

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”



Кафедра ЕОМ

**«Комп’ютерна логіка комбінаційних схем»**

**Навчальний посібник**

з дисципліни **«Комп’ютерна логіка»**

**для студентів першого (бакалаврського) рівня вищої освіти спеціальності 123  
«Комп’ютерна інженерія»**

Рекомендовано

Науково-методичною радою  
Національного університету  
“Львівська політехніка”

Львів – 2021

Укладач проф. каф. ЕОМ, д.т.н., проф. Глухов В. С.,

Рецензенти: проф. каф. СКС, д.т.н., проф. Дунець Р. Б.  
доц. каф. ЕОМ, к.т.н., доц. Березко Л. О.

Рекомендовано Науково-методичною радою  
Національного університету “Львівська політехніка”  
(протокол № \_\_ від \_\_.\_\_.2021 р.)

«Комп’ютерна логіка комбінаційних схем». Навчальний посібник з дисципліни «Комп’ютерна логіка» для студентів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп’ютерна інженерія» / Укл.: В. С. Глухов – Львів: Видавництво Національного університету “Львівська політехніка”, 2021. – 455 с..

До навчального посібника увійшли матеріали лекцій з навчальної дисципліни “Комп’ютерна логіка”. Матеріали містять теоретичні відомості, приклади розв’язання практичних задач, рисунки, таблиці та стосуються першої частини дисципліни – комбінаційних схем.

Навчальний посібник призначений для студентів спеціальності “Комп’ютерна інженерія” галузі знань “Інформаційні технології”.

© Глухов В. С., укладання, 2021

© Національний університет

"Львівська політехніка", 2021

## ЗМІСТ

ЗМІСТ .....	3
ВСТУП .....	6
<b>1 ІНФОРМАЦІЙНІ ОСНОВИ ЦИФРОВИХ АВТОМАТІВ .....</b>	<b>11</b>
1.1 Інформація і загальні принципи її перетворення.....	11
1.2 Обмін інформацією між різними інформаційними пристроями.....	18
1.3 Апаратні засоби збереження й обробки інформації .....	19
1.4 Загальні поняття про цифровий автомат і алгоритм .....	21
1.5 Опис абстрактного автомату .....	22
<b>2 ПРЕДСТАВЛЕННЯ ЧИСЛОВОЇ ІНФОРМАЦІЇ В ЦИФРОВОМУ АВТОМАТІ.....</b>	<b>25</b>
2.1 Системи числення і поняття коду .....	25
2.2 Вибір системи числення .....	32
2.3 Формальні правила двійкової арифметики.....	33
2.4 Переведення числа з однієї позиційної системи числення в іншу.....	37
2.5 Переведення чисел до десяткової системи числення з іншої однорідної позиційної системи числення з основою $k$ , коли дії виконуються в десятковій системі .....	38
2.6 Переведення чисел із десяткової системи числення до іншої однорідної позиційної системи числення з основою $k$ , коли дії виконуються в десятковій системі .....	39
2.7 Переведення цілої частини числа.....	39
2.8 Переведення дробової частини числа .....	40
2.9 Переведення чисел з шістнадцяткової й вісімкової систем до двійкової і зворотне переведення чисел .....	41
<b>3 ФОРМИ ПРЕДСТАВЛЕННЯ ЧИСЕЛ У ЦИФРОВИХ АВТОМАТАХ .....</b>	<b>43</b>
3.1 Форма представлення двійкових чисел з фіксованою комою.....	43
3.2 Представлення від'ємних чисел у форматі з фіксованою комою.....	45
3.3 Форма представлення чисел з рухомою комою.....	47
3.4 Перетворення форматів чисел.....	51
3.5 Похибки представлення чисел .....	54
<b>4 АРИФМЕТИЧНІ ДІЇ З ДВІЙКОВИМИ ЧИСЛАМИ .....</b>	<b>56</b>
4.1 Додавання двійкових чисел.....	56
4.2 Алгебраїчне додавання чисел з фіксованою комою.....	56
4.3 Переповнення розрядної сітки .....	59
4.4 Модифікований доповняльний код.....	60
4.5 Алгебраїчне додавання і віднімання чисел, представлених у формі з рухомою комою.....	61
4.6 Методи множення двійкових чисел .....	63
4.7 Множення чисел, представлених у форматі з фіксованою комою .....	63
4.8 Множення чисел, представлених у формі з рухомою комою .....	64
4.9 Прискорення операції множення .....	65
4.10 Ділення двійкових чисел, представлених у формі з фіксованою комою.....	66
4.11 Ділення двійкових чисел, представлених у форматі з рухомою комою.....	69
4.12 Оцінка точності виконання арифметичних операцій .....	70
4.13 Похибка округлення .....	71
<b>5 ВИКОНАННЯ ОПЕРАЦІЙ НАД ДЕСЯТКОВИМИ ЧИСЛАМИ .....</b>	<b>73</b>
5.1 Представлення десяткових чисел у Д-кодах.....	73
5.2 Формальні правила по-розрядного додавання в Д-кодах.....	74
5.3 Представлення від'ємних чисел у Д-кодах .....	76
5.4 Виконання операцій додавання і віднімання в Д-кодах.....	77
5.5 Множення чисел у Д-кодах .....	78
5.6 Ділення чисел у Д-кодах.....	78

	4
5.7 Переведення чисел з Д-коду в двійковий і з двійкового в Д-код.....	79
<b>6 КОНТРОЛЬ РОБОТИ ЦИФРОВОГО АВТОМАТА .....</b>	<b>81</b>
6.1 Основні поняття теорії кодування.....	81
6.2 Контроль методом парності-непарності .....	81
6.3 Коди Геммінга.....	82
6.4 Контроль за модулем .....	85
6.5 Контроль арифметичних операцій .....	86
<b>7 ОСНОВИ АЛГЕБРИ ЛОГІКИ .....</b>	<b>89</b>
7.1 Основні поняття алгебри логіки.....	89
7.2 Функції алгебри логіки нуля змінної (ФАЛ0) .....	92
7.3 Функції алгебри логіки однієї змінної (ФАЛ1) .....	93
7.4 Функції алгебри логіки двох змінних (ФАЛ2) .....	95
7.5 Основні правила виконання операцій у базисі Буля. ....	106
7.6 Основні правила виконання операцій у базисі Жегалкіна .....	108
7.7 Основні правила виконання операцій у монобазисі І-НЕ .....	108
7.8 Основні правила виконання операцій у монобазисі АБО-НЕ.....	109
7.9 Аналітичне представлення функцій алгебри логіки.....	109
7.10 Досконалі нормальні форми.....	113
7.11 Поліном Жегалкіна .....	116
7.12 Класи функцій алгебри логіки.....	116
7.13 Функціонально повні системи функцій алгебри логіки .....	118
7.14 Числове та геометричне подання логічних функцій .....	119
<b>8 СПРОЩЕННЯ І МІНІМІЗАЦІЯ ЛОГІЧНИХ ФУНКЦІЙ.....</b>	<b>121</b>
8.1 Задача мінімізації.....	121
8.2 Метод Квайна й імплікантні матриці.....	126
8.3 Мінімізація функцій методом Квайна-МакКласкі-Петрика .....	132
8.4 Мінімізація не повністю визначених функцій методом Квайна-МакКласкі-Петрика .....	136
8.5 Метод карт Карно (діаграм Вейча) .....	138
<b>9 АНАЛІЗ ТА СИНТЕЗ ЛОГІЧНИХ ЕЛЕКТРОННИХ СХЕМ.....</b>	<b>145</b>
9.1 Логічні оператори електронних схем.....	145
9.2 Задачі аналізу і синтезу електронних схем.....	148
9.3 Синтез логічних схем з одним виходом у базисі Буля на елементах з довільною кількістю вхідів.....	150
9.4 Синтез логічних схем з одним виходом у базисі Буля на елементах з обмеженою кількістю вхідів.....	153
9.5 Синтез логічних схем з одним виходом у монобазисі І-НЕ.....	154
9.6 Синтез логічних схем з одним виходом у монобазисі 2І-НЕ (Шеффера) .....	157
9.7 Синтез логічних схем з одним виходом у монобазисі АБО-НЕ .....	159
9.8 Синтез логічних схем з одним виходом у монобазисі 2АБО-НЕ (Пірса).....	162
9.9 Порівняння варіантів синтезу комбінаційних логічних схем .....	165
9.10 Синтез логічних схем з одним виходом у базисі Жегалкіна.....	171
<b>10 СИНТЕЗ ЛОГІЧНИХ СХЕМ З ДЕКІЛЬКОМА ВИХОДАМИ .....</b>	<b>174</b>
10.1 Базові комбінаційні вузли.....	174
10.2 Однорозрядний двійковий суматор.....	174
10.3 Однорозрядний двійковий напівсуматор.....	176
10.4 Дешифратор .....	176
10.5 Мультиплексори .....	183
10.6 Постійний запам'ятовуючий пристрій .....	187
10.7 Програмовані логічні матриці .....	191
10.8 Програмовані матриці логіки .....	196

	5
11 ТЕСТИ та відповіді.....	202
12 КОРОТКИЙ ТЛУМАЧЕВИЙ СЛОВНИК.....	442
БІБЛІОТЕЧНИЙ СПИСОК.....	446
13 Рекомендована література.....	452
14 Інформаційні ресурси.....	453
Додаток А Грецька абетка.....	454
НАВЧАЛЬНЕ ВИДАННЯ.....	455
Витяг.....	456
Витяг.....	457
РЕЦЕНЗІЯ.....	458
РЕЦЕНЗІЯ.....	460
РЕЦЕНЗІЯ.....	461
Закладки.....	463
ТАБЛИЦІ.....	465
РИСУНКИ.....	467

## ВСТУП

Навчальний посібник «Комп'ютерна логіка комбінаційних схем» частково замінює підручник з дисципліни «Комп'ютерна логіка».

Навчальний посібник містить Вступ, основний текст, питання та завдання для самоперевірки та контролю знань студентів, обов'язкові та додаткові задачі та приклади, довідково-інформаційні дані для розв'язання задач (таблиці істинності, схеми тощо), апарат для орієнтації в матеріалах кники (предметний покажчик).

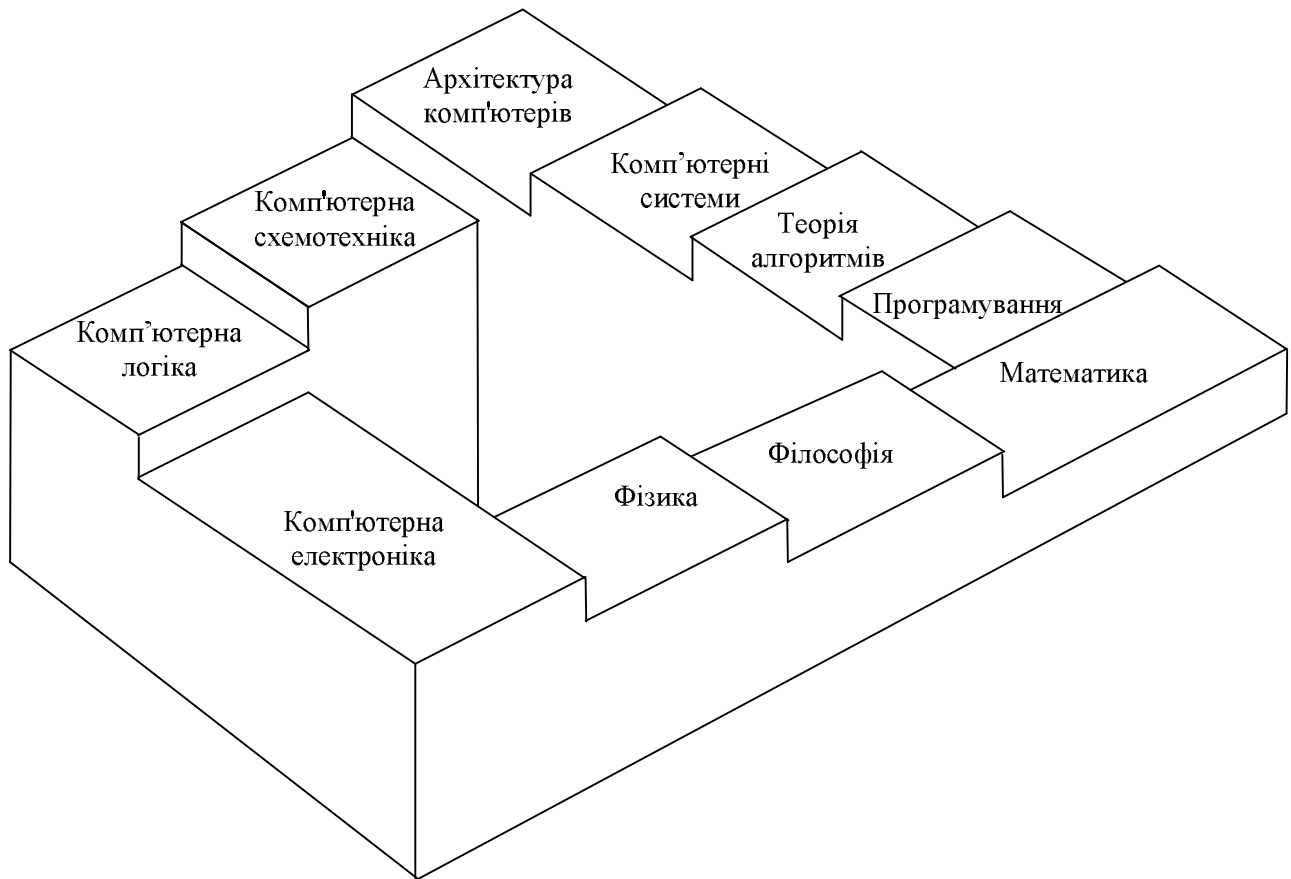
При вивченні дисципліни «Комп'ютерна логіка» кількість годин у навчальному плані, відведених для аудиторних занять складає 90, для самостійної роботи студентів – 90, коефіцієнт, що враховує продуктивність засвоєння 1 авторського аркушу навчальної літератури студентом з одну годину самостійної роботи з літературою, розв'язання задач, прикладів, тощо прийнято за 0,14. Коефіцієнт виду видання для навчального посібника прийнято від 0,5 до 1 При цьому обсяг навчального посібника складає не менше 13 авторських аркушів (не менше 277 сторінок даного видання).

«Комп'ютерна логіка комбінаційних схем» охоплює приблизно половину курсу Комп'ютерна логіка. За рамками посібника залишається розгляд логіки роботи комп'ютерних схем із пам'яттю – схем із зворотніми зв'язками.

Місце дисципліни «Комп'ютерна логіка» серед основних дисциплін напряму „Комп'ютерна інженерія” ілюструє Рис. І. Комп'ютерна логіка спирається на знання, визначення, термінологію, закони, які розвивають математика, фізика, філософія і, в свою чергу, є основою для розвитку комп'ютерної схемотехніки, архітектури комп'ютерів, теорії побудови комп'ютерних систем та мереж, теорії алгоритмів, програмування. А це веде для чергового етапу розвитку математики, фізики, філософії.

Метою вивчення дисципліни “Комп'ютерна логіка” є засвоєння термінології комп'ютерної інженерії та розуміння основних законів, правил та методів алгебри логіки і на їх основі оволодіння основними методами аналізу та синтезу операційних та керуючих вузлів сучасних комп'ютерів як цифрових автоматів (ЦА), що складаються з цифрових комбінаційних схем та схем з пам'яттю.

Вивчення навчальної дисципліни передбачає формування у здобувачів освіти таких загальних компетентностей:



*Рис. 1. Комп'ютерна логіка і „Комп'ютерна інженерія”*

ЗК1. Здатність до абстрактного мислення , аналізу і синтезу.

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність спілкуватися іноземною мовою.

ЗК15. Потенціал до подальшого навчання.

Вивчення навчальної дисципліни передбачає формування у здобувачів освіти таких фахових компетентностей:

ФК10. Здатність здійснювати організацію робочих місць, їхнє технічне оснащення, розміщення комп'ютерного устаткування, використання організаційних, технічних, алгоритмічних та інших методів і засобів захисту інформації.

ФК12. Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання.

У результаті вивчення навчальної дисципліни здобувач освіти повинен бути здатним продемонструвати такі результати навчання:

1. З письмової комунікації (на рівні передавання складної інформації);

2. З усної комунікації (на рівні передавання складних повідомлень);
3. Здатність спілкуватися першою (рідною) мовою, вміння правильно, логічно, ясно будувати своє усне й писемне мовлення;
4. Знання та розуміння предметної галузі, професії; основних концепцій, базових категорій, технічних понять;
5. Здатність до безперервного та активного навчання, самоосвіти, постійного підвищення кваліфікації;
6. Аналітичного мислення (на рівні застосування загального аналізу);
7. Здатність до абстрактного мислення, аналізу та синтезу;
8. Здатність визначати, формулювати та розв'язувати задачі, аналізувати соціально-значущі процеси та приймати обґрунтовані рішення;
9. Планування та організації (на рівні планування та організація багатокomпонентної складної діяльності);
10. Уміння знаходити, обробляти та аналізувати інформацію з різноманітних джерел; здатність використовувати інформаційно-комунікативні технології;
11. Знання дискретних структур і вміння застосовувати сучасні методи дискретної математики для аналізу і синтезу складних систем виконання кодування інформаційних повідомлень, у тому числі користуючись методами побудови завадозахисних кодів та кодами Хеммінга;
12. Знання теоретичних (логічних та арифметичних) основ побудови сучасних комп'ютерів та їх архітектури, вміння застосовувати їх в процесі побудови та експлуатації при рішенні професійних завдань;
13. Складання логічних виразів;
14. Читання і розуміння логічних виразів;
15. Спрощення логічних виразів;
16. Розроблення комбінаційних схем для реалізації системи перемикальних функцій у заданому елементному базисі, сформулювавши задачу її побудови у термінах теорії перемикальних функцій, виконавши мінімізацію функцій та отримавши необхідні операторні форми з урахуванням засобів уникнення збою в схемах;



17. Виконання абстрактний синтезу цифрового автомата, зробивши формальний опис алгоритму його функціонування у термінах теорії цифрових автоматів та виконавши процедуру мінімізації числа станів автомата;

18. Виконання структурного синтезу синхронних та асинхронних автоматів, застосовуючи способи мінімізації функцій збудження та виходів, а також уникнення збоїв в умовах використання для побудови схеми автомата заданого елементного базису, в тому числі інтегральних схем, що програмуються;

19. Розроблення алгоритму функціонування арифметичного пристрою на підставі форми представлення інформації, алгоритмів виконання арифметичних операцій в різних системах числення в умовах застосування методів контролю роботи пристрою з використанням систем автоматизованого проектування комп'ютерних засобів;

20. Уміння відповідально ставитись до виконуваної роботи та досягти поставленої мети;

21. Уміння застосовувати знання і розуміння для розв'язання задач синтезу та аналізу цифрових вузлів;

22. Уміння здійснювати пошук інформації в різних джерелах для розв'язання задач спеціальності;

23. Уміння сприймати критику, критикувати особистість, самокритично відноситись до своїх поступків та критикувати результати роботи;

24. Уміння застосовувати базові знання стандартів в області інформаційних технологій під час розробки цифрових вузлів;

25. Уміння опановувати та розробляти документацію на системи, продукти і сервіси інформаційних технологій.

У результаті вивчення навчальної дисципліни здобувач освіти повинен бути здатним продемонструвати такі програмні результати навчання:

ЗН4. Знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті.

УМ1. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

УМ3. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

УМ4. Вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.

УМ6. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

УМ7. Вміти ефективно працювати як індивідуально, так і у складі команди.

КОМ2. Використовувати інформаційні технології та інші методи для ефективного спілкування на професійному та соціальному рівнях.

АіВ1. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

АіВ2. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного

# 1 ІНФОРМАЦІЙНІ ОСНОВИ ЦИФРОВИХ АВТОМАТІВ

## 1.1 Інформація і загальні принципи її перетворення

В основі будь-якої науково-технічної, біологічної і соціальної системи керування і функціонування, а тим більше в основі систем обчислювальної техніки, лежать інформаційні процеси, зв'язані зі збором і обробленням інформації, її передачею, збереженням, розподілом, відображенням, реєстрацією, зчитуванням і т.д. Із усіх перерахованих інформаційних процесів у підсумку можна виділити чотири основних процеси або процедури: отримання інформації, пересилання (передача), збереження та її перетворення (оброблення). Комп'ютерна логіка вивчає три останні процедури (рис. 1.1).

