Міністерство освіти і науки України
Державний вищий навчальний заклад
„Чернівецький індустріальний коледж”

КОМП’ЮТЕРНА ЛОГІКА

навчальний посібник з дисципліни

Чернівці
2018

Комп’ютерна логіка: навчальний посібник з дисципліни для III курсу спеціальності 123 «Комп’ютерна інженерія» / Муринюк В.В. – Чернівці ДВНЗ”ЧІК”, 2018. – 60 с.

Укладач: Муринюк В.В. – викладач спецелектротехнічних дисциплін ДВНЗ «Чернівецький індустріальний коледж»

Рецензенти: Парфенюк О.А., доктор фізико-математичних наук, професор кафедри електроніки і енергетики Чернівецького національного університету імені Юрія Федьковича;

Гандабура М.С.,голова циклової комісії комп'ютерної інженерії ДВНЗ «Чернівецький індустріальний коледж»;

Розглянуто на засіданні циклової комісії комп'ютерної інженерії
Протокол від «09» січня 2018р. № 5

Схвалено Методичною радою Чернівецького індустріального коледжу. Протокол від «___» _____________2018р. №__

Зміст

Розділ 1. Арифметичні основи ЕОМ.

Розділ 2. Логічні основи ЕОМ

Розділ 1. Арифметичні основи ЕОМ

В розділі розглянуто загальні відомості про системи числення, позиційні системи числення, які застосовуються в ЕОМ, алгоритми перетворення цілих чисел, правильних дробів та змішаних десяткових чисел в системи з іншою основою, формати представлення даних і розміщення інформації в оперативній пам'яті ЕОМ, машинні коди чисел, алгоритми виконання операцій додавання та віднімання в арифметико-логічному пристрої ЕОМ, алгоритми множення та ділення двійкових чисел в АЛП.

Після вивчення матеріалу розділу Ви будете:

Знати	▪ *основні відомості про системи числення* ▪ *алгоритми перетворення цілих чисел, правильних дробів та змішаних десяткових чисел в системи з іншою основою* ▪ *формати представлення даних* ▪ *машинні коди чисел* ▪ *алгоритми виконання операцій додавання та віднімання в арифметико-логічному пристрої ЕОМ* ▪ *алгоритми множення та ділення двійкових чисел в АЛП*
Вміти	– *переводити числа з однієї системи числення в іншу* – *подавати двійкові числа у формі із плаваючою комою* – *подавати двійкові числа у формі із фіксованою комою* – *утворювати машинні коди чисел* – *здійснювати кодування інформації в ЕОМ;* – *виконувати арифметичні дії з двійковими числами;*

Ключові поняття та терміни	
• *система числення* • *основа системи числення* • *позиційні системи числення* • *непозиційні системи числення* • *розрядна сітка* • *з фіксованою комою* • *з «плаваючою» комою*	• *прямий код* • *зворотній код* • *додатковий код* • *модифікований код* • *порушення нормалізації.* • *переповнення розрядної сітки*

Тема 1.1. Загальні відомості про системи числення

Вступ

Актуальні завдання розвитку обчислювальної техніки, пов'язані з розвитком всіх галузей промисловості, вимагають застосування цифрової обробки інформації та використання сучасних інформаційних технологій. Обчислювальні пристрої – комп'ютери та калькулятори, системи контролю та сигналізації, системи керування промислових механізмів та установок створені за єдиними принципами.

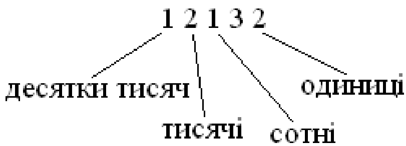
Мета предмету: вивчення принципів, за якими створюються обчислювальні пристрої.

Комп'ютер безпосередньо не сприймає десяткові числа. Стан електронних елементів будь-яких обчислювальних пристроїв (комп'ютерів) залежить від того, проходить чи ні в даний момент через них електричний струм. Ці два стани можна позначити як 0 і 1. тобто вся інформація, яку обробляє комп'ютер, повинна бути закодована тільки цифрами 0 та 1.

Десяткові числа спочатку переводяться в їхній аналог, представлений за допомогою цифр 0 і 1, і лише після цього обробляється. А потім для представлення результату в зручній для користувача формі, відбувається зворотній процес.

Основні відомості про системи числення

Система числення – це сукупність правил для позначення (запису) чисел за допомогою цифр, знаків. Для запису чисел в конкретній системі числення використовується деякий алфавіт, що складається цифр або інших символів.



Розрізняють два види систем числення: **позиційні і непозиційні**

Кількість символів, за допомогою яких можна записати будь-яке число в даній системі числення, називається **основою системи числення (S**). В різних системах числення зустрічається однакові записи чисел, але значення їхнє в різних системах різне. Для того щоб визначити, в якій системі числення писане дане число будемо вказувати індексом систему числення: 5_8, 10_{16}, 101_2, 18_{10}.

Оскільки першою цифрою у будь-якій системі числення є нуль, то остання можлива цифра в алфавіті системи завжди на одиницю менша за основу цієї системи: Наприклад:

$$?_{10} = 0,1,2,3,4,5,6,7,8,9$$

$$?_2 = 0,1$$

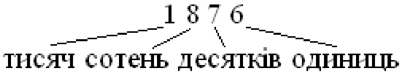
$$?_8 = 0,1,2,3,4,5,6,7\ .$$

Існує закономірність: $8_{10} = 10_8$; $5_{10} = 10_5$; $2_{10} = 10_2$

<u>Позиційні системи числення.</u>

<u>Позиційні</u> системи числення – це системи, в яких „вага" кожної цифри в числі залежить від її місцезнаходження в записі цього числа. Розглянемо десяткову систему числення.

Візьмемо число 555. Найправіша цифра 5 означає 5 одиниць. Найлівіша – 500 одиниць. Тобто „вага” кожної із цих п’ятірок зовсім різна. Позиції цифр в запису числа називаються розрядами. Тут „вага” кожного розряду в 10 разів більша від „ваги” попереднього. Наприклад



Значить кожне число в десятковій системі числення можна представити в вигляді суми різних цілих степенів числа 10 з відповідними коефіцієнтами a_i (0,1,2…9), взятими з алфавіту даної системи.

Наприклад: $245{,}83 = 2\cdot 10^2 + 4\cdot 10^1 + 5\cdot 10^0 + 8\cdot 10^{-1} + 3\cdot 10^{-2}$

Тобто записують тільки коефіцієнти при степенях.

Для представлення чисел в системах числення з основою S>10 недостатньо цифр арабських (0,1,…,9). Тому їх доповнюють іншими символами, знаками. Так наприклад для шістнадцяткової системи вводять:

10 – A, 11 – B, 12 – C, 13 – D, 14 – E, 15 – F.

Наприклад

$Å327 \rightarrow 14327$

Непозиційні системи числення.

Непозиційні_системи числення – це системи, в яких „вага” (кількісний еквівалент) кожної цифри в числі не залежить від її місцезнаходження в запису даного числа (див.мал).

Запис чисел паличками, хрестиками чи іншими символами, кожен з яких є еквівалентом одиниці, може бути прикладом непозиційної системи числення. Така система числення є найдревнішою.

У римській системі числення для запису різних цілих чисел використовують символи I, V, X, L, C, D, M і т.д., які позначають відповідно один, п’ять, десять, п’ятдесят, сто, п’ятсот, тисяча і т.д. Наприклад, запис у римській системі числення MCMLXXXV означає число 1985. Ця система також є непозиційною, оскільки в ній значення цифр не залежить від їх позиції в ряді інших цифр.

Загальним недоліком непозиційних систем є складність представлення в них достатньо великих чисел, оскільки при цьому отримується надзвичайно громіздкий запис чисел або потрібен великий алфавіт цифр, що використовуються. В зв’язку з цим в ЕОМ застосовують лише позиційні системи числення, у яких кількісний еквівалент кожної цифри алфавіту залежить не тільки від вигляду цієї цифри, але і від її місцерозміщення у запису чисел.

Таблиця 1. Представлення чисел у різних системах числення

Система числення				
Десяткова	**Двійкова**	**Вісімкова**	**Шістнад-цяткова**	**Двійково-десяткова**
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	00010000
11	1011	13	B	00010001
12	1100	14	C	00010010
13	1101	15	D	00010011
14	1110	16	E	00010100
15	1111	17	F	00010101
16	10000	20	10	00010110
17	10001	21	11	00010111
18	10010	22	12	00011000
19	10011	23	13	00011001
20	10100	24	14	00100000

Тема 1.2. Позиційні системи числення, які застосовуються в ЕОМ

Двійкова система числення

Основа (S) – 2 тобто 1 і 0.

Будь-яке дійсне число в двійковій системі представляють у вигляді суми цілих степенів її основи S = 2 помножених на відповідні коефіцієнти – 0 або 1, тобто

$$N_2 \sum_{i=-n}^{m} a_i \cdot 2^i$$

За цією формулою можна представити в двійковій системі,

наприклад такі числа:

$$27{,}25 = 16 + 8 + 2 + 1 + 0{,}25$$

$$27{,}25 = \underbrace{\underbrace{1\cdot 2^{4}}_{16} + \underbrace{1\cdot 2^{3}}_{8} + \underbrace{0\cdot 2^{2}}_{0} + \underbrace{1\cdot 2^{1}}_{2} + \underbrace{1\cdot 2^{0}}_{1}}_{27} + \underbrace{\underbrace{0\cdot 2^{-1}}_{0} + \underbrace{1\cdot 2^{-2}}_{1/4}}_{0{,}25}$$

$$27{,}25_{10} = 11011{,}01_{2}$$

Таблиця 2. Таблиця степенів числа 2.

2^{-5}	2^{-4}	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}	2^{6}	2^{7}	2^{8}
0,03125	0,0625	0,125	0,25	0,5	1	2	4	8	16	32	64	128	256

2^{9}	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
512	1024	2048	4096	8192	16384

Переваги двійкової системи:

Простота конструкції арифметичних і запам’ятовуючих пристроїв.

Можливість застосувати апарат математичної логіки при проектуванні функціональних схем та при розв’язування логічних задач.

Недоліки: 1. Громіздкість записів (двійковий запис числа як мінімум в 3,32 рази довший за десятковий).

2. Попередньо треба переводити всі вихідні дані з десяткової системи в двійкову, а потім навпаки.

Реалізація існуючих систем числення з основою більшою ніж 2 зазвичай досягається представленням цифр цієї системи групою двійкових цифр. Найбільшими можливостями в цьому напрямку володіють системи числення з основою, яка являє собою цілу степінь числа 2. З них використовують вісімкову і шістнадцяткову системи числення.

Вісімкова система числення.

Її використовують як допоміжну при підготовці задачі до розв’язування (в процесі програмування), при перевірці роботи машини і налаштуванні програми. В цій системі використовується вісім цифр від 0 до 7, а число представляється сумою цілих степенів основи S = 8 помножених на відповідні коефіцієнти, тобто $N_8 = \sum_{i=-n}^{m} a_i 8^i$,

де $i = 0 - 7$.

Тоді десяткове число 215 можна представити:

$$\underbrace{215_{10} = \underbrace{3 \cdot 8^2}_{192} + \underbrace{2 \cdot 8^1}_{16} + \underbrace{7 \cdot 8^0}_{7}}_{215} = 327_8$$

$$\underbrace{1386_{10} = \underbrace{2 \cdot 8^3}_{1024} + \underbrace{5 \cdot 8^2}_{320} + \underbrace{5 \cdot 8^1}_{40} + \underbrace{2 \cdot 8^0}_{2}}_{1386} = 2552_8$$

Вісімкова система числення зручна тим, що від неї легко можна перейти до двійкової системи.

Таблиця 3. Таблиця степенів числа 8.

8^0	8^1	8^2	8^3	8^4	8^5
1	8	64	512	4096	32768

Шістнадцяткова система числення.

Так само як і вісімкова ця система має основу, що дорівнює цілому степеню числа 2. Ця система також використовується як проміжна для внутрішнього запису чисел і команд програми.

В цій системі числі, як і в інших позиційних системах числення, теж можна представити сумою цілих степенів основи S = 16 помножених на відповідні коефіцієнти з алфавіту, тобто

$N_{16} = \sum_{i=-n}^{m} a_i 16^i$, де $i = 0,1,...9, A, B,..., E, F$.

Таблиця 4. Таблиця степенів числа 16.

16^0	16^1	16^2	16^3	16^4	16^5
1	16	256	4096	65536	1048576

$$215_{10} = \underbrace{D\cdot 16^1}_{13\cdot 16 = 208} + \underbrace{7\cdot 16^0}_{7\cdot 1 = 7} = D7_{16}$$

$$17_{10} = 1\cdot 16^1 + 1\cdot 16^0 = 11_{16}$$

$$18_{10} = 1\cdot 16^1 + 2\cdot 16^0 = 12_{16}$$

$$26_{10} = 1\cdot 16^1 + A\cdot 16^0 = 1A_{16}$$

$$27_{10} = 1\cdot 16^1 + B\cdot 16^0 = 1B_{16}$$

Двійково-десяткова система числення.

Числа в ній записують наступним чином:

За основу беруть десятковий запис числа, а потім кожну його цифру (від 0 до 9) записують в вигляді чотирьохрозрядного двійкового числа, що називається тетрадою.

Наприклад: 746,95

$$\frac{7}{0111}\frac{4}{0100}\frac{6}{0110},\frac{9}{1001}\frac{5}{0101}$$

Таким чином 746,95 відповідає двійково-десятковому коду 011101000110,10010101

Перевід навпаки з двійково-десяткової в десяткову систему теж простий. Треба число вліво і вправо від коми розбити на четвірки цифр – тетради і замінити кожну тетраду відповідною десятковою цифрою.

Тема 1.3. Алгоритми перетворення цілих чисел, правильних дробів та змішаних десяткових чисел в системи з іншою основою

Табличний спосіб перетворення десяткових чисел в систему з іншою основою.

Найдоступнішим і таким, що не потребує складних розрахунків, способом переводу чисел з десяткової системи числення в систему з новою основою є так званий табличний спосіб. Його суть полягає в тому, що вихідне десяткове число представляють у вигляді суми різних степенів нової основи. Ці степені підбирають з відповідної таблиці (звідси й назва способу). Послідовність розміщення цих степенів починається з тієї, яка дає число, що не перевищує (менше чи рівне) вихідне десяткове число. Потім для членів отриманого ряду степенів підбирають відповідні коефіцієнти таким чином, щоб їх сума дорівнювала вихідному десятковому числу.

Розглянемо цей спосіб на декількох прикладах. Нехай потрібно перевести десяткове число N_{10} = 377 у двійкову систему числення. Найбільший степінь нової основи S = 2, яка не перевищує число 377, дорівнює 8, тому що $2^8 = 256$. Отже, послідовність розміщення степенів починається з числа 2^8 і закінчиться числом 2^0. З цих членів підбирають ті, котрі будуть сумуватися для отримання результату, рівного 377. Потім перед членами, які приймали участь в сумуванні, ставимо коефіцієнт 1, а перед тими, які не приймали участі – 0. Кінцевий запис двійкового числа буде складатися з цих коефіцієнтів:

$$377_{10} = 1\cdot2^8 + 0\cdot2^7 + 1\cdot2^6 + 1\cdot2^5 + 1\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0 = 101111001_2.$$

$$256 + 0 + 64 + 32 + 16 + 8 + 0 + 0 + 1 = 377_{10}$$

Виконаємо тепер перевід цього ж чила 377_{10} у вісімкову систему числення. Найбільша степінь восьми, що не перевищує 377, дорівнює 2. Отже, послідовність степенів буде складатися з 8^2, 8^1, 8^0. Підбираючи до них коефіцієнти із алфавіту вісімкової системи числення (від 0 до 7), отримаємо:

$$377_{10} = 5\cdot8^2 + 7\cdot8^1 + 1\cdot8^0 = 571_8.$$

$$320 + 56 + 1 = 377_{10}$$

Аналогічно виконаємо перевід числа 377 в шістнадцяткову систему:

$$377_{10} = 1\cdot16^2 + 7\cdot16^1 + 9\cdot16^0 = 179_{16}.$$

$$256 + 112 + 9 = 377_{10}.$$

Слід зазначити, що вісімкові і шістнадцяткові числа легко переводяться в двійкову систему числення, завдяки чому ці системи використовують в якості проміжного при переводі з десяткової системи числення в двійкову. Це пояснюється тим, що їх основи (S = 8, S = 16) відповідають цілим степеням числа два ($2^3 = 8$, $2^4 = 16$).

