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»

«Розробка та аналіз трійкових обчислювальних пристроїв для цифрових систем. Мультиплексор, лічильник»

(тема кваліфікаційної роботи українською мовою)

«Development and analysis of ternary computing devices for digital systems. Multiplexer, counter»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 123 Комп'ютерна інженерія
(код, назва спеціальності)
Освітня програма Комп'ютерна інженерія
(назва)
Єфіменко Тімур Олександрович
(прізвище, ім'я, по-батькові здобувача)

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

Рецензент 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 № __5___
протокол № __від ____.____.20___ р.

Оцінка_____/_____/_____
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК Світлана АНТОЩУК

_____ _____
(підпис) (прізвище, ім'я)

Одеса 2025

АНОТАЦІЯ

Тема кваліфікаційної роботи «Розробка та аналіз трійкових обчислювальних пристроїв для цифрових систем: Мультиплексор, лічильник».

Кваліфікаційна робота присвячена дослідженню, розробці та аналізу трійкових обчислювальних пристроїв, зокрема мультиплексора і лічильника, з використанням принципів багатозначної логіки та сучасних підходів до проектування цифрових систем.

У роботі проведено порівняльний аналіз двійкової та трійкової логік, розглянуто їхні теоретичні основи, переваги та перспективи застосування в сучасних обчислювальних системах. Детально досліджено структури двійкових обчислювальних пристроїв, таких як мультиплексор і лічильник, та їхні аналоги в трійковій логіці. Особливу увагу приділено принципам побудови трійкових логічних елементів та їх функціонуванню в складі складних схем.

Окрему увагу приділено розробці трійкових мультиплексорів та лічильників, включаючи аналіз їх функціональних особливостей, синтез логічних схем та моделювання. Результати роботи демонструють потенціал трійкової логіки для підвищення ефективності цифрових систем, зокрема в аспекті зменшення апаратної складності та підвищення швидкодії у порівнянні з традиційними двійковими аналогами.

Робота складається з чотирьох розділів на 49 сторінках, 10 рисунків, 22 таблиці, списку використаних джерел з 14 джерел.

ANNOTATION

The topic of the thesis is “Development and analysis of ternary computing devices for digital systems: multiplexer, counter.”

The thesis is devoted to the research, development, and analysis of ternary computing devices, in particular multiplexers and counters, using the principles of multi-valued logic and modern approaches to the design of digital systems.

The thesis provides a comparative analysis of binary and ternary logic, considers their theoretical foundations, advantages, and prospects for application in modern computing systems. The structures of binary computing devices, such as multiplexers and counters, and their analogues in ternary logic are studied in detail. Particular attention is paid to the principles of constructing ternary logic elements and their functioning in complex circuits.

Special attention is paid to the development of ternary multiplexers and counters, including analysis of their functional features, synthesis of logic circuits, and modeling. The results of the work demonstrate the potential of ternary logic to improve the efficiency of digital systems, particularly in terms of reducing hardware complexity and increasing performance compared to traditional binary analogues.

The work consists of four sections on 49 pages, 10 figures, 22 tables, and a list of 14 sources used.

ЗМІСТ

стор.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

БПЕБЛ – багатопороговий елемент багатозначної логіки

БФП – блок формування порогів

ЕП – емітерний повторювач

СП – струмовий перемикач

ДДНФ – досконала диз’юнктивна нормальна форма

ШІ – штучний інтелект

ВСТУП

Сучасні цифрові системи потребують постійного вдосконалення з метою підвищення продуктивності, зменшення енергоспоживання та зниження апаратної складності. Традиційна двійкова логіка, яка лежить в основі більшості електронних пристроїв, досягає меж своїх можливостей у контексті мінімізації розмірів та енергоефективності. У зв’язку з цим актуальним стає пошук альтернативних підходів до логічного подання інформації, серед яких багатозначна логіка, зокрема трійкова, привертає дедалі більшу увагу.

Ідеї багатозначної логіки розвиваються ще з середини XX століття, проте лише з розвитком мікроелектроніки стало можливим створення фізично реалізованих трійкових схем. Існує низка наукових досліджень, присвячених моделюванню та аналізу трійкових логічних елементів, однак проблема масового впровадження залишається відкритою через низку технічних складнощів, зокрема відсутність стандартних елементів, складність реалізації трьох стійких станів, та потребу адаптації під існуючу інфраструктуру.

Метою кваліфікаційної роботи є розробка та аналіз трійкових обчислювальних пристроїв як мультиплексора та лічильника з використанням багатопорогової логіки для підвищення ефективності цифрових систем. Об’єктом дослідження є цифрові обчислювальні системи, що використовують логічні елементи. Предметом дослідження виступають принципи побудови, архітектура та функціональні особливості трійкових логічних пристроїв у порівнянні з їхніми двійковими аналогами.

1 ТЕОРЕТИЧНІ ОСНОВИ ДВІЙКОВОЇ ТА ТРІЙКОВОЇ ЛОГІКИ

1.1 Основи та принципи двійкової логіки

Двійкова логіка є фундаментальним принципом, що становить основу функціонування абсолютної більшості сучасних електронних цифрових систем. Її назва походить від використання лише двох дискретних станів, або значень: «0» (логічний нуль або хибність) і «1» (логічна одиниця або істина). Цей двійковий підхід ідеально відповідає фізичним реалізаціям в електронних схемах, де ці стани можуть бути представлені різними рівнями напруги (наприклад, низький рівень напруги для «0» і високий для «1»), різними рівнями струму, або станами фізичних перемикачів (відкрито/закрито). Простота та чіткість цих двох станів робить їх ідеальними для надійного та швидкого опрацювання інформації в електронних пристроях.

Двійкова логіка базується на використанні двійкових змінних та певного набору логічних операцій. Двійкові змінні можуть набувати лише одного з двох значень: «0» або «1». Основними логічними операціями є І (AND), АБО (OR) та НІ (NOT), кожна з цих операцій повертає двійковий результат. Важливо розрізняти двійкову логіку від двійкової арифметики, хоча операції І та АБО можуть нагадувати множення та додавання відповідно. У двійковій арифметиці змінна може містити довші числові значення, тоді як у логіці вона завжди обмежується лише двома станами: «0» або «1». Наприклад, у двійковій арифметиці 1 + 1 = 10, а в двійковій логіці 1 АБО 1 = 1 [1, 2].

Принцип роботи двійкової логіки полягає у перетворенні будь-якої інформації та логічних операцій у систему, яка оперує лише двома взаємовиключними станами: «0» і «1». Ці два стани є основою для представлення даних, виконання обчислень та керування в цифрових системах. Ключовим моментом, який дозволив двійковій логіці стати основою сучасної електроніки, є її проста та надійна фізична реалізація. В електронних схемах стани «0» і «1» представляються різними, чітко розрізнюваними

фізичними величинами [3].

Переваги двійкової логіки:

1) надійність та стійкість до перешкод; використання лише двох чітко розрізнюваних станів («0» і «1») робить цифрові схеми дуже стійкими до електричного шуму, коливань напруги та інших перешкод; великий «запас» між двома станами дозволяє легко їх розрізняти, забезпечуючи високу надійність передачі та обробки даних [2];

2) простота проектування та фізичної реалізації; двозначна природа двійкової логіки значно спрощує проектування логічних елементів та інтегральних схем; її можна легко реалізувати за допомогою простих електронних компонентів, таких як транзистори, які працюють як перемикачі;

3) легкість масового виробництва; простота конструкції цифрових схем на основі двійкової логіки сприяє їхньому легкому та економічному масовому виробництву [1];

4) ідеальна відповідність булевій алгебрі; двійкова логіка є прямою реалізацією булевої алгебри, що забезпечує чітку математичну основу для проектування та аналізу цифрових схем; це дозволяє ефективно реалізовувати логічні операції в процесорах та пам’яті;

5) висока швидкість обробки; простота кожного логічного елемента дозволяє досягати дуже високих частот перемикання, що забезпечує швидке виконання обчислень.

Недоліки двійкової логіки:

1) низька інформаційна щільність; для представлення складних даних або великих чисел двійкова логіка вимагає використання великої кількості бітів; це призводить до збільшення кількості необхідних логічних елементів і проводів, що відповідно збільшує розмір схеми, енергоспоживання та апаратні ресурси [4];

2) неефективність для аналогових та нечітких даних; двійкова логіка є дискретною за своєю природою; робота з аналоговими сигналами або даними, що мають невизначений або нечіткий характер, вимагає додаткових

перетворень (аналогово-цифрових та цифро-аналогових), що ускладнює схеми та може знижувати загальну швидкість обробки;

3) обмеження масштабування та енергоефективності; зі зростанням обсягів інформації та потребою у вищих обчислювальних потужностях, двійкова логіка призводить до експоненційного збільшення розрядності; це обмежує масштабування систем, особливо з огляду на строгі вимоги до енергоефективності та тепловиділення, оскільки більше бітів означає більше перемикань і, відповідно, більше споживання енергії;

4) жорсткість у представленні невизначеності; у двійковій логіці відсутній природний спосіб представлення «невідомих» або «проміжних» станів; будь-яка невизначеність мусить бути або істиною, або хибою, що може не відповідати реальним сценаріям [5].

1.2 Основи та принципи трійкової логіки

Трійкова логіка є першою історично багатозначною логікою, що розширює класичну двозначну систему («0» і «1») за рахунок введення третього значення. Це значення може інтерпретуватися як: «так» («1») – істина, «ні» («–1») – хиба, «невідомо» («0») – проміжний стан. У деяких варіантах таку логіку називають частковою (або симетричною), бо вона допускає стан невизначеності. На відміну від звичної логіки з законом виключеного третього, трійкова логіка не суперечить формальній логіці, а розглядається як її розширення. Вона дозволяє більш природно відображати реальні ситуації, де не завжди існує однозначна відповідь.

У трійковій логіці використовуються власні одиниці вимірювання інформації – аналогічні біту та байту в двійковій логіці. У трійковій логіці основними одиницями інформації є трит і трайт, які є аналогами біта та байта у двійковій системі. Трит – це мінімальний трійковий розряд, що може набувати одного з трьох значень: «–1», «0» або «+1». Завдяки трьом можливим станам, один трит містить приблизно 1,585 біта інформації. Це забезпечує

вищу щільність запису в порівнянні з бітами, де можливі лише два стани. Трайт, у свою чергу, складається з шести тритів і виконує роль повноцінної адресованої одиниці трійкової пам’яті, подібно до байта у двійковій. Він може представляти до 729 унікальних значень (3^6), що значно перевищує можливості байта (256 значень). Такий широкий діапазон дозволяє ефективно працювати з великими обсягами даних, зокрема з від’ємними числами, оскільки значення можуть розташовуватися симетрично від –364 до +364. Застосування тритів і трайтів дає змогу створювати компактніші та швидкодіючі системи з більшою інформаційною ємністю [9,10].

Принципи трійкової логіки базуються на використанні трьох логічних станів замість двох, як у класичній двійковій логіці. Змінна в трійковій системі може набувати значень «–1», «0» або «+1», що дозволяє точніше описувати невизначені або проміжні стани. Для реалізації трійкових операцій використовуються специфічні схеми обробки сигналів. Це може бути реалізовано, наприклад, через алгебраїчне додавання струмів або напруг, що відповідають кожному логічному стану. Поріг спрацьовування визначає логічний результат на виході, дозволяючи розрізняти три стани. Завдяки більшій кількості логічних станів, трійкова логіка дозволяє зменшити кількість розрядів у обчисленнях, що є критично важливим для мінімізації розміру схем та зменшення складності зв'язків. Це також сприяє підвищенню щільності кодування інформації та може спростити реалізацію деяких логічних функцій, які в двійковій системі потребують складніших комбінацій вентилів [6, 7].

