УДК 681.325
DOI: 10.15587/1729-4061.2020.220094

Розроблення методу образних перетворень для мінімізації булевих функцій в імплікативному базисі

М. Т. Соломко, Ю. В. Батишкіна, І. С. Войтович, Л. В. Зубик, С. М. Бабич, К. П. Музичук

Проведеними дослідженнями встановлена можливість зменшення обчислювальної складності, збільшення продуктивності спрощення булевих функцій у класі досконалих імплікативних нормальних форм (ДІНФ-1 та ДІНФ-2) методом образних перетворень.

Поширення методу образних перетворень на процес спрощення функцій імплікативного базису здійснено за допомогою розробленої алгебри імплікативного базису у вигляді правил спрощення ДІНФ-1 та ДІНФ-2 функцій імплікативного базису. Особливістю спрощення функцій імплікативного базису на бінарних структурах 2-(n, b)-блок-схем (англ. 2-(n, b)-designs) є використання аналогів досконалих диз'юктивних нормальних форм (ДДНФ) та досконалих кон'юктивних нормальних форм (ДКНФ) булевих функцій. Зазначені форми функцій визначають правила перетворення на бінарних структурах функцій імплікативного базису.

Показано, що досконалу імплікативну нормальну форму n-місткої функції імплікативного базису можна подати бінарними наборами або матрицею. Логічні операції над структурою матриці забезпечують результат спрощення функцій імплікативного базису. Це дозволяє зосередити принцип мінімізації у межах таблиці істинності заданої функції та обійтись без допоміжних об'єктів, як то карта Карно, діаграми Вейча та ін.

Розглянутий метод дозволяє:

– зменшити алгоритмічну складність спрощення ДІНФ-1 та ДІНФ-2;

– збільшити продуктивність спрощення функцій імплікативного базису на 100–200 %;

– демонструвати наочність процесу мінімізації ДІНФ-1 або ДІНФ-2;

Є підстави стверджувати, що мінімізація функцій імплікативного базису методом образних перетворень виводить проблему мінімізації ДІНФ-1 та ДІНФ-2 на рівень добре досліджених задач у класі диз'юнктивно-кон'юнктивних нормальних форм булевих функцій.

Ключові слова: метод образних перетворень, мінімізація функцій імплікативного базису, функція імплікації, ДІНФ-1, ДІНФ-2.

1. Вступ

Технологія проектування булевих функцій, які увійшли до логічних базисів, може використовувати реалізацію, що ґрунтується на тих чи інших фізичних явищах. Наприклад, властивостям напівпровідників відповідають функції Пірса (Вебба) та Шеффера, а за допомогою магнітних явищ можна реалізувати імплікативний базис.

Імплікація (від лат. implico – тісно пов’язаний) – це логічна операція, що відповідає зв’язці «якщо …, то …» (IF-THEN), коли з двох висловлювань *A* і *B* утворюється умовне висловлювання «якщо *A*, то *B*». Імплікацією часто називають і саме умовне висловлювання, а також його формалізовані аналоги, як то формули логічних обчислень, які містять знак імплікації (наприклад, «$\supset$» або «$\rightarrow$») і мають вигляд $A \supset B$ ($A \rightarrow B$), де *A* і *B* – формули для логічного обчислення. Операція імплікації використовується при опису лінгвістичних закономірностей за допомогою алгебри предикатів, а також для запису правил формальної граматики [1].

Логічна функція $f = x_1 \rightarrow x_2$ (імплікація пряма x_1 в x_2) є диз’юнкція –

$$f = x_1 \rightarrow x_2 = \overline{x_1} + x_2, \qquad (1)$$

тому значення функції «хибно» одержується тільки тоді, коли аргумент x_1 приймає значення «істинно», а аргумент x_2 приймають значення «хибно» (табл. 1).

Таблиця 1
Таблиця істинності функції $f = x_1 \rightarrow x_2$

x_1	x_2	$\overline{x_1}$	$f = x_1 \rightarrow x_2 = \overline{x_1} + x_2$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Логічна схема, що реалізує функцію (1), представлена на рис. 1.

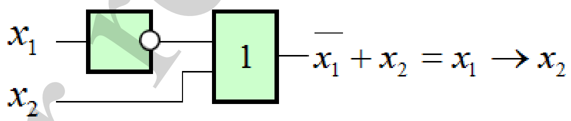

Рис. 1. Логічна схема, що реалізує функцію прямої імплікації $f = x_1 \rightarrow x_2$

Функція імплікації (1) може мати алгебричну форму:

$$f\left(x_1, x_2\right) = \begin{cases} 1, \text{ якщо } x_1 \le x_2, \\ 0, \text{ якщо } x_1 > x_2. \end{cases} \qquad (2)$$

На наборах змінних при яких функція (2) повертає «1», значення бітів у стовпчику «x_1» або співпадає зі значенням бітів у стовпчику «x_2», або є меншим за нього (табл. 2).

Таке відношення між бітами стовпців «x_1» та «x_2» означає, що стовпець «x_1» є складовою частиною стовпця «x_2» Таким чином, термін «імплікує» означає «є складовою частиною» (x_1 є складовою частиною x_2) [2].

Таблиця 2

Таблиця істинності на наборах змінних де функція f=x_1→x_2 повертає «1»

x_1	x_2	$\overline{x_1}$	$f = x_1 \to x_2 = \overline{x_1} + x_2$
0	0	1	1
0	1	1	1
1	1	0	1

Мінімальні логічні базиси за участю імплікації: {→, NOT}, {→, 0}, {→, ⊕}, {→, ←}. Базисами також є {←, NOT}, {←, 1}.

Функції канонічного базису {NOT, OR, AND} подаються через імплікацію наступним чином:

$$\overline{x} = x \to 0; \tag{3}$$

$$x_1 + x_2 = \overline{x_1} \to x_2; \tag{4}$$

або

$$x_1 + x_2 = \overline{x_1} \to x_2 = \left(x_1 \to 0\right) \to x_2;$$

$$x_1 \cdot x_2 = \overline{\overline{x_1} + \overline{x_2}} = \overline{x_1 \to \overline{x_2}}; \tag{5}$$

або

$$x_1 \cdot x_2 = \overline{\overline{x_1} + \overline{x_2}} = \overline{x_1 \to \overline{x_2}} = \overline{x_1 \to (x_2 \to 0)} = \left(x_1 \to \left(x_2 \to 0\right)\right) \to 0.$$

Імплікація забезпечує функціонально повний базис – кожна булева функція може бути реалізована за допомогою сполучення елементів NOT, або NOT-OR (рис. 2).

Функціональна повнота система перемикальних функцій забезпечує можливість подання довільної функціональної залежності від заданої кількості аргументів за допомогою мінімальної кількості базових функцій (операцій). Ці функції (операції) сукупно мають властивість функціональної повноти, а, отже, й можливість синтезу комбінаційної схеми, що відтворює функціональну залежність, за допомогою мінімальної кількості типів логічних елементів. Однак, при цьому не вирішується питання оптимальності комбінаційної схеми. Як показує практика проектування логічних схем об’єднання елементних базисів, які належать кільком функціонально повним системам (наприклад, системам {АБО-НЕ}, {І-НЕ}, {І, АБО, НЕ}), дозволяє будувати оптимальні комбінаційні схеми (за показниками апаратної складності та швидкодії). Використання елементного базису лише одніїї функціонально повної системи перемикальних функцій у загальному випадку не забезпечує отримання оптимальної комбінаційної схеми [2].

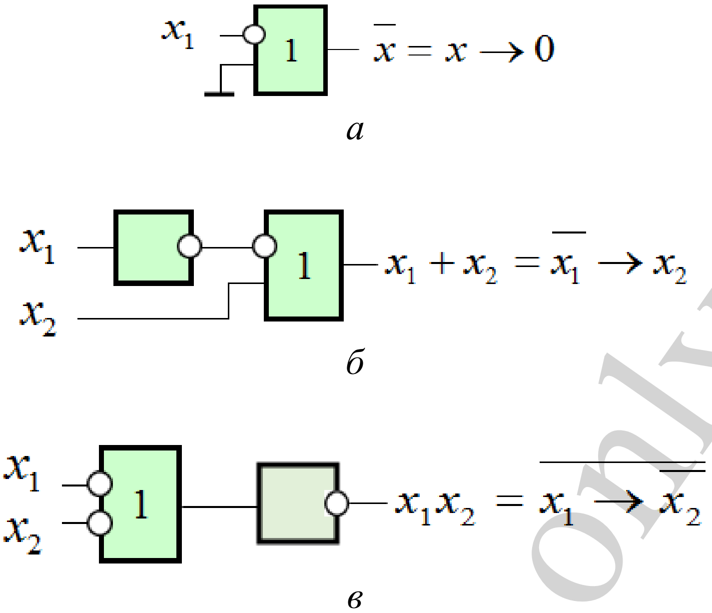

Рис. 2. Реалізація елементів канонічного базису {NOT, OR, AND} на елементах NOT, NOT-OR: *а* – інвертор, *б* – диз'юнкція, *в* – кон'юнкція

Процес мінімізації логічних функцій займає важливе положення всередині технології проектування цифрових компонентів. У зв'язку з цим актуальною проблемою залишається забезпечення адекватної відповідності розробленого продукту заданим специфікаціям собівартості, спрощення та гарантії отримання оптимального результату мінімізації різних представлень логічних функцій.

Оскільки імплікативний базис належить області оптимізації логічних функцій [3], актуальними є дослідження направлені, зокрема, на вдосконалення таких чинників, як:

– методи спрощення функцій імплікативного базису;
– мінімізація логічних схем на основі функцій імплікації;
– достовірність оптимального результату мінімізації імплікативного базису.

2. Аналіз літературних даних та постановка проблеми

Логічний затвор IMPLY та логічна схема на основі мемристорів представлені у роботі [4]. Мемристорні пристрої можуть застосовуватись як логічні схеми. У цій структурі логічної пам'яті кожен мемристор використовується як обчислювальний логічний елемент або як тригер на різних етапах обчислювального процесу. Логічний стан пристрою визначається опором мемристора. Зазначена структура може бути інтегрована в мемристорний кросбар, який зазвичай використовується для пам'яті. У роботі [4] представлена методологія його проектування. На основі розглянутої методології, проектування 8-bit повного суматора бінарних кодів подано як тематичне дослідження.

Мемрістори змінюють опір під контролем напруги, можуть зберігати своє значення після зняття напруги. Невеликий розмір мемристорів робить їх корисними для проектування надкомпактних систем пам'яті. Мемристори виконують логічні примітиви і, отже, можуть бути використаний для реалізації логічних функцій з використанням різних стилів логічного проектування У статті [5] також розглядається технологія виготовлення мемристорів, схеми моделей, методи реалізації логічних функцій та різні обчислювальні методології, які можуть бути використані в обчислювальних компонентах.

Метод генерації послідовностей керуючих сигналів, що дозволяє обчислити довільну n-вхідну 1-вихідну булеву функцію, використовуючи лише два робочих мемристори, представлено у роботі [6]. Описаний підхід заснований на використанні рекурсивної булевої формули, яка надає шлях для реалізації булевої функції над функціонально повною базою {imply, FALSE}, де impply є двовхідною булевою функцією імплікації $x \rightarrow y = \overline{x} + y$.

Характеристики мемрмстора, такі як висока швидкість, низька потужність та пасивне збереження пам’яті роблять його придатним для застосування в декількох областях (нейронні системи, цифрові та аналогові схеми, блоки пам’яті). Мемристор володіє різними властивостями, які дозволяють ряду галузей застосовувати його, скориставшись бажаними перевагами. Мемристор “Implication”, який реалізує логіку IMPLY, забезпечує всі логічні операції, що можливі в конструкціях, складених лише з мемрісторів. Основна увага статті [7] фокусується на повному суматорі бінарних кодів, логічна структура якого складена на основі мемристорів і яка використовує лише імплікативну логіку. У статті [7] запропонований алгоритм, що забезпечує меншої кількості кроків виконання логіки суматора та меншу кількість мемристорів.

Будь-яка логічна операція може бути реалізована в імплікативних мемристорних схемах, що характеризуються низьким споживанням енергії та розміром нанометрового рівня. У роботі [8] представлений метод перетворення графів And-Inverter Graphs (AIGs) для логічних функцій у мережі на основі імплікації. Оптимізований процес копіювання застосовується для зменшення затримки та площі ланцюгів пам’яті. Експериментальні дослідження проведені за набором тестів, що включає 33 функції з вхідними змінними від 3 до 41. Експериментальні результати порівнюються з результатами оригінального алгоритму та іншого методу відображення на основі графів (MIG). Продемонстровано, що вдосконалений алгоритм може отримати кращі показники затримки для більшості функцій тестового набору.

Логіка імплікації є однією з основних технологій для мемристорів. У роботі [9] представлена оптимальна конструкція повного суматора бінарних кодів на основі мемристора з використанням імплікативної логіки. Для розглянутої конструкції потрібно 27 мемристорів і менше площі у порівнянні з типовими 8-бітними повними суматорами на базі CMOS технології. Також представлений повний суматор, який потребує лише 184 обчислювальні кроки. Зазначається, що продуктивність роботи суматора збільшилась на 20 %.