Для переводу вісімкових чи шістнадцяткових чисел у двійкову систему числення достатньо кожну їхню цифру замінити відповідно трьох- чи чотирьохрозрядним двійковим числом. Переконаємось у цьому при переводі вісмікового числа 571_8 і шістнадцяткового числа 179_{16} у двійкову систему числення.

$$\begin{array}{cccccccc} 5 & 7 & 1 & \quad & 1 & 7 & 9 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 101 & 111 & 001 & & 0001 & 0111 & 1001 \end{array}.$$

В обох випадках отримаємо однаковий результат $571_8 = 179_{16} = 101111001_2$, який співпадає з тим, що раніше був отриманий шляхом безпосереднього переводу в двійкову систему числення вихідного десяткового числа 377_{10}.

$377_{10} = 101111001_2$ $\qquad$ $0{,}3125_{10} = 0{,}24_8$

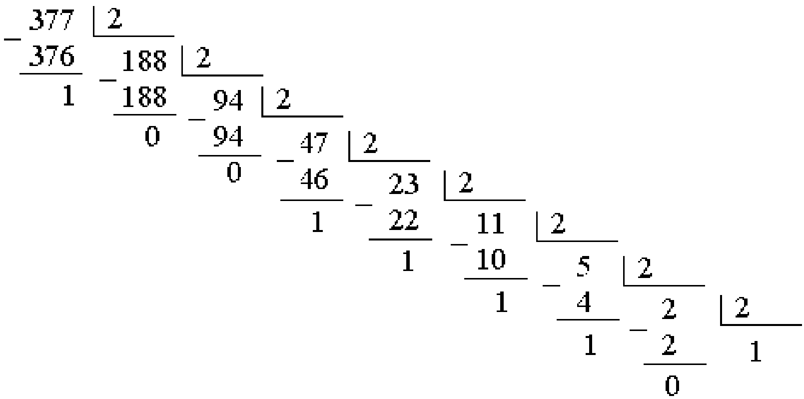
$377_{10} = 571_8$ $\quad$ $0,3125_{10} = 0,5_{16}$

$377_{10} = 179_{16}$ $\quad$ $0,3125_{10} = 0,0101_2$

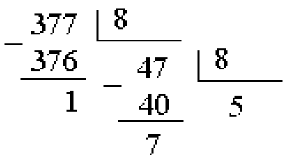
Переведення цілих десяткових чисел в іншу систему числення способом ділення

Для переводу цілого числа з десяткової системи числення в іншу систему з основою S треба це число послідовно ділити на основу S нової системи доти поки не отримаємо ділене менше від S.

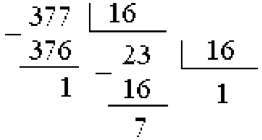
Число в новій системі запишеться в вигляді остач ділення, починаючи з останньої. Ця остання остача дає цифру старшого розряду в новій системі числення. Ділення виконують у вихідній системі числення.



Тобто $377_{10} = 101111001_2$



Тобто $377_{10} = 571_8$



Тобто $377_{10} = 179_{16}$

Переведення дробів з десяткової системи числення в іншу.

Щоб перевести правильний дріб (0,…) із десяткової системи числення в іншу, потрібно цей дріб послідовно множити на основу тієї системи, в яку він переводиться. При цьому перемножуються тільки дробові частини. Дріб в новій системі запишеться у вигляді цілих частин отриманих добутків починаючи з першої

$\begin{array}{r} \times\; 0,6875 \\ 2 \\ \hline \times\; 1,3750 \\ 2 \\ \hline \times\; 0,7500 \\ 2 \\ \hline \times\; 1,5000 \\ 2 \\ \hline 1,0000 \end{array}$ Тобто $0,6875_{10} = 0,1011_2$	$\begin{array}{r} \times\; 0,6875 \\ 8 \\ \hline \times\; 5,5000 \\ 8 \\ \hline 4,0000 \end{array}$ Тобто $0,6875_{10} = 0,54_8$	$\begin{array}{r} \times\; 0,6875 \\ 16 \\ \hline 41250 \\ 6875 \\ \hline 11.0000 \end{array}$ Тобто $0,6875_{10} = 0,B_{16}$

Тепер $0,6875_{10} = 0,1011_2 = 0,54_8 = 0,B_{16}$

При переводі змішаних десяткових чисел (наприклад 15,6875) необхідно, користуючись розглянутими правилами виконати окремо перевід цілої і дробової частин:

$$15,6875 = 15 + 0,6875$$

$$15_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1111_2$$

$0,6875_{10} = 0,1011$ тому $15,6875_{10} = 1111,1011_2$

Подібним чином виконаємо перевід цього ж дробу в вісімкову і шістнадцяткову системи числення:

$\begin{array}{r} 0,\quad 6875 \\ \text{x } 8 \\ \hline 5,\quad 5000 \\ \text{x } 8 \\ \hline 4,\quad 0000 \end{array}$	$\begin{array}{r} 0,\quad 6875 \\ 16 \\ \hline 11,\quad 0000 \end{array}$

У шістнадцятковій системі ціла частина добутку 11 запишеться символом В, тому отримаємо

$0,6875_{10} = 0,1011_2 = 0,54_8 = 0,B_{16}$.

Особливості двійково-десяткового кодування.

При переводі десяткових чисел в двійкову в якості проміжної застосовують двійкову-десяткову систему числення. Числа в цій системі записують наступним чином. За основу беруть десятковий запис числа, а потім кожну його цифру (від 0 до 9) записують у вигляді чотирьохрозрядного двійкового числа, що називається тетрадою. Таким чином, двійково-десятковий запис числа відрізняється від його десяткового запису тим, що для зображення кожної цифри застосовують не один знак, а чотири знаки.

Щоб дане десяткове число записати в двійково-десятковій системі, необхідно кожну його цифру замінити відповідною двійковою тетрадою. Візьмем, наприклад, десяткове число 746,95 і запишемо його в двійково-десятковій системі. Перепишемо спочатку це число з великими інтервалами між цифрами, а потім під кожною цифрою поставимо відповідну їй двійкову тетраду:

7 4 6 9 5
↓ ↓ ↓ ↓ ↓
0111 0100 0110 1001 0101

Таким чином, десяткове число 746,95 відповідає двійково-десятковому числу 011101000110,10010101.

Перевід чисел з десяткової системи у двійково-десяткову не пов’язаний з розрахунками. Тому можна наперед закодувати десяткові числа їх двійковими еквівалентами і переводити числа з десяткової систем в двійково-десяткову суто механічно.

Зворотній перевід числа з двійково-десяткової системи в десяткову також простий. Для цього двійково-десяткове число ліворуч і праворуч від коми необхідно розбити на четвірки цифр - тетради, а потім кожну тетраду замінити відповдною десятковою цифрою. Наприклад, маємо двійково-десяткове число 100101100101,01110001. Розіб’ємо його на тетради і, замінивши кожну тетраду десятковою цифрою, отримаємо

1001 0110 0101, 0111 0001
—— —— —— —— ------ = 965,71.
9 6 5 7 1

Перед вводом вихідних даних в машину десяткові числа спеціальними пристроями перетворюються в двійково-десяткові, які потім за спеціальною програмою переводяться самою машиною в двійкові. Після закінчення розрахунків машина автоматично переводить результат з двійкової в двійково-десяткову систему, після чого спеціальні пристрої видають кінцевий результат в десятковій системі. Слід зазначити, що існують машини, в яких обчислювальні операції здійснюються в двійково-десятковій системі.

Тема 1.4.Формати представлення даних і розміщення інформації в оперативній пам’яті ЕОМ

Двійкові числа в обчислювальних пристроях розміщуються у комірках пам'яті, причому для кожного розряду числа виділяється окрема комірка, що зберігає один біт інформації. Сукупність комірок, призна-чених для розміщення одного двійкового числа, називають ***розрядною сіткою***. Довжина розрядної сітки (число комірок ***n*** у розрядній сітці) обмежена і залежить від конструктивних особливостей обчислювального пристрою. Більшість існуючих електронних обчислювальних пристроїв мають розрядні сітки, що містять 16, 32 або 64 комірок.

Розміщення розрядів числа у розрядній сітці може відбуватися різними способами. Спосіб розміщення визначається формою подання двійкових чисел у ЕОМ. ***Розрізняють дві форми подання двійкових чисел: із фіксованою комою і з «плавучою» комою.*** Іноді ці форми називають відповідно *природною і напівлогарифмічною*.

Припустимо, що в розрядній сітці необхідно розмістити двійкове число, що містить цілу і дробову частини. Якщо для розміщення цілої частини числа виділяється k комірок n-розрядної сітки, то (якщо не враховувати знак) для розміщення дробової частини залишиться n-k вільних комірок (рис. 1).

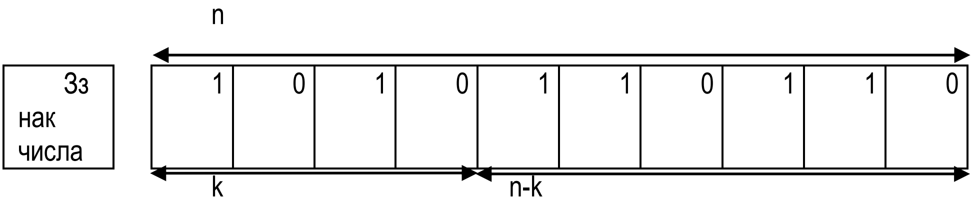


Рис. 1. Форма подання двійкових чисел із фіксованою комою.

Така форма подання двійкових чисел називається ***формою з фіксованою комою***. Дійсно, положення коми строго фіксовано стосовно розрядної сітки. Якщо кількість розрядів у дробовій частині числа перевищують n-k, то деякі молодші розряди виходять за межі розрядної сітки і не будуть сприйматися обчислювальним пристроєм. ***Отже, будь-яке двійкове число, менше ніж одиниця молодшого розряду розрядної сітки, сприймається як нуль і називається машинним нулем.***

У результаті відкидання молодших розрядів дробової частини числа, розташованої за межами розрядної сітки, виникає похибка подання. Максимальне значення абсолютної похибки подання не перевищує одиниці молодшого розряду сітки.

В універсальних ЕОМ форма з фіксованою комою, у зв'язку з властивою їй низькою точністю, застосовується лише для подання цілих чисел. Основною є форма подання чисел з «плаваючою» комою. Її використання дозволяє суттєво розширити діапазон і зменшити відносну похибку.

У цій формі числа подаються у вигляді суми деякого степеня основи системи числення (який називається характеристикою числа) і цифрової частини, що має вигляд правильного дробу:

$$N = \pm a q^{\pm p},$$

де p називають порядком числа, а правильний дріб a – його мантисою. Мантиса і порядок є знаковими числами. Тому для позначення знаків у розрядній сітці відводяться два додаткові розряди. Знак усього числа співпадає із знаком мантиси.

При запису двійкового числа у показовій формі, в розрядній сітці використовуються дві групи розрядів (без урахування знакових розрядів мантиси і порядку). Перша група (k розрядів) призначена для розміщення коду мантиси, друга (n-k розрядів) – для розміщення коду порядку (рис.2).

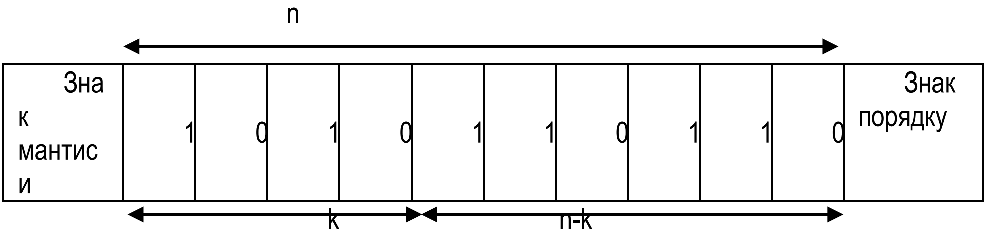


Рис.2. Форма подання двійкових чисел із „плаваючою" комою.

Отже, мантиса числа може мати необмежену кількість різних значень, менших за одиницю, при відповідних значеннях порядку (тобто кома може «плавати»). ***З усієї кількості подань числа у показовій формі те його подання, що не має в старшому розряді мантиси нуля, називають нормалізованим.*** Всі інші подання є ненормалізованими. У нормалізованій формі значення мантиси завжди більші або дорівнюють 1/2, але не перевищують одиниці.

У обчислювальних пристроях із «плаваючою» комою усі числа зберігаються у нормалізованому вигляді, при цьому не втрачаються молодші розряди мантиси і підвищується точність обчислень. Якщо після виконання будь-якої арифметичної операції результат виявляється ненормалізованим, то перед занесенням числа в пам'ять виконують його нормалізацію, тобто зсув мантиси ліворуч на відповідну кількість розрядів, і зменшення порядку числа на відповідну кількість одиниць.

Показова форма подання чисел має і свої вади, основною з яких є порівняно висока складність виконання арифметичних операцій, а отже, і більша вимогливість до ресурсів обчислювального пристрою. Це обмежує її застосування, наприклад, у спеціалізованих радіотехнічних обчислювальних пристроях, у системах управління технологічними процесами та обробки вимірювальної інформації у реальному часі.

Тема 1.5. Машинні коди чисел.

Технічна реалізація операції додавання в ЕОМ значно простіша, ніж операції віднімання. З цієї причини в ЕОМ застосовують тільки схеми додавання, а операцію віднімання замінюють додаванням з використанням спеціально побудованих кодів. Застосовують наступні коди чисел: прямий, зворотній, додатковий і модифіковані. Прямий код використовують для множення і ділення, а зворотній і додатковий коди - для заміни операції віднімання додаванням. Зображення додатніх чисел у всіх кодах збігаються, а від'ємних чисел відрізняються. Будемо розглядати способи кодування чисел за абсолютною величиною менші одиниці, тому що будь-яке інше число завжди можна представити у вигляді мантиси (за абсолютною величиною, меншою від одиниці) і цілого ступеня основи S=2.

***Прямий код*.** Число *X* у прямому коді позначається $[X]_{пр}$. Прямий код числа *X* отримують за наступним правилом: якщо $X= +0, X_1X_2, ..., X_n$, то $[X]_{пр} = 0, X_1,X_2,..., X_n$; якщо $X=-0, X_1X_2 ...,.X_n$, то $[X]_{пр} = 1, X_1X_2, ..., X_n$, де $X_1\ X_2\ ,..., X_n$ — двійкові розряди числа. Таким чином, прямий код двійкового числа збігається по зображенню з двійковим записом самого числа, але в знаковому розряді (на місці нуля цілих) ставиться 0, якщо число позитивне, і 1, якщо число негативне. Наприклад, маємо числа N_1= 0,101 і N_2 = -0,110. Потрібно їх представити в прямому коді. Якщо N_1 = 0,101, то $[N_1]_{пр}$ = 0,101; якщо N_2= -0,110, то $[N_2]_{пр}$=1, 110.