Переваги трійкової логіки:

1) вища інформаційна щільність; один трит може містити більше інформації (приблизно 1,585 біта) порівняно з одним бітом; це дозволяє більш компактно кодувати дані, що потенційно зменшує фізичний розмір пристроїв та підвищує ефективність зберігання та передачі інформації. Трайт (6 тритів) може представляти 729 унікальних значень, тоді як байт (8 бітів) – лише 256;

2) природна робота з від'ємними числами та симетрична система

кодування; трійкова логіка, зокрема симетрична (з використанням «-1», «0», «+1»), природно підтримує представлення від'ємних чисел, оскільки діапазон значень симетричний відносно нуля; це може спростити арифметичні операції з від'ємними числами порівняно з двійковою системою, де часто використовуються додаткові коди (наприклад, доповнення до двох);

3) краще відповідає людській логіці та перспективність для ШІ та нейромереж; наявність третього стану «невідомо» дозволяє трійковій логіці більш адекватно моделювати реальні ситуації, де інформація може бути неповною або невизначеною; це робить її перспективною для застосувань у штучному інтелекті, експертних системах та нейромережах, де часто потрібно працювати з неточними або частковими знаннями [6].

Недоліки трійкової логіки:

1) відсутність масового виробництва трійкових елементів; на відміну від двійкових транзисторів, які виробляються мільярдами, немає усталеної інфраструктури для масового виробництва трійкових логічних елементів [7];

2) складність реалізації тристабільних схем; фізична реалізація трійкових вентилів є значно складнішою порівняно з двійковими; потрібно розрізняти три чіткі стани напруги або струму, що вимагає більш точних компонентів та більшої чутливості до шуму [6];

3) підвищене енергоспоживання, хоча теоретично трійкова логіка може бути більш енергоефективною для певних завдань через меншу кількість елементів, на практиці, у сучасних експериментальних реалізаціях, складність розрізнення трьох рівнів та необхідність точніших компонентів часто призводить до більшого енергоспоживання на один логічний елемент.

4) проблеми сумісності з існуючими двійковими системами; переважна більшість сучасних комп'ютерних систем, програмного забезпечення та стандартів побудовані на двійковій логіці; інтеграція трійкових компонентів вимагатиме розробки додаткових перетворювачів, що додає складність та може знижувати продуктивність;

5) відсутність стандартизованих рішень та інфраструктури; відсутність

широко прийнятих стандартів для трійкових архітектур та інструментів розробки є серйозним бар'єром для її впровадження.

1.3 Порівняння двійкової та трійкової логіки та перспективи розвитку

Фундаментальна відмінність між двійковою та трійковою логікою полягає у їхній здатності представляти інформацію, що безпосередньо впливає на архітектуру обчислювальних систем та їхнє застосування. Двійкова логіка, що є основою сучасної цифрової електроніки, базується на використанні лише двох станів – «0» та «1». Її домінування зумовлене простотою фізичної реалізації: ці два стани легко розрізняються в електронних схемах, забезпечуючи високу надійність, стійкість до перешкод та зручність масового виробництва [7].

Однак, простота двійкової логіки має свої обмеження. Для представлення складних даних або великих числових значень їй потрібна значна кількість бітів, що призводить до зростання апаратних ресурсів, збільшення енергоспоживання та обмежень у масштабуванні систем. Крім того, двійкова логіка є неефективною для роботи з аналоговими сигналами або нечіткими, невизначеними даними, оскільки їй бракує природного способу їхнього представлення. Це вимагає додаткових перетворень, які ускладнюють систему та сповільнюють її роботу.

На противагу цьому, трійкова логіка розширює концепцію логічних станів, вводячи третє значення – «невідомо» або «0», до звичних «істини» («+1») та «хиби» («–1»). Ця здатність оперувати невизначеністю дозволяє трійковій логіці більш гнучко та природно відображати реальні ситуації, де однозначна відповідь відсутня. Такий підхід значно підвищує інформаційну щільність: один трит, що є мінімальним трійковим розрядом, вміщує приблизно 1,585 біта інформації. Відповідно, трайт, що складається з шести тритів, може кодувати 729 унікальних значень, тоді як стандартний двійковий

байт – лише 256. Це не лише дозволяє компактніше зберігати дані, а й спрощує роботу з від'ємними числами завдяки симетричній системі кодування [8, 9].

Порівнюючи ці дві парадигми, трійкова логіка теоретично обіцяє зменшення кількості логічних елементів для реалізації певних функцій та прискорення операцій порівняння завдяки можливості безпосередньо розрізняти «більше», «менше» та «дорівнює». Її здатність працювати з невизначеністю робить її особливо привабливою для таких галузей, як ШІ та нейромережі.

Однак, незважаючи на ці багатообіцяючі теоретичні переваги, трійкова логіка стикається зі значними практичними викликами. Відсутність масового виробництва трійкових елементів, складність їхньої фізичної реалізації та підвищене енергоспоживання у поточних експериментальних зразках створюють серйозні бар'єри.

Перспективи розвитку трійкової логіки тісно пов'язані з подоланням цих технологічних та інфраструктурних перешкод. Хоча двійкова логіка, ймовірно, залишатиметься домінуючою у більшості застосувань у найближчому майбутньому, трійкові системи можуть знайти своє місце у нішевих, але критично важливих сферах. Це може бути розробка спеціалізованих процесорів для ШІ, квантових обчислень або високоінтегрованих сенсорних систем, які вимагають ефективної обробки багатозначних даних. Подальші дослідження у матеріалознавстві та архітектурах можуть розкрити повний потенціал трійкової логіки, дозволивши їй доповнити, а у деяких аспектах і перевершити можливості двійкових систем.

2 АНАЛІЗ АНАЛОГІВ СТРУКТУР ДВІЙКОВИХ ЕЛЕМЕНТІВ

2.1 Двійковий мультиплексор

Мультиплексор є комбінаційним функціональним вузлом, що належить до пристроїв комутації цифрової інформації. Його основна функція полягає у спрямуванні одного з декількох інформаційних входів на єдиний вихід. Мультиплексори завжди оснащені кількома інформаційними входами, а також адресними входами, які визначають, який саме вхідний сигнал буде обраний. Додатково, вони можуть мати вхід дозволу мультиплексування, також відомий як стробуючий вхід. Цей пристрій виконує підключення одного з вхідних каналів до вихідного під дією керуючого (адресного) слова. Розрядність каналів може бути різною, а для комутації багаторозрядних слів використовуються однорозрядні мультиплексори. Зв'язок між кількістю інформаційних (n) та адресних (m) входів визначається співвідношенням $n \leq 2^m$. Залежно від кількості інформаційних та адресних входів розрізняють повні та неповні мультиплексори: якщо $n=2^m$, мультиплексор називається повним, а якщо $n<2^m$, то він є неповним.

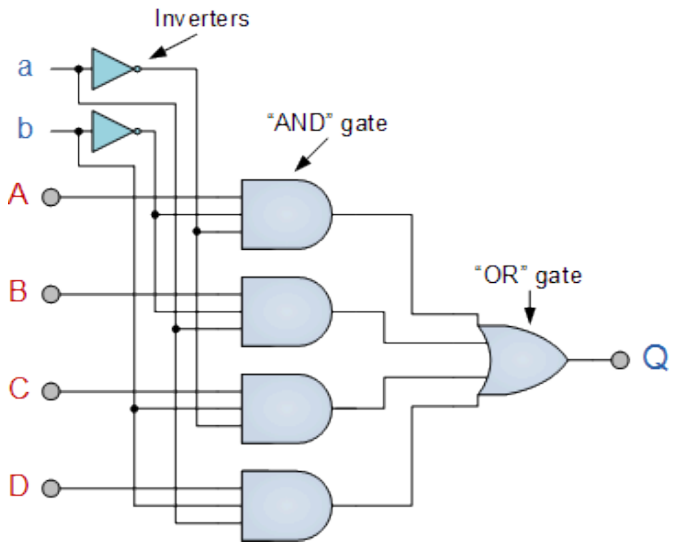

Рисунок 2.1.1 – Схема двійкового мультиплексора

Надана схема ілюструє реалізацію мультиплексора «3 до 1» (рис. 2.1.1). Вона включає чотири вхідні лінії даних – A, B, C, D, які є незалежними інформаційними входами, з яких буде обраний один для передачі на вихід Q. Вибір здійснюється за допомогою двох керуючих (адресних) входів – a та b. Важливо, що для формування всіх можливих комбінацій адрес використовуються як прямі, так і інвертовані версії цих керуючих сигналів.

Принцип роботи цієї схеми полягає в тому, що керуючі входи a та b формують двійковий адресний код. Цей код активує лише один з чотирьох вентилів «AND» через відповідні комбінації прямих та інвертованих керуючих сигналів. Коли певний вентиль «AND» активований, він пропускає відповідний вхід даних (A, B, C або D) на свій вихід. Оскільки лише один вентиль «AND» може бути активований одночасно для даної комбінації керуючих сигналів, лише один сигнал даних буде поданий на вхід вентиля «OR». Вентиль «OR» просто передає цей активований сигнал на єдиний вихід Q див. табл. 2.1.1 [10, 11].

Таблиця 2.1.1 – Таблиця істинності двійкового мультиплексора

Адресні входи		Вихід
a	b	
0	0	A
0	1	B
1	0	C
1	1	D

Таким чином, якщо керуючі входи a=0 та b=0, вихід Q буде дорівнювати A. При a=0 та b=1, вихід Q дорівнюватиме B. Якщо a=1 та b=0, вихід Q буде C. A при a=1 та b=1, вихід Q дорівнюватиме D.

2.2 Двійковий лічильник на основі Т-тригера

Лічильник – це послідовна схема, утворена каскадним з’єднанням Т-тригерів, призначена для підрахунку вхідних імпульсів, які можуть надходити як періодично, так і в довільному порядку (рис. 2.2.1). Амплітуда та тривалість

цих імпульсів мають відповідати технічним вимогам мікросхем, що застосовуються. Кількість Т-тригерів визначає розрядність лічильника (n). Кожен імпульс змінює стан лічильника, який зберігається до наступного імпульсу, а виходи тригерів $Q_n,...,Q_1$ відображають поточне значення лічби. З надходженням імпульсів лічильник послідовно переходить між станами у встановленому порядку. Кількість станів, що використовуються, називається модулем лічби $K_{лч}$. Один зі станів є початковим, і після підрахунку $K_{лч}-1$ імпульсів лічильник повертається до нього [12, 13].