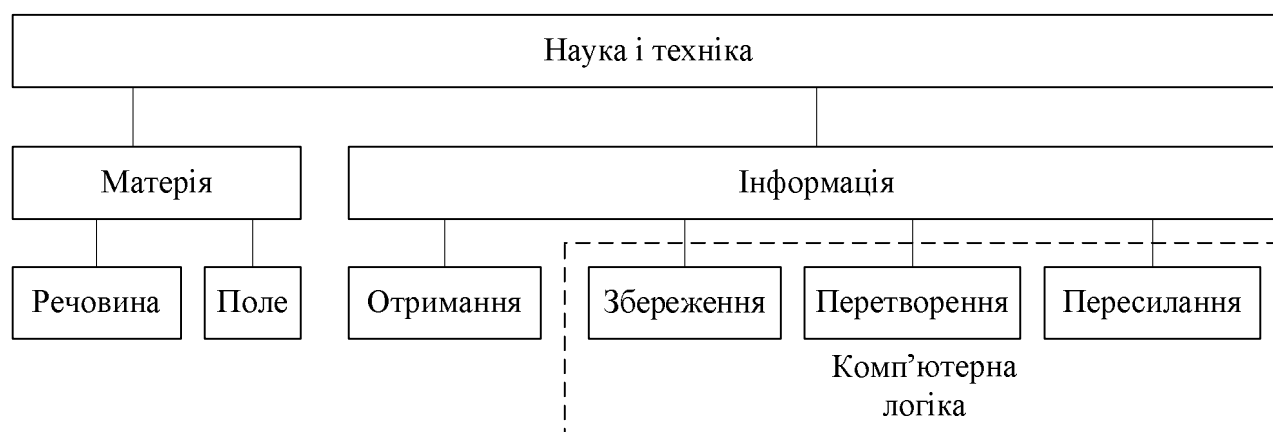


Рис. 1.1. Комп'ютерна логіка у системі наук інформаційної сфери

Реалізація всіх цих процедур у загальному випадку супроводжується перетворенням фізичного носія інформації і форми її представлення (рис. 1.2).

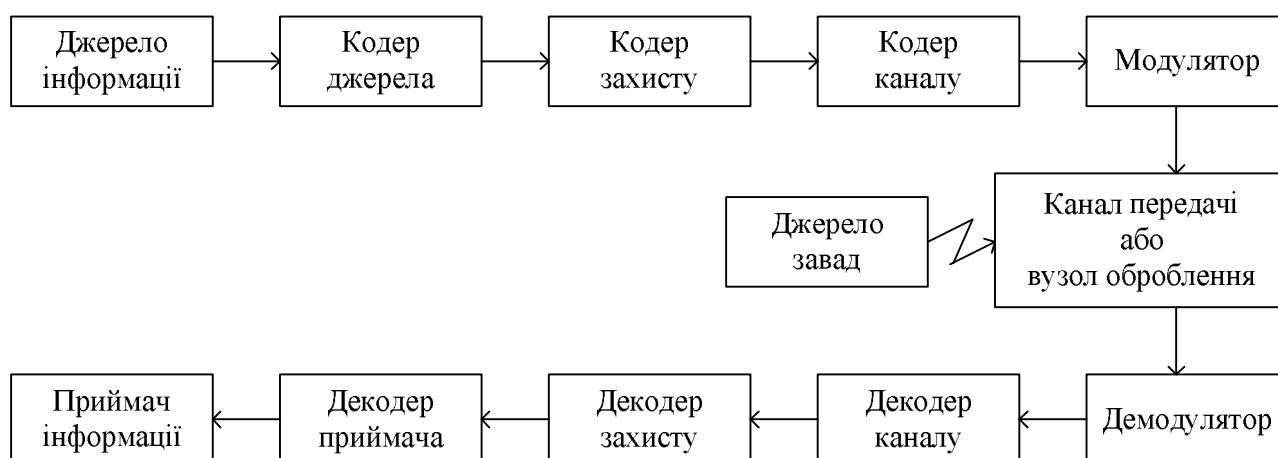


Рис. 1.2. Структурна схема процесу передачі або обробки інформації

Інформація - поняття дуже ємне, яке важко піддається чіткому визначенню. Не зупиняючись на складній проблемі суворого формалізованого і повного визначення

поняття інформації, прийmemo, що інформація є відомості про ті або інші явища або об'єкти, точніше - відомості про певні властивості або параметри цих явищ або об'єктів і про залежності між цими властивостями. Інформація, втілена і зафіксована в деякій матеріальній формі, називається повідомленням, а фізичний засіб передачі повідомлення - сигналом. Або інакше: сигнал - це процес зміни в часі деякого фізичного параметра  $s(t)$  якого-небудь об'єкта. Сигнал використовується для відображення, реєстрації і передачі повідомлення.

Характер зміни сигналу в часі може бути представлений графічно, у вигляді осцилограми; за допомогою таблиці, у яку вносяться значення  $s_i$  у  $i$ -ті моменти часу; сигнал також може бути описаний аналітично.

Для того, щоб зробити сигнал об'єктом теоретичного вивчення і розрахунків, використовують ті або інші способи математичного його опису, тобто створюють математичну модель сигналу. При створенні такої моделі описуються тільки ті властивості сигналу, що представляються об'єктивно важливими й ігнорується велике число другорядних ознак.

Як математична модель сигналу може бути прийнята, наприклад, функціональна залежність, аргументом якої є час:  $s = f(t)$ .

При такій моделі мається на увазі, що причиною зміни величини  $s$  є не сам час, а деяка інша фізична величина. Тому точніше говорити, що позначаючи сигнал через  $s(t)$ ,  $q(t)$ ,... приймається модель, що описує зміну величин  $s$ ,  $q$ ,... у часі.

Існують різного типу сигнали: безперервні (аналогові) сигнали, дискретні, квантовані, кодовані (цифрові), модульовані, імпульсні і т.д.

Як уже відзначалося, практично при будь-якій обробці інформації здійснюються ті або інші перетворення сигналів. Перетворення сигналів, зокрема, застосовуються при зміні носія інформації або самої переданої функції за певним законом, що забезпечує однозначну відповідність між вхідною функцією і вихідною.

У випадку безперервного вхідного сигналу (рис. 1.4) найчастіше першою процедурою зміни типу сигналу є його дискретизація і(або) квантування.

Дискретизація – перетворення неперервної функції в дискретну (це – математичний опис перетворення деякої математичної функції)

Квантування – наближене представлення аналогової величини дискретною

величиною, яка приймає одне з декількох наперед заданих значень [ЕК] (це – сам процес перетворення деякої фізичної величини).

Дискретизація (квантування) неперервного сигналу  $s(t)$  може провадитися в часі або за рівнем (параметром) [ЕК].

Квантування у часі - перетворення сигналу  $x(t)$  у послідовність імпульсів, що йдуть один за одним, амплітуда, тривалість або частота яких залежить від амплітуди вхідного сигналу.

Квантування за рівнем – перетворення сигналу  $x(t)$ , що полягає в заокругленні його миттєвого значення до деякої найближчої наперед заданої фіксованої величини  $xk$ , яка називається рівнем квантування.

Крок квантування (квант) – відстань між двома сусідніми рівнями квантування.

Імпульс - поштовх до дії. Електричний імпульс - короткочасне збільшення (зменшення) напруги або струму.

Амплітуда - найбільше відхилення періодично змінюваної величини від „нульового” значення.

Період (коливаний) – проміжок часу, за який система, що здійснює коливання, повертається до початкового стану.

Фронт - перехідна зона, де відбувається зміна рівня сигналу, перепад рівня сигналу (рис. 1.3).

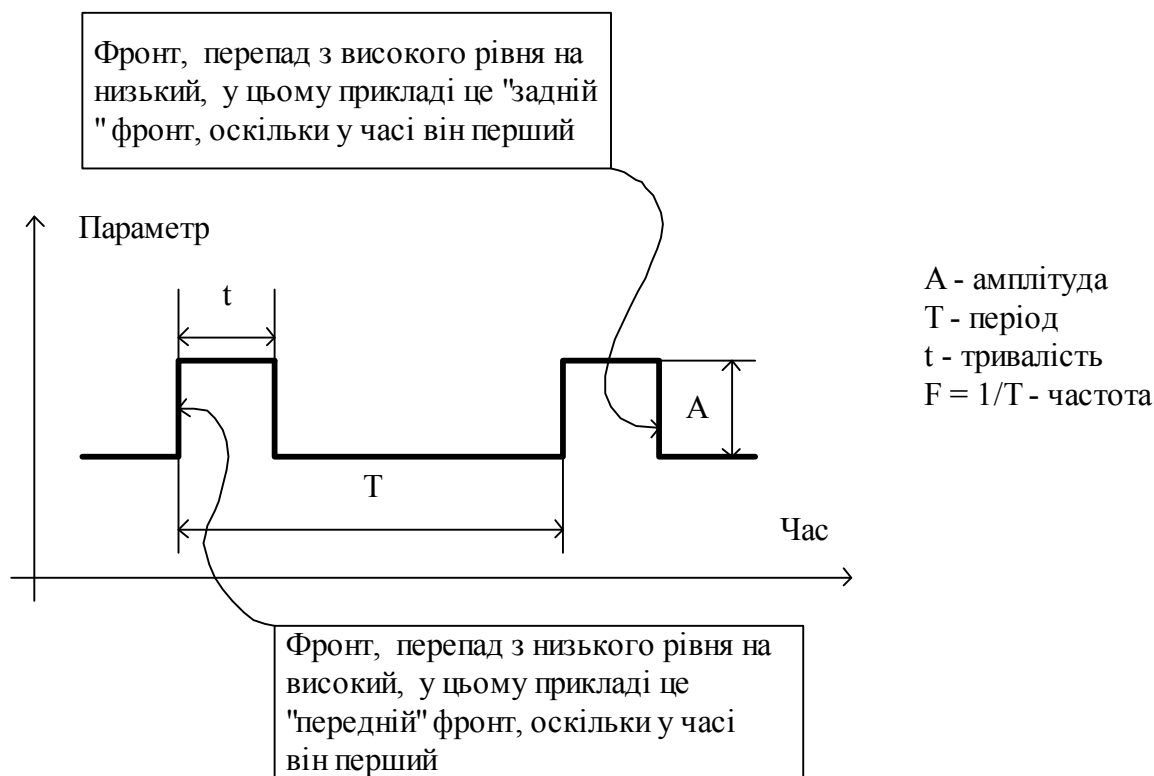


Рис. 1.3. Параметри імпульсу

При квантування у часі дискретний сигнал описується гратчастою функцією (послідовністю, часовим рядом)  $X(n\Delta t)$ , у якій величина  $X$  може приймати будь-які значення в деякому інтервалі  $[X_{min}, X_{max}]$ , у той час, як незалежна змінна  $n$  приймає тільки дискретні значення  $n = 0, 1, 2, \dots, \Delta t$  - інтервал (крок) дискретизації.

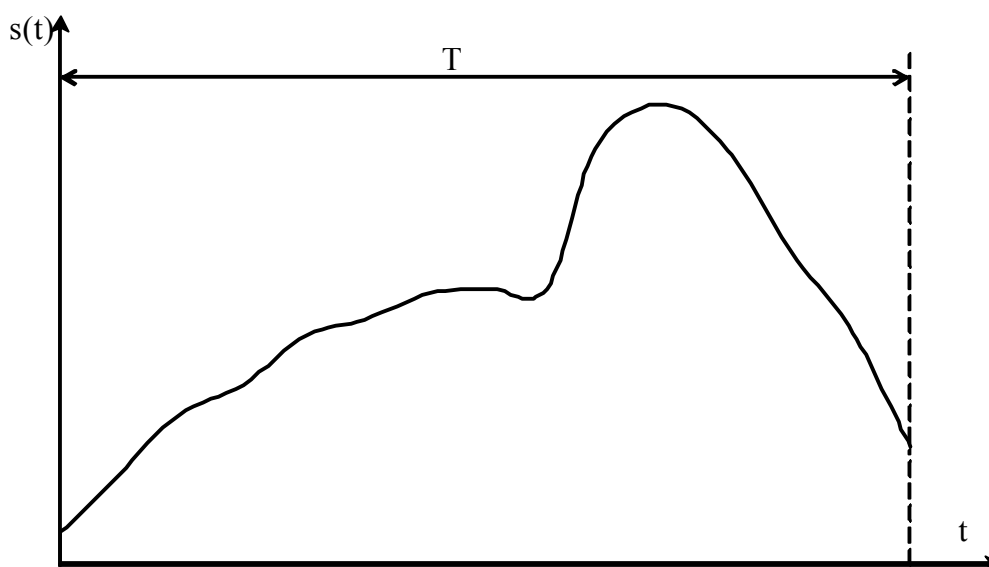


Рис. 1.4. Неперервний сигнал

Коли дискретизація сигналу  $s(t)$  здійснюється в часі, то безперервний сигнал  $s(t)$  замінюється (представляється) послідовністю імпульсних сигналів, амплітуда яких відповідає значенням безперервного сигналу в дискретні моменти часу  $n\Delta t$ . У даному

випадку значення  $\Delta t$  дорівнює часовому інтервалові між суміжними відліками. Тоді  $f = 1/\Delta t$  - це циклічна частота дискретизації, а  $\omega = 2\pi f$ , відповідно, кругова частота.

Прийmemo, що сигнал  $s(t)$  заданий або графічно або ж таблицею, до якої внесені його значення в межах періоду часу  $T$  через певні рівні інтервали часу, тобто період  $T$  розділений у цьому випадку на  $2n$  рівних частин). Тоді абсциси точок розбивки будуть рівні  $t_k = T/2n$ , а ординати в цих точках  $s_k = f(t_k)$ , де  $k=0, 1, 2, \dots, 2n$ .

Тут  $s_k$  значення амплітуди сигналу  $s(t)$  у  $k$ -тий момент часу.

Якщо для безперервного сигналу дійсний наступний вираз:  $s(t) = S_m \sin \omega t$ , то після дискретизації в часі цей сигнал описується рівнянням  $s(n\Delta t) = S_m \sin n\omega\Delta t$ .

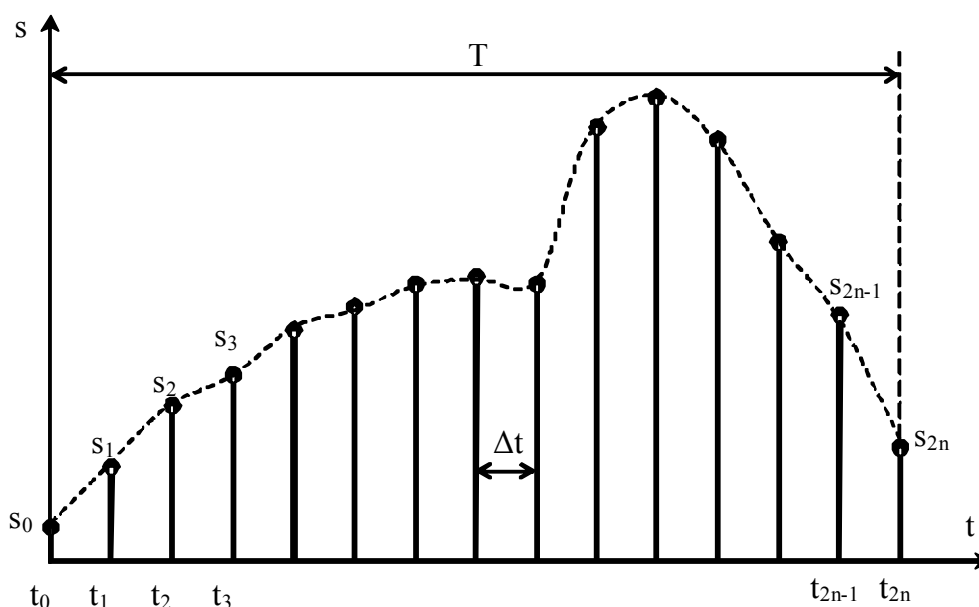


Рис. 1.5. Дискретизація (у часі)

У результаті дискретизації вихідна функція  $s(t)$  замінюється сукупність окремих значень  $s_k$ , за якими можна відновити вихідну функцію  $s(t)$  з деякою похибкою. Функцію, отриману в результаті такого відновлення (інтерполяції), називають відтворюючою. Для представлення відтворюючої функції найбільше часто використовуються степеневі алгебраїчні поліноми вигляду  $V(t) = \sum_{i=0}^n a_i t^i$ , де  $n$  - ступінь полінома,  $a_i$  - дійсні коефіцієнти.

При дискретизації сигналів доводиться вирішувати питання про те, як часто варто робити відліки функції, тобто який повинний бути крок дискретизації.

Відповідно до теореми В.А.Котельникова, якщо функція  $s(t)$  не містить частот

вище деякої  $F_m$ , то вона цілком визначається своїми миттєвими значеннями в моменти часу, що відстоять один від одного на величину  $1/(2F_m)$ , тобто

$$s(t) = \sum_{k=-\infty}^{\infty} s(t - k \cdot \Delta t) \cdot \frac{\sin[2\pi \cdot F_m \cdot (t - k \cdot \Delta t)]}{2\pi \cdot F_m \cdot (t - k \cdot \Delta t)},$$

де  $k$  - порядковий номер відліку функції;  $\Delta t = 1/(2F_m)$  - крок дискретизації за часом,  $s_k = s(t_k)$  - миттєві значення сигналу  $s(t)$  у  $k$ -ій відліковій точці  $t_k = k\pi/\omega_m = k/(2F_m) = k\Delta t$ .

З цієї теореми випливає, що для однозначного представлення функції з обмеженим спектром на інтервалі часу  $T$  досить мати деякі  $n$  значень цієї функції, де  $n = T/\Delta t = 2F_m T$ .

При виконанні цієї рівності (умови) безперервна і дискретна функції оборотні між собою, тобто тотожні. Таким чином, довільний сигнал, спектр якого не містить частот вище  $F_m$  може бути представлений у вигляді послідовності імпульсів, амплітуда яких дорівнює значенню вихідного сигналу в дискретні моменти часу  $k\Delta t$ , а інтервали між ними  $\Delta t = 1/(2F_m)$ .

З наведеного вище формулювання теореми Котельникова однозначно випливає, що для вибору оптимального кроку дискретизації необхідно попередньо провести кількісні оцінки всіх значущих гармонік спектрального розкладання початкового безперервного сигналу, для знаходження величини  $F_m$ , тобто,  $\omega_m$ .

Розглянемо тепер процедуру квантування сигналу за параметром.

У діапазоні безперервних значень функції  $f(t)$  вибирається кінцеве число дискретних значень функції, розподілених, наприклад, рівномірно по всьому діапазону. У довільний момент часу значення функції  $f(t)$  замінюється найближчим дискретним за параметрієм значенням. Тим самим функція здобуває східчастий вигляд (рис. 1.6).

Кроком квантування за параметром називається різниця між сусідніми дискретними значеннями функції. Для рівномірного квантування крок квантування постійний і дорівнює  $\Delta s = (s_{max} - s_{min})/k$ , де  $k$  - число кроків квантування.



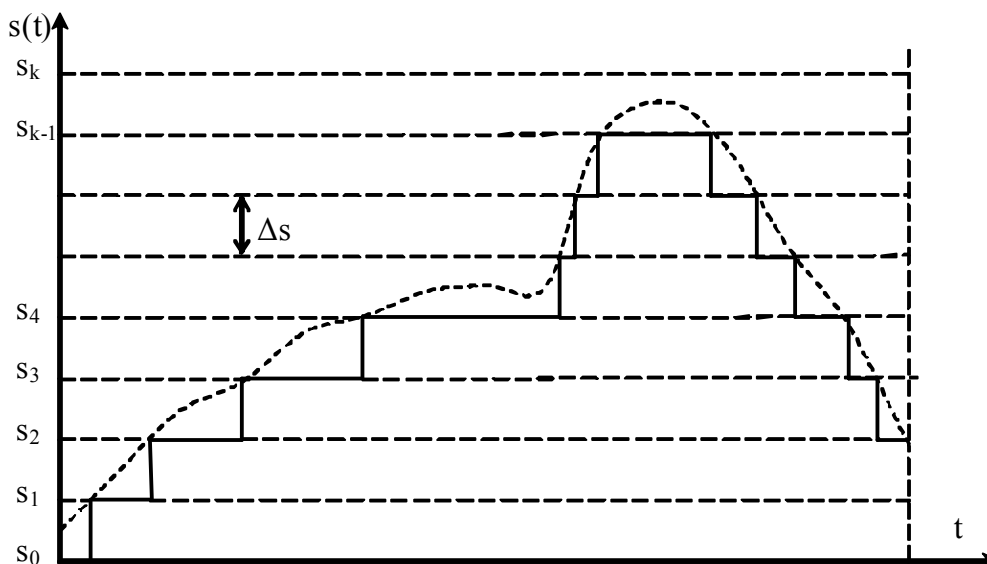


Рис. 1.6. Дискретизація за параметром

Абсолютна похибка квантування за рівнем в кращому випадку може змінюватися від  $0$  до  $\Delta s/2$  (у вище наведеному прикладі (рис. 1.6) вона змінюється від  $0$  до  $\Delta s$ ). Тоді середньоквадратична похибка квантування при рівномірному розподілі абсолютної похибки від  $0$  до  $0,5\Delta s$  дорівнює  $\sigma_s = \Delta s / 2$ .

Коли визначеному значенню квантованого параметра сигналу ставиться у відповідність деяке число (код) з кінцевої множини обраної системи числення, то виконується процедура кодування сигналу. Кодовані таким способом сигнали прийнято називати цифровими. Однак треба мати на увазі, що кодуванням у широкому значенні слова називається будь-яке перетворення повідомлення у певний сигнал шляхом установлення між ними однозначної відповідності.