***Зворотній код*.** Число *X* у зворотньому коді позначається $[X]_{зв}$. Як уже відзначалося, зворотній код додатнього числа збігається з його прямим кодом, тобто при $X > 0$ $[X]_{зв}=[X]_{пр}=X$. Для від'ємного числа зворотній код отримують за наступним правилом: у знаковий розряд числа записується одиниця, а в цифрових розрядах нулі заміняються одиницями, а одиниці - нулями. Таким чином, якщо маємо від'ємне число $X = -0, X_1X_2,...X_n$, то його зображення в зворотному коді буде $[X]_{зв}= 1, \overline{X}_1,\overline{X}_2, ..., \overline{X}_n$, де $\overline{X}_i$ (i= 1,2,..., n) означає звертання (чи інверсію) цифрових розрядів, тобто $\overline{X}_i$ = I, якщо $X_i = 0$, і $\overline{X}_i = 0$, якщо $X_i = 1$. Наприклад, переведемо число -0,1001 у зворотній код: якщо X= -0,1001, то $[X]_{зв}$ = 1,0110. В ЕОМ при додаванні зворотніх кодів за відповідними правилами виходить зворотній код суми.

Додатковий код. Число *X* у додатковому коді позначається $[X]_{дод}$. Додатковий код додатнього числа збігається з його прямим кодом, тобто при $X > 0$ $[X]_{дод} = [X]_{пр}= X$. Для від'ємного числа додатковий код отримуємо за наступним правилом: у знаковому розряді записується одиниця, у всіх цифрових розрядах нулі замінюються одиницями, а одиниці нулями (аналогічно тому, як це виконується в зворотньому коді), після чого до молодшого розряду додається одиниця. Таким чином, для від'ємного числа $X = -0, X_1X_2,...,X_n$ додатковий код буде представлений у виді $[X]_{дод} =1, \overline{X}_1,\overline{X}_2, ..., \overline{X}_n$ +0,00...01, де $\overline{X}_i$ = 1, якщо $X_i = 0$, і $\overline{X}_i$ = 0, якщо $X_i = 1$.

Слід зазначити, що додатковий код є доповненням його прямого коду до двох (до основи системи числення). При цьому необхідно мати на увазі, що доповнення числа до двох можна одержати, беручи доповнення кож-ного розряду двійкового числа до одиниці і додаючи потім одиницю в мо-лодший розряд. Це дає можливість одержувати доповнення до одиниці в кожному двійковому розряді простим звертанням (інверсією) цих розрядів. Дійсно, якщо поміняти нулі на одиниці й одиниці на нулі, наприклад, у чис-лі 1011, то одержимо той же результат, що і при відніманні його від одиниць:

$$\begin{array}{r} 1111 \\ -\;1011 \\ \hline 0100 \end{array}$$

З усього сказаного випливає, що $[X]_{дод} = [X]_{зв} + 0{,}00...01$. Таким чином, щоб одержати число в додатковому коді, його перетворюють у зворотній код (шляхом простого звертання розрядів), а потім до молодшого розряду отриманого числа додають одиницю. Наприклад, представимо числа $N_1 = 0{,}1111$ і $N_2 = -0{,}1111$ у прямому, зворотньому і додатковому кодах; $[N_1]_{пр} = [N_1]_{зв} = [N_1]_{дод} = 0{,}1111$; $[N_2]_{пр} = 1{,}1111$; $[N_2]_{зв} = 1{,}0000$; $[N_2]_{дод} = 1{,}0001$. При додаванні додаткових кодів за відповідними правилами в машині одержують додатковий код суми.

Модифіковані коди. Крім зворотнього і додаткового кодів у багатьох ЕОМ застосовують модифіковані зворотні і додаткові коди, що позначаються відповідно $[X]^{м}_{зв}$ і $[X]^{м}_{дод}$. Ці коди відрізняються від звичайних тим, що в них під знак числа виділяється не один, а два розряди. Так, для попереднього прикладу $N_1 = 0{,}1111$ і $N_2 = -0{,}1111$ будемо мати $[N_1]^{м}_{зв} = [N_1]^{м}_{дод} = 00{,}1111$; $[N_2]^{м}_{зв} = 11{,}0000$; $[N_2]^{м}_{дод} = 11{,}0001$. Як уже відзначалося, модифіковані коди зручні для виявлення переповнення розрядної сітки, що може відбутися при додаванні чисел.

Тема 1.6. Алгоритми виконання операцій додавання та віднімання в арифметико-логічному пристрої ЕОМ.

Додавання і віднімання чисел. Як уже відзначалося, в ЕОМ не роблять безпосередньо операцію віднімання. Її замінюють виконанням операції додавання з використанням зворотнього чи додаткового коду.

Спочатку розглянемо додавання чисел у машинах з фіксованою крапкою. Для цієї мети повинна бути виконана така послідовність дій: перетворення прямого коду вихідних чисел у зворотній (чи додатковий); порозрядне додавання кодів (з урахуванням можливих переносів з розряду в розряд); перетворення результату знову в прямий код при його передачі в інші пристрої машини.

При порозрядному додаванні зворотніх (чи додаткових) кодів знакові розряди додають як розряди мантис. Якщо в результаті додавання утвориться одиниця переносу зі знакового розряду, то при використанні зворотніх кодів її додають до молодшого розряду суми (ця операція називається циклічним переносом), а у випадку додаткових кодів цей перенос не враховують.

Нехай числа X і Y, що додаються, є правильними дробами, а їхня сума за абсолютною величиною менша від одиниці, тобто $|X + Y| < 1$. При додаванні зворотніх (чи додаткових) кодів можливі чотири різних випадки, які розглянемо на конкретних прикладах.

***1. X* > 0, У > 0, *X* + У > 0.** Цей випадок не відображає особливостей використання зворотнього чи додаткового коду, тому що для додатніх чисел ці коди збігаються із прямим кодом.

***2. X* > 0, Y < 0, *X* + Y < 0**. Нехай потрібно додати *X* = +3 і Y= -6. Оскільки X= 011_2 і Y = -110_2, то $[X]_{пр} = [X]_{зв} = [X]_{дод}$ = 0.011, а $[Y]_{пр}$ = 1.110, $[Y]_{зв}$ = 1.001 і $[Y]_{дод}$ = 1.010.

При додаванні зворотніх і додаткових кодів відповідно маємо

$[X]_{зв}$ = 0.011 $\qquad\qquad$ $[X]_{дод}$ = 0.011
+ $\qquad\qquad\qquad\qquad$ +
$\underline{[Y]_{зв} = 1.001}$ $\qquad\qquad$ $\underline{[Y]_{дод} = 1.010}$
$[X+Y]_{зв}$= 1.100 $\qquad\qquad$ $[X+Y]_{дод}$ = 1.101

Переведемо отримані результати в прямий код, двійковий та десятковий записи: $[1.100]_{зв} \rightarrow [1.011]_{пр} \rightarrow -011_2 \rightarrow -3_{10}$; $[1.101]_{дод} \rightarrow [1.100]_{зв} \rightarrow [1.011]_{пр} \rightarrow -011_2 \rightarrow -3_{10}$. Результат в обох випадках отримали вірний, тому що 3-6=-3.

3. *X* > 0, Y < 0, *X* + Y > 0. Нехай тепер потрібно додати X = + 6 і Y = -3. Для цього випадку маємо: $[X]_{пр} = [X]_{зв} = [X]_{дод}$ = 0.110; $[Y]_{пр}$ = 1.011; $[Y]_{зв}$ = 1.100; $[У]_{дод}$ = 1.101. При додаванні зворотних і додаткових кодів відповідно одержимо

$[X]_{зв}$ = 0.110 $\qquad\qquad$ $[X]_{дод}$ = 0.110
+ $\qquad\qquad\qquad\qquad$ +
$\underline{[Y]_{зв} = 1.100}$ $\qquad\qquad$ $\underline{[Y]_{дод} = 1.101}$
10.010 $\qquad\qquad$ $[X+Y]_{дод}$ = 10.011=0.011
+
$\underline{\qquad\qquad 1}$
$[X+Y]_{зв}$= 0.011

У даному випадку при використанні додаткових кодів одиницю переносу, що утвориться при додаванні знакових розрядів, відкидають, а при використанні зворотних кодів — циклічно додають до молодшого розряду мантиси. Сума в обох випадках вийшла додатня і дорівнює $[X+Y]_{зв} = [X+Y]_{дод} = [X+Y]_{пр}$ = 0.011 = $+011_2$= $+3_{10}$. Таким чином, отримано правильний результат додавання, тому що 6 - 3 = +3.

4. **X < 0, Y < 0, X + Y < 0**. Нехай потрібно додати X = -3 і

Y = -4. У результаті повинно вийти X + Y = -7_{10} = -111_2. Переконаємося в цьому. Для даного випадку маємо $[X]_{пр}$ =1.011, $[X]_{зв}$ = 1.100, $[X]_{дод}$ = 1.101; $[Y]_{пр}$ = 1.100, $[Y]_{зв}$ = 1.011, $[Y]_{дод}$ =1.100. При додаванні зворотніх і додаткових кодів відповідно отримаємо

$[X]_{зв}$ = 1.100 $\qquad\qquad$ $[X]_{дод}$ = 1.101
+ $\qquad\qquad\qquad\qquad$ +
$\underline{[Y]_{зв} = 1.011}$ $\qquad\qquad$ $\underline{[Y]_{дод} = 1.100}$
10.111 $\qquad$ $[X+Y]_{дод}$ = 11.001=1.001
+
$\underline{\qquad\qquad 1}$

$[X+Y]_{зв}= 1.000$

Переведемо отримані результати в прямий код. $[1.000]_{зв} \rightarrow [1.111]_{пр} \rightarrow -111_2 \rightarrow -7_{10}$; $[1.001]_{дод} \rightarrow [1.000]_{зв} \rightarrow [1.111]_{пр} \rightarrow -111_2 \rightarrow -7_{10}$. В обох випадках отримали правильний результат.

Розглянемо тепер виконання операції додавання в машинах з плаваючою комою. Для виконання даної операції необхідно спочатку вирівняти порядки доданків, для чого менший порядок збільшується до більшого, а його мантиса нормалізується на відповідне число розрядів. Порядком суми є загальний порядок доданків. Потім додають мантиси з використанням зворотного чи додаткового коду (звичайного чи модифікованого). Це додавання виконують за тими самими правилами, що і в машинах з фіксованою крапкою. При цьому можуть зустрічатися три різних випадки, які розгялнемо на конкретних прикладах.

1. Додавання відбувається без переповнення розрядної сітки і порушення нормалізації. Нехай потрібно додати два числа $X=-2_{10}=-10_2$ і $Y=+7_{10} = +111_2$. В нормальній формі ці числа представляють у вигляді $X=-0{,}1 \cdot 2^{10}$ і $Y=+0{,}111 \cdot 2^{11}$. В машині з плаваючою комою їм відповідають наступні записи:

	зн. мантиси	мантиса	зн. порядку	порядок
X	1	100	0	10
Y	0	111	0	11

Порядки у цих чисел різні, тому потрібно збільшити на одиницю порядок числа X, зсунувши при цьому його мантису на один розряд вправо. Після вирівнювання порядків отримаємо

X	1	010	0	11
Y	0	111	0	11

Виконаємо тепер додавання мантис в модифікованому зворотньому коді:

$$
\begin{array}{r}
[X]^{м}_{зв} = 11\ 101 \\
+ \\
\underline{[Y]^{м}_{зв} = 00\ 111} \\
100\ 000 \\
+ \\
\underline{1} \\
[X+Y]^{м}_{зв} = 00\ 101
\end{array}
$$

в результаті отримали додатнє число, його прямий код співпадає зі зворотнім, тому X+Y 0 101 0 11, що відповідає $X+Y=+0{,}101 \cdot 2^{11} =+101_2 = +5_{10}$, тобто результат додавання правильний, тому що –2+7=+5.

2. Додавання відбувається без переповнення розрядної сітки, але результат після переводу в прямий код виявляється ненормалізованим (старший розряд мантиси рівний нулю). Цей випадок називається **порушення нормалізації вправо**. Потрібну нормалізацію результату здійснюють зсувом мантиси вліво із відповідним зменшенням порядку суми.

Виконаємо алгебраїчне додавання чисел X=+5 і Y=-7. Результат повинен бути рівний X+Y = 5-7= -2. У формі з плаваючою комою доданки X і Y запишуться у вигляді

X 0 101 0 11
Y 1 111 0 11

Вирівнювання порядків не потрібно,тому що вони однакові і дорівнюють $3_{10}=11_2$. Виконаємо додавання мантис в модифікованому зворотньому коді:

$$\begin{array}{l} \quad [X]^{м}_{зв} = 00\ 101 \\ + \\ \quad \underline{[Y]^{м}_{зв} = 11\ 000} \\ [X+Y]^{м}_{зв} = 11\ 101 \end{array}$$

Переведемо мантису в прямий код: $[X+Y]^{м}_{пр}$=11.010. У формі з плаваючою комою результат запишеться у вигляді: X+Y 1 010 0 11. Нормалізуємо результат зсувом мантиси вліво на один розряд і відповідним зменшенням порядку: X+Y 1 100 0 10. Таким чином, отримали результат: $X+Y = -0{,}100\cdot 2^{10} = -10_2 = -2_{10}$.

<u>При додаванні мантис відбувається переповнення розрядної сітки.</u> Причиною переповнення є те, що сума $|X + Y| \geq 1$. При використанні модифікованих кодів в знакових розрядах з'являються різні цифри, причому комбінація 01 має місце для додатньої суми,а комбінація 10 – для від'ємної. Отже, знак суми встановлюється по лівому з двох знакових розрядів. Даний випадок переповнення розрядної сітки прийнято називати **порушенням нормалізації вліво**.

При появі переповнення розрядної сітки ЕОМ автоматично виконує зсув результату на один розряд вправо, збільшує порядок на одиницю і заносить в правий знаковий розряд цифру, яка стоїть в лівому знаковому розряді. Розглянемо цей випадок на прикладі додавання чисел X = -2 і Y= -7, які у формі з плаваючою комою запишуться у вигляді

X 1 1000 0 10
Y 1 1110 0 11

Після вирівнювання порядків отримаємо

X 1 0100 0 11
Y 1 1110 0 11

Виконаємо додавання мантис в модифікованому зворотньому коді:

$$\begin{array}{l} \quad [X]^{м}_{зв} = 11\ 1011 \\ + \\ \quad \underline{[Y]^{м}_{зв} = 11\ 0001} \\ \qquad\qquad 110\ 1100 \\ \qquad\qquad\qquad + \\ \qquad\qquad\qquad \underline{1} \\ [X+Y]^{м}_{зв} = 10\ 1101 \end{array}$$

Відбулося порушення нормалізації вліво. Після зсуву на один розряд вправо і збільшення результату порядку суми на одиницю переведемо результат в прямий код. Отримаємо правильний результат: $X+Y = -0{,}1001\cdot 2^{100} = -1001_2 = -9_{10}$.

Розглянуті приклади свідчать про те,що в ЕОМ з плаваючою комою операцію додавання виконати дещо складніше, ніж в ЕОМ з фіксованою крапкою.

Таблиці додавання у вісімковій та шістнадцятковій системах числення.

Дія додавання з числами у вісімковій системі числення виконується так, як вказано в таблиці 5.

Таблиця 5.

+	0	1	2	3	4	5	6	7	10
0	0	1	2	3	4	5	6	7	10
1	1	2	3	4	5	6	7	10	11
2	2	3	4	5	6	7	10	11	12
3	3	4	5	6	7	10	11	12	13
4	4	5	6	7	10	11	12	13	14
5	5	6	7	10	11	12	13	14	15
6	6	7	10	11	12	13	14	15	16
7	7	10	11	12	13	14	15	16	17
10	10	11	12	13	14	15	16	17	20

Арифметична дія додавання з шістнадцятковими числами виконується так, як показано в таблиці 6.

Таблиця 6.

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Тема 1.7. Алгоритми множення та ділення двійкових чисел в АЛП.

Множення чисел.