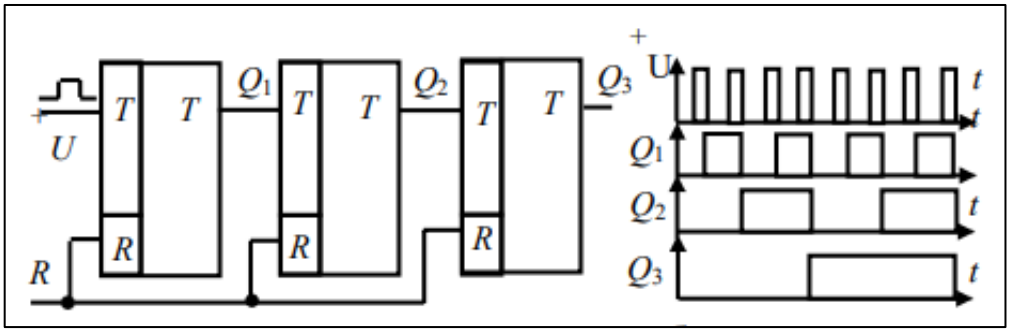

Рисунок 2.2.1 – Схема асинхроного двійкового лічильника на основі Т-тригерів та його часова діаграма

Принцип роботи двійкового лічильника, особливо того, що базується на Т-тригерах, ґрунтується на послідовному перемиканні станів його складових тригерів у відповідь на вхідні імпульси. Кожен вхідний імпульс змінює стан лічильника, який зберігається до надходження наступного імпульсу. Значення виходів тригерів лічильника відображають результат лічби.

Кожен Т-тригер у лічильника працює як дільник частоти на два, перемикаючи свій стан (з «0» на «1» або з «1» на «0») при надходженні керуючого імпульсу (як правило, по фронту тактового сигналу), якщо його вхід Т активний (логічна «1»). У синхронного лічильника всі тригери отримують спільний тактовий сигнал, що дозволяє їм змінювати стан одночасно. Щоб забезпечити двійковий підрахунок, вихід молодшого розряду (Q0) подається на вхід Т наступного тригера (Т1), вихід Q1 подається на Т2, і

так далі. Це призводить до того, що кожен наступний тригер перемикається вдвічі рідше за попередній, ефективно ділячи частоту вхідного сигналу і формуючи двійковий код.

По мірі надходження вхідних імпульсів, лічильник послідовно перебирає свої стани у визначеному для даної схеми порядку. Довжина списку цих станів, що використовуються, називається коефіцієнтом (модулем) лічби $K_{лч}$. Один зі станів обирається за початковий, і після підрахунку $K_{лч}-1$ імпульсів лічильник повертається у початковий стан.

Для 3-розрядного двійкового лічильника коефіцієнт лічби $K_{лч}$ дорівнює $2^3=8$. Це означає, що лічильник буде перебирати 8 унікальних станів (від 000_2 до 111_2), після чого повернеться до початкового стану 000 див. табл. 2.2.1.

Таблиця 2.2.1 – Таблиця істинності двійкового 3-х розрядного лічильника

Такти	Q2 (старший розряд)	Q1	Q0 (молодший розряд)	Десяткове значення
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

На основі Т-тригерів двійковий лічильник є важливим елементом цифрових систем, що забезпечує ефективний підрахунок імпульсів для реалізації таймерів, частотомірів, систем керування виробничими процесами, генерації адрес пам’яті, послідовного перемикання станів у кінцевих автоматах та передачі даних у послідовних інтерфейсах, сприяючи надійній роботі процесорів і комунікаційних пристроїв.

2.3 Двійковий D-тригер

Тригер – це базовий елемент цифрової електроніки, який використовується для зберігання одного біта інформації. Він є двостабільним пристроєм, тобто має два стійкі стани: логічну «1» та логічний «0». Перехід між цими станами відбувається під впливом вхідних сигналів. Тригери належать до елементів послідовної логіки, де вихід залежить не лише від поточного вхідного сигналу, а й від попереднього стану системи.

D-тригер – це тип тригера, що має один інформаційний вхід D, один тактовий вхід C і два виходи: Q та $\overline{Q}$ (інверсний). Основною функцією D-тригера є збереження значення входу D у момент активного фронту тактового сигналу (рис. 2.3.1).

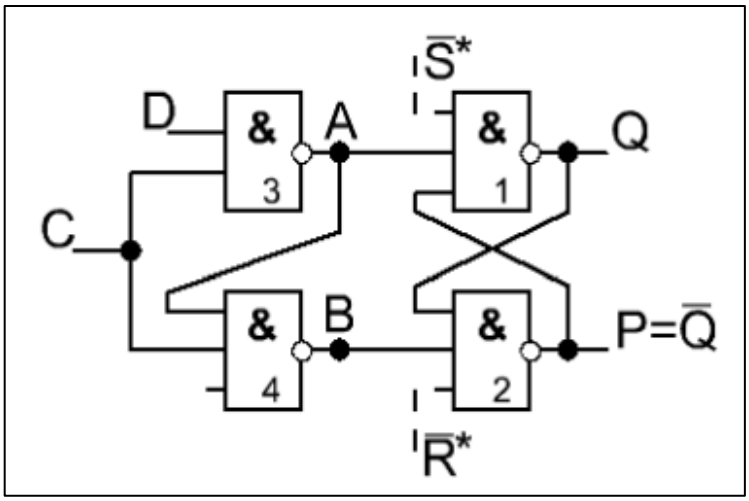

Рисунок 2.3.1 – Схема D-тригера

Цей тригер часто називають тригером затримки, оскільки він реалізує передачу вхідного сигналу на вихід із затримкою на один такт. Це робить його ключовим елементом у побудові регістрової пам’яті, лічильників, зсувних регістрів тощо [14].

Принцип дії D-тригера полягає в тому, що: під час позитивного фронту тактового імпульсу (C) значення з входу D записується на вихід Q; в інші моменти (тобто за відсутності активного фронту) – стан тригера не

змінюється, зберігаючи попереднє значення. Це дозволяє забезпечити синхронізовану передачу інформації, виключаючи виникнення нестабільних станів див. табл. 2.3.1.

Таблиця 2.3.1 – Таблиця істинності двійкового D-тригера

C	D	Q_n	Режим роботи
0	X	Q_{n-1}	Зберігання
1	0	0	Передача нуля
1	1	1	Передача одиниці

D-тригер широко застосовується в цифрових системах завдяки своїй здатності синхронно зберігати та обробляти один біт інформації, що робить його ключовим елементом у регістрах для тимчасового збереження багаторозрядних даних у мікроконтролерах, лічильниках для підрахунку імпульсів у таймерах чи частотомірах, а також у синхронізаторах для узгодження сигналів із різними частотами в комунікаційних системах; крім того, він використовується в регістрах зсуву та оперативній пам’яті для передачі та зберігання даних, забезпечуючи надійну роботу цифрових пристроїв, таких як процесори, пам’ять і системи зв’язку.

3 ПОБУДОВА ТРІЙКОВИХ ПРИСТРОЇВ НА ОСНОВІ БАГАТОПОРОГОВОГО ЕЛЕМЕНТУ БАГАТОЗНАЧНОЇ ЛОГІКИ

3.1 Теоретичні основи створення трійкових елементів за допомогою багатопорогової логіки

Багатопороговий елемент багатозначної логіки (БПЕБЛ) – це запропонована схема, яка може використовуватися для побудови інших елементів багатозначних логік, включаючи трійкову (рис. 3.1.1). Він розроблений для подолання недоліків існуючих трійкових елементів. БПЕБЛ є струмовим елементом, що підтримує алгебраїчне додавання вхідних струмів. Цей елемент може застосовуватися для логік будь-якої значності. Використання БПЕБЛ дозволяє отримати вигідні рішення, що дозволять побудувати пристрої багатозначної обчислювальної техніки, а також піднести обчислювальні системи на новий рівень.

Принцип роботи БПЕБЛ базується на послідовній обробці струмових сигналів трьома основними блоками. На вхід БПЕБЛ надходять дискретні струмові сигнали від попередніх елементів, які можуть приймати значення +1, 0 або -1. Кількість вхідних сигналів k може бути довільною і визначається як сума сигналів, поточні значення яких «+1» (k_{+1}), «-1» (k_{-1}) та «0» (k_0): $k=k_{+1}+k_{-1}+k_0$. Залежно від алгебраїчної суми вхідних струмів, БФП формує необхідну кількість порогів, активуючи відповідні виходи. Активовані виходи БФП подають сигнали на входи ЕП. Кожен ЕП, залежно від отриманого сигналу, формує напругу, яка далі передається на відповідний СП. Таким чином, ЕП виступають в ролі буферів, що перетворюють струмові сигнали від БФП у відповідні напруги для СП. У свою чергу, СП, отримуючи напругу від ЕП, перемикаються та формують стандартний струм на одному зі своїх двох виходів (L або R). Комбінуючи вихідні струми з різних СП, можна реалізувати бажану логічну функцію. Отже, БПЕБЛ здійснює багатопорогову обробку

вхідних струмів, перетворюючи їх через порогові рівні та відповідні перемикання у вихідні струми, які можуть бути об'єднані для виконання складних логічних операцій [7].

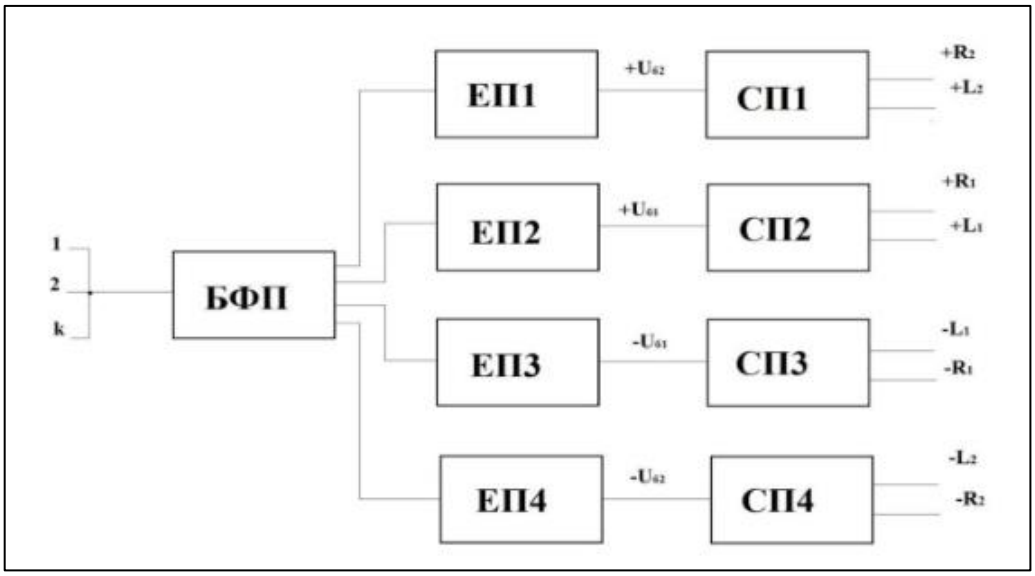

Рисунок 3.1.1 – Багатопороговий елемент багатозначної логіки

Активні ЕП формують сигнали на підключених до них струмових перемикачах (СП). Для 4-х порогової реалізації БПЕБЛ використовуються чотири емітерні повторювачі (ЕП1–ЕП4). Значення напруг, що формуються на виходах ЕП та відповідно подаються на входи струмових перемикачів, наведені в табл. 3.1.1.

Таблиця 3.1.1 – Напруги на виходах ЕП1–ЕП4

Сума вхідних струмів (terlev)	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 струмовий перемикач (СП) формує стандартний струм $I_ф$ на одному зі своїх двох виходів, залежно від вхідного сигналу. У чотирьох пороговій реалізації БПЕБЛ є чотири струмові перемикачі (СП1–СП4). Виходи струмових перемикачів можуть об'єднуватися у довільних комбінаціях для формування необхідної логіки функціонування багатопорогового елемента багатозначної логіки. У випадку, коли сигнал на вході СП активний, на виході L відповідного СП формується струм, інакше – струм формується на виході R. Значення вихідних сигналів СП1–СП4, залежно від суми вхідних струмів (terlev), наведені в таблиці 3.1.2 [7].

