МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Навчально-науковий інститут комп'ютерного моделювання,
прикладної фізики та математики

Ю.І. Под'ячий, Г.Ю. Під'ячий

ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВОЇ ЕЛЕКТРОНІКИ

Конспект лекцій для студентів спеціальностей
105 – Прикладна фізика та наноматеріали,
172 – Електронні комунікації та радіотехніка.

Харків
2023

В наш час цифрові сигнали і пристрої широко застосовуються в усіх галузях науки і техніки. Вони глибоко проникли у всі області життя і майже повністю замінили аналогові сигнали і системи, що оперують ними. Це пов'язано з тими перевагами, що надають цифрова техніка порівняно з аналоговою. Без перебільшення можна сказати, що мікроелектроніка і цифрові сигнали значно змінили наше життя. Стільниковий зв'язок, цифрове телебачення і радіо, інтернет, цифрові системи управління і контролю, "пластикові" гроші і банківський сервіс – це далеко не повний перелік систем, в яких використовуються цифрові сигнали.

Математичною основою побудови і функціонування цифрових пристроїв є логічна (або булева) алгебра. В конспекті розглянуті основні функції алгебри логіки, аксіоми і закони, якими вони керуються, їх реалізація електронними логічними елементами. Спрощення виду логічних функцій є однією з основних задач алгебри логіки. В конспекті основна увага приділяється мінімізації логічних функцій за допомогою карт Карно (діаграм Вейча).

Зміст лекцій відповідає вимогам бакалаврату спеціальностей 105 – Прикладна фізика та наноматеріали, 172 – Електронні комунікації та радіотехніка з навчальних дисципліни "Цифрова електроніка" і "Схемотехніка телекомунікаційних пристроїв".

ЛОГІЧНІ КОНСТАНТИ І ЗМІННІ

Для опису алгоритмів роботи цифрових пристроїв потрібний відповідний математичний апарат. Такий апарат для вирішення завдань формальної логіки в середині минулого століття розробив ірландський математик Джорж Буль. На його ім'я математичний апарат і отримав назву булевої алгебри або алгебри логіки.

Булева алгебра – це математична система, що оперує двома поняттями: подія істинна і подія хибна. Природно асоціювати ці поняття з цифрами, що використовуються в двійковій системі числення. Далі будемо їх називати відповідно логічними одиницею (1) та нулем (0).

Два елементи булевої алгебри, саме подія істинна і подія хибна, називаються її константами. Будемо розуміти під ними значення відповідно логічну 1 і логічний 0.

Для того щоб описати за допомогою булевої алгебри поведінку та структуру цифрової схеми, її вхідним, вихідним і внутрішнім вузлам ставлять у відповідність булеві змінні, які можуть набувати лише двох значень:

$$X=0, \text{ якщо } X\neq 1,$$

$$X=1, \text{ якщо } X\neq 0.$$

ОПЕРАЦІЇ БУЛЬОВОЇ АЛГЕБРИ

Основні операції, що здійснюються в булевій алгебрі над константами та змінними.

Логічне складання. Цю операцію називають операцією *АБО* чи диз'юнкцією. Вона визначається постулатом, який можна виразити у вигляді таблиці, яка називається *таблицею істинності* чи *таблицею відповідності*. Така таблиця для двох змінних має вигляд:

A	B	F = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Слід зазначити, що ця операція справедлива для довільного числа змінних. Число змінних, над якими виконується операція, позначається цифрою, що стоїть перед її позначенням (тут 2*АБО*).

Результат операції логічного складання дорівнює одиниці, якщо хоч одна змінна дорівнює одиниці; якщо всі змінні дорівнюють нулю, то результат операції логічного складання дорівнює нулю.

Логічне множення. Цю операцію називають операцією *І* чи кон'юнкцією. Її постулат визначається такою таблицею істинності.

A	B	F = A · B
0	0	0
0	1	0
1	0	0
1	1	1

Слід зазначити, що ця операція також справедлива для довільної кількості змінних. Число змінних, над якими виконується операція, також позначається цифрою (тут 2*І*).

Результат операції логічного множення дорівнює нулю, якщо хоч одна змінна дорівнює нулю; якщо всі змінні дорівнюють одиниці, то результат операції логічного складання дорівнює одиниці.

Заперечення. Операцію заперечення називають операцією *НІ* чи інверсією. Для її позначення використовують риску над відповідним значенням чи виразом.

A	F = $\bar{A}$
0	1
1	0

Результат операції логічного заперечення дорівнює одиниці, якщо змінна дорівнює нулю; якщо змінна дорівнюють нулю, то результат операції заперечення дорівнює одиниці.

СПОСОБИ ЗАПИСУ ФУНКЦІЙ АЛГЕБРИ ЛОГІКИ

Нехай на вході цифрового логічного пристрою діє двійковий код {X}, а на виході – двійковий код {Y}. Для того щоб описати роботу пристрою необхідно визначити залежність кожного елемента вихідного коду від усіх елементів вхідного коду. Тобто. визначити функціональну залежність {Y} = F ({X}). Ця залежність називається *функцією алгебри логіки* − ФАЛ.

Пристрої, робота яких описується за допомогою ФАЛ, називаються логічними. Існує кілька способів визначення ФАЛ.

Словесне (***вербальне***). Наприклад: значення змінної на виході пристрою F дорівнює 1, якщо рівні одиниці не менше двох вхідних змінних з трьох. Даний вид опису найчастіше застосовується для початкового, вихідного опису поведінки логічного устрою (технічне завдання).

Таблиця істинності (***відповідності***). Визначимо її для попереднього прикладу. Нехай пристрій має 3 входи A,B,C і один вихід F. Очевидно, таблиця буде мати такий вигляд:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Опис ФАЛ у вигляді алгебраїчного виразу.

Спочатку визначення виду алгебраїчного виразу.

Елементарна кон'юнкція – логічне множення змінних та їх заперечень.

$\overline{X}1 \cdot X2$ – ранг 2.

$A1 \cdot \overline{A}2 \cdot \overline{A}3 \cdot A4$ – ранг 4.

Якщо в елементарну кон'юнкцію входять усі аргументи функції, вона називається *конституентою одиниці*:

$F(A, B, C, D) = \overline{A} \cdot B \cdot \overline{C} \cdot D$.