Числа в ЕОМ множаться в прямому коді за звичайними правилами арифметики двійкових чисел. Знак добутку визначається сумуванням знакових розрядів співмножників, причому одиницю переносу в старший розряд, яка утворюється при додаванні, не враховують. При однакових знаках у співмножників знак добутку отримується "+", а при різних – знак " – " : 0+0=0, 0+1=1, 1+0=1, 1+1=0.

Розглянемо спочатку виконання операції множення в ЕОМ з фіксованою крапкою. Процес множення зводиться до порозрядного множення першого множника на кожний розряд другого множника, потім до зсуву і додаванню отриманих проміжних добутків. При цьому проміжні добутки або дорівнюють множнику, на який множать, якщо в даному розряді множника стоїть одиниця, або дорівнюють нулю. Тому множення фактично складається з послідовних зсувів і додавання проміжних добутків, отриманих в результаті зсувів.

Процес множення може здійснюватися з зсувом вліво чи вправо. В першому випадку множення починають з молодшої цифри множника, а у другому – зі старшої. Якщо даний розряд множника містить нуль, то просто здійснюють зсув проміжного добутку, тобто пропускають такт додавання з проміжним добутком, рівним нулю.

Якщо співставити процес множення на машині з послідовністю дій, які виконуємо ручним способом, то можна побачити багато спільного. Розглянемо спочатку множення ручним способом на наступному прикладі:

```
  0,1001
     x
    0,0101
    1001
     0000
   +
    1001
   0000
0,00101101
```

Тепер виконаємо це множення на машині, використовуючи зсув множника вправо. При цьому в суматорі Σ, який здійснює сумування проміжних добутків, передбачено подвійна кількість розрядів порівнно зі співмножниками:

```
Σ          00000000                 Множник
Додати     10010000                    1
(множник)           ________
Σ                   10010000
Зсунути    01001000
                    (не додавати)      0
```

```
Зсунути 00100100
Додати  10010000                 1
                  ________
Σ       10110100
Зсунути 01011010
                  (не додавати)  0
Зсунути 00101101
        ________
Σ       00101101
```

Таким чином, отримали той же результат, що й при ручному рахунку. Аналогічно реалізуєтья множення зсувом множника вліво.

Після виконання операції множення необхідно здійснити округлення результату до потрібної кількості розрядів. Якщо в старшому, який відкидаємо, розряді нуль, то молодший з тих, що залишаються, розрядів не змінюється, якщо одиниця, то вона додається до молодшого з тих, що залишилися. В останньому прикладі при заокругленні добутку до чотирьох розрядів (як і у співмножників) необхідно до четвертого розряду після коми додати "1" з п'ятого розряду, тобто отримаємо

```
  0,0010
+
______1
  0,0011
```

В ЕОМ з плаваючою комою процес множення протікає аналогічно, з тією лише різницею, що при цьому додається операція визначення порядку добутку алгебраїчним дродаванням порядків співмножників і проводиться, якщо необхідно, нормалізація результату з відповідною зміною порядку добутку. Розглянемо приклад множення чисел X=0,00101101 і Y= –10000,1. В машині з плаваючою комою ці числа запишуться у вигляді

X	0	101101	1	010
Y	1	100001	0	101

Знак добутку дорівнює: 0+1=1. Порядок добутку дорівнює алгебраїчній сумі порядків співмножників, тобто (-010)+101=+011. Перемножимо мантиси співмножників:

```
    0,101101
     x
____0,100001
      101101
+
      101101    5 зсувів
   0,010111001101
```

Після округлення до шести розрядів після коми результат добутку запишеться у вигляді X · Y 1 010111 0 011. Виконаємо нормалізацію вліво на один розряд, а потім зменшимо порядок добутку на одиницю, в результаті отримаємо X · Y 1 101110 0 010.

Ділення чисел.

Якщо множення виконується в ЕОМ з допомогою чисельних зсу-вів і додавань, то ділення, яке є операцією оберненою до множення, ви-конується з допомогою чисельних зсувів і віднімань. Знак частки, як і при множенні, визначається шляхом сумування знакових розрядів діле-ного і дільника без врахування можливого переносу в старший розряд.

Так я к і при ручному рахунку, розряди частки при діленні чисел визначаються (починаючи зі старшого) послідовним відніманням дільника від остачі, отриманої від попереднього віднімання.

При визначенні першої цифри частки за остачу приймають все ді-лене. Після кожного віднімання остача зсувається вліво по відношенню до дільника. Якщо остачу отримали додатню (чи рівну нулю), то у відповідний розряд частки записується одиниця, а якщо від’ємну то нуль.

Щоб отримати наступну після нуля цифру частки, необхідно від останньої додатньої остачі відняти дільник, додатково зсунутий на один розряд вправо. Зазвичай замість зсуву дільника вправо здійснюють зсув остачі вліво, що, по суті, нічого не змінює.

Оскільки в ЕОМ операція віднімання в безпосередньому вигляді не виконується, послідовне віднімання замінюють додаванням остач з оберненим чи додатковим кодом дільника. При цьому остачі отримують у відповідному коді.

Спочатку розглянемо виконання операції ділення у машинах з фіксованою крапкою. Для таких машин ділення можливо за умови,що ділене за абсолютною величиною менше від дільника. Інакше виникне переповнення розрядної сітки.

Нехай дані два числа: X=0,1001 і Y=0,1101, які потрібно поділити. Виконаємо спочатку ділення ручним способом. Після кожного віднімання дільник зсуваємо вправо по відношенню до діленого. Якщо остачу після віднімання отримуємо додатню, то в даний розряд частки записується одиниця, а якщо від’ємну, то нуль. На практиці зазвичай від’ємну остачу не записують, просто дільник додатково зсувається на один розряд вправо і віднімається від останньої додатньої остачі.

```
 _10010     |1101
   1101
 _10100
   1101
   1110
    1101
       1 і т.д.
```

Операція ділення реалізується в машинах за двома алгоритмами – з відновленням остачі і без її відновлення. В першому випадку при отриманні від'ємної остачі в даному розряді частки записується нуль і відновлюється попередня додатня остача додаванням до отриманої від'ємної остачі дільника в прямому коді. Відновлена остача після цього зсувається додатково на один розряд вліво (аналогічно додатковому зсуву дільника при обчисленні вручну).

Весь процес ділення виконується в три такти, які послідовно повторюються до закінчення операції: перший такт – зсув остачі вліво; другий такт – віднімання від зсунутої остачі дільника (додавання в додатковому чи зворотньому коді); третій такт – запис одиниці у відповідний розряд частки, якщо залишок після віднімання отримався додатній, і нуля, якщо залишок отримали від'ємний (в останньому випадку в цьому ж такті виконують відновлення попередньої остачі).

Алгоритм ділення без відновлення остачі отримав ширше застосування. Він полягає в тому, що при отриманні від'ємної остачі у відповідний розряд частки записується нуль, а потім ця остача зсувається вліво на один розряд і до неї додається дільник в прямому коді. В результаті такого додавання отримуємо наступну остачу, у якої аналізується знак і в залежності від цього виконуються наступні дії.

В ЕОМ з плаваючою комою при діленні окрім розглянутих дій визначається порядок частки відніманням від порядку діленого порядку дільника, а також виконується при необхідності нормалізація результату вправо, якщо отримаємо частку більшу від одиниці (в цих машинах не потрібно, щоб ділене було менше дільника). В подальшому операція ділення протікає аналогічно.

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Опишіть перетворення цілих чисел в системи з іншою основою методом ділення.
2. Опишіть перетворення цілих чисел в системи з іншою основою табличним методом.
3. Опишіть перетворення правильних дробів в системи з іншою основою.
4. Опишіть форму подання з фіксованою комою чисел в комп'ютерах.
5. Опишіть форму подання з плаваючою комою чисел в комп'ютерах.
6. Опишіть форми кодування чисел в комп'ютерах.
7. Опишіть операції з фіксованою комою.
8. Опишіть операції з плаваючою комою.
9. Опишіть множення двійкових чисел в АЛП.
10. Опишіть ділення двійкових чисел в АЛП.

Розділ 2
Логічні основи ЕОМ

В розділі розглянуто основні визначення алгебри логіки, операції алгебри логіки, основні закони алгебри логіки, алгебри перемикальних функцій, досконалі нормальні форми, метод Квайна для мінімізації логічних функцій, мінімізація логічних функцій за допомогою карт Карно, методи синтезу цифрових автоматів з пам'яттю

Після вивчення матеріалу розділу Ви будете:

Знати	▪ *основні визначення алгебри логіки* ▪ *операції алгебри логіки* ▪ *основні закони алгебри логіки* ▪ *алгебри перемикальних функцій* ▪ *досконалі нормальні форми* ▪ *метод квайна для мінімізації логічних функцій* ▪ *методи синтезу цифрових автоматів з пам'яттю* ▪ *мінімізація логічних функцій за допомогою карт карно*
Вміти	– *представляти логічні функції* – *виконувати Операції алгебри логіки* – *виконувати логічне множення* – *виконувати логічне додавання* – *застосовувати Основні закони алгебри логіки* – *утворювати Досконалі нормальні форми* – *спрощувати аналітичні вирази логічної функції*

Ключові поняття та терміни	
• *числення висловів* • *кон'юнкція* • *диз'юнкція* • *еквівалентність* • *імплікація* • *закон інверсії* • *досконалі нормальні форми*	• *диз'юнктивна досконала нормальна форма* • *кон'юнктивна досконала нормальна форма* • *метод Квайна* • *імпліканта* • *метод Карно-Вейча*

Тема 2.1. Операції алгебри логіки. Основні закони алгебри логіки.

<u>Поняття про логічну функцію.</u>

В електронних обчислювальних машинах інформація піддається не лише арифметичній, але й логічній обробці. При цьому машина виконує певні перетворення над двійковими числами, в результаті яких утворюються двійкові числа, які є результатом виконання відповідної логічної операції.

В основі роботи логічних схем і пристроїв ЕОМ лежить спеціальний математичний аппарат, який називається математичною логікою, в якій вивчаються питання застосування математичних методів для розв'язку різних логічних задач.

В ЕОМ використовується в основному початковий розділ математичної логіки – алгебра логіки, яка часто називається ***числення висловів***. Під численням розуміють будь яке твердження, про яке можемо сказати, що воно істинне чи хибне. У логіці висловів цікавляться не змістом висловлювань, а лише їхньою істинністю чи хибністю; ніякі інші ознаки висловів в алгебрі логіки не розглядаються. Одне й те саме висловлювання не може бути одночасно істинним і хибним чи не істинним і не хибним.

Граматичними засобами в розмовній мові з кількох простих висловлювань можна скласти складне висловлювання. Наприклад, за допомогою сполучників І та АБО і заперечної частки НЕ можна з простих висловів: хибного – "Київ стоїть на березі Пруту" та істинного – "Чернівці стоять на березі Пруту" скласти наступні складні висловлювання: "Київ не стоїть на березі Пруту", "Київ не стоїть на березі Пруту і Чернівці стоять на березі Пруту", "Київ стоїть на березі Пруту чи Чернівці стоять на березі Пруту". Усі три вислови істинні.

Якщо вислів істинний то кажуть, що його значення дорівнює одиниці, якщо вислів хибний, то його значення дорівнює нулю. Таким чином, значення висловів можна розглядати як змінну величину, яка приймає лише два дискретних значення – 0 або 1. Це приводить до повної відповідності між логічними висловлюваннями в математичній логіці і двійковими цифрами у двійковій системі числення, що дозволяє описувати роботу логічних схем і проводити їх аналіз і синтез за допомгою математичного апарату алгебри логіки.

Будь який пристрій ЕОМ, який виконує арифметичні дії над двійковими числами, можна розглядати як функціональний перетворювач, вхідними змінними (аргументами) якого є вихідні двійкові числа, а вихідною функцією від них – нове двійкове число, яке утворилось в результаті виконання даної операції. При цьому як вхідні змінні, так і вихідні функції можуть приймати лише одне з двох можливих значень – 0 і 1.

У кожному конкретному випадку кількість вхідних змінних може бути різна. В найпростішому випадку це може бути одна змінна X, яка приймає значення або 0, або 1. У загальному випадку таких змінних може бути n: X_1, X_2,…,X_n, причому як і раніше, кожна з цих змінних приймає лише одне з двох можливих значень – 0 або 1.

Функція f (X_1, X_2,…,X_n), яка визначається набором вхідних двійкових змінних X_1, X_2,…,X_n і приймає в якості своїх можливих значень 0 або 1, називається ***логічною функцією*** або функцією алгебри логіки (**ФАЛ**).

Табличне представлення логічних функцій.

Як уже зазначалось, кількість аргументів (вхідних змінних) логічної функції може бути різним. Оскільки кожна змінна при цьому може бути рівною чи 0 чи 1, то в результаті цього можуть бути утворені їх найрізноманітніші сполучення чи, як кажуть, набори вхідних змінних.

В алгебрі логіки строго доводиться, що для n змінних існує 2^n різних наборів. Переконаємось в цьому на кількох прикладах. Так, для однієї змінної X існують тільки два набори: чи <0>, чи <1>, тобто $2^1=2$. Для двох змінних можуть бути утворені вже чотири різних набори, тому що $2^2=4$. Дійсно, <0,0>; <0,1>; <1,0>; <1,1>. Для трьох змінних буде існувати вже вісім різних наборів (див. таблицю), оскільки $2^3=8$. Табличне представлення логічних функцій дає значення цих функцій для різних наборів вхідних змінних. За допомогою аналогічних таблиць зручно описувати функціонування різних логічних елементів, вузлів і пристроїв ЕОМ.

Таблиця 7

X_1	X_2	X_3
0	0	0
0	0	1
0	1	0
1	0	0
0	1	1
1	0	1
1	1	0
1	1	1

Приймемо без доведення, що загальна кількість різних логічних функцій від n аргументів дорівнює 2^{2n}. Наприклад, для n=1,2,3 і т.д. їх буде відповідно 4,16,256 і т.д.

3. Операції алгебри логіки

Розглянемо для двох змінних різні логічні функції (див. таблицю) і відповідні їм логічні операції, які можуть бути реалізовані в ЕОМ. Розглянемо ці функції не в порядку їх нумерації, а в послідовності, яка дозволить виявити їх загальні і найхарактерніші властивості.

Таблиця 8. Значення логічних функцій двох змінних

X_1	X_2	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7	ψ_8	ψ_9	ψ_{10}	ψ_{11}	ψ_{12}	ψ_{13}	ψ_{14}	ψ_{15}	ψ_{16}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

1. $\psi_9(X_1, X_2)$ – кон'юнкція (логічне множення чи операція І) змінних X_1, X_2, яка має значення одиниці лише в тому випадку, коли аргументи X_1 і X_2 одночасно дорівнюють

1, і нуль – у всіх інших випадках, тобто коли хоча б один аргумент дорівнює нулю. Іншими словами, функція кон'юнкції дорівнює min (X_1, X_2). Позначається кон'юнкція знаком $\wedge$, який читається як І, тобто запис $X_1 \wedge X_2$ слід читати "X_1 і X_2". Значення кон'юнкції за значеннями аргументів знаходиться за правилами логічного множення (див. таблицю). У цій таблиці $\overline{X}$ означає заперечення змінної X, тобто $\overline{X}$ =0, якщо X=1, і $\overline{X}$ =1, якщо X=0. В загальному випадку функція кон'юнкції (або логічного множення) може бути визначена для будь-якої кількості аргументів, тобто $X_1 \wedge X_2 \wedge X_3 \wedge \ldots$ Знак $\wedge$ може опускатися чи замінюватися крапкою, тобто може бути записано: $X_1X_2X_3\ldots$

Правила логічного множення

$0\wedge0=0$	$X\wedge0=0$
$0\wedge1=0$	$X\wedge1=X$
$1\wedge0=0$	$X\wedge X=X$
$1\wedge1=1$	$X\wedge\overline{X}=0$

2. $\psi_8(X_1, X_2)$ – заперечення кон'юнкції (чи функція Шеффера), тобто $\overline{X_1 \wedge X_2}$. Дана функція має значення нуль тільки в тому випадку, коли аргументи X_1 та X_2 одночасно дорівнюють одиниці, і одиницю – у всіх інших випадках, тобто коли хоча б один з аргументів дорівнює нулю.