Таблиця 3.1.2 – Значення виходів СП1–СП4

Сума вхідних струмів	Вихідні сигнали струмових перемикачів							
	СП1		СП2		СП3		СП4	
terlev	$+R_2$	$+L_2$	$+R_1$	$+L_1$	$-R_1$	$-L_1$	$-R_2$	$-L_2$
− −	0	+	0	+	−	0	−	0
−	+	0	0	+	−	0	−	0
0	+	0	+	0	−	0	−	0
+	+	0	+	0	0	-	−	0
+ +	+	0	+	0	0	-	0	−

Сукупність вихідних сигналів та поєднання однотипних чотирьохпорогових БПЕБЛ дозволяє створювати багато різноманітних логічних та арифметичних пристроїв. Схеми пристроїв, побудованих на основі чотирьохпорогового БПЕБЛ, є економнішими за кількістю обладнання та мають більшу швидкодію.

Метод побудови трійкових пристроїв базується на використанні двовходових БПЕБЛ, які обробляють сигнали зі значеннями «-1», «0» або «+1». Комбінація цих сигналів створює до дев’яти варіантів, хоча основними є п’ять рівнів сумарного сигналу «–» від «-2» до «+2». Така гнучкість дозволяє застосовувати БПЕБЛ у створенні складних трійкових елементів, зокрема

дешифраторів, мультиплексорів, суматорів та тригерів. Принципи трійкової логіки нагадують двійкову, але зі специфічним підходом до формування вихідних сигналів, подібним до ДДНФ, де активні рівні задаються вибірково, а нуль використовується як значення за замовчуванням.

Метод відзначається високою точністю завдяки аналізу внутрішніх сигналів і застосуванню порогової логіки, що дозволяє скоротити кількість необхідних елементів, підвищуючи енергоефективність і зменшуючи розміри пристроїв. Це робить метод перспективним для сучасних обчислювальних систем і подальших досліджень.

3.2 Формування трійкових логічних функцій через порогові сигнали

Для побудови логічних елементів використовується спеціалізована структура на основі БПЕБЛ. Багатопороговий елемент складається з БФП, чотирьох порогових елементів (ЕП1-ЕП4) та чотирьох струмових перемикачів (СП1-СП4). Вихідні сигнали цих перемикачів позначаються як +R1, +L1, –R1, –L1, +R2, +L2, –R2, –L2. Ці вихідні сигнали формуються шляхом арифметичного додавання заданих вихідних сигналів з урахуванням їхніх знаків [7].

Таблиця 3.2.1 – Можливі сигнали на виходах двовходового БПЕБЛ

Сума вхідних струмів	Вихідні сигнали струмових перемикачів							
	СП1		СП2		СП3		СП4	
terlev	$+R_2$	$+L_2$	$+R_1$	$+L_1$	$-R_1$	$-L_1$	$-R_2$	$-L_2$
− −	0	+	0	+	−	0	−	0
−	+	0	0	+	−	0	−	0
0	+	0	+	0	−	0	−	0
+	+	0	+	0	0	-	−	0
+ +	+	0	+	0	0	-	0	−

Згідно з табл. 3.2.1, деякі значення вихідних струмів, виділені в помаранчевим, відповідають одновходовому елементу; при подвоєнні суми вхідних сигналів значення вихідних струмів не змінюються. Значення +L2 та –L2, виділені синім, мають значення «+» для +L2 при вхідному сигналі», та «–» для –L2 при вхідному сигналі «++».

Кількість можливих вихідних сигналів N для елемента з k=2 входами визначається як $N=((m)^m)^k$, де m – це кількість значень елемента (наприклад, m=2 для двійкового елемента). Для двійкового елемента це $N=((2)^2)^2=16$ комбінацій. Для трійкової системи кількість комбінацій значно зростає, наприклад, $((3)^3)^2=729$ можливих комбінацій. Отримання кожної з цих комбінацій є дуже складною та затратною справою, тому виникає необхідність розробити метод, який це дозволить зробити.

Пропонується система, в якій за допомогою двох БПЕБЛ розрізняються рівні кожного з вхідних сигналів. На основі цієї інформації на виході кожного БПЕБЛ формуються чотири сигнали (+R1, +L1, -R1, -L1). Для формування кожного значення вихідної функції пропонується використовувати чотири порогові БПЕБЛ, залишаючи в них тільки той поріг, що реагує на подвійні значення: +L2 реагує на сигнал «– –», а -L2 реагує на сигнал «++». Необхідно підібрати таку комбінацію внутрішніх сигналів, щоб вони давали потрібне значення лише при даних значеннях вхідних аргументів. Ця комбінація подається на пороговий елемент рівня, номер якого відповідає кількості вхідних аргументів (+-L2) [7].

3.3 Реалізація трійкового мультиплексора

Трійковий мультиплексор «3 до 1» є фундаментальним компонентом для маршрутизації даних у трійкових системах. Він дозволяє обирати один із трьох інформаційних входів (D0, D1, D2), кожен з яких може набувати значень «-1», «0» або «+1», залежно від значення адресного входу A. Практичне застосування методу побудови трійкових пристроїв на основі БПЕБЛ до

мультиплексора демонструє його ефективність у реалізації функцій вибору. Метод базується на арифметичному додаванні вхідних сигналів (інформаційних та адресного, за допомогою відповідних ваг), а потім аналізі їхньої суми через порогові рівні +L2 та -L2, що забезпечує точне перемикання та вибір потрібного інформаційного каналу в трійковій системі.

Логікою роботи трійкового мультиплексора передбачає обробку дев'яти комбінацій вхідних сигналів, оскільки вхід A може мати три стани «-1», «0» або «+1», в той час по методу описаним у підрозділі 3.2, ми будемо ставити у пару з входом A по кожному інформаційному входу тобто буде утворено по три пари A-D0, A-D1, A-D2, що буде утворювати лише дев'ять можливих комбінацій, через те що будуть повторюватися значення для інформаційних входів, та вони будуть задіяні лише при певних значеннях адресного входу A, тобто не потрібно формувати таблицю з 81 різними значеннями, бо ми не враховуємо комбінації інформаційних входів між одне одним, лише парою між одним адресним та одним інформаційним, дана логіка відображається знаком «*», тобто значення даного сигналу не впливає на вихідне значення мультиплексора. Дана логіка мультиплексора описана у табл. 3.3.1 з відображеннями кожного входу та виходу БПЕБЛ: A3, B3, C3, D3 – для адресного входу A; A0, B0, C0, D0 – для інформаційного входу D0; A1, B1, C1, D1 – для інформаційного входу D1; A2, B2, C2, D2 – для інформаційного входу D2. Ці сигнали являють собою проміжні стани, що відображають електричну поведінку схеми через рівні струмів. Вони необхідні для генерації порогових сигналів +L2 та -L2, які, в свою чергу, визначають кінцеві вихідні значення див. табл. 3.3.2.

Логіка трійкового мультиплексора ґрунтується на виборі одного з трьох інформаційних входів за допомогою одного адресного, кожні значення входів можуть набувати «-1», «0» або «+1». В залежності від вхідного значення на адресний вхід, на вихід схеми буде подано лише один інформаційний вхід, тобто якщо на адресний вхід подати «-1», то на вихід схеми буде йти лише значення першого інформаційного входу(D0) в той же момент часу.

Ефективність трійкового мультиплексора «3 до 1» та його переваги над двійковими аналогами проявляються у кількох ключових аспектах. По-перше, для вибору одного з трьох інформаційних входів трійковому мультиплексора потрібен лише один адресний вхід, оскільки цей вхід може приймати три значення («-1», «0», «+1»). На противагу цьому, двійковий мультиплексор, щоб здійснити подібний вибір (наприклад, з трьох або чотирьох входів), вимагає двох адресних входів. Це значно зменшує кількість фізичних з'єднань, спрощує розведення на друкованій платі та мінімізує необхідну площу. По-друге, один трійковий розряд здатний переносити більше інформації, ніж один двійковий біт. Це означає, що трійкові схеми потенційно можуть обробляти більший обсяг даних, використовуючи при цьому меншу кількість компонентів або зв'язків, що в кінцевому підсумку може призвести до створення більш компактних і продуктивних пристроїв.

Таблиця 3.3.1 – Таблиця істинності трійкового мультиплексора

				A				D0				D1				D2				Вихід
				A3	B3	C3	D3	A0	B0	C0	D0	A1	B1	C1	D1	A2	B2	C2	D2	
A	D0	D1	D2	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Y
–	–	*	*	0	–	+	0	0	–	+	0	*	*	*	*	*	*	*	*	–
–	0	*	*	0	–	+	0	0	–	0	+	*	*	*	*	*	*	*	*	0
–	+	*	*	0	–	+	0	–	0	0	+	*	*	*	*	*	*	*	*	+
0	*	–	*	0	–	0	+	*	*	*	*	0	–	+	0	*	*	*	*	–
0	*	0	*	0	–	0	+	*	*	*	*	0	–	0	+	*	*	*	*	0
0	*	+	*	0	–	0	+	*	*	*	*	–	0	0	+	*	*	*	*	+
+	*	*	–	–	0	0	+	*	*	*	*	*	*	*	*	0	–	+	0	–
+	*	*	0	–	0	0	+	*	*	*	*	*	*	*	*	0	–	0	+	0
+	*	*	+	–	0	0	+	*	*	*	*	*	*	*	*	–	0	0	+	+

Таблиця 3.3.2 – Вихідні комбінації для мультиплексора

Y	Вихідна комбінація
–	–L2(C3, C0)
0	
+	+L2(B3,A0,D3)
–	–L2(D3, C1, A3)
0	
+	+L2(B3, A1, C3)
–	–L2(D3,C2,B3)
0	
+	+L2(A3, A2)