Цифровий сигнал описується квантованою гратчастою функцією (квантованою послідовністю, квантованим часовим рядом)  $X(n \cdot \Delta t)$ , що приймає ряд дискретних значень рівнів квантування  $s_0, s_1, s_2, s_3, \dots, s_k$ , у тий час як незалежна змінна  $n = 0, 1, 2, \dots$  (рис. 1.7). Кожному з рівнів квантування ставиться у відповідність деяке число і тим самим сигнал кодується. Тому передача або оброблення відліку цифрового кодованого сигналу зводиться до операцій над безрозмірними числами (кодами). Представлення сигналу в цифровій формі практично завжди дає істотну перевагу при передачі, збереженні й обробці інформації.

Отже в даному випадку інформація кодується кінцевим набором символів (цифр, літер), що вибираються з деякого кінцевого алфавіту. Будь-яку кінцеву послідовність символів називають словом у даному алфавіті. Метод зображення будь-яких чисел з

допомогою обмеженої кількості цифр називається системою числення.

Кодування аналогового сигналу звичайно здійснюється за допомогою різного типу аналого-цифрових перетворювачів (АЦП). Зворотна процедура реалізується цифро-аналоговими перетворювачами (ЦАП). Для перетворення сигналів, зокрема, використовуються також різні модулятори і демодулятори, широко застосовувані, у так званих, модемах, за допомогою яких комп'ютери підключаються до різного типу обчислювальних мереж. Часто перетворюють і фізичну природу сигналу. Зокрема, перетворення оптичного сигналу в електричний і навпаки, виконують за допомогою різних оптоелектронних пристроїв.

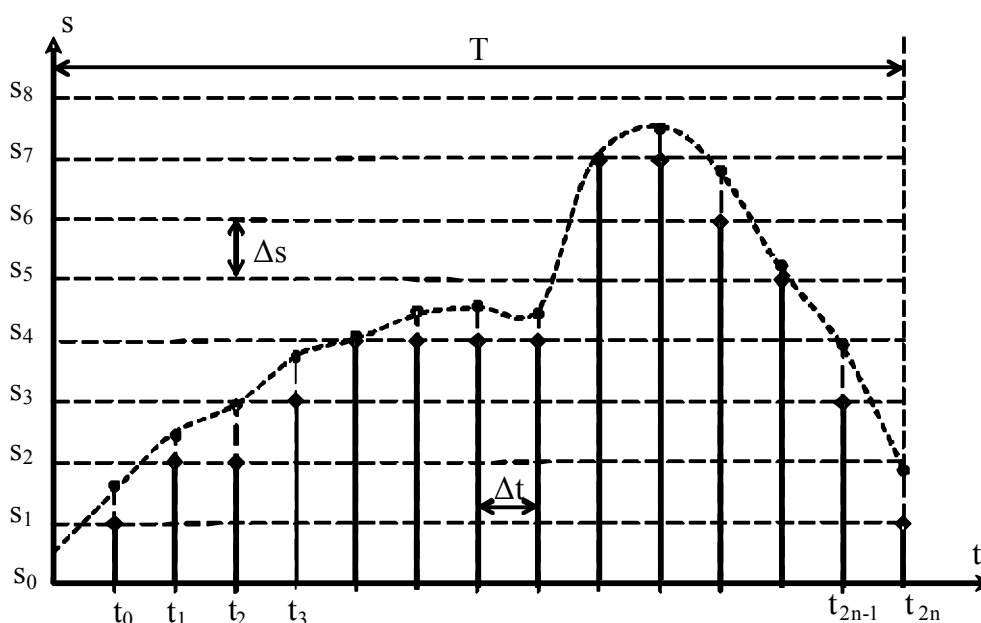


Рис. 1.7. Дискретизація за параметром та часом

## 1.2 Обмін інформацією між різними інформаційними пристроями

Обмін інформацією між різними пристроями, які так чи інакше реалізують інформаційні процеси, тобто приймання і передача інформації цими пристроями, забезпечується так званими інтерфейсами.

Сукупність усіх, в основному уніфікованих, технічних (апаратних) і програмних засобів, що забезпечують інформаційну взаємодію між інформаційними пристроями (ІП), називається інтерфейсом. Інакше кажучи, інтерфейс - це сукупність апаратних і програмних засобів для організації інформаційного зв'язку між інформаційними пристроями.

Інформація передається інтерфейсом за певним "протоколом", тобто, з

дотриманням певних правил, характерних для даного інтерфейсу.

Розрізняють ведучий ІІ (ініціатор) і ведений (виконавець). Ведучий ініціює обмін інформацією (даними), ведений передає або приймає інформацію під керуванням ведучого. До основних інтерфейсних процедур, зокрема, відносяться: арбітраж - запит і захоплення ведучим каналу зв'язку (магістралі) інтерфейсу; адресація потрібного веденого; обмін даними, виконуваний за протоколом конкретного інтерфейсу.

Стандартний інтерфейс - це сукупність уніфікованих апаратних, програмних і конструктивних засобів, необхідних для реалізації взаємодії різних ІІ при умовах, запропонованих стандартом і спрямованих на забезпечення інформаційної, електричної і конструктивної сумісності цих пристроїв.

Усі перераховані засоби стандартного інтерфейсу (СІ) повинні забезпечити суворе виконання протоколу специфічного для даного СІ.

На практиці використовують різні стандартні інтерфейси, наприклад: ISA, PCI, VME, IDE, SCSI, Ethernet, RS-232, USB і т.д.

### **1.3 Апаратні засоби збереження й обробки інформації**

Основним апаратним засобом будь-якої обробки інформації є комп'ютер, що призначений для обробки кодованої інформації, приймання, передачі і збереження її під керуванням відповідних програм.

Передача і приймання інформації в комп'ютерах здійснюється через відповідні порти, що з'єднують комп'ютер за допомогою певних інтерфейсів з його периферією (дисплей, клавіатура, принтер, сканер і т.д.) і з необхідною апаратурою користувача. Інформація зберігається в пам'яті комп'ютера.

Пам'ять комп'ютера - це функціональна його частина, призначена для запам'ятовування і (або) видачі різної інформації, проміжних і остаточних результатів обробки інформації.

У пам'яті, зокрема, знаходяться також програми рішення певних задач обробки інформації. Вся інформація в пам'яті комп'ютера представляється символами внутрішнього алфавіту.

Основними параметрами, що характеризують пам'ять, є ємність пам'яті і час доступу до неї.

Для побудови запам'ятовуючих пристроїв як фізичні елементи використовують електронні схеми, феритові магнітні матеріали, магнітні стрічки і диски, оптичні диски і т.д. Оперативний запам'ятовуючий пристрій (ОЗП) комп'ютера звичайно формується з електронних схем. У ОЗП кожне інформаційне слово розміщується в так званій комірці, якій присвоюється певна адреса. При відключенні комп'ютера інформація, записана в ОЗП, пропадає. При вмиканні комп'ютера відбувається автоматичне перевантаження операційної системи з відповідного магнітного диска в ОЗП. У пам'яті на магнітній стрічці або на дисках інформація групується у файли, що мають певні імена. Інформація, записана в пам'яті такого типу, зберігається при відключенні комп'ютера.

Обробку інформації виконує процесор, або мікропроцесор (МП) комп'ютера, що керує також всією апаратурою комп'ютера в цілому, включаючи зовнішню апаратуру, підключену до нього. До складу МП звичайно входять наступні основні блоки (вузли):

- арифметично-логічний пристрій (АЛП);
- ряд регістрів, кожний з яких призначений для збереження одного інформаційного слова;
- пам'ять мікропрограм (ПМП), у якій розміщуються мікропрограми, що керують функціями АЛП і регістрами МП.

Арифметично-логічний пристрій (АЛП) - функціональна частина МП, що виконує логічні й арифметичні дії, необхідні для обробки інформації, що зберігається в пам'яті комп'ютера.

АЛП може виконувати такі елементарні дії, як порівняння, додавання, віднімання, множення, ділення, зсуви вправо або вліво, додатній або від'ємний приріст, інверсію і цілий набір елементарних логічних операцій.

Функції АЛП визначають архітектуру МП у цілому.

Регістри МП беруть участь у реалізації усіх функцій МП, зокрема функцій його АЛП. Майже всі МП мають шість основних регістрів: стану, буферні, команд, адреси пам'яті або порту комп'ютера, лічильник команд і акумулятор. Акумулятор призначений для збереження слова даних, посланого з АЛП або пам'яті комп'ютера. Він є головним регістром МП при різних операцій з даними. Більшість арифметичних і логічних операцій здійснюються шляхом використання АЛП і акумулятора.

Структура мікропрограм визначає склад базових команд асемблера даного МП, кожній з яких транслятор ставить у відповідність певний код (звичайно в двійковій системі числення). У свою чергу кожній команді, операторів мови програмування високого рівня (Паскаль, Сі, Фортран і т.д.) відповідний компілятор або інтерпретатор ставить у відповідність певну групу команд асемблера.

Коли МП витягає з наступної адреси комірки ОЗП комп'ютера код команди асемблера, то за цим кодом активізується відповідна мікропрограма, яка, у свою чергу, генерує послідовність кодів команд, які керують мікропроцесором у цілому, зокрема, арифметично-логічним пристроєм і регістрами МП.

У такий спосіб спрощено виглядає ієрархічна структура керування роботою комп'ютера. Отже, оброблення інформації програмою будь-якої складності і будь-якого характеру зводиться в підсумку до виконання відповідної послідовності елементарних команд, виконуваних АЛП мікропроцесора. Програмне введення і виведення інформації у всіх випадках здійснюється "крізь" МП, через його регістри.

#### 1.4 Загальні поняття про цифровий автомат і алгоритм

Необхідність формального опису комп'ютера і його окремих частин у процесі його проектування вимагає застосування спеціального математичного апарату, який необхідний при будь-яких розробках різних методів обробки інформації, при синтезі й аналізі будь-яких інформаційних процесів. Для цього вводиться поняття абстрактного цифрового автомата (ЦА).

Пристрої, призначені для обробки, перетворення цифрової інформації називаються цифровими автоматами. Найбільш розповсюдженим у даний час видом цифрових автоматів є електронні цифрові обчислювальні машини з програмним керуванням, тобто комп'ютери.

Введемо інше визначення поняття цифрового автомата.

Цифровий автомат - це пристрій, що характеризується набором деяких внутрішніх станів  $A = \{a_0(t), a_1(t), a_2(t), a_3(t), \dots, a_n(t)\}$ , у які він попадає під впливом вхідних сигналів і команд програми рішення задачі.

Нехай є автомат з одним входом і одним виходом (рис. 1.8). Тоді математичною моделлю цифрового автомата є деякий абстрактний автомат, заданий у такий спосіб: у початковий момент часу  $T = T_0$  автомат знаходиться в стані  $a(T_0) = a_0$  і залишається в

ньому до моменту  $T = T_1$ , коли з'являється вхідний сигнал  $x(T_1)$ . Під впливом сигналу  $x(T_1) = x_1$  автомат переходить зі стану  $a_0$  у стан  $a(T_1) = a_1$ . При цьому виникає вихідний сигнал  $y(T_1) = y_1$ , визначений як функція виходів  $y(T_1) = \lambda[a(T_0), x(T_1)]$  або  $y_1 = \lambda[a_0, x_1]$ . Загалом можна сказати, що  $y_{t+1} = \lambda[a_t, x_{t+1}]$ , де  $t = 0, 1, 2, \dots$  - моменти дискретного часу.

У такий спосіб можна прийняти, що при подачі довільного сигналу  $x(T_{t+1}) = x_{t+1}$  автомат переходить зі стану  $a_t$  у стан  $a_{t+1}$ , що описується функцією переходів  $a_{t+1} = \delta[a_t, x_{t+1}]$ , і в результаті виробляє вихідний сигнал  $y_{t+1}$ .

Вихідні сигнали можуть вироблятися при кожному переході автомата зі стану  $a_t$  у стан  $a_{t+1}$  або тільки при певних сполученнях вхідного сигналу і станів автомата.

Поняття стану автомата використовується для опису систем, виходи яких залежать не тільки від вхідних сигналів у даний момент часу, але і від деякої передісторії, тобто і від сигналів, що надходили на входи системи раніше. Стан автомата відповідає деякій пам'яті про минуле, дозволяючи усунути час як явну змінну і виразити вихідні сигнали як функцію станів і вхідних сигналів.

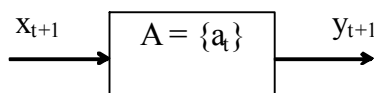


Рис. 1.8. Цифровий автомат

## 1.5 Опис абстрактного автомату

У такий спосіб абстрактний автомат можна описати за допомогою наступних параметрів:

$a_0$  - початковий стан автомата;

$A = \{a_0, a_1, a_2, \dots, a_m\}$  - множина внутрішніх станів;

$X = \{x_0, x_1, x_2, \dots, x_f\}$  - множина вхідних сигналів;

$Y = \{y_0, y_1, y_2, \dots, y_h\}$  - множина вихідних сигналів;

$\Delta = \{\delta_0, \delta_1, \delta_2, \dots, \delta_n\}$  - сукупність функцій переходу з одного стану в інше;

$\Lambda = \{\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_p\}$  - сукупність функцій виходів.

Отже, абстрактний автомат реалізує деякі відображення множини слів вхідного алфавіту  $X$  у множину слів вихідного алфавіту  $Y$ .

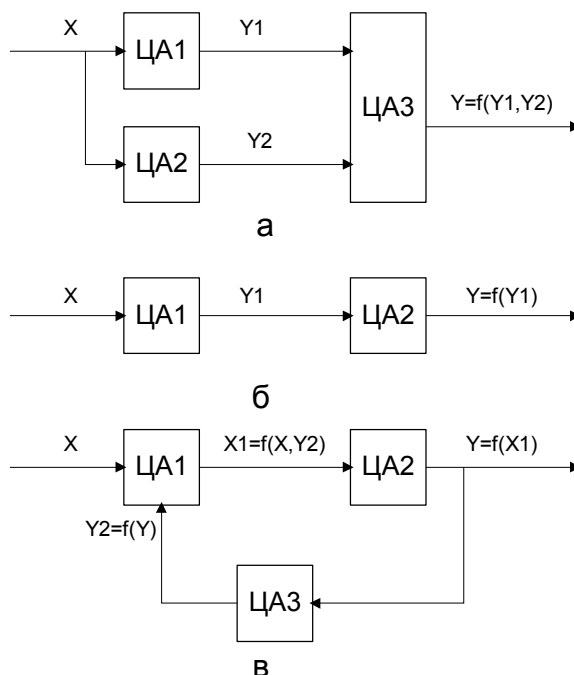
Цифрові автомати можуть бути об'єднані між собою для формування більш складного пристрою обробки цифрової інформації. Існує кілька способів з'єднання автоматів. Наприклад, паралельне з'єднання автоматів ЦА1 і ЦА2 (рис. 1.9, а), послідовне з'єднання (рис. 1.9, б) і з'єднання із зворотний зв'язок (рис. 1.9, в).

У першому випадку пристрій ЦА3 об'єднує виходи двох автоматів і тим самим утворюється новий автомат, у якого  $Y = f\{Y_1, Y_2\}$ .

В другому випадку вихідний алфавіт першого автомата є входним для другого. У третьому випадку є деякий перетворювач ЦА1, який є автоматом без пам'яті, що реалізує відображення  $X_1 = f\{X, Y_2\}$ . Очевидно, що у всіх цих випадках у підсумку формується якийсь цифровий автомат з новими характеристиками.

Сукупність правил переходу автомата з одного стану в інше в залежності від входної інформації і внутрішніх станів автомата називається алгоритмом перетворення (переробки) інформації. Взагалі алгоритмом називається кінцева сукупність точно сформульованих правил рішення якоїсь задачі.

Можна навести ще одне визначення поняття алгоритму. Алгоритм - це суворо формальний опис кінцевої послідовності деяких "елементарних" дій або процедур, яку треба виконати над вихідними даними і над проміжними результатами, що виникли в ході виконання цих операцій, для того щоб прийти до інформації, яка є результатом обробки вихідних даних. Докладний опис принципів розробки алгоритмів і їхніх блок-схем наведено в розділі 1.4.



*Рис. 1.9. Об'єднання автоматів*

Перераховані в цьому розділі поняття відносяться до абстрактної теорії цифрових автоматів, у якій розглядаються автомати, що мають один вхід і один вихід. Тому застосувати все це до комп'ютера можна тільки в самому загальному вигляді, обмежуючи коло розглянутих пристроїв пристроями, що входять до складу комп'ютера, або ж у випадку коли розглядається взагалі деякий вузол комп'ютера, призначений для досить елементарної обробки цифрової інформації. Справді, у залежності від команд, подаваних пристроєм керування процесора, арифметично-логічний пристрій здійснює відповідні дії (зміну внутрішніх станів) з видачею необхідних результатів. Однак зміна внутрішніх станів усього комп'ютера у цілому носить настільки складний характер, що цей процес неможливо відобразити в аналітичному вигляді. Тому поняття цифрового автомата доцільно використовувати стосовно до алгоритмів, що реалізують якусь програму або послідовність операцій. При цьому кожна операція представляється як елементарна дія, здійснювана в процесі переробки інформації.

І все-таки надалі, якщо це не буде спеціально обговорено, завжди під поняттям "цифровий автомат" будемо мати на увазі комп'ютер без деталізації, про який з його вузлів мова йде, або ж узагалі прямо будемо використовувати термін "комп'ютер".



## 2 ПРЕДСТАВЛЕННЯ ЧИСЛОВОЇ ІНФОРМАЦІЇ В ЦИФРОВОМУ АВТОМАТІ

### 2.1 Системи числення і поняття коду

Як уже відзначалося, у процесі переробки інформації комп'ютери оперують числами, що представляються в деякій системі числення.

Система числення - це сукупність прийомів і правил для запису чисел цифровими знаками. Запис числа в деякій системі числення часто називають кодом числа.

Елементи (символи) алфавіту, що використовуються для запису чисел у деякій системі числення, прийнято називати цифрами. Кожній цифрі даного числа однозначно зіставляється її кількісний (числовий) еквівалент.

Розрізняють позиційні і непозиційні системи числення.

Непозиційна система числення - це система, для якої значення символу, тобто цифри, не залежить від його положення в числі. До таких систем відноситься, зокрема, римська система (правда, з деякими застереженнями). Тут, наприклад, символ V завжди означає п'ять, поза залежністю від місця його появи в записі числа. Є й інші сучасні непозиційні системи.

Позиційна система числення - це система, у якій значення кожної цифри залежить від її числового еквівалента і від її місця (позиції) у числі, тобто, той самий символ (цифра) може приймати різні значення.

Найбільш відомою позиційною системою числення є десяткова система числення. Наприклад, у десятковому числі 555 перша цифра праворуч означає 5 одиниць, сусідня з нею - 5 десятків, а ліва - 5 сотень.

У зв'язку з тим, що в цифрових автоматах в основному використовуються позиційні системи числення, те ми надалі будемо розглядати тільки такі системи.

Будь-яка позиційна система числення характеризується основою.

Основа або базис  $q$  природної позиційної системи числення це кількість знаків або символів, використовуваних для зображення числа в даній системі.

Тому, можлива незліченна множина позиційних систем, оскільки за основу можна прийняти будь-як число, утворивши нову систему числення.

Коли ми представляємо, тобто записуємо деяке число в позиційній системі числення, ми розміщаємо відповідні цифри числа в окремих потрібних позиціях, які

прийнято називати розрядами числа в даній позиційній системі числення. Кількість розрядів у записі числа називається розрядністю числа і збігається з його довжиною.

Для позиційної системи числення справедливо:

$$\text{Формула 2.1: } A_{(q)} = a_n q^n + a_{n-1} q^{n-1} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m} \text{ або } A_{(q)} = \sum_{i=-m}^{i=n} a_i q^i,$$

де  $A_{(q)}$  це довільне число, записане в системі числення з основою  $q$ ;

$a_i$  - коефіцієнти ряду, тобто цифри системи числення;

$n+1, m$  - кількість цілих і дробових розрядів відповідно.

Наприклад,

$$1961,32_{10} = 1 \cdot 10^3 + 9 \cdot 10^2 + 6 \cdot 10^1 + 1 \cdot 10^0 + 3 \cdot 10^{-1} + 2 \cdot 10^{-2},$$

$$124,537_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 4 \cdot 8^0 + 5 \cdot 8^{-1} + 3 \cdot 10^{-2} + 7 \cdot 8^{-2},$$

$$1001,1101_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4}.$$

Індекс, приписуваний до числа, вказує систему числення, у якій представлене дане число.

