DOI: 10.15587/2312-8372.2017.118336

ЗАСТОСУВАННЯ АЛГЕБРИЧНОЇ ОПЕРАЦІЇ СУПЕР-СКЛЕЮВАННЯ ЗМІННИХ ДЛЯ МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ КОМБІНАТОРНИМ МЕТОДОМ

Різник В. В., Соломко М. Т.

1. Вступ

Мінімізація булевих функцій все ще популярна в різних областях цифрових технологій, таких як дизайн PLA, вбудований самотест (BIST), проектування систем управління тощо.

Проблема мінімізації диз'юнктивної нормальної форми (ДНФ) є однією з багато-екстремальних 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 (ДКНФ).

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. Зокрема, карта Карно зазвичай важко піддається розпізнаванню при зростанні кількості змінних більше чотирьох-п'яти, тому цей метод недоцільно використовувати з більшим числом змінних. Незважаючи на більшу досконалість методу Куайна-Мак-Класкі порівняно з картами Карно, він також має обмежене практичне застосування з-за експоненціального зростання часу обчислення зі збільшенням кількості змінних. Можна показати, що для функції від n змінних верхня межа кількості основних імплікант дорівнює $3^n ln(n)$ [1]. Наприклад, відомо, що для n=32 кількість основних імплікант може перевершувати $6{,}5 \cdot 10^{15}$.

Від результату мінімізації булевої функції залежить швидкодія обчислювального пристрою, його надійність та енергозбереження.

Особливості комбінаторного методу [2] 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 комбінаторним методом 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 практично важливих проблем, яка постає під час проектування обчислювальних пристроїв. Тому вивчення нових правил алгебри логіки для спрощення алгоритму мінімізації булевої функції без втрати її функціональності при збільшенні кількості змінних є актуальним.

2. Об'єкт дослідження та його технологічний аудит

Об'єктом спрощення задачі мінімізації булевої функції комбінаторним методом є нова процедура алгебри логіки – *супер-склеювання* змінних, яка здійснюється за наявності у структурі таблиці істинності повної бінарної комбінаторної системи з повторенням або неповної бінарної комбінаторної системи з повторенням.

Процедура скорочення повної досконалої диз'юнктивної нормальної форми

(ДДНФ) логічної функції дає одиницю. 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 застосовувати правила супер-склеювання змінних.

Процедура простого склеювання змінних є частковим випадком процедури супер-склеювання змінних. Змінні, що утворюють повну бінарну комбінаторну систему з повторенням або неповну бінарну комбінаторну систему з повторенням, можуть займати будь-який розряд мінтерма логічної функції.

Ефективність алгебричної операції супер-склеювання змінних суттєво спрощує алгоритм мінімізації булевої функції, що дозволяє здійснювати ручну мінімізацію функцій з числом змінних до 10. Середня складність алгоритму мінімізації логічної функції комбінаторним методом з використанням процедури супер-склеювання змінних оцінюється за динамікою зростання кількості образних перетворень комбінаторного методу мінімізації зі збільшенням розрядності булевої функції. Для n<7 динаміка характеризується лінійним законом $O(n)$, а зі збільшенням числа змінних до 10 – за $O(n^2)$.

Недоліки комбінаторного методу ручної мінімізації з використанням процедури супер-склеювання змінних пов’язані зі швидким зростанням алгоритмічної складності при збільшенні числа змінних логічної функції. Мінімізація функції з числом змінних більше 12÷14 потребує оновлення бібліотеки підматриць, на яких ґрунтується процедура супер-склеювання.

3. Мета та задачі дослідження

Метою роботи є спрощення комбінаторного методу мінімізації булевої функції за допомогою нової процедуру алгебри логіки – супер-склеювання змінних та встановлення властивостей такої процедури.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Встановити адекватність застосування алгебричної процедури супер-склеювання змінних для процесу мінімізації булевої функції.
2. Визначити властивості операції супер-склеювання змінних при використанні структур повної бінарної комбінаторної системи з повторенням та неповної бінарної комбінаторної системи з повторенням.
3. Здійснити верифікацію комбінаторного методу при застосуванні правила супер-склеювання змінних та оцінити складність алгоритму мінімізації булевої функції комбінаторним методом.
4. Провести порівняльний аналіз продуктивності та складності алгоритмів мінімізації булевих функцій, отриманих за допомогою правила супер-склеювання змінних, з іншими методами мінімізації.

4. Дослідження існуючих рішень проблеми

Класичний об’єктно-орієнтований алгоритм мінімізації булевої функції за допомогою карт Карно описаний у [3], де представлені мовні стереотипи та діаграми класів, а також поданий аналіз продуктивності уніфікованої моделі мінімізації булевої функції. У [4] розглянуто кубічні методи мінімізації булевих функцій як ще один варіант пошуку мінімальної функції. Головна мета роботи [4] полягає в тому, щоб використати переваги кубічного методу мінімізації логічних функцій, зокрема через досягнення мінімальної вартості рішення.

Швидкий і ефективний евристичний алгоритм мінімізації булевих функцій (ESOP) розглянуто у [5]. Цей алгоритм ґрунтується на нових перетвореннях куба. Його автори доводять, що якість відповідного покриття відповідає, а в окремих випадках перевершує сучасний рівень евристичної мінімізації. У роботі [6] представлений *розширений* QMC алгоритм (*e* QMC), який підвищує продуктивність процесу мінімізації булевої функції методом Квайна-Мак-Класкі. У роботі подана демонстрація збільшення швидкості та продуктивності роботи пам'яті комп'ютера шляхом моделювання процесу мінімізації функції.

У публікації [7] представлений ефективний алгоритм синтезу та точної мінімізації ESCT (Exclusive or Sum of Complex Terms) булевих функцій не більше шести змінних. Цей вид логічних виразів можна перетворити на спеціальну стільникову архітектуру, яку називають архітектурою реверсивної хвилі каскаду. Доведено, що така топологія є оборотною та може допомогти у розробці квантових схем. Запропонований алгоритм є першим, який дає рішення проблеми пошуку мінімальних виразів ESCT для перемикання функції до шести вхідних змінних. Комплексне обстеження методів мінімізації логічних функцій демонструється у роботі [8]. Методи розглядаються за їх метою, методологією, реалізацією та перевагами. Представлено порівняння переглянутих підходів до мінімізації логічних функцій.

Нова техніка двоетапного процесу оптимізації комбінаційної логіки описується у [9]. Техніка може бути застосована до довільних комбінаційних логічних завдань, і часто приносить поліпшення навіть після оптимізації за стандартними методами. Зазначена техніка оптимізації використовується для підвищення продуктивності програмного забезпечення. У [10] демонструється метод, коли процес оптимізації може включати в себе не тільки пошук еквівалентного логічного виразу, але й визначення конкретних умов, за яких логічні вирази вдається 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 у підсумку може забезпечити оптимальний булевий вираз. У роботі [11] представлена багатозначна логіка як узагальнення класичної булевої логіки на більш високих рівнях абстракції, де змінні часто варіюються над набором символічних кодів. Використання багатозначної логіки може зробити завдання дизайну більш інтуїтивним. Дизайнер може спочатку маніпулювати та оптимізувати багатозначну логіку, а потім виконати відповідне кодування і вивести задачу до булевої алгебри. Це дозволяє краще вивчати дизайн-простір, оскільки бінарне кодування відкладено і багатозначна оптимізація не впливає на достовірність таких рішень на заключному етапі. В роботі описані спроби побудувати багатозначні інтегральні схеми (IC), починаючи з 3-х значних конструкцій, що можна простежити до 1970 року. У [12] представлена оптимізація схеми 2-бітного компаратора шляхом порівняння різних логічних стилів, які використовуються для проектування схеми компаратора. Порівняння між різними конструкціями обчислюється моделюванням, яке виконується для технології 90 nm в Tanner EDA Tool. Після симуляції всіх проектів отримуються остаточні результати стосовно споживання електроенергії, затримки сигналу, потужності. Зокрема порівнюються PTL, NMOS, CMOS технології.

На відміну від публікацій [3–12], у даній роботі об'єктом спрощення процесу мінімізації булевої функції є нова процедура алгебри логіки – супер-склеювання 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. 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 застосовувати 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 у вигляді двовимірних матриць за своїми властивостями мають більшу інформаційну ємність, тому спроможні з ефектом замінити вербальні процедури алгебричних перетворень.

5. Методи дослідження

5.1. Бінарна комбінаторна система з повторенням

Якщо задана деяка множина A, то можна розглядати нову множину $M(A)$ – множину всіх її підмножин. Через $M_k(A)$, будемо позначати множину всіх підмножин A, що мають k елементів.

Приклад 1. Нехай $A=\{a, b, c\}$, тоді:

$$M(A)=\{\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\},\varnothing\};$$
$$M_2(A)=\{\{a,b\},\{a,c\},\{b,c\}\}.$$

Переконуємося, що:

$$N(M(A))=8=2^3,\quad N(M_2(A))=3.$$

Число всіх k-елементних підмножин множини із n елементів дорівнює:

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

Має місце також рівність:

$$\sum_{k=0}^{n} C_n^k = 2^n. \tag{1}$$

Оскільки C_n^k – число k-елементних підмножин множини із n елементів, то сума у лівій частині виразу (1) є число всіх підмножин.

Приклад 2. За формулою (1) 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 також визначати номери позицій, на яких знаходиться елемент α. Так, наприклад, *a* може означати першу позицію, *b* може означати 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*, таким чином елемент α повторюється на множині *A*.

Нехай $\alpha=1$, тоді позиції, на яких відсутній елемент α, позначаються нулем.

Приклад 3. Для множини $A=\{a, b, c, d\}$, що визначає номери позицій, приймаємо $\alpha=1$. Тоді підмножини множини *A* будуть мати такий вигляд:

$$
\begin{array}{ll}
(0,0,0,0); & (1,0,0,0); \\
(0,0,0,1); & (1,0,0,1); \\
(0,0,1,0); & (1,0,1,0); \\
(0,0,1,1); & (1,0,1,1); \\
(0,1,0,0); & (1,1,0,0); \\
(0,1,0,1); & (1,1,0,1); \\
(0,1,1,0); & (1,1,1,0); \\
(0,1,1,1); & (1,1,1,1).
\end{array} \tag{2}
$$

Число всіх *k*-елементних підмножин множини $A=\{a, b, c, d\}$, що визначає номери позицій, обчислюється за формулою (1).

$$
\begin{aligned}
&N(M_0(A)) = C_4^0 = 1, \\
&N(M_1(A)) = C_4^1 = 4, \\
&N(M_2(A)) = C_4^2 = 6, \\
&N(M_3(A)) = C_4^3 = 4, \\
&N(M_3(A)) = C_4^4 = 1. \\
&N(M(A)) = N(M_0(A)) + N(M_1(A)) + N(M_2(A)) + N(M_3(A)) + N(M_4(A)) = 16.
\end{aligned}
$$

Конфігурація (2) складає повну комбінаторну систему з повторенням елемента α, яку позначимо:

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 (2). Наприклад, 2-(4, 16)-design є повна бінарна комбінаторна система з повторенням, що складається з 4-розрядних блоків, кількість блоків – 16.

5.2. Алгебрична операція супер-склеювання змінних

Комбінаторні властивості блок-схеми з повторенням дозволяють доповнити правило алгебри логіки склеювання змінних [2], правилом супер-склеювання змінних.

Для 4-розрядної логічної функції правило супер-склеювання змінних має такий вигляд:

– перше правило:

$$\begin{vmatrix} 0 & 0 & 0 & x \\ 0 & 0 & 1 & x \\ 0 & 1 & 0 & x \\ 0 & 1 & 1 & x \\ 1 & 0 & 0 & x \\ 1 & 0 & 1 & x \\ 1 & 1 & 0 & x \\ 1 & 1 & 1 & x \end{vmatrix} = x; \tag{3}$$

– друге правило:

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

– третє правило:

$$\begin{vmatrix} 0 & x & y & z \\ 1 & x & y & z \end{vmatrix} = xyz. \tag{5}$$

Перше правило використовує 2-(3, 8)-design. Друге правило використовує 2-(2, 4)-design. Третє правило використовує 2-(1, 2)-design.