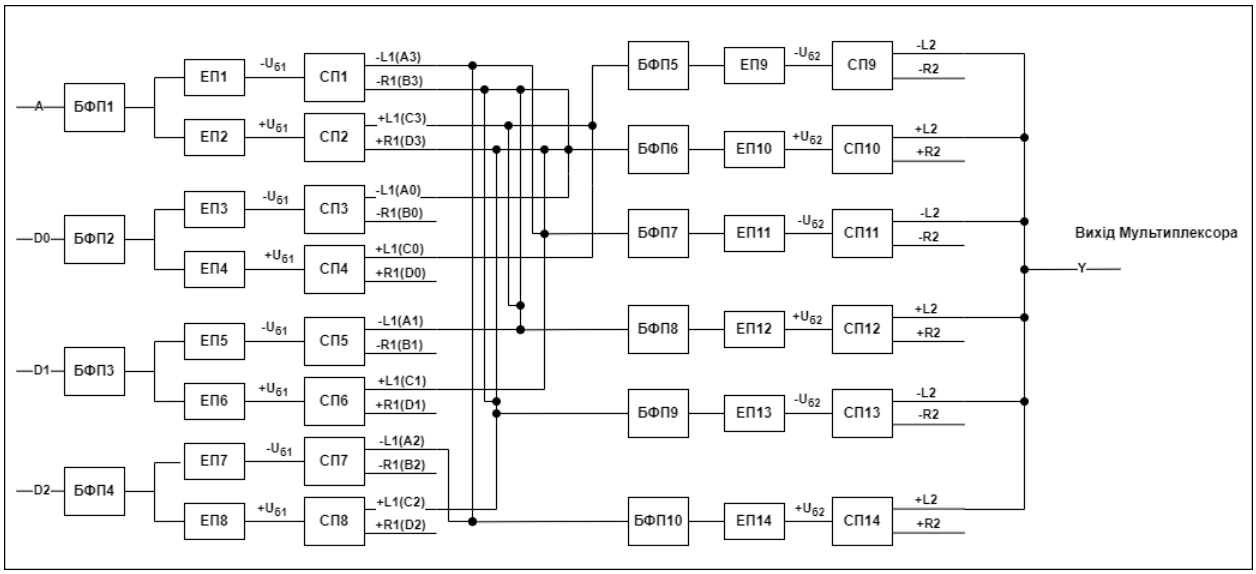

Рисунок 3.3.1 – Структурна схема трійкового мультиплексора

Схема трійкового мультиплексора складається з десяти БФП, чотири з яких основних, які відповідають за вхідні сигнали схеми та шість проміжних які формують вихідні значення мультиплексора на основі вхідних сигналів (рис. 3.3.1). Кожен з перших чотирьох БФП блоків з’єднанні з двома ЕП (ЕП1–ЕП8), які формують проміжні сигнали які надходять до наступних блоків, а саме до СП (СП1–СП8), які в свою чергу формують вихідні сигнали, що йдуть у наступну частину схеми, на основі яких будуть формуватися вихідні значення мультиплексора. Внутрішня структура побудована таким чином, що

потрібно просумувати відповідні виходи перших чотирьох БФП, які ми порахували у таблиці 3.3.2, для отримання «++» або «– –», де кожен рядок табл. 3.3.2 це окремий БПЕБЛ на який йде обрана комбінація виходів з БФП1–БФП4. Після складання кожної комбінації виходів, з БФП5–БФП10 йдуть з'єднання по одному ЕП (ЕП9–ЕП14), які формують проміжні сигнали які надходять до наступних СП (СП9–СП14) та формують вихідні значення мультиплексора, та кожне вихідне значення сумується для утворення одного єдиного виходу схеми.

3.4 Реалізація трійкового інкременту

Трійковий інкремент є базовим елементом для послідовної зміни станів у трійкових обчислювальних системах. Він змінює своє значення циклічно за схемою: «0» → «-1», «-1» → «+1», «+1» → «0», реагуючи лише на єдиний вхідний сигнал. Реалізація трійкового інкремента на основі БПЕБЛ демонструє ефективність порогової логіки в обробці багатозначних сигналів. Робота трійкового інкремента ґрунтується на додаванні вихідних значень єдиного обробленого через порогові рівні вхідного сигналу, подібно до методу реалізації мультиплексора, але без проміжних обчислень. Це забезпечує точний перехід між трьома допустимими логічними станами та дозволяє використовувати інкремент як ключовий компонент у трійкових лічильниках, генераторах послідовностей та інших автоматичних пристроях.

Логіка трійкового інкремента ґрунтується на циклічній зміні вхідного значення між трьома допустимими станами: «0», «-1» та «+1». При подачі одного сигналу інкремент зчитує його і змінює його відповідно до визначеного порядку: якщо вхід дорівнює «0», то вихід стає «-1»; якщо «-1» – переходить у «+1»; якщо «+1» – змінюється назад на «0». Суть логіки полягає в тому, що кожне нове значення визначається лише вхідним сигналом без необхідності сигналу активації чи тактового сигналу. Реалізація цієї логіки на основі БПЕБЛ передбачає аналіз вхідного сигналу через порогові рівні, що дозволяє точно

фіксувати момент переходу до наступного стану. Логіка даного інкременту описана у табл. 3.4.1 разом з вихідними комбінаціями.

Таблиця 3.4.1 – Таблиця істинності трійкового інкременту та його вихідні комбінації

Вхідні значення	+L1	+L1	-R1	Сума виходів
0	0	0	–	–
–	+	+	–	+
+	0	0	0	0

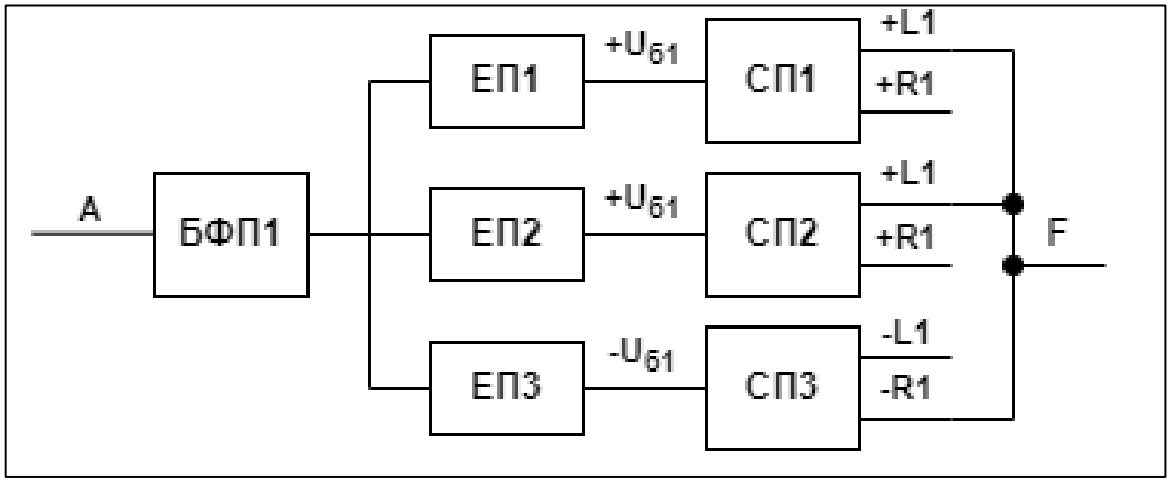

Рисунок 3.4.1 – Структурна схема трійкового інкременту

Схема трійкового інкремента реалізована на основі БФП та порогових елементів (рис. 3.4.1). Основу роботи становить блок БФП1, який приймає вхідний трійковий сигнал А та формує відповідні керуючі сигнали, що надходять до трьох елементів ЕП1–ЕП3. Кожен з ЕП аналізує свій сигнал на відповідність певному рівню та передає його до відповідного СП1–СП3.

Елементи СП виконують логічне перетворення відповідно до порогових значень +L1 та -R1, формуючи часткові вихідні сигнали. Ці сигнали потім сумуються та отримаємо вихідне значення F. Кожен сигнал на вході А викликає зміну стану згідно з циклічною трійковою послідовністю: «0» → «-1» → «+1» → «0». Така побудова схеми дозволяє точно керувати переходами між станами та забезпечує стабільну роботу пристрою.

3.5 Реалізація трійкового інвертора

Трійковий інвертор є ключовим логічним елементом у багатозначних обчислювальних системах, зокрема у трійковій логіці. Його основна функція полягає в перетворенні вхідного сигналу на протилежне логічне значення згідно з поставленою задачею. Логіка роботи інвертора передбачає, що при подачі значення «-1» або «0» на вхід, на виході формується логічне «+1», тоді як при подачі «+1» результатом є «-1».

Реалізація трійкового інвертора на основі БПЕБЛ демонструє гнучкість цієї технології в обробці умовних логічних функцій. Метод ґрунтується на аналізі вхідного сигналу через декілька фіксованих порогових рівнів. Така побудова забезпечує однозначне формування вихідного сигналу відповідно до заданої логіки. Завдяки використанню БПЕБЛ, інвертор функціонує без складних обчислень або часових затримок, що дозволяє ефективно застосовувати його у високошвидкісних цифрових схемах див. табл. 3.5.1.

Таблиця 3.5.1 – Таблиця істинності трійкового інвертору та його вихідні комбінації

Вхідні значення	+L1	–L1	–L1	+R1	Сума виходів
0	0	0	0	+	+
–	+	0	0	0	+
+	0	–	–	+	–

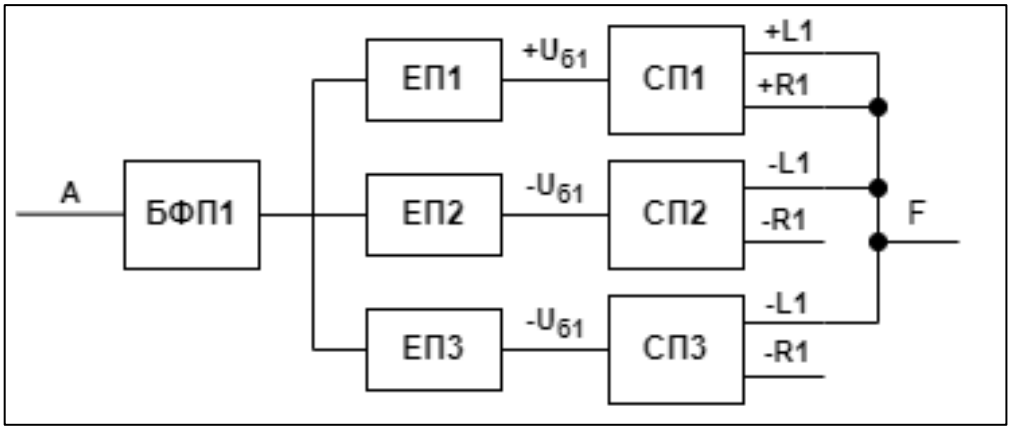

Рисунок 3.5.1 – Структурна схема трійкового інвертору

Схема трійкового інвертора реалізована на основі одного БФП (БФП1), ЕП (ЕП1, ЕП2, ЕП3) та СП (СП1, СП2, СП3) (рис. 3.5.1). На вхід А подається трійковий сигнал, що може набувати значень «-1», «0» або «+1». Блок БФП1 формує пороги, які далі подаються на ЕП1 і ЕП2. Елементи ЕП1–ЕП3 аналізують порогові значення, і передають проміжні сигнали на СП1–СП3. Кожен СП формує частковий вихідний сигнал на основі порівняння з відповідним порогом. Ці часткові сигнали підсумовуються для формування кінцевого вихідного значення F, які працюють з вихідними струмами +L1, +R1 і -L1. В результаті роботи такої схеми реалізується трійковий інвертор вхідного сигналу: якщо на вхід подано значення «-1» або «0», то на виході буде «+1»; якщо на вхід подано «+1», то результатом буде «-1». Така структура забезпечує стабільну та передбачувану роботу інвертора в системах трійкової логіки.

3.6 Реалізація трійкового D-тригера

Трійковий D-тригер є одним із ключових елементів трійкової логіки, призначеним для зберігання одного трійкового значення («-1», «0» або «+1») на основі двох входів: інформаційного входу D та керуючого входу C. Цей тригер зберігає значення, яке на вході D, у момент активного стану керуючого сигналу C. Таким чином, коли C перебуває в активному стані (наприклад, «+1»), тригер зчитує та зберігає значення з D. Якщо ж C знаходиться в неактивному стані, вихід тригера F зберігає попереднє значення без змін.

Реалізація трійкового D-тригера на основі БПЕБЛ дозволяє ефективно забезпечити затримку та збереження стану за допомогою порогового аналізу вхідних сигналів. Як і в інших трійкових схемах, логіка функціонування базується на перетворенні струмових сигналів у відповідності до порогових рівнів напруг, що дозволяє стабільно розрізняти стани «-1», «0» та «+1».