Основа системи числення показує скільки різних значень у межах  $i$ -того розряду може приймати кожна цифра  $a_i$  числа  $A$ . Номера розрядів у позиційній системі числення відраховуються в цілій частині вліво від коми, а в дробовій - вправо від коми. Причому, нумерація розрядів починається з 0 (наймолодший цілий розряд, перший розряд зліва від коми). Величина основи позиційної системи числення визначає її назву: для десяткової системи це буде 10, для вісімкової - 8, для двійкової - 2 і т.д. Як уже відзначалося, звичайно замість назви системи числення використовують термін "код числа". Наприклад, під поняттям двійковий код мається на увазі число, представлене в двійковій системі числення, під поняттям десятковий код - у десятковій системі числення і т.д.

Для запису числа в десятковій системі використовується 10 різних цифр від 0 до 9:  $a_{10} = (0,1,2,3,4,5,6,7,8,9)$ ;

у шістнадцяткової - 16:  $a_{16} = (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)$ ,

де  $A = 10_{10}$ ,  $B = 11_{10}$ ,  $C = 12_{10}$ ,  $D = 13_{10}$ ,  $E = 14_{10}$ ,  $F = 15_{10}$ ;

у вісімкової - 8:  $a_8 = (0,1,2,3,4,5,6,7)$ ;

у двійковій - 2:  $a_2 = (0,1)$  (таблиця 2.).

**Таблиця 2.1. Цифри і числа (в дужках) у поширених системах числення**

Система числення з основою 10	Система числення з основою 2	Система числення з основою 8	Система числення з основою 16
0	0	0	0
1	1	1	1
2	(10)	2	2
3	(11)	3	3
4	(100)	4	4
5	(101)	5	5
6	(110)	6	6
7	(111)	7	7
8	(1000)	(10)	8
9	(1001)	(11)	9
(10)	(1010)	(12)	A
(11)	(1011)	(13)	B
(12)	(1100)	(14)	C
(13)	(1101)	(15)	D
(14)	(1110)	(16)	E
(15)	(1111)	(17)	F
(16)	(10000)	(20)	10

Вага кожного розряду числа в двійковій системі числення ліворуч від коми представляється двійкою у відповідному додатньому степені, а праворуч від коми - двійкою у від'ємному степені (таблиця 2.).

Таблиця 2.2. Степені 2

n	2 <sup>n</sup>			
	10-ва система числення	2-ва система числення	16-ва система числення	8-ва система числення
10	1024	1000000000	400	2000
9	512	100000000	200	1000
8	256	10000000	100	400
7	128	1000000	80	200
6	64	100000	40	100
5	32	10000	20	40
4	16	1000	10	20
3	8	100	8	10
2	4	10	4	4
1	2	1	2	2
0	1	1	1	1
-1	0,5	0,1	0,8	0,4
-2	0,25	0,01	0,4	0,2
-3	0,125	0,001	0,2	0,1
-4	0,0625	0,0001	0,1	0,04
-5	0,03125	0,00001	0,08	0,02
-6	0,015625	0,000001	0,04	0,01
-7	0,0078125	0,0000001	0,02	0,004
-8	0,00390625	0,00000001	0,01	0,002
-9	0,00195313	0,000000001	0,008	0,001
-10	0,00097656	0,0000000001	0,004	0,0004

Приклади представлення чисел у різних системах числення:

$$10_{10} = 1010_2 = 12_8 = A_{16},$$

$$16_{10} = 10000_2 = 20_8 = 10_{16},$$

$$255_{10} = 1111111_2 = 377_8 = FF_{16}.$$

Для обробки інформації в комп'ютері звичайно використовується двійкова система числення. Це пояснюється, зокрема, тим, що для розміщення чисел (операндів) у комп'ютерах використовуються регістри і комірки пам'яті, що складаються з тригерів або елементів із тригерною характеристикою, які, як відомо, мають два стійких стани. Одному з цих станів ставиться у відповідність 1, а іншому - 0. Кожний із тригерів призначається для розміщення найменшої інформаційної одиниці в двійковій системі числення - двійкового розряду, що називається бітом.

Вісім біт є байт.

Кількість тригерів, тобто двійкових розрядів, у регістрі або комірці пам'яті визначає довжину слова, характерну для даного комп'ютера, а сукупність цих двійкових розрядів називається розрядною сіткою. Номер розряду такої сітки, відведеної для зображення цілого числа в двійковій системі числення, збігається з відповідним показником степеня двійки.

Таким чином, довжина числа - це кількість позицій (або розрядів) у записі числа. Для різних систем числення характерна різна довжина розрядної сітки, необхідна для запису того самого числа. Наприклад,  $96_{10} = 1100000_2 = 140_8 = 60_{16}$ , тобто, як правило, чим менша основа системи числення, тим більша довжина числа.

У будь-яких цифрових автоматах довжина розрядної сітки обраної системи числення фіксована, що принципово обмежує точність і діапазон представлення чисел.

Нехай довжина розрядної сітки дорівнює будь-якому додатньому числу, наприклад  $n$ . Тоді

$$(A_q)_{max} = q^n - 1; (A_q)_{min} = -(q^n - 1).$$

Наприклад, у двійковій системі числення, якщо  $n = 3$ , то  $(A_2)_{max} = 111_2 = 7_{10}$ , тобто максимальне додатнє число, що може бути записано дорівнює 7, а мінімальне дорівнює -7, представлення якого в двійковій системі буде обговорюватися пізніше.

Діапазон представлення чисел у заданій системі числення - це інтервал числової осі, який знаходиться між максимальним і мінімальним числами, значення яких, як ми бачимо, залежить від довжини розрядної сітки.

Вага розряду  $P_i$  числа в позиційній системі числення - це відношення  $P_i = q^i / q^0 = q^i$ , де  $i$  - номер розряду, а  $q^0$  це вага першого розряду цілої частини числа, його номер дорівнює 0, а значення ваги дорівнює 1.

Якщо в даному розряді нагромадилося значення одиниць рівне або більше  $q$ , то повинна відбуватися передача одиниці в сусідній старший розряд. При додаванні чисел, представлених у будь-якій позиційній системі числення, такі передачі інформації називають переносами, а при відніманні - позиками. Передача переносів і позик відбувається послідовно від розряду до розряду. Ці процедури добре відомі усім із правил арифметичних дій з десятковими числами.

Крім звичайної двійкової системи числення були запропоновані двійкові

системи числення, у яких для зображення чисел використовуються символи 1, -1 або 0, -1. Існують ще, так називані, надлишкові двійкові системи числення, наприклад, із символами 0, 1, -1. Але на практиці в переважній більшості випадків використовується звичайна двійкова система числення.

При обміні даними між комп'ютером і зовнішніми пристроями виникає необхідність в обміні знаковими і літерними символами. Цим символам у комп'ютері також ставиться у відповідність деякий код у двійковій системі числення. Для представлення цифр і літер у двійковій системі в даний час найбільш розповсюджений код ASCII (American Standard Code for Information Interchange – американський стандартний код для обміну інформацією, таблиця 2.). Для представлення будь-якого символу в цьому коді приділяється 7 двійкових розрядів. Приклади коду ASCII наведені нижче (таблиця 2.). У текстах, де додатково до латинських літер використовуються також літери кирилиці (українські та російські) використовуються інші 8-розрядні коди (8-розрядні кодові таблиці):

кодова таблиця фірми Microsoft CP-1251 (кодування Windows, таблиця 2.);

кодова таблиця, застосовувана в ОС Unix KOI 7-U (таблиця 2.);

альтернативна кодова таблиця CP-866 (таблиця 2.).

Також можуть використовуватися і інші кодові таблиці.

**Таблиця 2.3. Код ASCII (приклади)**

Символи	Десятковий код	Двійковий код	Вісімковий код	Шістнадцятковий код
0	48	0110000	60	30
1	49	0110001	61	31
2	50	0110010	62	32
A	65	1000001	101	41
B	66	1000010	102	42
F	70	1000110	106	46
:	58	0111010	72	3A
(	40	0101000	50	28

Таблиця 2.4. Код ASCII (Шістнадцятковий код)

0	nul	1	soh	2	stx	3	etx	4	eot	5	enq	6	ask	7	bel
8	bs	9	ht	0a	nl	0b	vt	0c	np	0d	cr	0e	so	0f	si
10	dle	11	dc1	12	dc2	13	dc3	14	dc4	15	nak	16	syn	17	etb
18	can	19	em	1a	sub	1b	esc	1c	fs	1d	gs	1e	rs	1f	us
20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(	29	)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5a	Z	5b	[	5c	\	5d	]	5e	^	5f	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del

Таблиця 2.5. Кодова таблиця фірми Microsoft CP-1251 (кодування Windows)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				©	Є	§	€	.		°						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2		!	"	#	\$	%	&		(	)	^	+				/
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	Ђ	Ѓ	„	ѓ	„	…	†	‡		‰	Љ	«	Њ	ќ	ћ	џ
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	ђ	'	'	"	"		—	—		™	љ	»	њ	ќ	ћ	џ
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A		У	Ў	Ј	Ѡ	Ѓ	Ѕ	Є	©	€	«		-	®	Ї	
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	°	±	І	і	Ѓ	µ		·	ё	№	є	»	ј	Ѕ	ѕ	ї
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Таблиця 2.6. Кодова таблиця КОІ 7-У

—		Г	Г	Г	Г	Г	Г	Г	Г	Г	■	■	■	■	
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
▤	▥	▦	Г	■	●	√	≈	≤	≥	nbsp	Г	○	2	•	÷
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
=		Г	ё	ё	Г	і	й	Г	Г	Г	Г	Г	Г	Г	Г
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Г	Г	Г	ё	ё	Г	і	й	Г	Г	Г	Г	Г	Г	Г	©
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Таблиця 2.7. Альтернативна кодова таблиця CP-866

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		⊙	⊖	♥	♠	♣	♠	●		○							
	1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	▶	◀					—		↑	↓	→	←		↔	▲	▼	
	2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/		
	3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
	4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
	5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	
	6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
	7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	␣	
	8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
8	A	B	B	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	
	9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
9	P	C	T	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
	A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
A	a	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	
	B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
B	▤	▥	▦		Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	
	C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
C	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	
	D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
D	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	
	E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	
	F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
F	ё	ё	≥	≤			÷	*	°	·	·	√	π	z	■	␣	

## 2.2 Вибір системи числення

При виборі системи числення для ЕОМ необхідно враховувати, що по-перше, основа системи числення визначає кількість стійких станів, що повинні мати



функціональний елемент, обраний для зображення розрядів числа; по-друге - довжина числа істотно залежить від основи системи числення; у третій - система числення повинна забезпечити прості алгоритми виконання арифметичних і логічних операцій.

Якщо є  $n$  розрядів для зображення числа в системі числення з основою  $q$ , тоді максимальне число  $M$ , яке можна зобразити в межах даної розрядної сітки, буде дорівнювати  $M = q^n - 1$ . Звідки випливає  $n = \log_q(M + 1) = \ln(M + 1) / \ln q$ .

Для оцінки економічності системи числення з погляду витрат устаткування цифрового автомата вводиться відповідний показник  $N = qn$ .

З наведених рівностей випливає, що  $N = q \cdot \ln(M + 1) / \ln q$ . Використовуючи отриману залежність, можна знайти основу системи числення, при якій потрібен мінімум устаткування. Визначивши  $dN/dq$  і дорівнявши її до нуля, одержимо екстремум при  $q = e$ . Але  $e$  не ціле число, тому потрібно використовувати системи з  $q = 2$  або  $q = 3$ . Трійкова система більш ефективна ніж двійкова тому, що при великому  $M$   $M = 2^{n_2} - 1$ ,  $n_3 = \log_3(M + 1) = \ln(M + 1) / \ln 3 \approx \ln(2^{n_2} - 1 + 1) / \ln 3 = \ln(2^{n_2}) / \ln 3 = n_2 \ln 2 / \ln 3$ ,  $N_3 / N_2 = 3n_3 / 2n_2 = 3 \ln 2 / 2 \ln 3 = 0,95$ . Подібне порівняння десяткової і двійкової систем числення показує, що десяткова в 1,5 разів менш економічна двійкової.

Найбільш зручні умови реалізації двійкових цифр, тому що фізичних процесів, що мають два стійких стани, набагато більше, ніж процесів з числом чітко помітних станів більше двох. До того ж у процесах із двома стійкими станами розходження між цими станами носить якісний, а не кількісний характер, що забезпечує надійну реалізацію двійкових цифр.

Таким чином, простота арифметичних і логічних дій, мінімум використовуваного устаткування для представлення чисел і найбільш зручні умови реалізації тільки двох стійких станів визначили застосування двійкових систем числення практично у всіх існуючих і проєктованих цифрових обчислювальних машинах.

### 2.3 Формальні правила двійкової арифметики

Перед тим, як розглянути формальні правила двійкової арифметики наголосимо на загальному принципі додавання і віднімання чисел, представлених у будь-якій позиційній системі числення.

У загальному випадку процедури додавання і віднімання двох чисел  $A \pm B \pm C_i = C_o, S$  у будь-якій позиційній системі числення починаються з молодших розрядів ( $A$  – перше число,  $B$  – друге число,  $C_i$  – вхідний перенос або позика,  $C_o$  – вихідний перенос або позика,  $S$  – сума).

Коди переносу до старшого розряду  $c_o$  і суми кожного  $i$ -того розряду  $s_i$  утворюється у результаті додавання  $a_i + b_i + c_i$ , де  $c_i$  - перенос з молодшого ( $i - 1$ )-розряду в  $i$ -тий,  $c_i$  рівний 1, якщо в молодшому розряді код суми вийшов більшим або рівним основі системи числення, і  $c_i$  рівний 0 в інших випадках.

Коди позики від старшого розряду  $c_o$  і різниці кожного  $i$ -того розряду утворюється в результаті віднімання  $a_i - b_i - c_i$ , де  $c_i$  відповідає позиці, якщо вона була, до молодших розрядів величини, рівна основі системи числення.

Отже, правила і методи додавання і віднімання в будь-якій позиційній системі числення в принципі залишаються такими ж, як у десятковій системі.

Тепер розглянемо правила арифметики з числами, представленими в двійковому коді.

Додавання двох чисел виконується по-розрядно, починаючи з молодшого розряду. У кожному розряді виконується додавання двох цифр доданків і одиниці переносу із сусіднього молодшого розряду (таблиця 2.). Сумування двійкових чисел у комп'ютерах таким способом здійснюється за допомогою повних однорозрядних двійкових суматорів.

Додавання без вхідного переносу (таблиця 2.) здійснюється за допомогою двійкових напівсуматорів.

Наприклад:

$$0101_2 = 5_{10}$$

$$+0011_2 = 3_{10}$$

---


$$1000_2 = 8_{10}$$

**Таблиця 2.8. Повний однорозрядний двійковий суматор**

Входи			Виходи	
$a$	$b$	$c_i$	$c_o$	$s$
0	0	0	0	0

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

**Таблиця 2.9. Двійковий напівсуматор**

Входи		Виходи	
$a$	$b$	$c_o$	$s$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Віднімання також провадиться по-розрядно, починаючи з молодшого розряду (таблиця 2.). Як видно (таблиця 2.), при відніманні в даному розряді з нуля одиниці необхідно позичити одиницю із сусіднього старшого розряду, що дорівнює двом одиницям даного розряду. Це ускладнює структуру пристроїв за допомогою яких виконується віднімання - двійкових віднімачів. Але як буде показано надалі, віднімання можна організувати також за допомогою процедури додавання, тобто за допомогою двійкових суматорів, якщо від'ємник представити в "доповняльному" або "оберненому" коді і тим самим виключити необхідність у двійкових віднімачах .

Наприклад:  $6 - 3 = 3$ .

$$\begin{array}{r}
 0110_2 = 6_{10} \\
 -0011_2 = 3_{10} \\
 \hline
 0011_2 = 3_{10}
 \end{array}$$

Умовні схемотехнічні позначення двійкових напівсуматора і суматора показує рис. 2.1.

**Таблиця 2.10. Двійковий віднімач**

Входи			Виходи	
$a$	$b$	$c_i$	$c_o$	$s$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	2	0

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

Множення двійкових чисел провадиться шляхом утворення проміжних добутоків і наступного їхнього сумування. Проміжні по-розрядні добутки формуються за правилами множення двійкових розрядів  $a$  і  $b$  (таблиця 2.).

Наприклад:  $5_{10} \times 3_{10} = 15_{10}$

$$\begin{array}{r}
 \phantom{+} \phantom{+} \phantom{+} 1 \ 0 \ 1 \\
 \phantom{+} \phantom{+} \phantom{+} \phantom{+} \phantom{+} x \ 0 \ 1 \ 1 \\
 \hline
 \phantom{+} \phantom{+} \phantom{+} + \ 1 \ 0 \ 1 \\
 \phantom{+} \phantom{+} + \ 1 \ 0 \ 1 \\
 + \ 0 \ 0 \ 0 \\
 \hline
 0 \ 1 \ 1 \ 1 \ 1
 \end{array}$$

**Таблиця 2.11. Множення двійкових кодів**

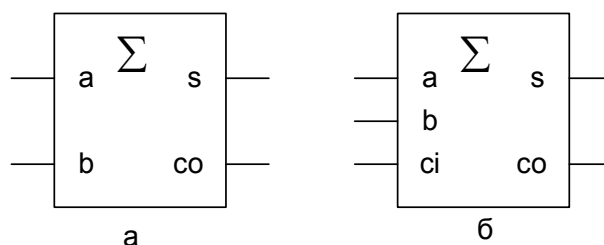
Входи		Вихід
$a$	$b$	$s = ab$
0	0	0
0	1	0
1	0	0
1	1	1

Ділення чисел у двійковій системі провадиться за правилами множення і віднімання.

Наприклад:

$$6_{10} : 3_{10} = 2_{10}; \quad 110 : 11 = 10.$$

Арифметичні дії з двійковими числами докладно будуть розглянуті далі.



*Рис. 2.1. Умовне позначення напівсуматора (а) і двійкового суматора (б).*

Якщо для представлення двійкових чисел  $A$ ,  $B$ ,  $S$  та їхніх знаків виділена  $n$ -розрядна сітка, то очевидно, що для організації процедури додавання необхідно  $n$  двійкових однорозрядних суматорів, які з'єднуються між собою за певною схемою, яка залежить від того, в якому коді представляються ці двійкові числа: прямому,

оберненому або доповняльному.

Крім арифметичних операцій у цифрових автоматах реалізуються також логічні операції, які докладно розглядаються в наступних главах.

Крім цих операцій у цифрових автоматах, комп'ютерах, виконується ще одна операція над двійковими числами - це зсув числа вздовж розрядної сітки ліворуч або праворуч. У випадку зсуву ліворуч фактично здійснюється множення двійкового числа на  $2\alpha$ , а при зсуві праворуч - ділення на  $2\alpha$ , де  $\alpha$  - кількість розрядів, на яке зсувається двійкове число. Наприклад:  $000011_2 = 3_{10}$  зсунемо ліворуч на 2 розряди, одержимо  $001100_2 = 12_{10}$ , тобто  $3_{10} \times 4_{10} = 12_{10}$ , а тепер  $001000_2 = 8_{10}$  зсунемо на 2 розряди праворуч, одержимо  $000010_2 = 2_{10}$ , тобто  $8_{10} : 4_{10} = 2_{10}$ .

У комп'ютерах часто використовується циклічний зсув, при виконанні якого розрядна сітка, відведена для операнда, представляється замкнутою в кільце. Тоді при зсуві ліворуч вміст старшого розряду потрапляє в молодший розряд операнда, а при зсуві праворуч - навпаки.

## 2.4 Переведення числа з однієї позиційної системи числення в іншу

Як уже відзначалося, будь-яка оброблення інформації в комп'ютері звичайно здійснюється в двійковій системі числення. У той же час, при обміні інформації між комп'ютером і користувачем для більшої наочності представлення даних використовуються десяткова, двійкові-десяткові, вісімкова або шістнадцяткова системи. Кожний розряд числа у вісімковому і шістнадцятковому коді еквівалентний трьом і чотирьом двійковим розрядам відповідно. Тому, представлення чисел у цих системах числення виходить більш компактним і наочним.

Для переведення вісімкового (шістнадцяткового) числа в двійкову форму досить замінити кожну цифру цього числа відповідної трирозрядним (чотирирозрядним) двійковим числом, при цьому відкидають непотрібні нулі в старших і молодших розрядах.

Наприклад:

$$305,4_8 = 011\ 000\ 101,100_2 = 11000101,1_2;$$

$$7B2, E_8 = 0111\ 1011\ 0010,1110_2 = 11110110010,111_2.$$

Для переходу від двійкової до вісімкової (шістнадцяткової) системи роблять

так: рухаючись від коми ліворуч і праворуч, розбивають двійкове число на групи по три (чотири) розряди, доповнюючи, при необхідності, нулями крайні ліву і праву групи. Потім групу з трьох (чотирьох) розрядів заміняють відповідною вісімковою (шістнадцятковою) цифрою.