Процедура скорочення повної досконалої диз'юктивної нормальної форми (ДДНФ) логічної функції дає одиницю. Наприклад, скорочення 3-розрядної повної ДДНФ виглядає так:

$$\begin{aligned} &\overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2 x_3 + \overline{x}_1 x_2\overline{x}_3 + \overline{x}_1 x_2 x_3 + x_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2 x_3 + x_1 x_2\overline{x}_3 + x_1 x_2 x_3 = \\ &= \overline{x}_1\overline{x}_2\left(\overline{x}_3 + x_3\right) + \overline{x}_1 x_2\left(\overline{x}_3 + x_3\right) + x_1\overline{x}_2\left(\overline{x}_3 + x_3\right) + x_1 x_2\left(\overline{x}_3 + x_3\right) = \\ &= \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 + x_1\overline{x}_2 + x_1 x_2 = \overline{x}_1\left(\overline{x}_2 + x_2\right) + x_1\left(\overline{x}_2 + x_2\right) = \overline{x}_1 + x_1 = 1. \end{aligned}$$

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 (3)–(5). Далі, застосувавши закон ідемподентності до змінної, що залишилась – *x* (*xy*; *xyz*) отримуємо результат скорочення за правилом супер-склеювання змінних. Правило (5) проявляє себе як просте склеювання змінних та є частковим випадком правил (3) і (4).

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

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

– перше правило:

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

– друге правило:

$$\begin{vmatrix} 0 & 0 & 0 & x & y \\ 0 & 0 & 1 & x & y \\ 0 & 1 & 0 & x & y \\ 0 & 1 & 1 & x & y \\ 1 & 0 & 0 & x & y \\ 1 & 0 & 1 & x & y \\ 1 & 1 & 0 & x & y \\ 1 & 1 & 1 & x & y \end{vmatrix} = xy; \tag{7}$$

– третє правило:

$$\begin{vmatrix} 0 & 0 & x & y & z \\ 0 & 1 & x & y & z \\ 1 & 0 & x & y & z \\ 1 & 1 & x & y & z \end{vmatrix} = xyz; \tag{8}$$

– четверте правило:

$$\begin{vmatrix} 0 & x & y & z & t \\ 1 & x & y & z & t \end{vmatrix} = xyzt. \tag{9}$$

Перше правило (6) використовує 2-(4, 16)-design. Друге правило (7) використовує 2-(3, 8)-design. Третє правило (8) використовує 2-(2, 4)-design. Четверте правило (9) 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.

Інший варіант застосування правила супер-склеювання змінних демонструє комбінаторна конфігурація, де використовується комбінаторна система 2-(3, 8)-design у варіанті конфігурації, коли присутній один стовпчик з однаковими змінними y, а другий стовпчик вміщує порівну змінні x та $\bar{x}$:

$$\begin{vmatrix} 0 & 0 & 0 & \bar{x} & y \\ 0 & 0 & 1 & \bar{x} & y \\ 0 & 1 & 0 & \bar{x} & y \\ 0 & 1 & 1 & \bar{x} & y \\ 1 & 0 & 0 & x & y \\ 1 & 0 & 1 & x & y \\ 1 & 1 & 0 & x & y \\ 1 & 1 & 1 & x & y \end{vmatrix} = \begin{vmatrix} 0 & & & \bar{x} & y \\ 1 & & & x & y \end{vmatrix}. \qquad (10)$$

Аналогічно правилам супер-склеювання змінних для функцій з чотирма чи п’ятьма змінними, можна представити правила супер-склеювання для функцій шести і більше змінних.

У загальному випадку конфігурація таблиці істинності заданої функції, крім підматриці повної комбінаторної системи з повторенням 2-(n, b)-design, вміщує й підматриці неповної комбінаторної системи з повторенням 2-(n, x/b)-design. У цьому випадку x – число блоків неповної комбінаторної системи з повторенням. Властивості неповної комбінаторної системи з повторенням 2-(n, x/b)-design дозволяють також встановлювати правила, що забезпечують ефективну мінімізацію булевих функцій.

Виділимо клас неповних комбінаторних систем з повторенням 2-(n, $b/2$)-design, у яких число блоків складає половину від можливої кількості блоків повної комбінаторної системи з повторенням. Для 2-(n, $b/2$)-design при n=3 правила мінімізації є такі:

A. $\begin{vmatrix} x & 0 & 0 & 1 \\ x & 0 & 1 & 1 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & & & 1 \end{vmatrix}.$

B. $\begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 1 & 0 \\ x & 1 & 0 & 0 \\ x & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} x & & & 0 \end{vmatrix}.$

C. $\begin{vmatrix} x & 1 & 0 & 0 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 0 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 1 & & \end{vmatrix}.$

D. $\begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 0 & 1 \\ x & 0 & 1 & 0 \\ x & 0 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & & \end{vmatrix}.$

E. $\begin{vmatrix} x & 0 & 1 & 0 \\ x & 0 & 1 & 1 \\ x & 1 & 0 & 0 \\ x & 1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 1 & \\ x & 0 & 1 & \\ x & 1 & 0 & \\ x & 1 & 0 & \end{vmatrix} = \begin{vmatrix} x & 0 & 1 & \\ x & 1 & 0 & \end{vmatrix}.$

F. $\begin{vmatrix} x & 0 & 1 & 1 \\ x & 1 & 0 & 0 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} x & 0 & 1 & 1 \\ x & 1 & 0 & \\ x & 1 & 0 & \\ x & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} x & 0 & 1 & 1 \\ x & 1 & 0 & \\ x & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} x & 0 & 1 & 1 \\ x & 1 & 0 & \\ x & 1 & & 0 \end{vmatrix}.$

Правила, подібні A – F, існують для всіх можливих 2-(*n*, *b*/2)-design з 2-(*n*, *b*)-design.

Для інших випадків 4-розрядної логічної функції, використовуючи структуру 2-(*n*, *x*/*b*)-design, правила мінімізації логічної функції можуть бути, наприклад, такі:

$$1.\ \begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 0 & 1 \\ x & 1 & 0 & 0 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 0 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & 1 & 0 & \\ x & 1 & 1 & \end{vmatrix}. \tag{11}$$

У першій матриці блок-схеми (11) наявна неповна комбінаторна система з повторенням 2-(3, 6/8)-design. Подальша мінімізація блок-схеми (11) можлива двома варіантами, кожен з яких дає однаковий результат:

1) $\begin{vmatrix} x & 0 & 0 & \\ x & 1 & 0 & \\ x & 1 & 1 & \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & 1 & & \end{vmatrix} = \begin{vmatrix} x & & 0 & \\ x & 1 & & \end{vmatrix},$

2) $\begin{vmatrix} x & 0 & 0 & \\ x & 1 & 0 & \\ x & 1 & 1 & \end{vmatrix} = \begin{vmatrix} x & & 0 & \\ x & 1 & 1 & \end{vmatrix} = \begin{vmatrix} x & & 0 & \\ x & 1 & & \end{vmatrix}.$

У двох вищенаведених варіантах першою здійснена операція склеювання змінних, другою проведена операція заміщення змінних. У підсумку для блок-схеми (11) отримуємо таке правило скорочення функції:

$$\begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 0 & 1 \\ x & 1 & 0 & 0 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 0 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & & 0 & \\ x & 1 & & \end{vmatrix}. \qquad (12)$$

2. $$\begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 0 & 1 \\ x & 0 & 1 & 1 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & 1 & 0 & 1 \\ x & & 1 & 1 \end{vmatrix}, \qquad (13)$$

або

$$\begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 0 & 1 \\ x & 0 & 1 & 1 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & 0 & 1 & 1 \\ x & 1 & & 1 \end{vmatrix}. \qquad (14)$$

У першій матриці блок-схеми (13) наявна неповна комбінаторна система з повторенням 2-(3, 5/8)-design. Подальша мінімізація блок-схеми (13) можлива двома варіантами, кожен з яких дає однаковий результат:

1) $$\begin{vmatrix} x & 0 & 0 & \\ x & 1 & 0 & 1 \\ x & & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & & 0 & 1 \\ x & & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & & & 1 \end{vmatrix},$$

2) $$\begin{vmatrix} x & 0 & 0 & \\ x & 1 & 0 & 1 \\ x & & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & 1 & & 1 \\ x & & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & 1 & & 1 \\ x & & 1 & 1 \\ x & & 0 & 1 \end{vmatrix} =$$
$$= \begin{vmatrix} x & 0 & 0 & \\ x & 1 & & 1 \\ x & & & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & & & 1 \end{vmatrix}.$$

У підсумку для блок-схеми (13) отримуємо таке правило скорочення функції:

$$\begin{vmatrix} x & 0 & 0 & 0 \\ x & 0 & 0 & 1 \\ x & 0 & 1 & 1 \\ x & 1 & 0 & 1 \\ x & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & 0 & 0 & \\ x & & & 1 \end{vmatrix}. \qquad (15)$$

$$3.\ \begin{vmatrix} 0 & 0 & x & 0 \\ 0 & 0 & x & 1 \\ 0 & 1 & x & 0 \\ 1 & 0 & x & 0 \\ 1 & 0 & x & 1 \\ 1 & 1 & x & 0 \\ 1 & 1 & x & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & x & \\ 0 & 1 & x & 0 \\ 1 & 0 & x & \\ 1 & 1 & x & \end{vmatrix} = \begin{vmatrix} 0 & 0 & x & \\ 0 & 1 & x & 0 \\ 1 & & x & \end{vmatrix}. \tag{16}$$

У першій матриці блок-схеми (16) наявна неповна комбінаторна система з повторенням 2-(3, 7/8)-design. Подальша мінімізація блок-схеми (16) можлива двома варіантами, кожен з яких дає однаковий результат.

$$1)\ \begin{vmatrix} 0 & 0 & x & \\ 0 & 1 & x & 0 \\ 1 & & x & \end{vmatrix} = \begin{vmatrix} 0 & 0 & x & \\ 0 & & x & 0 \\ 1 & & x & \end{vmatrix} = \begin{vmatrix} 0 & 0 & x & \\ & & x & 0 \\ 1 & & x & \end{vmatrix} = \begin{vmatrix} & 0 & x & \\ & & x & 0 \\ 1 & & x & \end{vmatrix},$$

$$2)\ \begin{vmatrix} 0 & 0 & x & \\ 0 & 1 & x & 0 \\ 1 & & x & \end{vmatrix} = \begin{vmatrix} 0 & 0 & x & \\ & 1 & x & 0 \\ 1 & & x & \end{vmatrix} = \begin{vmatrix} & 0 & x & \\ & 1 & x & 0 \\ 1 & & x & \end{vmatrix} = \begin{vmatrix} & 0 & x & \\ & & x & 0 \\ 1 & & x & \end{vmatrix}.$$

У підсумку для блок-схеми (16) отримуємо таке правило скорочення функції:

$$\begin{vmatrix} 0 & 0 & x & 0 \\ 0 & 0 & x & 1 \\ 0 & 1 & x & 0 \\ 1 & 0 & x & 0 \\ 1 & 0 & x & 1 \\ 1 & 1 & x & 0 \\ 1 & 1 & x & 1 \end{vmatrix} = \begin{vmatrix} & 0 & x & \\ & & x & 0 \\ 1 & & x & \end{vmatrix}. \tag{17}$$

$$4.\ \begin{vmatrix} 0 & 0 & x & 0 \\ 0 & 0 & x & 1 \\ 1 & 0 & x & 1 \\ 1 & 1 & x & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & x & \\ 1 & & x & 1 \end{vmatrix}. \tag{18}$$

$$5.\ \begin{vmatrix} x & y & 0 & 1 \\ x & y & 1 & 0 \\ x & y & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & y & 0 & 1 \\ x & y & 1 & \end{vmatrix} = \begin{vmatrix} x & y & & 1 \\ x & y & 1 & \end{vmatrix}. \tag{19}$$

$$6.\ \begin{vmatrix} x & y & 0 & 1 \\ x & y & 1 & 0 \\ x & y & 0 & 0 \end{vmatrix} = \begin{vmatrix} x & y & 0 & 1 \\ x & y & & 0 \end{vmatrix} = \begin{vmatrix} x & y & 0 & \\ x & y & & 0 \end{vmatrix}. \tag{20}$$

$$7.\ \begin{vmatrix} x & y & 1 & 1 \\ x & y & 1 & 0 \\ x & y & 0 & 0 \end{vmatrix} = \begin{vmatrix} x & y & 1 & 1 \\ x & y & & 0 \end{vmatrix} = \begin{vmatrix} x & y & 1 & \\ x & y & & 0 \end{vmatrix}. \tag{21}$$

$$8.\begin{vmatrix} x & y & 1 & 1 \\ x & y & 0 & 1 \\ x & y & 0 & 0 \end{vmatrix} = \begin{vmatrix} x & y & 1 & 1 \\ x & y & 0 & \end{vmatrix} = \begin{vmatrix} x & y & & 1 \\ x & y & 0 & \end{vmatrix}. \qquad (22)$$

Правила перетворення блок-схем (19)–(22) дозволяють встановлювати нові правила мінімізації, наприклад:

$$9.\begin{vmatrix} x & \bar{y} & 0 & 1 \\ x & \bar{y} & 1 & 0 \\ x & \bar{y} & 1 & 1 \\ x & y & 0 & 1 \\ x & y & 1 & 0 \\ x & y & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & & 1 \\ x & \bar{y} & 1 & \\ x & y & & 1 \\ x & y & 1 & \end{vmatrix} = \begin{vmatrix} x & & & 1 \\ x & & 1 & \end{vmatrix}. \qquad (23)$$

Правила A–F, (12), (15), (17)–(23) складають бібліотеку правил для процесу мінімізації булевих функцій як стандартні процедури, тому застосування окремого такого правила для змінних булевої функції зводиться до здійснення одного алгебричного перетворення.