Фізична реалізація мемристорів або мемристорних пристроїв, що поєднують електричні властивості елемента пам’яті та резистора вперше були представлені Леоном Чуа у 1971 р. Такі пристрої характеризуються однією або кількома змінними стану, що визначають опір вимикача, залежно від його історії напруги. У роботі [10] показано, що це сімейство нелінійних пристроїв динамічної пам'яті також може бути використано для логічних операцій. Продемонстровано, що розглянуті пристрої можуть виконувати матеріальну імплікацію (IMP), що є фундаментальною логічною операцією булевої логіки над двома змінними p і q, така що pIMPq еквівалентна (NOTp) ORq. Таким чином, включені до відповідної схеми, пам'ятні комутатори можуть виконувати логічні операції, що відповідають стану, для яких ті самі пристрої одночасно виконують функції затворів (логіка) та засувок (пам'ять), які використовують опір замість напруги або заряду як змінну фізичного стану.

Яка кількість мемристорів потрібна для виконання заданої логічної операції розглядається у роботі [11]. Продемонстровано, що мемристори, природно придатні для виконання логіки імплікації, замість булевої логіки. Зазначається також, що мемристор можна використовувати в якості логічного затвору і тригера. Будучи функціонально повною, логіка імплікації може бути використана для обчислення будь-якої булевої функції. Однак, виконуючи логічну імплікацію з пристроями, що містять дані, для збереження проміжних результатів потрібні додаткові мемристори. У роботі [11] представлений ефективний спосіб обчислення будь-якої з великою кількістю мемристорів. Також враховується довжина відповідної обчислювальної послідовності.

Нові методи логічного синтезу для неповних багаторівневих двійкових ланцюгів з використанням імплікаційних елементів на основі мемристорів розглянуто у роботі [12]. Перший метод перевіряє припущення про використання лише двох працюючих мемристорів. Алгоритм мінімізує кількість імплікативних (IMPLY) шлюзів, що відповідає мінімізації кількості імпульсів або часу затримки. Перший метод перевіряється з іншими методами синтезу, такими як модифікований SOP та Exclusive-Or Sum of Products (ESOP) з мінімальною кількістю працюючих мемристорів. Здійснений аналіз проблеми зменшення кількості шлюзів IMPLY, додавши більше працюючих мемристорів. Впроваджено діаграми послідовності та нове позначення, подібне до того, що використовується в оборотній логиці.

У роботі [13] продемонстровано, що всі булеві функції можна обчислити за допомогою двох мемристорів. Для цього потрібна рекурсивна сполучна форма для введення булеві функції. Представлено також процедуру синтезу відповідної обчислювальні послідовності. Результат є важливий для мінімізації складних логічних ланцюгів щодо кількості використовуваних мемристорів.

Розглянуті літературні джерела [4–13] представляють реалізацію четвертого базового елемента схемотехніки – мемристора, який являє собою доповнення до резистора, конденсатора та індуктивності.

За пропозицією Леонома Чуа [14], є четвертий базовий елемент електричних ланцюгів – наряду з індуктивністю, конденсатором та резистором, який повинен зв'язувати заряд зі змінами магнітного поля таким співвідношенням:

$$d\varphi = Mdq.$$

Було показано [14], коли мемристивність M є постійною величиною, то мемристор веде себе як звичайний резистор. Однак, якщо мемристивність M є функцією заряду q, співвідношення між напругою на клемах мемристора і зарядом, який пройшов через елемент, визначається формулою:

$$v(t) = M(q)i(t) = M\left(\int_{-\infty}^{t} \mathrm{i}(\tau)\,d\tau\right)i(t).$$

У кожен момент часу поведінка мемристора подібна поведінці резистора, фактичне значення опору якого залежить від часового інтегралу струму, що протікає через пристрій. Історія функціонування пристрою визначає його властивості у кожен конкретний момент часу. Таким чином, термін «мемристор» означає «резистор з пам'яттю».

За думкою дослідників з компанії Hewlett-Packard, мемристори найбільш ефективні, коли використовується логіка, що ґрунтується на операції імплікації [15].

Паралельне підключення двох мемристорів реалізує функцію матеріальної імплікації (material implication) [16]. Поряд з універсальними елементами AND-NOT і OR-NOT функція імплікації, разом з функцією константа нуль, формує функціонально повний базис (табл. 3) [17].

Таблиця 3

Обчислювальна універсальність операцій IMP (імплікація) і FALSE: 16 бінарних булевих операцій над двома логічними величинами

Операція	Таблиця істинності				Еквівалентна операція
p	1	1	0	0	=*p*
q	1	0	1	0	=*q*
TRUE	1	1	1	1	=*p* IMP *p*
p OR *q*	1	1	1	0	=(*p* IMP 0) IMP *q*
q IMP *p*	1	1	0	1	= *q* IMP *p*
p	1	1	0	0	= (*p* IMP 0) IMP 0
p IMP *q*	1	0	1	1	= *p* IMP *q*
q	1	0	1	0	=(*q* IMP 0) IMP 0
p EQUAL *q*	1	0	0	1	=((*p* IMP *q*) IMP ((*q* IMP *p*) IMP 0)) IMP 0
p AND *q*	1	0	0	0	=(*p* IMP (*q* IMP 0)) IMP 0
p NAND *q*	0	1	1	1	=*p* IMP (*q* IMP 0)
p XOR *q*	0	1	1	0	=(*p* IMP *q*) IMP ((*q* IMP *p*) IMP 0)
NOT *q*	0	1	0	1	=*q* IMP 0
p NIMP *q*	0	1	0	0	=(*p* IMP *q*) IMP 0
NOT *p*	0	0	1	1	=*p* IMP 0
q NIMP *p*	0	0	1	0	=(*q* IMP *p*) IMP 0
p NOR *q*	0	0	0	1	=((*p* IMP 0) IMP *q*) IMP 0
FALSE	0	0	0	0	=0

Це дозволяє виконувати всі 16 перемикальних функцій двох змінних. Але до цих пір такий базис не використовується в обчислювальній техніці [17].

Літературні джерела [4, 7, 9, 11] підтверджують не використання функціонально повного імплікативного базису для мінімізації логічних функцій. Тут результат мінімізації подається у булевому базисі. Тільки після такої мінімізації спеціальні алгоритми проводять заміну елементів базису {AND, OR, NOT} на елементи базису {→, NOT} або {→, 0}.

Для належного використання функціонально повного імплікативного базису необхідна алгебра у складі правил спрощення імплікативних функцій.

Метод образних перетворень забезпечує мінімізацію логічних функцій безпосередньо в імплікативному базисі. Таким чином, розглянуті алгоритми та методи мінімізації перемикальних функцій [4, 7, 9, 11] і метод образних перетворень посідають відмінні підходи, а відтак вбачають різні перспективи стосовно технологічної можливості мінімізації функцій в імплікативному базисі. Зокрема перспективою є застосування алгебри у складі правил рівносильного перетворення функцій імплікативного базису {→, NOT}, {→, 0}, що дасть розширення можливостей аналітичного методу.

У зв'язку з цим є підстави вважати, що процедура мінімізації перемикальних функцій, яка представлена алгоритмами та методами мінімізації [4, 7, 9, 11], є недостатньою для проведення теоретичних досліджень стосовно оптимальної мінімізації функцій імплікативного базису. Це обумовлює необхідність досліджень з рівносильними образними перетвореннями імплікативних функцій, зокрема для базисів {→, NOT}, {→, 0}. У прикладному відношенні зазначений підхід дозволяє розширити можливості технології проектування цифрових компонентів на основі імплікативних базисів {→, NOT}, {→, 0}.

3. Мета та задачі дослідження

Метою роботи є поширення методу образних перетворень на мінімізацію булевих функцій у класі досконалих імплікативних нормальних форм (ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1). Це дає можливість спростити, збільшити продуктивність мінімізації функцій імплікативного базису, розробивши алгебричні правила логічних перетворень.

Для досягнення поставленої мети необхідно вирішити такі задачі:

– встановити адекватність застосування методу образних перетворень для мінімізації булевих функцій імплікативного базису, зокрема встановити герменевтику логічних операцій в імплікативному базисі;

– створити алгебру імплікативного базису у частині необхідних правил мінімізації булевих функцій;

– провести аналіз ефективності мінімізації функцій імплікативного базису методом образних перетворень та прикладів мінімізації функцій в імплікативному базисі з метою порівняння.

4. Досконалі імплікативні нормальні форми булевих функцій

Всі визначення для функцій алгебри логіки у базисі {І, АБО, НЕ} мають свої аналоги і для імплікативного базису $\{\rightarrow, \mathrm{NOT}\}$ (табл. 4). Заміна базису {І, АБО, НЕ} на базис $\{\rightarrow, \mathrm{NOT}\}$ можлива на підставі формул (6)–(8).

Таблиця 4
Тезáуруси логічних базисів

№ з/п	Тезáурус базису {AND, OR, NOT}	Тезáурус базису $\{\rightarrow, \mathrm{NOT}\}$
1	Досконала диз’юктивна нормальна форма (ДДНФ)	Досконала імплікативна нормальна форма –2 (ДІНФ-2)
		Досконала імплікативна нормальна форма –2.1 (ДІНФ-2.1)
2	Досконала кон’юктивна нормальна форма (ДКНФ)	Досконала імплікативна нормальна форма –1 (ДІНФ-1)
		Досконала імплікативна нормальна форма –1.1 (ДІНФ-1.1)
3	Мінімальна диз’юктивна нормальна форма (МДНФ)	Мінімальна імплікативна нормальна форма –2 (МІНФ-2)
		Мінімальна імплікативна нормальна форма –2.1 (МІНФ-2.1)
4	Мінімальна кон’юктивна нормальна форма (МКНФ)	Мінімальна імплікативна нормальна форма –1 (МІНФ-1)
		Мінімальна імплікативна нормальна форма –1.1 (МІНФ-1.1)

Отримання ДІНФ-1 та ДІНФ-2 демонструють приклади 1 і 2.

4. 1. ДІНФ-1.

Досконала диз’юктивна нормальна форма (ДДНФ) та досконала кон’юктивна нормальна форма (ДКНФ) булевих функцій можуть бути виражені через функції, відмінні від кон’юнкції і заперечення або диз’юнкції і заперечення. Представити ДДНФ або ДКНФ булевих функцій можна, наприклад, за допомогою заперечення та імплікації.

Теорема 1 [18]. Будь-яка функція алгебри логіки, крім тотожної одиниці, може бути представлена у наступному вигляді:

$$f\left(x_1, x_2, \dots, x_n\right) = \underset{0}{\&}\left(\overline{\overline{x_1^{\delta_1} \rightarrow x_2^{\overline{\delta_2}}} \rightarrow \dots \rightarrow x_{n-1}^{\overline{\delta_{n-1}}}} \rightarrow x_n^{\overline{\delta_n}}\right). \quad (6)$$

Доведення теореми 1 можна знайти у [18].

Імплікативна форма (6) є аналогом ДКНФ. Присвоїмо запису (6) класифікацію *досконалої імплікативної нормальної форми* – 1 (ДІНФ-1) булевої функції.

Для представлення булевої функції у ДІНФ-1 необхідно входження всіх аргументів x_i, крім x_1, у терми ДІНФ-1 функції з запереченням, якщо $x_i^{\delta_i} = 1$, і без заперечення – у протилежному випадку. Для x_1 умови входження у терми ДІНФ-1 функції протилежні.

Будь-якому бінарному набору відповідає терм ДІНФ-1 функції $\overline{\overline{x_1^{\delta_1}\to x_2^{\overline{\delta_2}}}\to\ldots\to x_{n-1}^{\overline{\delta_{n-1}}}}\to x_n^{\overline{\delta_n}}$ і, навпаки, – терму ДІНФ-1 функції $\overline{\overline{x_1^{\delta_1}\to x_2^{\overline{\delta_2}}}\to\ldots\to x_{n-1}^{\overline{\delta_{n-1}}}}\to x_n^{\overline{\delta_n}}$ відповідає бінарний набір (кортеж). Наприклад, набору <1100> відповідає терм ДІНФ-1 $\overline{\overline{x_1\to\overline{x_2}}\to x_3}\to x_4$, а терму ДІНФ-1 $\overline{\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3}\to x_4}\to\overline{x_5}$ відповідає набір <01001>.

Приклад 1. Функцію $f(x_1, x_2, x_3, x_4)$ (табл. 5) представити у ДІНФ-1.