Наприклад:

1) переведення  $1101111001,1101_2$  у вісімкове

$$001\ 101\ 111\ 001,110\ 100_2 = 1571,64_8;$$

2) переведення  $1111111011,100111_2$  у шістнадцяткове

$$0111\ 1111\ 1011,10011100_2 = 7FB,9C_{16}.$$

Двійково-десятковий код (*D*-код) орієнтований на найбільш зручну для людини десяткову систему числення. У ньому для запису чисел використовуються тільки двійкові цифри 0 і 1. Двійковий-десятковий код утвориться заміною кожного десяткового розряду в десятковому числі 4-х бітовим двійковим представленням цього розряду.

Наприклад,

$$0001\ 1001\ 1000\ 0100_2 = 1984_{10}.$$

## 2.5 Переведення чисел до десяткової системи числення з іншої однорідної позиційної системи числення з основою *k*, коли дії виконуються в десятковій системі

В однорідних позиційних системах числення при безпосередньому представленні цифр число записується у вигляді

$$X = \{x_s x_{s-1} \dots x_l x_0 x_{-1} \dots x_{-m}\}$$

Кількісний еквівалент, який виражається цим записом, визначається так:

$$\text{Формула 2.1 } X = k^s x_s + k^{s-1} x_{s-1} + \dots + k^l x_l + k^0 x_0 + k^{-1} x_{-1} + \dots + k^{-m} x_{-m},$$

де *k* - основа системи числення,

*s+1* - розрядність цілої частини числа,

*m* - розрядність дробової частини числа,

*x<sub>i</sub>* - цифри *i*-го розряду запису числа ( $x_i = 0, 1, \dots, k-1$ ),

$k^i$  - вага *i*-го розряду. У даному випадку вага *i*-го розряду в *k* разів більша за вагу (*i-1*)-го розряду. Такі системи числення називають системами з природним порядком

ваги,  $i$  до них належать двійкова, вісімкова, десяткова і шістнадцяткова системи числення.

Цифри цих систем та їхні еквіваленти містить таблиця 2..

Формула 2.1 подає один із методів переведення чисел із системи числення з основою  $k$  до десяткової системи. Для переведення необхідно записати число  $X$  як цю формулу, замінити цифри  $k$ -тої системи числення та основу  $k$  їхніми десятковими еквівалентами, а потім обчислити вираз за правилами десяткової арифметики.

Приклад 2.1. Перевести число  $X_2 = 1011,1001_2$  до десяткової системи числення.

$$\begin{aligned} 1011,1001_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = \\ &= 8 + 0 + 2 + 1 + 0,5 + 0 + 0 + 0,0625 = 11,5625; \end{aligned}$$

$$X_{10} = 11,5625.$$

Приклад 2.2. Перевести вісімкове число  $X_8 = 105,71$  до десяткової системи числення.

$$\begin{aligned} 105,71_8 &= 1 \cdot 8^2 + 0 \cdot 8^1 + 5 \cdot 8^0 + 7 \cdot 8^{-1} + 1 \cdot 8^{-2} = \\ &= 64 + 0 + 5 + 0,875 + 0,015625 = 69,890625; \end{aligned}$$

$$X_{10} = 69,890625.$$

Приклад 2.3. Перевести шістнадцяткове число  $X_{16} = 2ED,0A$  до десяткової системи числення.

$$\begin{aligned} 2ED,0A_{16} &= 2 \cdot 16^2 + 14 \cdot 16^1 + 13 \cdot 16^0 + 0 \cdot 16^{-1} + 10 \cdot 16^{-2} = 512 + 224 + 13 + 0 + 0,0390625 = 849,0390625; \\ X_{10} &= 849,0390625. \end{aligned}$$

## 2.6 Переведення чисел із десяткової системи числення до іншої однорідної позиційної системи числення з основою $k$ , коли дії виконуються в десятковій системі

При даному переведенні окремо виконується переведення цілої частини числа й окремо - дробової, результати потім додаються.

### 2.7 Переведення цілої частини числа

Цілу частину числа  $X_{10}$  ділять на основу системи числення  $k$  за правилами десяткової арифметики до отримання залишку, який буде десятковим еквівалентом молодшої цифри результату. Якщо частка від ділення не дорівнює 0, то вона стає діленим і процес ділення на  $k$  продовжується. Як тільки чергова частка стане рівною 0,

процес ділення на  $k$  припиняється. Залишок, який отримали при першому діленні на  $k$ , є цифрою розряду результату з вагою  $k^0$ , залишок при другому діленні - цифрою з вагою  $k^1$  і т.д. Останній залишок є старшою цифрою результату.

## 2.8 Переведення дробової частини числа

**Таблиця 2.12. Переведення дробової частини числа до двійкової системи числення**

Крок	Дріб	Результат множення на $k=2$	Ціла частина результату множення, яка вилучається	Вага двійкового розряду
1	0,5625	1,125	1	Старший (перший після коми)
2	0,125	0,25	0	
3	0,25	0,5	0	
4	0,5	1,0	1	
5	0,0	0,0	0	Молодший

Дробова частина числа  $X_{10}$  помножується на  $k$  за правилами десяткової арифметики. В отриманому добутку від'єднується ціла частина, яка може дорівнювати 0, а дробова частина знову помножується на  $k$  із наступним від'єднанням цілої частини. Ця операція повторюється або до отримання нульової дробової частини добутку, або до отримання необхідної кількості розрядів числа  $X_k$ . Старша цифра результату переведення (тобто, перша після коми) збігається з першою від'єднаною цілою частиною, друга цифра результату - із другою від'єднаною цілою частиною і т.д. При цьому від'єднані цілі частини необхідно представити в системі числення з основою  $k$ .

Приклад 2.1. Перевести десяткове число 11,5625 до двійкової системи числення з точністю п'ять розрядів після коми.

Переведення цілої частини:

$11 : 2 = 5$ , залишок 1 (молодший розряд результату),

$5 : 2 = 2$ , залишок 1,

$2 : 2 = 1$ , залишок 0,

$1 : 2 = 0$ , залишок 1 (старший розряд результату).

Результат  $X_{2ц} = 1011$ .

Хід переведення дробової частини подає таблиця 2..



Результат  $X_{20} = 0,10010$ .

Повний результат  $X_2 = X_{24} + X_{20} = 1011 + 0,10010 = 1011,10010$ .

## 2.9 Переведення чисел з шістнадцяткової й вісімкової систем до двійкової і зворотне переведення чисел

Для переведення чисел з 16-кової і 8-кової систем числення до двійкової необхідно кожну цифру числа, яке переводиться, замінити відповідно чотири- або трирозрядним двійковим еквівалентом (табл. 1.1.1) - тетрадою або тріадою, а отримані двійкові цифри розташувати на місцях 16-кових або 8-кових цифр.

При необхідності переведення чисел із 10-кової системи числення до 8-кової, 16-кової та двійкової переведення робиться тільки до однієї системи (8-кової або 16-кової), подальше переведення виконується через двійкову систему з використання тріад і тетрад.

Приклад 2.1. Перевести число 12345,67 з десяткової системи числення до двійкової, 8-кової, 16-кової.

1) переведення цілої частини числа до 8-кової системи:

$$12345 : 8 = 1543, \text{ залишок } 1;$$

$$1543 : 8 = 192, \text{ залишок } 7;$$

$$192 : 8 = 24, \text{ залишок } 0;$$

$$24 : 8 = 3, \text{ залишок } 0;$$

$$3 : 8 = 0, \text{ залишок } 3;$$

результат  $30071_8$ .

2) переведення дробової частини числа до 8-кової системи:

$$0,67 \times 8 = 5,36;$$

$$0,36 \times 8 = 2,88;$$

$$0,88 \times 8 = 7,04;$$

$$0,04 \times 8 = 0,32;$$

наближений результат переведення дробової частини  $0,5270\dots$

Повний результат  $30071,5270\dots$

3) переведення до двійкової та 16-кової систем числення показує таблиця 2..

**Таблиця 2.13. Переведення з вісімкової до шістнадцяткової системи  
числення**

3			0			0			7			1			,	5			2			7			0			8-кові цифри
0	1	1	0	0	0	0	0	0	1	1	1	0	0	1	,	1	0	1	0	1	0	1	1	1	0	0	0	2-кові цифри
3			0			3			9			,	A			B			8			16- кові цифри						

Розбиття двійкового числа на тріади і тетради починається від коми ліворуч і праворуч.

Результат:  $12345,67_{10} = 30071,5270_8 = 11000000111001,101010111_2 = 3039,AB8_{16}$ .

### 3 ФОРМИ ПРЕДСТАВЛЕННЯ ЧИСЕЛ У ЦИФРОВИХ АВТОМАТАХ

Формою представлення чисел у цифрових автоматах називається сукупність правил, що дозволяють установити взаємну відповідність між записом числа і його кількісним еквівалентом.

Машинне (автоматне) зображення числа це є представлення числа в розрядній сітці цифрового автомата. Умовну позначку машинного зображення числа, наприклад,  $A$  будемо представляти як  $[A]$ .

Через обмежену довжину машинних слів, множина чисел, який можна представити в машині кінцева. Порівняння різних форм представлення чисел у комп'ютерах звичайно провадиться на основі оцінки діапазону і точності представлення числа.

У повсякденній практиці найбільш розповсюдженої є форма представлення чисел у вигляді послідовності цифр, розділеною комою на цілу і дробову частини. Числа, представлені в такій формі, називаються числами з природною комою або числами в природній формі. Наприклад, 12560 - ціле число, 0,003572 – дробове число, 4,89760 – змішане число..

При представленні чисел у такій формі обов'язково потрібною для кожного числа є вказівка про положення його коми в розрядній сітці, виділеній для представлення числа в машині, що вимагає доповняльних апаратних витрат досить великого обсягу. Тому в комп'ютерах одержали поширення дві інші форми представлення: з фіксованою комою і з рухомою комою.

#### 3.1 Форма представлення двійкових чисел з фіксованою комою

Необхідність у вказівці положення коми відпадає, якщо місце коми в розрядній сітці машини заздалегідь фіксовано раз і назавжди. Така форма представлення чисел називається представленням з фіксованою комою (крапкою).

Оскільки числа бувають додатні і від'ємні, то формат (розрядна сітка) машинного зображення розбивається на два поля: поле знаку (знакову частину) і поле числа. У полі числа розміщується саме зображення числа, що ми будемо умовно називати мантисою числа. Для кодування знака числа використовується самий старший розряд розрядної сітки, відведеної для зображення двійкового числа, а інші розряди приділяються під мантису числа. Положення коми в розрядній сітці строго

фіксується, звичайно або правіше самого молодшого розряду мантиси, або лівіше самого старшого. У першому випадку число представляється як ціле, у другому - як правильний дріб. В даний час, переважно, в комп'ютерах у форматі з фіксованою крапкою представляються цілі числа.

У знакову частину записується інформація про знак числа. Прийнято, що знак додатнього числа "+" зображується символом 0, а знак від'ємного числа "-" зображується символом 1.

Наприклад, у двійковому коді, використовуючи 6-розрядну сітку, число 7 у формі з фіксованою комою можна представити у вигляді  $0.00111_2$ , де цифра лівіше крапки це знак числа, а п'ять цифр правіше крапки - мантиса числа в прямому коді. Тут мається на увазі, що кома фіксована правіше молодшого розряду, а крапка в зображенні числа в даному випадку просто розділяє знаковий біт від мантиси числа.

Надалі часто буде використовуватися в прикладах такий вигляд представлення числа в машинній формі. Можна використовувати й іншу форму представлення числа в машинній формі  $[0]00111_2$ , де знаковий розряд виділяється квадратними дужками.

Кількість розрядів у розрядній сітці, відведене для зображення мантиси числа, визначає діапазон і точність представлення числа з фіксованою комою. Максимальне за абсолютною величиною двійкове число зображується одиницями у всіх розрядах, крім знакового, тобто для цілого числа  $|A|_{max} = (2^{n-1} - 1)$ , де  $n$  - повна довжина розрядної сітки. У випадку 16-розрядної сітки  $|A|_{max} = (2^{16-1} - 1) = 32767_{10}$ , тобто, діапазон представлення цілих чисел у цьому випадку буде від  $+32767_{10}$  до  $-32767_{10}$ .

Для випадку, коли кома фіксується правіше молодшого розряду мантиси, тобто для цілих чисел, числа, у яких модуль більше, ніж  $(2^{n-1} - 1)$  і менше одиниці, не представляються у формі з фіксованою комою. Числа, за абсолютною величиною менші одиниці молодшого розряду розрядної сітки, називаються в цьому випадку машинним нулем. Від'ємний нуль заборонений.

У деяких випадках, коли можна оперувати тільки модулями чисел, уся розрядна сітка, включаючи самий старший розряд, приділяється для представлення числа, що дозволяє розширити діапазон зображення чисел.

### 3.2 Представлення від'ємних чисел у форматі з фіксованою комою

У комп'ютерах з метою спрощення виконання арифметичних операцій застосовуються спеціальні двійкові коди для представлення від'ємних чисел: обернений і доповняльний. За допомогою цих кодів спрощується визначення знака результату операції при алгебраїчному додаванні. Операція віднімання (або алгебраїчного додавання) зводиться до арифметичного додавання операндів, полегшується вироблення ознак переповнення розрядної сітки. У результаті спрощуються пристрої комп'ютера, що виконують арифметичні операції.

Відомо, що одним зі способів виконання операції віднімання є заміна знака від'ємника на протилежний і додавання його до зменшуваного:

$$A - B = A + (-B).$$

Цим операцію арифметичного віднімання заміняють операцією алгебраїчного додавання, яку можна виконати за допомогою двійкових суматорів.

Для машинного представлення від'ємних чисел використовують коди прямий, доповняльний, обернений. Спрощене визначення цих кодів може бути дане в такий спосіб. Якщо додатнє число  $A$  в звичайному двійковому коді - прямому двійковому коді, зобразити як  $[A]_{ПК} = 0.a_n a_{n-1} a_{n-2} \dots a_1 a_0$ . Тоді від'ємне число  $-A$  в цьому ж коді представляється як  $[-A]_{ПК} = 1.a_n a_{n-1} a_{n-2} \dots a_1 a_0$ , а в оберненому (інверсному) коді це число буде мати вигляд  $[-A]_{ОК} = 1.\overline{a_n} \overline{a_{n-1}} \overline{a_{n-2}} \dots \overline{a_1} \overline{a_0}$ , де рисочками над символами позначена операція інверсії (обернення):  $\overline{a_i} = 1$ , якщо  $a_i = 0$ ,  $\overline{a_i} = 0$ , якщо  $a_i = 1$ ,  $a_i$  - цифра  $i$ -того розряду двійкового числа. Отже, при переході від прямого коду до оберненого всі цифри розрядів мантиси числа інвертуються.

Тоді число  $-A$  у доповняльному коді зображується у вигляді  $[-A]_{ДК} = [-A]_{ОК} + 1$ .

Таким чином, для одержання доповняльного коду від'ємних чисел потрібно спочатку інвертувати цифрову частину вихідного числа, у результаті чого утворюється його обернений код, а потім додати одиницю в молодший розряд цифрової частини числа.

Доповняльний код деякого числа утворюється його заміною на нове число, що доповнює його до числа, рівного вазІ розряду, що іде за самим старшим розрядом розрядної сітки, використовуваної для представлення мантиси числа у форматі з фіксованою комою. Тому такий код числа називається доповняльним.

Припустимо, що ми маємо тільки два розряди для представлення чисел у десятковій системі числення. Тоді максимальне число, яке можна зобразити буде 99, а вага третього неіснуючого старшого розряду буде 102, тобто 100. У такому випадку для числа 20 доповняльним буде число 80, що доповнює 20 до 100 ( $100 - 20 = 80$ ). Отже, за визначенням, віднімання  $50 - 20 = 30$  можна замінити на додавання:

$$50 + 80 = (1)30.$$

Тут старша одиниця виходить за межі виділеної розрядної сітки, у якій залишається тільки число 30, тобто результат віднімання з 50 числа 20.

А тепер розглянемо схожий приклад для чисел, представлених 4-х розрядним двійковим кодом. Знайдемо доповняльне число для  $0010_2 = 2_{10}$ . Треба з  $[1]0000$  відняти  $[0]0010$ , одержимо  $[0]1110$ , що і є доповняльним кодом 2. Розряд, зображений у квадратних дужках насправді не існує. Але тому що в нас 4-х розрядна сітка, то виконати таке віднімання в принципі неможливо, а тим більше ми намагаємося позбутися від віднімання. Тому доповняльний код числа одержують способом, описаним раніше, тобто спочатку одержують обернений код числа, а потім додають до нього 1. Проробивши все це з нашим числом (2), неважко переконатися, що вийде аналогічна відповідь.

Підкреслимо, що доповняльний і обернений коди використовуються тільки для представлення від'ємних двійкових чисел у формі з фіксованою комою. Додатні числа в цих кодах не змінюють свого зображення і представляються як у прямому коді.

Нагадаємо, що цифрові розряди від'ємного числа в прямому коді залишаються незмінними, а в знаковій частині записується одиниця.

Розглянемо прості приклади.

Сім у прямому коді представляється так:  $[7]_{ПК} = 0.000111_2$ .

Число -7 у прямому коді  $[-7]_{ПК} = 1.000111_2$ , а в оберненому коді буде мати вигляд  $[-7]_{ПК} = 1.111000_2$ , тобто. одиниці замінюються нулями, а нулі одиницями. Те ж число в доповняльному коді буде  $[-7]_{ДК} = 1.111001_2$ .

Розглянемо ще раз як процедура віднімання, за допомогою представлення від'ємника в доповняльному коді, зводиться до процедури додавання. Віднімемо з 10 число 7:  $10 - 7 = 3$ . Якщо обидва операнда представлено в прямому коді, то процедура віднімання виконується так:

$$\begin{array}{r}
 0.001010 \\
 - 1.000111 \\
 \hline
 0.000011
 \end{array}$$

А якщо від'ємник, тобто  $-7$ , представити в доповняльному коді, то процедура віднімання зводиться до процедури додавання:

$$\begin{array}{r}
 0.001010 \\
 + 1.111001 \\
 \hline
 (1)0.000011
 \end{array}$$

Цифра 1, яку взято в дужки, виходить за межі розрядної сітки і губиться (зникає).

У даний час у комп'ютерах для представлення від'ємних чисел у форматі з фіксованою комою звичайно використовується доповняльний код.

### 3.3 Форма представлення чисел з рухомою комою

У загальному випадку число у формі з рухомою комою (інша назва – з комою, що плаває) представляється у вигляді  $A = \pm mq^{\pm p}$ , де  $m$  - мантиса числа,  $q$  - основа системи числення,  $q^{\pm p} = P$  - порядок числа, що для спрощення в прикладах будемо іноді зображати як  $P$ . Тоді очевидно, що  $p$  - це показник степеня порядку, що звичайно називають просто порядком числа. Отже, попередній вираз можна записати як  $A = \pm mP$ , маючи на увазі, що в комп'ютерах звичайно  $q = 2$ . Якщо  $q$  додатне число, то  $P$  – завжди додатне число.

Для забезпечення однозначного і максимально точного представлення чисел прийнято представляти число з рухомою комою в так званому нормалізованому вигляді. Якщо виконується нерівність  $q^{-1} \leq |m| < 1$ , а у випадку двійкової системи числення  $0,5 \leq |m| < 1$ , тобто, якщо старший двійковий розряд мантиси дорівнює 1, то вважається, що число представлено в нормалізованому вигляді.

Таким чином, у двійкового нормалізованого числа у формі з рухомою комою мантиса в старшому розряді мантиси завжди стоїть 1. Операція приведення числа до нормалізованого вигляду називається нормалізацією. Нормалізація чисел у комп'ютері виконується або автоматично або ж спеціальною програмою.

Оскільки система числення для заданого цифрового автомата (комп'ютера) є постійною, то при представленні числа у форматі з рухомою комою немає

необхідності вказувати її основу, досить лише представити показник степеня порядку числа.

Для представлення двійкового числа у формі з рухомою комою у розрядній сітці, виділеній для цієї мети, приділяється по одному розряду для представлення знака числа  $S_m$  (поле знаку числа) і знака показника степеня порядку  $S_p$  (поле знаку порядку); певне число розрядів для представлення значення самого показника  $p$  (поле показника порядку), а також розряди для розміщення значення модуля мантиси  $m$  (поле мантиси). Наприклад, можливий наступний варіант, коли формат числа складається з чотирьох полів, тобто,  $[A] = S_p p S_m m$  (рис. 3.1).

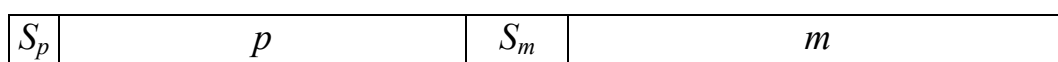


Рис. 3.1. Формат числа з рухомою комою (чотири поля)

Звичайно у форматі з рухомою комою замість показника  $p$  використовують так звану характеристику ("зміщений порядок")  $r = \pm p + l$ , де  $l$  - надлишок (зсув), значення якого підбирається таким чином, щоб при зміні значення показника від деякого мінімального значення  $-|p_{min}|$  до максимального  $+|p_{max}|$ , характеристика  $r$  мінялася від 1 до  $r_{max}$ . Отже, характеристика не змінює свого знака. У такому випадку відпадає необхідність у відображенні знака порядку  $S_p$ . Для цього приймається, що  $l = 2^{k-1}$ , де  $k$  - число розрядів, виділених для представлення порядку числа у форматі з рухомою комою.