5.3. Мінімізація 4-розрядних булевих функцій

Приклад 4. Мінімізувати логічну функцію $F(x_1,x_2,x_3,x_4)$ комбінаторним методом, яка задана наступною таблицею істинності (табл. 1) [13].

Таблиця 1

Таблиця істинності логічної функції $F(x_1,x_2,x_3,x_4)$

№ з/п	x_1	x_2	x_3	x_4	F	№ з/п	x_1	x_2	x_3	x_4	F
0	0	0	0	0	**1**	8	1	0	0	0	**0**
1	0	0	0	1	**1**	9	1	0	0	1	**1**
2	0	0	1	0	**1**	10	1	0	1	0	**1**
3	0	0	1	1	**1**	11	1	0	1	1	**1**
4	0	1	0	0	**0**	12	1	1	0	0	**0**
5	0	1	0	1	**0**	13	1	1	0	1	**1**
6	0	1	1	0	**1**	14	1	1	1	0	**1**
7	0	1	1	1	**0**	15	1	1	1	1	**1**

Складемо досконалу диз'юктивну нормальну форму (ДДНФ) заданої функції з блоків при яких функція отримує значення одиниці, тобто для наборів 0, 1, 2, 3, 6, 9, 10, 11, 13, 14, 15.

$$\begin{aligned} F(x_1,x_2,x_3,x_4) = {} & \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + \\ + {} & \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 + \\ + {} & x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + \\ + {} & x_1x_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4 + x_1x_2x_3x_4. \end{aligned}$$

Перший крок, це склеювання, заміщення та узагальнене склеювання змінних. З множини варіантів мінімізації, отриманих на першому кроці, розглянемо два варіанти мінімізації 4-розрядної функції.

1-й варіант: мінімізація функції з використанням правила супер-склеювання змінних за наявності повної бінарної комбінаторної системи з повторенням 2-(n, b)-design.

$$F = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ 0 & 1 & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ 0 & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{vmatrix} = \tag{24}$$

$$= \begin{vmatrix} 0 & 0 & & \\ & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ & & 1 & 0 \\ 1 & & & 1 \end{vmatrix}.$$

Для блоків 1–4 першої матриці блок-схеми (24) використано друге правило супер-склеювання (4), де наявна повна комбінаторна система з повторенням 2-(2, 4)-design; блок 5 не змінюється; для блоків 6–11 використано правило супер-склеювання (23), де наявна неповна бінарна комбінаторна система з повторенням 2-(3, 6/8)-design. Результат операції супер-склеювання записано у другу матрицю блок-схеми (24).

Алгебричні перетворення другої матриці (результат перетворення записаний у третю матрицю):

– заміщення змінних у першому та другому блоках другої матриці блок-схеми (24):

$$\overline{x_1}\,\overline{x_2} + \overline{x_1}x_2x_3\overline{x_4} = \overline{x_1}\left(\overline{x_2} + x_2x_3\overline{x_4}\right) =$$
$$= \overline{x_1}\left(\overline{x_2} + x_3\overline{x_4}\right) = \overline{x_1}\left(\overline{x_2} + \overline{x_1}x_3\overline{x_4}\right),$$

$$\begin{matrix} 0 & 0 & & \\ 0 & 1 & 1 & 0 \end{matrix} \to \begin{matrix} 0 & 0 & & \\ 0 & & 1 & 0 \end{matrix}.$$

Алгебричні перетворення третьої матриці, результат якої записаний у четверту матрицю:

– заміщення змінних у другому та четвертому блоках матриці блок-схеми (24):

$$\overline{x_1}x_3\overline{x_4} + x_1x_3 = x_3\left(\overline{x_1}\,\overline{x_4} + x_1\right) =$$
$$= x_3\left(\overline{x_4} + x_1\right) = x_1x_3 + x_3\overline{x_4},$$

$$\begin{array}{cccc} 0 & & 1 & 0 \\ 1 & & 1 & \end{array} \rightarrow \begin{array}{cccc} & & 1 & 0 \\ 1 & & 1 & \end{array}.$$

Алгебричні перетворення четвертої матриці, результат якої записаний у п'яту матрицю:
– узагальнене склеювання змінних 2-го, 3-го та 4-го блоків 4-ї матриці 3 (24):

$$x_3\overline{x_4} + x_1x_4 + x_1x_3 = x_3\overline{x_4} + x_1x_4.$$

$$\begin{array}{cccc} & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{array} \rightarrow \begin{array}{cccc} & & 1 & 0 \\ 1 & & & 1 \end{array}.$$

У підсумку отримуємо мінімальну функцію:

$$F = \overline{x_1}\,\overline{x_2} + x_1x_4 + x_3\overline{x_4}. \tag{25}$$

2-й варіант: мінімізація функції з використанням правила супер-склеювання змінних за наявності неповної бінарної комбінаторної системи з повторенням 2-(n, x/b)-design.

На блоках 1–5 першої матриці блок-схеми (24) виділяємо лівий стовпчик з загальними нулями. Інші три стовпчики складуть неповну комбінаторну систему з повторенням 2-(3, 5/8)-design. Для мінімізації блоків 1–5 використовуємо правило супер-склеювання (13):

$$\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 & \\ 0 & 0 & 1 & \\ 0 & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ 0 & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 & & \\ 0 & & 1 & 0 \end{vmatrix}.$$

На блоках 6–11 першої матриці блок-схеми (24) виділяємо лівий стовпчик з загальними одиницями. Інші три стовпчики складуть неповну комбінаторну систему з повторенням 2-(3, 6/8)-design. Для блоків 6–11 використовуємо правило (23):

$$\begin{vmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & & & 1 \\ 1 & & 1 & \end{vmatrix}.$$

Склавши результати мінімізації 1–5 та 6–11 блоків в одну матрицю, отримаємо третю матрицю блок-схеми (24):

$$F=\begin{vmatrix} 0 & 0 & & \\ 0 & & 1 & 0 \\ 1 & & & 1 \\ 1 & & 1 & \end{vmatrix}.$$

Мінімізація матриці F аналогічна процедурі мінімізації першого варіанту.

Другий крок – верифікація отриманої мінімізованої функції (25) за допомогою вихідної таблиці істинності (табл. 1).

Мінімізована логічна функція (25) задовольняє вихідну таблицю істинності (табл. 1).

У табл. 2 представлені результати мінімізації функції $F(x_1,x_2,x_3,x_4)$ за допомогою ациклічного графа [13] та комбінаторним методом.

Таблиця 2

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

За допомогою ациклічного графа	Комбінаторним методом
$F=\bar{x}_1\bar{x}_2+x_1x_4+x_1x_3\bar{x}_4+\bar{x}_1x_2x_3\bar{x}_4$	$F=\bar{x}_1\bar{x}_2+x_1x_4+x_3\bar{x}_4$

З огляду табл. 2 бачимо, що комбінаторний метод дає функцію з меншим числом вхідних змінних.

Приклад 5. Мінімізувати комбінаторним методом логічну функцію [14]:

$$F(x_1,x_2,x_3,x_4)=(0,0,1,1,1,0,1,1,1,1,1,1,0,0,0,1).$$

Складемо таблицю істинності заданої 4-розрядної функції з блоків, при яких функція отримує значення одиниці, тобто для наборів: 2, 3, 4, 6, 7, 8, 9, 10, 11, 15 та проведемо мінімізацію:

$$F=\begin{vmatrix} 2 \\ 3 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 15 \end{vmatrix}\begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{vmatrix}=\begin{vmatrix} 0 & 0 & 1 & \\ 0 & 1 & & 0 \\ 1 & 0 & & \\ & 1 & 1 & 1 \end{vmatrix}=\begin{vmatrix} & 0 & 1 & \\ 0 & 1 & & 0 \\ 1 & 0 & & \\ & 1 & 1 & 1 \end{vmatrix}=\begin{vmatrix} & & 0 & 1 \\ 0 & 1 & & 0 \\ 1 & 0 & & \\ & & 1 & 1 \end{vmatrix}.$$

До блоків 8–11 (виділені червоним кольором) першої матриці застосовано правило супер-склеювання змінних, де присутня комбінаторна система 2-(2, 4)-design. Прості склеювання змінних виділені іншими кольорами. В останніх двох матрицях проведено заміщення (неповне склеювання) змінних.

У підсумку отримуємо мінімальну функцію:

$$F = x_1\overline{x_2} + \overline{x_1}x_2\overline{x_4} + \overline{x_2}x_3 + x_3x_4.$$

У табл. 3 представлені результати мінімізації функції $F(x_1,x_2,x_3,x_4)$ за допомогою паралельного розчеплення кон'юнктермів [14] та комбінаторним методом.

Таблиця 3

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

Методом паралельного розчеплення кон'юнктермів	Комбінаторним методом
$F = x_1\overline{x_2} + \overline{x_1}x_2\overline{x_4} + \overline{x_1}x_3 + x_3x_4$	$F = x_1\overline{x_2} + \overline{x_1}x_2\overline{x_4} + \overline{x_2}x_3 + x_3x_4$

З огляду табл. 3 бачимо, що обидві функції мають однакові параметри і проходять верифікацію, хоч відрізняються складом змінних у третій імпліканті. Приклад 5 демонструє меншу обчислювальну складність мінімізації булевої функції комбінаторним методом.

Приклад 6. Мінімізувати комбінаторним методом логічну функцію, що задана у канонічній формі [15]:

$$F(x_1,x_2,x_3,x_4) = (0{,}1{,}6{,}8{,}11{,}14{,}15).$$

$$F = \left|\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & 0 & 1 \\ & 0 & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} 0 & 0 & 0 & \\ & 0 & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & & 1 & 1 \end{array}\right|.$$

Результати мінімізації функції $F(x_1,x_2,x_3,x_4)$ за допомогою паралельного розчеплення кон'юнктермів [15] та комбінаторним методом представлені у табл. 4.

Таблиця 4

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

Методом паралельного розчеплення кон'юнктермів	Комбінаторним методом
$\{(000\sim),(\sim000),(\sim110),(1\sim11)\}$	$\left|\begin{array}{cccc} 0 & 0 & 0 & \\ & 0 & 0 & 0 \\ & 1 & 1 & 0 \\ 1 & & 1 & 1 \end{array}\right|$

З табл. 4 можна бачити, що результати мінімізації двох порівнюваних методів однакові. Збігається й показник мінімізації $\frac{k_\theta}{k_l} = \frac{4}{12}$, де k_θ – число простих імплікант; k_l – число вхідних змінних. Однак, обчислювальна складність мінімізації булевої функції комбінаторним методом є меншою.

Приклад 7. Мінімізувати систему 4-розрядних булевих функцій f_1, f_2, f_3 [14] комбінаторним методом:

$$\begin{cases} f_1 = 2,5,6,13,14, \\ f_2 = 5,7,13,14, \\ f_3 = 2,6,7,13,15. \end{cases}$$

Складемо таблицю істинності заданої системи 4-розрядних функції з блоків, при яких функція отримує значення одиниці (табл. 5).

Таблиця 5

Таблиця істинності системи булевих функцій f_1, f_2, f_3

	x_1	x_2	x_3	x_4	f_1	f_2	f_3
2	0	0	1	0	**1**	**0**	**1**
5	0	1	0	1	**1**	**1**	**0**
6	0	1	1	0	**1**	**0**	**1**
7	0	1	1	1	**0**	**1**	**1**
13	1	1	0	1	**1**	**1**	**1**
14	1	1	1	0	**1**	**1**	**0**
15	1	1	1	1	**0**	**0**	**1**

Є два підходи до мінімізації системи булевих функцій від *n* змінних:

1) мінімізацію здійснюють окремо для кожної функції;
2) сумісна мінімізація системи, коли метод мінімальної системи використовує загальні кон’юнктерми окремих функцій.

