

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 004.421.2:519.7

[https://doi.org/10.32515/2664-262X.2025.11\(42\).2.45-51](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 швидкодіючих 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 напрямку 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 посиленням інтересом наукової спільноти до альтернативних логічних основ, зокрема до трійкової логіки, останніми роками було опубліковано низку праць, присвячених розробці як теоретичних,

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 проектування трійкових логічних воріт та схем, таких як інвертори, NAND, NOR, трійковий декодер, мультиплексор і напівсуматор.

Особливо важливими є дослідження, що ґрунтуються 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 елемента 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 елемента. 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 елемента; 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

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 комбінації 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.

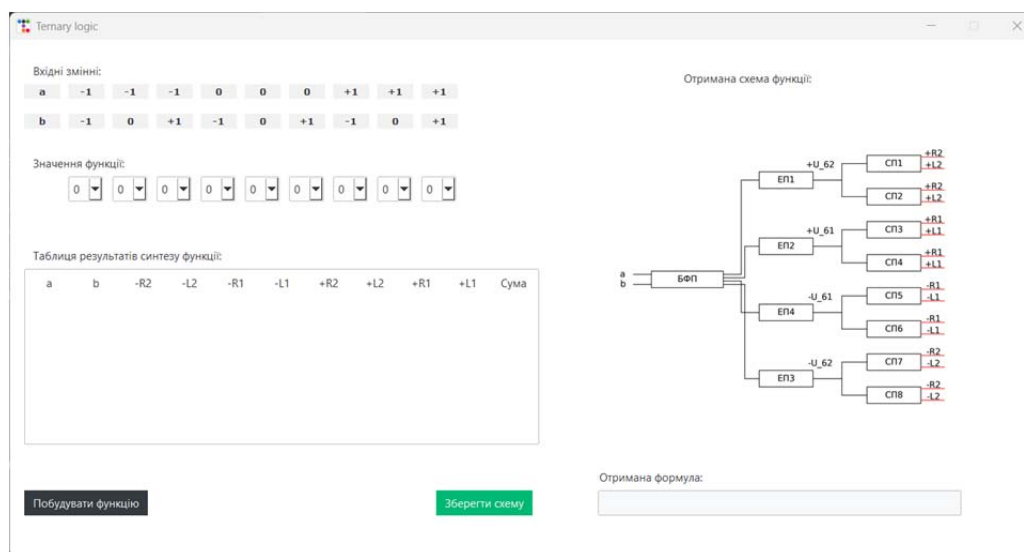


Рисунок 1 – Головне вікно програми

Джерело: розроблено авторами

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 проектування 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 «I». 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 функцій 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.