Важливо зауважити, що рівна 0 характеристика  $r$  ( $r=0$ ) використовується як ознака рівності 0 усього числа  $A$  ( $A=0$ ). При цьому мантиса  $m$  ніколи не дорівнює 0 ( $m > 0$ ).

Тоді формат числа з рухомою комою можна представити з використанням трьох полів, тобто,  $[A] = S_m r m$  (рис. 3.2).

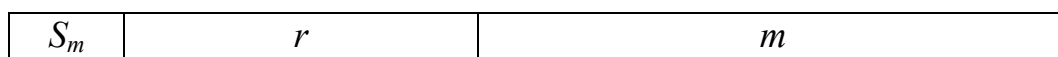


Рис. 3.2. Формат числа з рухомою комою (три поля)

Одиниця самого старшого розряду нормалізованої мантиси звичайно є "схованою", тобто, є уявною і не відображається у форматі числа. Розряд слова, у якому повинна була бути відображена ця одиниця, використовується як молодший розряд характеристики, що дозволяє збільшити діапазон представлення чисел у форматі з рухомою комою.



Таким чином, мантиса в такому варіанті відображається, починаючи з розряду, що йде після самого старшого. При всіх операціях з мантисою числа цю обставину треба враховувати і перед початком операцій відновлювати старший розряд мантиси.

Після завершення операцій під час формування відображення нормалізованого результату у відведеній для нього розрядній сітці машинних слів, старша одиниця мантиси знову відкидається.

$n$ -розрядне поле порядку дозволяє змінювати показник порядку в межах від  $-2^{n-l} + 1$  до  $+2^{n-l} - 1$  (таблиця 3.,  $n=3$ ).

**Таблиця 3.1. Показник порядку і характеристика**

Показник порядку	Прямий код показника	Характеристика (показник+4)	Примітки
+3	011	111	3+4=7
+2	010	110	2+4=6
+1	001	101	1+4=5
0	000	100	0+4=4
-1	101	011	-1+4=3
-2	110	010	-2+4=2
-3	111	001	-3+4=1
		000	Ознака нуля

Як уже відзначалося, характеристика  $r$  - це показник порядку  $p$  з надлишком  $l=2^{n-l}$ , вона не змінює свого знака і змінюється від 1 (при  $p=-2^{n-l}$ ) до  $2^{n-l}$  (при  $p=+2^{n-l} - 1$ ).

Виключення складає число 0: нуль виражається нульовою характеристикою і (не обов'язково) нульовою мантисою.

Основною перевагою представлення чисел у формі з рухомою комою є великий діапазон машинних чисел і висока точність їхнього представлення. Діапазон визначається довжиною розрядної сітки, виділеної під характеристику, а точність - визначається довжиною розрядної сітки, виділеної під мантису.

Для формату чисел з рухомою комою (таблиця 3.) значення деяких кодів подає таблиця 3..

**Таблиця 3.2. Приклад формату з рухомою комою**

Номер розряду	Призначення поля	Примітки
---------------	------------------	----------

7	знак числа	0 – число додатне; 1 – число від’ємне
6..4	порядок, зміщений на 4	ознака рівності числа нулю – усі розряди порядка дорівнюють 0
3..0	мантиса числа	мантиса 5-розрядна, нормалізована, старший розряд мантиси завжди дорівнює 1 й у форматі не відображається

Довжина розрядної сітки, виділеної під характеристику, визначає діапазон представлення чисел у форматі з рухомою комою.

**Таблиця 3.3. Приклади чисел у форматі з рухомою комою**

Число (16-кова система числення)	7f	70	10	0X	90	ff
Число (2-ва система числення)	01111111	01110000	00010000	0000xxxx	10010000	11111111
Число у форматі з рухомою комою, розбите на поля	0.111.1111	0.111.0000	0.001.0000	0.000.xxxx	1.001.0000	1.111.1111
Знак	+ (0)	+ (0)	+ (0)	+ (0)	- (1)	- (1)
Характеристика	7=111	7=111	1=001	0=000	1=001	7=111
Показник порядку	3=7-4	3=7-4	-3=1-4	не задіяний	-3=1-4	3=7-4
Порядок	$8=2^3$	$8=2^3$	$1/8=2^{-3}$	не задіяний	$1/8=2^{-3}$	$8=2^3$
Мантиса	$31/32=$ $=0,11111$	$1/2=$ $=0,10000$	$1/2=$ $=0,10000$	не задіяна	$1/2=$ $=0,10000$	$31/32=$ $=0,11111$
Величина числа	$+8*31/32=$ $=+7^3/4$	$+8*1/2=+4$	$+1/8*1/2=$ $=+1/16$	0	$-1/8*1/2=$ $=-1/16$	$-8*31/32=$ $=-7^3/4$
Примітка	Найбільше додатне число		Найменше додатне число	Число 0	Найменше за модулем від’ємне число	Найбільше за модулем від’ємне число

Як уже відзначалося, модуль мантиси нормалізованого числа -  $m$ , представленого у форматі з рухомою комою лежить у межах  $2^{-1} \leq |m| < 1 - 2^{-n}$ , де  $n$  - число розрядів, виділених для представлення мантиси числа  $A$ . Якщо для

представлення порядку  $p$  виділено  $k$  розрядів, то  $p_{min} = -2^k$ ,  $p_{max} = 2^k - 1$ .

Тоді діапазон, у якому може бути представлене число у форматі з рухомою комою, буде дорівнює  $2^{p_{min}} \cdot 2^{-1} \leq |A| \leq (1 - 2^{-n}) 2^{p_{max}}$  або  $2^{-2^k} \cdot 2^{-1} \leq |A| \leq (1 - 2^{-n}) 2^{2^k - 1}$ .

Основною перевагою представлення чисел у формі з рухомою комою є великий діапазон машинних чисел і висока точність їхнього представлення. Діапазон визначається довжиною розрядної сітки, виділеної під характеристику, а точність - визначається довжиною розрядної сітки, виділеної під мантису.

### 3.4 Перетворення форматів чисел

Перед описом перетворення формату з фіксованою комою до формату з рухомою комою і навпаки треба попередньо описати деякі особливості чисел з рухомою комою:

збільшення мантиси в 2 рази здійснюється зсувом двійкового значення мантиси ліворуч (у бік старших розрядів);

зменшення мантиси в 2 рази здійснюється зсувом двійкового значення мантиси праворуч (у бік молодших розрядів);

величина числа не зміниться, якщо збільшити мантису в 2 рази і одночасно зменшити показник порядку на 1;

величина числа не зміниться, якщо зменшити мантису в 2 рази і одночасно збільшити показник порядку на 1.

На початку розглянемо алгоритм переведення числа з формату з фіксованою комою у формат з рухомою комою.

Переведення числа з формату з фіксованою комою  $[A]_{\phi k}$  у формат з рухомою комою  $[A]_{p k}$  можливий тільки у випадку виконання наступної умови  $n_m \geq n_\phi$ , де  $n_m$  це довжина мантиси для числа у форматі з рухомою комою, а  $n_\phi$  - довжина мантиси для числа у форматі з фіксованою комою.

Число з фіксованою комою  $\pm m_{\phi k}$ , яке записане у прямому коді і яке потрібно перевести до формату з рухомою комою, представимо у вигляді добутка  $A = \pm m_{\phi k} \cdot q^{\pm p}$ ,  $q^{\pm p} = 1$ ,  $q = 2$ , тобто,  $p=0$ . Результатом переведення повинні бути:

знак числа  $S_m$  ( $\pm$ );

нормалізована мантиса  $m$ ;

характеристика  $r$ ,

які треба записати у форматі (рис. 3.2).

Якщо мантиса  $m_{\Phi K}$  денормалізована і  $m_{\Phi K} \geq 1$ , її треба нормалізувати зсувами праворуч у бік молодших розрядів. Зсуви мантиси (її зменшення) потрібно компенсувати збільшенням показника  $p$  (на скільки розрядів зсунули мантису, на стільки одиниць треба збільшити  $p$ ).

Якщо мантиса  $m_{\Phi K}$  денормалізована і  $m_{\Phi K} \leq 1/2$ , її треба нормалізувати зсувами ліворуч у бік старших розрядів. Зсуви мантиси (її збільшення) потрібно компенсувати зменшенням показника  $p$  (на скільки розрядів зсунули мантису, на стільки одиниць треба зменшити  $p$ ).

Коли мантиса буде нормалізована, треба визначити характеристику  $r=p+4$  (рис. 3.2).

Наприклад, переведемо  $5_{10} = 00101_2$  у формат з рухомою комою,  $m_{\Phi K} = 5 > 1$ .

Послідовність виконання процедури нормалізації ілюструють таблиця 3. і таблиця 3., де показані зміни мантиси, порядку і характеристики в десятковому і двійковому коді.

**Таблиця 3.4. Утворення формату з рухомою комою для числа  $\geq 1$**

Крок	Мантиса			Порядок		Характеристика	
	$m_{10}$	$m_2$	нормалізація	$p_{10}$	$p_2$	$r_{10}$	$r_2$
1	5	101	денормалізована	0	000	0+4=4	100
2	2,5	10,1	денормалізована	1	001	1+4=5	101
3	1,25	1,01	денормалізована	2	010	2+4=6	110
4	0,625	0,101	нормалізована	3	011	3+4=7	111

Результат перетворення:

додатне число, у полі знаку 0;

у полі характеристики 111;

підкреслені розряди мантиси 0,10100 (без прихованої 1) заносяться до поля мантиси.

Результат показує рис. 3.3. Результат: число +5 кодується у форматі з рухомою комою як  $01110100_2 = 74_{16} = 116_{10}$ .

$S_m$	$r$	$m$
0	111	0100

Рис. 3.3. Число (більше або рівне 1) у форматі з рухомою комою

Наприклад, переведемо  $3/16 = 0,1875_{10} = 0,0011_2$  у формат з рухомою комою,  $m_{\text{фк}} = 3/16 \leq 1/2$ .

Таблиця 3.5. Утворення формату з рухомою комою для числа  $<1/2$ 

Крок	Мантиса			Порядок		Характеристика	
	$m_{10}$	$m_2$	нормалізація	$p_{10}$	$p_2$	$r_{10}$	$r_2$
1	0,1875	0,0011	денормалізована	0	000	0+4=4	100
2	0,375	0,0111	денормалізована	-1	-001	-1+4=3	011
3	0,75	0,11	нормалізована	-2	-010	-2+4=2	010

Результат перетворення:

додатнє число, у полі знаку 0;

у полі характеристики 010;

підкреслені розряди мантиси 0,11000 (без прихованої 1) заносяться до поля мантиси.

Результат показує рис. 3.4. Результат: число  $3/16 = 0,1875_{10} = 0,0011_2$  кодується у форматі з рухомою комою як  $00101000_2 = 28_{16} = 40_{10}$ .

$S_m$	$r$	$m$
0	010	1000

Рис. 3.4. Число (менше  $1/2$ ) у форматі з рухомою комою

А тепер наведемо принцип переведення числа з формату з рухомою комою у формат з фіксованою комою. Розглянемо той же випадок, коли  $n_m \geq n_\phi$ .

Для переведення необхідно значення полів формату числа з рухомою комою підставити у вираз  $A = \pm m_{\text{фк}} \cdot q^{\pm p}$  і обрахувати значення числа.

Наприклад, визначити величину числа, закодованого у відповідності з форматом (рис. 3.2), код якого рівний  $74_{16} = 01110100_2 = 0\_111\_0100_2$  (формат, розбитий на поля відповідно знаку, характеристики і мантиси). Звідси маємо:

поле знаку 0, тобто, число додатнє (знак +);

поле характеристики 111, характеристика  $r = 111_2 = 7_{10}$ , тобто показник порядку  $p = r - 4 = 7 - 4 = +3$ ;

поле мантиси 0100, мантиса  $m = 0,10100_2 = 5/8$ .

Тепер можна визначити величину числа:  $A = \pm m_{\text{фк}} \cdot q^{\pm p} = +(5/8) \cdot 2^{+3} = +(5/8) \cdot 8 = +5$ .

### 3.5 Похибки представлення чисел

Як уже відзначалося, представлення числової інформації в цифровому автоматі, як правило, спричиняє появу похибок (помилки), величина яких залежить як від форми представлення чисел, так і від довжини розрядної сітки цифрового автомата, тобто комп'ютера. Отже, при переведенні чисел з однієї системи числення в іншу неминуче виникають похибки.

Абсолютна похибка представлення  $\Delta[A]$  - це різниця між істинним значенням вхідної величини  $A$  та її значенням, отриманим з машинного представлення  $A_M$ , тобто

$$\Delta[A] = |A - A_M|$$

Тоді відносна похибка представлення дорівнює  $\delta[A] = |\Delta[A]/A|$

Максимальна похибка переведення десяткової інформації в двійкову не перевищує одиниці молодшого розряду розрядної сітки цифрового автомата. Мінімальна похибка переведення дорівнює нулеві.

Усереднена абсолютна похибка  $\Delta[A]_c$  переведення чисел у двійкову систему числення дорівнює:

$$\Delta[A]_c = 0,5 \cdot 2^{-n} \text{ для дробів та змішаних чисел і}$$

$$\Delta[A]_c = 0,5 \cdot 2^0 \text{ для цілих чисел}$$

Для представлення чисел у формі з фіксованою комою абсолютне значення  $|[A]_\phi|$  машинного зображення числа лежить у межах:

$$0 \leq [A]_\phi \leq 1 - 2^{-n} \text{ для дробів і}$$

$$0 \leq [A]_\phi \leq 2^n - 1 \text{ для цілих чисел.}$$

Найменше не нульове значення чисел з фіксованою комою за абсолютною величиною дорівнює

$$|[A]_{\phi min}| = 2^{-n} \text{ для дробів і}$$

$$|[A]_{\phi min}| = 2^0 \text{ для цілих чисел.}$$

Отже, відносні похибки представлення для мінімального не рівного 0 значення числа рівні:

$$\delta[A]_{\phi min} = \frac{\Delta[A]}{[A]_{\phi min}} = \frac{0,5 \cdot 2^{-n}}{2^{-n}} = 0,5 \text{ для дробів і}$$

$$\delta[A]_{\text{фmin}} = \frac{\Delta[A]}{[A]_{\text{фmin}}} = \frac{0,5 \cdot 2^0}{2^0} = 0,5 \text{ для цілих чисел.}$$

Аналогічно для максимального значення:

$$\delta[A]_{\text{фmax}} = \frac{\Delta[A]}{[A]_{\text{фmax}}} = \frac{0,5 \cdot 2^{-n}}{1 - 2^{-n}} \approx 2^{-(n+1)} \text{ для дробів і}$$

$$\delta[A]_{\text{фmax}} = \frac{\Delta[A]}{[A]_{\text{фmax}}} = \frac{0,5 \cdot 2^0}{2^n - 1} \approx 2^{-(n+1)} \text{ для цілих чисел.}$$

Як бачимо похибки представлення малих чисел у формі з фіксованою комою можуть бути дуже значними (50%).

Для представлення чисел у формі з рухомою комою абсолютне значення мантиси лежить у межах  $2^{-1} \leq m \leq 1 - 2^{-n}$ .

Тому що похибка представлення числа залежить тільки від розрядності мантиси, то для знаходження похибки представлення числа у формі з рухомою комою величину цієї похибки треба помножити на величину порядку числа  $P$ :

$$\delta[A]_{\text{нmax}} = \frac{0,5 \cdot 2^{-n} \cdot P}{1 - 2^{-n} \cdot P} \approx 2^{-(n+1)} \quad \text{і} \quad \delta[A]_{\text{нmin}} = \frac{0,5 \cdot 2^{-n} \cdot P}{2^{-1} \cdot P} = 2^{-n},$$



де  $n$  - кількість розрядів для представлення мантиси числа. З останньої формули випливає, що відносна точність представлення чисел у формі з рухомою комою майже не залежить від величини числа.

## 4 АРИФМЕТИЧНІ ДІЇ З ДВІЙКОВИМИ ЧИСЛАМИ

### 4.1 Додавання двійкових чисел

Арифметичні операції можна виконувати з двійковими числами, представленими в прямому, оберненому і доповняльному коді. Якщо операнди представлені в прямому коді і мають однакові знаки, те над ними при алгебраїчному додаванні просто виконується процедура додавання. Якщо ж операнди мають різні знаки - процедура віднімання. Як уже відзначалося, для спрощення апаратних засобів комп'ютера процедура віднімання замінюється додаванням завдяки тому, що від'ємний операнд представляється в оберненому або доповняльному коді.

### 4.2 Алгебраїчне додавання чисел з фіксованою комою

Структурні схеми двійкових суматорів, призначених для виконання алгебраїчного додавання чотирирозрядних чисел  $A = (a_3, a_2, a_1, a_0)$  і  $B = (b_3, b_2, b_1, b_0)$  (у дужках позначені двійкові розряди чисел, з індексом 0 – молодший, з індексом 3 – старший) у прямому, оберненому і доповняльному кодах містить рис. 4.1 а, б і в відповідно. Результатом є число  $S = (s_3, s_2, s_1, s_0)$ . На схемах (рис. 4.1) символами  $a_3, b_3, s_3$  позначено знакові біти операндів і результату (від слова „знак”, інші розряди – значущі (від слова „значення”). Символом  позначені входи, а символом  - виходи, D1-D4 – позиційні позначення цифрових елементів на схемі.



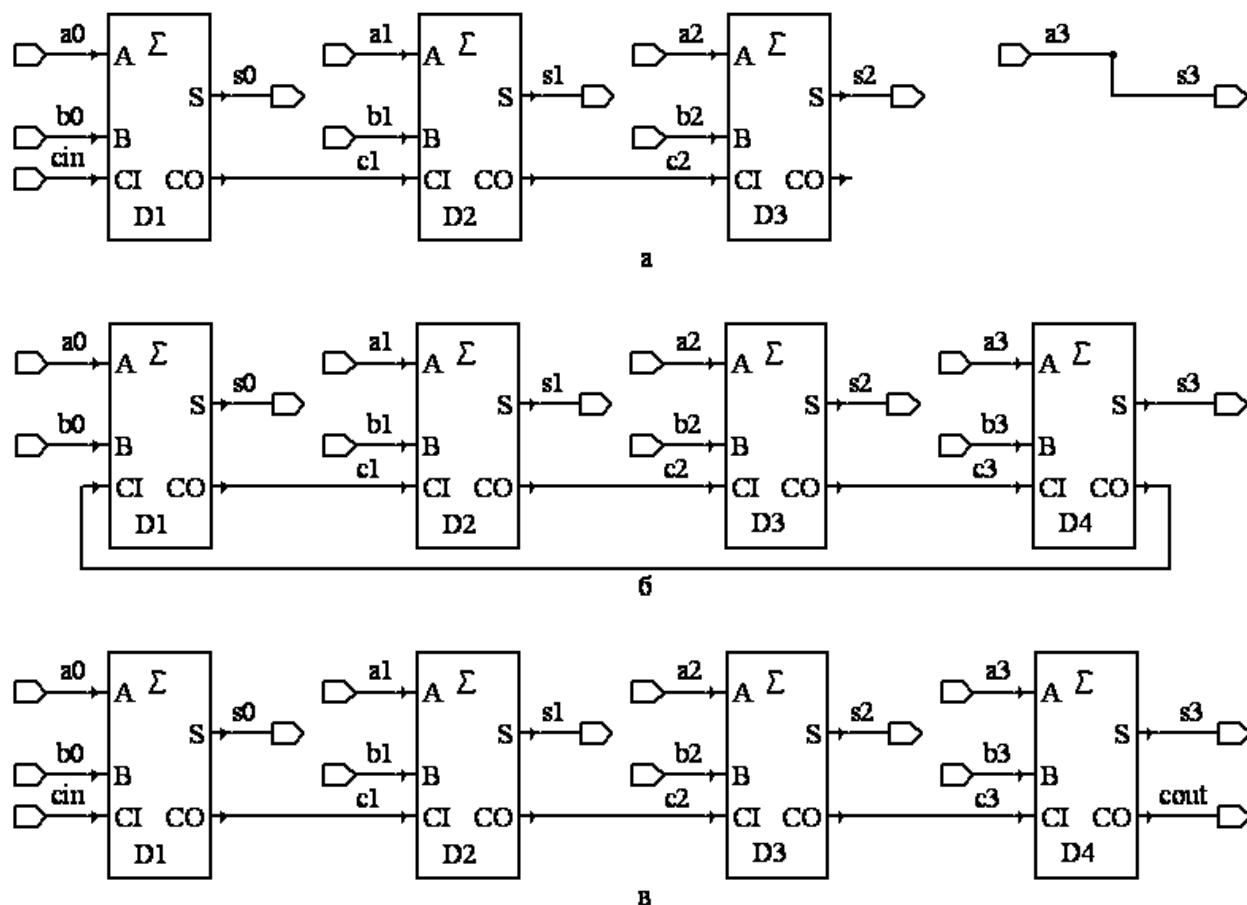


Рис. 4.1. Чотирирозрядні суматори

При додаванні двох чисел, представлених в оберненому або доповняльному коді, їхні знакові розряди додаються аналогічно значущим. При додаванні в оберненому коді можливий циклічний перенос одиниці зі старшого розряду в молодший.