Усунення надлишкових кон’юнктермів в окремій функції не гарантує усунення надлишковості у самій системі. З іншого боку, сумісна мінімізація системи не завжди може бути кращою. Тому для ряду систем функцій, доводиться застосовувати обидва методи мінімізації. Сумісна мінімізація системи є більш громіздкою, порівняно з першим методом.

Для сумісної мінімізації об’єднуємо всі різні кон’юнктерми окремих функцій у функцію Y системних кон’юнктермів:

$$Y = 0010_{(1,3)} + 0101_{(1,2)} + 0110_{(1,3)} + 0111_{(2,3)} + \\ + 1101_{(1,2,3)} + 1110_{(1,2)} + 1111_{(3)}.$$

Системним кон’юнктермом називається мінтерм булевої функції з індексами, які показують, до яких функцій він належить [14]. Серед системних кон’юнктермів розрізняють тотожні елементи – такі, що мають однакові індекси, та нетотожні – що мають неоднакові індекси, але перетин яких не порожній. Наприклад, $(101)_{2,4}$ і $(111)_{2,4}$ утворюють тотожний елемент $(1\text{-}1)_{2,4}$, а $(101)_{2,4}$ і $(001)_4$ утворюють нетотожний елемент $(\text{-}01)_4$ [14].

Функцію Y представимо таблицею істинності.

Для сумісної мінімізації системи будемо застосовувати такі правила:

– склеювання змінних у системних кон’юнктермах функції *Y* здійснюється тільки для тих кон’юнктермів, які мають хоча б один загальний індекс;

– результату склеювання кон'юнктермів присвоюється множина індексів, яка є перетином вихідних множин індексів склеюваних кон'юнктермів;

– якщо кон'юнктерми не мають спільних індексів, склеювання не відбувається;

– тотожні кон'юнктерми склеюються з іншими тотожними кон'юнктермами;

– нетотожні кон'юнктерми склеюються з іншими нетотожними кон'юнктермами.

Після операції склеювання нетотожні кон'юнктерми переносяться до наступної таблиці для подальшої мінімізації, крім випадку, коли індекси результату склеювання збігаються з індексами одного з нетотожних кон'юнктермів. Поглинання одного кон'юнктерма іншим здійснюється тільки за умови збігу множин індексів двох кон'юнктермів.

$$Y=\begin{vmatrix} 0 & 0 & 1 & 0 & (1,3) \\ 0 & 1 & 1 & 0 & (1,3) \\ 0 & 1 & 0 & 1 & (1,2) \\ 0 & 1 & 1 & 1 & (2,3) \\ 1 & 1 & 0 & 1 & (1,2,3) \\ 1 & 1 & 1 & 0 & (1,2) \\ 1 & 1 & 1 & 1 & (3) \end{vmatrix}=\begin{vmatrix} 0 & & 1 & 0 & (1,3) \\ 0 & 1 & & 1 & (2) \\ & 1 & 0 & 1 & (1,2) \\ 0 & 1 & 0 & 1 & (1,2) \\ 1 & 1 & 1 & 0 & (1,2) \\ & 1 & 1 & 1 & (3) \\ 0 & 1 & 1 & 1 & (2,3) \\ 1 & 1 & 0 & 1 & (1,2,3) \\ 1 & 1 & & 1 & (3) \end{vmatrix}=\begin{vmatrix} 0 & & 1 & 0 & (1,3) \\ 0 & 1 & & 1 & (2) \\ & 1 & 0 & 1 & (1,2) \\ 1 & 1 & 1 & 0 & (1,2) \\ & 1 & 1 & 1 & (3) \\ 0 & 1 & 1 & 1 & (2,3) \\ 1 & 1 & 0 & 1 & (1,2,3) \\ 1 & 1 & & 1 & (3) \end{vmatrix}.$$

Перетворення першої матриці:

– склеювання тотожних кон'юнктермів $0010_{(1,3)}$ та $0110_{(1,3)}$ (виділені червоним кольором); результат склеювання – $0\sim10_{(1,3)}$ переноситься до другої матриці; після склеювання кон'юнктерми $0010_{(1,3)}$ та $0110_{(1,3)}$ в інших операціях склеювання першої матриці участі не беруть;

– склеювання нетотожних кон'юнктермів $0101_{(1,2)}$ та $0111_{(2,3)}$; результат склеювання – $01\sim1_{(2)}$ переноситься до другої матриці; оскільки кон'юнктерми $0101_{(1,2)}$ та $0111_{(2,3)}$ є нетотожними, вони можуть брати участь в інших операціях склеювання першої матриці; кон'юнктерми $0101_{(1,2)}$ та $0111_{(2,3)}$ переносяться до другої матриці для подальшої мінімізації;

– склеювання нетотожних кон'юнктермів $0101_{(1,2)}$ та $1101_{(1,2,3)}$; результат склеювання – $\sim101_{(1,2)}$ переноситься до другої матриці; оскільки кон'юнктерми $0101_{(1,2)}$ та $1101_{(1,2,3)}$ є нетотожними, вони можуть брати участь в інших операціях склеювання першої матриці; кон'юнктерми $0101_{(1,2)}$ та $1101_{(1,2,3)}$ переносяться до другої матриці для подальшої мінімізації;

– нетотожний кон'юнктерм $1110_{(1,2)}$ не склеюється з жодним кон'юнктермом, переноситься до другої матриці для подальшої мінімізації;

– склеювання нетотожних кон’юнктермів $0111_{(2,3)}$ та $1111_{(3)}$; результат склеювання – $\sim111_{(3)}$ переноситься до другої матриці; оскільки результат склеювання – $\sim111_{(3)}$ та кон’юнктерм – $1111_{(3)}$ мають однакові індекси, кон’юнктерм – $1111_{(3)}$ не переноситься до другої матриці; індекси кон’юнктерма $0111_{(2,3)}$ не збігаються з індексами результату склеювання, тому цей кон’юнктерм переноситься до другої матриці для подальшої мінімізації;

– склеювання нетотожних кон’юнктермів $1101_{(1,2,3)}$ та $1111_{(3)}$; результат склеювання – $11\sim1_{(3)}$ переноситься до другої матриці; оскільки результат склеювання – $11\sim1_{(3)}$ та кон’юнктерм – $1111_{(3)}$ мають однакові індекси, кон’юнктерм – $1111_{(3)}$ не переноситься до другої матриці; індекси кон’юнктерма $1101_{(1,2,3)}$ не збігаються з індексами результату склеювання, тому цей кон’юнктерм переноситься до другої матриці для подальшої мінімізації.

Перетворення другої матриці: поглинання тотожних кон’юнктермів $\sim101_{(1,2)}$ та $0101_{(1,2)}$ (виділені синім кольором); результат поглинання – $\sim101_{(1,2)}$ переноситься до третьої матриці.

Третя матриця представляє тупикову ДНФ функції *Y*. Далі завдання пошуку мінімальної ДНФ функції *Y* вирішується на підставі таблиці покриття Рицара Б. [14] (рис. 1), у якій необхідно забрати усі зайві прості імпліканти.

	$01\sim1_{(2)}$		$01\sim1_{(2)}$	$11\sim1_{(3)}$		$11\sim1_{(3)}$
$0\sim10_{(1,3)}$		$0\sim10_{(1,3)}$				
	$\sim101_{(1,2)}$		$\sim111_{(3)}$	$\sim101_{(1,2)}$		$\sim111_{(3)}$
$0010_{(1,3)}$	$0101_{(1,2)}$	$0110_{(1,3)}$	$0111_{(2,3)}$	$1101_{(1,2,3)}$	$1110_{(1,2)}$	$1111_{(3)}$

Рис. 1. Таблиця покриття функції $F(x_1,x_2,x_3,x_4)$

На рис. 1 кольором виділені елементи мінімального покриття, що мінімізують функцію Y методом сумісної мінімізації системи:

$$\begin{aligned} Y = {} & \overline{x_1}x_3\overline{x_4}(1,3) + x_2\overline{x_3}x_4(1,2) + \overline{x_1}x_2x_3x_4(2,3) + \\ & + x_1x_2x_4(3) + x_1x_2x_3\overline{x_4}(1,2). \end{aligned} \tag{26}$$

Результат мінімізації функції системних кон’юнктермів (26) комбінаторним методом збігається з результатом мінімізації методом паралельного розчеплення кон’юнктермів [14].

Оскільки для мінімізації функції *Y* комбінаторним методом у прикладі 6 операція склеювання застосовувалась для тотожних кон’юнктермів і не застосовувалась між тотожними та нетотожними кон’юнктермами, це дало зменшення кількості зайвих простих імплікант та розміру таблиці покриття (рис. 1).

Після розподілу системних кон’юнктермів функції (26) отримуємо мінімізовану систему булевих функцій:

$$
\begin{cases}
f_1 = \overline{x_1} x_3 \overline{x_4} + x_2 \overline{x_3} x_4 + x_1 x_2 x_3 \overline{x_4}, \\
f_2 = x_2 \overline{x_3} x_4 + \overline{x_1} x_2 x_3 x_4 + x_1 x_2 x_3 \overline{x_4}, \\
f_3 = \overline{x_1} x_3 \overline{x_4} + \overline{x_1} x_2 x_3 x_4 + x_1 x_2 x_3.
\end{cases}
$$

Приклад 8. Мінімізувати комбінаторним методом логічну функцію [16]:

$$F(x_1, x_2, x_3, x_4) = (1,1,1,1,1,1,0,0,1,1,1,1,0,0,0,0).$$

Складемо таблицю істинності заданої 4-розрядної функції з блоків при яких функція отримує значення одиниці, тобто для наборів: 0, 1, 2, 3, 4, 5, 8, 9, 10, 11 та проведемо мінімізацію:

$$
\left|\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
2 & 0 & 0 & 1 & 0 \\
3 & 0 & 0 & 1 & 1 \\
4 & 0 & 1 & 0 & 0 \\
5 & 0 & 1 & 0 & 1 \\
8 & 1 & 0 & 0 & 0 \\
9 & 1 & 0 & 0 & 1 \\
10 & 1 & 0 & 1 & 0 \\
11 & 1 & 0 & 1 & 1
\end{array}\right| = \left|\begin{array}{cccc}
 & 0 & & \\
0 & 1 & 0 &
\end{array}\right| = \left|\begin{array}{cccc}
 & 0 & & \\
0 & & 0 &
\end{array}\right|.
$$

До блоків 0–3 та 8–11(виділені червоним кольором) першої матриці застосовано правило супер-склеювання змінних, де присутня комбінаторна система 2-(3, 8)-design. Просте склеювання змінних виділено чорним кольорам. В останній матриці проведено заміщення (неповне склеювання) змінних.

У підсумку отримуємо мінімальну функцію:

$$F = \overline{x_1}\,\overline{x_3} + \overline{x_2}.$$

Результат мінімізації комбінаторним методом збігається з результатом мінімізації, отриманим за допомогою методу самопонижуючих циклів [16]. Метод самопонижуючих циклів для мінімізації заданої функції використовує чотири понижуючих цикли і таблицю покриття для видалення зайвих імплікант. Комбінаторний метод задану функцію мінімізує за три образних перетворення. Оскільки метод самопонижуючих циклів для мінімізації булевої функції використовує повну комбінаторну систему з повторенням 2-(n, b)-design [16], але не використовує неповну комбінаторну систему з повторенням 2-(n, x/b)-design, цей метод можна віднести до часткового випадку мінімізації комбінаторним методом.

5.4. Мінімізація 5-розрядних булевих функцій

Приклад 9. Мінімізувати логічну функцію $F(x_1, x_2, x_3, x_4, x_5)$ комбінаторним методом, яка задана наступною таблицею істинності (табл. 6) [17].