У трійковому D-тригері також ефективно використовується ідея зменшення кількості з'єднань: замість того, щоб мати окремі входи для

установки, скидання та утримання стану (як у двійковій логіці), трійковий тригер виконує всі ці функції через одну пару сигналів D і C. Така схема забезпечує більшу щільність логіки на одиницю площі та кращу масштабованість у складних трійкових схемах. Але сама реалізація схеми вийшла досить великою, через те що немає ще методів побудови елементів та наукової бази для трійкових елементів.

У момент активного імпульсу керуючого сигналу C значення з D надходить до виходу Q(0). Завдяки використанню трьох стійких логічних станів, трійковий D-тригер забезпечує більшу інформаційну ємність і дозволяє створювати складніші логічні структури з меншою кількістю елементів у порівнянні з аналогами у двійковій логіці. Така особливість робить його незамінним у проєктуванні трійкових регістрів, лічильників, пам’яті та інших цифрових автоматів. Тепер поетапно опишемо роботу спроектованого D-тригера, логіка роботи представлена у табл. 3.6.1.

Таблиця 3.6.1 – Таблиця істинності трійкового D-тригера

Вхідний сигнал (D)	Керуючий сигнал (C)	Попередній стан тригера (Q(-1))	Вихід (Q(0))
–	–	–	–
–	–	0	0
–	–	+	+
–	0	–	–
–	0	0	0
–	0	+	+
–	+	–	–
–	+	0	–
–	+	+	–
0	–	–	–
0	–	0	0
0	–	+	+
0	0	–	–
0	0	0	0
0	0	+	+
0	+	–	0

Продовження таблиці 3.6.1

Вхідний сигнал (D)	Керуючий сигнал (C)	Попередній стан тригера (Q(-1))	Вихід (Q(0))
0	+	0	0
0	+	+	0
+	–	–	–
+	–	0	0
+	–	+	+
+	0	–	–
+	0	0	0
+	0	+	+
+	+	–	+
+	+	0	+
+	+	+	+

У даній таблиці різними кольорами виділено різні рядки, це описується тим що саме для цих рядків була спроектована складна структура схеми для того щоб тригер працював правильно, тобто використано багато БФП, ЕП та СП для побудови правильно працюючого D-тригера.

З табл. 3.6.1 у синіх, зелених, фіолетових та помаранчевих рядках вихідні значення не збігалися, тому довелось компенсувати дані втрати, для цього були додані додаткові таблиці (тобто частини схеми) для правильної роботи тригера.

Для роботи тригера спочатку були побудовані дві таблиці 3.6.2-3.6.3 (кожна таблиця мається на увазі компонент схеми як у підрозділі 3.3, тобто два входи, з яких йдуть по чотири виходи, потім з яких формуються вихідні комбінації), але після перевірки значень на виході значення не збігалися з вихідною таблицею. У табл. 3.6.2 описується основна логіка роботи D-тригера, тобто при подачі на вхід C сигналу «+1», то значення D повино пройти далі у схемі, тобто на вихід.

Таблиця 3.6.2 – Таблиця комбінацій інформаційного та керуючого входів для формування виходу Q1(0)

		D				C				Вихід	Вихідна комбінація
		A1	B1	C1	D1	A0	B0	C0	D0		
D	C	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q1(0)	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	-	- L2(c1, d0, b0)
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	0	
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	+	+L2(a1, a0)

У табл. 3.6.3 були спроби зробити зв’язок з попереднім станом, тобто при подачі на С сигналу «+1», попередній стан не враховувався у вихідному сигналі, тобто йшов лише «0», але при перевірці цього не вистачило, і деякі рядки табл. 3.6.1 не виконувалися, тому потрібно додавати ще таблиці для компенсування.

Таблиця 3.6.3 – Таблиця комбінацій керуючого входу та попереднього стану для формування виходу Q2(0)

		C				Q(-1)				Вихід	Вихідна комбінація
		A1	B1	C1	D1	A2	B2	C2	D2		
C	Q(-1)	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q2(0)	
–	–	0	–	+	0	0	–	+	0	–	-L2(c0, c2)
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	+	+L2(b0, a2, d0)
0	–	0	–	0	+	0	–	+	0	–	-L2(d0, c2, a0)
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	+	+L2(b0, a2, c0)

Продовження таблиці 3.6.3

		C				Q(-1)				Вихід	Вихідна комбінація
		A1	B1	C1	D1	A2	B2	C2	D2		
C	Q(-1)	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q2(0)	
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	0	

Після створення цих двох таблиць потрібно їх поєднати щоб мати один єдиний вихід, для цього створена табл. 3.6.4 яка поєднує два виходи Q1(0) та Q2(0) з таблиць 3.6.2-3.6.3, для того щоб мати зв'язок між трьома вхідними змінними та мати один єдиний вихід. Після перевірок дана схема не вдалась, і треба компенсувати сині рядки матриці спочатку. Видно що для компенсування потрібно пов'язати входи D та Q(-1), бо при значеннях D «-1», «0» та «+1» та значеннях Q(-1) «+1» вихідні значення були «-1» або «0», в той час вони повинні бути «+1», тому була створена дана табл. 3.6.5 для компенсування даної проблеми.

Таблиця 3.6.4 – Таблиця комбінацій виходів Q1(0) та Q2(0) для формування виходу Q(0)

		Q1(0)				Q2(0)				Вихід	Вихідна комбінація
		A4	B4	C4	D4	A3	B3	C3	D3		
Q1(0)	Q2(0)	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q(0)	
–	–	0	–	+	0	0	–	+	0	–	-L2(c4, c3)
–	0	0	–	+	0	0	–	0	+	–	-L2(c4, d3, a3)
–	+	0	–	+	0	–	0	0	+	–	-L2(c4, d3, b3)
0	–	0	–	0	+	0	–	+	0	–	-L2(d4, c3, a3)
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	0	
+	–	–	0	0	+	0	–	+	0	+	+L2(a4, b3, d3)
+	0	–	0	0	+	0	–	0	+	+	+L2(a4, b3, c3)
+	+	–	0	0	+	–	0	0	+	+	+L2(a4, a3)

Таблиця 3.6.5 – Таблиця комбінацій виходів D та Q(-1) для формування виходу Q(0)

		D				Q(-1)				Вихід	Вихідна комбінація
		A1	B1	C1	D1	A0	B0	C0	D0		
D	Q(-1)	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q(0)	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	+	+L2(b1, a2, d1)
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	+	+L2(b1, a2, c1)
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	+	+L2(a1, a2)

Компенсування проблем на цьому не закінчується, залишаються ще проблеми з помаранчевою, фіолетовою та зеленою зоною. Для виправлення неправильного значення у помаранчевій зоні були створені додаткових три таблиці 3.6.6-3.6.8 які пов’язані між собою для задіяння певних комбінацій, а саме для тих випадків коли керуючий та попередній стани мають значення сигналу «+1», а сигнал інформаційного входу неважливий. Саме при таких значеннях буде виправлена ситуація на даному наборі сигналів. У табл. 3.6.6 ще раз перераховуються сигнали D та C, та отримаємо вихід DC+ який буде далі заподіяний у виходах схеми, для забезпечення правильної компенсації. У табл. 3.6.7 також ще раз перераховані сигнали C та Q(-1) для подальшої обробки у наступні табл. 3.6.8, та вихід даної таблиці буде подано на проміжній вихід CQ+. У табл. 3.6.8 використовують вихідні значення DC+ та CQ+ з попередніх двох таблиць для того щоб пов’язати саме три вхідних значення між собою, а саме три набори комбінацій («–++», «0++», «+++»), обробити їх та правильно компенсувати значення на виході.

Таблиця 3.6.6 – Таблиця комбінацій виходів D та C для формування виходу DC+

		D				C				Вихід	Вихідна комбінація
		A1	B1	C1	D1	A0	B0	C0	D0		
D	C	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	DC+	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	–	-L2(c1, d0, b0)
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	0	
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	+	+L2(a1, a0)

Таблиця 3.6.7 – Таблиця комбінацій виходів C та Q(-1) для формування виходу CQ+

		C				Q(-1)				Вихід	Вихідна комбінація
		A0	B0	C0	D0	A2	B2	C2	D2		
C	Q(-1)	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	CQ+	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	0	
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	0	
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	+	+L2(a0, a2)

Таблиця 3.6.8 – Таблиця комбінацій виходів DC+ та CQ+ для формування виходу Q(0)

		DC+				CQ+				Вихід	Вихідна комбінація
		A6	B6	C6	D6	A5	B5	C5	D5		
DC+	CQ+	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q(0)	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	–	-L2(c6, d5, b5)
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	0	
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	+	+L2(a6, a5)

Після даної компенсації залишаються ще два рядки табл. 3.6.1, які не збігаються з вихідними значеннями, а саме зелена та фіолетова зона таблиці. Для кожного з цих рядків були додані по одній таблиці 3.6.9-3.6.10, які компенсують втрати значень на цих наборах. У таблиці 3.6.9 були обраховані комбінації зі значеннями DC+ та CQ+ при таких значеннях щоб на наборі «0++» після компенсації значень буде правильний вихідний сигнал «0», все через зв’язану комбінацію даних входів «0++» за допомогою виходів DC+ та CQ+, та на виході отримали компенсуючий вихід Q(0). У табл. 3.6.10 знову обраховані значення DC+ та CQ+, але при інших комбінаціях вхідних сигналів, а саме при «+++», та на виході отримали компенсуючий вихід Q(0), який компенсує плюс який з’являється на тому місці, хоча він там не повинен бути для правильності роботи тригера. У результаті ми отримали повністю функціонуючий D-тригер, який працює правильно при усіх вхідних комбінаціях та при усіх попередніх станах.

Таблиця 3.6.9 – Таблиця комбінацій виходів DC+ та CQ+ для формування виходу Q(0)

		DC+				CQ+				Вихід	Вихідна комбінація
		A6	B6	C6	D6	A5	B5	C5	D5		
DC+	CQ+	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q(0)	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	0	
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	0	
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	–	-L2(d6, d5, b5, b6)

Таблиця 3.6.10 – Таблиця комбінацій виходів DC+ та CQ+ для формування виходу Q(0)

		DC+				CQ+				Вихід	Вихідна комбінація
		A6	B6	C6	D6	A5	B5	C5	D5		
DC+	CQ+	-L1	-R1	+L1	+R1	-L1	-R1	+L1	+R1	Q(0)	
–	–	0	–	+	0	0	–	+	0	0	
–	0	0	–	+	0	0	–	0	+	0	
–	+	0	–	+	0	–	0	0	+	0	
0	–	0	–	0	+	0	–	+	0	0	
0	0	0	–	0	+	0	–	0	+	0	
0	+	0	–	0	+	–	0	0	+	–	-L2(d6, d5, b5, a6)
+	–	–	0	0	+	0	–	+	0	0	
+	0	–	0	0	+	0	–	0	+	0	
+	+	–	0	0	+	–	0	0	+	0	