Якщо обидва операнда представлені в прямому коді і мають однакові знаки, то над такими операндами виконується процедура додавання і результату присвоюється знак одного з операндів (наприклад, знак числа  $A - a_3$ ).

Розглянемо приклади (крапка відділяє знак від значущих розрядів, а не цілу частину від дробової, усі числа – цілі!):

$$\begin{array}{r}
 1) \quad 2_{10} + 4_{10} = 6_{10} \\
 \begin{array}{r}
 0.010 \\
 +0.100 \\
 \hline
 0.110 = 6_{10}
 \end{array}
 \end{array}
 \quad
 \begin{array}{r}
 2) \quad -2_{10} - 4_{10} = -6_{10} \\
 \begin{array}{r}
 1.110 \\
 +1.100 \\
 \hline
 1\leftarrow 1.110 = -6_{10}
 \end{array}
 \end{array}
 \quad
 \begin{array}{r}
 3) \quad -2_{10} - 4_{10} = -6_{10} \\
 \begin{array}{r}
 1.101 \\
 +1.011 \\
 \hline
 1\leftarrow 1.000 \\
 \quad \quad \quad \rightarrow +1 \\
 \hline
 1.001 = -6_{10}
 \end{array}
 \end{array}$$

В другому прикладі обидва від'ємні доданки представлені в доповняльному коді, а в третьому - в оберненому. У першому випадку результат отриманий у

доповняльному коді, а в другому, після циклічного переносу, - в оберненому.

Якщо ж операнди мають різні знаки, то повинна виконуватися процедура віднімання. Але, як уже відзначалося, процедуру віднімання заміняють процедурою додавання, від'ємний операнд (від'ємник) представляють в оберненому або доповняльному коді, а додатній операнд - у прямому коді.

Наприклад:

$$\begin{array}{l}
 1) 4_{10} - 2_{10} = 2_{10} \\
 \begin{array}{r}
 \text{a) } 0.100 \\
 +1.110 \\
 \hline
 1\leftarrow 0.010 \\
 = 2_{10}
 \end{array}
 \end{array}
 \qquad
 \begin{array}{l}
 \text{b) } 0.100 \\
 +1.101 \\
 \hline
 1\leftarrow 0.001 \\
 \rightarrow +1 \\
 \hline
 0.010 \\
 = 2_{10}
 \end{array}$$

$$\begin{array}{l}
 2) 4_{10} - 4_{10} = 0_{10} \\
 \begin{array}{r}
 \text{a) } 0.100 \\
 +1.100 \\
 \hline
 1\leftarrow 0.000 \\
 = 0_{10}
 \end{array}
 \end{array}
 \qquad
 \begin{array}{l}
 \text{b) } 0.100 \\
 +1.011 \\
 \hline
 1.111 \\
 = -0_{10}
 \end{array}$$

$$\begin{array}{l}
 3) 2_{10} - 4_{10} = -2_{10} \\
 \begin{array}{r}
 \text{a) } 0.010 \\
 +1.100 \\
 \hline
 1.110 \\
 = -2_{10}
 \end{array}
 \end{array}
 \qquad
 \begin{array}{l}
 \text{b) } 0.010 \\
 +1.011 \\
 \hline
 1.101 \\
 = -2_{10}
 \end{array}$$

У прикладах 1а) і 2а) одиниця, що виходить за розрядну сітку, не враховується ("губиться").

Як видно з прикладів, процедура алгебраїчного додавання з доповняльними кодами простіша, ніж з оберненими кодами, тому що, в останньому випадку при виникненні одиниці переносу за межі розрядної сітки, виділеної для числа у форматі з фіксованою комою, доводиться коректувати результат за допомогою додаткової процедури додавання одиниці до результату. Нульовий результат утворюється або в прямому коді або ж в оберненому, що також вимагає корекції результату. Тому в даний час на практиці для представлення від'ємних операндів використовується в основному доповняльний код.

Таким чином, перед виконанням самої процедури алгебраїчного додавання в доповняльному коді потрібно проаналізувати знаки доданків. Якщо вони різні, то виконується алгебраїчне додавання (фактично віднімання) без перевірки на переповнення результату, тому що його в цьому випадку просто не може бути. Якщо ж

у операндів знаки однакові, то такий же знак попередньо присвоюється результату і виконується додавання. Якщо знак остаточного результату не збігається з попередньо присвоєним знаком, то це є ознакою неправильного результату, так званого переповнення.

Ще раз підкреслимо, що результат алгебраїчного додавання операндів представлених в оберненому (доповняльному) кодах утворюється також в оберненому (доповняльному) коді.

### 4.3 Переповнення розрядної сітки

При додаванні чисел однакового знака, представлених у формі з фіксованою комою, може виникнути переповнення розрядної сітки.

Ознакою переповнення розрядної сітки при додаванні чисел у прямому коді є поява одиниці переносу зі старшого значущого розряду числа.

Наприклад (додаються два додатніх числа у доповняльному коді, а результат виходить від'ємний):

$$\begin{array}{r} 0.011 \quad = +3_{10} \\ +0.110 \quad = +6_{10} \\ \hline 1.001 \quad = -7_{10} ? \end{array}$$

Ознака переповнення розрядної сітки при додаванні чисел у доповняльному й оберненому кодах - одержання знака результату, протилежного знакам операндів.

Наприклад:

$$\begin{array}{l} 1) \begin{array}{r} 0.011 \quad = +3_{10} \\ +0.110 \quad = +6_{10} \\ \hline 1.001 \quad = -7_{10} ? \end{array} \quad 2) \begin{array}{r} 1.101 \quad = -3_{10} \\ +1.010 \quad = -6_{10} \\ \hline 1\leftarrow 0.111 \quad = +7_{10} ? \end{array} \quad 3) \begin{array}{r} 1.100 \quad = -3_{10} \\ +1.001 \quad = -6_{10} \\ \hline 1\leftarrow 0.101 \\ \quad \quad \quad \rightarrow +1 \\ \hline 0.110 \quad = +6_{10} ? \end{array} \end{array}$$

1) додавання додатніх чисел  $3+6$ , результат неправильний, в доповняльному коді він буде сприйматися як  $-7$ , а в оберненому коді – як  $-6$ ;

2) додавання від'єних чисел у доповняльному коді  $(-3)+(-6)$ , результат неправильний, в доповняльному коді він буде сприйматися як  $+7$ ;

3) додавання від'єних чисел в оберненому коді  $(-3)+(-6)$ , результат неправильний, в оберненому коді він буде сприйматися як  $+6$ .

При множенні будь-яких чисел також може виникнути переповнення розрядної сітки.

Для виявлення переповнення розрядної сітки в складі цифрового автомата повинні бути передбачені апаратні засоби, що автоматично виробляють ознаку переповнення - сигнал переповнення.

Один з методів виявлення переповнення розрядної сітки припускає введення допоміжного розряду в знакову частину зображення числа. Цей додатковий розряд називають розрядом переповнення. Таке представлення числа називають модифікованим. Здебільшого використовується модифікований доповняльний код.

#### 4.4 Модифікований доповняльний код

Модифікований доповняльний код відрізняється від звичайного доповняльного тим, що має два знакових розряду. При виконанні арифметичних дій над двійковими числами ці два знаки дозволяють легко визначити переповнення розрядної сітки. Якщо вміст цих двох розрядів збігається, то переповнення відсутнє. У протилежному випадку в комп'ютері виробляється керуючий сигнал (сигнал переповнення), який викликає або призупинення комп'ютера, або запуск програми усунення переповнення розрядної сітки.

Приклади:

$$\begin{array}{r}
 1) \begin{array}{r} 00.011 \\ +00.110 \\ \hline 01.001 \end{array} \begin{array}{l} =+3_{10} \\ =+6_{10} \\ ? \end{array} \\
 \downarrow \\
 0.1001 =+9_{10}
 \end{array}
 \quad
 \begin{array}{r}
 2) \begin{array}{r} 11.101 \\ +11.010 \\ \hline 1\leftarrow 10.111 \end{array} \begin{array}{l} =-3_{10} \\ =-6_{10} \\ ? \end{array} \\
 \downarrow \\
 1.0111 =-9_{10}
 \end{array}
 \quad
 \begin{array}{r}
 3) \begin{array}{r} 11.101 \\ +00.110 \\ \hline 00.011 \end{array} \begin{array}{l} =-3_{10} \\ =+6_{10} \\ =+3_{10} \end{array}
 \end{array}
 \quad
 \begin{array}{r}
 4) \begin{array}{r} 00.011 \\ +11.010 \\ \hline 11.101 \end{array} \begin{array}{l} =-3_{10} \\ =+6_{10} \\ =-3_{10} \end{array}
 \end{array}$$

1) додавання додатніх чисел  $3+6$ , знакові розряди результату різні, тому результат неправильний. Правильний результат можна визначити, якщо знаковим розрядом вважати тільки один самий старший з двох знакових. Але при цьому кількість значущих розрядів збільшується на один (відбулося переповнення розрядної сітки, далі з таким числом комп'ютер працювати вже не може). Відтворений правильний результат у доповняльному коді буде сприйматися як  $+9$ ;

2) додавання від'єних чисел у доповняльному коді  $(-3)+(-6)$ , знакові розряди результату різні, тому результат неправильний. Відтворений правильний результат у

доповняльному коді буде сприйматися як -9;

3) додавання чисел з різними знаками у доповняльному коді  $(-3)+6$ , знакові розряди результату однакові, тому результат правильний, у доповняльному коді він буде сприйматися як +3;

4) додавання чисел з різними знаками у доповняльному коді  $3+(-6)$ , знакові розряди результату однакові, тому результат правильний, у доповняльному коді він буде сприйматися як -3.

#### **4.5 Алгебраїчне додавання і віднімання чисел, представлених у формі з рухомою комою**

Потрібно підкреслити, що при виконанні будь-яких арифметичних дій над операндами, представленими у формі з рухомою комою, операції, виконувані над мантисами і порядками (або характеристиками) цих чисел, різні. Тому перед початком будь-якої арифметичної процедури кожний з операндів "розчленовується": порядок (характеристика) відокремлюється від мантиси операнда, щоб можна було над ними виконувати необхідні окремі процедури. Після виконання конкретної арифметичної дії й обов'язкової процедури нормалізації результату, його порядок або характеристика і мантиса "упаковуються" у звичайний формат з рухомою комою.

У наведених нижче поясненнях вважається, що „прихований” старший двійковий розряд мантиси перед початком операції вже зайняв своє місце у мантисі, а після виконання операції – він має бути додатково вилученим з формату числа.

У випадку алгебраїчного додавання і віднімання порядки операндів обов'язково повинні бути однаковими. Тому на початку процедури додавання і віднімання чисел провадиться, при необхідності, вирівнювання характеристик операндів. Для цього мантиса операнда з меншою характеристикою зсувається вздовж розрядної сітки праворуч з додаванням одиниці до його характеристики при кожному зсуві на один розряд. Ця процедура продовжується доти, поки характеристики обох операндів не стануть рівними.

Отримана в такий спосіб характеристика, однакова для обох операндів, присвоюється, як попередня, результатів операції. Далі здійснюється додавання або віднімання мантис за правилами, аналогічним для чисел з фіксованою комою, зокрема

мантиса одного з операндів перетвориться в доповняльний код для того, щоб процедуру віднімання звести до додавання. Якщо відповідь вийшла в доповняльному коді, то його перетворюють у прямій, тому що мантиса числа з рухомою комою - це завжди модуль числа. Далі, при необхідності, виконується нормалізація й округлення відповіді. У процесі нормалізації мантиса по-розрядно зсувається вліво або вправо. При зсуві вліво на кожний розряд віднімається одиниця з характеристики, попередньо привласненій відповіді. При кожнім зсуві вправо - до неї додається одиниця. Відзначимо, що зсув мантиси вправо необхідний у тих випадках, коли при додаванні мантис відбулося переповнення розрядної сітки. У процесі цих останніх операцій визначаються остаточні значення характеристики і мантиси відповіді.

Таким чином, операція нормалізації числа складається з перевірки виконання умови  $0,1_2 \leq m < 1$  і зсуву зображення мантиси в ту або іншу сторону з відповідною корекцією характеристики. Зсуви можуть здійснюватися на один і більш розрядів у ліву сторону, або на один розряд у праву сторону в межах розрядної сітки машини.

Порушення нормалізації може бути двох видів: порушення праворуч, тобто коли величина результату більше або дорівнює 1, і ліворуч, коли величина результату виявляється менше  $0,1_2$ .

Треба відзначити, що при реалізації алгоритмів математичних операцій у форматі з рухомою комою щораз, коли здійснюється та або інша процедура, що торкається характеристики операндів або результату, провадиться контроль над переповненням і зникненням порядку, тобто контролюється умова  $0 \leq r \leq r_{max}$ , де  $r_{max} = l + p_{max}$ .

Розглянемо приклад:

Маємо 8-розрядну мантису і 6-розрядний порядок, зміщення дорівнює  $1000_2$ . Складемо 2 числа з мантисами  $m_1 = 0,10100000$ ,  $m_2 = 0,10000000$  і з характеристиками  $r_1 = 001011$ ,  $r_2 = 001010$ . Тому що порядки різні, треба їх вирівнювати:  $r_1 - r_2 = 000001$ , значить треба зсунути  $m_2$  на 1 розряд вправо, а до  $r_2$  додати 1. Після перетворень одержимо:

→  $m_2 = 0,01000000$ ,  $r_2 = 001010 + 000001 = 001011$ . Додаємо мантиси, одержуємо:

$$m = m_1 + m_2 = 0,10100000 + 0,01000000 = 0,11100000, r = 001011.$$

Нормалізація відповіді не потрібна.

#### 4.6 Методи множення двійкових чисел

Стосовно до двійкової системи числення найбільш відомі наступні основні способи виконання операції множення:

1) множення починаючи з молодших розрядів множника і зсувом часткових добутків ліворуч:

$$\begin{array}{r}
 \begin{array}{r}
 \times 1101 = 13_{10} \text{ - множене} \\
 1101 = 13_{10} \text{ - множник} \\
 \hline
 + \quad 1101 \\
 + \quad 0000 \\
 + \quad 1101 \\
 + \quad 1101 \\
 \hline
 10101001 = A9_{16} = 169_{10} \text{ - добуток}
 \end{array}
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{часткові} \\ \text{добутки} \end{array}$$

2) множення починаючи зі старших розрядів множника і зсувом часткових добутків праворуч:

$$\begin{array}{r}
 \begin{array}{r}
 \times 1101 = 13_{10} \text{ - множене} \\
 1101 = 13_{10} \text{ - множник} \\
 \hline
 + \quad 1101 \\
 + \quad 1101 \\
 + \quad 0000 \\
 + \quad 1101 \\
 \hline
 10101001 = A9_{16} = 169_{10} \text{ - добуток}
 \end{array}
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{часткові} \\ \text{добутки} \end{array}$$

В обох випадках операція множення складається з ряду послідовних операцій додавання часткових добутків. Операціями додавання керують розряди множника: якщо в якомусь розряді множника знаходиться одиниця, то до суми часткових добутків додається множене з наступним його зсувом (вліво або вправо), якщо в розряді множника - нуль, то замість множеного додаються нулі, але зсув множеного все одно відбувається.

Таким чином, крім операції додавання чисел для одержання добутку необхідна операція зсуву числа.

#### 4.7 Множення чисел, представлених у форматі з фіксованою комою

Розглянемо один з можливих варіантів алгоритму множення, коли операнди представлені в прямому коді. Перед виконанням самої процедури множення за

звичайними арифметичними правилами множення визначається і запам'ятовується знак добутку. Далі обидва операнда представляються в прямому коді і виконується сама процедура множення одним із двох раніше описаних методів з обов'язковим контролем переповнення розрядної сітки. Якщо знак добутку від'ємний, то відповідь, при необхідності, представляється в доповняльному коді.

Множення чисел представлених у формі з фіксованою комою можна організувати не тільки на двійковому суматорі прямого коду, але і на двійкових суматорах доповняльного коду. У цьому випадку добуток доповняльних кодів співмножників дорівнює доповняльному кодові відповідно тільки у випадку додатнього множника. Якщо ж множник від'ємний, то провадиться корекція результату: при множенні на старший (знаковий) розряд множника при його одиничному значенні замість додавання множеного здійснюється віднімання множеного.

Розглянемо приклади множення від'ємних чисел у доповняльних кодах:

$$\begin{array}{r}
 \times 1.0011 = -13_{10} \text{ - множене} \\
 \quad 1.0011 = -13_{10} \text{ - множник} \\
 \hline
 + 11111.0011 \quad \left. \vphantom{\begin{array}{r} + \\ + \\ + \end{array}} \right\} \text{ часткові від'ємні добутки} \\
 + 1111.0011 \\
 + \quad 0.0000 \\
 + \quad 0.0000 \quad \left. \vphantom{\begin{array}{r} + \\ + \end{array}} \right\} \text{ часткові нульові добутки} \\
 \hline
 1.11011001 \\
 + 0.1101 \quad \text{- корекція (+13}_{10}\text{, віднімання множеного)} \\
 \hline
 0.10101001 = +A9_{16} = +169_{10} \text{ - добуток}
 \end{array}$$

При множенні на значущі розряди утворюються часткові добутки, які сумуються з врахуванням їхніх знаків. При цьому знак розповсюджується ліворуч на всі розряди, які приймають участь у сумуванні. Особливо на це слід звернути увагу коли множене – від'ємне.

#### 4.8 Множення чисел, представлених у формі з рухомою комою

При операції множення чисел, представлених у формі з рухомою комою їхні мантиси перемножуються, як числа з фіксованою комою, а порядки додаються. В обох випадках перевірка на переповнення обов'язкова. Але, тому що реально додаються характеристики чисел, то множення виконується за наступною формулою:

$$A_1 \times A_2 = m_1 2^{r_1} \times m_2 2^{r_2} = (m_1 \times m_2) 2^{r_1+r_2-l}, \text{ де } r_1 + r_2 - l \text{ - характеристика результату.}$$



Знак результату визначається звичайним чином. Якщо відповідь вийшла ненормалізованою, то виконуються процедури нормалізації та округлення відповіді.

Нагадаємо, що при реалізації алгоритмів математичних операцій у форматі з рухомою комою щораз, коли здійснюється та або інша процедура, що торкається характеристики операндів або результату, провадиться контроль над переповненням і зникненням порядку, тобто контролюється умова  $0 \leq r \leq r_{max}$ , де  $r_{max} = l + p_{max}$ .

Розглянемо приклад:

помножимо числа з мантисами  $m_1 = 0,101$ ,  $m_2 = 0,1$  і характеристиками  $r_1 = 1011$ ,  $r_2 = 1010$ ,  $l = 1000$ ,  $p_1 = 3$ ,  $p_2 = 2$ ,  $p = p_1 + p_2 = 5$ ,  $r = r_1 + r_2 - l = 001011 + 001010 - 001000 = 001101 = 13_{10}$ ,  $m = m_1 \times m_2 = 0,01010$ , необхідно нормалізувати відповідь. Для цього в даному випадку зсуваємо мантису вліво на 1 розряд і віднімаємо 1 з  $p$ , одержуємо  $p = 4$ . У підсумку маємо  $m = 0,1010_2$ ,  $p = 4_{10} = 0100_2$ ,  $r = 12_{10} = 1100_2$ .

#### 4.9 Прискорення операції множення

За часом виконання операція множення відноситься до довгих операцій. Витрати часу на множення двох чисел у прямому коді можна оцінити наступною формулою (для випадку послідовного аналізу розрядів множника):

$$t_{mn} = \sum_{i=1}^n (t_{зсв} + p_i t_c),$$

де  $t_{зсв}$  - час виконання зсуву числа на один розряд;  $t_c$  - час сумування на суматорі;  $p_i$  - імовірність появи одиниці в розрядах множника;  $n$  - кількість розрядів множника.

Існують кілька методів прискорення процедури множення:

аналіз двох розрядів множника одночасно,

аналіз довільної кількості розрядів множника,

множення в системі числення з підставою  $q = 2^k i$ ,

матричні методи множення,

інші.

#### 4.10 Ділення двійкових чисел, представлених у формі з фіксованою комою.

Ділення двійкових чисел багато в чому аналогічно діленню десяткових чисел.

В універсальних обчислювальних машинах, як правило, реалізується "шкільний" алгоритм ділення чисел. "Шкільний" алгоритм ділення полягає в тому, що дільник на кожному кроці віднімається з діленого стільки разів (починаючи зі старших розрядів), скільки це можливо для одержання найменшого додатнього залишку. Тоді в черговий розряд частки записується цифра, рівна числу дільників, що утримуються в діленому на даному кроці (формування частки починається з її старших розрядів). Інакше кажучи, при діленні операцію віднімання повторюють доти, поки зменшуване не стане менше від'ємника. Число цих повторень показує, скільки разів від'ємник укладається в зменшуваному.