Таблиця 6

Таблиця істинності логічної функції $F(x_1,x_2,x_3,x_4,x_5)$

№ з/п	x_1	x_2	x_3	x_4	x_5	F	№ з/п	x_1	x_2	x_3	x_4	x_5	F
0	0	0	0	0	0	**0**	16	1	0	0	0	0	**1**
1	0	0	0	0	1	**1**	17	1	0	0	0	1	**1**
2	0	0	0	1	0	–	18	1	0	0	1	0	**1**
3	0	0	0	1	1	–	19	1	0	0	1	1	**0**
4	0	0	1	0	0	**1**	20	1	0	1	0	0	–
5	0	0	1	0	1	**1**	21	1	0	1	0	1	**0**
6	0	0	1	1	0	**0**	22	1	0	1	1	0	**1**
7	0	0	1	1	1	**1**	23	1	0	1	1	1	**0**
8	0	1	0	0	0	**0**	24	1	1	0	0	0	**0**
9	0	1	0	0	1	**1**	25	1	1	0	0	1	**0**
10	0	1	0	1	0	**0**	26	1	1	0	1	0	–
11	0	1	0	1	1	**1**	27	1	1	0	1	1	**0**
12	0	1	1	0	0	**1**	28	1	1	1	0	0	**1**
13	0	1	1	0	1	**1**	29	1	1	1	0	1	**0**
14	0	1	1	1	0	–	30	1	1	1	1	0	–
15	0	1	1	1	1	–	31	1	1	1	1	1	**1**

Використовуючи табл. 6 складемо ДДНФ заданої 5-розрядної функції з блоків при яких функція отримує значення одиниці, тобто для наборів 1, 4, 5, 7, 9, 11, 12, 13, 16, 17, 18, 22, 28, 31:

$$\begin{aligned}
F(x_1,x_2,x_3,x_4,x_5) = {} & \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4x_5 + \bar{x}_1\bar{x}_2x_3\bar{x}_4\bar{x}_5 + \\
+ & \bar{x}_1\bar{x}_2x_3\bar{x}_4x_5 + \bar{x}_1\bar{x}_2x_3x_4x_5 + \bar{x}_1x_2\bar{x}_3\bar{x}_4x_5 + \\
+ & \bar{x}_1x_2\bar{x}_3x_4x_5 + \bar{x}_1x_2x_3\bar{x}_4\bar{x}_5 + \bar{x}_1x_2x_3\bar{x}_4x_5 + \\
+ & x_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5 + x_1\bar{x}_2\bar{x}_3\bar{x}_4x_5 + x_1\bar{x}_2\bar{x}_3x_4\bar{x}_5 + \\
+ & x_1\bar{x}_2x_3x_4\bar{x}_5 + x_1x_2x_3\bar{x}_4\bar{x}_5 + x_1x_2x_3x_4x_5.
\end{aligned} \qquad (27)$$

Нагадаємо, значення «–» функції F означає довільний стан, який вказує на те, що такого набору вхідних змінних не очікується і значення функції може бути довільним – нулем або одиницею у процесі мінімізації.

Доповнимо визначення функції $F(x_1,x_2,x_3,x_4,x_5)$ заміною значення «–» функції на одиницю. Після цієї заміни таблиця істинності (табл. 6) набуває такого вигляду (табл. 7).

Використовуючи табл. 7, треба виписати ДДНФ 5-розрядної функції з блоків, при яких функція отримує значення одиниці, тобто для наборів 1, 2, 3, 4, 5, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 20, 22, 26, 28, 30, 31:

$$\begin{aligned}
F(x_1,x_2,x_3,x_4,x_5) = {} & \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4x_5 + \bar{x}_1\bar{x}_2\bar{x}_3x_4\bar{x}_5 + \\
+ & \bar{x}_1\bar{x}_2\bar{x}_3x_4x_5 + \bar{x}_1\bar{x}_2x_3\bar{x}_4\bar{x}_5 + \bar{x}_1\bar{x}_2x_3\bar{x}_4x_5 + \\
+ & \bar{x}_1\bar{x}_2x_3x_4x_5 + \bar{x}_1x_2\bar{x}_3\bar{x}_4x_5 + \bar{x}_1x_2\bar{x}_3x_4x_5 + \\
+ & \bar{x}_1x_2x_3\bar{x}_4\bar{x}_5 + \bar{x}_1x_2x_3\bar{x}_4x_5 + \bar{x}_1x_2x_3x_4\bar{x}_5 + \\
+ & \bar{x}_1x_2x_3x_4x_5 + x_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5 + x_1\bar{x}_2\bar{x}_3\bar{x}_4x_5 + \\
+ & x_1\bar{x}_2\bar{x}_3x_4\bar{x}_5 + x_1\bar{x}_2x_3\bar{x}_4\bar{x}_5 + x_1\bar{x}_2x_3x_4\bar{x}_5 + \\
+ & x_1x_2\bar{x}_3x_4\bar{x}_5 + x_1x_2x_3\bar{x}_4\bar{x}_5 + x_1x_2x_3x_4\bar{x}_5 + x_1x_2x_3x_4x_5.
\end{aligned} \qquad (28)$$

Таблиця 7

Таблиця істинності логічної функції $F(x_1,x_2,x_3,x_4,x_5)$ після заміни значення «–» функції на одиницю

№ з/п	x_1	x_2	x_3	x_4	x_5	F	№ з/п	x_1	x_2	x_3	x_4	x_5	F
0	0	0	0	0	0	**0**	16	1	0	0	0	0	**1**
1	0	0	0	0	1	**1**	17	1	0	0	0	1	**1**
2	0	0	0	1	0	**1**	18	1	0	0	1	0	**1**
3	0	0	0	1	1	**1**	19	1	0	0	1	1	**0**
4	0	0	1	0	0	**1**	20	1	0	1	0	0	**1**
5	0	0	1	0	1	**1**	21	1	0	1	0	1	**0**
6	0	0	1	1	0	**0**	22	1	0	1	1	0	**1**
7	0	0	1	1	1	**1**	23	1	0	1	1	1	**0**
8	0	1	0	0	0	**0**	24	1	1	0	0	0	**0**
9	0	1	0	0	1	**1**	25	1	1	0	0	1	**0**
10	0	1	0	1	0	**0**	26	1	1	0	1	0	**1**
11	0	1	0	1	1	**1**	27	1	1	0	1	1	**0**
12	0	1	1	0	0	**1**	28	1	1	1	0	0	**1**
13	0	1	1	0	1	**1**	29	1	1	1	0	1	**0**
14	0	1	1	1	0	**1**	30	1	1	1	1	0	**1**
15	0	1	1	1	1	**1**	31	1	1	1	1	1	**1**

На **першому кроці** здійснюють склеювання конституант і заміщення змінних. Використовуючи неповну бінарну комбінаторну систему з повторенням 2-(*n*, *x/b*)-design для 5-розрядної логічної функції, на блоках 1–12 (табл. 8) виділяють лівий стовпчик з загальними нулями. Інші чотири стовпчики складуть неповну комбінаторну систему з повторенням 2-(4, 12/16)-design.

Таблиця 8

Таблиця істинності досконалої диз'юнктивної нормальної форми $F(x_1,x_2,x_3,x_4,x_5)$, блоки якої отримують значення одиниці

№ з/п	x_1	x_2	x_3	x_4	x_5	F	№ з/п	x_1	x_2	x_3	x_4	x_5	F
1	0	0	0	0	1	**1**	12	0	1	1	1	1	**1**
2	0	0	0	1	0	**1**	13	1	0	0	0	0	**1**
3	0	0	0	1	1	**1**	14	1	0	0	0	1	**1**
4	0	0	1	0	0	**1**	15	1	0	0	1	0	**1**
5	0	0	1	0	1	**1**	16	1	0	1	0	0	**1**
6	0	0	1	1	1	**1**	17	1	0	1	1	0	**1**
7	0	1	0	0	1	**1**	18	1	1	0	1	0	**1**
8	0	1	0	1	1	**1**	19	1	1	1	0	0	**1**
9	0	1	1	0	0	**1**	20	1	1	1	1	0	**1**
10	0	1	1	0	1	**1**	21	1	1	1	1	1	**1**
11	0	1	1	1	0	**1**							

Процес мінімізації блоків 1–12 (табл. 8) можливий двома варіантами.

1-й варіант: склеювання, заміщення і склеювання змінних.

$$\left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 0 & 1 \\
\bar{x} & \bar{y} & 0 & 1 & 0 \\
\bar{x} & \bar{y} & 0 & 1 & 1 \\
\bar{x} & \bar{y} & 1 & 0 & 0 \\
\bar{x} & \bar{y} & 1 & 0 & 1 \\
\bar{x} & \bar{y} & 1 & 1 & 1 \\
\bar{x} & y & 0 & 0 & 1 \\
\bar{x} & y & 0 & 1 & 1 \\
\bar{x} & y & 1 & 0 & 0 \\
\bar{x} & y & 1 & 0 & 1 \\
\bar{x} & y & 1 & 1 & 0 \\
\bar{x} & y & 1 & 1 & 1
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & 0 \\
\bar{x} & & 0 & 0 & 1 \\
\bar{x} & & 0 & 1 & 1 \\
\bar{x} & & 1 & 0 & 0 \\
\bar{x} & & 1 & 0 & 1 \\
\bar{x} & y & 1 & 1 & 0 \\
\bar{x} & & 1 & 1 & 1
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & 0 \\
\bar{x} & & 0 & & 1 \\
\bar{x} & & 1 & 0 & \\
\bar{x} & y & 1 & 1 & 0 \\
\bar{x} & & 1 & 1 & 1
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 0 & & 1 \\
\bar{x} & & 1 & 0 & \\
\bar{x} & y & 1 & 1 & \\
\bar{x} & & 1 & 1 & 1
\end{array}\right| =$$

$$= \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 0 & & 1 \\
\bar{x} & & 1 & 0 & \\
\bar{x} & y & 1 & & \\
\bar{x} & & 1 & & 1
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & y & 1 & & \\
\bar{x} & & & & 1
\end{array}\right|.$$

2-й варіант: склеювання, заміщення, склеювання, заміщення, склеювання і заміщення змінних.

$$\left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 0 & 1 \\
\bar{x} & \bar{y} & 0 & 1 & 0 \\
\bar{x} & \bar{y} & 0 & 1 & 1 \\
\bar{x} & \bar{y} & 1 & 0 & 0 \\
\bar{x} & \bar{y} & 1 & 0 & 1 \\
\bar{x} & \bar{y} & 1 & 1 & 1 \\
\bar{x} & y & 0 & 0 & 1 \\
\bar{x} & y & 0 & 1 & 1 \\
\bar{x} & y & 1 & 0 & 0 \\
\bar{x} & y & 1 & 0 & 1 \\
\bar{x} & y & 1 & 1 & 0 \\
\bar{x} & y & 1 & 1 & 1
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 0 & 1 \\
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & \bar{y} & 1 & 0 & \\
\bar{x} & \bar{y} & 1 & 1 & 1 \\
\bar{x} & y & 0 & & 1 \\
\bar{x} & y & 1 & 0 & \\
\bar{x} & y & 1 & 1 &
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 0 & 1 \\
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & \bar{y} & 1 & 1 & 1 \\
\bar{x} & y & 0 & & 1 \\
\bar{x} & y & 1 & 1 &
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & & 1 \\
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & & 1 & 1 & 1 \\
\bar{x} & y & 0 & & 1 \\
\bar{x} & y & 1 & 1 &
\end{array}\right| =$$

$$= \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & & 1 & 1 & 1 \\
\bar{x} & & 0 & & 1 \\
\bar{x} & y & 1 & 1 &
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & & 1 & & 1 \\
\bar{x} & & 0 & & 1 \\
\bar{x} & y & 1 & 1 &
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & & & & 1 \\
\bar{x} & y & 1 & 1 &
\end{array}\right| = \left|\begin{array}{ccccc}
\bar{x} & \bar{y} & 0 & 1 & \\
\bar{x} & & 1 & 0 & \\
\bar{x} & & & & 1 \\
\bar{x} & y & 1 & &
\end{array}\right|.$$

На блоках 13–21 (табл. 8) виділяють лівий стовпчик із загальними одиницями. Інші чотири стовпчики складуть неповну комбінаторну систему з повторенням 2-(4, 9/16)-design.

Процес мінімізації блоків 13–21 (табл. 8) можливий двома варіантами.