3. $\psi_{15}(X_1, X_2)$ – диз'юнкція (логічне додавання чи операція АБО) змінних X_1, X_2, яка має значення нуль тільки в тому випадку, якщо аргументи X_1 і X_2 одночасно дорівнюють нулю, у всіх інших випадках вона дорівнює одиниці, тобто коли хоча б один аргумент дорівнює одиниці. Іншими словами, функці диз'юнкції дорівнює max(X_1,X_2). Позначається диз'юнкція значком $\vee$, який читається як АБО, тобто запис $X_1\vee X_2$ слід читати "X_1 або X_2". Значення диз'юнкції за значеннями аргументів знаходиться за правилами логічного додавання (таблиця).

Правила логічного додавання

$0\vee0=0$	$X\vee0=X$
$0\vee1=1$	$X\vee1=1$
$1\vee0=1$	$X\vee X=X$
$1\vee1=1$	$X\vee\overline{X}=1$

В загальному випадку функція диз'юнкції (або логічного додавання) може бути визначена для будь-якої кількості аргументів, тобто $X_1\vee X_2\vee X_3\vee\ldots$

4. $\psi_2(X_1, X_2)$ – заперечення диз'юнкції (чи операція Пірса), тобто $\overline{X_1 \vee X_2}$. Дана функція має значення одиниці тільки в тому випадку, коли аргументи X_1 і X_2

одночасно дорівнюють нулю, у всіх інших випадках вона дорівнює 0 (тобто коли хоча б один аргумент =1).

5. $\psi_{10}(X_1, X_2)$ – еквівалентність (чи рівнозначність) змінних X_1, X_2. Дана функція має значення 1 у тому випадку, коли співпадають значення аргументів X_1 і X_2, у інших випадках вона =0. Позначається еквівалентність знаком ∞, який читається як "рівнозначно", тобто запис $X_1 \infty X_2$ слід читати "X_1 рівнозначно X_2".

6. $\psi_7(X_1, X_2)$ – заперечення еквівалентності (чи нерівнозначність) змінних X_1, X_2. Запис $\overline{X_1 \infty X_2}$ читається як "X_1 нерівнозначно X_2". Можна переконатися, що значення функції нерівнозначності отримуються порозрядним додаванням двійкових змінних X_1 і X_2 по модулю 2, тобто без врахування переносів у старший розряд.

7. $\psi_{12}(X_1, X_2)$ – імплікація X_1 в X_2, яка має значення 0 тільки в тому випадку, коли змінна $X_1 = 1$, а $X_2 =0$, у інших випадках функція імплікації X_1 в $X_2 = 1$. Дана функція позначається $X_1 \rightarrow X_2$ і читається як "якщо X_1, то X_2".

8. $\psi_{14}(X_1, X_2)$ – імплікація X_2 в X_1, яка має значення 0 тільки в тому випадку, якщо змінна $X_2=1$, а $X_1=0$, у інших випадках функція імплікації X_2 в $X_1 =1$. Дана функція позначається $X_2 \rightarrow X_1$ і читається як "якщо X_2, то X_1".

9. $\psi_5(X_1, X_2)$ – заперечення імплікації X_1 в X_2, тобто $\overline{X_1 \rightarrow X_2}$. Дану функцію можна розглядати як функцію заборони з боку вхідної змінної X_2. Це означає, що вихідна функція має значення 0, якщо вхідна змінна $X_2 = 1$, у інших випадках вона повторює змінну X_1.

10. $\psi_3(X_1, X_2)$ – заперечення імплікації X_2 в X_1, тобто $\overline{X_2 \rightarrow X_1}$. Дану функцію можна розглядати як функцію заборони з боку вхідної змінної X_1. Це означає, що вихідна функція має значення 0, якщо вхідна змінна $X_1 = 1$, у інших випадках вона повторює змінну X_2.

Шість функцій, що залишилися ψ_1, ψ_4, ψ_6, ψ_{11}, ψ_{13} і ψ_{16} є або константами, або залежними тільки від однієї змінної X_1 чи X_2. Так, функції ψ_1 і ψ_{16} є відповідно тотожньо рівними нулю і одиниці, тобто $\psi_1(X_1, X_2)\equiv 0$ і $\psi_{16}(X_1, X_2)\equiv 1$. Функції ψ_{13} і ψ_{11} повторюють відповідно змінні X_1 і X_2, тобто $\psi_{13}(X_1, X_2)=X_1$ і $\psi_{11}(X_1, X_2)=X_2$. Функції ψ_4 і ψ_6 є запереченнями відповідно змінних X_1 і X_2, тобто $\psi_4(X_1, X_2) = \overline{X}_1$ і $\psi_6(X_1, X_2)= \overline{X}_2$.

Основні закони алгебри логіки.

В алгебрі логіки є чотири основні закони: переставний, сполучний, розподільний і закон інверсії. Наведемо співвідношення, які відображають ці закони.

1. Переставний закон: для логічного додавання $X_1 \vee X_2 = X_2 \vee X_1$; для логічного множення $X_1 \wedge X_2 = X_2 \wedge X_1$

2. Сполучний закон: для логічного додавання $(X_1 \vee X_2) \vee X_3 = X_1 \vee (X_2 \vee X_3)$; для логічного множення $(X_1 \wedge X_2) \wedge X_3 = X_1 \wedge (X_2 \wedge X_3)$.

3. Розподільний закон: для логічного додавання $(X_1 \vee X_2) \wedge X_3 = (X_1 \wedge X_3) \vee (X_2 \wedge X_3)$; для логічного множення $(X_1 \wedge X_2) \vee X_3 = (X_1 \vee X_3) \wedge (X_2 \vee X_3)$.

4. Закон інверсії: для логічного додавання $\overline{X_1 \vee X_2} = \overline{X_1} \wedge \overline{X_2}$; для логічного множення $\overline{X_1 \wedge X_2} = \overline{X_1} \vee \overline{X_2}$.

Закони переставний і сполучний, а також розподільний для логічного додавання мають повну аналогію з відповідними законами звичайної алгебри і тому не потребують спеціального доведення. Така аналогія відсутня для розподільного закону логічного множення і закону інверсії.

Справедливість законів алгебри логіки можна довести табличним способом, знаходячи значення лівої й правої частин логічного виразу, що доводиться, для усіх можливих наборів вхідних змінних. Якщо ці вирази співпадуть, то закон буде доведено.

Доведемо, наприклад, справедливість закону інверсії для логічного додавання і множення. З цією метою складемо таблицю для чотирьох можливих комбінацій (наборів) вхідних змінних X_1 і X_2, а потім у стовбчиках цієї таблиці обчислимо за відповідними правилами алгебри логіки значення лівих і правих частин виразів, що доводяться. Отримані результати занесемо у таблицю:

Таблиця 9.

X_1	X_2	$X_1 \vee X_2$	* $\overline{X_1 \vee X_2}$	$X_1 \wedge X_2$	** $\overline{X_1 \wedge X_2}$	$\overline{X_1}$	$\overline{X_2}$	* $\overline{X_1} \wedge \overline{X_2}$	** $\overline{X_1} \vee \overline{X_2}$
0	0	0	1	0	1	1	1	1	1
0	1	1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	1	0	1
1	1	1	0	1	0	0	0	0	0

Порівнюючи в ній стовбці, відмічені відповідно однією чи двома зірочками, переконуємося в тому, що вони співпадають. Отже, доведена справедливість законів інверсії для логічного додавання і множення.

Аналогічним чином можна довести справедливість інших співвідношень в алгебрі логіки, що існують між різними логічними функціями двох змінних. Так, справедливі наступні співвідношення для функцій імплікації: $X_1 \rightarrow X_2 = \overline{X_1} \vee X_2$; $X_2 \rightarrow X_1 = \overline{X_2}$;

заперечення імплікації: $\overline{X_1 \rightarrow X_2} = X_1 \wedge \overline{X_2}$; $\overline{X_1 \rightarrow X_2} = \overline{X_1} \wedge X_2$;

рівнозначності: $X_1 \infty X_2 = (\overline{X_1} \wedge \overline{X_2}) \vee (X_1 \wedge X_2)$;

нерівнозначності: $\overline{X_1 \infty X_2} = (\overline{X_1} \vee \overline{X_1}) \wedge (X_1 \vee X_2)$.

З розподільного закону для логічного додавання і множення витікають так звані формули склеювання: $(X_1 \wedge \overline{X}_2) \vee (X_1 \wedge X_2) = X_1$; $(X_1 \vee \overline{X}_2) \wedge (X_1 \vee X_2) = X_1$

і формули поглинання: $X_1 \vee (X_1 \wedge X_2) = X_1$; $X_1 \wedge (X_1 \vee X_2) = X_1$;

$X_1 \vee (\overline{X}_1 \wedge X_2) = X_1 \vee X_2$; $X_1 \wedge (\overline{X}_1 \vee X_2) = X_1 \wedge X_2$.

Слід зазначити, що розглянуті закони справедливі не лише для двох, але і для будь якої кількості аргументів. Наприклад, закон інверсії логічного додавання і множення в загальному випадку має вигляд

$\overline{X_1 \vee X_2 \vee X_3 \ \ldots} = \overline{X}_1 \wedge \overline{X}_2 \wedge \overline{X}_3 \wedge \ldots$; $\overline{X_1 \wedge X_2 \wedge X_3 \ \ldots} = \overline{X}_1 \vee \overline{X}_2 \vee \overline{X}_3 \vee \ldots$

Відповідно для розподільного закону можливі наступні співвідношення:

$X_1 \wedge (X_2 \vee X_3 \vee \ldots) = (X_1 \wedge X_2) \vee (X_1 \wedge X_3) \vee \ldots$; $X_1 \vee (X_2 \wedge X_3 \wedge \ldots) = (X_1 \vee X_2) \wedge (X_1 \vee X_3) \wedge \ldots$

Використовуючи наведені вище закони і співвідношення алгебри логіки, можна істотньо спростити складні логічні вирази і отримати простіші формули. Розглянемо приклад перетворення складної логічної функції трьох аргументів:

$f(X_1, X_2, X_3) = X_1 \wedge \overline{X}_3 \vee \overline{X_1 \wedge \overline{X}_2 \vee \overline{X}_1 \wedge X_3} \vee X_2 \wedge X_3 \wedge (X_1 \vee \overline{X}_1 \wedge X_2 \wedge \overline{X}_3)$.

Користуючись законом інверсії і розподільним законом, перетворимо спочатку

$\overline{X_1 \wedge \overline{X}_2 \vee \overline{X}_1 \wedge X_3} = \overline{X_1 \wedge \overline{X}_2} \wedge \overline{\overline{X}_1 \wedge X_3} = (\overline{X}_1 \vee X_2) \wedge (X_1 \vee \overline{X}_3) = \overline{X}_1 \wedge X_2 \vee \overline{X}_1 \wedge \overline{X}_3 \vee X_1 \wedge X_2 \vee X_2 \wedge \overline{X}_3 = \overline{X}_1 \wedge \overline{X}_3 \vee X_1 \wedge X_2 \vee X_2 \wedge \overline{X}_3$ (тут враховувалось також, що $\overline{\overline{X}} = X$ і $\overline{X} \wedge X = 0$). Тепер перетворимо наступну складову заданої функції, розкриваючи в ній дужки:

$X_2 \wedge X_3 \wedge (X_1 \vee \overline{X}_1 \wedge X_2 \wedge \overline{X}_3) = X_1 \wedge X_2 \wedge X_3 \vee X_2 \wedge X_3 \wedge X_2 \wedge \overline{X}_3 = X_1 \wedge X_2 \wedge X_3$ (враховуючи, що другий член останнього виразу містить кон'юнкцію $X_3 \wedge \overline{X}_3$, яка перетворює його в нуль).

Підставляючи отримані вирази перетворених членів в вихідну формулу заданої функції, маємо

$f(X_1, X_2, X_3) = X_1 \wedge \overline{X}_3 \vee \overline{X}_1 \wedge \overline{X}_3 \vee X_1 \wedge X_2 \vee X_2 \wedge \overline{X}_3 \vee X_1 \wedge X_2 \wedge X_3$.

Застосовуючи до отриманого виразу формули поглинання і склеювання, отримаємо

$f(X_1, X_2, X_3) = \overline{X}_3 \wedge (X_1 \vee \overline{X}_1) \vee X_1 \wedge X_2 \wedge (1 \vee X_3) \vee X_2 \wedge \overline{X}_3 = X_3 \vee X_1 \wedge X_2 \vee X_2 \wedge X_3 = \overline{X}_3 \wedge (1 \vee X_2) \vee X_1 \wedge X_2 = \overline{X}_3 \vee X_1 \wedge X_2$. Таким чином, в результаті перетворення вихідна логічна функція значно спростилась і в результаті має вигляд $f(X_1, X_2, X_3) = X_1 \wedge X_2 \vee \overline{X}_3$.

Базиси перемикальних функцій.

При використанні аналітичних форм представлення логічних функцій широко використовують принцип суперпозиції, який полягає в заміні одних аргументів даної функції іншими. Наприклад, якщо аргументи функції Z = Z (X,Y) є, в свою чергу, функціями інших аргументів X = X (a,b) і Y = Y (c,d), то можна утворити функцію вигляду Z = Z (a,b,c,d).

Спрощення логічних виразів можна досягнути виразивши складні логічні функції через інші функції. При цьому система логічних функцій f_0, f_1, f_2,..., f_k називається ***функціонально повною***, якщо будь яку функцію алгебри логіки можна представити в аналітичній формі через ці функції.

Прикладом функціонально повної системи логічних функцій можуть бути функції інверсії, диз'юнкції і кон'юнкції. Оскільки закон інверсії дозволяє перетворювати диз'юнкцію в кон'юнкцію і навпаки, то функціонально повними системами будуть диз'юнкція і інверсія, а також кон'юнкція і інверсія.

Функціонально повними є також функції заперечення кон'юнкції $\overline{X \wedge Y}$, або заперечення диз'юнкції $\overline{X \vee Y}$ тому що через будь яку з цих функцій можна представити будь яку логічну функцію від n змінних при будь якому значенні n. По цій причині ці дві функції називаються також ***універсальними логічними функціями***.

Тема 2.2. Нормальні форми. Досконалі нормальні форми.

Розглядуване нами раніше табличне представлення логічних функцій суттєво ускладнюється зі збільшенням числа аргументів, наприклад, вже для трьох аргументів буде $(2^2)^3$ = 256 логічних функцій. Для цієї мети зручнішим є аналітичне представлення логічних функцій у вигляді відповідних формул. Найраціональнішим є представлення логічних функцій в нормальних формах. В основі цих форм лежать поняття елементарних кон'юнкцій і елементарних диз'юнкцій.

Елементарна кон'юнкція (мінтерм) утворюється кон'юнкцією кінцевої множини логічних змінних та їх заперечень, наприклад, P(X,Y,Z) = $\overline{X} \wedge Y \wedge \overline{Z}$. Елементарна диз'юнкція (макстерм) утворюється диз'юнкцією кінцевої множини логічних змінних та їх заперечень, наприклад, Q(X,Y,Z) = $X \vee \overline{Y} \vee \overline{Z}$. Елементарна кон'юнкція (мінтерм) приймає значення одиниці при одному із усіх можливих наборів вхідних аргументів, а елементарна диз'юнкція (макстерм) навпаки, приймає нульове значення при одному з можливих наборів аргументів і значення одиниці – при усіх інших.

