УДК 519.718
Дяк М.С., бакалавр, Круліковський Б.Б., канд. техн. наук, доцент,завідувач кафедри
НУ водного господарства та природокористування

МІНІМІЗАЦІЯ 5-РОЗРЯДНИХ БУЛЕВИХ ФУНКЦІЙ МЕТОДОМ РІЗНИКА-СОЛОМКА

Дяк М.С., Круліковський Б.Б. Мінімізація 5-розрядних булевих функцій методом Різника-Соломка. Розглянуто поширення принципу мінімізації за допомогою алгебричних перетворень на метод мінімізації з використанням комбінаторної блок – схеми з повторенням Математичний апарат блок – схеми з повторенням дає більше інформації стосовно ортогональності, суміжності, однозначності блоків комбінаторної системи, якою є власне таблиця істинності заданої функції, тому застосування такої системи мінімізації функцій є більш ефективним.
Ключові слова: булева функція, метод мінімізації, мінімізація логічної функції, блок-схема з повторенням, мінтерм.

Дяк М.С., Круликовский Б.Б. Минимизация 5-разрядных булевых функций методом Ризника-Соломка. Рассмотрено распространение принципа минимизации с помощью алгебраических преобразований на метод минимизации с использованием комбинаторной блок – схемы с повторением. Математический аппарат блок – схемы с повторением даёт больше информации относительно ортогональности, смежности, однозначности блоков комбинаторной системы, которой является собственно таблица истинности заданной функции, поэтому применение такой системы минимизации функций есть более эффективным.
Ключевые слова: булева функция, метод минимизации, минимизация логической функции, блок-схема с повторением, минтерм.

Dyak M.S., Krulikovskyi B. Krulikovskyi B. Krulikovskyi B. Krulikovskyi V.V.Minimization of 5-bit Boolean functions by method Riznyk-Solomko. The distribution of the principle of minimization by means of algebraic transformations to the minimization method using combinatorial block diagrams with repetition. The mathematical apparatus of the block diagram with repetition gives more information regarding the orthogonally, adjacency, uniqueness of the combinatorial system blocks, which is the actual table of the truth of the given function, therefore the use of such the function minimization system is more effective.
Keywords: Boolean function, minimization method, logic function minimization, block-design with repetition, mìnterm

Вступ.

Мінімізація булевих функцій популярна у різних областях цифрових технологій, таких як дизайн PLA, вбудований самотест (BIST), проектування систем управління тощо. Проблема мінімізації ДНФ є однією з багатоекстремальних логіко-комбінаторних задач і зводиться до оптимального зменшення кількості логічних елементів вентильної схеми без втрати її функціональності. Слід зазначити, що у загальній постановці дана задача до тепер не вирішена, однак добре досліджена у класі диз'юнктивно-кон'юнктивних нормальних форм.

Відомі такі методи мінімізації булевих функцій у класі ДДНФ [1–5]:

- метод Блейка - Порецького;
- метод Нельсона;
- метод Карта Карно;
- матричний метод;
- метод Квайна;
- метод Квайна – Мак-Класкі;
- метод Діаграма Вейча;
- метод алгебричних перетворень;
- метод Петрика;
- метод Рота;
- метод мінімізації функцій у базисах ТА-НІ і АБО-НІ (базиси Шеффера та Пірса);
- метод невизначених коефіцієнтів;
- метод гіперкубів;
- метод функціональної декомпозиції;
- евристичний алгоритм мінімізації Espresso.

Недоліки відомих методів мінімізації булевих функцій пов'язані зі стрімким зростанням обсягу обчислень, наслідком чого є збільшення розрядності обчислювальних операцій, і, отже, збільшенням числа змінних логічної функції. Наприклад, карта Карно зазвичай важко піддається розпізнаванню при зростанні кількості змінних більше чотирьох-п’яти, тому цей метод недоцільно використовувати з більше ніж шістьма змінними. Незважаючи на більшу досконалість методу Квайна – Мак-Класкі порівняно з картами Карно, він також має обмежене практичне застосування з-за експоненціального зростання часу обчислення зі збільшенням кількості змінних.

Від результату мінімізації булевої функції залежить швидкодія обчислювального пристрою, його надійність та енергозбереження. Особливості мінімізації методом Різника - Соломка [6.7] полягають у більшій інформативності процесу вирішення задачі порівняно з алгебричним способом мінімізації функції за рахунок табличної організації та впровадження апарату образного перетворення. Об’єктом вирішення задачі мінімізації булевої функції методом Різника – Соломка є блок – схема з повторенням, властивості якої, у свою чергу, дозволяють доповнити правила алгебри логіки новими правилами спрощення логічної функції. Алгоритм мінімізації булевої функції є однією з центральних та практично важливих проблем, яка постає під час проектування обчислювальних пристроїв. У зв’язку з цим вивчення нових правил алгебри логіки, встановлення їх властивостей є актуальним для спрощення алгоритму мінімізації булевої функції без втрати її функціональності при збільшенні кількості змінних.

Дослідження існуючих рішень проблеми.