1-й варіант: склеювання, склеювання і заміщення змінних.

$$\begin{vmatrix} x & \bar{y} & 0 & 0 & 0 \\ x & \bar{y} & 0 & 0 & 1 \\ x & \bar{y} & 0 & 1 & 0 \\ x & \bar{y} & 1 & 0 & 0 \\ x & \bar{y} & 1 & 1 & 0 \\ x & y & 0 & 1 & 0 \\ x & y & 1 & 0 & 0 \\ x & y & 1 & 1 & 0 \\ x & y & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & 0 & 0 & \\ x & & 0 & 1 & 0 \\ x & & 1 & 0 & 0 \\ x & & 1 & 1 & 0 \\ x & y & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & 0 & 0 & \\ x & & 1 & 0 & 0 \\ x & & & 1 & 0 \\ x & y & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & 0 & 0 & \\ x & & 1 & & 0 \\ x & & & 1 & 0 \\ x & y & 1 & 1 & \end{vmatrix}.$$

2-й варіант: склеювання, склеювання і заміщення змінних.

$$\begin{vmatrix} x & \bar{y} & 0 & 0 & 0 \\ x & \bar{y} & 0 & 0 & 1 \\ x & \bar{y} & 0 & 1 & 0 \\ x & \bar{y} & 1 & 0 & 0 \\ x & \bar{y} & 1 & 1 & 0 \\ x & y & 0 & 1 & 0 \\ x & y & 1 & 0 & 0 \\ x & y & 1 & 1 & 0 \\ x & y & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & 0 & 0 & \\ x & \bar{y} & 0 & 1 & 0 \\ x & \bar{y} & 1 & & 0 \\ x & y & 0 & 1 & 0 \\ x & y & 1 & & 0 \\ x & y & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & 0 & 0 & \\ x & & 0 & 1 & 0 \\ x & & 1 & & 0 \\ x & y & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x & \bar{y} & 0 & 0 & \\ x & & & 1 & 0 \\ x & & 1 & & 0 \\ x & y & 1 & 1 & \end{vmatrix}.$$

На **другому кроці** виконується заміщення і узагальнене склеювання змінних.

Для подальшої мінімізації ДДНФ $F(x_1,x_2,x_3,x_4,x_5)$ об’єднаємо результати мінімізації 1–12 та 13–21 стовпчиків табл. 8 в одну матрицю:

$$\begin{vmatrix} \bar{x} & \bar{y} & 0 & 1 & \\ \bar{x} & & 1 & 0 & \\ \bar{x} & & & & 1 \\ \bar{x} & y & 1 & & \\ x & \bar{y} & 0 & 0 & \\ x & & & 1 & 0 \\ x & & 1 & & 0 \\ x & y & 1 & 1 & \end{vmatrix} = \begin{vmatrix} \bar{x} & \bar{y} & 0 & 1 & \\ \bar{x} & & 1 & 0 & \\ \bar{x} & & & & 1 \\ \bar{x} & y & 1 & & \\ x & \bar{y} & 0 & 0 & \\ x & & & 1 & 0 \\ x & & 1 & & 0 \\ & y & 1 & 1 & \end{vmatrix} = \begin{vmatrix} \bar{x} & \bar{y} & 0 & 1 & \\ \bar{x} & & 1 & 0 & \\ \bar{x} & & & & 1 \\ x & \bar{y} & 0 & 0 & \\ x & & & 1 & 0 \\ x & & 1 & & 0 \\ & y & 1 & 1 & \end{vmatrix} = \begin{vmatrix} \bar{x} & \bar{y} & 0 & 1 & \\ \bar{x} & & 1 & 0 & \\ \bar{x} & & & & 1 \\ x & \bar{y} & 0 & 0 & \\ x & & & 1 & 0 \\ & & 1 & 0 & 0 \\ x & & 1 & & 0 \\ & y & 1 & 1 & \end{vmatrix} = \begin{vmatrix} \bar{x} & \bar{y} & 0 & 1 & \\ \bar{x} & & & & 1 \\ x & \bar{y} & 0 & 0 & \\ x & & & 1 & 0 \\ & & 1 & 0 & 0 \\ & y & 1 & 1 & \end{vmatrix}.$$

У першій об'єднаній матриці проведено заміщення змінної, у 2 – 4 об'єднаних матрицях проведено узагальнене склеювання змінних.

Алгебричні перетворення другої об'єднаної матриці, результат якого записаний у третю об'єднану матрицю, визначають узагальнене склеювання змінних для 2, 4 та 8 блоків.

$$\overline{x}_1 x_3 \overline{x}_4 + \overline{x}_1 x_2 x_3 + x_2 x_3 x_4 = \overline{x}_1 x_3 \overline{x}_4 +$$
$$+ x_2 x_3 x_4 + \overline{x}_1 x_2 x_3 = \overline{x}_1 x_3 \overline{x}_4 + x_2 x_3 x_4.$$

Алгебричні перетворення третьої об'єднаної матриці, результат якого записаний у четверту об'єднану матрицю, визначають узагальнене склеювання змінних для 2, та 7 блоків:

$$\overline{x}_1 x_3 \overline{x}_4 + x_1 x_3 \overline{x}_5 = \overline{x}_1 x_3 \overline{x}_4 + x_1 x_3 \overline{x}_5 +$$
$$+ x_3 x_3 \overline{x}_4 \overline{x}_5 = \overline{x}_1 x_3 \overline{x}_4 + x_1 x_3 \overline{x}_5 + x_3 \overline{x}_4 \overline{x}_5.$$

Алгебричні перетворення четвертої об'єднаної матриці, результат якого записаний у п'яту об'єднану матрицю, визначають як:

– узагальнене склеювання змінних для 5, 6 та 7 блоків:

$$x_3 \overline{x}_4 \overline{x}_5 + x_1 x_4 \overline{x}_5 = x_3 \overline{x}_4 \overline{x}_5 + x_1 x_4 \overline{x}_5 + x_1 x_3 \overline{x}_5 \overline{x}_5 =$$
$$= x_3 \overline{x}_4 \overline{x}_5 + x_1 x_4 \overline{x}_5 + x_1 x_3 \overline{x}_5 = x_3 \overline{x}_4 \overline{x}_5 + x_1 x_4 \overline{x}_5.$$

– узагальнене склеювання змінних для 2, 6 та 7 блоків:

$$\overline{x}_1 x_5 + x_3 \overline{x}_4 \overline{x}_5 + \overline{x}_1 x_3 \overline{x}_4 = \overline{x}_1 x_5 + x_3 \overline{x}_4 \overline{x}_5,$$

Спроби подальшого застосування операцій алгебричного перетворення не дають результату, що приводить до тупикової ДНФ функції $F(x_1, x_2, x_3, x_4, x_5)$, яка представлена у табл. 8.

$$F(x_1, x_2, x_3, x_4, x_5) = \overline{x}_1 x_5 + x_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 +$$
$$+ x_1 x_4 \overline{x}_5 + \overline{x}_1 \overline{x}_2 \overline{x}_3 x_4 + x_2 x_3 x_4 + x_3 \overline{x}_4 \overline{x}_5.$$

Третій крок передбачає перевірку кожної простої імпліканти у ДДНФ на надлишковість з метою її видалення та веріфікація отриманої функції за допомогою таблиці істинності (табл. 8).

Далі завдання пошуку мінімальної ДНФ вирішується на підставі таблиці покриття (табл. 9). У загальному випадку для одержання мінімальної ДНФ необхідно забрати з тупикової ДНФ усі зайві прості імпліканти.

У стовпцях табл. 9 знаходяться прості імпліканти скороченої ДНФ функції, представленої п'ятою об'єднаною матрицею. Рядки табл. 9 представляють конституанти одиниці ДДНФ функції, що представлена табл. 7.

Таблиця 9

Таблиця покриття функції $F(x_1,x_2,x_3,x_4,x_5)$

Конституанти	$\overline{x_1}x_5$	$x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}$	$x_1x_4\overline{x_5}$	$\overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4$	$x_2x_3x_4$	$x_3\overline{x_4}\,\overline{x_5}$
00001	●	–	–	–	–	–
00010	–	–	–	●	–	–
00011	●	–	–	–	–	–
00100	–	–	–	–	–	●
00101	●	–	–	–	–	–
00111	●	–	–	–	–	–
01001	●	–	–	–	–	–
01011	●	–	–	–	–	–
01100	–	–	–	–	–	●
01101	●	–	–	–	–	–
01110	–	–	–	–	●	–
01111	●	–	–	–	●	–
10000	–	●	–	–	–	–
10001	–	●	–	–	–	–
10010	–	–	●	–	–	–
10100	–	–	–	–	–	●
10110	–	–	●	–	–	–
11010	–	–	●	–	–	–
11100	–	–	–	–	–	●
11110	–	–	●	–	●	–
11111	–	–	–	–	●	–

Проста імпліканта поглинає певну конституанту одиниці тоді, коли є її власною частиною. Відповідна клітинка табл. 9 на перетині стовпця (з розглянутою простою імплікантою) і рядка (з конституантою одиниці) позначена кружком ● чорного кольору.

З огляду табл. 9 можна бачити, що зайві імпліканти відсутні, і, отже, табл. 9 представляє мінімальну ДНФ функції (28), що представлена табл. 7.

$$F(x_1,x_2,x_3,x_4,x_5)=\overline{x_1}x_5+x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}+ \\ +x_1x_4\overline{x_5}+\overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4+x_2x_3x_4+x_3\overline{x_4}\,\overline{x_5}. \qquad (29)$$

Таблиця істинності (табл. 7) допомагає зручніше здійснювати мінімізацію. Слід зауважити, що вихідна логічна функція (28) представлена таблицею істинності (табл. 6), у якій присутні набори непередбачуваних змінних. Значення функції F для таких наборів позначається «–» і означає її довільний стан.

У зв’язку з цим пошук мінімальної ДНФ функції, що представлена вихідною таблицею істинності (табл. 6) вирішується за допомогою таблиці покриття (табл. 9), видаленням з її рядків наборів непередбачуваних змінних. Після цього таблиця набуває такого вигляду (табл. 10).

З огляду табл. 10 бачимо, що зайвою є імпліканта $\overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4$, яку видаляють з виразу функції (29):

$$F(x_1,x_2,x_3,x_4,x_5)=\overline{x_1}x_5+x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}+ \\ +x_1x_4\overline{x_5}+x_2x_3x_4+x_3\overline{x_4}\,\overline{x_5}. \qquad (30)$$

Таблиця 10

Таблиця покриття функції $F(x_1,x_2,x_3,x_4,x_5)$ з видаленими наборами непередбачуваних змінних

Конституанти	$\overline{x_1}x_5$	$x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}$	$x_1x_4\overline{x_5}$	$\overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4$	$x_2x_3x_4$	$x_3\overline{x_4}\,\overline{x_5}$
00001	●	–	–	–	–	–
00100	–	–	–	–	–	●
00101	●	–	–	–	–	–
00111	●	–	–	–	–	–
01001	●	–	–	–	–	–
01011	●	–	–	–	–	–
01100	–	–	–	–	–	●
01101	●	–	–	–	–	–
10000	–	●	–	–	–	–
10001	–	●	–	–	–	–
10010	–	–	●	–	–	–
10110	–	–	●	–	–	–
11100	–	–	–	–	–	●
11111	–	–	–	–	●	–

Вираз (30) представляє тупикову і мінімальну ДНФ вихідної функції (27), що представлена у табл. 6.

У табл. 11 подані результати мінімізації функції $F(x_1,x_2,x_3,x_4,x_5)$ методом «симетричних карт» [17] та комбінаторним методом.

Таблиця 11

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

Методом «симетричних карт»

$$F(x_1,x_2,x_3,x_4,x_5)=\overline{x_1}x_5+x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}+x_1\overline{x_2}\,\overline{x_5}+x_2x_3x_4+x_3\overline{x_4}\,\overline{x_5}.$$

$$F(x_1,x_2,x_3,x_4,x_5)=\overline{x_1}x_5+\overline{x_2}\,\overline{x_3}\,\overline{x_4}x_5+x_1\overline{x_2}\,\overline{x_5}+x_2x_3x_4+x_3\overline{x_4}\,\overline{x_5}.$$

Комбінаторним методом

$$F(x_1,x_2,x_3,x_4,x_5)=\overline{x_1}x_5+x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}+x_1x_4\overline{x_5}+x_2x_3x_4+x_3\overline{x_4}\,\overline{x_5}.$$

Основну відмінність мінімальних функцій (табл. 11) демонструє третя імпліканта. Для функції мінімізації методом «симетричних карт» імпліканта $x_1\overline{x_2}\,\overline{x_5}$ вимагає здійснення двох інвертувань для підтримки своєї функціональності, тоді як для функції мінімізованої комбінаторним методом (імпліканта – $x_1x_4\overline{x_5}$) – одного. Для апаратної реалізації функції (30) потрібно буде у даному випадку на один інвертор менше, якщо обрати, наприклад, технологію К-МОН (комплементарна структура метал-окис-напівпровідник).

Мінімізована логічна функція (30) задовольняє задану таблицю істинності (табл. 6).