Таблиця 5
Таблиця істинності логічної функції $f(x_1, x_2, x_3, x_4)$

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	1	1	0	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	1	1	0	1	0	1
0	0	1	1	1	1	0	1	1	0
0	1	0	0	1	1	1	0	0	1
0	1	0	1	0	1	1	0	1	1
0	1	1	0	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	0

Складемо терми φ_i імплікативної функції для наборів табл. 5, на яких $f(x_1, x_2, x_3, x_4)=0$:

$$\varphi_1=\overline{\overline{\overline{x_1}\to x_2}\to x_3}\to\overline{x_4};\ \ \varphi_2=\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3}\to\overline{x_4};$$

$$\varphi_3=\overline{\overline{x_1\to x_2}\to\overline{x_3}}\to\overline{x_4};\ \ \varphi_4=\overline{\overline{x_1\to\overline{x_2}}\to\overline{x_3}}\to\overline{x_4}.$$

Тоді запис (7)

$$\begin{aligned} f(x_1,x_2,x_3,x_4)=&\left(\overline{\overline{\overline{x_1}\to x_2}\to x_3}\to\overline{x_4}\right)\ \&\\ &\&\left(\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3}\to\overline{x_4}\right)\ \&\\ &\&\left(\overline{\overline{x_1\to x_2}\to\overline{x_3}}\to\overline{x_4}\right)\&\\ &\&\left(\overline{\overline{x_1\to\overline{x_2}}\to\overline{x_3}}\to\overline{x_4}\right). \end{aligned}\qquad(7)$$

буде представляти функцію $f(x_1, x_2, x_3, x_4)$ у ДІНФ-1.

4. 2. Двійковий еквівалент ДІНФ-1

Для методу образних перетворень доцільно використовувати бінарний аналог заданої булевої функції, у тому числі й функції імплікативного базису.

Оскільки ДІНФ-1 є аналогом ДКНФ функції булевого базису, спрощення ДІНФ-1 здійснюється за методом Нельсона [19]. Двійковий еквівалент змінних ДІНФ-1 функції $f\,(x_1, x_2, x_3, x_4)$ (11) можна подати двома варіантами:

$$
\begin{aligned}
F_{\text{ДІНФ-1}} = {} & \left(\overline{\overline{\overline{x_1} \to x_2} \to x_3} \to x_4\right)\left(\overline{\overline{\overline{x_1} \to \overline{x_2}} \to x_3} \to x_4\right) \times \\
& \times \left(\overline{\overline{\overline{x_1} \to \overline{x_2}} \to x_3} \to \overline{x_4}\right)\left(\overline{\overline{\overline{x_1} \to \overline{x_2}} \to \overline{x_3}} \to x_4\right) \times \\
& \times \left(\overline{\overline{\overline{x_1} \to \overline{x_2}} \to \overline{x_3}} \to \overline{x_4}\right)\left(\overline{\overline{x_1 \to x_2} \to x_3} \to x_4\right) \times \\
& \times \left(\overline{\overline{x_1 \to \overline{x_2}} \to x_3} \to x_4\right)\left(\overline{\overline{x_1 \to \overline{x_2}} \to x_3} \to \overline{x_4}\right) \times \\
& \times \left(\overline{\overline{x_1 \to \overline{x_2}} \to \overline{x_3}} \to x_4\right)\left(\overline{\overline{x_1 \to \overline{x_2}} \to \overline{x_3}} \to \overline{x_4}\right).
\end{aligned}
$$

(8)

Варіант 1. Двійковий еквівалент функції (8) представити її таблицею істинності (табл. 6) з наступним інвертуванням бінарних значень змінних.

Таблиця 6

Таблиця істинності функції $f\,(x_1, x_2, x_3, x_4)$

№	x_1	x_2	x_3	x_4	f	№	x_1	x_2	x_3	x_4	f
0	0	0	0	0	0	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	0
6	0	1	1	0	0	14	1	1	1	0	0
7	0	1	1	1	0	15	1	1	1	1	0

Двійковий еквівалент функції (8), згідно з першим варіантом представлення буде мати вигляд:

$$F_{\text{ДІНФ-1}} = \left|\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 5 & 0 & 1 & 0 & 1 \\ 6 & 0 & 1 & 1 & 0 \\ 7 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \end{array}\right| = \left|\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right|. \quad (9)$$

Варіант 2. Змінні двійкового еквіваленту приймають одиничне значення, якщо змінна x_i, крім x_1, у термах функції (8) представлена у прямому коді. І, навпаки, змінна двійкового еквіваленту приймає нульове значення, якщо змінна x_i, крім x_1, у термах функції (8) представлена в інверсному коді. Для x_1 у термах функції (8) умови входження до двійкового еквіваленту протилежні. Двійковий еквівалент змінної x_1 приймає одиничне значення, якщо змінна x_1 представлена в інверсному коді. І, навпаки, змінна двійкового еквіваленту приймає нульове значення, якщо змінна x_1 у термах функції (8) представлена у прямому коді (табл. 7).

Таблиця 7
Відповідність змінних x_i та x_1 ДІНФ-1 булевої функції другому варіанту двійкового еквіваленту

Змінні ДІНФ-1 функції x_i	Змінні двійкового еквіваленту
x_i	1
$\overline{x_i}$	0
$\overline{x_1}$	1
x_1	0

Двійковий еквівалент функції (8), згідно з другим варіантом представлення буде мати вигляд:

$$F_{\text{ДІНФ-1}} = \left|\begin{array}{c|cccc} 0 & 1 & 1 & 1 & 1 \\ 4 & 1 & 0 & 1 & 1 \\ 5 & 1 & 0 & 1 & 0 \\ 6 & 1 & 0 & 0 & 1 \\ 7 & 1 & 0 & 0 & 0 \\ 8 & 0 & 1 & 1 & 1 \\ 12 & 0 & 0 & 1 & 1 \\ 13 & 0 & 0 & 1 & 0 \\ 14 & 0 & 0 & 0 & 1 \\ 15 & 0 & 0 & 0 & 0 \end{array}\right|. \tag{10}$$

Двійкові еквіваленти (9) і (10) ДІНФ-1 булевої функції (8) однакові.

4. 3. ДІНФ-1.1

У (6) кон'юнкцію можна замінити імплікацією з запереченням на підставі формули (5):

$$x_1x_2 = \overline{x_1 \to \overline{\overline{x_2}}}.$$

Після застосування формули (5) функція $f\,(x_1, x_2, x_3, x_4)$ (7) з прикладу 1 буде мати вигляд:

$$f\left(x_1, x_2, x_3, x_4\right) = \overline{\varphi_1 \to \left[\varphi_2 \to \left(\varphi_3 \to \overline{\varphi_4}\right)\right]},$$

або

$$f\left(x_1, x_2, x_3, x_4\right) = \overline{\begin{array}{l} \left(\overline{\overline{\overline{x_1} \to x_2} \to x_3} \to \overline{x_4}\right) \to \\ \to \left[\begin{array}{l} \left(\overline{\overline{\overline{x_1} \to \overline{\overline{x_2}}} \to x_3} \to \overline{x_4}\right) \to \\ \to \left(\left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}} \to \overline{x_4}\right) \to \left(\overline{\overline{\overline{x_1 \to \overline{x_2}} \to \overline{x_3}} \to \overline{x_4}}\right)\right) \end{array}\right] \end{array}}.$$

Твердження 1. Будь-яка функція алгебри логіки, крім тотожної одиниці, може бути представлена у наступному вигляді:

$$f\left(x_1, x_2, \ldots, x_n\right) = \overline{\varphi_1 \to \left[\varphi_2 \to \ldots \to \left(\varphi_{n-1} \to \overline{\varphi_n}\right)\right]},$$

де

$$\varphi_i = \left(\overline{\overline{x_1^{\delta_1} \to x_2^{\overline{\overline{\delta_2}}} \to \ldots \to x_{n-1}^{\overline{\delta_{n-1}}} \to x_n^{\overline{\delta_n}}}} \right). \tag{11}$$

Присвоїмо запису (11) класифікацію досконалої імплікативної нормальної форми – 1.1 (ДІНФ-1.1) булевої функції.

Для представлення булевої функції у ДІНФ-1.1 необхідно входження всіх аргументів x_i, крім x_1, у терми ДІНФ-1.1 функції з запереченням, якщо $x_i^{\delta_i} = 1$, і без заперечення – у протилежному випадку. Для x_1 умови входження у терми ДІНФ-1.1 функції протилежні.

4. 4. ДІНФ-2

Аналогом ДДНФ є друга форма імплікативного запису булевої функції (ДІНФ-2).

Теорема 2 [18]. Будь-яка функція алгебри логіки, крім тотожного нуля, може бути представлена у наступному вигляді:

$$f\left(x_1, x_2, \ldots, x_n\right) = \overline{\bigvee_1 \left(\overline{\overline{\overline{x_1^{\delta_1} \to x_2^{\overline{\overline{\delta_2}}} \to \ldots \to x_{n-1}^{\overline{\delta_{n-1}}} \to x_n^{\overline{\delta_n}}}}} \right)}. \tag{12}$$

Доведення теореми 2 можна знайти у [18].

Тут зазначимо, що функції $\phi_i = \overline{\overline{\overline{x_1^{\delta_1} \to x_2^{\overline{\overline{\delta_2}}} \to \ldots x_n^{\overline{\delta_n}}}}}$ налаштовуються так, що $\underset{1}{}$ кожна така функція повертає «1» на наборі, що відповідає набору $< \delta_1\, \overline{\delta_2} \ldots \overline{\delta_n} >$ і повертає «0» на решту наборах.

Присвоїмо запису (12) класифікацію досконалої імплікативної нормальної форми – 2 (ДІНФ-2) булевої функції.

Входження аргументів x_i та x_1 у терми ДІНФ-2 функції є аналогічним входженню аргументів у терми ДІНФ-1.

Будь-якому бінарному набору відповідає терм ДІНФ-2 функції $\overline{\overline{\overline{x_1^{\delta_1} \to x_2^{\overline{\overline{\delta_2}}} \to \ldots \to x_{n-1}^{\overline{\delta_{n-1}}} \to x_n^{\overline{\delta_n}}}}}$ і, навпаки, – терму ДІНФ-2 функції $\overline{\overline{\overline{x_1^{\delta_1} \to x_2^{\overline{\overline{\delta_2}}} \to \ldots \to x_{n-1}^{\overline{\delta_{n-1}}} \to x_n^{\overline{\delta_n}}}}}$ відповідає бінарний набір (кортеж). Наприклад, набору <1100> відповідає терм ДІНФ-2 $\overline{\overline{\overline{x_1 \to \overline{\overline{x_2}} \to x_3} \to x_4}}$, а терму ДІНФ-2 $\overline{\overline{\overline{\overline{\overline{x_1} \to \overline{\overline{x_2}} \to x_3} \to x_4} \to \overline{x_5}}}$ відповідає набір <01001>.

Приклад 2. Функцію $f\,(x_1, x_2, x_3)$ (табл. 8) представити у ДІНФ-2.

Складемо терми ϕ_i імплікативної функції для наборів табл. 8, на яких $f(x_1, x_2, x_3)$=1:

$$\phi_1 = \overline{\overline{x_1 \to x_2} \to x_3};$$

$$\phi_2 = \overline{\overline{x_1 \to x_2} \to \overline{x_3}}.$$

Таблиця 8

Таблиця істинності логічної функції $f(x_1, x_2, x_3)$

№	x_1	x_2	x_3	$f(x_1, x_2, x_3)$	№	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0	4	1	0	0	1
1	0	0	1	0	5	1	0	1	1
2	0	1	0	0	6	1	1	0	0
3	0	1	1	0	7	1	1	1	0

Тоді запис (13)

$$f(x_1, x_2, x_3) = \left(\overline{\overline{x_1 \to x_2} \to x_3}\right) + \left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}}\right) \qquad (13)$$

буде представляти функцію $f(x_1, x_2, x_3)$ у ДІНФ-2.

4. 5. Двійковий еквівалент ДІНФ-2

Оскільки ДІНФ-2 є аналогом ДДНФ функцій булевого базису, спрощення ДІНФ-2 здійснюється за правилами спрощення ДДНФ [19, 20].

Змінні двійкового еквіваленту приймають одиничне значення, якщо змінна x_i, крім x_1, у термах функції (14) представлена в інверсному коді, і, навпаки,

$$f(x_1, x_2, x_3) = \left(\overline{\overline{x_1 \to x_2} \to x_3}\right) + \left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}}\right). \qquad (14)$$

змінна двійкового еквіваленту приймає нульове значення, якщо змінна x_i, крім x_1, представлена у прямому коді. Для x_1 у термах функції (17) умови входження до двійкового еквіваленту протилежні. Двійковий еквівалент змінної x_1 приймає одиничне значення, якщо змінна x_1, представлена у прямому коді. І, навпаки, двійковий еквівалент змінної x_1 приймає нульове значення, якщо змінна x_1 у термах функції (14) представлена в інверсному коді (табл. 9).

Таблиця 9

Відповідність змінних x_i та x_1 ДІНФ-2 булевої функції двійковому еквіваленту

Змінні ДІНФ-2 функції x_i	Змінні двійкового еквіваленту
x_i	0
$\overline{x_i}$	1
$\overline{x_1}$	0
x_1	1

Двійковий еквівалент функції (14) буде мати вигляд:

$$F_{\text{ДІНФ-2}} = \left| \begin{array}{c|ccc} 4 & 1 & 0 & 0 \\ 5 & 1 & 0 & 1 \end{array} \right| \tag{15}$$

Таким чином логічна функція (14) подається бінарною матрицею (15).