Умови логічного зведення до мінімуму булевої функції, поданої у ДНФ, розглядаються у [8]. Якщо функція задовольняє таким умовам, то для її спрощення застосовують класичний алгоритм мінімізації Квайна–Мак-Класкі, що допускає автоматизацію. Зазначається, що число змінних функції для коду програми обмежується пам’яттю комп’ютера. Автор публікації [9] описують метод оптимізації, коли процес включає в себе не тільки пошук еквівалентного логічного виразу, але й залучає визначення конкретних умов, за яких логічні вирази можна ще більше скоротити. Ці типи елементів у логічному дизайні розглядаються як "ступінь свободи". У таких випадках користувач може оптимізувати заданий дизайн на підставі ступеня свободи. Тому пошук альтернативних рішень є бажаним, оскільки він може забезпечити оптимальний булевий вираз у підсумку. У публікації [10] розглянуто узагальнені правила спрощення кон’юнктермів у поліноміальному теоретико-множинному форматі, які ґрунтуються на запропонованих теоремах для різних початкових умов перетворення пари кон’юнктермів, геммінгова відстань між якими може бути довільна. Зазначені правила можуть бути корисними для мінімізації у поліноміальному теоретико-множинному форматі довільних логічних функцій від *n* змінних. Ефективність запропонованих правил демонструється прикладами мінімізації функції, запозичених з робіт відомих авторів з метою порівняння результатів. З огляду на порівняльні приклади запропоновані правила дають підставу для підтвердження доцільності застосування їх у процедурах мінімізації будь-якої логічної функції від *n* змінних у поліноміальній формі. У роботі [11] представлена мінімізація булевої функції з використанням таблиці істинності, в якій послідовно зменшуються одна змінна поки всі змінні не вичерпаються. У стандартному методі таблиця істинності (ТІ) готується за заданою логічною функцією. Тоді функція виражається як сума мінімальних умов, що відповідають наборам змінних, на яких функція отримує значення одиниці. Нарешті, ця функція зменшується за допомогою булевих ідентичностей. Таким чином, всі спрощення концентруються в одному місці після ТІ. Ця процедура не завжди приводить до мінімальної реалізації. У роботі [11] розглянуто спрощення, що наприкінці кожного етапу здійснює скорочення ТІ. Показано, що метод є системним і безумовно веде до мінімальної функції. Він простіший в експлуатації, ніж на основі тільки булевих топонімів, карт Карно, Quine-McClusky та може обробляти будь-яку кількість змінних. Це пояснюється декількома прикладами. Алгоритм і програма для мінімізації комбінаційних логічних функцій до 20 змінних представлені у [12], де число змінних обмежується пам'яттю комп'ютерної системи. Алгоритм заснований на послідовній кластеризації термів, починаючи з групування термів з однією зміною. Алгоритм кластеризації закінчується тоді, коли змінні не можуть більше бути згруповані. Цей алгоритм аналогічний алгоритму Квайна – Мак-Класкі, але він є простішим, оскільки усуває ряд дій алгоритму Квайна – Мак-Класкі. У роботі [13] демонструється метод логіко-мінімізаційного стиснення зображень, який залежить від логічної функції. Процес мінімізації розглядає сусідні пікселі зображення як роз’єднані мінтерми, що представляють логічну функцію та стискає 24-розрядні кольорові зображення за допомогою процедури мінімізації функції. Коефіцієнт стиснення такого методу у середньому на 25

% більший порівняно з існуючими методами стиснення зображень. Робота [14] демонструє використання генетичного алгоритму для вибору побічних об'єктів процедури мінімізації логічної функції за допомогою карти Карно. У [15] запропоновано новий евристичний алгоритм для максимальної мінімізації булевих функцій. Для реалізації запропонованого алгоритму використовуються графічні дані і представлені умови для досягнення максимального рівня мінімізації булевої функції.

На відміну від публікацій [8–15], у даній роботі об'єктом вирішення задачі мінімізації булевої функції є комбінаторна блок – схема з повторенням, а об'єктом спрощення процесу мінімізації - процедура алгебри логіки – суперсклеювання змінних, яка здійснюється за наявності у структурі таблиці істинності повної або неповної бінарних комбінаторних систем з повторенням. Процедура скорочення повної досконалої диз'юнктивної нормальної форми (ДДНФ) логічної функції дає одиницю. А оскільки повна ДДНФ однозначно визначає повну бінарну комбінаторну систему з повторенням і навпаки, це дає підставу видаляти всі блоки повної бінарної комбінаторної системи з повторенням з таблиці істинності, структура якої дозволяє проводити правила супер-склеювання змінних. Математичний апарат блок – схеми з повторенням дає можливість отримати більше інформації стосовно ортогональності, суміжності, однозначності блоків таблиці істинності (комбінаторної системи). Рівносильні перетворення графічними образами, що за своїми властивостями мають більшу інформаційну ємність, спроможні з ефектом замінити вербальні процедури алгебричних перетворень.

Мінімізація булевих функцій метод Різника – Соломка.