Отже, мінтерм алгебраїчно представляє кон'юнкцію аргументів та їх заперечень, а макстерм – диз'юнкцію аргументів та їх заперечень. Число мінтермів чи макстермів співпадає з числом наборів різних аргументів (див. таблицю), тобто для n аргументів їх буде відповідно N = = 2^n (N = 2^2 = 4, знаки кон'юнкції в таблиці пропущені).

Таблиця 10. Набори елементарних кон'юнкцій і диз'юнкцій для двох аргументів

Значення аргумен-тів		**Значення елементарних кон'юнкцій**				**Значення елементарних диз'юнкцій**			
X	Y	P= $\overline{X}\,\overline{Y}$	P= $\overline{X}Y$	P= $X\overline{Y}$	P=XY	P=X $\vee$ Y	P=X $\vee \overline{Y}$	P= $\overline{X} \vee$ Y	P= $\overline{X} \vee \overline{Y}$
0	0	1	0	0	0	0	1	1	1
0	1	0	1	0	0	1	0	1	1
1	0	0	0	1	0	1	1	0	1
1	1	0	0	0	1	1	1	1	0

Якщо логічна функція виражена через інверсію, кон'юнкцію і диз'юнкцію, то така форма її представлення називається ***нормальною***. Розрізняють ***диз'юнктивну*** і ***кон'юнктивну*** нормальні форми. **Диз'юнктивна нормальна форма** (ДНФ) є логічною сумою елементарних кон'юнкцій (мінтермів), наприклад, P(X,Y) = $\overline{X}Y \vee X\overline{Y}$ **Кон'юнктивна нормальна форма** (КНФ) є логічним добутком елементарних диз'юнкцій (макстермів), наприклад, Q(X,Y) = (X $\vee$ Y) $\wedge$ ($\overline{X} \vee$ Y).

Кількість аргументів, які утворюють елементарну кон'юнкцію чи диз'юнкцію, є її **рангом**. Наприклад, функція P (X_1, X_2, X_3, X_4) = $X_1X_2X_3X_4$ є елементарною кон'юнкцією четвертого рангу, а функція Q(X_1,X_2,X_3) = $X_1 \vee X_2 \vee X_3$ – елементарною диз'юнкцією третього рангу.

Якщо до складу логічної формули входять набори елементарних кон'юнкцій однакового рангу, пов'язані диз'юнкцією, то така форма представлення логічної функції отримала назву **диз'юнктивна досконала нормальна форма** (ДДНФ). Логічну функцію, яка містить елементарні диз'юнкції одного рангу, пов'язані кон'юнкцією, прийнято називати **кон'юнктивною досконалою нормальною формою** (КДНФ).

Слід зазначити, що одна й та сама логічна функція може бути представлена різними ДНФ і КНФ. З усієї сукупності цих нормальних форм виділяють одну ДНФ і одну КНФ, а саме ті форми, які є інверсними одна до одної, тобто якщо одна з них дорівнює 1, то інша при цьому рівна 0 і навпаки. Саме ці форми називаються досконалими, тобто ДДНФ і КДНФ.

В алгебрі логіки суворо доводиться, що будь яка логічна функція, окрім f $\equiv$ 0, може бути представлена в ДДНФ, а будь яка функція, окрім f $\equiv$ 1, може бути представлена в КДНФ.

Прикладом ДДНФ логічної функції Р можна назвати наступну функцію: $P(X,Y,Z) = XYZ \vee \overline{X}Y\overline{Z} \vee \overline{X}YZ$. Дана функція відповідає вимогам, які пред'являються до ДДНФ, тому що у ній нема двох одинакових кон'юнкцій; жодна кон'юнкція не містить двох однакових змінних; жодна кон'юнкція не містить двійкову змінну разом з її запереченням; усі кон'юнкції одного рангу.

Якщо логічна функція містить кон'юнкції різних рангів, то слід використати властивість $X \vee \overline{X} = 1$ для кон'юнкції молодшого рангу і підвищити її ранг для утворення ДДНФ. Наприклад, $P(X,Y,Z) = \overline{X}Y \vee XYZ = \overline{X}Y(Z \vee \overline{Z}) \vee XYZ = \overline{X}YZ \vee \overline{X}Y\overline{Z} \vee XYZ$.

При аналізі та синтезі логічних схем зазвичай використовують опис їхньої роботи, заданий у вигляді відповідної таблиці, з якої легко можуть бути отримані логічні формули в ДДНФ чи КДНФ. Правило утворення ДДНФ функції від n аргументів полягає в наступному:

1. По кожному набору двійкових змінних, при якому логічна функція приймає значення одиниці, скласти елементарні кон'юнкції (мінтерми).
2. В елементарні кон'юнкції записати неінвертованими змінні, задані одиницею в табличному представленні функції, а інвертованими – ті змінні, які у цій таблиці задані нулем.
3. З'єднати елементарні кон'юнкції знаком диз'юнкції.

Нехай деяка функція Р від трьох аргументів X,Y,Z задана в наступній таблиці. Ця функція приймає значення одиниці для чотирьох наборів аргументів, тому функція Р в ДДНФ буде складатися із логічної суми чотирьох елементарних кон'юнкцій (мінтермів) третього рангу, тобто $P(X,Y,Z) = \overline{X}Y\overline{Z} \vee \overline{X}YZ \vee X\overline{Y}Z \vee XYZ$.

Таблиця 11. Значення функції Р від трьох змінних

Значення аргументу			Зна-чення функції Р	ДДНФ	КДНФ	Значення аргументу			Зна-чення функції Р	ДДНФ	КДНФ
X	Y	Z		мін-терм	макс-терм	X	Y	Z		мін-терм	макстерм
0	0	0	0	-	$X \vee Y \vee Z$	1	0	0	0	-	$\overline{X} \vee Y \vee Z$
0	0	1	0	-	$X \vee Y \vee \overline{Z}$	1	0	1	1	$X\overline{Y}Z$	-
0	1	0	1	$\overline{X}Y\overline{Z}$	-	1	1	0	0	-	$\overline{X} \vee \overline{Y} \vee Z$
0	1	1	1	$\overline{X}YZ$	-	1	1	1	1	XYZ	-

Кон'юнктивна досконала нормальна форма відповідає наступним вимогам: у ній нема двох однакових елементарних диз'юнкцій; жодна диз'юнкція в ній не містить двох одинакових змінних; жодна диз'юнкція не містить змінну разом з її запереченням; усі диз'юнкції одного рангу. Прикладом КДНФ деякої функції P(X,Y,Z) може служити, зокрема, функція вигляду $P(X,Y,Z) = (X \vee Y \vee Z) \wedge (\overline{X} \vee Y \vee \overline{Z})$.

За табличним представленням логічної функції можна також отримати кон'юнктивну досконалу нормальну форму. Правила утворення КДНФ полягає в наступному:

1. По кожному набору двійкових змінних, при якому дана функція приймає значення нуля, скласти елементарні диз'юнкції (макстерми).
2. В елементарні диз'юнкції записати неінвертованими змінні, задані нулем у відповідному наборі, а інвертованими – ті змінні, які задані одиницею.
3. Елементарні диз'юнкції поєднати знаками кон'юнкції.

Задана в останній таблиці логічна функція трьох змінних P(X,Y,Z) приймає значення нуля при чотирьох наборах аргументів. Отже, КДНФ цієї функції буде складатися із логічного добутку чотирьох елементарних диз'юнкцій (макстермів) третього рангу: $P(X,Y,Z) = (X \vee Y \vee Z) \cdot (X \vee Y \vee \overline{Z}) \cdot (\overline{X} \vee Y \vee Z) \cdot (\overline{X} \vee \overline{Y} \vee Z)$.

Аналітичне представлення логічних функцій є зручнішим порівняно з табличним представленням при аналізі та синтезі лоігчних пристроїв в ЕОМ.

Тема 2.3. Мінімізація логічних функцій

Логічна функція, яка описує принцип функціонування синтезованого цифрового пристрою, може бути представлена у вигляді таблиці, відповідно до якої складаються ДДНФ і КДНФ даної функції.

Потім застосовують низку методів для спрощення аналітичного виразу логічної функції. Основними вимогами до цієї задачі є мінімальна кількість елементарних кон'юнкцій чи диз'юнкцій в логічній формулі і однорідність виконуваних операцій. Перетворення логічної функції з метою спрощення її аналітичного виразу отримало назву мінімізації. Розглянемо деякі методи мінімізації.

Метод безпосередніх перетворень (метод Квайна). На першому етапі застосування цього методу будується ДДНФ логічної функції. Потім дана форма перетворюється і спрощується з використанням різних законів і співвідношень алгебри логіки.

Спочатку доцільно виявити у вихідній ДДНФ так звані сусідні елементарні кон'юнкції, тобто такі, у яких є по одній неспівпадаючій змінній. Наприклад, $X_1, \overline{X}_2, X_3$ і X_1, X_2, X_3; $\overline{X}_1, \overline{X}_2, \overline{X}_3$ і $\overline{X}_1, X_2, \overline{X}_3$ і т.д.

Відносно таких кон'юнкцій застосовується метод склеювання, в результаті чого ранг цих кон'юнкцій знижується на одиницю. Наприклад, $X_1\overline{X}_2X_3 \vee X_1X_2X_3 = X_1X_3(\overline{X}_2 \vee X_2) = X_1X_3$. Кон'юнкції, утворені в результаті склеювання, отримали назву ***імплікант***.

Отримані після першого склеювання імпліканти при можливості склеюються повторно і так до тих пір, поки склеювання можливо. Несклеювані імпліканти називають ***простими***, а формула, яка складається з простих імплікантів, називається ***тупіковою***.

Розглянемо даний метод мінімізації на конкретному прикладі. Нехай логічна функція трьох аргументів F (X_1, X_2, X_3) задана таблицею.

Таблиця 12.

Значення аргументу			Значення функції F	Значення аргументу			Значення функції F
X_1	X_2	X_3		X_1	X_2	X_3	
0	0	0	1	1	0	0	0
0	0	1	1	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	1	1	1	1	0

Утворюємо ДДНф даної функції: F = $\overline{X}_1\overline{X}_2\overline{X}_3 \vee \overline{X}_1\overline{X}_2X_3 \vee \overline{X}_1X_2\overline{X}_3 \vee \overline{X}_1X_2X_3$. Тут перша і друга кон'юнкції, а також третя і четверта є сусідніми. Застосувавши склеювання цих кон'юнкцій, отримаємо F = $\overline{X}_1\overline{X}_2(\overline{X}_3 \vee X_3) \vee \overline{X}_1X_2(\overline{X}_3 \vee X_3) = \overline{X}_1\overline{X}_2 \vee \overline{X}_1X_2$.

До отриманих імплікантів застосуємо повторне склеювання: F = $\overline{X}_1\overline{X}_2 \vee \overline{X}_1X_2 = \overline{X}_1(\overline{X}_2 \vee X_2) = \overline{X}_1$.

Таким чином, в результаті проведених перетворень вихідна функція трьох аргументів виявилась інверсією лише однієї змінної $\overline{X}_1$. В цьому можна також переконатись за таблицею, наведеною вище.

Якщо на певному етапі склеювання утворюється форма R, яка містить члени виду aX_i і $b\overline{X}_i$, то для неї справделиво $R = R \vee ab$. Це дає можливість додати до вихідної форми R декілька членів виду ab в залежності від кількості наявних пар членів aX_i і $b\overline{X}_i$. Після додавання вказаних членів зазвичай стає можливим продовжити подальше склеювання за формулою $ab \vee a\overline{b} = a(b \vee \overline{b}) = a$.

Розглянемо приклад, який ілюструє такий підхід. Нехай була отримана формула для логічної функції трьох аргументів: F (X_1, X_2, X_3) = $X_1X_2 \vee X_1\overline{X}_2\overline{X}_3 \vee X_1\overline{X}_2X_3 \vee \overline{X}_2\overline{X}_3 \vee X_1\overline{X}_3$. У перших двох кон'юнкціях містяться змінні X_2 і $\overline{X}_2$, тому можна додати добуток наявних при них змінних, тобто $X_1 \cdot X_1\overline{X}_3 = X_1\overline{X}_3$. У третій і четвертій кон'юнкціях містяться змінні X_3 і $\overline{X}_3$, тому до них можна додати $X_1\overline{X}_2 \cdot \overline{X}_2 = X_1\overline{X}_2$. В результаті отримаємо F (X_1, X_2, X_3) = $X_1X_2 \vee X_1\overline{X}_2\overline{X}_3 \vee X_1\overline{X}_2X_3 \vee \overline{X}_2\overline{X}_3 \vee X_1\overline{X}_3 \vee X_1X_3 \vee X_1\overline{X}_2$.

Тепер можна продовжити подальше спрощення: F (X_1, X_2, X_3) = $X_1(X_2 \vee \overline{X}_2) \vee X_1\overline{X}_2(\overline{X}_3 \vee X_3) \vee \overline{X}_2\overline{X}_3 \vee X_1\overline{X}_3 = X_1 \vee X_1\overline{X}_2 \vee \overline{X}_2\overline{X}_3 \vee X_1\overline{X}_3 = X_1(1 \vee \overline{X}_2 \vee \overline{X}_3) \vee \overline{X}_2\overline{X}_3 = X_1 \vee \overline{X}_2\overline{X}_3$. В отриманому виразі можна замінити кон'юнкцію заперечень $\overline{X}_2\,\overline{X}_3$ заперченням диз'юнкції $\overline{X_2 \vee X_3}$. Після такої заміни операції, які використовуються в лгічному виразі, стали однорідними, тобто F (X_1, X_2, X_3) = $X_1 \vee \overline{X_2 \vee X_3}$.

Метод Карно-Вейча.

Метод діаграм Вейча, удосконалений Карно, застосовують у тому випадку, коли число аргументів не перевищує 5-6. Для мінімізації функції використовують карти Карно, що представляють собою таблиці, розділені на клітинки. Число клітинок дорівнює числу різних наборів аргументів, причому кожній клітинці відповідає певний набір аргументів.

На мал.1,а показана картка Карно для двох аргументів. Аналогічно складають карти для трьох (мал.1,б) і чотирьох (мал.1,в) аргументів, які містять відповідно 8 і 16 клітинок.

	X_2	$\overline{X}_2$
X_1	$X_1 X_2$	$X_1 \overline{X}_2$
$\overline{X}_1$	$\overline{X}_1 X_2$	$\overline{X}_1 \overline{X}_2$

Мал. 1а

	$\overline{X}_1$		X_1	
$\overline{X}_3$	$\overline{X}_1\overline{X}_2\overline{X}_3$	$\overline{X}_1 X_2\overline{X}_3$	$X_1 X_2\overline{X}_3$	$X_1\overline{X}_2\overline{X}_3$
X_3	$\overline{X}_1\overline{X}_2 X_3$	$\overline{X}_1 X_2 X_3$	$X_1 X_2 X_3$	$X_1\overline{X}_2 X_3$
	$\overline{X}_2$	X_2		$\overline{X}_2$

Мал. 1б

	X_2		$\overline{X}_2$		
X_1	$X_1 X_2\overline{X}_3\overline{X}_4$	$X_1 X_2\overline{X}_3 X_4$	$X_1\overline{X}_2\overline{X}_3 X_4$	$X_1\overline{X}_2\overline{X}_3\overline{X}_4$	$\overline{X}_3$
	$X_1 X_2 X_3\overline{X}_4$	$X_1 X_2 X_3 X_4$	$X_1\overline{X}_2 X_3 X_4$	$X_1\overline{X}_2 X_3\overline{X}_4$	X_3
$\overline{X}_1$	$\overline{X}_1 X_2 X_3\overline{X}_4$	$\overline{X}_1 X_2 X_3 X_4$	$\overline{X}_1\overline{X}_2 X_3 X_4$	$\overline{X}_1\overline{X}_2 X_3\overline{X}_4$	
	$\overline{X}_1 X_2\overline{X}_3\overline{X}_4$	$\overline{X}_1 X_2\overline{X}_3 X_4$	$\overline{X}_1\overline{X}_2\overline{X}_3 X_4$	$\overline{X}_1\overline{X}_2\overline{X}_3\overline{X}_4$	$\overline{X}_3$
	$\overline{X}_4$	X_4		$\overline{X}_4$	

Мал. 1в

Таблиця 13. Значення функції F від трьох змінних

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	1	0	0	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	0
0	1	1	1	1	1	1	1

В клітинки карти Карно, що відповідають тим наборам аргументів, для яких функція, що мінімізується, у відповідності до заданої таблиці приймає значення 1, заносяться одиниці. Наприклад, для функції трьох аргументів $F(X_1, X_2, X_3)$, яка задана таблицею 13, карта Карно має вигляд, показаний на мал.2а.

