**Лабораторна робота №1-2**

**Логічні елементи. Мінімізація перемикальних функцій**

**Ціль роботи:** дослідити роботу перемикальних функцій. Використовуючи Діаграму Вейча мінімізувати функцію та навчитися будувати комбінаційні схеми різної складності.

**Короткі теоретичні відомості**

На логічному та функціональному рівнях опису комп’ютер є системою, до складу якої входять відповідні функціональні вузли, блоки та пристрої.

Будь-який функціональний вузол (схему, пристрій) можна вважати перетворювачем інформації, що має n входів та m виходів (рис. 1), на входи якого в моменти дискретного (автоматного) часу надходять вхідні (логічні) послідовності *xn xn–1 … x1*, а значення на кожному виході *yj ( j = 1, 2, …, m )* є функцією від вхідних двійкових послідовностей: *yj = fj(xn, xn–1, …, x1).* Функції *y1, y2, …, ym* називають перемикальними.



Рис. 1. Умовно графічне позначення перетворювача інформації

Функція *y = f (xn, xn–1, …, x1)* називається *перемикальною* (логічною, булевою), якщо сама функція y і кожен з її аргументів *xi (і = 1, 2, …, n)* можуть набувати лише двох значень: 0 або 1, тобто якщо *xi, y ∈ {0, 1}.* Отже, поведінку схеми (вузла, пристрою) можна описати системою перемикальних функцій:

(1)

Перемикальну функцію можна задати: таблицею істинності, словесним описом (вербально), геометричним зображенням, аналітично (формулою).

Найпоширенішим є спосіб задання перемикальної функції таблицею істинності (табл. 1), у якій перелічують всі можливі набори значень аргументів функції та вказують, яке саме значення функція набуває на відповідному наборі. Набором (кортежем) називають упорядковану послідовність значень аргументів. Кожному набору відповідає десятковий номер, який є кількісним еквівалентом двійкового числа, утвореного значеннями аргументів.

Таблиця 1. Таблиця істинності перемикальної функції y = f(x3, x2, x1)



Перевагами табличного задання перемикальної функції є наочність та універсальність – у вигляді таблиці істинності можна задати функцію від довільної кількості змінних (аргументів).

Розглянемо логічні елементи та перемикальні функції, що мають найбільше практичне значення. В таблиці 2 та 3 представлені функції одного та двох аргументів відповідно.

Таблиця 2. Функція одного аргументу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Табл. істинності | Назва ф-ї | Запис | УГП логічного елемента | Назва логічного елемента |
| |  |  | | --- | --- | | x | y | | 0 | 0 | | 1 | 1 | | Повторення, ТАК |  |  | Повторювач |
| |  |  | | --- | --- | | x | y | | 0 | 1 | | 1 | 0 | | Інверсія, заперечення, НІ |  |  | Інвертор |

Таблиця 3. Функція двох аргументів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Табл. істинності | Назва ф-ї | Запис | УГП логічного елемента | Назва логічного елемента |
| |  |  |  | | --- | --- | --- | | X2 | X1 | y | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | | Кон’юнкція, І, логічний добуток |  |  | Елемент І, кон’юнктор |
| |  |  |  | | --- | --- | --- | | X2 | X1 | y | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | | Диз’юнкція, АБО, логічний сума |  |  | Елемент АБО, диз’юнктор |
| |  |  |  | | --- | --- | --- | | X2 | X1 | y | | 0 | 0 | 1 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 0 | | І-НЕ,  ф-я Шефера |  |  | Елемент  І-НЕ,  елемент Шефера |
| |  |  |  | | --- | --- | --- | | X2 | X1 | y | | 0 | 0 | 1 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 0 | | АБО-НЕ,  ф-я Пірса |  |  | Елемент АБО-НЕ,  елемент Пірса |
| |  |  |  | | --- | --- | --- | | X2 | X1 | y | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 0 | | ВИКЛЮЧНЕ АБО,  сума по модулю два |  |  | Елемент ВИКЛЮЧНЕ АБО |

Перемикальній функції відповідає логічна схема, що є сукупністю відповідним чином з’єднаних логічних елементів. Логічна схема є апаратною реалізацією перемикальної функції. І навпаки, якщо відома логічна система, то її поведінку можна описати системою перемикальних функцій (кожному виходу логічної схеми відповідає одна перемикальна функція).

Розрізняють два види логічних схем – комбінаційні схеми та послідовнісні схеми. Якщо сукупність вихідних сигналів логічної схеми з n входами і m виходами в даний момент часу повністю визначається сукупністю вхідних сигналів у цей самий момент часу і не залежить від вхідних сигналів, що діяли на входах в попередні моменти часу, то таку логічну схему називають *комбінаційною схемою*. Вважають, що комбінаційна схема має один стан. Поведінку комбінаційної схеми можна описати системою перемикальних функцій (1).