4. 6. ДІНФ-2. 1

У (12) диз'юнкцію можна замінити імплікацією з запереченням на підставі формули (4):

$$x_1 + x_2 = \overline{x_1} \to x_2.$$

Після застосування формули (4) функція $f\,(x_1, x_2, x_3)$ (13) з прикладу 2 буде мати вигляд:

$$f\left(x_1, x_2, x_3\right) = \left(\overline{x_1 \to x_2} \to x_3\right) \to \left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}}\right).$$

Твердження 2. Будь-яка функція алгебри логіки, крім тотожного нуля, може бути представлена у наступному вигляді:

$$f\left(x_1, x_2, \ldots, x_n\right) = \phi_1 \to \left[\phi_2 \to \ldots \to \left(\phi_{n-1} \to \overline{\phi_n}\right)\right],$$

де

$$\phi_i = \left(\overline{\overline{x_1^{\delta_1} \to x_2^{\overline{\delta_2}}} \to \ldots \to x_{n-1}^{\overline{\delta_{n-1}}}} \to x_n^{\overline{\delta_n}}\right). \tag{16}$$

Присвоїмо запису (16) класифікацію досконалої імплікативної нормальної форми – 2.1 (ДІНФ-2.1) булевої функції.

Для представлення булевої функції у ДІНФ-2.1 входження всіх аргументів x_i та x_1 у терми ДІНФ-2.1 функції є аналогічним входженню аргументів у терми ДІНФ-1.

Приклад 3. Представити булевий вираз $x_1\left(x_2 + x_3\right) + \overline{x_2}x_3$ в імплікативному базисі {→, NOT}.

Рішення:

$$x_1\left(x_2 + x_3\right) + \overline{x_2}x_3 = \overline{x_1\left(x_2 + x_3\right)} \to \overline{x_2}x_3 = \overline{\overline{\overline{x_1} + \overline{\left(x_2 + x_3\right)}}} \to \overline{\overline{\overline{x_2}} + \overline{x_3}} =$$

$$= \overline{x_1} + \overline{\left(x_2 + x_3\right)} \to \overline{x_2 + \overline{\overline{x_3}}} = \left(x_1 \to \overline{\left(x_2 + x_3\right)}\right) \to \left(\overline{\overline{x_2} \to \overline{x_3}}\right) =$$

$$= \left(x_1 \to \overline{\left(\overline{x_2} \to x_3\right)}\right) \to \left(\overline{\overline{x_2} \to \overline{x_3}}\right).$$

Вираз $x_1\left(x_2+x_3\right)+\overline{x_2}x_3$ представлений у базисі $\{\rightarrow, \mathrm{NOT}\}$.

5. Аксіоми і перетворення в імплікативному базисі

Для імплікації справедливі такі аксіоми:

$$\left.\begin{aligned}
&x \rightarrow x = 1;\\
&x \rightarrow \overline{x} = \overline{x};\\
&x \rightarrow 1 = 1;\\
&x \rightarrow 0 = \overline{x};\\
&x_1 \rightarrow x_2 \rightarrow x_1 = x_1.
\end{aligned}\right\} \qquad (17)$$

Справедливість представлених аксіом доводиться таблицями істинності.

З (17) випливає, що для імплікації справедливий тільки переставний закон у зміненому вигляді:

$$x_1 \rightarrow x_2 = \overline{x_2} \rightarrow \overline{x_1}.$$

Асоціативний закон для імплікації несправедливий.

Правила виконання імплікації забезпечують наступні перетворення алгебричних виразів (табл. 10).

Таблиця 10

Рівносильні перетворення в імплікативному базисі

№ з/п	Перетворення в імплікативному базисі
1	$x_1 \leftrightarrow x_2 = \left(x_1 \rightarrow x_2\right)\left(x_2 \rightarrow x_1\right) = \left(\overline{x_1} + x_2\right)\left(\overline{x_2} + x_1\right) = \left(x_1x_2\right) + \left(\overline{x_1}x_2\right)$
2	$x_1 \rightarrow x_2 = 1 \oplus x_1 \oplus x_1x_2$
3	$x_1 \leftarrow x_2 = x_1 \oplus x_1x_2$
4	$x_2 \leftarrow x_1 = \overline{x_1}x_2$
5	$x_1 \leftarrow x_2 = x_1\overline{x_2}$
6	$x_2 \rightarrow x_1 = \overline{x_1}\,\overline{x_2} + x_1\overline{x_2} + x_1x_2$
7	$x_1 \rightarrow x_2 = \overline{x_1}\,\overline{x_2} + \overline{x_1}x_2 + x_1x_2$
8	$x_1\ NOR\ x_2 = \left(\left(x_2 \rightarrow 0\right) \rightarrow x_1\right) \rightarrow 0$
9	$x_1\ NAND\ x_2 = x_2 \rightarrow \left(x_1 \rightarrow 0\right)$
10	$x_1\ XOR\ x_2 = \left\{\left(\left(x_1 \rightarrow 0\right) \rightarrow x_2\right) \rightarrow \left[\left(x_1 \rightarrow \left(x_2 \rightarrow 0\right)\right) \rightarrow 0\right]\right\} \rightarrow 0$

Перетворення табл. 10 утворюють алгебричний апарат рівносильних між базисних переходів та спрощення логічних функцій.

6. Результати мінімізації функцій імплікативного базису методом образних перетворень

Рівносильні образні перетворення при мінімізації функцій імплікативного базису дають наступний результат:

– визначають герменевтику логічних операцій на бінарних структурах функцій імплікативного базису;

– утворюють алгебру імплікативного базису у частині спрощення ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1 булевих функцій.

6. 1. Герменевтика логічних операцій в імплікативному базисі

В імплікативному базисі герменевтика логічних операцій подібна до герменевтики раніш розглянутих базисів [19].

Для подання досконалих імплікативних нормальних форм, наприклад ДІНФ-1 n-містких булевих функцій бінарним еквівалентом або матрицею потрібно змінні з інверсією $\overline{x}_n$ замінити на 1_n, а змінні без інверсії x_n – на 0_n (п. 4.2), де n – числовий індекс, який визначає розрядність символа-змінної «1» або «0» у термах функції імплікативного базису. Для змінної x_1, у термах функції імплікативного базису, умови входження до двійкового еквіваленту протилежні (п. 4. 2).

Досконалу імплікативну нормальну форму 3-місткої функції імплікативного базису

$$F = \left(\overline{\overline{x_1} \to x_2} \to x_3\right)\left(\overline{\overline{x_1} \to x_2} \to \overline{x_3}\right), \tag{18}$$

можна подати бінарними наборами (кортежами)

$$F = \left(0_1 0_2 0_3\right)\left(0_1 0_2 1_3\right), \tag{19}$$

або матрицею

$$F = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \tag{20}$$

Назвемо матрицю (20) екземпляром класу бінарних матриць функцій імплікативного базису.

Герменевтика логічних операцій для матриці (20) полягає у тому, що матриця (20) подає терми ДІНФ-1 функції імплікативного базису та операцію кон'юнкції для них. Зазначену герменевтику доцільно застосовувати при виведенні результату логічних операцій у класі бінарних матриць функцій імплікативного базису.

6. 2. Рівносильні перетвореннями у ДІНФ-1 булевих функцій імплікативного базису

У загальному випадку під час мінімізації булевих функцій імплікативного базису методом образних перетворень можливими є наступні правила алгебри логіки.

Склеювання змінних 2-містких термів ДІНФ-1 функції можна здійснювати за допомогою перетворення

$$\left(\overline{x_1} \to x_2\right)\left(\overline{x_1} \to \overline{x_2}\right) = x_1 = 1 \to x_1. \tag{21}$$

Рівносильні перетворення для правила склеювання змінних 2-містких термів ДІНФ-1 (21) мають ілюстрацію комбінаторного образу (22), де двійковий еквівалент функції (24) поданий другим варіантом його побудови (п. 4. 2).

$$\begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & \end{vmatrix} = x_1 = 1 \to x_1. \tag{22}$$

Отримана мінімальна булева функція у ДІНФ-1 має вигляд:

$$f_{\text{МІНФ-1}} = 1 \to x_1.$$

Склеювання змінних 3-містких термів ДІНФ-1 можна здійснювати за допомогою перетворення

$$\left(\overline{\overline{x_1} \to x_2} \to x_3\right)\left(\overline{\overline{x_1} \to \overline{x_2}} \to x_3\right) = \overline{x_1} \to x_3. \tag{23}$$

Рівносильні перетворення для правила склеювання змінних 3-містких термів ДІНФ-1 (23) мають ілюстрацію образу (24), де двійковий еквівалент функції (23) поданий другим варіантом його побудови (п. 4. 2).

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & & 1 \end{vmatrix} = x_1 + x_3 = \overline{x_1} \to x_3. \tag{24}$$

Склеювання змінних 4-містких термів ДІНФ-1 можна здійснювати за допомогою перетворення

$$\left(\overline{\overline{\overline{x_1} \to x_2} \to x_3} \to \overline{x_4}\right)\left(\overline{\overline{\overline{x_1} \to \overline{x_2}} \to x_3} \to \overline{x_4}\right) = \\ = \overline{\overline{x_1} \to x_3} \to \overline{x_4}. \tag{25}$$

Рівносильні перетворення для правила склеювання змінних 4-містких термів ДІНФ-1 (25) мають ілюстрацію образу (26):

$$\begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & & 1 & 0 \end{vmatrix} = \\ = x_1 + x_3 + \overline{x_4} = \overline{\overline{x_1} \to x_3} \to \overline{x_4}. \tag{26}$$

Правило супер-склеювання змінних.

Для 4-містких термів ДІНФ-1 правило супер-склеювання змінних [18] може мати такий, наприклад, вигляд:

$$\left(\overline{\overline{x_1 \to x_2} \to x_3} \to x_4\right)\left(\overline{\overline{x_1 \to x_2} \to x_3} \to \bar{x}_4\right)\left(\overline{\overline{x_1 \to x_2} \to \bar{x}_3} \to x_4\right)\left(\overline{\overline{x_1 \to x_2} \to \bar{x}_3} \to \bar{x}_4\right) =$$
$$= \left(\overline{x_1 \to x_2} \to x_3 + x_4\right)\left(\overline{x_1 \to x_2} \to x_3 + \bar{x}_4\right)\left(\overline{x_1 \to x_2} \to \bar{x}_3 + x_4\right)\left(\overline{x_1 \to x_2} \to \bar{x}_3 + \bar{x}_4\right) =$$
$$= \left(x_1 \to x_2 + x_3 + x_4\right)\left(x_1 \to x_2 + x_3 + \bar{x}_4\right)\left(x_1 \to x_2 + \bar{x}_3 + x_4\right)\left(x_1 \to x_2 + \bar{x}_3 + \bar{x}_4\right) =$$
$$= \left(\overline{x_1} + x_2 + x_3 + x_4\right)\left(\overline{x_1} + x_2 + x_3 + \bar{x}_4\right)\left(\overline{x_1} + x_2 + \bar{x}_3 + x_4\right)\left(\overline{x_1} + x_2 + \bar{x}_3 + \bar{x}_4\right) =$$
$$= \left(\overline{x_1} + x_2 + x_3\right)\left(\overline{x_1} + x_2 + x_3\right)\left(\overline{x_1} + x_2 + \bar{x}_3\right)\left(\overline{x_1} + x_2 + \bar{x}_3\right) =$$
$$= \left(\overline{x_1} + x_2\right)\left(\overline{x_1} + x_2\right)\left(\overline{x_1} + x_2\right)\left(\overline{x_1} + x_2\right) = \overline{x_1} + x_2 = x_1 \to x_2.$$

(27)

Рівносильні перетворення для правила супер-склеювання змінних 4-містких термів ДІНФ-1 (27) мають ілюстрацію образу (28):

$$\begin{vmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{vmatrix} = \overline{x_1} + x_2 = x_1 \to x_2. \tag{28}$$

У правилі (28) використовується 2-(2, 4)-design [20].

Правило неповного супер-склеювання змінних.

Комбінаторні властивості неповної комбінаторної системи з повторенням 2-(n, x/b)-design [20] забезпечують правило неповного супер-склеювання змінних в імплікативному базисі.

Для 2-містких термів ДІНФ-1 правило неповного супер-склеювання змінних може мати такий, наприклад, вигляд:

$$
\begin{aligned}
& f\left(x_1, x_2\right) = \left(\overline{x_1} \to x_2\right)\left(x_1 \to x_2\right)\left(x_1 \to \overline{x_2}\right) = \\
& = \left(x_1 + x_2\right)\left(\overline{x_1} + x_2\right)\left(\overline{x_1} + \overline{x_2}\right) = \\
& = x_2\left(\overline{x_1} + \overline{x_2}\right) = \overline{x_1} x_2 = \overline{x_1 + \overline{x_2}} = \overline{\overline{x_1} \to \overline{x_2}}.
\end{aligned}
\tag{29}
$$

Рівносильні перетворення для правила неповного супер-склеювання змінних 2-містких термів ДІНФ-1 (29) мають ілюстрацію образу (30), де двійковий еквівалент функції (32) поданий другим варіантом його побудови (п. 4. 2).

$$
\begin{vmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} & 1 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} & 1 \\ 0 & \end{vmatrix} = \overline{x_1} x_2 = \overline{x_1 + \overline{x_2}} = \overline{\overline{x_1} \to \overline{x_2}}.
\tag{30}
$$

У правилі (30) використовується 2-(2, 3/4)-design [20].