	$\overline{X}_1$		X_1	
$\overline{X}_3$				1
X_3		1	1	1
	$\overline{X}_2$	X_2		$\overline{X}_2$

Мал. 2а

Дана функція приймає значення одиниці при чотирьох наборах змінних. У відповідних клітинках карти Карно проставлені одиниці. Правила застосування карт Карно полягають у наступному:

Якщо одиниці знаходяться в двох сусідніх клітинках рядка, стовбчика чи на протилежних кінцях будьякого рядка чи стовбчика, то відповідні цим одиницям кон'юнкції (див. мал.2,б) замінюються однією кон'юнкцією на ранг нижче, причому в неї включаються змінні з одинаковими показниками інвертування.

Якщо чотири клітинки складають великий квадрат, рядок чи стовбчик, то відповідні їм кон'юнкції заміняються однією на два ранги нижче, в якій включені змінні з однаковими показниками інвертування.

Кон'юнкції, які відповідають решті одиниць, зберігаються без змін.

Розглянемо застосування карт Карно на прикладах. Для функції, заданої таблицею 13, карта Карно представлена на мал.2,а. В правому крайньому стовбчику цієї карти сусіднім одиницям відповідають кон'юнкції (див. мал.1,б) $X_1\overline{X}_2\overline{X}_3$ і $X_1\overline{X}_2X_3$, які замінюються однією кон'юнкцією на ранг менше з одиниковими показниками інвертування, тобто $X_1\overline{X}_2$. Дві останні одиниці знаходяться в сусідніх клітинках другого рядка, тому відповідні їм кон'юнкції $\overline{X}_1X_2X_3$ і $X_1X_2X_3$ замінюються однією X_2X_3. В результаті скорочена (чи мінімальна) диз'юнктивна нормальна форма даної функції прийме вигляд $F(X_1, X_2, X_3) = X_1\overline{X}_2 \vee X_2X_3$. Дана форма значно простіша ДДНФ вихідної функції, яка відповідно до таблиці 13 має вигляд $F(X_1, X_2, X_3)$ $= \overline{X}_1 X_2 X_3 \vee X_1 \overline{X}_2 \overline{X}_3 \vee X_1 \overline{X}_2 X_3 \vee X_1 X_2 X_3$.

Розглянемо тепер приклад мінімізації функції чотирьох аргументів, ДДНФ якої задана у вигляді $F(X_1, X_2, X_3, X_4) = X_1 \overline{X}_2 \overline{X}_3 X_4 \vee X_1 \overline{X}_2 \overline{X}_3 \overline{X}_4 \vee X_1 \overline{X}_2 X_3 X_4 \vee X_1 \overline{X}_2 X_3 \overline{X}_4 \vee X_1 X_2 X_3 \overline{X}_4 \vee \overline{X}_1 X_2 X_3 \overline{X}_4 \vee \overline{X}_1 X_2 X_3 X_4$.

За картою Карно (мал.2,б) кон’юнкції, що утворюють великий квадрат, знижуються на два ранги; кон’юнкції в двох сусідніх клітинках знижуються на один ранг, а одинична кон’юнкція залишається без змін.

	X_2		$\overline{X}_2$		
X_1			1	1	$\overline{X}_3$
	1		1	1	X_3
$\overline{X}_1$	1	1			
					$\overline{X}_3$
	$\overline{X}_4$	X_4		$\overline{X}_4$	

Мал. 2б

В результаті отримаємо:

$F(X_1, X_2, X_3, X_4) = X_1 \overline{X}_2 \vee X_2 X_3 \overline{X}_4 \vee \overline{X}_1 X_2 X_3 X_4$.

Метод Карно-Вейча спрощує знаходження зклеюваних кон’юнкцій в ДДНФ вихідної логічної функції.

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Наведіть основні визначення алгебри логіки
2. Опишіть операції алгебри логіки
3. Наведіть основні закони алгебри логіки.
4. Опишіть алгебри перемикальних функцій
5. Опишіть досконалі нормальні форми.
6. Опишіть метод Квайна для мінімізації логічних функцій.
7. Опишіть мінімізацію логічних функцій за допомогою карт Карно.
8. Опишіть синтез комбінаційних схем у різних елементних базисах
9. Опишіть методи синтезу цифрових автоматів з пам'яттю.

Додаток А
(рекомендований)

Інструкції для виконання лабораторних робіт

ЛАБОРАТОРНЕ ЗАНЯТТЯ №1

Тема: "Переведення десяткових чисел у двійкову систему числення".

Мета: засвоєння на практиці правил перетворення десяткових чисел у двійкову систему числення.

Теоретичні відомості

1. *Табличний спосіб перетворення десяткових чисел в систему з іншою основою.*

Сутність цього способу полягає у тому, що вихідне десяткове число представляється у вигляді суми цілих ступенів нової основи, вибраних відповідної таблиці цих степенів. Послідовність розміщення цих степенів починається з тої, яка дає число, що не перевищує вихідне десяткове. Підбирають для степенів відповідні коефіцієнти з алфавіту нової системи таким чином, щоб їх сума дорівнювала вихідному десятковому числу

2. *Переведення цілих десяткових чисел в іншу систему числення способом ділення*

Для переводу цілого числа з десяткової системи числення в іншу систему з основою S треба це число послідовно ділити на основу S нової системи доти поки не отримаємо ділене < S.

Число в новій системі запишеться в вигляді остач ділення, починаючи з останньої. Ця остання остача дає цифру старшого розряду в нові системі числення. Ділення виконують у вихідній системі числення.

Хід роботи

1. ***Виконайте завдання по варіантах: (№ варіанта=номер за списком групи-8, наприклад для студента, що в списку групи під номером 15, №варіанта=15-8=7)***

Завдання №1. Перевести задане число з десяткової системи числення в двійкову систему числення табличним методом:

№ з/п	№ варіанту							
	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
1	1548	4120	6321	5422	2345	3246	6151	5555
2	259	186	223	179	355	312	441	363
3	55	74	62	87	92	73	49	88

Завдання №2. Перевести задане число з десяткової системи числення в двійкову систему числення методом ділення:

№ з/п	№ варіанту							
	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
1	7451	6325	5595	6254	5564	4651	7129	6333
2	366	238	150	342	129	201	415	445
3	59	71	86	64	91	77	83	69

Завдання №3. Перевести правильний дріб з десяткової системи числення в двійкову систему числення з точністю до чотирьох знаків після коми:

№ з/п	№ варіанту							
	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
1	0,865	0,715	0,692	0,763	0,809	0,677	0,755	0,555
2	0,5142	0,3538	0,6565	0,7221	0,6383	0,8141	0,5160	0,4982

Завдання №4. Перевести задане число з десяткової системи числення в двійкову систему числення з точністю до п'яти знаків після коми:

№ з/п	№ варіанту							
	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
1	452,33	612,35	541,19	436,48	579,22	608,46	397,68	566,92
2	99,71	65,85	81,37	69,56	73,41	82,82	75,67	88,55

2. Заповніть таблицю результатів виконання роботи:

Десяткові числа	Двійкові числа

3. Оформіть звіт.

Контрольні запитання

1. В чому полягає суть табличного способу перетворення десяткових чисел у двійкову систему числення?

В чому полягає суть способу ділення для перетворення десяткових чисел у двійкову систему числення?

ЛАБОРАТОРНА РОБОТА №2

Тема: "Додавання та віднімання двійкових чисел".

Мета: засвоєння на практиці правил додавання і віднімання двійкових чисел.

Теоретичні відомості

Для додавання чисел у машинах з фіксованою крапкою повинна бути виконана така послідовність дій: перетворення прямого коду вихідних чисел у зворотний (чи додатковий); порозрядне додавання кодів (з урахуванням можливих переносів з розряду в розряд); перетворення результату знову в прямий код при його передачі в інші пристрої машини.

При порозрядному додаванні зворотних (чи додаткових) кодів знакові розряди складають як розряди мантис. Якщо в результаті додавання утвориться одиниця переносу зі знакового розряду, то при використанні зворотних кодів її додають до молодшого розряду суми (ця операція називається циклічним переносом), а у випадку додаткових кодів цей перенос не враховують.

Хід роботи

3. Виконайте завдання по варіантах:

Завдання №1. Переведіть задані десяткові числа в двійкову систему числення, подайте у формі з фіксованою комою і додайте їх в зворотніх і додаткових кодах, а результат запишіть в десятковій формі:

	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
X_1	-8	10	11	-6	13	-5	6	-8
Y_1	5	-7	-5	7	-6	10	-7	12
X_2	-9	-8	-6	-12	-4	-14	-5	-7
Y_2	-14	-6	-9	-3	-9	-6	-9	-7

Завдання №2. Переведіть задані десяткові числа в двійкову систему числення, подайте у формі з плаваючою комою і додайте їх в модифікованих кодах, а результат запишіть в десятковій формі:

	Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
X	-11	14	12	-10	15	-8	12	-13
Y	5	-7	-5	7	-6	10	-7	9

4. Заповніть таблицю результатів роботи:

Десяткові числа	Двійкові числа	Сума чисел

3. Зробіть висновок. Оформіть звіт.

Контрольні запитання

2. Яку послідовність дій треба виконати, щоб виконати додавання чисел у машинах з фіксованою крапкою?

Яку послідовність дій треба виконати, щоб виконати додавання чисел у машинах з плаваючою крапкою?

ЛАБОРАТОРНА РОБОТА №3

Тема: “ **Мінімізація логічних функцій** ”.

Мета: засвоєння на практиці правил мінімізації логічних функцій.

Теоретичні відомості

Метод безпосередніх перетворень (метод Квайна). На першому етапі застосування цього методу будується ДДНФ логічної функції. Потім дана форма перетворюється і спрощується з використанням різних законів і співвідношень алгебри логіки. Спочатку доцільно виявити у вихідній ДДНФ так звані сусідні елементарні кон’юнкції, тобто такі, у яких є по одній неспівпадаючій змінній. Відносно таких кон’юнкцій застосовується метод склеювання, в результаті чого ранг цих кон’юнкцій знижується на одиницю. Кон’юнкції, утворені в результаті склеювання, отримали назву ***імплікант***. Отримані після першого склеювання імпліканти при можливості склеюються повторно і так до тих пір, поки склеювання можливо.

Метод карт Карно. Для мінімізації функції використовують карти Карно, що представляють собою таблиці, розділені на клітинки. Число клітинок дорівнює числу різних наборів аргументів, причому кожній клітинці відповідає певний набір аргументів. В клітинки карти Карно, що відповідають тим наборам аргументів, для яких функція, що мінімізується, у відповідності до заданої таблиці приймає значення 1, заносяться одиниці. Якщо одиниці знаходяться в двох сусідніх клітинках рядка, стовбчика чи на протилежних кінцях будьякого рядка чи стовбчика, то відповідні цим одиницям кон’юнкції замінюються однією кон’юнкцією на ранг нижче, причому в неї включаються змінні з одинаковими показниками інвертування. Якщо чотири клітинки складають великий квадрат, рядок чи стовбчик, то відповідні їм кон’юнкції заміняються однією на два ранги нижче, в якій включені змінні з однаковими показниками інвертування. Кон’юнкції, які відповідають решті одиниць, зберігаються без змін.

Хід роботи

5. Виконайте завдання по варіантах:

Складіть ДДНФ за заданою таблицею аргументів та спростіть її методами Квайна та карт Карно:

Варіант 1

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	1	0	0	1	1	1	0	0	1	0	1	0	0	1	0

Вріант 2

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	0	0	0	1	1	0	0	1	1	1	0	1	0	0	1

Вріант 3

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1

Вріант 4

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	0

Вріант 5

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	0	0	0	0	1	0	1	1	1	0	1	1	0	0	1

Вріант 6

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	0	1	1	0	1	1	0	1	0	1	0	1	0	1	0

Вріант 7

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	1	0	0	1	1	0	0	0	1	1	0	0	0	1	1

Вріант 8

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	0	0	1	0	1	0	1	1	0	0	0	1	1	0	1

6. Зробіть висновок.

3. Оформіть звіт.

Контрольні запитання

3. Що таке мінімізація логічної функції?
4. Що таке сусідні елементарні кон’юнкції?

В чому полягає метод склеювання?

Додаток Б
(рекомендований)

Інструкції для виконання практичних робіт

ПРАКТИЧНЕ ЗАНЯТТЯ №1

Тема: Перетворення цілих чисел в системи з іншою основою.

Мета: засвоїти на практиці методи та правила перетворення цілих чисел в системи з іншою основою.

Хід роботи

1. Повторіть правила перетворення цілих чисел в системи з іншою основою.

2. Виконайте завдання по варіантах: (номер варіанта відповідає або кратний номерові студента в списку групи, тобто якщо номер студента в списку групи-1, то його віріант-№1, якщо номер у списку-8, то варіант-№8; для студентів, у яких номер у списку групи більший ніж 8, номер варіанта можна обчислити за виразом: № варіанта=номер за списком групи-8, наприклад для студента, що в списку групи під номером 15, №варіанта=15-8=7)

Завдання №1. Перевести задане число з десяткової системи числення в двійкову, вісімкову та шістнадцяткову системи числення табличним методом:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
223	179	312	259	441	355	186	274

Завдання №2. Перевести задане число з десяткової системи числення в двійкову та вісімкову системи числення методом ділення:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
366	238	150	342	129	201	415	288

Завдання №3. Перевести задане число в десяткову систему числення:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
177_8	451_8	407_8	17_8	263_8	146_8	274_8	244_8
$C5_{16}$	$4E_{16}$	$A2_{16}$	$5B_{16}$	312_{16}	CA_{16}	143_{16}	$A4_{16}$
100111_2	101101_2	110011_2	111100_2	110010_2	100011_2	110101_2	1000011_2

Завдання №4. Перевести задане число з десяткової системи числення в двійкову та шістнадцяткову системи числення з точністю до п’яти знаків після коми:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
29,51	38,35	50,65	34,22	29,38	20,81	45,16	36,42

3. Зробіть висновок.

4. Оформіть звіт.

Контрольні запитання

1. Як називається сукупність правил для позначення (запису) чисел за допомогою цифр?
2. Яку помилку містить запис $[X]_{дод}=01,10100$?
3. Що називається сукупністю цифр або інших символів для позначення (запису) чисел в певній системі числення?

ПРАКТИЧНЕ ЗАНЯТТЯ №2

Тема: Утворення машинних кодів.

Мета: засвоїти на практиці методи та правила утворення машинних кодів.

Хід роботи

1. Повторіть правила утворення машинних кодів.