Елементарна диз'юнкція – логічна сума змінних та їх заперечень.

$\overline{X}1 + X2$ – ранг 2.

$A1 + \overline{A}2 + \overline{A}3 + A4$ – ранг 4.

Якщо в елементарну диз'юнкцію входять усі аргументи функції, вона називається *конституентою нуля*:

$F(A, B, C, D) = A + B + C + D$.

Диз'юнктивна нормальна форма (ДНФ) – логічна сума елементарних кон'юнкцій, у кожній з яких аргумент або його інверсія входять один раз:

$F(A,B,C,D) = \overline{A} \cdot B \cdot \overline{C} \cdot D + A \cdot \overline{B} + B \cdot \overline{C} + C \cdot D$.

Досконала диз'юнктивна нормальна форма (ДДНФ) – логічна сума конституент одиниці:

$F(A,B,C,D) = \overline{A} \cdot B \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot D + A \cdot B \cdot \overline{C} \cdot D + A \cdot B \cdot C \cdot D$.

Кон'юнктивна нормальна форма (КНФ) – логічний добуток елементарних диз'юнкцій, у кожній з яких аргумент або його інверсія входять один раз:

$F(A,B,C,D)=(A+C) \cdot (A+D) \cdot (A+B+C) \cdot (A+B)$.

Досконала кон'юнктивна нормальна форма (ДКНФ) – логічне множення конституент нуля:

$F(A,B,C,D) = (A+B+C+D) \cdot (A+B+C+D) \cdot (A+B+C+D) \cdot (A+B+C+D)$.

Отримання ДДНФ з таблиці істинності

Описані нижче процедури мають велике значення при проектуванні цифрових електронних пристроїв. Існують кілька способів отримання функцій алгебри логіки. Розглянемо найпростіший. Будемо вважати, що таблиця істинності сформована для пристрою, на виході якого формується одиниця, якщо не менше двох вхідних змінних є одиницями.

а) Для кожного набору змінних, у якому ФАЛ дорівнює одиниці, записуємо елементарні кон'юнкції (елементарні логічні множення) вхідних змінних, причому змінні, рівні нулю, записуємо з запереченням (з інверсією). Отримаємо конституенти одиниці.

б) Логічно підсумовують отриманні конституенти одиниці.

A	B	C	F	Конституента одиниці
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{A}BC$
1	0	0	0	
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	1	ABC

$$F(A, B, C) = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC.$$

Отримання ДКНФ із таблиці істинності

Отримаємо цю форму для такого ж пристрою, тобто для такої ж таблиці істинності.

а) Для кожного набору змінних, у якому ФАЛ дорівнює нулю, записуємо елементарні логічні суми вхідних змінних, причому змінні, що дорівнюють одиниці, записуються з інверсією. Отримаємо конституенти нуля.

б) Логічно перемножуємо отримані конституенти нуля.

A	B	C	F	Конституента нуля
0	0	0	0	$A+B+C$
0	0	1	0	$A+B+\bar{C}$
0	1	0	0	$A+\bar{B}+C$
0	1	1	1	
1	0	0	0	$\bar{A}+B+C$
1	0	1	1	
1	1	0	1	
1	1	1	1	

$$F(A,B,C) = (A+B+C)\cdot(A+B+\bar{C})\cdot(A+\bar{B}+C)\cdot(\bar{A}+B+C).$$

Опис ФАЛ як послідовності десяткових чисел.

Іноді для скорочення запису логічної функції її представляють як послідовність десяткових чисел. При цьому послідовно записують десяткові еквіваленти двійкових кодів вхідних сигналів, що відповідають конституантам одиниці чи нуля. Запишемо в такому вигляді одержані вище функції.

Таблиця для ДДНФ: четвертий рядок - $(011)_2 = (3)_{10}$; 6-й рядок - $(101)_2 = (5)_{10}$; 7-й рядок - $(110)_2 = (6)_{10}$; 8-й рядок – $(111)_2=(7)_{10}$. Тому функція записується так:

$$F(A,B,C)= (3,5,6,7).$$

Таблиця для ДКНФ: Перший рядок - $(000)_2 = (0)_{10}$; Другий рядок – $(001)_2=(1)_{10}$; 3-й рядок – $(010)_2=(2)_{10}$; 5-й рядок – $(100)_2=(4)_{10}$. Тому функція записується так:

$$F(A,B,C)= \mathbf{\Pi}\ (0,1,2,4).$$

(Тут Σ − символ підсумовування; **П** − символ перемноження).

Короткі висновки

- Алгоритм функціонування цифрового логічного пристрою зручно представляти у вигляді таблиці істинності.
- За допомогою такої таблиці легко одержати логічну функцію, що описує алгоритм роботи пристрою у досконалій диз'юнктивній нормальній формі (ДДНФ) або у досконалій кон'юнктивній нормальній формі (ДКНФ).
- Спеціальними процедурами мінімізації (які будуть розглянуті нижче) форми ДДНФ і ДКНФ перетворюють у форми ДНФ і КНФ. Ці форми дозволяють більш просто побудувати електронні схеми, що реалізують даний алгоритм.

АКСІОМИ І ЗАКОНИ АЛГЕБРИ ЛОГІКИ

Аксіоми

Аксіомою називають твердження, яке не потребує доведення. Добре відомі аксіоми евклідової геометрії. На них базується геометрія на площині. Більшість наук в своїй основі мають перелік аксіом. Такі аксіоми сформульовані і в логічній алгебрі.

Запишемо першу групу аксіом, аналізуючи таблицю істинності операції *АБО* (логічне складання):

Рядок	A	B	F = A + B
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

1) $A + 0 = A$ (рядки 1 і 3);
2) $A + 1 = 1$ (рядки 2 і 4);
3) $A + A = A$ (рядки 1 і 4);
4) $A + \overline{A} = 1$ (рядки 2 і 3);

Другу групу аксіом запишемо на основі таблиці істинності операції *І* (логічне множення):

Рядок	A	B	F = A + B
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

5) $A \cdot 0 = 0$ (рядки 1 і 3);
6) $A \cdot 1 = A$ (рядки 2 і 4);
7) $A \cdot A = A$ (рядки 1 і 4);
8) $A \cdot \overline{A} = 0$ (рядки 2 і 3);

Остання аксіома – подвійне заперечення:

$$9)\ \overline{\overline{X}} = X.$$

Закони

Аксіоми і закони алгебри логіки застосовуються в операціях перетворення логічних функцій, зокрема – при їх спрощенні. Деякі закони зрозумілі інтуїтивно і не потребують доказів; інші закони таких доказів потребують.