5.5. Мінімізація 6-розрядних булевих функцій

Приклад 10. Мінімізувати комбінаторним методом логічну функцію [18]:

$F(x_1, x_2, x_3, x_4, x_5, x_6) = (0,1,0,1,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,$
$0,0,1,1,0,0,1,1,0,0,1,1,1,1,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1,1,1,1).$

Складемо таблицю істинності заданої 6-розрядної функції з блоків при яких функція отримує значення одиниці, тобто для наборів: 1, 3, 10, 11, 12, 13, 26, 27, 28, 29, 32, 33, 36, 37, 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 54, 55, 56, 57, 58, 59, 60, 61, 62. 63.

$$\left|\begin{array}{c|cccccc} 1 & 0&0&0&0&0&1\\ 3 & 0&0&0&0&1&1\\ 10 & 0&0&1&0&1&0\\ 11 & 0&0&1&0&1&1\\ 12 & 0&0&1&1&0&0\\ 13 & 0&0&1&1&0&1\\ 26 & 0&1&1&0&1&0\\ 27 & 0&1&1&0&1&1\\ 28 & 0&1&1&1&0&0\\ 29 & 0&1&1&1&0&1\\ 32 & 1&0&0&0&0&0\\ 33 & 1&0&0&0&0&1\\ 36 & 1&0&0&1&0&0\\ 37 & 1&0&0&1&0&1\\ 40 & 1&0&1&0&0&0\\ 41 & 1&0&1&0&0&1\\ 42 & 1&0&1&0&1&0\\ 43 & 1&0&1&0&1&1\\ 44 & 1&0&1&1&0&0\\ 45 & 1&0&1&1&0&1\\ 46 & 1&0&1&1&1&0\\ 47 & 1&0&1&1&1&1\\ 50 & 1&1&0&0&1&0\\ 51 & 1&1&0&0&1&1\\ 54 & 1&1&0&1&1&0\\ 55 & 1&1&0&1&1&1\\ 56 & 1&1&1&0&0&0\\ 57 & 1&1&1&0&0&1\\ 58 & 1&1&1&0&1&0\\ 59 & 1&1&1&0&1&1\\ 60 & 1&1&1&1&0&0\\ 61 & 1&1&1&1&0&1\\ 62 & 1&1&1&1&1&0\\ 63 & 1&1&1&1&1&1 \end{array}\right| = \left|\begin{array}{cccccc} 0&0&0&0&&1\\ 0&0&1&0&1&\\ 0&0&1&1&0&\\ 0&1&1&0&1&\\ 0&1&1&1&0&\\ 1&0&0&0&0&\\ 1&0&0&1&0&\\ 1&0&1&&&\\ 1&1&0&0&1&\\ 1&1&0&1&1&\\ 1&1&1&&& \end{array}\right| = \left|\begin{array}{cccccc} 0&0&0&0&&1\\ 0&&1&0&1&\\ 0&&1&1&0&\\ 1&0&0&&0&\\ 1&&1&&&\\ 1&1&0&&1& \end{array}\right| = \left|\begin{array}{cccccc} 0&0&0&0&&1\\ &&1&0&1&\\ &&1&1&0&\\ 1&0&&&0&\\ 1&&1&&&\\ 1&1&&&1& \end{array}\right| \quad (31)$$

До блоків 40–47 та 56–63 першої матриці блок-схеми (31) застосовано правило супер-склеювання змінних, де системи 2-(3, 8)-design виділені червоним кольором. Для решти блоків першої матриці здійснено просте склеювання змінних зі записом результату образних перетворень у другу матрицю. У другій матриці блок-схеми (31) проведено просте склеювання змінних, а у третій – заміщення змінних.

У підсумку отримуємо мінімальну функцію:

$$F = \overline{x}_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 x_6 + x_3 \overline{x}_4 x_5 + x_3 x_4 \overline{x}_5 + x_1 \overline{x}_2 \overline{x}_5 + x_1 x_3 + x_1 x_2 x_5. \qquad (32)$$

Результат мінімізації комбінаторним методом (32) збігається з результатом мінімізації, отриманим за допомогою тривимірної карти Карно [18]. Приклад 10 демонструє меншу обчислювальну складність мінімізації булевої функції комбінаторним методом.

5.6. Мінімізація 8-розрядних булевих функцій

Приклад 11. Мінімізувати логічну функцію $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ [19] комбінаторним методом, яка задана наступною таблицею істинності (33).

Для кожних восьми блоків першої матриці блок-схеми (33) застововано правило супер-склеювання змінних, оскільки у кожній «вісьмірці» блоків наявна повна комбінаторна система з повторенням 2-(3,8)-design (виділена червоним кольором). У другій матриці блок-схеми (33) застосовано правило суперсклеювання (присутня 2-(2,4)-design, блоки матриці виділені червоним кольором) та правило (10) (блоки матриці виділені синім кольором). Результат склеювання записаний у третю матрицю. У третій матриці здійснене неповне склеювання змінних із записом результату до четвертої матриці.

$$
\begin{vmatrix}
0&0&0&0&0&1&0&1&1&0&0&0&1&1&1&1\\
0&0&0&1&0&1&0&1&1&0&0&1&1&1&1&1\\
0&0&1&0&0&1&0&1&1&0&1&0&1&1&1&1\\
0&0&1&1&0&1&0&1&1&0&1&1&1&1&1&1\\
0&1&0&0&0&1&0&1&1&1&0&0&1&1&1&1\\
0&1&0&1&0&1&0&1&1&1&0&1&1&1&1&1\\
0&1&1&0&0&1&0&1&1&1&1&0&1&1&1&1\\
0&1&1&1&0&1&0&1&1&1&1&1&1&1&1&1\\
\hline
1&0&0&0&0&0&0&0&1&0&0&0&1&0&1&0\\
1&0&0&1&0&0&0&0&1&0&0&1&1&0&1&0\\
1&0&1&0&0&0&0&0&1&0&1&0&1&0&1&0\\
1&0&1&1&0&0&0&0&1&0&1&1&1&0&1&0\\
1&1&0&0&0&0&0&0&1&1&0&0&1&0&1&0\\
1&1&0&1&0&0&0&0&1&1&0&1&1&0&1&0\\
1&1&1&0&0&0&0&0&1&1&1&0&1&0&1&0\\
1&1&1&1&0&0&0&0&1&1&1&1&1&0&1&0\\
\hline
1&0&0&0&0&0&0&1&1&0&0&0&1&0&0&1\\
1&0&0&1&0&0&0&1&1&0&0&1&1&0&0&1\\
1&0&1&0&0&0&0&1&1&0&1&0&1&0&0&1\\
1&0&1&1&0&0&0&1&1&0&1&1&1&0&0&1\\
1&1&0&0&0&0&0&1&1&1&0&0&1&0&0&1\\
1&1&0&1&0&0&0&1&1&1&0&1&1&0&0&1\\
1&1&1&0&0&0&0&1&1&1&1&0&1&0&0&1\\
1&1&1&1&0&0&0&1&1&1&1&1&1&0&0&1\\
\hline
1&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0\\
1&0&0&1&0&0&1&0&1&0&0&1&1&0&0&0\\
1&0&1&0&0&0&1&0&1&0&1&0&1&0&0&0\\
1&0&1&1&0&0&1&0&1&0&1&1&1&0&0&0\\
1&1&0&0&0&0&1&0&1&1&0&0&1&0&0&0\\
1&1&0&1&0&0&1&0&1&1&0&1&1&0&0&0\\
1&1&1&0&0&0&1&0&1&1&1&0&1&0&0&0\\
1&1&1&1&0&0&1&0&1&1&1&1&1&0&0&0\\
\hline
1&0&0&0&0&1&1&1\\
1&0&0&1&0&1&1&1\\
1&0&1&0&0&1&1&1\\
1&0&1&1&0&1&1&1\\
1&1&0&0&0&1&1&1\\
1&1&0&1&0&1&1&1\\
1&1&1&0&0&1&1&1\\
1&1&1&1&0&1&1&1
\end{vmatrix}
=
\begin{vmatrix}
0&&0&1&0&1\\
1&&0&0&0&0\\
1&&0&0&0&1\\
1&&0&0&1&0\\
1&&0&1&1&1\\
1&&1&1&1&1\\
1&&1&0&1&0\\
1&&1&0&0&1\\
1&&1&0&0&0
\end{vmatrix}
=
$$

$$
=\begin{vmatrix}
0&&0&1&0&1\\
1&&&0&0&\\
1&&&0&1&0\\
1&&&1&1&1
\end{vmatrix}
=
\begin{vmatrix}
0&&0&1&0&1\\
1&&&0&0&\\
1&&&0&&0\\
1&&&1&1&1
\end{vmatrix}. \qquad (33)
$$

У підсумку отримуємо мінімальну функцію:

$$F = \overline{x_1}\,\overline{x_5}x_6\overline{x_7}x_8 + x_1\overline{x_6}\,\overline{x_7} + x_1\overline{x_6}\,\overline{x_8} + x_1x_6x_7x_8. \quad (34)$$

У табл. 12 подані результати мінімізації функції $F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)$ картою Карно [19] та комбінаторним методом.

Таблиця 12

Результат мінімізації функції $F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)$

Картою Карно
$F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8) = x_1\overline{x_6}\,\overline{x_7} + x_1\overline{x_6}x_7\overline{x_8} + x_1x_6x_7x_8 + \overline{x_1}\,\overline{x_5}x_6\overline{x_7}x_8$
Комбінаторним методом
$F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8) = x_1\overline{x_6}\,\overline{x_7} + x_1\overline{x_6}\,\overline{x_8} + x_1x_6x_7x_8 + \overline{x_1}\,\overline{x_5}x_6\overline{x_7}x_8$

З огляду табл. 12 легко бачити, що комбінаторний метод дає другий кон'юнктерм булевої функції з меншим числом вхідних змінних.

6. Результати дослідження

Складність алгоритму мінімізації булевої функції – це кількісна характеристика, що відображає споживані алгоритмом ресурси під час його виконання. Основними ресурсами, якими оцінюється складність алгоритму, є час обчислення і простір пам'яті, необхідний для здійснення цього обчислення за даним алгоритмом. Для оцінки алгоритмічної складності процесу мінімізації булевої функції комбінаторним методом в якості споживаних алгоритмічних ресурсів беруться операції образних перетворень, які здійснюються під час пошуку мінімальної функції. Одна операція супер-склеювання, простого склеювання, узагальненого склеювання, поглинання та заміщення змінних приймається як одне перетворення. Кількість зазначених перетворень залежить від розрядності булевої функції, числа вихідних кон'юнктермів функції та структури таблиці істинності. Можлива кількість таких перетворень у залежності від розрядності булевої функції представлена у табл. 13.

Таблиця 13

Витрачені образні перетворення комбінаторного методу

Розрядність функції	Кількість образних перетворень комбінаторного методу
4	4–5
5	7–18
6	8–32
7	10–58
8	15–117

На рис. 2 представлена динаміка зростання кількості образних перетворень комбінаторним методом мінімізації зі збільшенням розрядності булевої функції.

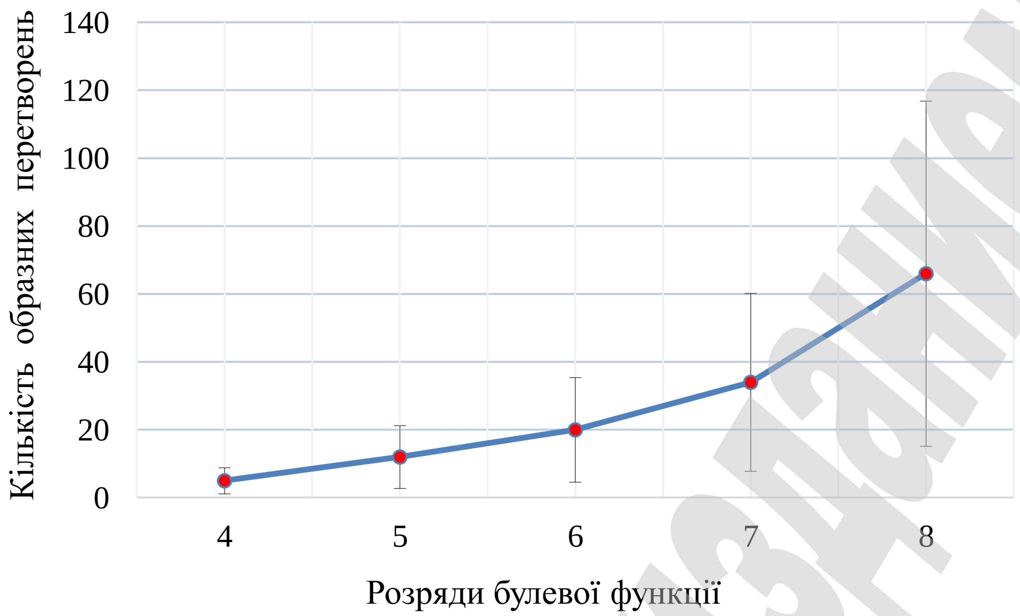

Рис. 2. Динаміка зростання кількості образних перетворень комбінаторного методу мінімізації, зі збільшенням розрядності булевої функції

За доступними даними можна у першому наближенні вважати складність алгоритму за комбінаторним методом лінійно залежним від числа образних перетворень з оцінкою складності – O(n) для n<7. Зі збільшенням числа змінних від n=6 до 8 динаміка зростання кількості перетворень характеризується законом O(n^2) з подальшим зростанням O($f(n)$) зі збільшенням розрядності булевої функції за поліноміальним законом.