Поняття булевих функцій і ДНФ тісно пов'язані з багатьма поняттями комбінаторного аналізу, зокрема з поняттям покриття. Нехай C = (X1, ..., Xn) – деяке сімейство підмножин множини X, і нехай $Y \subseteq X$. Тоді Y є покриттям для C, якщо для будь-якого Xi з C виконується умова $X_i \cap Y \neq \varnothing$. Покриття Y називається приведеним для C, якщо будь-яка його власна підмножина не є покриттям для C. Множина всіх приведених покриттів для C позначається через P(C).

При використанні блок – схеми з повторенням (3) процес мінімізації у частині склеювання змінних зводиться до пошуку блоків з однаковими змінними у відповідних розрядах, за виключенням однієї змінної. Враховуючи табличну організацію методу Різника - Соломка, це дає змогу підвищити ефективність пошуку мінімальної функції.

Мінімізація логічної функції методом Різника - Соломка здійснюється наступним чином. На першому кроці виявляють блоки (конституанти) зі змінними, для яких можлива операція супер-склеювання змінних. У випадку відсутності операції супер-склеювання, проводиться операція простого склеювання змінних. Наступним кроком здійснюють пошук наборів пар блоків (імплікант) з можливістю їх мінімізації заміщенням (склеюванням, поглинанням) змінних у цих парах. Отримані набори блоків знову мінімізують подібним способом, і т. д. – до отримання тупикової ДНФ (ТДНФ). У загальному випадку на прикінцевих кроках мінімізації можливим є застосування методу Блейка-Порецького, а також збільшення числа змінних зі значенням логічної одиниці. Серед множини ТДНФ містяться і мінімальні функції (МДНФ). Після мінімізації логічної функції проводиться верифікація мінімізованої функції, застосовуючи задану таблицю істинності.

Початок процедури мінімізації за методом Різника – Соломка зводиться до пошуку локального екстремуму мінімальної функції. Однак апарат алгебричних перетворень методу дозволяє здійснювати переходи з одного локального екстремуму мінімальної функції до іншого, і, таким чином, знаходити глобальний екстремум мінімальної функції.

У загальному випадку, під час мінімізації функції методом Різника - Соломка, використовуються такі правила алгебри логіки:

- склеювання змінних - $ab + a\bar{b} = a$,
- узагальнене склеювання змінних - $xy + \bar{x}z = xy + \bar{x}z + yz$,
- заміщення змінної - $a + \bar{a}b = a + b$,
- поглинання змінної - $ab + a = a(b+1) = a$,
- ідемподентність змінних - $a + a = a$, $aa = a$.

- доповнення змінної - $a + \overline{a} = 1$, $a\overline{a} = 0$,
- повторення константи - $a + 0 = a$, $a \cdot 1 = a$,
- та інші.

Алгебричні перетворення доцільно замінити рівносильними перетвореннями за допомогою підматриць (графічних образів). Процедуру склеювання за допомогою підматриць можна продемонструвати так:

$$\overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 = \overline{x}_1(\overline{x}_2 + x_2) = \overline{x}_1,$$

$$\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array} \rightarrow \begin{array}{c} 0 \\ 0 \end{array} \rightarrow 0$$

$$x_1 x_2 + \overline{x}_1 x_2 + x_2(x_1 + \overline{x}_1) = x_2,$$

$$\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array} \rightarrow \begin{array}{c} 1 \\ 1 \end{array} \rightarrow 1$$

Застосовуючи підматриці (графічні образи) можна продемонструвати й інші алгебричні перетворення [6, 7].

Приклад 1. Мінімізувати логічну функцію f(a,b,c,d) = (3,7,11,12,13,14,15) алгебричним методом.

3	0011	a\`b\`cd
7	0111	a\`bcd
11	1011	ab\`cd
12	1100	abc\`d\`
13	1101	abc\`d
14	1110	abcd\`
15	1111	abcd

f(a,b,c,d) = (3,7,11,12,13,14,15) =
= a\`b\`cd + a\`bcd + ab\`cd + abc\`d\`+ abc\`d + abcd\` + abcd =
= cd(a\`b\` + a\`b + ab\`) + ab(c\`d\` + c\`d + cd\` + cd) =
= cd(a\`[b\` + b] + ab\`) + ab(c\`[d\` + d] + c[d\` + d]) =
= cd(a\`[1] + ab\`) + ab(c\`[1] + c[1]) = ab+ab\`cd + a\`cd =
= ab+cd(ab\` + a\`) = ab+ cd(a + a\`)(a\`+b\`) = ab + a\`cd + b\`cd =
= ab +cd(a\` + b\`) = ab +cd

Приклад 2. Мінімізувати логічну функцію f(a,b,c,d) = (3,7,11,12,13,14,15) методом Різника - Соломка.

$$f(a.b.c.d) = \left|\begin{array}{c|cccc} 3 & 0 & 0 & 1 & 1 \\ 7 & 0 & 1 & 1 & 1 \\ 11 & 1 & 0 & 1 & 1 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & & \end{array}\right| = \left|\begin{array}{cccc} & 0 & 1 & 1 \\ & 1 & 1 & 1 \\ 1 & 1 & & \end{array}\right| = \left|\begin{array}{cccc} & & 1 & 1 \\ 1 & 1 & & \end{array}\right|$$

Мінімізована функція – $F = ab + cd$. Операція супер-склеювання змінних у першій матриці проведена для блоків 12 – 15, які виділені червоним кольором. Результат мінімізації методом Різника – Соломка (приклад 2) збігається з результатом мінімізації, отриманим за допомогою алгебричного методу (приклад 1), однак процес мінімізації булевої функції методом Різника – Соломка є простішим, порівняно з мінімізацією логічної функції алгебричним методом.