Переміщувальний закон

$\mathbf{A + B = B + A}$

$\mathbf{A \cdot B = B \cdot A}$

Сполучний закон

$\mathbf{A + (B + C) = (A + B) + C}$

$\mathbf{A \cdot (B \cdot C) = (A \cdot B) \cdot C}$

Закон розподілу

$\mathbf{A \cdot (B + C) = A \cdot B + A \cdot C}$

$\mathbf{A + B \cdot C = (A + B) \cdot (A + C)}$

Закон інверсії

Якщо $\mathbf{A = B}$, то $\mathbf{\overline{A} = \overline{B}}$

Закон поглинання

$\mathbf{A + A \cdot B = A}$

{Доказ: $A + A \cdot B = A\,(1+B) = A \cdot 1 = A$}

$\mathbf{A \cdot (A + B) = A}$

{Доказ: $A \cdot (A + B) = A \cdot A + A \cdot B = A + AB = A(1 + B) = A \cdot 1 = A$}

Закон склеювання

$\mathbf{A \cdot B + A \cdot \bar{B} = A}$

{Доказ: $A \cdot B + A \cdot \bar{B} = A(B + \bar{B}) = A \cdot 1 = A$}

$\mathbf{(A + B) \cdot (A + \bar{B}) = A}$

{Доказ: $(A + B) \cdot (A + \bar{B}) = A \cdot A + A \cdot \bar{B} + A \cdot B + B \cdot \bar{B} = A + A \cdot \bar{B} + A \cdot B + 0 =$

$= A + A(\bar{B} + B) = A + A \cdot 1 = A + A = A$}

Правило викреслення

$\mathbf{\bar{A} + A \cdot B = \bar{A} + B}$

{Доказ: $\bar{A} + A \cdot B = (\bar{A} + A) \cdot (\bar{A} + B) = 1 \cdot (\bar{A} + B) = \bar{A} + B$}

$\mathbf{\bar{A}(A + B) = \bar{A} \cdot B}$

{Доказ: $\bar{A}(A + B) = \bar{A} \cdot A + \bar{A} \cdot B = 0 + \bar{A} \cdot B = \bar{A} \cdot B$}

Теорема де Моргана

Одна з найважливіших теорем алгебри логіки, тому що вона відіграє важливу роль в перетворенні логічних функцій.

Запишемо таблиці істинності для операції *АБО* у прямій та інверсній формах:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

$\Rightarrow$

$\bar{A}$	$\bar{B}$	$\bar{F}$
1	1	1
1	0	0
0	1	0
0	0	0

Вочевидь, що якщо $F = A + B$, то $\bar{F} = \bar{A} \cdot \bar{B}$. Інверсія першого виразу $\bar{F} = \overline{A + B}$.

Тоді справедливо:

$$\overline{A + B} = \bar{A} \cdot \bar{B}.$$

Тепер запишемо таблиці істинності для операції *І* у прямій та інверсній формах:

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

$\bar{A}$	$\bar{B}$	$\bar{F}$
1	1	1
1	0	1
0	1	1
0	0	0

Вочевидь, що якщо $F=A{\cdot}B$, то $\bar{F}=\bar{A}+\bar{B}$. Інверсія першого виразу $\bar{F} = \overline{A + B}$.

Тоді справедливо:

$$\overline{A \cdot B} = \bar{A} + \bar{B}.$$

Ці формули виражають теорему де Моргана: заперечення функції *АБО* можна замінити функцією *І* заперечення змінних; заперечення функції *І* можна замінити функцією *АБО* заперечення змінних

Важливі наслідки теореми де Моргана:

$$\overline{\overline{A+B}} = A+B = \overline{\bar{A}\cdot\bar{B}}.$$
$$\overline{\overline{A\cdot B}} = A\cdot B = \overline{\bar{A}+\bar{B}}.$$

Теорема виражає принцип двоїстості (дуальності) логічних функцій *АБО* та *І*. Це означає, що функцію *АБО* операціями заперечення можна замінити функцією *І*. Таким же способом операцію *І* можна замінити операцією *АБО*.

Важливим практичним наслідком принципу двоїстості є те що, записи логічних висловів і, отже, побудову логічних схем, можна здійснити лише двома типами операцій, наприклад операціями *І, НІ* чи *АБО, НІ*.

Поширені логічні операції

Розглянемо три логічні операції, які не відносяться до основних, але часто використовуються на практиці.

Логічне слідування – ***імплікація***.

A	B	F
1	1	1
1	0	0
0	1	1
0	0	1

Операція пов'язує два простих логічних вирази, з яких перше є умовою (А), а друге (В) – наслідком цієї умови. Результатом ІМПЛІКАЦІЇ є хибна подія тільки тоді, коли умова А істинна, а наслідок В хибний. Виражається словами ЯКЩО … , ТО …

Позначення операції і її логічна формула:

$$A \rightarrow B = \bar{A} + B.$$

Логічна рівнозначність – ***еквівалентність***.

A	B	F
1	1	1
1	0	0
0	1	0
0	0	1

Операція визначає результат порівняння двох простих логічних виразів A і B. Результатом еквівалентності є новий логічний вираз, який буде істинним тоді і тільки тоді, коли обидва вихідні вирази одночасно істинні або хибні.

Позначення операції і її логічна формула:

$$A \leftrightarrow B = \overline{A}\cdot\overline{B} + A\cdot B$$

Складання по модулю 2 – ***виключне АБО***.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Ця операція відрізняється від операції *АБО* тим, що при вхідних змінних A = 1 і B = 1 результат на виході C = 0.