Для 3-містких термів ДІНФ-1 правило неповного супер-склеювання змінних може мати такий, наприклад, вигляд:

$$
\begin{aligned}
& \left(\overline{x_1 \to \overline{x_2}} \to \overline{x_3}\right)\left(\overline{x_1 \to \overline{x_2}} \to x_3\right)\left(\overline{\overline{x_1} \to \overline{x_2}} \to \overline{x_3}\right)\left(\overline{\overline{x_1} \to \overline{x_2}} \to x_3\right) \times \\
& \times\left(\overline{x_1 \to x_2} \to \overline{x_3}\right)\left(\overline{x_1 \to x_2} \to x_3\right)\left(\overline{\overline{x_1} \to x_2} \to \overline{x_3}\right) = \\
& = \left(x_1 \to \overline{x_2} + \overline{x_3}\right)\left(x_1 \to \overline{x_2} + x_3\right)\left(\overline{x_1} \to \overline{x_2} + \overline{x_3}\right)\left(\overline{x_1} \to \overline{x_2} + x_3\right) \times \\
& \times\left(x_1 \to x_2 + \overline{x_3}\right)\left(x_1 \to x_2 + x_3\right)\left(\overline{x_1} \to x_2 + \overline{x_3}\right) = \\
& = \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right)\left(\overline{x_1} + \overline{x_2} + x_3\right)\left(x_1 + \overline{x_2} + \overline{x_3}\right)\left(x_1 + \overline{x_2} + x_3\right) \times \\
& \times\left(\overline{x_1} + x_2 + \overline{x_3}\right)\left(\overline{x_1} + x_2 + x_3\right)\left(x_1 + x_2 + \overline{x_3}\right) = \\
& = \left(\overline{x_1} + \overline{x_2}\right)\left(\overline{x_1} + \overline{x_2}\right)\left(x_1 + \overline{x_2}\right)\left(x_1 + \overline{x_2}\right)\left(\overline{x_1} + x_2\right)\left(\overline{x_1} + x_2\right)\left(x_1 + x_2 + \overline{x_3}\right) = \\
& = \left(\overline{x_1} + \overline{x_2}\right)\left(x_1 + \overline{x_2}\right)\left(\overline{x_1} + x_2\right)\left(x_1 + x_2 + \overline{x_3}\right) = \\
& = \overline{x_2}\left(\overline{x_1} + x_2\right)\left(x_2 + \overline{x_3}\right) = \overline{x_1}\,\overline{x_2}\left(x_2 + \overline{x_3}\right) = \\
& = \overline{x_1}\,\overline{x_2}\,\overline{x_3} = \overline{x_1 + x_2 + x_3} = \overline{\overline{x_1 + x_2} \to x_3} = \overline{\overline{\overline{x_1} \to x_2} \to x_3}.
\end{aligned}
\tag{31}
$$

Рівносильні перетворення для правила неповного супер-склеювання змінних 3-містких термів ДІНФ-1 (31) мають ілюстрацію образу (32), де двійковий еквівалент функції (31) поданий другим варіантом його побудови (п. 4. 2).

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} & 0 & \\ 0 & 1 & \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} & 0 & \\ 0 & 1 & \\ & 1 & 0 \end{vmatrix} = \begin{vmatrix} & 0 & \\ 0 & & \\ & & 0 \end{vmatrix} = \tag{32}$$

$$= \bar{x}_1 \bar{x}_2 \bar{x}_3 = \overline{x_1 + x_2 + x_3} = \overline{\overline{x_1 + x_2} \to x_3} = \overline{\overline{\bar{x}_1 \to x_2} \to x_3}.$$

У правилі (32) використовується 2-(3, 7/8)-design [20].

Узагальнене склеювання змінних в імплікативному базисі можна здійснювати за допомогою перетворення

$$\left(x_1 \to \bar{x}_2\right)\left(x_1 \to x_3\right)\left(x_2 \to \bar{x}_3\right) = \left(x_1 \to x_3\right)\left(x_2 \to \bar{x}_3\right). \tag{33}$$

Рівносильні перетворення для правила узагальненого склеювання змінних (33) мають ілюстрацію образу (34), де двійковий еквівалент виразу (33) поданий другим варіантом його побудови (п. 4. 2).

$$\begin{vmatrix} 0 & 0 & \\ 0 & & 1 \\ & 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & & 1 \\ & 0 & 0 \end{vmatrix} = \left(\bar{x}_1 + x_3\right)\left(\bar{x}_2 + \bar{x}_3\right) = \left(x_1 \to x_3\right)\left(x_2 \to \bar{x}_3\right). \tag{34}$$

Інший варіант правила узагальненого склеювання змінних для ДІНФ-1:

$$\left(x_1 \to x_3\right)\left(x_2 \to \bar{x}_3\right) = \left(x_1 \to \bar{x}_2\right)\left(x_1 \to x_3\right)\left(x_2 \to \bar{x}_3\right). \tag{35}$$

$$\begin{vmatrix} 0 & & 1 \\ & 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 & \\ 0 & & 1 \\ & 0 & 0 \end{vmatrix} = \left(\bar{x}_1 + \bar{x}_2\right)\left(\bar{x}_1 + x_3\right)\left(\bar{x}_2 + \bar{x}_3\right) = \left(x_1 \to \bar{x}_2\right)\left(x_1 \to x_3\right)\left(x_2 \to \bar{x}_3\right).$$

Правило поглинання змінних зводиться до перетворень:

$$1.\ \left(\bar{x}_1 \to 0\right)\left(\bar{x}_1 \to x_2\right) = \bar{x}_1 \to 0. \tag{36}$$

Рівносильні перетворення для правила поглинання змінних ДІНФ-1 (36) мають ілюстрацію образу (37).

$$\left(\overline{x_1}\to 0\right)\left(\overline{x_1}\to x_2\right)=x_1\left(x_1+x_2\right)=\left|\begin{array}{cc}1 & \\ 1 & 1\end{array}\right|=\left|\begin{array}{cc}1 & \end{array}\right|=x_1=\overline{x_1}\to 0. \quad (37)$$

2. $\left(x_1\to 0\right)\left(x_1\to x_2\right)=x_1\to 0.$ (38)

3. $\left(x_1\to x_2\right)\left(\overline{x_1\to x_2}\to x_3\right)=x_1\to x_2.$ (39)

$$\left|\begin{array}{ccc}0 & 1 & \\ 0 & 1 & 1\end{array}\right|=\left|\begin{array}{ccc}0 & 1 & \end{array}\right|=\overline{x_1}+x_2=x_1\to x_2.$$

4. $\left(x_1\to x_2\right)\left(\overline{\overline{x_1\to x_2}\to x_3}\to x_4\right)=x_1\to x_2.$ (40)

$$\left|\begin{array}{cccc}0 & 1 & & \\ 0 & 1 & 1 & 1\end{array}\right|=\left|\begin{array}{cccc}0 & 1 & & \end{array}\right|=\overline{x_1}+x_2=x_1\to x_2.$$

Правило напівсклеювання змінних в імплікативному базисі можна здійснювати за допомогою наступних перетворень:

$$\left(\overline{x_1}\to x_2\right)\left(\overline{x_1\to x_2}\to x_3\right)=\left(\overline{x_1}\to x_2\right)\left(\overline{x_2}\to x_3\right). \quad (41)$$

Правило напівсклеювання змінних (41) має ілюстрацію образу (42):

$$\left|\begin{array}{ccc}1 & 1 & \\ 0 & 1 & 1\end{array}\right|=\left|\begin{array}{ccc}1 & 1 & \\ & 1 & 1\end{array}\right|=$$
$$=\left(x_1+x_2\right)\left(x_2+x_3\right)=\left(\overline{x_1}\to x_2\right)\left(\overline{x_2}\to x_3\right). \quad (42)$$

Правило

$$\left(\overline{x_1}\to 0\right)\left(x_1\to x_2\right)=\overline{x_1\to\overline{x_2}}, \quad (43)$$

доводиться наступними перетвореннями:

$$\left(\overline{x_1}\to 0\right)\left(x_1\to x_2\right)=x_1\left(\overline{x_1}+x_2\right)=x_1x_2=\overline{\overline{x_1}+\overline{x_2}}=\overline{x_1\to\overline{x_2}}.$$

$$x_1\left(\overline{x_1}+x_2\right)=\left|\begin{array}{cc}1 & \\ 0 & 1\end{array}\right|=\left|\begin{array}{cc}1 & \\ & 1\end{array}\right|=x_1x_2=\overline{\overline{x_1}+\overline{x_2}}=\overline{x_1\to\overline{x_2}}. \quad (44)$$

Перетворення результату образних перетворень (44) – x_1x_2 до представлення його в імплікативному базисі здійснюється за допомогою формули де Моргана.

Приклад 4. Спростити логічну функцію $f(x_1, x_2, x_3, x_4)$ (табл. 11) у досконалій імплікативній нормальній формі –1 (ДІНФ-1).

Таблиця 11

Таблиця істинності функції $f\,(x_1, x_2, x_3, x_4)$

№	x_1	x_2	x_3	x_4	$f\,(x_1, x_2, x_3, x_4)$	№	x_1	x_2	x_3	x_4	$f\,(x_1, x_2, x_3, x_4)$
0	0	0	0	0	1	8	1	0	0	0	1
1	0	0	0	1	1	9	1	0	0	1	0
2	0	0	1	0	0	10	1	0	1	0	0
3	0	0	1	1	0	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	0
6	0	1	1	0	1	14	1	1	1	0	1
7	0	1	1	1	0	15	1	1	1	1	1

Застосовуємо перший варіант бінарного еквіваленту ДІНФ-1 функції $f(x_1, x_2, x_3, x_4)$ (п. 4. 2). Мінімізація ДІНФ-1 $f(x_1, x_2, x_3, x_4)$ проводиться наступними образними перетвореннями:

$$F_{\text{МІНФ-1}} = \left|\begin{array}{c|cccc} 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 1 \\ 4 & 0 & 1 & 0 & 0 \\ 5 & 0 & 1 & 0 & 1 \\ 7 & 0 & 1 & 1 & 1 \\ 9 & 1 & 0 & 0 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \end{array}\right| = \left|\begin{array}{cccc} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right| = \left|\begin{array}{cccc} & 1 & 0 & 1 \\ 1 & & 0 & 0 \\ & 0 & 1 & \\ 0 & 1 & 1 & 0 \end{array}\right| = \left|\begin{array}{cccc} & 1 & 0 & 1 \\ 1 & & 0 & 0 \\ & 0 & 1 & \\ 0 & & 1 & 0 \end{array}\right| =$$

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

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

$$= \left(\overline{\overline{x_2} \to \overline{x_3}} \to x_4\right)\left(\overline{\overline{x_1} \to \overline{x_3}} \to \overline{x_4}\right)\left(x_2 \to x_3\right)\left(\overline{x_1 \to x_3} \to \overline{x_4}\right).$$

МІНФ-1 функції $f(x_1,x_2,x_3,x_4)$ (табл. 11):

$$F_{\text{МІНФ-1}} = \left(\overline{\overline{x_2} \to \overline{x_3}} \to x_4\right)\left(\overline{\overline{x_1} \to \overline{x_3}} \to \overline{x_4}\right)\left(x_2 \to x_3\right)\left(\overline{x_1 \to x_3} \to \overline{x_4}\right). \quad (45)$$

6. 3. Рівносильні перетвореннями у ДІНФ-1.1 булевих функцій імплікативного базису

Перед проведенням операції склеюванням змінних 4-містких термів ДІНФ-1. 1, проведемо склеюванням змінних 4-містких термів ДІНФ-1:

$$\begin{aligned}
&f\left(x_1, x_2, x_3, x_4\right) = \\
&= \left(\overline{\overline{\overline{x_1} \to x_2} \to x_3} \to \overline{x_4}\right)\left(\overline{\overline{\overline{x_1} \to \overline{x_2}} \to x_3} \to \overline{x_4}\right) \times \\
&\times \left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}} \to \overline{x_4}\right)\left(\overline{\overline{x_1 \to \overline{x_2}} \to \overline{x_3}} \to \overline{x_4}\right) = \\
&= \left(\overline{\overline{x_1} \to x_2} \to x_3 + \overline{x_4}\right)\left(\overline{\overline{x_1} \to \overline{x_2}} \to x_3 + \overline{x_4}\right) \times \\
&\times \left(\overline{x_1 \to x_2} \to \overline{x_3} + \overline{x_4}\right)\left(\overline{x_1 \to \overline{x_2}} \to \overline{x_3} + \overline{x_4}\right) = \\
&= \left(\overline{x_1} \to x_2 + x_3 + \overline{x_4}\right)\left(\overline{x_1} \to \overline{x_2} + x_3 + \overline{x_4}\right) \times \\
&\times \left(x_1 \to x_2 + \overline{x_3} + \overline{x_4}\right)\left(x_1 \to \overline{x_2} + \overline{x_3} + \overline{x_4}\right) = \\
&= \left(x_1 + x_2 + x_3 + \overline{x_4}\right)\left(x_1 + \overline{x_2} + x_3 + \overline{x_4}\right) \times \\
&\times \left(\overline{x_1} + x_2 + \overline{x_3} + \overline{x_4}\right)\left(\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4}\right) = \\
&= \left(x_1 + x_3 + \overline{x_4}\right)\left(\overline{x_1} + \overline{x_3} + \overline{x_4}\right) = \\
&= \left(\overline{\overline{x_1} \to x_3} \to \overline{x_4}\right)\left(\overline{x_1 \to \overline{x_3}} \to \overline{x_4}\right).
\end{aligned} \tag{46}$$