Приклад 3. Мінімізувати логічну функцію $F(x_1, x_2, x_3, x_4)$ методом Різника - Соломка, яка задана наступною таблицею істинності (0,1,2,3,5,7,8,10,11,12,13) [16].

У [16] мінімізація функції зводиться до синтезу інфімумної диз'юктивної нормальної форми (ІДНФ) логічної функції, за допомогою досконалого матричного розміщення (ДМР) 4-вимірного куба E^4 (рис. 1). Вершини куба E^4 заданої функції, на яких $F(x_1, x_2, x_3, x_4) = 1$ виділені затемненням. Затемнені вершини відповідають блокам таблиці істинності (0,1,2,3,5,7,8,10,11,12,13) логічної функції.

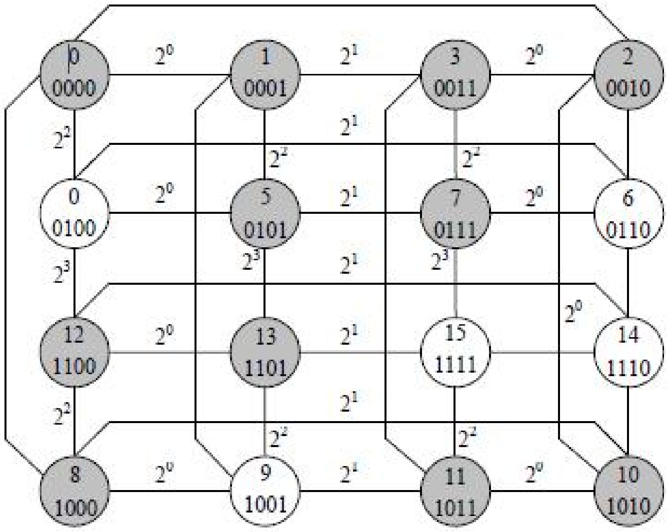

Рис. 1. Досконале матричне розміщення 4-вимірного куба E^4

Мінімізація функції $F(x_1, x_2, x_3, x_4)$ методом Різника – Соломка зводиться до наступної процедури.

$$F = \left|\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 1 \\ 5 & 0 & 1 & 0 & 1 \\ 7 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 \\ 10 & 1 & 0 & 1 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \end{array}\right| = \left|\begin{array}{cccc} & 0 & 0 & 0 \\ 0 & & 0 & 1 \\ & 0 & 1 & \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & \end{array}\right| = \left|\begin{array}{cccc} & 0 & & 0 \\ 0 & & 0 & 1 \\ & 0 & 1 & \\ 0 & & 1 & 1 \\ 1 & 1 & 0 & \end{array}\right| = \left|\begin{array}{cccc} & 0 & & 0 \\ 0 & & & 1 \\ & 0 & 1 & \\ 1 & 1 & 0 & \end{array}\right|.$$

Блоки 2,3,10,11 (виділені червоним кольором) мінімізовані за протоколом супер-склеювання змінних. Інші блоки мінімізовані за протоколами простого склеювання та напівсклеювання.

Мінімізована функція

$$F = \overline{x_2}\,\overline{x_4} + \overline{x_1} x_4 + \overline{x_2} x_3 + x_1 x_2 \overline{x_3}. \qquad (1)$$

Результат мінімізації (1) збігається з результатом синтезу інфімумної диз’юктивної нормальної форми логічної функції [16], однак метод Різника – Соломка є більш простою процедурою.

Мета та задачі дослідження

Метою роботи є спрощення процесу мінімізації 5-розрядних булевих функції, використовуючи логічну операцію – супер-склеювання змінних.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1) встановити адекватність застосування логічної операції супер-склеювання змінних для процесу мінімізації булевих функцій;

2) визначити властивості операції супер-склеювання змінних при використанні структур повної та неповної бінарних комбінаторних систем з повторенням;

3) провести порівнянний аналіз продуктивності та якості алгоритму мінімізації булевих функцій, отриманих методом Різника – Соломка, з прикладами мінімізації функції іншими методами.