У табл. 14 представлено порівняння процесів мінімізації булевих функцій методом паралельного розчеплення кон'юнктермів [14] та комбінаторним методом.

Таблиця 14

Порівняльна таблиця двох методів мінімізації булевих функцій

Метод паралельного розчеплення кон'юнктермів	Комбінаторний метод
1	2
Перший етап мінімізації булевої функції	
Метод паралельного розчеплення кон'юнктермів відноситься до класу евристичних методів мінімізації логічної функції. Процедура паралельного розчеплення кон'юнктермів полягає у накладанні на двійкові мінтерми m_1, m_2,..., m_k функції f масок літералів матриці-стовпця, внаслідок чого утворюється матриця кон'юнктермів з підматрицями розчеплених кон'юнктермів 1-, 2-, ..., (n–1)-, n-рангів. Визначається кількість тотожних розчеплених кон'юнктермів-копій, яка залежить від числа k заданих мінтермів. У спеціальній матриці виділяється утворені кон'юнктерми-копії, а відбір елементів покриття виконується з пріоритетом для матриць меншого рангу	Мінімізація логічної функції комбінаторним методом ґрунтується на образних перетвореннях, що збільшує інформативність процесу мінімізації, порівняно з вербальним алгебричним способом мінімізації функції. На першому кроці виявляють блоки зі змінними, які можна склеювати. На другому кроці здійснюють пошук наборів пар блоків з можливістю їх мінімізації склеюванням, поглинанням, заміщенням та узагальненим склеюванням змінних. Отримані набори блоків знову мінімізують подібним способом, і т. д. – до отримання тупикової ДНФ

Продовження таблиці 14

1	2
Другий етап мінімізації булевої функції	
Здійснюється видалення надлишкових імплікант і отримання МДНФ. Для цього будується таблиця покриття конституант. Виявлення мінімальної функції можливе за алгебричним методом Петрика	Спроба мінімізації тупикової ДНФ за методом Блейка-Порецького: здійснюють всі узагальнені склеювання змінних – $xy+\bar{x}z=xy+\bar{x}z+yz$ (1) з наступним проведенням усіх поглинань, або проведенням щонайменше двох операцій узагальненого склеювання змінних за формулою $xy+\bar{x}z+yz=xy+\bar{x}z$ на одну операцію (1). Зазначена процедура перетворення функції можлива й на першому етапі мінімізації
Автоматизація процесу мінімізації булевої функції	
Оскільки метод використовує математичний апарат матриць, підматриць, масок та інші обчислення, застосування методу потребує його автоматизації	Застосовуючи операцію супер-склеювання змінних, комбінаторний метод може обходитись без автоматизації процесу мінімізації булевих функцій з числом змінних до 10

Технологія мінімізації булевої функції комбінаторним методом дозволяє обходитись без автоматизації процесу мінімізації функції з числом змінних до 10. Для ряду випадків ручна мінімізація логічної функції можлива й з числом змінних більше 10.

7. SWOT-аналіз результатів досліджень

Strengths. До сильної сторони комбінаторного методу можна віднести зменшення складності алгоритму мінімізації булевої функції, що дозволяє обходитись без автоматизації процесу мінімізації булевих функцій з числом змінних до 10. Це вигідніше у порівнянні з аналогами за такими чинниками:

− меншою вартістю розробки та впровадження, оскільки значна доля функцій, що мінімізуються припадає на функції з числом змінних не більше 16, і, отже, у загальному випадку потреба в автоматизації процесу мінімізації функції зменшується;

− збільшенням продуктивності ручної мінімізації 4–10 розрядних функцій, що сприяє контролю та вивченню алгоритму мінімізації логічної функції.

Weaknesses. Слабка сторона комбінаторного методу при ручній мінімізації пов’язана зі зростанням трудомісткості обчислення зі збільшенням числа змінних логічної функції. Негативні внутрішні фактори притаманні комбінаторному методу ручної мінімізації булевої функції та полягають у збільшенні часу отримання мінімальної функції при зростанні числа змінних заданої функції.

Opportunities. Перспективою подальших досліджень комбінаторного методу може бути вироблення протоколу оптимального чергування алгебричних перетворень з метою підвищення точності розв’язку задачі мінімізації функції. Додаткові можливості, що можуть принести впровадження комбінаторного методу мінімізації булевої функції, полягають у створенні та підтримці бібліотеки графічних образів з метою оптимізації алгоритму пошуку мінімальної функції за обраними критеріями.

Threats. Протокол мінімізації булевої функції комбінаторного методу є незалежним від протоколів інших методів мінімізації, тому загроза негативної дії на об’єкт дослідження зовнішніх чинників мінімальна. До певної міри аналогом комбінаторного методу мінімізації булевої функції є метод Куайна-Мак-Класкі. На даний момент метод Куайна-Мак-Класкі кращий тим, що для нього вже створений алгоритм автоматизованого пошуку мінімальної функції.

8. Висновки

1. Впровадження алгебричної операції супер-склеювання змінних дає змогу спростити процедуру мінімізації булевої функції без втрати її функціональності.
2. 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.
3. 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. Оцінка складності алгоритму пошуку мінімальної функції комбінаторним методом складає $O(n)$ і є лінійною для $n<7$. Зі збільшенням числа змінних від $n=6$ до 8 динаміка зростання кількості перетворень характеризується законом $O(n^2)$ з подальшим зростанням $O(f(n))$ зі збільшенням розрядності булевої функції за поліноміальним законом.
4. Ефективність комбінаторного методу демонструється прикладами мінімізації функції, запозичених з робіт інших авторів з метою порівняння:

- *приклад* 4 [13], *приклад* 5 [14], *приклад* 6 [15], *приклад* 8 [16] – мінімізація 4-розрядних булевих функцій;
- *приклад* 7 [14] – мінімізація системи 4-розрядних булевих функцій;
- *приклад* 9 [17] – мінімізація 5-розрядних булевих функцій;
- *приклад* 10 [18] – мінімізація 6-розрядних булевих функцій;
- *приклад* 11 [19] – мінімізація 8-розрядних булевих функцій.

З огляду на зазначені приклади комбінаторний метод мінімізації функції дає підставу для доцільності застосування його у процесах мінімізації логічної функції.

Література

1. Quine–McCluskey algorithm [Electronic resource] // Wikipedia. – August 1, 2017 – Available at: \www/URL: https://en.wikipedia.org/wiki/Quine%E2%80%93McCluskey_algorithm
2. Riznyk, V. Minimization of Boolean functions by combinatorial method [Text] / V. Riznyk, M. Solomko // Technology Audit and Production Reserves. – 2017. – Vol. 4, No. 2 (36). – P. 49–64. doi:10.15587/2312-8372.2017.108532
3. Solairaju, A. Optimal Boolean Function Simplification through K-Map using Object-Oriented Algorithm [Text] / A. Solairaju, R. Periyasamy // International Journal of Computer Applications. – 2011. – Vol. 15, No. 7. – P. 28–32. doi:10.5120/1959-2621
4. Kumar, R. Cubical Representation and Minimization through Cubical Technique A Tabular Approach [Text] / R. Kumar, S. Rawat // International Journal of Applied Engineering Research. – 2016. – Vol. 11, No. 7. – P. 4822–4829.

5. Stergiou, S. A Fast and Efficient Heuristic ESOP Minimization Algorithm [Text] / S. Stergiou, K. Daskalakis, G. Papakonstantinou // Proceedins of the 14th ACM Great Lakes symposium on VLSI – GLSVLSI '04. – Boston, 2004. doi:10.1145/988952.988971

6. Dusa, A. Enhancing the Minimization of Boolean and Multivalue Output Functions WitheQMC [Text] / A. Dusa, A. Thiem // The Journal of Mathematical Sociology. – 2015. – Vol. 39, No. 2. – P. 92–108. doi:10.1080/0022250x.2014.897949

7. Voudouris, D. Exact ESCT minimization for functions of up to six input variables – PRELIMINARY VERSION [Electronic resource] / D. Voudouris, M. Sampson, G. Papakonstantinou. – 2005. – 17 p. – Available at: \www/URL: http://mule.cslab.ece.ntua.gr/xor/docs/xmin6.pdf

8. Valli, M. A State of Appraoches on Minimization of Boolean Functions [Text] / M. Valli, R. Periyasamy, J. Amudhavel // Journal of Advanced Research in Dynamical and Control Systems. – 2017. – No. 12. – P. 1322–1341.

9. Boyar, J. A New Combinational Logic Minimization Technique with Applications to Cryptology [Text] / J. Boyar, R. Peralta // Lecture Notes in Computer Science. – 2010. – P. 178–189. doi:10.1007/978-3-642-13193-6_16

10. Eungi, K. Derivations of Single Hypothetical Don't-Care Minterms Using the Quasi Quine-McCluskey Method [Text] / K. Eungi // Journal of the Korea Industrial Information Systems Research. – 2013. – Vol. 18, No. 1. – P. 25–35. doi:10.9723/jksiis.2013.18.1.025

11. Dubrova, E. Minimization of Multiple-Valued Functions in Post Algebra [Electronic resource] / E. Dubrova, Y. Jiang, R. Brayton. – 2001. – 5 p. Available at: \www/URL: https://pdfs.semanticscholar.org/e9fa/a422aad8b92c0448eb8d41425852717cb637.pdf

12. Anjuli, S. A. 2-Bit Magnitude Comparator Design Using Different Logic Styles [Text] / A. S. Anand // International Journal of Engineering Science Invention. – 2013. – Vol. 2, No. 1. – P. 13–24. – Available at: \www/URL: http://www.ijesi.org/papers/Vol(2)1%20(version%202)/C211324.pdf

13. Buniak, A. Elektronika ta mikroskhemotekhnika [Text] / A. Buniak. – Kyiv: Aston, 2001. – 382 p.

14. Rytsar, B. Ye. Minimization of logic functions system by konjuncterms parallel splittingmethod [Text] / B. Ye. Rytsar // Bulletin of the Lviv Polytechnic National University. Radio Electronics and Telecommunications. – 2013. – No. 766. – P. 18–27.

15. Rytsar, B. Ye. New minimization method of logical functions in polynomial set-theoretical format. 1. Generalized rules of conjuncterms simplification [Text] / B. Ye. Rytsar // Upravliaiushchie sistemy i mashiny. – 2015. – Vol. 2. – P. 39–57.

16. Samofalov, K. G. Prikladnaia teoriia tsifrovyh avtomatov [Text] / K. G. Samofalov, A. M. Romlinkevich, V. N. Valuiskii, Yu. S. Kanevskii, M. M. Pinevich. – Kyiv: Vishcha shkola, 1987. – 375 p.

17. Plehanov, A. Simmetrichnye karty kak sredstvo minimizatsii bulevyh funktsii [Electronic resource] / A. Plehanov // Geektimes. – March 8, 2016. – Available at: \www/URL: https://geektimes.ru/post/272294/

18. Plehanov, A. Eshche raz o minimizatsii bulevyh funktsii [Electronic resource] / A. Plehanov // Habrahabr. – May 5, 2016. – Available at: \www/URL: https://habrahabr.ru/post/283030/

19. Triohmernaia karta Karno [Electronic resource] // Cyclowiki. – February 9, 2016. – Available at: \www/URL: http://cyclowiki.org/wiki/Трёхмерная_карта_Карно

20. Karta Karno [Electronic resource] // Wikipedia. – September 30, 2017. Available at: \www/URL: https://ru.wikipedia.org/wiki/Карта_Карно