Позначення операції і її логічна формула:

$$A \oplus B = A\cdot\overline{B} + \overline{A}\cdot B$$

Перетворення основних логічних функцій через додавання по модулю 2:

$$\overline{A} = A \oplus 1$$

$$A + B = A \oplus B \oplus A \cdot B$$

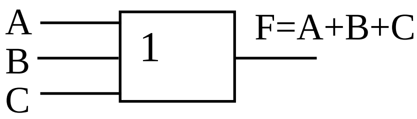
$$A \cdot B = (A \oplus B) \oplus (A + B)$$

ЛОГІЧНІ ЕЛЕМЕНТИ

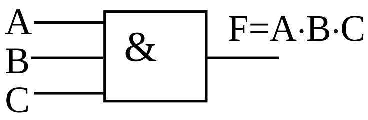
В сучасній електроніці елементи, що виконують логічні операції, виробляються промисловістю у вигляді цифрових інтегральних схем (мікросхем), які називають *логічними елементами*. За їх допомогою реалізуються логічні функції будь якої складності. Розглянемо логічні елементи, що виконують основні операції булевої алгебри.

Схемне позначення основних логічних елементів.

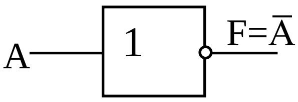
1. Елемент **АБО (OR)** - диз'юнктор:



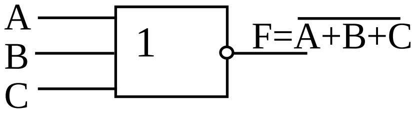
2. Елемент **І (AND)** – кон'юнктор:



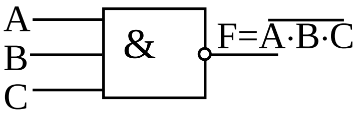
3. Елемент **НІ (NOT)** – інвертор:



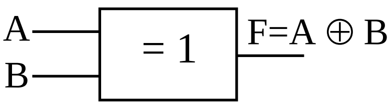
4. Елемент **АБО-НІ (NOR)** - стрілка Пірса:



5. Елемент **І-НІ (NAND)** – штрих Шеффера:



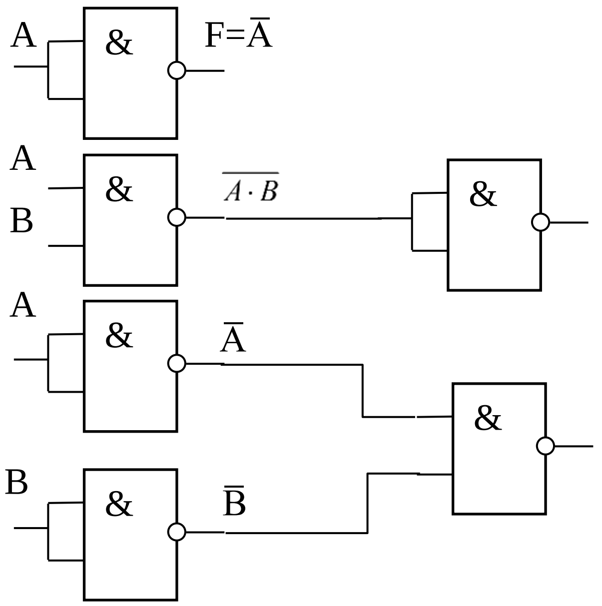
6. Елемент **Виключне АБО (XOR)**:



Функціонально повна система логічних елементів

Функціонально повною системою називається сукупність логічних елементів, що дозволяє реалізувати логічну схему довільної складності. Таким чином, системи двох елементів І і НІ, а також АБО та НІ нарівні з системою з трьох елементів (І, АБО, НІ) є функціонально повними. На практиці широке застосування знайшли логічні елементи, що поєднують функції елементів зазначених вище функціонально повних систем. Це елементи І-НІ та АБО-НІ. За визначенням кожен із цих елементів сам по собі утворює функціонально повну систему. Щоб довести це необхідно за допомогою тільки одного такого елемента створити пристрої, які реалізують основні логічні операції І, АБО і НІ.

Як приклад розглянемо схеми, що виконують ці операції на логічних елементах І-НІ.



НІ: $A \cdot A = A$ Використана аксіома 5

І: $\overline{\overline{A \cdot B}} = A \cdot B$ Використана аксіома 9

АБО: $\overline{\overline{A} \cdot \overline{B}} = A + B$ Використано наслідок теореми де Моргана

ПРИКЛАД СТРУКТУРНОЇ СХЕМИ ЛОГІЧНОГО ПРИСТРОЮ

Побудуємо схему пристрою, що реалізує алгоритм:

- Пристрій має три входи A,B,C і один вихід F.
- Сигнал на виході буде рівним 1, якщо дорівнюють одиниці не менше двох вхідних сигналів.

Таблиця істинності:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

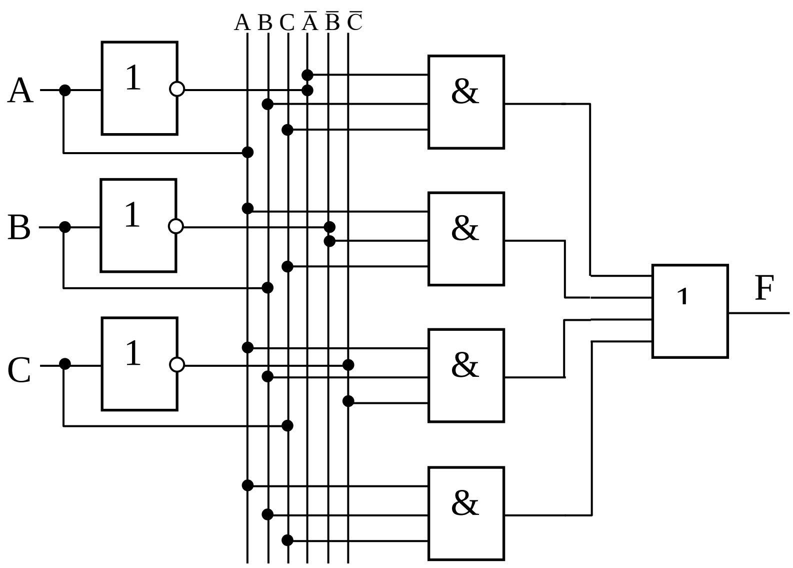
Логічна функція у формі ДДНФ:

$$F(A,B,C) = \overline{A}\cdot B\cdot C + A\cdot \overline{B}\cdot C + A\cdot B\cdot \overline{C} + A\cdot B\cdot C.$$

Які логічні елементи будуть потрібні?