Виклад основного матеріалу та обговорення отриманих результатів дослідження

Бінарна комбінаторна система з повторенням

Якщо задана деяка множина *A*, то множина всіх її підмножин, що мають *k* елементів $M_k(A)$ і число *N* всіх *k*-елементних підмножин множини із *n* елементів, дорівнює

$$N(M_k(A)) = C_n^k = \frac{n!}{k!(n-k)!}.$$

Крім цього має місце $\sum_{k=0}^{n} C_n^k = 2^n$. (2)

Оскільки C_n^k – число *k*-елементних підмножин множини з *n* елементів, то сума у лівій частині виразу (2) дорівнює числу всіх підмножин. За формулою (2) легко можна обчислити кількість всіх підмножин множини $A = \{a, b, c, d\}$:

$$N(M(A)) = C_4^0 + C_4^1 + C_4^2 + C_4^3 + C_4^4 = 1+4+6+4+1 = 16 = 2^4.$$

Зазначимо, що множина $A = \{a, b, c, d\}$, крім перерахунку своїх елементів, може визначати номери позицій, на яких знаходиться елемент α. Так, наприклад, *a* може означати першу, *b* - другу позицію множини $A = \{a, b, c, d\}$ і т. д. Підмножинами множини $A = \{a, b, c, d\}$ у такому випадку будуть підмножини, що містять елемент α на *k* позиціях, $k = 0, \ldots, n$, де *n* – кількість позицій множини *A*. У загальному випадку елемент α може займати декілька позицій на множині *A*, і таким чином елемент α повторюється на множині *A*.

Нехай α = 1, тоді позиції, на яких відсутній елемент α слід позначати нулями.

Приклад 4. Для множини $A = \{a, b, c, d\}$, що визначає номери позицій, приймемо α = 1. Тоді підмножини множини A будуть мати такий вигляд:

$$\begin{array}{ll} (0,0,0,0); & (1,0,0,0); \\ (0,0,0,1); & (1,0,0,1); \\ (0,0,1,0); & (1,0,1,0); \\ (0,0,1,1); & (1,0,1,1); \\ (0,1,0,0); & (1,1,0,0); \\ (0,1,0,1); & (1,1,0,1); \\ (0,1,1,0); & (1,1,1,0); \\ (0,1,1,1); & (1,1,1,1). \end{array} \qquad (3)$$

Конфігурація (3) складає повну комбінаторну систему з повторенням елемента α, яку назвемо 2-(*n*,*b*)- design, де *n* – розрядність блоку системи, *b* – кількість блоків повної системи, що визначається за формулою $b = 2^n$, число 2 перед дужками означає бінарну структуру конфігурації (3). Наприклад, 2-(4,16)- design є повною бінарною комбінаторною системою з повторенням, що складається з 4-

розрядних блоків, загальна кількість блоків – 16.

Алгебрична операція супер-склеювання змінних

Комбінаторні властивості блок – схеми з повторенням дозволяють доповнити правило алгебри логіки склеювання змінних правилом супер-склеювання змінних [7].

Для 5-розрядної логічної функції правила супер-склеювання змінних будуть такі:

– перше правило:

$$\begin{vmatrix} 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 1 & x \\ 0 & 0 & 1 & 0 & x \\ 0 & 0 & 1 & 1 & x \\ 0 & 1 & 0 & 0 & x \\ 0 & 1 & 0 & 1 & x \\ 0 & 1 & 1 & 0 & x \\ 0 & 1 & 1 & 1 & x \\ 1 & 0 & 0 & 0 & x \\ 1 & 0 & 0 & 1 & x \\ 1 & 0 & 1 & 0 & x \\ 1 & 0 & 1 & 1 & x \\ 1 & 1 & 0 & 0 & x \\ 1 & 1 & 0 & 1 & x \\ 1 & 1 & 1 & 0 & x \\ 1 & 1 & 1 & 1 & x \end{vmatrix} = x; \tag{4}$$

– друге правило:

$$\begin{vmatrix} 0 & 0 & 0 & x & y \\ 0 & 0 & 1 & x & y \\ 0 & 1 & 0 & x & y \\ 0 & 1 & 1 & x & y \\ 1 & 0 & 0 & x & y \\ 1 & 0 & 1 & x & y \\ 1 & 1 & 0 & x & y \\ 1 & 1 & 1 & x & y \end{vmatrix} = xy; \tag{5}$$

– третє правило:

$$\begin{vmatrix} 0 & 0 & x & y & z \\ 0 & 1 & x & y & z \\ 1 & 0 & x & y & z \\ 1 & 1 & x & y & z \end{vmatrix} = xyz; \tag{6}$$

– четверте правило:

$$\begin{vmatrix} 0 & x & y & z & t \\ 1 & x & y & z & t \end{vmatrix} = xyzt. \tag{7}$$

Перше правило (4) використовує 2-(4, 16)-design. Друге правило (5) використовує 2-(3, 8)-design. Третє правило (6) використовує 2-(2, 4)-design. Четверте правило (7) використовує 2-(1, 2)-design.

Змінні *x*, *y*, *z*, *t*, що утворюють повну бінарну комбінаторну систему з повторенням 2-(*n*, *b*)-design, можуть займати будь-який розряд мінтерма 5-розрядної логічної функції.