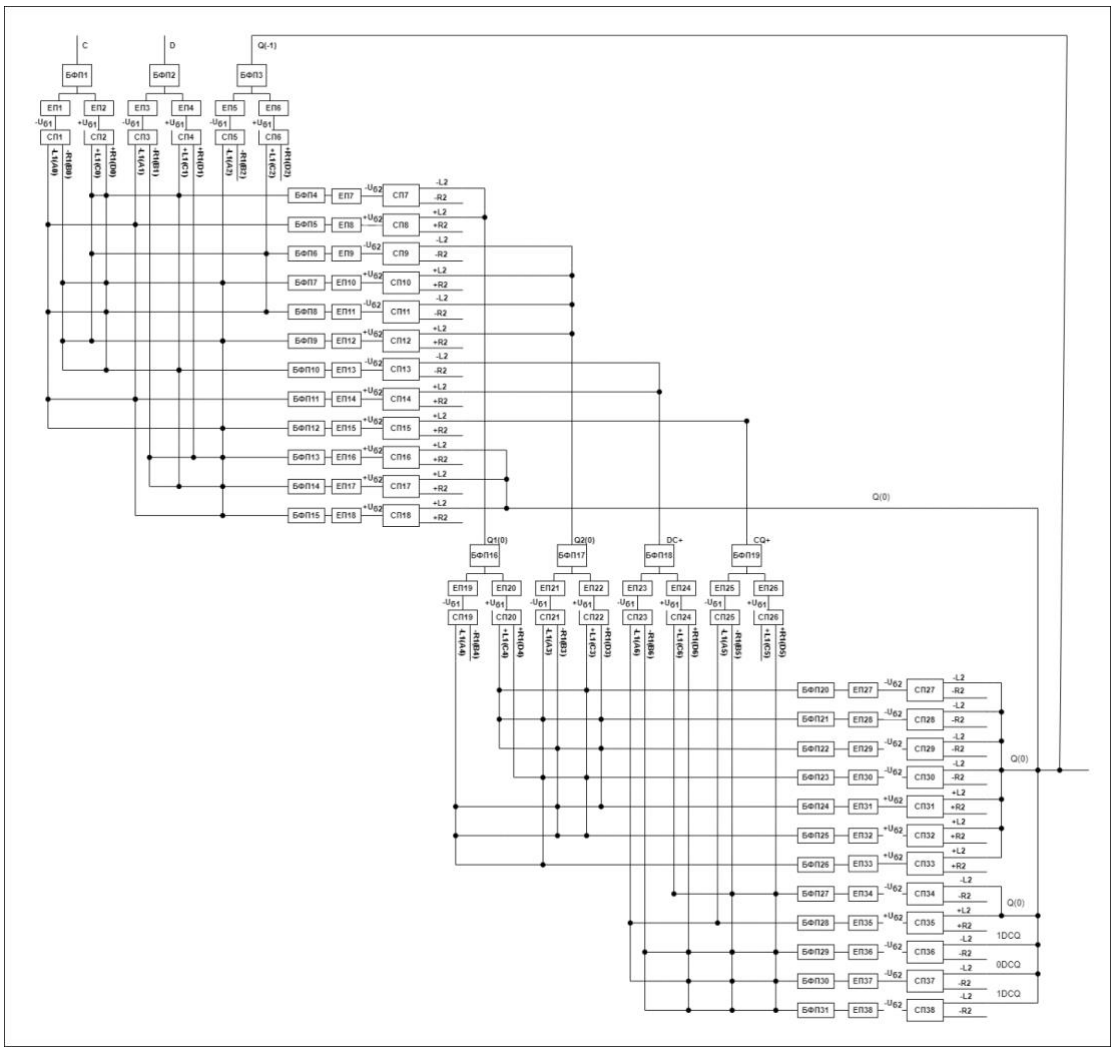

Рисунок 3.6.1 – Структурна схема трійкового D-тригера

Схема трійкового D-тригера складається з двох частин (рис. 3.6.1). Перша частина оброблює вхідні сигнали D і C та попередній стан Q(-1), які містять три блоки БФП (БФП1–БФП3), кожен з яких поєднано з двома ЕП (ЕП1–ЕП6), які формують проміжні сигнали які йдуть на СП (СП1–СП6), які визначають порогові рівні для подальшої обробки. Після цього виходи СП1–СП6 поєднуються між собою для утворення вихідних комбінацій першої частини проміжних значень, для цих комбінацій задіяні такі виходи СП: A0, B0, C0, D0, A1, B1, C1, D1, A2 та C2. Дані виходи прямують до наступних блоків БФП (БФП4–БФП15), кожен з яких поєднано лише з одним ЕП (ЕП7–ЕП18), які в свою чергу також поєднані лише з одними СП (СП7–СП18), які формують вихідні проміжні сигнали для подальшої роботи схеми для

утворення проміжних змінних як Q1(0), Q2(0), DC+, CQ+ та одне вихідне значення Q(0). У другій частині утворюються проміжні та вихідні зміні. У даній частині використовуються чотири блоки БФП (БФП16–БФП19), кожен з яких поєднано з двома ЕП (ЕП19–ЕП26), які формують проміжні сигнали які йдуть на наступні СП (СП19–СП26), які визначають порогові рівні для подальшої обробки схеми. Після цього СП19–СП26 поєднуються між собою для утворення вихідних комбінацій другої частини проміжних значень, для цих комбінацій задіяні такі виходи СП: A4, C4, D4, A3, B3, C3, D3, A6, B6, C6, D6, A5, B5 та D5. Дані виходи прямують до наступних блоків БФП (БФП20–БФП31), кожен з яких поєднано лише з одним ЕП (ЕП27–ЕП38), які в свою чергу також поєднані лише з одними СП (СП27–СП38), які формують вихідні сигнали «+–L2», які потім об’єднуються між собою та формують вихідні значення схеми.

3.7 Реалізація трійкового Т-тригера

При проєктуванні синхронних послідовних логічних схем критично важливим є забезпечення стабільності та передбачуваності роботи елементів пам'яті, таких як тригери. Однією з ключових проблем, що виникає в простих тригерах, які реагують на рівень сигналу, є ефект «гонок» (race around condition). Цей ефект проявляється тоді, коли вихідний стан тригера змінюється та повертається на вхід ще до завершення тактового імпульсу, що призводить до багаторазових, неконтрольованих перемикань протягом одного циклу тактового сигналу та, як наслідок, до нестабільної поведінки.

Для ефективного усунення проблеми «гонок» застосовується архітектура двоступеневого D-тригера, також відомого як Master-Slave D Flip-Flop. Теоретична основа цього підходу полягає в розділенні операції фіксації вхідних даних та оновлення вихідного стану на два незалежні, послідовні етапи, кожен з яких активується різними фазами тактового сигналу.

Ця архітектура складається з двох послідовно з'єднаних D-засувок:

1) майстер-засувка є першим ступенем; вона функціонує в режимі «прозорості, тобто її вихід безпосередньо відображає її вхід, під час однієї фази тактового сигналу (наприклад, коли тактовий сигнал перебуває у високому логічному стані); у цей період майстер-засувка здійснює захоплення вхідних даних; важливою особливістю є те, що в цей самий час слейв-засувка перебуває в режимі «утримання», що означає стабільність її вихідного стану незалежно від змін на вході майстер-засувки;

2) коли тактовий сигнал переходить у протилежну фазу (наприклад, з високого на низький), майстер-засувка переходить у режим «утримання», надійно фіксуючи значення, яке присутнє на її вході в момент зміни тактового сигналу (фронту); саме в цей момент слейв-засувка стає «прозорою» (оскільки її тактовий вхід, як правило, інвертований відносно майстер-засувки); вона негайно передає на свій вихід (який є кінцевим виходом всього двоступеневого D-тригера) значення, що зафіксоване майстер-засувкою.

Таким чином, оновлення вихідного стану D-тригера відбувається лише в момент зміни тактового сигналу (наприклад, спадаючого фронту, якщо майстер-засувка активується високим рівнем, а слейв – низьким, і навпаки для зростаючого фронту). Цей механізм гарантує, що вихід змінюється лише один раз за тактовий цикл і виключно на заданому фронті, ефективно запобігаючи виникненню ефекту «гонок», оскільки майстер-засувка вже зафіксувала вхідні дані до того, як слейв-засувка починає їх поширення [2].

Таблиця 3.7.1 – Таблиця істинності трійкового Т-тригера

Вхідний сигнал (С)	Вихідний сигнал (Q(0))
0	–
–	+
+	0

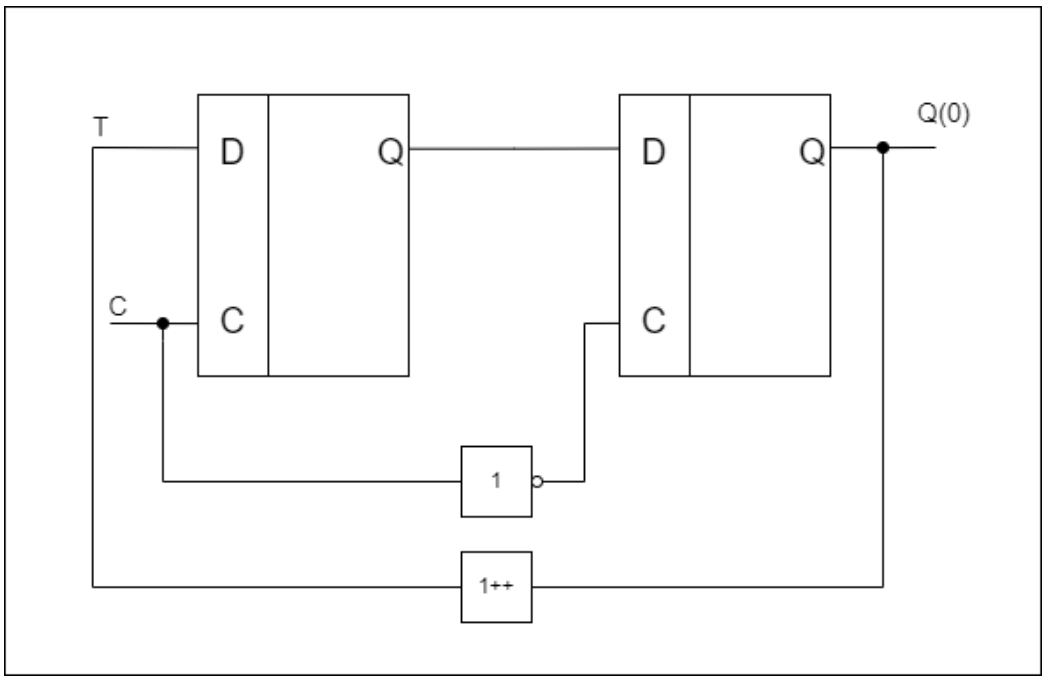

Рисунок 3.7.1 – Структурна схема трійкового Т-тригера

Для реалізації функціоналу трійкового Т-тригера на основі трійкового двоступеневого D-тригера застосовується елегантна схема зворотного зв'язку зображено на рисунку 3.7.1. Вихід першого D-тригера поєднано з інформаційним входом другого D-тригера, на тактовий вхід першого D-тригера подається тактовий сигнал напряму, а до другого D-тригера через інвертор описаний у підрозділі 3.5. Вихід другого D-тригера йде на вихід та одночасно йде напряму на інформаційний вхід першого D-тригера через інкремент описаний у підрозділі 3.4. Таким чином отримуємо трійковий Т-тригер який працює від спалу сигналу, щоб уникнути ефекту «гонок». Логіка роботи така, що значення змінюються від «0» до «-1», від «-1» до «+1» та від «+1» до «0» циклічно. У табл. 3.7.1 зображені усі можливі значення та комбінації Т-тригера, що дозволяє в майбутньому будувати лічильники на його основі. У даній схемі використовуються елементи як D-тригер, інкремент та інвертор, тому на рисунку 3.7.1 наведені лише їх структурні елементи.

