Patent No. 149386). Odesa I. I. Mechnikov National University.
7. Gunchenko, Y. O., Lenkov, S. V., Shvorov, S. A., Mezhuev, V. I., Lendel, T. I., Lukin, V. Y., Zahrebniuk, V. I., Lenkov, Y. S., & Levchuk, V. V. (2018). Universal device for constructing ternary unary operations (Ukrainian Patent No. 130182). National University of Life and Environmental Sciences of Ukraine.
8. Levchuk, V., Gunchenko, Y., Kuznichenko, S., et al. (2018). The concept of constructing logical and arithmetic devices for multivalued logics. *In Informatsiini tekhnolohii ta kompiuterne modeliuvannia: Proceedings of the International Scientific and Practical Conference* (Ivano-Frankivsk, May 14–18, 2018) (pp. 216–219). Ivano-Frankivsk. https://journal.comp-sc.if.ua/test/index.php/ITCM/article/view/428/220 [in Ukrainian]
9. Zahoor, F., Ahmad, I., Kazim, R., et al. (2024). Design implementations of ternary logic systems: A critical review. *Results in Engineering*, Article ID 102761. https://doi.org/10.1016/j.rineng.2024.102761
10. Martynovych, L., Gunchenko, Y., Shugailo, Y., & Bercov, Y. (2022). Design and synthesis of ternary logic elements. *Computer Systems and Information Technologies*, (4), 52–60. https://doi.org/10.31891/csit-2022-4-8

Kateryna Budat, **Larysa Martynovych**
Odesa I. I. Mechnykov National University, Odesa, Ukraine

Software Implementation of the Synthesis of Ternary Two-Input Logic Functions

The article proposes a method for synthesizing ternary binary logic functions using a multi-threshold element, which significantly extends the capabilities of ternary logic compared to traditional binary logic. The aim of the work is to develop an algorithm for synthesizing ternary logic functions, based on the use of a multi-threshold element, which simplifies the design of logical circuits and reduces energy consumption while maintaining high data processing speed.

The study includes a comparative analysis of ternary and binary logic functions, allowing an assessment of the advantages of ternary logic, particularly its ability to process more values than binary logic, which opens up new possibilities for creating more efficient computing devices. Specifically, ternary logic can reduce the complexity of circuits and hardware resource usage, while also enhancing performance in tasks where energy efficiency is critical. An essential part of the research is the development of an algorithm for synthesizing ternary binary logic functions based on a four-threshold multi-valued element. Furthermore, a software module was implemented to automate the synthesis of ternary functions. The program not only generates a graphical representation of the logical circuit but also formulates the analytical expression corresponding to the given logical function. This significantly simplifies the design and analysis of logical systems and enhances the efficiency of computations.

Considering the high potential of ternary logic, the results of this work are significant not only for theoretical research but also for practical applications in the development of new computing devices and technologies. The further development of these ideas could contribute to the progress in computational systems working on new principles and open up new possibilities for optimizing technological processes in various fields of science and engineering.

ternary logic, multi-threshold element of multi-valued logic, threshold element of ternary logic, ternary logic element, multi-valued logic

Одержано (Received) 18.04.2025 *Прорецензовано (Reviewed) 29.04.2025*
Прийнято до друку (Approved) 02.05.2025

Рисунок Б.7 – Текст статті (7)

ДОДАТОК В

Диплом за наукову конкурсну роботу І ступеня

ДИПЛОМ

І ступеня

НАГОРОДЖУЄТЬСЯ

БУДАТ Катерина Володимирівна

студентка 4-го курсу ОР «бакалавр»,
спеціальності 123 «Комп'ютерна інженерія»

за наукову роботу

«Моделювання трійкових логічних функцій»

Науковий керівник – ***Мартинович Лариса Ярославівна*** –
старший викладач кафедри комп'ютерних систем та технологій
факультету математики, фізики та інформаційних технологій
Одеського Національного Університету імені І. І. Мечникова

Голова конкурсної комісії з
ІТ спеціальностей ФМФІТ ОНУ
імені І. І. Мечникова
доцент
Алла РАЧИНСЬКА

Голова Вченої ради ФМФІТ
ОНУ імені І. І. Мечникова
професор
Юрій НІЩУК

Затверджено рішенням Вченої Ради ФМФІТ 25.04.2025, протокол № 6

Рисунок В.1 – Диплом за наукову конкурсну роботу