Якщо сукупність вихідних сигналів логічної схеми в даний момент часу залежить не лише від вхідних сигналів, що діють у цей самий момент часу, але й від стану, в якому перебуває схема (тобто й від сигналів, що діяли в попередній момент часу), то таку логічну схему називають *послідовнісною схемою* або схемою з пам’яттю (рис. 2).

|  |  |
| --- | --- |
|  |  |

а б

Рис. 2. Логічні схеми: а – комбінаційна; б – послідовнісна

Основною ознакою послідовнісної схеми (схеми з пам’яттю) є наявність у ній петель. *Петля* – це електричне з’єднання від виходу деякого логічного елемента до одного з входів цього самого елемента (можливо й через інші логічні елементи). У комбінаційних схемах петлі відсутні.

Як комбінаційні, так і послідовнісні схеми називають *цифровими автоматами*. При цьому комбінаційні схеми є *тривіальними автоматами* (тобто автоматами, що не мають пам’яті), а послідовнісні схеми є *автоматами з пам’яттю.*

*Канонічні форми алгебри Буля.*

Для перетворення аргументів (змінних) та виразів в алгебрі Буля використовують 3 логічні операції (функції) – І, АБО, НЕ (AND, OR, NOT). Операція НЕ – одномісна, операції І, АБО – n-місні.

Розглянемо основні закони (властивості) алгебри Буля.

1. *Закон комутативності⋅ (переставний закон):*

На практиці закон комутативності означає, що всі входи логічних елементів І та елементів АБО рівноцінні, і вхідні змінні можна подавати на будь-які входи цих елементів.x

1. *Закон асоціативності(сполучний закон):*

На практиці властивість асоціативності (розставляння дужок) означає, що за рахунок каскадування (каскадного з’єднання) логічних елементів можна реалізувати функції І та АБО від довільної кількості аргументів, використовуючи 2-входові логічні елементи І, АБО

1. *Закон абсорбції (поглинання):*
2. *Закон склеювання:*
3. Закони (правила) де Моргана (De Morgan’s laws) встановлюють зв'язок між функціями (операціями) І, АБО, І-НЕ, АБО-НЕ:

1) (NOR→AND) – заперечення диз’юнкції еквівалентне кон’юнкції заперечень,

2) (NAND→OR) – заперечення кон’юнкції еквівалентне диз’юнкції заперечень,

3) (OR→NAND) – диз’юнкція еквівалентна кон’юнкції заперечень із загальним запереченням,

4) (AND→NOR) – кон’юнкція еквівалентна диз’юнкції заперечень із загальним запереченням.

Наслідком правил де Моргана на практиці є те, що в логічних схемах елемент АБО можна замінити на елемент І-НЕ з інверсними входами (і навпаки), а елемент І – на елемент АБО-НЕ з інверсними входами (і навпаки).

При виконанні перетворень в аналітичних виразах булевої алгебри необхідно дотримуватись пріоритету операцій: спочатку виконують операції в дужках, далі найвищий пріоритет має операція заперечення, потім кон’юнкція, а диз’юнкція має найнижчий пріоритет. Пріоритет операцій слід враховувати при застосуванні правил де Моргана.

Будь-яку перемикальну функцію можна подати в досконалій диз’юнктивній нормальній формі. Досконала диз’юнктивна нормальна форма функції (**ДДНФ**) – це диз’юнкція конституент одиниці (мінтермів), що відповідають наборам, на яких перемикальна функція набуває значення 1.

ДДНФ – називають також формою І/АБО (AND/OR), підкреслюючи що в ДДНФ внутрішньою функцією є І, а зовнішньою – АБО.

Досконала кон’юнктивна нормальна форма функції (**ДКНФ**) – це кон’юнкція контитуент нуля (макстермів), що відповідають наборам, на яких перемикальна функція набуває значення 0.

ДКНФ називають також формою АБО/І (OR/AND), де АБО – внутрішня функція, а І – зовнішня.

Наприклад, для ДДНФ внутрішньою функцією є кон’юнкція, а зовнішньою – диз’юнкція, тобто ДДНФ є диз’юнкцією кон’юнкцій; для ДКНФ внутрішньою функцією є диз’юнкція, а зовнішньою – кон’юнкція, тобто ДКНФ є кон’юнкцією диз’юнкцій. На практиці властивість нормальності канонічної форми означає, що якщо немає обмежень на кількість входів логічних елементів, то комбінаційна схема, що реалізує нормальну форму, має два каскади (стовпці) логічних елементів.