2. Виконайте завдання по варіантах:

Завдання №1. Запишіть двійкове число в прямому, зворотному, додатковому і модифікованому кодах:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
0,1000011	0,111001	0,110101	0,110010	0,101010	0,100000	0,1000110	0,100011
-0,10011010	-0,10101010	-0,1101010	-0,1101100	-0,101100	-0,111000	-0,100011	-0,101010
-0,1111001	-0,1001101	-0,1101101	-0,1101110	-0,111110	-0,100011	-0,110011	-0,110001
0,1100	0,1010	0,1011	0,1010	0,1111	0,1000	0,1001	0,1100

Завдання №2. Числа, що записані в зворотному коді перетворіть в звичайний двійковий запис:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
0,0100	0,0100	0,01000	0,01111	0,001101	1,010101	1,11011	0,101101
1,011100	1,00010	1,0100010	1,011010	1,001101	0,00110	0,00110	1,001001
0,00001	0,00101	0,010101	0,011011	0,101101	0,10101	0,00011	0,111101

Завдання №3. Числа, що записані в додатковому коді перетворіть в звичайний двійковий запис:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
0,001101	1,010101	1,11011	0,01000	0,0100	0,01111	0,0100	0,11000
1,001101	0,00110	0,00110	1,0100010	1,011100	1,011010	1,00010	1,0101110
0,101101	0,10101	0,00011	0,010101	0,00001	0,011011	0,00101	0,110111

Завдання №4. Числа, що записані в додатковому модифікованому коді перетворіть в звичайний двійковий запис:

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5	Варіант 6	Варіант 7	Варіант 8
00,111011	00,111001	11,010010	00,011101	11,010010	00,011011	11,011100	11,011010
11,01110	11,01110	00,00110	11,01110	00,10001	11,10101	00,11110	00,10110

3. Зробіть висновок.

4. Оформіть звіт.

Контрольні запитання

1. Чому в ЕОМ застосовують тільки схеми додавання?
2. Яку помилку містить запис $[X]_{зв}=01,10100$?
3. За яким правилом отримується додатковий код від'ємного числа?

ПРАКТИЧНЕ ЗАНЯТТЯ №3

Тема: Мінімізація логічних функцій методом Квайна.

Мета: засвоїти на практиці методи та правила мінімізації логічних функцій методом Квайна.

Хід роботи

1. Повторіть правила мінімізації логічних функцій методом Квайна.

2. Виконайте завдання по варіантах:

Завдання №1. Складіть таблицю істинності для заданої логічної функції, утворіть ДДНФ та спростіть її методами Квайна:

Варіант 1: $f = x_1 \sim (x_2 \rightarrow x_3) \vee \overline{x_1}$

Варіант 2: $f = x_1 \leftarrow (x_2 \wedge x_3) \vee \overline{x_1}$

Варіант 3: $f = x_2 \sim (\overline{x_2} \rightarrow x_3) \vee x_1$

Варіант 4: $f = x_1 \vee (x_2 \rightarrow x_3) \sim \overline{x_3}$

Варіант 5: $f = x_1 \leftarrow ((x_2 \sim x_1) \vee \overline{x_3})$

Варіант 6: $f = x_3 \wedge (x_2 \rightarrow \overline{x_3}) \sim x_1$

Варіант 7: $f = (x_3 \wedge x_1) \leftarrow (x_2 \rightarrow \overline{x_3})$

Варіант 8: $f = (x_1 \sim x_3) \vee (x_2 \rightarrow \overline{x_3})$

Завдання №2. Складіть ДДНФ за заданою таблицею аргументів та спростіть її методом Квайна:

Варіант 1

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1

Вріант 2

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0

Вріант 3

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	0	0	1	1	0	0	1	1	0	1	1	1	1	1	0

Вріант 4

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	1	1	0	0	1	0	1	0	1	1	0	0	1	1	1

Вріант 5

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	0	0	1	1	1	0	1	0	1	0	1	0	0	1	0

Вріант 6

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	0	0	1	1	0	0	1	1	0	0	1	0	1	1	1

Вріант 7

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	0	0	1	1	1	1	1	1	0	0	0	1	1	0	0	0

Вріант 8

X_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
X_2	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1
X_3	0	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1
X_4	0	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1
F	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1

3. Зробіть висновок.

4. Оформіть звіт.

Контрольні запитання

5. Що таке ранг кон'юнкції?
6. Що таке імпліканти?
7. Які імпліканти називають простими?

Яка формула називається тупиковою?

ПРАКТИЧНЕ ЗАНЯТТЯ №4

Тема: Мінімізація логічних функцій за допомогою карт Карно.

Мета: засвоїти на практиці методи та правила мінімізації логічних функцій методом карт Карно.

Теоретичні відомості

Метод карт Карно. Для мінімізації функції використовують карти Карно, що представляють собою таблиці, розділені на клітинки. Число клітинок дорівнює числу різних наборів аргументів, причому кожній клітинці відповідає певний набір аргументів. В клітинки карти Карно, що відповідають тим наборам аргументів, для яких функція, що мінімізується, у відповідності до заданої таблиці приймає значення 1, заносяться одиниці. Якщо одиниці знаходяться в двох сусідніх клітинках рядка, стовбчика чи на протилежних кінцях будь-якого рядка чи стовбчика, то відповідні цим одиницям кон'юнкції замінюються однією кон'юнкцією на ранг нижче, причому в неї включаються змінні з однаковими показниками інвертування. Якщо чотири клітинки складають великий квадрат, рядок чи стовбчик, то відповідні їм кон'юнкції заміняються однією на два ранги нижче, в якій включені змінні з однаковими показниками інвертування. Кон'юнкції, які відповідають решті одиниць, зберігаються без змін.

Хід роботи

1. Повторіть правила мінімізації логічних функцій методом карт Карно.

2. Виконайте завдання по варіантах:

Завдання №1. Складіть ДДНФ за заданою таблицею аргументів та спростіть її методом карт Карно:

Варіант 1								
X	0	1	1	0	0	1	0	1
Y	0	0	0	0	1	1	1	1
Z	0	1	0	1	1	0	0	1
F	0	1	1	0	1	1	0	1

Варіант 2								
X	1	0	0	1	1	0	0	1
Y	0	0	0	0	1	1	1	1
Z	0	1	0	1	1	1	0	0
F	0	1	1	0	1	0	1	1

Варіант 3								
X	1	1	1	0	0	0	0	1
Y	1	0	1	0	0	1	1	0
Z	0	1	1	1	0	1	0	0
F	0	1	0	1	1	0	1	1

Варіант 4								
X	0	1	0	0	1	1	0	1
Y	0	0	1	0	1	0	1	1
Z	0	0	0	1	0	1	1	1
F	0	0	1	0	1	1	1	1

Варіант 5								
X	0	1	0	0	1	0	1	1
Y	0	0	1	0	1	1	0	1
Z	0	0	0	1	0	1	1	1
F	1	1	1	0	0	0	1	1

Варіант 6								
X	1	1	0	1	1	0	0	0
Y	1	1	1	0	0	1	0	0
Z	1	0	1	1	0	0	1	0
F	0	0	0	1	1	1	1	1

Варіант 7								
X	1	1	0	0	0	1	1	0
Y	0	1	1	0	1	0	1	0
Z	0	1	0	1	1	1	0	0
F	1	1	1	0	1	0	0	1

Варіант 8								
X	1	1	1	1	0	0	0	0
Y	0	0	1	1	1	1	0	0
Z	1	0	0	1	1	0	1	0
F	0	1	0	1	1	0	1	1

Завдання №2. Спростіть логічну функцію методом карт Карно:

Варіант 1

	X_2		$\overline{X}_2$		
X_1			1	1	$\overline{X}_3$
	1		1	1	X_3
$\overline{X}_1$	1	1			
					$\overline{X}_3$
	$\overline{X}_4$	X_4		$\overline{X}_4$	

Варіант 2

	X_2		$\overline{X}_2$		
X_1	1	1	1	1	$\overline{X}_3$
					X_3
$\overline{X}_1$	1	1			
			1	1	$\overline{X}_3$

	$\bar{X}_4$	X_4		$\bar{X}_4$	

Варіант 3

	X_2		$\bar{X}_2$		
X_1		1	1		$\bar{X}_3$
		1	1		X_3
$\bar{X}_1$		1	1		
	1			1	$\bar{X}_3$
	$\bar{X}_4$	X_4		$\bar{X}_4$	

Варіант 4

	X_2		$\bar{X}_2$		
X_1	1		1	1	$\bar{X}_3$
	1		1	1	X_3
$\bar{X}_1$	1				
	1		1		$\bar{X}_3$
	$\bar{X}_4$	X_4		$\bar{X}_4$	

Варіант 5

	X_2		$\bar{X}_2$		
X_1	1	1		1	$\bar{X}_3$
	1	1			X_3
$\bar{X}_1$					
	1	1		1	$\bar{X}_3$
	$\bar{X}_4$	X_4		$\bar{X}_4$	

Варіант 6

	X_2		$\overline{X}_2$		
X_1			1	1	$\overline{X}_3$
X_1	1		1	1	X_3
$\overline{X}_1$	1	1			X_3
$\overline{X}_1$					$\overline{X}_3$
	$\overline{X}_4$	X_4	X_4	$\overline{X}_4$	

Варіант 7

	X_2		$\overline{X}_2$		
X_1		1		1	$\overline{X}_3$
X_1		1	1	1	X_3
$\overline{X}_1$	1	1	1		X_3
$\overline{X}_1$	1				$\overline{X}_3$
	$\overline{X}_4$	X_4	X_4	$\overline{X}_4$	

Варіант 8

	X_2		$\overline{X}_2$		
X_1	1	1	1	1	$\overline{X}_3$
X_1	1				X_3
$\overline{X}_1$					X_3
$\overline{X}_1$	1	1	1	1	$\overline{X}_3$
	$\overline{X}_4$	X_4	X_4	$\overline{X}_4$	

3. Зробіть висновок.

4. Оформіть звіт.

Контрольні запитання

8. Чому повинна дорівнювати кількість клітинок в карті Карно?

9. В якому випадку дві кон'юнкції замінюються однією кон'юнкцією на ранг нижчою, причому в неї включаються змінні з однаковими показниками інвертування?

10. В якому випадку чотири кон'юнкції замінюються однією на два ранги нижчою, в якій включені змінні з однаковими показниками інвертування?

ПРАКТИЧНЕ ЗАНЯТТТЯ №5.

Синтез та аналіз комбінаційних схем

<u>Мета роботи:</u> Навчитись на практиці проводити обрахунок логічних функцій та на їх основі створювати та аналізувати логічні схеми.

Хід заняття

1. Повторіть правила синтезу та аналізу логічних схем.

2. Виконайте завдання по варіантах:

Завдання 1. Здійснити синтез логічних схем для таких булевих функцій:

$f = x_1 \sim (x_2 \to x_3) \vee \overline{x_1}$

$f = x_1 \leftarrow (x_2 \wedge x_3) \wedge \overline{x_1}$

$f = x_2 \sim (\overline{x_2} \to x_3) \vee x_1$

$f = x_1 \vee (x_2 \to x_3) \sim \overline{x_3}$

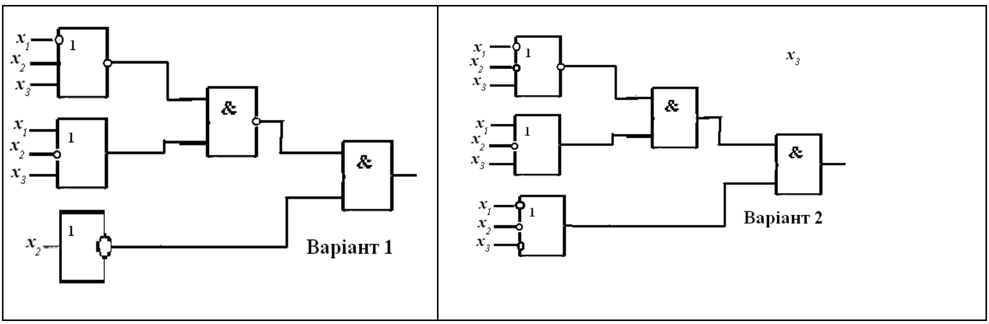
$f = x_3 \leftarrow ((x_2 \sim x_1) \vee \overline{x_1})$

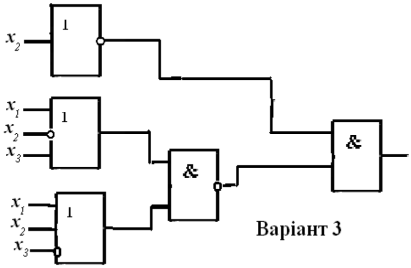
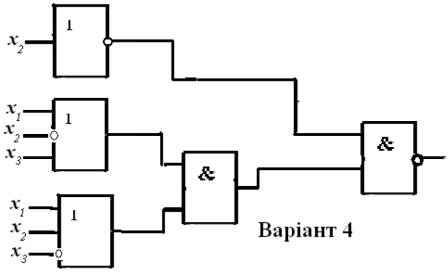
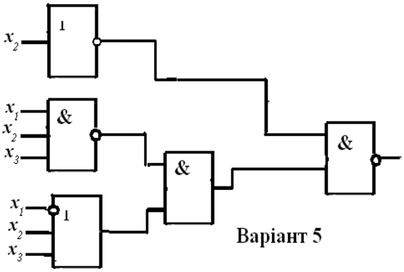
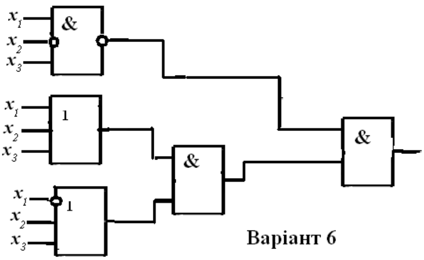
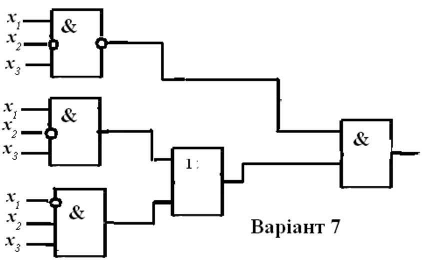
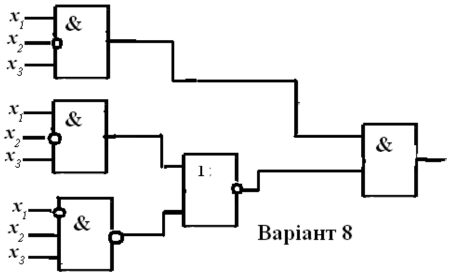
$f = x_3 \wedge (x_2 \to \overline{x_3}) \sim x_1$

$f = (x_3 \wedge x_1) \leftarrow (x_2 \to \overline{x_3})$

$f = (x_1 \sim x_3) \vee (x_2 \to \overline{x_3})$

Завдання 2. Провести аналіз таких логічних ланцюгів:



 Варіант 3	 Варіант 4
 Варіант 5	 Варіант 6
 Варіант 7	 Варіант 8

3. Зробіть висновок.

4. Оформіть звіт.

Контрольні запитання

11. Що виконують на першому етапі синтезу комбінаційних схем?
12. Що виконують на третьому етапі синтезу комбінаційних схем?
13. Яке графічне зображення логічного елемента АБО-НІ?
14. Що виконують на другому етапі аналізу комбінаційних схем?

Список використаної та рекомендованої літератури

1. Костинюк Л.Д. , Паранчук Я.С., Щур І.З. Мікропроцесорні засоби та ситеми.- 2-ге видання. – Львів:Видавництво Національного університету „Львівська політехніка”, 2002. -200 с.
2. Семененко В..А., Балтрушевич А.В. Электронно вычислительное машины:: Учеб. Пособие.-М.: Высшая школа, 1985.-272с.
3. Скаржепа В. А. Луценко А. Н. Электроника и микроэлектроника. - К.:Высшая школа, 1989. - 430 с.
4. Шило В. Л. Популярные цифровые микросхемы. Справочник. -Челябинск: Металургия, 1988. -340с.
5. Аналоговые и цифровые и интегральные микросхемы. Справочное пособие. Под редакцией Якубовского С. В. - 2-е издание переработаное и дополненое - М.: Радио и связь, 1984. -432с.

58,3,56,5,54,7,52,9,50,11,48,13,46,15,44,17,42,19,40,21,38,23,36,25,34,27,32,29

30,31,28,33,26,35,24,37,22,39,20,41,18,43,16,45,14,47,12,49,10,51,8,53,6,55,4,57