- Для отримання інверсних значень аргументів A, B, C потрібно 3 (елемента *НІ*);
- Для визначення логічних добутків ($\overline{A}\cdot B\cdot C$, $A\cdot \overline{B}\cdot C$ і т.д.) потрібно 4 елементи 3*І*;
- Для отримання остаточної логічної суми потрібен один елемент 4*АБО*.

При побудові схеми використана спільна шина.



МІНІМІЗАЦІЯ ЛОГІЧНИХ ПРИСТРОЇВ

Отже, було показано, що логічну схему, що реалізує заданий алгоритм перетворення сигналів, можна синтезувати безпосередньо за виразом, представленим у вигляді ДДНФ або ДКНФ, які просто визначити з таблиці істинності. Однак отримана при цьому схема, як правило, не оптимальна з точки зору її практичної реалізації – кількості елементів і з'єднань. Тому вихідні ФАЛ зазвичай спрощують (мінімізують). При цьому одержують ФАЛ у вигляді мінімальних ДНФ та КНФ.

Метою мінімізації логічної функції є зменшення вартості її технічної реалізації та збільшення швидкодії. При проектуванні апаратури із застосуванням ЦІС ставиться вимога зменшення числа корпусів ІВ та їх з'єднань.

Принципи мінімізації ФАЛ

Принципи мінімізації ґрунтуються на аксіомах і законах алгебри логіки.

- аксіоми:

  $A + \overline{A} = 1;$

  $A \cdot \overline{A} = 0;$

  $A+A=A;$

  $A{\cdot}A=A.$

- закон поглинання:

  $A + A\,B = A;$

  $A\cdot (A+B) = A.$

- закон склеювання:

  $A{\cdot}B + A{\cdot}\overline{B} = A;$

  $(A + B){\cdot}(A + \overline{B}) = A.$

- правило викреслення:

  $\overline{A} + A{\cdot}B = \overline{A} + B$

  $\overline{A}(A + B) = \overline{A}{\cdot}B$

У лівій частині всіх наведених виразів більше змінних, ніж у правій.

Очевидно, що в результаті дії законів алгебри логіки вихідні вираження значно спрощуються.

Головне! Зауважимо, що спрощення відбувається лише в тому випадку, якщо в сусідніх доданках або співмножниках (часто їх називають термами) є однакові змінні. У всіх розглянутих випадках такою змінною є А. І саме ця змінна і залишається в результаті перетворень.

Розглянемо конкретний приклад. Тут доданки відрізняються лише однією перемінною – С і $\overline{C}$.

$\overline{A}\cdot B\cdot C\cdot D + \overline{A}\cdot B\cdot \overline{C}\cdot D = \overline{A}\cdot B\cdot D\cdot (C+\overline{C}) = \overline{A}\cdot B\cdot D\cdot 1 = \overline{A}\cdot B\cdot D$

Логічне множення, що повторюється в обох термах, залишається в результаті перетворення.

Потрібно звернути увагу на таке: вихідний вираз записано у ДДНФ, а результат вийшов у ДНФ.

Ще приклад. Тут співмножники також відрізняються лише однією змінною – C та $\overline{C}$.

$(\overline{A}+B+C+D)\cdot(\overline{A}+B+\overline{C}+D) = \overline{A}+B+D+C\cdot\overline{C} = \overline{A}+B+D+0 = \overline{A}+B+D.$

У цьому прикладі вихідний вираз записано у ДКНФ, а результат – у КНФ.

Таким чином, головним завданням при мінімізації ДДНФ і ДКНФ є пошук термів, що містять однакові змінні (крім однієї), придатних до склеювання з подальшим поглинанням. Одним з наочних методів пошуку таких термів є використання карт Карно.

Мінімізація ФАЛ за допомогою карток Карно (Вейча)

Карти Карно були винайдені в 1952 році Едвардом В. Вейчем, вдосконалені в наступному році Морісом Карно і були покликані допомогти спростити цифрові електронні схеми.

Даний метод базується на табличному вигляді уявлення ФАЛ. Він широко використовується при ручній, без застосування ЕОМ, мінімізації ФАЛ, кількість змінних в якій зазвичай не перевищує п'яти-шести.

Карта Карно є спеціальним чином організована таблиця істинності, на якій наочно здійснюються операції склеювання з метою спрощення булевої функції та мінімізації форми. Стовпці та рядки таблиці відповідають різноманітним наборам змінних. Ці набори розташовані в такому порядку, що кожний наступний відрізняється від попереднього лише однією змінною. Завдяки цьому, сусідні клітинки карти Карно по горизонталі та вертикалі теж відрізняються значенням лише однієї змінної.

Для ФАЛ n-змінних число клітин у карті Карно дорівнює 2^n.

Розглянемо побудову карт Карно для функцій 2-х, 3-х та 4-х змінних.

1. Функція двох змінних.

Функція представлена в ДКНФ:

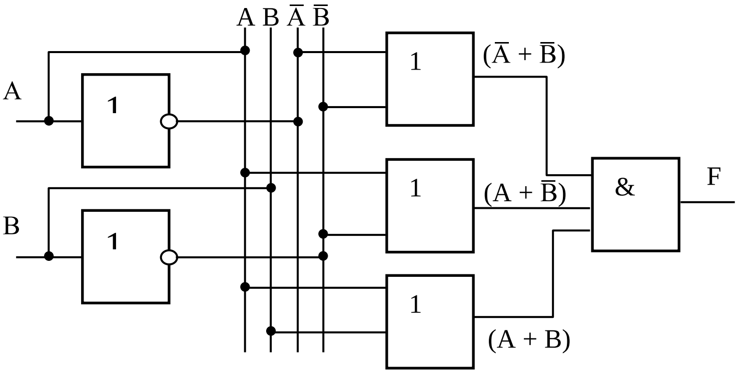
F (A, B) = (A + B)(A + B)(A + B)

Таблиця істинності:

A=0; B=0: F=(1+1)(0+1)(0+0)=0
A=0; B=1: F=(1+0)(0+0)(0+1)=0
A=1; B=0: F=(0+1)(1+1)(1+0)=1
A=1; B=1: F=(0+0)(1+0)(1+1)=0

A	B	F
0	0	0
0	1	0
1	0	1
1	1	0

Схемна реалізація логічної функції без її мінімізації:



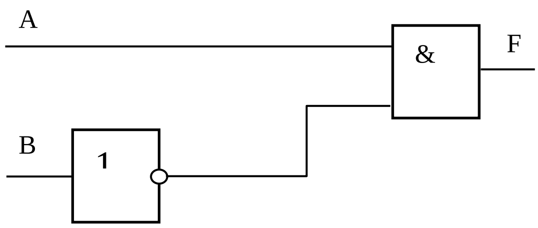
Карта Карно має $2^2 = 4$ клітинки. Це також таблиця істинності у спеціальному вигляді.