Процедура скорочення повної досконалої диз'юнктивної нормальної форми (ДДНФ) логічної функції дає одиницю. Наприклад, для 3-ох змінних:

$$\overline{x_1}\,\overline{x_2}\,\overline{x_3}+\overline{x_1}\,\overline{x_2}x_3+\overline{x_1}x_2\overline{x_3}+\overline{x_1}x_2x_3+x_1\overline{x_2}\,\overline{x_3}+x_1\overline{x_2}x_3+x_1x_2\overline{x_3}+x_1x_2x_3=$$
$$=\overline{x_1}\,\overline{x_2}\left(\overline{x_3}+x_3\right)+\overline{x_1}x_2\left(\overline{x_3}+x_3\right)+x_1\overline{x_2}\left(\overline{x_3}+x_3\right)+x_1x_2\left(\overline{x_3}+x_3\right)=$$
$$=\overline{x_1}\,\overline{x_2}+\overline{x_1}x_2+x_1\overline{x_2}+x_1x_2=\overline{x_1}\left(\overline{x_2}+x_2\right)+x_1\left(\overline{x_2}+x_2\right)=\overline{x_1}+x_1=1.$$

Оскільки повна ДДНФ однозначно визначає повну комбінаторну систему з повторенням 2-(*n*, *b*)-design, і навпаки, це дає підставу видаляти всі блоки повної комбінаторної системи з матриць, які демонструють правила супер-склеювання (4) – (7). Правило (7) проявляє просте склеювання та є частковим випадком правил супер-склеювання (4) – (6).

Правила супер-склеювання змінних можна представити для функцій щість змінних і більше.

У загальному випадку конфігурація таблиці істинності крім підматриці 2-(*n*,*b*)-design, вміщує й підматриці 2-(*n*, *x/b*)-design, де *x* – число блоків неповної комбінаторної системи з повторенням. Властивості 2-(*n*, *x/b*)-design також встановлюють правила, що забезпечують ефективну мінімізацію булевих функцій [15].

Мінімізація 5-розрядних булевих функцій

Приклад 5. Мінімізувати логічну функцію $F(x_1,x_2,x_3,x_4,x_5)$ методом Різника - Соломка, яка задана наступною таблицею істинності (2,3,4,5,6,7,10,11,12,13,14,15,18,19,20,21,22,23,16,17, 28, 29,30,31).

$$F=\left|\begin{array}{c|ccccc} 2&0&0&0&1&0\\ 3&0&0&0&1&1\\ 4&0&0&1&0&0\\ 5&0&0&1&0&1\\ 6&0&0&1&1&0\\ 7&0&0&1&1&1\\ 10&0&1&0&1&0\\ 11&0&1&0&1&1\\ 12&0&1&1&0&0\\ 13&0&1&1&0&1\\ 14&0&1&1&1&0\\ 15&0&1&1&1&1\\ 18&1&0&0&1&0\\ 19&1&0&0&1&1\\ 20&1&0&1&0&0\\ 21&1&0&1&0&1\\ 22&1&0&1&1&0\\ 23&1&0&1&1&1\\ 26&1&1&0&1&0\\ 27&1&1&0&1&1\\ 28&1&1&1&0&0\\ 29&1&1&1&0&1\\ 30&1&1&1&1&0\\ 31&1&1&1&1&1 \end{array}\right|=\left|\begin{array}{cccc} 0&0&0&1\\ 0&0&1&\\ 0&1&0&1\\ 0&1&1&\\ 1&0&0&1\\ 1&0&1&\\ 1&1&0&1\\ 1&1&1& \end{array}\right|\left|\begin{array}{c} \bullet\\ *\\ \bullet\\ \circ\\ \bullet\\ *\\ \bullet\\ \circ \end{array}\right|=\left|\begin{array}{ccc} &0&1\\ 0&1&\\ 1&1& \end{array}\right|\left|\begin{array}{c} \bullet\\ *\\ \circ \end{array}\right|=\left|\begin{array}{cc} 0&1\\ 1& \end{array}\right|=\left|\begin{array}{cc} &1\\ 1& \end{array}\right|$$

Мінімізована функція – $F=x_3+x_4$. Операція супер-склеювання змінних у першій матриці проведена для блоків:

- 4 – 7 (виділені синім кольором);
- 12 – 15 (виділені зеленим кольором);

- 20 – 23 (виділені бузинковим кольором);
- 28 – 31 (виділені фіолетовим кольором).

Операція супер-склеювання змінних у другій матриці проведена для блоків, які позначені значком ($\bullet$).

Приклад 6. Мінімізувати логічну функцію $F(x_1,x_2,x_3,x_4,x_5)$ методом Різника - Соломка, яка задана наступною таблицею істинності (0,1,3,4,7,13,15,19,20,22,23,29,31) [11].