У (46) проведемо заміну кон'юнкції на імплікацією з запереченням на підставі формули (5),

$$x_1 x_2 = \overline{x_1 \to \overline{x_2}}.$$

після чого проведемо операцію склеюванням змінних 4-містких термів ДІНФ-1. 1.

$$f\left(x_1,x_2,x_3,x_4\right)=$$

$$=\overline{\left(\overline{\overline{\overline{x_1}}\to x_2\to x_3\to\overline{x_4}}\right)\to\left[\left(\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3\to\overline{x_4}}\right)\to\left(\begin{array}{c}\left(\overline{\overline{x_1\to x_2}\to\overline{x_3}\to\overline{x_4}}\right)\to\\ \to\left(\overline{\overline{x_1\to\overline{x_2}}\to\overline{x_3}\to\overline{x_4}}\right)\end{array}\right)\right]}=$$

$$=\overline{\left(\overline{\overline{\overline{x_1}}\to x_2\to x_3\to\overline{x_4}}\right)\to\left[\left(\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3\to\overline{x_4}}\right)\to\left(\begin{array}{c}\overline{\overline{\overline{x_1\to x_2}\to\overline{x_3}}\to\overline{x_4}}+\\ \overline{+\left(\overline{\overline{x_1\to\overline{x_2}}\to\overline{x_3}\to\overline{x_4}}\right)}\end{array}\right)\right]}=$$

$$=\overline{\left(\overline{\overline{\overline{x_1}}\to x_2\to x_3\to\overline{x_4}}\right)\to\left[\begin{array}{l}\left(\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3\to\overline{x_4}}\right)+\overline{\overline{\overline{x_1\to x_2}\to\overline{x_3}}\to\overline{x_4}}+\\ +\overline{\overline{\overline{x_1\to\overline{x_2}}\to\overline{x_3}}\to\overline{x_4}}\end{array}\right]}=$$

$$=\overline{\begin{array}{l}\overline{\overline{\overline{\overline{x_1}}\to x_2}\to x_3\to\overline{x_4}}+\overline{\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3}\to\overline{x_4}}+\\ +\overline{\overline{\overline{x_1\to x_2}\to\overline{x_3}}\to\overline{x_4}}+\overline{\overline{\overline{x_1\to\overline{x_2}}\to\overline{x_3}}\to\overline{x_4}}\end{array}}=$$

$$=\overline{\begin{array}{l}\overline{\overline{\overline{\overline{x_1}}+x_2}\to x_3\to\overline{x_4}}+\overline{\overline{\overline{\overline{x_1}+\overline{x_2}}\to x_3}\to\overline{x_4}}+\\ +\overline{\overline{\overline{\overline{x_1}+x_2}\to\overline{x_3}}\to\overline{x_4}}+\overline{\overline{\overline{\overline{x_1}+\overline{x_2}}\to\overline{x_3}}\to\overline{x_4}}\end{array}}=$$

$$=\overline{\begin{array}{l}\overline{\overline{\overline{x_1}+x_2+x_3}\to\overline{x_4}}+\overline{\overline{x_1}+\overline{x_2}+x_3\to\overline{x_4}}+\\ +\overline{\overline{\overline{x_1}+x_2+\overline{x_3}}\to\overline{x_4}}+\overline{\overline{x_1}+\overline{x_2}+\overline{x_3}\to\overline{x_4}}\end{array}}=\overline{\begin{array}{l}\overline{x_1+x_2+x_3+\overline{x_4}}+\overline{x_1+\overline{x_2}+x_3+\overline{x_4}}+\\ +\overline{\overline{x_1}+x_2+\overline{x_3}+\overline{x_4}}+\overline{\overline{x_1}+\overline{x_2}+\overline{x_3}+\overline{x_4}}\end{array}}=$$

$$=\overline{\overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4+\overline{x_1}x_2\overline{x_3}x_4+x_1\overline{x_2}x_3x_4+x_1x_2x_3x_4}=$$

$$=\overline{\overline{x_1}\,\overline{x_3}x_4+x_1x_3x_4}=\overline{\overline{\overline{x_1}\,\overline{x_3}x_4}\to x_1x_3x_4}=\overline{\left(x_1+x_3+\overline{x_4}\right)\to\left(\overline{\overline{x_1}+\overline{x_3}+\overline{x_4}}\right)}=$$

$$=\overline{\left(\overline{x_1}\to x_3+\overline{x_4}\right)\to\left(\overline{x_1\to\overline{x_3}+\overline{x_4}}\right)}=\overline{\left(\overline{\overline{x_1}\to x_3}\to\overline{x_4}\right)\to\left(\overline{\overline{x_1\to\overline{x_3}}\to\overline{x_4}}\right)}.\qquad(47)$$

Перетворення (47) має ілюстрацію образу (48):

$$f\left(x_1,x_2,x_3,x_4\right)=\begin{vmatrix}0&0&0&1\\0&1&0&1\\1&0&1&1\\1&1&1&1\end{vmatrix}=\begin{vmatrix}1&1&1&0\\1&0&1&0\\0&1&0&0\\0&0&0&0\end{vmatrix}=\begin{vmatrix}1&&1&0\\0&&0&0\end{vmatrix}=$$

$$=\left(x_1+x_3+\overline{x_4}\right)\left(\overline{x_1}+\overline{x_3}+\overline{x_4}\right)= \tag{48}$$

$$=\overline{\left(\overline{\overline{x_1}\to x_3}\to\overline{x_4}\right)\to\left(\overline{\overline{x_1\to\overline{x_3}}\to\overline{x_4}}\right)}.$$

Образні перетворення (48) мають свою герменевтику: терми функції імплікативного базису об’єднуються операцією імплікації «→», останній терм інвертується, інвертується весь отриманий вираз.

6. 4. Рівносильні перетвореннями у ДІНФ-2 булевих функцій імплікативного базису

Склеювання змінних 4-містких термів ДІНФ-2.

$$\left(\overline{\overline{\overline{\overline{x_1}\to x_2}\to x_3}\to\overline{x_4}}\right)+\left(\overline{\overline{\overline{\overline{x_1}\to\overline{x_2}}\to x_3}\to\overline{x_4}}\right)=\overline{\overline{\overline{x_1}\to x_3}\to\overline{x_4}}. \tag{49}$$

Рівносильні перетворення для правила склеювання змінних 4-містких термів ДІНФ-2 (49) мають ілюстрацію образу (50):

$$\begin{vmatrix}0&0&0&1\\0&1&0&1\end{vmatrix}=\begin{vmatrix}0&&0&1\end{vmatrix}=\overline{x_1}\,\overline{x_3}x_4=\overline{x_1+x_3+\overline{x_4}}=\overline{\overline{\overline{x_1}\to x_3}\to\overline{x_4}}. \tag{50}$$

Оскільки ДІНФ-2 є аналогом ДДНФ функції булевого базису, образні перетворення у бінарній матриці імплікативної функції (45) здійснюються за правилами ДДНФ [19, 20].

Правило супер-склеювання змінних.

Для 4-містких термів ДІНФ-2 правило супер-склеювання змінних [20] може мати такий, наприклад, вигляд:

$$
\begin{aligned}
&\left(\overline{\overline{\overline{x_1 \to x_2} \to x_3} \to x_4}\right) + \left(\overline{\overline{\overline{x_1 \to x_2} \to x_3} \to \bar{x}_4}\right) + \\
&+\left(\overline{\overline{\overline{x_1 \to x_2} \to \bar{x}_3} \to x_4}\right) + \left(\overline{\overline{\overline{x_1 \to x_2} \to \bar{x}_3} \to \bar{x}_4}\right) = \\
&=\left(\overline{\overline{x_1 \to x_2} \to x_3 + x_4}\right) + \left(\overline{\overline{x_1 \to x_2} \to x_3 + \bar{x}_4}\right) + \\
&+\left(\overline{\overline{x_1 \to x_2} \to \bar{x}_3 + x_4}\right) + \left(\overline{\overline{x_1 \to x_2} \to \bar{x}_3 + \bar{x}_4}\right) = \\
&=\left(\overline{x_1 \to x_2 + x_3 + x_4}\right) + \left(\overline{x_1 \to x_2 + x_3 + \bar{x}_4}\right) + \\
&+\left(\overline{x_1 \to x_2 + \bar{x}_3 + x_4}\right) + \left(\overline{x_1 \to x_2 + \bar{x}_3 + \bar{x}_4}\right) = \\
&=\left(\overline{\overline{x_1} + x_2 + x_3 + x_4}\right) + \left(\overline{\overline{x_1} + x_2 + x_3 + \bar{x}_4}\right) + \\
&+\left(\overline{\overline{x_1} + x_2 + \bar{x}_3 + x_4}\right) + \left(\overline{\overline{x_1} + x_2 + \bar{x}_3 + \bar{x}_4}\right) = \\
&= x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4} + x_1\overline{x_2}\,\overline{x_3}x_4 + x_1\overline{x_2}x_3\overline{x_4} + x_1\overline{x_2}x_3x_4 = \\
&= x_1\overline{x_2}\left(\overline{x_3}\,\overline{x_4} + \overline{x_3}x_4 + x_3\overline{x_4} + x_3x_4\right) = \\
&= x_1\overline{x_2} = \overline{\overline{x_1} + x_2} = \overline{x_1 \to x_2}.
\end{aligned}
\tag{51}
$$

Рівносильні перетворення для правила супер-склеювання змінних 4-містких термів ДІНФ-2 (51) мають ілюстрацію образу (52):

$$
\begin{vmatrix}
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1
\end{vmatrix}
= x_1\overline{x_2} = \overline{\overline{x_1} + x_2} = \overline{x_1 \to x_2}.
\tag{52}
$$

У правилі (52) використовується 2-(2, 4)-design [20].

ДІНФ-2 є аналогом ДДНФ функції булевого базису, тому рівносильні перетворення у бінарній матриці імплікативної функції (52) здійснюються за правилами ДДНФ [19, 20].

Інші логічні операції у ДІНФ-2 булевих функцій імплікативного базису здійснюються подібно до розглянутих операцій (50), (52).

Оскільки комбінаторна структура таблиць істинності логічних функцій імплікативного базису дає більше інформації стосовно ортогональності, суміжності, однозначності блоків таблиці істинності, застосування комбінаторних образів для пошуку об’єктів рівносильного перетворення при спрощенні функцій імплікативного базису є ефективним.

6. 5. Рівносильні перетвореннями у ДІНФ-2.1 булевих функцій імплікативного базису

Перед проведенням операції склеюванням змінних 3-містких термів ДІНФ-2. 1, проведемо склеюванням змінних 3-містких термів ДІНФ-2:

$$
\begin{aligned}
f(x_1,x_2,x_3) &= \left(\overline{\overline{x_1 \to x_2} \to x_3}\right) + \left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}}\right) = \\
&= \left(\overline{x_1 \to x_2 + x_3}\right) + \left(\overline{x_1 \to x_2 + \overline{x_3}}\right) = \\
&= \overline{\overline{x_1} + x_2 + x_3} + \overline{\overline{x_1} + x_2 + \overline{x_3}} = x_1\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}x_3 = \\
&= x_1\overline{x_2} = \overline{\overline{x_1} + x_2} = \overline{x_1 \to x_2}.
\end{aligned}
\tag{53}
$$

$$
\begin{aligned}
f(x_1,x_2,x_3) &= \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & \end{vmatrix} = \\
&= x_1\overline{x_2} = \overline{\overline{x_1} + x_2} = \overline{x_1 \to x_2}.
\end{aligned}
$$

У (53) зробимо заміну диз'юнкції на імплікацією на підставі формули (5):

$$x_1 + x_2 = \overline{x_1} \to x_2,$$

після чого проведемо склеюванням змінних 3-містких термів ДІНФ-2.1.

$$
\begin{aligned}
f(x_1,x_2,x_3) &= \left(\overline{x_1 \to x_2} \to x_3\right) \to \left(\overline{\overline{x_1 \to x_2} \to \overline{x_3}}\right) = \\
&= \overline{\overline{x_1 \to x_2} \to x_3} + \overline{\overline{x_1 \to x_2} \to \overline{x_3}} = \\
&= \overline{x_1 \to x_2 + x_3} + \overline{x_1 \to x_2 + \overline{x_3}} = \\
&= \overline{\overline{x_1} + x_2 + x_3} + \overline{\overline{x_1} + x_2 + \overline{x_3}} = \\
&= x_1\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}x_3 = x_1\overline{x_2} = \overline{\overline{x_1} + x_2} = \overline{x_1 \to x_2}.
\end{aligned}
\tag{54}
$$

Результати склеювання 3-містких термів ДІНФ-2 (53) і ДІНФ-2.1 (54) співпадають.