A \ B	0	1
0	0	0
1	1	0

Як видно з цієї таблиці функція може бути рівна одиниці тільки при кон'юнкції A=1, $\bar{B}$=1. Тому мінімізована функція у ДНФ:

$$\mathbf{F(A,B) = A \cdot \bar{B}.}$$

Схемна реалізація цієї функції набагато простіше, ніж попередня:



2. Функція трьох змінних.

Вже розглянутий раніше приклад.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Логічне вираження у ДДНФ:

$$F(A,B,C)=\bar{A}\cdot B\cdot C+A\cdot\bar{B}\cdot C+A\cdot B\cdot\bar{C}+A\cdot B\cdot C.$$

Карта Карно містить $2^3 = 8$ клітинок:

AB		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	1

S_2 S_1 S_3

Стовпці пронумеровані відповідно до коду Грея.

Код Грея − система числення, у якій два сусідніх значення різняться лише в одному розряді.

2-бітний код Грея: 00, 01, 11, 10.

3-бітний код Грея: 000, 001, 011, 010, 110, 111, 101, 100.

4-бітний код Грея: 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000.

Перші і останні елементи коду також відрізняються лише в одному розряді.

Як працювати з карткою?

а) Виділяють прямокутні області клітинок з одиницями (для отримання ***ДНФ***) або з нулями (для отримання ***КНФ***). У кожній області має бути 2^k клітинок ($k = 0,1,2,3, \dots$).

б) Крайні клітинки кожного рядка та кожного стовпчика також межують між собою і їх можна включати в область.

в) Кожна одиниця (або нуль) має потрапити до якоїсь області.

г) Для отримання ***ДНФ*** вибираємо першу область з одиницями та дивимося, які змінні не змінюються в межах цієї області. Записуємо кон'юнкцію цих перемінних – це перший *терм*. (Якщо змінна нульова, проставляємо над нею інверсію). Вибираємо другу область, виконуємо те саме, що й для першої, – отримуємо другий терм. І т. д. для всіх областей. Отримані терми логічно складаємо.

Для побудови мінімізованої ***ДНФ*** в картці Карно необхідно виділити прямокутні області, що містять одиниці.

Таких областей на карті три.

Область **S1** (синя): в ній не змінюються значення змінних A і B. Тому терм

$$\mathbf{S1 = A \cdot B}.$$

Область **S2** (зелена): для неї не змінюються значення C і B. Тому терм

$$\mathbf{S2 = B \cdot C}.$$

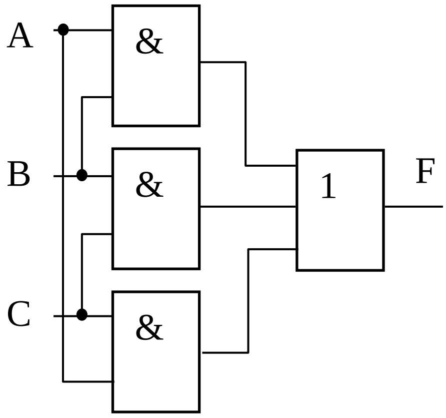
Область **S3** (червона): в ній не змінюються значення A і C. Тому терм

$$\mathbf{S3 = A \cdot C}.$$

Логічно складаємо визначені терми. Мінімізована функція в ДНФ має вигляд:

$$\mathbf{F(A,B,C) = S1 + S2 + S3 = A \cdot B + B \cdot C + A \cdot C.}$$

Схемна реалізація:



На сторінці 14 наведена схема, що виконує той же алгоритм, але побудована за не мінімізованою функцією ДДНФ. Для її реалізації потрібно 8 логічних елементів. Для реалізації одержаної мінімізованої функції в ДНФ потрібно лише 4 елементи. В цьому і полягає мета мінімізації.

Для побудови мінімізованої ***КНФ*** в картці Карно необхідно виділити прямокутні області, що містять нулі.

AB		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	1

S_2 S_1 S_3

Таких областей на карті також три.

Область S1 (синя): не змінюються нульові значення A і C. Тому терм

$$\mathbf{S1 = A + C}.$$

Область S2 (зелена): не змінюються значення A і B. Тому терм

$$\mathbf{S2 = A + B}.$$

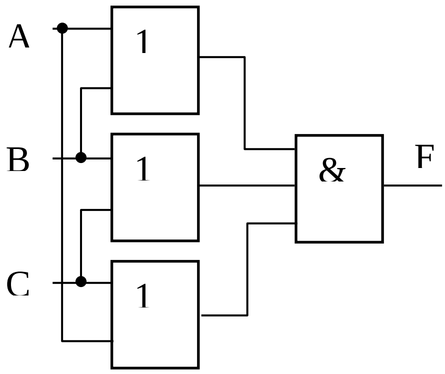
Область S3 (червона): не змінюються значення B і C. Тому терм

$$\mathbf{S3 = B + C}.$$

Таким чином, мінімізована функція в КНФ має вигляд:

$$\mathbf{F(A,B,C) = S1 \cdot S2 \cdot S3 = (A + C) \cdot (A + B) \cdot (B + C).}$$

Схемна реалізація:



Зверніть увагу на симетрію формул і схем ДНФ та КНФ.

3. Функція чотирьох змінних.

Нехай F(A,B,C,D)=1, якщо не менше двох аргументів дорівнюють одиниці.