$$F = \begin{array}{c|ccccc|} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \\ 7 & 0 & 0 & 1 & 1 & 1 \\ 13 & 0 & 1 & 0 & 1 & 1 \\ 15 & 0 & 1 & 1 & 1 & 1 \\ 19 & 1 & 0 & 0 & 1 & 0 \\ 20 & 1 & 0 & 1 & 0 & 1 \\ 22 & 1 & 0 & 1 & 1 & 0 \\ 23 & 1 & 0 & 1 & 1 & 1 \\ 29 & 1 & 1 & 0 & 1 & 1 \\ 31 & 1 & 1 & 1 & 1 & 1 \end{array} = \begin{array}{|ccccc|} 0 & 0 & 0 & 0 & \\ & 0 & 0 & 1 & 0 \\ & 0 & 1 & 0 & 1 \\ & 1 & 0 & 1 & 1 \\ & & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{array} = \begin{array}{|ccccc|} 0 & 0 & 0 & 0 & \\ & 0 & 0 & 1 & 0 \\ & 0 & 1 & & 1 \\ & 1 & & 1 & 1 \\ & & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & \end{array} = \begin{array}{|ccccc|} 0 & 0 & 0 & 0 & \\ & 0 & 0 & 1 & 0 \\ & 0 & 1 & & 1 \\ & 1 & & 1 & 1 \\ 1 & 0 & 1 & 1 & \end{array}$$

Мінімізована функція – $F = \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} + \overline{x_2}\,\overline{x_3} x_4 \overline{x_5} + \overline{x_2} x_3 x_5 + x_2 x_4 x_5 + x_1 \overline{x_2} x_3 x_4$. Операція супер-склеювання змінних у першій матриці проведена для блоків 7, 15, 23, 31, які виділені червоним кольором. Результат мінімізації методом Різника – Соломка збігається з результатом мінімізації методом скорочення таблиці істинності [11], однак процес мінімізації методом Різника – Соломка є простішим.

Приклад 7. Мінімізувати логічну функцію $F(x_1,x_2,x_3,x_4,x_5)$ методом Різника - Соломка, яка задана наступною таблицею істинності (0,4,5,13,16,21,22,23,24,25,28,29,30,31) [11].

$$F = \begin{array}{c|ccccc|} 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \\ 13 & 0 & 1 & 1 & 0 & 1 \\ 16 & 1 & 0 & 0 & 0 & 0 \\ 21 & 1 & 0 & 1 & 0 & 1 \\ 22 & 1 & 0 & 1 & 1 & 0 \\ 23 & 1 & 0 & 1 & 1 & 1 \\ 24 & 1 & 1 & 0 & 0 & 0 \\ 25 & 1 & 1 & 0 & 0 & 1 \\ 28 & 1 & 1 & 1 & 0 & 0 \\ 29 & 1 & 1 & 1 & 0 & 1 \\ 30 & 1 & 1 & 1 & 1 & 0 \\ 31 & 1 & 1 & 1 & 1 & 1 \end{array} = \begin{array}{|ccccc|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ & & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & \\ 1 & 1 & 0 & 0 & \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & \end{array} = \begin{array}{|ccccc|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \\ & & 1 & 0 & 1 \\ 1 & & 1 & 1 & \\ 1 & 1 & 0 & 0 & \\ 1 & 1 & 1 & 0 & \end{array} = \begin{array}{|ccccc|} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \\ & & 1 & 0 & 1 \\ 1 & & 1 & 1 & \\ 1 & 1 & & 0 & \end{array}$$

Мінімізована функція – $F = \overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5} + \overline{x_1}\,\overline{x_2} x_3 \overline{x_4} + x_3 \overline{x_4} x_5 + x_1 x_3 x_4 + x_1 x_2 \overline{x_4}$. Операція супер-склеювання змінних у першій матриці проведена для блоків 5, 13, 21, 29, які виділені синім кольором. У табл. 1 подані результати мінімізації функції $F(x_1,x_2,x_3,x_4,x_5)$ методом скорочення таблиці істинності [11] та методом Різника - Соломка.

Таблиця 1

Результат мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$

Мінімізація методом скорочення таблиці істинності
$F(x_1,x_2,x_3,x_4,x_5) = x_1x_3x_4 + x_1x_2\overline{x_4} + \overline{x_1}\,\overline{x_2}\,\overline{x_4}\,\overline{x_5} + \overline{x_2}x_3\overline{x_4}x_5 + x_1x_3\overline{x_4}\,\overline{x_5}$
Мінімізація методом Різника - Соломка
$F(x_1,x_2,x_3,x_4,x_5) = x_1x_3x_4 + x_1x_2\overline{x_4} + \overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5} + \overline{x_1}\,\overline{x_2}x_3\overline{x_4} + x_3\overline{x_4}x_5$

З огляду табл. 1 бачимо, що метод Різника-Соломка а останньому мінтермі мінімізованої функції дає на одну вхідну змінну менше.

Висновки

1. Алгебрична операція супер-склеювання змінних здійснюється при наявності у структурі таблиці істинності повної бінарної комбінаторної системи з повторенням або неповної бінарної комбінаторної системи з повторенням. Операція супер-склеювання змінних є найбільш ефективною за наявності повної бінарної комбінаторної системи з повторенням. Ефективність операції супер-склеювання змінних за наявності неповної бінарної комбінаторної системи з повторенням зменшується не суттєво.