Перетворення (54) має ілюстрацію образу (55):

$$
f(x_1,x_2,x_3) = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & \end{vmatrix} = x_1\overline{x_2} = \overline{\overline{x_1} + x_2} = \overline{x_1 \to x_2}. \tag{55}
$$

Приклад 5. Мінімізувати систему рівнянь 1-bit повного суматора бінарних кодів (табл. 12) в імплікативному базисі [7].

Таблиця 12

Таблиця істинності 1-bit суматора бінарних кодів

№	x_i	y_i	$p_{i\text{-}1}$	s_i	p_i
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

Споглядаючи табл. 12 бачимо, що всюди, крім наборів <0,0,0> і <1,1,1>, має місце співвідношення $s_i = \overline{p_i}$. Складемо таблицю істинності, в якій чотири аргументи x_i, y_i, $p_{i\text{-}1}$, p_i і одна функція s_i (табл. 13) [21].

Таблиця 13

Таблиця істинності, в якій чотири аргументи x_i, y_i, $p_{i\text{-}1}$, p_i і одна функція s_i

№	x_i	y_i	$p_{i\text{-}1}$	p_i	s_i
0	0	0	0	0	0
1	0	0	0	1	*
2	0	0	1	0	1
3	0	0	1	1	*
4	0	1	0	0	1
5	0	1	0	1	*
6	0	1	1	0	*
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	*
10	1	0	1	0	*
11	1	0	1	1	0
12	1	1	0	0	*
13	1	1	0	1	0
14	1	1	1	0	*
15	1	1	1	1	1

Довизначемо функцію s_i (табл.14).

Таблиця 14

Таблиця істинності довизначеної функція s_i

№	x_i	y_i	$p_{i\text{-}1}$	p_i	s_i
0	0	0	0	0	
1	0	0	0	1	

2	0	0	1	0	1
3	0	0	1	1	
4	0	1	0	0	1
5	0	1	0	1	
6	0	1	1	0	1
7	0	1	1	1	
8	1	0	0	0	1
9	1	0	0	1	
10	1	0	1	0	1
11	1	0	1	1	
12	1	1	0	0	1
13	1	1	0	1	
14	1	1	1	0	1
15	1	1	1	1	1

Методом образних перетворень проведемо мінімізацію довизначеної функції s_i, замінивши змінні: $x_i - x_1$, $y_i - x_2$, $p_{i\text{-}1} - x_3$, $p_i - x_4$.

$$s_i = \begin{array}{|c|cccc|}\hline \text{№} & x_1 & x_2 & x_3 & x_4 \\ \hline 2 & 0 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 10 & 1 & 0 & 1 & 0 \\ 12 & 1 & 1 & 0 & 0 \\ 14 & 1 & 1 & 1 & 0 \\ 15 & 1 & 1 & 1 & 1 \\ \hline \end{array} = \begin{vmatrix} & & 1 & 0 \\ & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & & 1 & 0 \\ & 1 & 0 & 0 \\ 1 & & 0 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} & & 1 & 0 \\ & 1 & & 0 \\ 1 & & & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} =$$

$$= \begin{vmatrix} & & 1 & 0 \\ & 1 & & 0 \\ 1 & & & 0 \\ 1 & 1 & 1 & \end{vmatrix} = \left(x_1 + x_2 + x_3\right)\overline{x_4} + x_1x_2x_3 = \overline{\left(x_1 + x_2 + x_3\right)\overline{x_4}} \to x_1x_2x_3 =$$

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

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

$$= \left(\left(\overline{\overline{x_1} \to x_2} \to x_3\right) \to x_4\right) \to \left(\overline{\overline{x_1 \to \overline{x_2}} \to \overline{x_3}}\right).$$

Мінімальна довизначена функція s_i в імплікативному базисі (табл. 14):

$$s_i = \left(\left(\overline{\overline{x_1} \to x_2} \to x_3\right) \to x_4\right) \to \overline{\left(\overline{x_1 \to \overline{x_2}} \to \overline{x_3}\right)}. \qquad (56)$$

Змінна x_4 (p_i) для імплікативної функції (56) є фіктивною, оскільки булева похідна довизначеної функції s_i по змінній x_4 дорівнює нулю.

Мінімальна довизначена функція s_i у булевому базисі має вигляд:

$$s_i = (x_1 + x_2 + x_3)\overline{x_4} + x_1 x_2 x_3.$$

Обчислення булевої похідної функції s_i по змінній x_4 має вигляд:

$$\frac{\partial s_i}{\partial x_4} = \left((x_1 + x_2 + x_3)\overline{1} + x_1 x_2 x_3\right) \oplus \left((x_1 + x_2 + x_3)\overline{0} + x_1 x_2 x_3\right) =$$
$$= \left((x_1 + x_2 + x_3)0 + x_1 x_2 x_3\right) \oplus \left((x_1 + x_2 + x_3)1 + x_1 x_2 x_3\right) =$$
$$= x_1 x_2 x_3 \oplus (x_1 + x_2 + x_3 + x_1 x_2 x_3) = x_1 x_2 x_3 \oplus (x_1 + x_2 + x_3) =$$
$$= x_1 x_2 x_3 \cdot \left(\overline{x_1 + x_2 + x_3}\right) + \overline{x_1 x_2 x_3} \cdot (x_1 + x_2 + x_3) =$$
$$= x_1 x_2 x_3 \cdot \overline{x_1}\,\overline{x_2}\,\overline{x_3} + \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) \cdot (x_1 + x_2 + x_3) =$$
$$= \left(\overline{x_1} + \overline{x_2} + \overline{x_3}\right) \cdot (x_1 + x_2 + x_3) = 0.$$

Оскільки змінна x_4 довизначеної функції s_i є фіктивною, реалізація функції s_i буде мати вигляд:

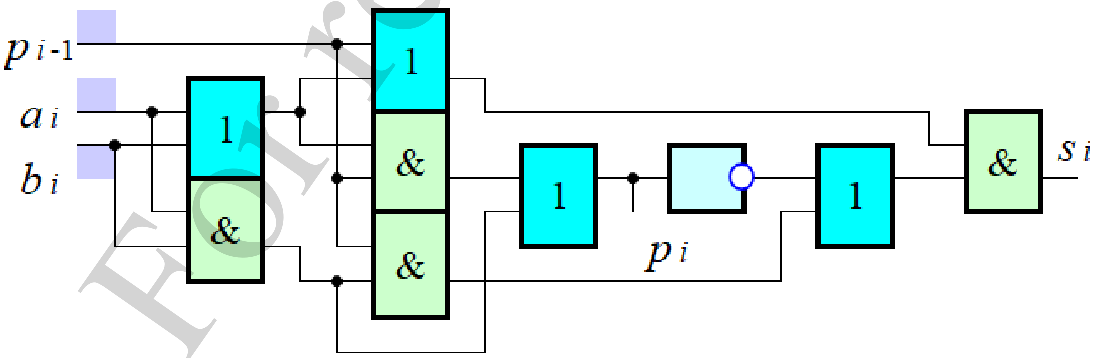

Рис. 3. 1-bit суматор бінарних кодів

Функція p_i в імплікативному базисі має вигляд:

$$p_i = a_i b_i + (a_i + b_i)\,p_{i-1} = \overline{a_i b_i} \to \left((a_i + b_i)\,p_{i-1}\right) = (\overline{a}_i + \overline{b}_i) \to \left((a_i + b_i)\,p_{i-1}\right) =$$
$$= \left(a_i \to \overline{b}_i\right) \to \left((a_i + b_i)\,p_{i-1}\right) = \left(a_i \to \overline{b}_i\right) \to \left(\overline{\overline{(a_i + b_i)} + \overline{p_{i-1}}}\right) =$$
$$= \left(a_i \to \overline{b}_i\right) \to \overline{\left((a_i + b_i) \to \overline{p_{i-1}}\right)} = \left(a_i \to \overline{b}_i\right) \to \overline{\left((\overline{a}_i \to b_i) \to \overline{p_{i-1}}\right)}.$$

Функції S і C_{OUT} 1-bit суматора в імплікативному базисі [7] мають такий вигляд:

$$S = \left[(\overline{a} \to b) \to \left((a \to \overline{b}) \to c\right)\right] \to \left(\overline{\left(\overline{a \oplus b}\right) \to \overline{c}}\right); \tag{57}$$

$$C_{out} = \left[\overline{(\overline{a} \to b) \to \left((a \to \overline{b}) \to c\right)}\right]. \tag{58}$$

Рівняння суми s_i (56), порівняно з рівнянням (57), містить на один літерал менше.

Приклад 6. Застосовуючи рівносильні перетворення для булевої функції в імплікативному базисі $f = \left(\overline{x}_1 \to \overline{x}_2\right) \to \left(x_2 x_3 \to \overline{x}_1 x_3\right)$, знайти мінімальну ДНФ [22].

Рішення:

$$f = \left(\overline{x}_1 \to \overline{x}_2\right) \to \left(x_2 x_3 \to \overline{x}_1 x_3\right) = \left(x_1 + \overline{x}_2\right) \to \left(\overline{x_2 x_3} + \overline{x}_1 x_3\right) =$$
$$= \left(\overline{x_1 + \overline{x}_2}\right) + \left(\left(\overline{x}_2 + \overline{x}_3\right) + \overline{x}_1 x_3\right) = \overline{x}_1 x_2 + \overline{x}_2 + \overline{x}_3 + \overline{x}_1 x_3 =$$
$$= \overline{x}_1 x_2 + \overline{x}_1 x_3 + \overline{x}_2\left(x_1 + \overline{x}_1\right) + \overline{x}_3\left(x_1 + \overline{x}_1\right) =$$
$$= \overline{x}_1 x_2 + \overline{x}_1 x_3 + x_1 \overline{x}_2 + \overline{x}_1 \overline{x}_2 + x_1 \overline{x}_3 + \overline{x}_1 \overline{x}_3 =$$
$$= \overline{x}_1\left(x_2 + \overline{x}_2\right) + \overline{x}_1\left(x_3 + \overline{x}_3\right) + x_1 \overline{x}_2 + x_1 \overline{x}_3 =$$
$$= \overline{x}_1 + x_1\left(\overline{x}_2 + \overline{x}_3\right) = \left(x_1 + \overline{x}_1\right)\left(\overline{x}_1 + \overline{x}_2 + \overline{x}_3\right) = \overline{x}_1 + \overline{x}_2 + \overline{x}_3.$$

Пошук мінімальної ДНФ [22] образними перетвореннями має вигляд:

$$f = \left(\overline{x_1} \to \overline{x_2}\right) \to \left(x_2 x_3 \to \overline{x_1} x_3\right) = \left(x_1 + \overline{x_2}\right) \to \left(\overline{x_2 x_3} + \overline{x_1} x_3\right) =$$

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

$$= \left| \begin{array}{ccc} 0 & 1 & \\ & 0 & \\ & & 0 \\ 0 & & 1 \end{array} \right| = \left| \begin{array}{ccc} 0 & & \\ & 0 & \\ & & 0 \\ 0 & & \end{array} \right| = \left| \begin{array}{ccc} 0 & & \\ & 0 & \\ & & 0 \end{array} \right| = \overline{x_1} + \overline{x_2} + \overline{x_3}.$$

Результат спрощення двома методами однаковий, однак метод образних перетворень є простішим.

8. Обговорення результатів мінімізації функцій імплікативного базису методом образних перетворень

Математичний апарат образних перетворень розглянуто у роботах [3, 19, 20, 23] та ін. Тут подано вербальне і образне представлення інформації, протоколи образних перетворень, нові логічні операції, ознака мінімальної логічної функції, переваги мінімізації булевих функцій на повній таблиці істинності, контролюючі властивості методу, алгоритм аналітичного методу та його автоматизація, поширення методу образних перетворень на логічні базиси. Герменевтика логічних операцій на бінарних структурах забезпечує дидактику спрощення булевих функцій, у тому числі і для класу досконалих імплікативних нормальних форм.

Алгебра, що створена, у складі правил спрощення імплікативних функцій з ілюстрацією образних перетворень логічних процедур, дозволяє впроваджувати метод образних перетворень для мінімізації булевих функцій до імпікативного базису.

Таким чином, метод стає альтернативою для технології проектування обчислювальних компонентів на основі імплікативних функцій, оскільки без алгебри спрощення логічних функцій в імплікативному базисі, залишається оптимізація функцій у булевому базисі. І тільки після такої мінімізації логічних функцій стає можливим спеціальними алгоритми проводити заміну елементів основного базису {AND, OR, NOT}, на елементи імплікативного базису {→, NOT} або {→, 0}. Однак такому підходу властиві вербальні процедури, які можуть не виявляти логічні операції, і, таким чином, зменшувати можливості аналітичного методу. Рівносильні перетворення комбінаторними образами за своїми властивостями мають більшу інформаційну ємність, і, отже, спроможні з ефектом замінити вербальні процедури алгебричних перетворень.