Таблиця істинності:

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1

A	B	C	D	F
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

ДДНФ для таблиці істинності:

$$F(A,B,C,D)=\bar{A}\bar{B}CD+\bar{A}B\bar{C}D+\bar{A}BC\bar{D}+\bar{A}BCD+A\bar{B}\bar{C}D+A\bar{B}C\bar{D}+A\bar{B}CD+AB\bar{C}\bar{D}$$
$$+AB\bar{C}D+ +ABC\bar{D}+ABCD$$

Для схемної реалізації потрібні наступні елементи:

4 – НІ; 11 - 4І; 3 - 5 АБО (всього 18 логічних елементів).

Карта Карно (рядки та стовпці нумеруються кодом Грея):

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S1**: для всіх клітинок області повторюються значення змінних **C** і **D**. Тому терм

S1 = CD.

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S2**: для всіх клітинок області повторюються значення змінних **A** і **B**. Тому терм

S2 = AB.

C**D**		00	01	11	10
A**B**	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S3**: для всіх клітинок області повторюються значення змінних **B** і **D**. Тому терм

S3 = BD.

C**D**		00	0**1**	1**1**	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S4**: для всіх клітинок області повторюються значення змінних **A** і **D**. Тому терм

S4 = AD.

CD		00	01	**1**1	**1**0
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S5**: для всіх клітинок області повторюються значення змінних **A** і **C**. Тому терм

S5 = AC.

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S6**: для всіх клітинок області повторюються значення змінних **B** і **C**. Тому терм

S6 = BC.

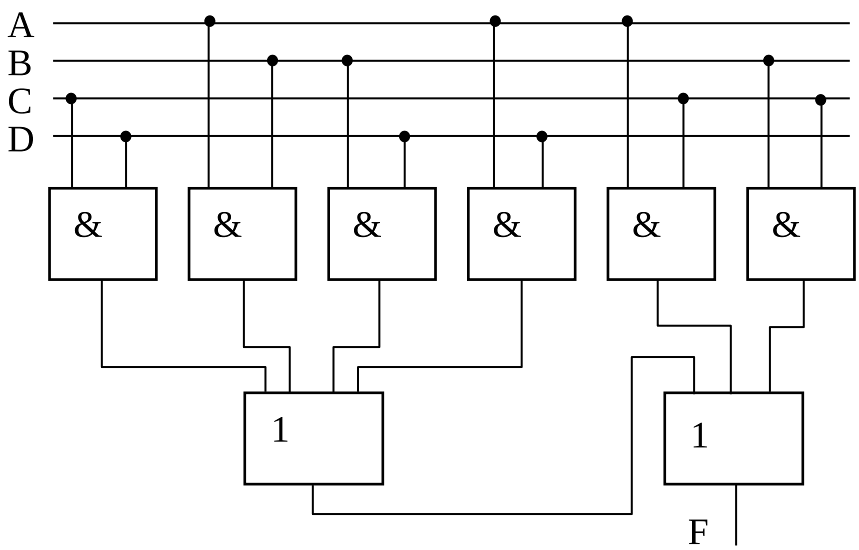
Диз'юнкція термів:

F = S1 + S2 + S3 + S4 + S5 + S6.

Таким чином, мінімізована логічна функція у ДНФ:

F(A,B,C,D) = CD + AB + BD + AD + AC + BC.

Схемна реалізація:



Як видно, вісім логічних елементів у мінімізованій схемі замість вісімнадцяти у схемі без мінімізації.

Визначимо КНФ.

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S1**: для клітинок області повторюються значення змінних **A**,**B**,**C**. Тому терм

S1 = A+B+C.

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S2**: для клітинок області повторюються значення змінних **A**,**C**,**D**. Тому терм

S2 =A+C+D.

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S3**: для клітинок області повторюються значення змінних **A**,**B**,**D**. Тому терм

S3 =A+B+D.

CD		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Область **S4**: для клітинок області повторюються значення змінних **B**,**C**,**D**. Тому терм

S4 =B+C+D.

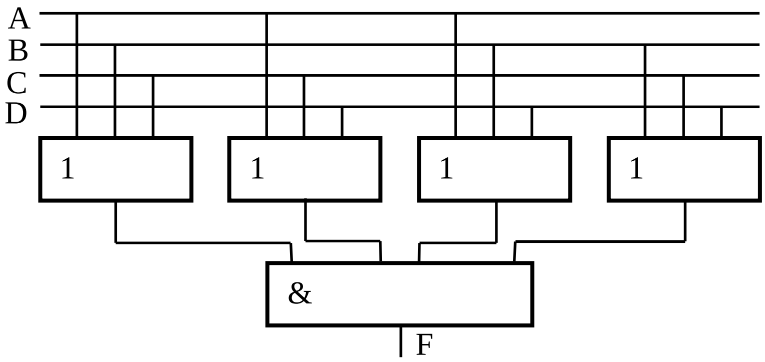
Кон'юнкція термів:

F = S1·S2·S3·S4.

Таким чином, мінімізована ФАЛ у ДНФ:

F(A,B,C,D) =(A+B+C)(A+C+D)(A+B+D)(B+C+D).

Схемна реалізація:



Спрощення логічних виразів за допомогою карток Карно

У розглянутих раніше прикладах ми будували карти Карно за таблицями істинності, а потім, аналізуючи карти Карно, визначали спрощений логічний вираз.

При розв'язанні задач логічної алгебри часто виникає завдання спростити отримане якимось чином логічний вираз. В принципі це можна зробити, безпосередньо перетворюючи складну логічну формулу, використовуючи аксіоми та закони алгебри логіки. Але такі перетворення, як правило, дуже стомлюючі та займають багато часу. Набагато простіше і швидше це можна зробити за допомогою карток Карно.

Розглянемо кілька прикладів.

<u>Приклад 1.</u>

Спростити ФАЛ трьох змінних, поданої в ДДНФ:

$$F(A,B,C) = ABC + A\overline{B}C + ABC\overline{C} + A\overline{B}\overline{C}.$$

номер доданку → 1 2 3 4

В принципі, цей вираз не важко спростити аналітично:

$$F(A,B,C)=ABC + A\overline{B}C + AB\overline{C} + A\overline{B}\overline{C}= A(BC + \overline{B}C + B\overline{C} + \overline{B}\overline{C})=$$

$$= A(C(B + \overline{B}) + \overline{C}(B + \overline{B}))= A(C\cdot 1 + \overline{C}\cdot 1) = A(C + \overline{C}) = A\cdot 1 = A.$$

Зробимо це за допомогою карти Карно, яка має такий вигляд.