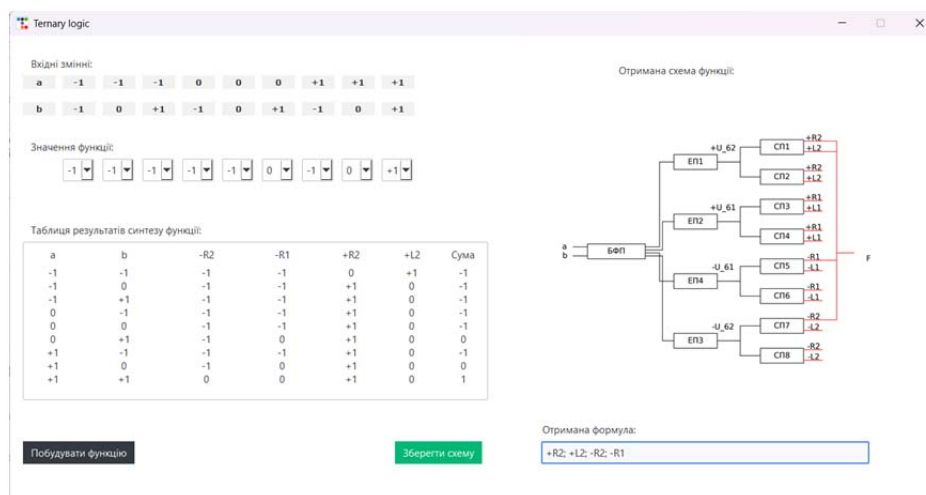


Рисунок 4 – Побудова сильної кон'юнкції

Джерело: розроблено авторами

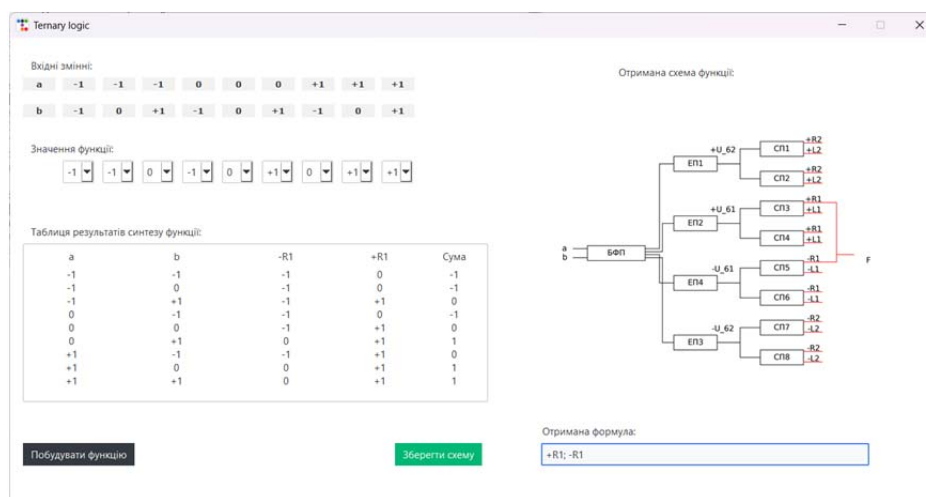


Рисунок 5 – Побудова функції «Приймати все»

Джерело: розроблено авторами

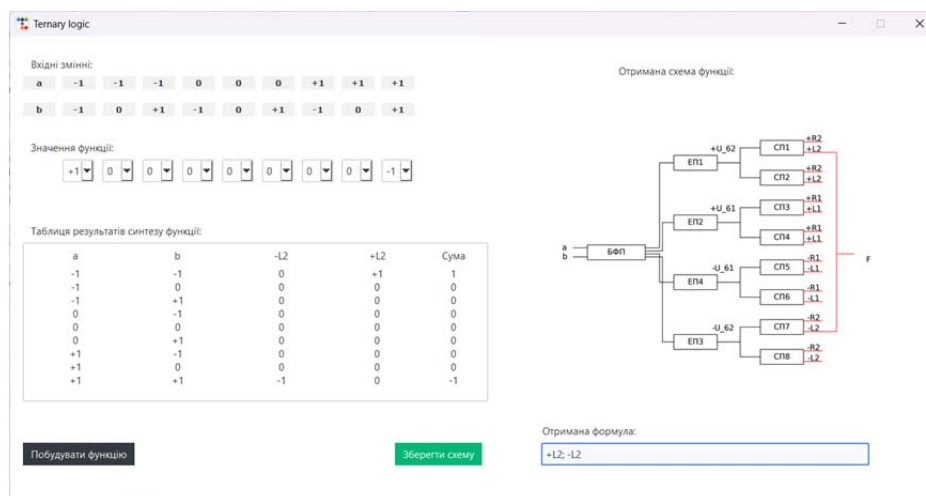


Рисунок 6 – Побудова консенсусу

Джерело: розроблено авторами

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., Кузнichenko, 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 Kyivskoho 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.

4. Gunchenko, Y. O., Lenkov, S. V., Malakhov, Y. V., Shvoro, S. A., Ustymchuk, V. V., Lukin, V. Y., Mezhuiev, 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., Shvoro, S. A., Mezhuiev, V. I., Levchenko, A. O., Kuznichenko, S. D., Lenkov, Y. S., Nikolaenko, O. Y., Shvoro, 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., Mezhuiev, V. I., Masliy, N. D., Shugailo, Y. B., Berkov, Y. M., & Fastyskovskyi, P. P. (2021). Ternary RS flip-flop (Ukrainian Patent No. 149386). Odesa I. I. Mechnikov National University.
7. Gunchenko, Y. O., Lenkov, S. V., Shvoro, S. A., Mezhuiev, 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 modeliuвання: 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., & Berkov, 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. Mechnikov 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