**Графічний метод мінімізації перемикальних функцій**

Особливістю аналітичних методів мінімізації перемикальних функцій – Квайна, Квайна – Мак-Класкі, невизначених коефіцієнтів, Блейка-Порецького, є їх двоетапність – на першому етапі визначають скорочену ДНФ, а на другому – знаходять тупикові ДНФ, з яких вибирають мінімальну ДНФ. Ці методи є універсальними – їх можна застосувати для мінімізації перемикальних функцій від довільної кількості змінних, поданих в диз’юнктивній формі (найчастіше – в ДДНФ).

Якщо перемикальна функція залежить від відносно невеликої кількості змінних (до 7-ми), то замість аналітичних застосовують спрощені методи, що дістали назву графічних методів мінімізації перемикальних функцій. Розрізняють два графічні методи мінімізації перемикальних функцій:

• метод мінімізації на основі діаграм Вейча;

• метод мінімізації на основі карт Карно.

Перевагами обох методів графічної мінімізації є:

1) наочність – для отримання мінімальної форми функції використовується лише одна таблиця;

2) простота та швидкість отримання результату – мінімальну форму функції отримують з таблиці одразу, минаючи етап знаходження скороченої ДНФ та етап визначення тупикових ДНФ функції;

3) з таблиці можна отримати як мінімальну ДНФ, так і мінімальну КНФ функції.

Розглянемо метод мінімізації перемикальних функцій на основі **діаграм Вейча**. Діаграма Вейча перемикальної функції від n змінних є прямокутною (якщо n – непарне) або квадратною (якщо n – парне) таблицею, що складається з 2n клітинок (рис. 3). Така таблиця є аналогом таблиці істинності перемикальної функції. Кожній клітинці на діаграмі Вейча відповідає двійковий набір, для якого може бути записана конституента одиниці або конституента нуля. Якщо на деякому наборі аргументів функція дорівнює 1, то у клітинку, що відповідає даному набору, записують 1. Клітинки, що відповідають наборам, на яких функція дорівнює 0, або заповнюють нулями, або залишають незаповненими.

  

Рис. 3 Діаграма Вейча перемикальної функції від двох, трьох та чотирьох змінних відповідно (0, 1,...,15 – номери двійкових наборів)

**Приклад 1.** Мінімізувати перемикальну функцію ***y*** від трьох змінних, задану таблицею істинності.



**Розв’язування.** Заповнимо діаграму Вейча функції – одиниці слід розташувати в клітинках з номерами 2, 3, 4, 6 та 7, нулі – в решті клітинках. Об’єднаємо одиниці в прямокутники (групи) максимально можливого розміру так, щоб покрити всі одиниці на діаграмі.



Таких груп – дві: першу утворюють чотири клітинки з номерами 6, 7, 2, 3; другу – дві клітинки з номерами 6, 4. Першій групі клітинок відповідає терм (імпліканта) X2 (вона є спільною для всіх 4-х клітинок), другій – терм (імпліканта) .

Тому мінімальною ДНФ функції є:

**Завдання до лабораторної роботи**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x*1 | *x*2 | *x*3 | *y*, згідно з номером варіанту (1,2,…,30 – порядковий номер в списку групи!!!) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Елемент І, Елемент НІ | | | + | + | + | + | + | + | + | + | + | + | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Ел-т Пірса  (2АБО-НІ) | | | - | - | - | - | - | - | - | - | - | - | + | + | + | + | + | + | + | + | + | + | - | - | - | - | - | - | - | - | - | - |
| Ел-т АБО,  Елемент НІ | | | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | + | + | + | + | + | + | + | + | + | + |

**Порядок виконання роботи**

1. Дослідити елементи, що реалізують базові функції алгебри логіки: інвертор, диз’юнктор, кон’юнктор. Використовуючи логічний конвертор довести, що функції для обох систем потенціалів дають одну й ту саму таблицю дійсності, а значить і реалізуються одним і тим же елементом.
2. Дослідити елементи Шеффера та Пірса, що реалізують універсальні функції. Довести, що за допомогою кожного з них можна реалізувати будь-яку з базових функцій (можна використовувати декілька однотипних елементів).
3. Для наведеної таблиці істинності (**Завдання до лабораторної роботи**, згідно з номером варіанту) реалізувати функцію за допомогою елементів заданого типу (можна використовувати будь- яку кількість елементів, але тільки заданих типів).
4. Використовуючи аксіоми та закони алгебри Буля спростити заданий вираз та представити схему.
5. Провести порівняльний аналіз комбінаційних схем до спрощення та після.
6. Використовуючи Діаграму Вейча мінімізувати задану функцію.

**Додаткове завдання**

Самостійно дослідити алгебру Шеффера, алгебру Пірса, алгебру Жегалкіна