	B		$\overline{B}$	
C	0	1_1	1_2	0
$\overline{C}$	0	1_3	1_4	0
	$\overline{A}$	A		$\overline{A}$

Тут стовпці і рядки позначені тими змінними, які в них не змінюються. У першому і другому стовпцях не змінюється B, в третьому і четвертому – $\overline{B}$. У другому і третьому стовпцях не змінюється A, а першому і четвертому – $\overline{A}$. У першому рядку не змінюється C, а у другому – $\overline{C}$.

Головне! На одній грані таблиці має бути одна змінна − пряме значення та інверсне. Різні змінні не повинні займати одні і ті ж стовпці та одні й ті ж рядки.

Карта заповнюється так.

Перше доданок функції – ABC. Йому має відповідати певна клітина (точка) карти. Воно має бути істинним, тобто. повинно бути рівним 1. Це може бути тільки тоді, коли і A і B і C дорівнюють 1. Отже, точка цього доданка повинна знаходитися в клітці на перетині рядків і стовпців, відповідним змінним A, B, C. Така клітина єдина. Вона позначена так: 1_1. Рядкова одиниця – значення доданку, а підрядкова – його номер у вихідній формулі.

Другий доданок – $A\bar{B}C$. Його точка знаходиться на перетині стовпців і клітин, що відповідають змінним A, $\bar{B}$, C. Ця точка знаходиться в клітині, позначеної 1_2.

Аналогічно знаходяться і інші точки доданків. Незаповнені одиницями клітини заповнюються нулями, або залишаються порожніми.

Очевидно, на карті є лише одна область, що містить чотири одиниці. У цій області не змінюється єдина змінна – А. Отже, мінімізована логічна формула має вигляд:

$$F(A,B,C) = A.$$

Цей результат відповідає тому, що отримано за допомогою аналітичних перетворень.

Приклад 2.

За допомогою карти Карно спростити вираз

$$F(A,B,C) = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC.$$

номер доданку → 1 2 3 4

Карта Карно:

	B		$\bar{B}$	
A	1_3	1_4	1_2	
$\bar{A}$		1_1		
	$\bar{C}$	C		$\bar{C}$

Тут три області одиниць:

область S_1 – $\{1_1;1_4\}$ − терм S_1=BC;
область S_2 – $\{1_3;1_4\}$ − терм S_2=AB;
область S_3 – $\{1_4;1_2\}$ − терм S_3=AC.

Спрощена ФАЛ в ДНФ є диз'юнкція термів:

$$F(A,B,C) = BC + AB + AC$$

Приклад 3.

За допомогою карти Карно спростити вираз

$$,B,C,D) = A\bar{B}C\bar{D}+A\bar{B}\bar{C}\bar{D}+AB\bar{C}\bar{D}+\bar{A}B\bar{C}\bar{D}+\bar{A}\bar{B}CD+\bar{A}BCD+ABC\bar{D}.$$

№ доданку → *1* *2* *3* *4* *5* *6* *7*

Карта Карно:

	B		$\bar{B}$		
A	1_3	0	0	1_2	$\bar{C}$
	1_7	0	0	1_1	C
$\bar{A}$	0	1_6	1_5	0	
	1_4	0	0	0	$\bar{C}$
	$\bar{D}$	D		$\bar{D}$	

Знаходимо клітинки картки всіх доданків при тих значеннях змінних, кон'юнкція яких дорівнює одиниці.

Три області. Знаходимо їхні терми.

Область S_1 – 1_1, 1_2, 1_3, 1_7. Не змінюються A, $\overline{D}$. Терм $S1=A\overline{D}$.
Область S_2 – 1_5, 1_6. Не змінюються $\overline{A}$, C, D. Терм $S2=\overline{A}CD$.
Область S_3 – 1_3, 1_4. Не змінюються B, $\overline{C}$, $\overline{D}$. Терм $B\overline{C}\overline{D}$.

Диз'юнкція термів дає спрощену ФАЛ у ДНФ:

$$F(A,B,C,D) = A\cdot\overline{D}+\overline{A}\cdot C\cdot D+B\cdot\overline{C}\cdot\overline{D}.$$

Зазначимо: чим більше одиниць входять в область, тим більше спрощується доданок.

ОСНОВНІ ВИСНОВКИ

1. Для опису алгоритмів роботи цифрових пристроїв, що працюють у двійковому (бінарному) коді, використовується математичний апарат алгебри логіки, яка також носить ім'я її творця − алгебра Буля.

2. В алгебрі логіки доводиться твердження, що алгоритм функціонування будь-якого цифрового пристрою може бути описаний за допомогою:

- трьох функцій АБО, І, НІ;
- двох функцій АБО та НІ;
- двох функцій І та НІ;
- однієї функції АБО-НІ;
- однієї функції І-НІ.

3. Алгоритм роботи цифрової системи може бути описаний за допомогою логічних виразів, що мають форми:

- ДНФ – диз'юнктивна нормальна форма;
- КНФ – кон'юнктивна нормальна форма;
- ДДНФ – досконала диз'юнктивна нормальна форма;
- ДКНФ – досконала кон'юнктивна нормальна форма.

Дві останні форми легко одержати за допомогою таблиці істинності.

5. За допомогою логічних виразів у будь-якій формі неважко побудувати електронний цифровий пристрій, використовуючи електронні логічні елементи АБО, І, НІ, АБО-НІ, І-НІ. Ці пристрої не будуть оптимальними в сенсі кількості елементів, числа міжелементних з'єднань, а, отже, і в сенсі вартості та швидкодії.

6. З метою зменшення кількості елементів і числа міжелементних з'єднань проводиться процедура мінімізації логічної формули функціонування алгоритму цифрового пристрою. Нині використовуються кілька методів такої мінімізації. При ручній мінімізації (без використання комп'ютерних програм) найбільшу популярність набув метод карт Карно як найбільш простий і очевидний. Його доцільно використовувати, якщо кількість аргументів логічного вираження, що мінімізується, не перевищує п'яти-шести.