3.8 Реалізація трійкового дворозрядного лічильника

Лічильники в цифровій електроніці поділяються на синхронні та асинхронні. У синхронних лічильниках усі елементи пам'яті (тригери) отримують єдиний тактовий сигнал, що забезпечує їхню одночасну зміну стану та спрощує синхронізацію у складних схемах. Натомість, асинхронні лічильники характеризуються послідовним поширенням тактового імпульсу: лише перший тригер керується безпосередньо зовнішнім тактовим генератором, тоді як тактові входи наступних тригерів підключаються до виходів попередніх. Цей «хвильовий» принцип дозволяє спростити схему керування, хоча й може призводити до затримок поширення сигналу в багаторозрядних лічильниках. У межах даної роботи зосереджено увагу саме на асинхронній архітектурі.

Основою запропонованого лічильника є трійковий Т-тригер, розроблений для роботи з 3-ма дискретними логічними станами (рис. 3.8.1). Для цілей даної реалізації ці стани позначено як «0», «−» та «+», які відповідають числовим значенням 0, 1 та 2 відповідно. Цей Т-тригер має один вхід для керування перемиканням та один вихід, що відображає поточний трійковий стан. При кожному активному фронті тактового сигналу, за умови активації входу Т, тригер циклічно змінює свій вихідний стан у послідовності: «0» → «−» → «+» → «0». Дворозрядний трійковий асинхронний лічильник формується шляхом послідовного каскадування двох ідентичних трійкових Т-тригерів. Структурна схема передбачає наступне підключення:

1) перший Т-тригер (Т0) його тактовий вхід (С) підключається безпосередньо до зовнішнього тактового генератора, який є основним тактовим входом усього лічильника; вхід Т цього тригера постійно перемикається з активного стану в неактивний, що забезпечує його перемикання при кожному вхідному тактовому імпульс;

2) другий Т-тригер (Т1) його тактовий вхід (С) з'єднується з виходом Q0 першого тригера, це означає, що другий тригер буде перемикати свій стан

лише тоді, коли вихід Q0 генерує керуючий фронт, що відповідає завершенню повного циклу молодшого розряду; вхід T цього тригера також залишається постійно активним, вихід цього тригера, позначений як Q1, є старшим розрядом лічильника.

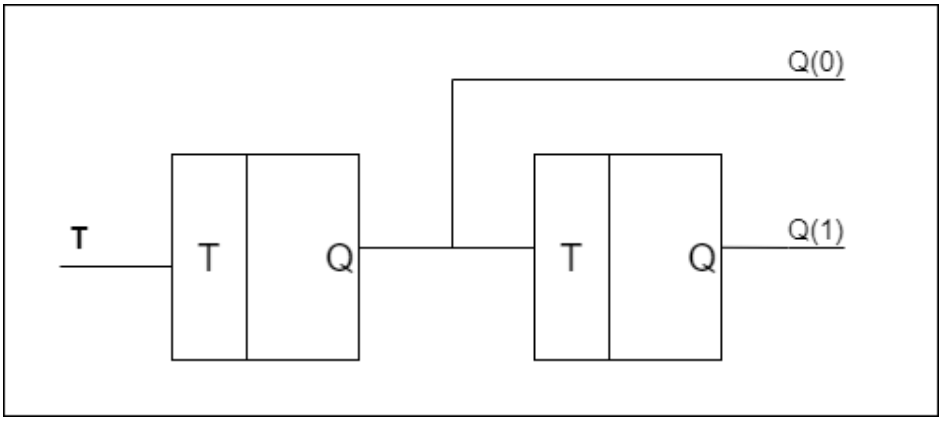

Рисунок 3.8.1 – Структурна схема двохрозрядного лічильника на основі трійкових Т-тригерів

Внаслідок такого каскадування, кожен Т-тригер функціонує як дільник частоти на 3. Лічильник послідовно проходить через 3^2=9 унікальних станів, що відповідають десятковим значенням від 0 до 8. Після досягнення стану, що відповідає десятковій 8, лічильник циклічно повертається до початкового стану 0. Вихідні сигнали лічильника у трійковому вигляді (0, −, +) можуть бути перетворені в еквівалентне десяткове число. Відповідність символів числовим значенням встановлюється таким чином: «0» → 0, «−» → 1, «+» → 2. Десяткове значення лічильника (D) розраховується за формулою позиційної системи числення:

$$D = (Q_1 \times 3^1) + (Q_0 \times 3^0) \quad (3.8.1)$$

де Q_0 – молодший розряд лічильника;

Q_1 – .старший розряд лічильника.

Для забезпечення функціональності даного лічильника, зокрема циклічного перемикання станів трійкових Т-тригерів та їхньої взаємодії, була розроблена табл. 3.8.1. Ця таблиця детально описує відповідність вхідних сигналів вихідним станам для кожного тригера та відображає послідовність переходу лічильника між його дев'ятьма унікальними станами.

Таблиця 3.8.1 – Таблиця істинності трійкового лічильника

Номер такту	Вихід першого розряду	Вихід другого розряду	Десяткове значення
0	0	0	$0*3^0+0*3^1=0$
1	-	0	$1*3^0+0*3^1=1$
2	+	0	$2*3^0+0*3^1=2$
3	0	-	$0*3^0+1*3^1=3$
4	-	-	$1*3^0+1*3^1=4$
5	+	-	$2*3^0+1*3^1=5$
6	0	+	$0*3^0+2*3^1=6$
7	-	+	$1*3^0+2*3^1=7$
8	+	+	$2*3^0+2*3^1=8$
9	0	0	$0*3^0+0*3^1=0$

Цей дворозрядний трійковий лічильник функціонує циклічно, приймаючи лише один вхідний тактовий сигнал. Він відраховує значення від 0 до 8, після чого скидається назад до 0. Лічильник має один вхід і два виходи, що відповідають за його розряди, сумарне значення яких інтерпретується як десяткове число.

Ефективність даного трійкового лічильника, порівняно з двійковим, полягає у його здатності кодувати більше інформації на один розряд. У той час як дворозрядний двійковий лічильник може представляти лише 4 унікальні стани, наш дворозрядний трійковий лічильник здатен відраховувати до 9 станів, охоплюючи діапазон від 0 до 8. Це означає, що для досягнення тієї ж лічильної ємності трійковій системі потрібно менше логічних елементів. Така зменшена апаратна складність призводить до меншого розміру схеми, потенційно нижчого споживання енергії та спрощення виробництва, підкреслюючи значну перевагу трійкової логіки.

ВИСНОВКИ

У дипломній роботі розроблено та реалізовано трійкові обчислювальні пристрої як мультиплексор «3 до 1», D-тригер, T-тригер, інкремент, інвертор та дворозрядний трійковий лічильник на основі багатопорогових елементів багатозначної логіки. Проведено порівняльний аналіз двійкової та трійкової логіки, визначено їхні теоретичні основи, переваги, недоліки та перспективи застосування в цифрових системах.

Спроєктовано структурні схеми трійкових логічних елементів, здійснено їх моделювання, перевірено коректність роботи згідно з таблицями істинності. Показано, що реалізація трійкових пристроїв дозволяє в деяких випадках зменшити апаратну складність та підвищити інформаційну щільність за рахунок використання трьох логічних станів.

У роботі також обґрунтовано доцільність використання трійкової логіки в перспективних напрямках цифрової техніки зокрема в системах штучного інтелекту, нейромережах та високопродуктивних процесорах. При цьому ідентифіковано чинники, що стримують масове впровадження, зокрема технологічні обмеження та потребу в розвитку елементної бази.

Таким чином, у роботі досягнуто поставлену мету, виконано повний цикл дослідження: від теоретичного аналізу до побудови й апробації трійкових схем, що демонструє можливість ефективного застосування багатозначної логіки у цифрових пристроях нового покоління.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Morris Mano, Michael D. Ciletti (2017). Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog. [Електронний ресурс] – Режим доступу: https://mrce.in/ebooks/Digital%20Design%20With%20an%20Introduction%20to%20the%20Verilog%20HDL,%20VHDL%20and%20Systemverilog%206th%20Ed.pdf

2. Thomas L. Floyd (2015). Digital Fundamentals, Pearson Education Limited. [Електронний ресурс] – Режим доступу: https://bpcbirgunj.edu.np/wp-content/uploads/2019/10/DIGITAL_ELECTRONICS-by-Flyod.pdf

3. Charles H. Roth, Jr., Larry L. Kinney, Eugene B. John (2021). Fundamentals of Logic Design, Enhanced Seventh Editions. [Електронний ресурс] – Режим доступу: https://mrce.in/ebooks/Logic%20Design%20Fundamentals%207th%20Ed.pdf

4. John L. Hennessy, David A. Patterson (2011). Computer Architecture: A Quantitative Approach. Morgan Kaufmann. [Електронний ресурс] – Режим доступу:https://acs.pub.ro/~cpop/SMPA/Computer%20Architecture%20A%20Quantitative%20Approach%20(5th%20edition).pdf

5. Date C. J., An Introduction to Database Systems / Addison-Wesley, 2004. – 1024 p.

6. D. Michael Miller; Mitchell A. Thornton (2008). Multiple valued logic: concepts and representations. Synthesis lectures on digital circuits and systems 12. Morgan & Claypool Publishers. ISBN 9781598291902.

7. Мартинович Л. Я. Проектування та синтез трійкових логічних елементів. Комп’ютерні системи та інформаційні технології. Міжнародний науковий журнал Хмельницького національного університету. Вип. 4 (2022): 52-60. URL: https://doi.org/10.31891/csit-2022-4-8.

8. Лебедєва, В. В., Ю. О. Гунченко. Інформатика, інформаційні системи та технології. Огляд тризначної логіки. Інформатика, інформаційні системи та

технології: тези доповідей тринадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 8 квітня 2016р. - Одеса, 2016. – с. 61-63. [Електронний ресурс] – Режим доступу: https://atl.pdpu.edu.ua/doc/content/2016/inf/Zbirka_tez_IIS_T-2016.pdf#page=61

9. Левчук В., Гунченко Ю. Метод побудови трійкових одномісних функцій // Інформатика, інформаційні системи та технології: матеріали 15-Ї всеукраїнська конференція студентів та молодих науковців(27 квітня 2018 р., м. Одеса), С. 208. [Електронний ресурс] – Режим доступу: https://atl.pdpu.edu.ua/doc/content/2018/inf/Zbirka_tez_IIS_T-2018.pdf#page=16

10. Навчальний посібник. – Одеса: ОНАЗ ім. О. С. Попова. – 2004, Ч. 2. – 172с.: іл.

11. Що таке мультиплексор і типи мультиплексування. [Електронний ресурс] – Режим доступу: https://www.watelectronics.com/what-is-multiplexer-and-types/

12. Лічильники та його основні положення. [Електронний ресурс] – Режим доступу: https://studfile.net/preview/5083036/page:11/

13. Схемотехніка цифрових елементів, двійкові лічильники, тригери та регістри. [Електронний ресурс] – Режим доступу: http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/lekcija-6_tryhery-lichylnyky-rehistry.pdf

14. Цифрова схемотехніка: навчальний посібник для самостійної роботи студентів (2021). [Електронний ресурс] – Режим доступу: https://ela.kpi.ua/server/api/core/bitstreams/6ff22a52-988f-4b6a-a471-3ea7cb1daece/content