Об’єктом вирішення задачі спрощення булевих функцій в імплікативному базисі методом образних перетворень є бінарні структури з повторенням, якими є власне таблиці істинності заданих функцій. Це дозволяє обійтись без допоміжних об’єктів, як то карти Карно, карти Махоні, діаграми Вейча, ациклічний граф, ненаправлений граф, таблиці покриття, куби та ін.

При спрощенні логічних формул не завжди очевидно, який із законів алгебри логіки необхідно застосувати на тому чи іншому кроці. Наглядні комбінаторні структури бінарних матриць та уніфікація оригінальних процедур до певної міри дозволяють вирішувати цю проблему.

Особливістю розглянутого методу спрощення логічних виразів у ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1 є використання аналогів досконалих форм ДДНФ та ДКНФ представлення булевих функцій. Зазначені форми булевих функцій визначають правила перетворення на бінарних структурах функцій імплікативного базису.

Наглядна структура образних перетворень дозволяє здійснювати ручний спосіб спрощення функцій імплікативного базису (з використанням математичного редактора, наприклад MathType v. 7.0) орієнтовно у межах до десяти вхідних змінних.

Застосування методу образних перетворень для мінімізації функцій імплікативного базису виводить проблему спрощення ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1 на рівень добре дослідженої задачі у класі диз'юнктивно-кон'юнктивних нормальних форм (ДКНФ) булевих функцій, а також у класі досконалих нормальних форм функцій алгебри Шеффера (ДНФШ-1 та ДНФШ-2).

Алгебра, що створена для перетворення функцій в імплікативному базисі, представлена наступними логічними операціями (табл. 15):

Таблиця 15
Логічні операції в імплікативному базисі

№ з/п	Назва логічної операції	Номер посилання у тексті	Форма представлення
1	Склеювання змінних	(21), (23), (25), (46)	ДІНФ-1
2	Супер-склеювання змінних	(27)	ДІНФ-1
3	Неповне супер-склеювання змінних	(29), (31)	ДІНФ-1
4	Узагальнене склеювання змінних	(33), (35)	ДІНФ-1
5	Поглинання змінних	(36), (38), (39), (40)	ДІНФ-1
6	Напівсклеювання змінних	(41)	ДІНФ-1
7	Правило без назви	(43)	ДІНФ-1
8	Склеювання змінних	(47)	ДІНФ-1.1
9	Склеювання змінних	(49), (53)	ДІНФ-2
10	Супер-склеювання змінних	(51)	ДІНФ-2
11	Склеювання змінних	(54)	ДІНФ-2.1

Обмеженнями застосування методу образних перетворень є випадки, коли перемикальна функція представлена у змішаному базисі. У цьому випадку функцію необхідно представити одним логічним базисом.

Слабка сторона розглянутого методу полягає у малому практичному застосуванні рівносильних образних перетворень для процесу мінімізації функцій імплікативного базису з подальшим виготовленням відповідних обчислювальних компонентів. Негативні внутрішні фактори методу пов'язані з додатковими часовими витратами на встановлення протоколів спрощення функцій імплікативного базису з подальшим створенням бібліотеки правил алгебри логіки, що мають ілюстрацію відповідних образних перетворень. Перспективою подальших досліджень може бути, наприклад, використання методу для мінімізації булевих функцій у класі змішаних базисів.

9. Висновки

1. Встановлено, що спрощення булевих функцій імплікативного базису методом образних перетворень ґрунтується на блок-схемі з повторенням, якою є власне таблиця істинності заданої функції. Це дозволяє зосередити принцип спрощення у межах таблиці істинності функції і, таким чином, обійтись без допоміжних об'єктів, як то таблиці покриття, карти Карно, діаграми Вейча, ациклічний граф, кубічне представлення та ін.

Досконалу нормальну форму *n*-місткої функції імплікативного базису можна подати бінарними наборами (19) або матрицею (20), яка у цьому випадку буде подавати терми функції імплікації та операцію кон'юнкції для них. Така герменевтика має ефективне застосовувати при спрощенні логічних функцій та при виведенні результату логічних операцій у класі бінарних матриць функцій імплікативного базису.

2. Для належного спрощення функцій імплікативного базису методом образних перетворень була розроблена алгебра імплікативного базису у частині правил спрощення ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1 булевих функцій імплікативного базису. Створення алгебри імплікативного базису вирішує проблему мінімізації функцій в імплікативному базисі.

3. Рівносильні перетворення комбінаторними образами, що за своїми властивостями мають більшу інформаційну ємність, спроможні з ефектом замінити вербальні процедури алгебричних перетворень.

Алгебра, що створена, у складі правил спрощення функцій імплікативного базису забезпечує безпосереднє перетворення логічних виразів. А метод образних перетворень, використовуючи наглядні комбінаторні структури бінарних матриць та уніфікацію оригінальних процедур, забезпечує належне перетворення логічних виразів та функцій. У свою чергу, вербальні процедури мають меншу інформаційну ємність, вимагають активного спостереження, що створює початок для не виявлення логічних операції (наприклад, узагальненого склеювання змінних, суперсклеювання змінних, неповного супер-склеювання змінних, напівсклеювання змінних), а відтак, зменшують можливості аналітичного методу.

Мемристори найбільш ефективні, коли використовується логіка, що ґрунтується на операції імплікації. Паралельне підключення двох мемристорів реалізує функцію матеріальної імплікації. Разом з універсальними елементами AND-NOT і OR-NOT функція імплікації та функція константа нуль, формує функціонально повний базис. Таким чином на його основі можна виконувати всі 16 перемикальних функцій двох змінних. Створена алгебра у складі правил спрощення імплікативних функцій з ілюстрацією образних перетворень логічних процедур дає змогу поширювати застосування функціонально повного імплікативного базису у галузях обчислювальної техніки, де зазначений базис на даний час використовується не в повній мірі.

На структурі кросбару з мемристорами можна реалізувати операцію імплікації, а на її основі – й інші логічні операції. Все, що може бути обчислено на кремнії, можна зробити за допомогою мемристорів.

Література

1. Булкин В.И. Моделирование отношения импликации с использованием направленных реляционных сетей // Восточно-Европейский журнал передовых технологий. 2014 – № 6/4 (72). – С. 30 – 37. DOI: 10.15587/1729-4061.2014.30567 URL: http://journals.uran.ua/eejet/article/viewFile/30567/31312 (дата обращения: 30.11.2020).

2. Дичка І. А., Тарасенко В. П., Онай М. В. Основи прикладної теорії цифрових автоматів. – Київ, – КПІ ім. Ігоря Сікорського, 2019, 505 с. – URL:

1. https://core.ac.uk/download/pdf/323531874.pdf (дата звернення: 30.11.2020).

2. Різник, В.В., Соломко М.Т., Тадеєв П. О., Назарук В. Д., Зубик Л. В., Волошин В. С. Алгоритм мінімізації булевих функцій методом оптимального комбінування послідовності образних перетворень // Східно-Європейський журнал передових технологій. 2020 – № 3/4 (105). – С. 43 – 51. URL: http://journals.uran.ua/eejet/article/view/206308 (дата звернення: 30.11.2020).

3. Shahar Kvatinsky, Avinoam Kolodny (2014) Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies. *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, VOL. 22, NO. 10, OCTOBER 2014. URL: http://www2.ece.rochester.edu/users/friedman/papers/TVLSI_IMPLY_14.pdf

4. Saeideh Shirinzadeh, Kamalika Datta, Rolf Drechsler. (2017) Logic Design using Memristors: An Emerging Technology. Department of Mathematics and Computer Science, University of Bremen, Bremen, Germany. 2017. URL: http://www.informatik.uni-bremen.de/agra/doc/konf/ISMVL-tutrial-18.pdf

5. Predrag Teodorovic, Bogdan Vukobratovic, Rastislav Struharik, S. Dautovic (2012) Sequence generator for computing arbitrary n-input Boolean function using two memristors / Conference: Telecommunications Forum (TELFOR), 2012 20th, November 2012. DOI: 10.1109/TELFOR.2012.6419391. URL:https://www.researchgate.net/publication/261311914_Paper_title_Sequence_generator_for_computing_arbitrary_n-input_Boolean_function_using_two_memristors

6. Shokat Ganjeheizadeh Rohani and Nima Taheri Nejad. (2017) An Improved Algorithm for IMPLY Logic Based Memristive Full-Adder. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE). URL: https://www.semanticscholar.org/paper/An-improved-algorithm-for-IMPLY-logic-based-Rohani-Taherinejad/271018ce172c3670bbb8561d82d2e5ee2e165c54

7. Xiaoxiao Wang, Jiaxin Han, Yifang Yang, Yu Li (2019) An Improved Mapping and Optimization Method for Implication-based Memristive Circuits Using And-Inverter Graph // Journal of Physics Conference Series 1237:032026, August

2019. DOI: 10.1088/1742-6596/1237/3/032026. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1237/3/032026/pdf

8. Mehri Teimoory, Amirali Amirsoleimani, Jafar Shamsi, Arash Ahmadi, Shahpour Alirezaee, Majid Ahmadi (2012) Optimized Implementation of Memristor-Based Full Adder by Material Implication Logic. *Department of Electrical Engineering, Islamic Azad University, Science and Research Branch, Kermanshah, Iran.* 2012. URL: https://arxiv.org/ftp/arxiv/papers/1501/1501.00606.pdf

9. Julien Borghetti, Gregory S. Snider, Philip J. Kuekes, J. Joshua, YangDuncan R. StewartR. Stanley Williams (2010) Memristive' switches enable 'stateful' logic operations via material implication. *Nature* volume 464, pp. 873–876 April 2010. URL: https://www.nature.com/articles/nature08940

10. Eero Lehtonen, Mika Laiho (2009) Stateful implication logic with memristors. *IEEE/ACM International Symposium on Nanoscale Architectures.* 30-31 July 2009. DOI: 10.1109/NANOARCH.2009.5226356. URL: https://ieeexplore.ieee.org/document/5226356

11. Anika Raghuvanshi, Marek Perkowski (2014) Logic synthesis and a generalized notation for memristor-realized material implication gates. *IEEE/ACM International Conference on Computer-Aided Design (ICCAD).* 2-6 Nov. 2014. DOI: 10.1109/ICCAD.2014.7001393 . URL: https://ieeexplore.ieee.org/document/7001393

12. Eero Lehtonen, Jussi Poikonen, Mika Laiho (2010) Two Memristors Suffice to Compute All Boolean Functions. Electronics Letters 46(3). February 2010. DOI: 10.1049/el.2010.3407 . URL: https://www.researchgate.net/publication/49288024_Two_Memristors_Suffice_to_Compute_All_Boolean_Functions

13. Chua L.O. (1971) Memristor – the missing circuit element. *IEEE Trans. Circuit Theory*, 1971, v.18, p.507–519.

14. Borghetti J., Snider G.S., Kuekes P.J. et al. (2010) 'Memristive' switches enable 'stateful' logic operations via material implication. – Nature letters, 2010, v.464, p.873–876.

15. Кривуля Г., Сыревич Е., Власов И., Павлов О. Особенности применения наномемристорной логики для проектирования цифровых систем // Весник ХНТУ №1(46), 2013. ст. 280-285. URL: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=Vkhdtu_2013_1_54 (дата звернення: 30. 11. 2020).

16. Елисеев, Н. мемристоры и кроссбары: нанотехнологии для процессоров // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес 8 / 2010? - С. 84 – 89. URL: https://www.electronics.ru/files/article_pdf/0/article_149_323.pdf

17. Поспелов Д. А. Логические методы анализа и синтеза схем. – М. «Энергия», Изд. 3, перераб. и доп. 1974. 368 с. URL: http://urss.ru/cgi-bin/db.pl?lang=Ru&blang=ru&page=Book&id=25326 (дата звернення: 30. 11. 2020).

18. Різник, В.В., Соломко М.Т. Мінімізація кон'юктивних нормальних форм булевих функцій комбінаторним методом // Технологічний аудит та резерви виробництва. 2018 – № 5/2 (43). – С. 42 – 55. URL: http://journals.uran.ua/tarp/article/view/146312 (дата звернення: 30. 11. 2020).

19. Різник В.В., Соломко М.Т. Застосування алгебричної операції супер-склеювання змінних для мінімізації булевих функцій комбінаторним методом // Технологічний аудит та резерви виробництва. – Вип. 6/2 (38), 2017. Р. 60 – 76. – URL: http://journals.uran.ua/tarp/article/viewFile/118336/112951 (дата звернення: 30. 11. 2020).

20. Никишечкин А.П. Дискретная математика и дискретные системы управления. Учебное пособие для академического бакалаврата. 2-е изд. М.: Юрайт, - 2018. – 299 с. URL: https://urait.ru/book/diskretnaya-matematika-i-diskretnye-sistemy-upravleniya-442305

21. Примеры решений:минимизация ДНФ. URL: https://www.matburo.ru/ex_dm.php?p1=bfmin (дата обращения: 30. 11. 2020).

22. Різник, В.В., Соломко, М.Т. Дослідження протоколів мінімізації 5-bit булевих функцій комбінаторним методом // Технологічний аудит та резерви виробництва, 4/2 (42), 2018. – 41–52. DOI: 10.15587/2312-8372.2018.140351. URL:http://journals.uran.ua/tarp/article/view/140351 (дата звернення: 30. 11. 2020).