2. Встановлено, що результати верифікації мінімізованої функції, отриманої з використанням правила супер-склеювання змінних, задовольняють вихідний протокол обчислення заданої функції і, отже, засвідчують оптимальне зменшення кількості змінних функції без втрати її функціональності. Впровадження алгебричної операції супер-склеювання змінних дає змогу спростити процедуру мінімізації булевої функції без втрати її функціональності

3. Ефективність методу Різника - Соломка демонструється прикладами мінімізації функції, запозичених з робіт інших авторів з метою порівняння:

– *приклад* 6 [11] – мінімізація 5-розрядних булевих функцій;

– *приклад* 7 [11] – мінімізація 5-розрядних булевих функцій;

З огляду на зазначені приклади та приклади 2, 3 і 5 метод мінімізації функції Різника - Соломка дає підставу для доцільності застосування його у процесах мінімізації логічних функцій.

1. Матвієнко, М. П. Комп'ютерна логіка [Текст] / М. П. Матвієнко – Київ: ТОВ "Центр навчальної літератури", 2012. – 288 с.
2. Игошин, В. И. Математическая логика и теория алгоритмов [Учебное пособие] / В. И. Игошин – М.: Изд.центр "Академия", 2007. – 304 с.
3. Кутюра, Л. Алгебра логики [Текст] / Л. Кутюра – М.: Либроком, 2011 – 128 с.
4. Колмогоров, А. Н. Математическая логика. [Текст] / А. Н. Колмогоров, А. Г. Драгалин – М.: КомКнига, Изд. 3-е, стереотипное, 2006. – 240 с.
5. Venn, J. (July 1880). On the Diagrammatic and Mechanical Representation of Propositions and Reasonings. Philosophical Magazine and Journal of Science. 5 10 (59).
6. Riznyk V. Minimization of boolean functions by combinatorial method [Text] / V. Riznyk, M. Solomko // Technology audit and production reserves.- Vol 4/2 (36), 2017. – P. 49–64. http://journals.uran.ua/tarp/article/view/108532 .
7. Riznyk V. Application of super-sticking algebraic operation of variables for boolean functions minimization by combinatorial method [Text] / V. Riznyk, M. Solomko // Technology audit and production reserves.- Vol 6/2 (38), 2017. P. 60 – 76. http://journals.uran.ua/tarp/article/view/118336/112951
8. Manojlović, Vladislav (2013) Minimization of Switching Functions using Quine-McCluskey Method. *International Journal of Computer Applications (0975 – 8887) Volume 82 – No 4, November 2013*, 12-16. (2010) http://research.ijcaonline.org/volume82/number4/pxc3892127.pdf
9. Eungi Kim (2013) Derivations of Single Hypothetical Don't-Care Minterms Using the Quasi Quine-McCluskey Method. *J Korea Industr Inf Syst Res Volume 18 Number 1 February 2013* 25-35 https://www.researchgate.net/publication/267819138_An_Innovative_procedure_to_minimize_Boolean_function
10. Rytsar, Bohdan (2015) The Minimization Method of Boolean Functionns in Polynomial Set-theoretical Format. Conference: Proc. 24th Inter. Workshop, CS@P'2015, Sept. 28-30, 2015, 130-146 pp. (17), At Rzeszow, Poland, Volume: vol.2 http://dspace.nbuv.gov.ua/handle/123456789/87194
11. Rathore, T. S. (2014) Minimal Realizations of Logic Functions Using Truth Table Method with Distributed Simplification. IETE JOURNAL OF EDUCATION, VOL 55, NO 1, JAN_JUN 2014, 26-32 http://www.tandfonline.com/doi/pdf/10.1080/09747338.2014.921412

12. Dan, Rotar (2010) Software for The Minimization of The Combinational Logic Functions. The Romanian Review Precision Mechanics, Optics & Mechatronics, 2010 (20), No. 37, 95-99
https://www.researchgate.net/publication/268270733_Software_for_The_Minimization_of_The_Combinational_Logic_Functions_SOFTWARE_FOR_THE_MINIMIZATION_OF_THE_COMBINATIONAL_LOGIC_FUNCTIONS
13. Zolfaghari, Behrouz, Sheidaeian, Hamed (2011) A NEW CASE FOR IMAGE COMPRESSION USING LOGIC FUNCTION MINIMIZATION. The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.2, May 2011, 45-62. http://aircconline.com/ijma/V3N2/3211ijma04.pdf
14. Nosrati M., Karimi R., Nariri M. (2012) MINIMIZATION OF BOOLEAN FUNCTIONS USING GENETIC ALGORITHM. Anale. Seria Informatica. Vol. X fasc. 1 – 2012, 73–77. https://pdfs.semanticscholar.org/c53d/2240a2aa5531832a7707ad186dee23129ed8.pdf
15. Nosrati M., Karimi R. (2011) An Algorithm for Minimizing of Boolean Functions Based on Graph DS. World Applied Programming, Vol (1), No (3), August 2011. 209–214. http://waprogramming.com/papers/50ae59d04ee143.95681909.pdf
16. Иванов Ю.Д. Алгоритм синтеза инфимумных дизъюктивных нормальных форм логических функций [Текст] / Ю.Д. Иванов, О.С. Захарова // Труды Одесского политехнического университета, 2004, вып. 2(22). ст. 1-7. http://pratsi.opu.ua/app/webroot/articles/1313074117.pdf