Наприклад:

розділимо число 35 на 7 :

1)  $35 - 7 = 28$ , 2)  $28 - 7 = 21$ , 3)  $21 - 7 = 14$ , 4)  $14 - 7 = 7$ , 5)  $7 - 7 = 0$ .

Відповідь дорівнює 5, тому що процедура віднімання була повторена 5 разів.

Розглянемо ще один приклад:

розділимо  $204_{10}$  на  $12_{10}$ , тобто  $11001100_2 : 1100_2$

ділене -	1 1 0 0 1 1 0 0	1 1 0 0	- дільник
	- 1 1 0 0	1 0 0 0 1	- частка
залишок з наступним розрядом -	0 0 0 0 1		
	- 0		
залишок з наступним розрядом -	1 1		
	- 0		
залишок з наступним розрядом -	1 1 0		
	- 0		
залишок з наступним розрядом -	1 1 0 0		
	- 1 1 0 0		
залишок -	0		

Двійкове ділення, як і десяткове ділення, починається з аналізу діленого (11001100) і дільника (1100). Відразу ж виявляється, що дільник укладається в 1100, а тому записується 1 у старший розряд поля частки. Дільник перемножується з 1 і віднімається зі старших розрядів діленого (1100), різниця дорівнює 0. Об'єднується 0 залишку зі значенням наступного розряду діленого, рівного 1. Оскільки дільник (1100) 0 разів укладається в 01, записуємо 0 у наступний за старшим розряд поля частки, а

число 01 об'єднується з наступним розрядом діленого і т.д. доти, поки ділене не виявляється вичерпаним.

Звичайно комп'ютер не може робити припущення щодо того, скільки разів дільник укладається в тому або іншому числі, тому весь процес ділення зводиться до операцій віднімання і зсуву. Продемонструємо це на тому ж прикладі, але спочатку дільник (1100) представимо в доповняльному коді, що дозволить обмежитися додаванням у всіх випадках, коли потрібно виконувати додавання або віднімання:  $1100_{ПК} = 1.0100_{ДК}$ . Частка формується в деякому регістрі  $Ч$ , незаповнені розряди якого будемо позначати через  $X$ .

Починаємо віднімати дільник з діленого. Якщо залишок виходить додатнім, то в розряд частки записується 1, у противному випадку - 0.

$$\begin{array}{r} 0.11001100 \text{ - ділене } (204_{10}) \\ + 1.01000000 \text{ - дільник } (12_{10}) \\ \hline 0.00001100 \text{ - перший залишок (додатній)} \end{array}$$

Перший (старший) біт частки дорівнює 1, тому що залишок вийшов додатнім:  $Ч = 1XXXX$ . Далі зсувається перший залишок на один розряд вліво і від нього віднімається дільник:

$$\begin{array}{r} + 0.00011000 \text{ - зсунутий перший залишок} \\ + 1.01000000 \text{ - дільник } (12_{10}) \\ \hline 1.01011000 \text{ - другий залишок (від'ємний)} \end{array}$$

Залишок від'ємний, тому в наступний розряд частки записується 0,  $Ч = 10XXX$ . Крім того необхідно біти дільника повернути назад першому залишкові, тобто додати дільник (у прямому коді) і другий залишок:

$$\begin{array}{r} + 1.01011000 \text{ - другий залишок (від'ємний)} \\ + 0.11000000 \text{ - дільник } (12_{10}) \\ \hline 0.00011000 \text{ - зсунутий перший залишок} \end{array}$$

Далі ще раз зсувається зсунутий перший залишок на один розряд вліво і віднімається від нього дільник:

$$\begin{array}{r} + 0.00110000 \text{ - двічі зсунутий перший залишок} \\ + 1.01000000 \text{ - дільник } (12_{10}) \\ \hline 1.01110000 \text{ - третій залишок (від'ємний)} \end{array}$$

Третій залишок від'ємний, тому наступний (третій) розряд частки дорівнює 0,

$C = 100XX$ . Тому повертаємо дільник третьому залишкові,

$$\begin{array}{r} 1.0\ 11\ 10\ 000 \\ + 0.1\ 1000000 \\ \hline 0.0\ 01\ 10000 \end{array}$$

- третій залишок (від'ємний)  
- дільник ( $12_{10}$ )  
- відновлений двічі зсунутий перший залишок

Зсуваємо двічі зсунутий перший залишок на один розряд вліво і віднімаємо дільник:

$$\begin{array}{r} 0.0\ 11\ 00000 \\ + 1.0\ 1000000 \\ \hline 1.1\ 01\ 00000 \end{array}$$

- тричі зсунутий перший залишок  
- дільник ( $12_{10}$ )  
- четвертий залишок

Четвертий залишок знову від'ємний, тому  $C = 1000X$ . Додаємо дільник до четвертого залишку, результат зсуваємо на один розряд уліво, а потім знову віднімаємо дільник:

$$\begin{array}{r} 0.1\ 1000000 \\ + 1.0\ 1000000 \\ \hline 0.0\ 0000000 \end{array}$$

- чотири рази зсунутий перший залишок  
- дільник ( $12_{10}$ )  
- п'ятий залишок (рівний 0)

Залишок додатній, значить  $C = 10001 = 17_{10}$  - це і є відповідь.

Такий метод ділення називається діленням з відновленням залишку.

Ділення чисел, представлених у формі з фіксованою комою можна також здійснити на двійкових суматорах оберненого і доповняльного коду.

Перед виконанням самої процедури ділення чисел у форматі з фіксованою комою визначається і запам'ятовується знак частки. Далі обидва операнда представляються в прямому коді, а дільник ще й у доповняльному для того, щоб віднімання дільника замінити додаванням, і виконується сама процедура ділення за описаним вище методом з обов'язковим контролем переповнення розрядної сітки. Якщо знак частки від'ємний, то відповідь, при необхідності, представляється в доповняльному коді.

Метод ділення без відновлення залишку. Як уже відзначалося, основою виконання ділення є операція віднімання з метою одержання залишку, знак якого визначає цифру частки. Алгоритм виконання ділення має такий вигляд:

$$X - Y = a_0,$$

де  $a_0$  - це перший залишок,  $X$  і  $Y$  - відповідно ділене і дільник, вирівняні за

старшими розрядами. Якщо  $a_i \geq 0$ , то черговий розряд частки  $Ч_i = 1$ , якщо  $a_i < 0$ , то  $Ч_i = 0$ . Для визначення наступної цифри частки необхідно виконати наступні дії: при  $a_i \geq 0$  треба виконати операцію  $2a_i - Y = a_{i+1}$ , а при  $a_i < 0$  треба виконати  $2a_i + Y = a_{i+1}$ . Як видно в даному випадку знак залишку визначає не тільки чергову цифру частки, але і характер наступної процедури: додавання дільника до зсунутого залишку, якщо цей залишок менше 0, і віднімання дільника від зсунутого залишку, якщо залишок більше або дорівнює 0. Цей метод ділення одержав назву ділення без відновлення залишку.

Розглянемо приклад:

розділимо  $35_{10}$  на  $5_{10}$ , тобто  $0.100011_2$  на  $0.101_2$

$$\begin{array}{r}
 0.100011 - \text{ділене} \\
 + 1.011000 - \text{віднімання дільника } (5_{10}) \\
 \hline
 1.111011 - a_0 < 0, Ч = 0 \\
 + 1.110110 - 2a_0 \\
 + 0.101000 - \text{додавання дільника } (5_{10}) \\
 \hline
 0.011110 - a_1 > 0, Ч = 01 \\
 + 0.111100 - 2a_1 \\
 + 1.011000 - \text{віднімання дільника } (5_{10}) \\
 \hline
 0.010100 - a_2 > 0, Ч = 011 \\
 + 0.101000 - 2a_2 \\
 + 1.011000 - \text{віднімання дільника } (5_{10}) \\
 \hline
 0.000000 - a_3 = 0, Ч = 0111 = 7_{10}
 \end{array}$$

#### 4.11 Ділення двійкових чисел, представлених у форматі з рухомою комою.

При операції ділення чисел, представлених у формі з рухомою комою їхні мантиси діляться, як числа з фіксованою комою, а порядки віднімаються. В обох випадках перевірка на переповнення обов'язкова. Але, тому що реально віднімаються характеристики чисел, то ділення виконується за наступною формулою:

$$A_1 : A_2 = m_1 2^{r_1} : m_2 2^{r_2} = (m_1 : m_2) 2^{r_1 - r_2 + l}$$

Знак результату визначається звичайним чином. Якщо результат вийшов ненормалізований, то виконується процедура нормалізації й округлення результату.

Оскільки мантиси операндів нормалізовані, то можливі випадки, коли  $|m_A| \geq |m_B|$  або  $|m_A| < |m_B|$ . У першому випадку перед початком ділення треба відняти дільник з діленого й до цілої частини мантиси частки записати 1. Потім продовжувати ділити числа звичайним методом (визначити дробову частину мантиси частки). Після

одержання частки очевидно, що вона не буде нормалізованою. Значить треба нормалізувати частку, тобто в даному випадку зрушити його на 1 розряд вправо, а до порядку частки додати 1.

Нагадаємо, що при реалізації алгоритмів математичних операцій у форматі з рухомої комою щораз, коли здійснюється та або інша процедура, що торкається характеристики операндів або результату, провадиться контроль над переповненням і зникненням порядку, тобто контролюється умова  $0 \leq r \leq r_{max}$ , де  $r_{max} = l + p_{max}$ .

Розглянемо приклад:

$A = 1010 = 0.1010_2$ ,  $p = 4$ ,  $m = 0.1010$ ,  $B = 2_{10} = 0.10_2$ ,  $p = 2$ .  $p = p_A - p_B = 2$ ,  $m = 0.1$ ,  $(-m)_{DK} = 1.1000$ ,  $|m_A| \geq |m_B|$ . При першому відніманні мантис записуємо в цілу частину частки 1:

$$\begin{array}{r} 0.1010 - \text{ділене} \\ + 1.1000 - \text{дільник } (2_{10}) \\ \hline 0.0010 - a_0 \text{ перший залишок} \end{array}$$

$Ч = 1,XX$ , далі, будемо ділити методом ділення без відновлення залишку.

$$\begin{array}{r} 0.0100 - 2a_0 \text{ зсунутий перший залишок} \\ + 1.1000 - \text{дільник } (2_{10}) \text{ віднімається} \\ \hline 1.1100 - a_1 < 0, \text{ другий залишок від'ємний, } Ч = 1,0 \\ + 1.1000 - 2a_1 \text{ зсунутий другий залишок} \\ + 0.1000 - \text{дільник } (2_{10}) \text{ додається} \\ \hline 0.0000 - a_2 = 0, \text{ третій залишок додатній } Ч = 1,01 \end{array}$$

зсуваємо вправо  $Ч = 0,101$ ,  $p = p + 1 = 3$ .

#### 4.12 Оцінка точності виконання арифметичних операцій

Як уже неодноразово відзначалося, для представлення чисел у будь-якій формі в комп'ютері приділяється кінцева довжина розрядної сітки. Тому числа, у загальному випадку, через неминучі процедури округлення, представляються з деякою похибкою. У зв'язку з цим обчислення в комп'ютері виконуються з похибкою, що у деяких випадках може істотно накопичуватися.

Нехай величини  $A$  і  $B$  задані з абсолютними похибками  $\Delta A$  і  $\Delta B$ :  $A = [A] \pm \Delta A$ ,  $B = [B] \pm \Delta B$  де  $[A]$  і  $[B]$  - машинне представлення чисел  $A$  і  $B$ . Тоді, як нам уже відомо, відносні похибки представлення цих чисел будуть рівні:

$$\delta A = \Delta A / |A| \text{ і } \delta B = \Delta B / |B|.$$

Можна показати, що відносна похибка алгебраїчної суми буде дорівнює:

$$\delta(A \pm B) = \frac{[A]}{[A] \pm [B]} \delta A \pm \frac{[B]}{[A] \pm [B]} \delta B,$$

відносна похибка добутку -  $\delta AB = |\delta A| + |\delta B|$ ,

а відносна похибка частки -  $\delta \frac{A}{B} = |\delta A| + |\delta B|$ .

З приведених співвідношень видно, що операція множення і ділення незначно збільшують відносну похибку, а віднімання майже рівних чисел може сильно її збільшити. Дійсно, якщо прийняти, що  $\delta = \delta A \approx \delta B$ , а  $[A]$  мало відрізняється від  $[B]$ , то

$$\delta(A - B) \approx \delta \cdot \frac{2[A]}{[A] - [B]}$$

У зв'язку з тим, що величина  $(A - B)$  у даному випадку мала, а  $\delta \neq 0$ , то  $\delta(A - B)$  може бути досить великою величиною.

Через неминучі похибки машинних обчислень може спостерігатися порушення деяких основних законів математики. Наприклад, може не виконуватися точно закон асоціативності для множення. Закон дистрибутивності, що зв'язує операції множення і додавання, може значно порушуватися, тобто не буде строго виконуватися рівність:

$$A(B+C) = AB+AC.$$

Тому потрібно дуже ретельно оцінювати конкретні умови обчислень і, при необхідності, переходити до представлення чисел з подвійною точністю і приймати різні інші міри, щоб у якійсь мірі зменшувати вплив накопиченої похибки на результат обчислень.

### 4.13 Похибка округлення

Тому що в комп'ютері приходиться прибігати до округлення чисел (відкидання зайвих розрядів), то неминучі похибки округлення. Якщо для представлення у системі числення з основою  $q$   $p$ -розрядної мантиси  $A$  використовується тільки  $n$  її старших розрядів ( $n < p$ ), то зображення мантиси можна розбити на дві частин:

$n$ -розрядне поле  $[msb]$  – старші розряди, число  $A_m$ , яке записується за допомогою цих розрядів має величину  $A_m = [msb]q^{-n}$ ;

$(p-n)$ -розрядне поле  $[lsb]$  - молодші розряди, "хвіст" числа, що не потрапив до розрядної сітки, число  $A_l$ , яке записується за допомогою цих розрядів має величину  $A_l = [lsb]q^{-p} < (1 \times q^{-n}) = q^{-n}$ .

$$A = A_m + A_l = [msb]q^{-n} + [lsb]q^{-p}.$$

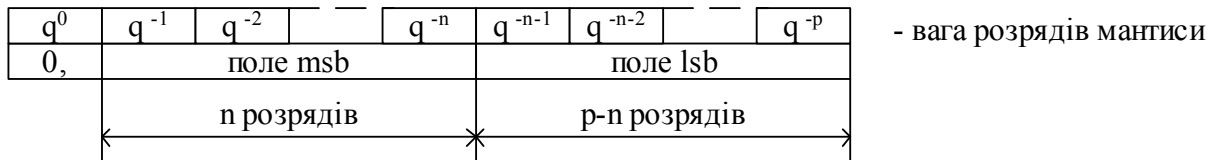


Рис. 4.2. Заокруглення мантиси

У залежності від того, як враховується величина  $[lsb]$  у машинному зображенні, існує кілька способів округлення.

1. Відкидання  $[lsb]$ . При цьому відносна похибка дорівнює  $\delta_{окр} = A_l/A$ .

Оскільки  $q^{-l} \leq |A| < 1$ ; а  $0 \leq |A_l| < q^{-n}$ , тому  $\delta_{окр} < q^{-n}/q^{-l} = q^{-n+l}$ , тобто відносна похибка округлення не залежить від величини самого числа, а залежить тільки від кількості розрядів у машині для будь-якої системи числення.

2. Симетричне округлення. При цьому провадиться аналіз величини  $A_l = [lsb]q^{-p}$ .

$$\text{Тоді } [A] = \begin{cases} A_m, & \text{якщо } A_l < q^{-n}/2; \\ A_m + q^{-n}, & \text{якщо } A_l \geq q^{-n}/2. \end{cases}$$

За умови  $A_l \geq q^{-n}/2$  (частина, яку відкидаємо, більша або дорівнює половині молодшого розряду з тих, що залишаються) провадиться додавання одиниці до молодшого розряду мантиси. Максимально можливе значення модуля абсолютної похибки дорівнює  $q^{-n}/2$ , тобто помилка не перевищує половини одиниці молодшого розряду мантиси.

$$\text{Максимальна відносна похибка дорівнює } \delta_{окр} \leq q^{-n}/2q^{-l} = 0,5q^{-n+l}.$$

Спосіб симетричного округлення найбільше часто застосовують на практиці.

Для двійкової системи числення (при  $q = 2$ ) спосіб симетричного округлення спрощується. Для округлення аналізується інформація, що міститься в  $(n+1)$ -му розряді. Якщо в  $(n+1)$ -му розряді міститься 1, то перед відкиданням зайвих розрядів до  $n$ -го розряду додається 1; якщо ж там нуль, розряди правіше  $n$ -го просто відкидаються.



## 5 ВИКОНАННЯ ОПЕРАЦІЙ НАД ДЕСЯТКОВИМИ ЧИСЛАМИ

### 5.1 Представлення десяткових чисел у Д-кодах

Операції над десятковими числами (десятькова арифметика) часто включається до складу основних команд універсальних комп'ютерів. Крім того, десяткова арифметика широко реалізується в електронних калькуляторах і персональних комп'ютерах. Тому крім загальної інформації про можливості представлення десяткових чисел розроблювачеві необхідно знати і алгоритми виконання арифметичних операцій.

Двійково-десятьковий код (Д-код) десяткового числа, як уже відзначалося, це таке його представлення, у якому кожна десяткова цифра зображується чотирма двійковими розрядами (тетрадою з двійкових символів):

$$A = \{a_{3,n} a_{2,n} a_{1,n} a_{0,n}\}_n \{a_{3,n-1} a_{2,n-1} a_{1,n-1} a_{0,n-1}\}_{n-1} \dots \{a_{3,0} a_{2,0} a_{1,0} a_{0,0}\}_0,$$

де  $a_{i,j}$  - двійкові розряди тетради,  $i$  - номер розряду усередині тетради,  $j$  - номер самої тетради.

Для однозначності переведення чисел у Д-код і назад бажано, щоб розряди тетрад мали визначену вагу. Тоді значення десяткової цифри  $a_i$  відповідає виразу:  $a_i = a_3d_3 + a_2d_2 + a_1d_1 + a_0d_0$ , де  $d_i$  - вага розряду тетради. Максимальне припустиме число в тетраді - 9. Якщо виникає число 10 і більше, то одиниця переходить у наступну старшу тетраду. Існують різні Д-коди. Ми будемо розглядати Д-код, вагу розрядів тетради якого наступна: 8, 4, 2, 1. Цей код називається також кодом  $D_1$  прямого заміщення або кодом 8421. Але є Д-коди з іншими вагами розрядів тетрадами: 5, 1, 2, 1; 2, 4, 2, 1 і т.д. Загальна кількість комбінацій, які можуть створювати 4 двійкових розрядів - 16. Комбінації, які відповідають десятковим цифрам (їх 10), в Д-кодах дозволені. Всі інші комбінації - заборонені. Наявність дозволених і заборонених комбінацій - дуже важлива властивість Д-кодів. Воно відрізняє їх від звичайних позиційних систем числення, у яких усі комбінації - дозволені.

У розглянутому Д-коді прямого заміщення (8421) дозволені комбінації відповідають двійковим еквівалентам десяткових цифр із вагами розрядів, рівних степеням основи 2. Цей Д-код найчастіше використовується в різних комп'ютерах.

Нижче наведені приклади запису десяткових цифр у різних двійково-десяткових кодах (таблиця 5.), а також приклади запису деяких десяткових чисел у кодї 8421 прямого заміщення (Таблиця 5.).

**Таблиця 5.1. Двійково-десяткові коди**

Десяткові цифри	Код 8421	Код 8421+3	Код 2421	Код 5121
0	0000	0011	0000	0000
1	0001	0100	0001	0001
2	0010	0101	0010	0010
3	0011	0110	0011	0011
4	0100	0111	0100	0111
5	0101	1000	1011	1000
6	0110	1001	1100	1100
7	0111	1010	1101	1101
8	1000	1011	1110	1110
9	1001	1100	1111	1111

Як видно з таблиці 5.1 коди 8421+3, 2421, 5121 „обернено” симетричні відносно горизонтальної лінії, яка проходить посередині таблиці між кодами цифр 4 і 5. Цим полегшується утворення обернених кодів і, тим самим, полегшується виконання операції віднімання.

**Таблиця 5.2. Двійково-десятковий код 8421**

Десяткові числа	Код $D_1$	Десяткові числа	Код $D_1$
10	0001 0000	20	0010 0000
11	0001 0001	21	0010 0001
12	0001 0010	22	0010 0010
13	0001 0011	33	0011 0011
14	0001 0100	34	0011 0100
15	0001 0101	45	0100 0101
16	0001 0110	56	0101 0110
17	0001 0111	67	0110 0111
18	0001 1000	78	0111 1000
19	0001 1001	89	1000 1001

## 5.2 Формальні правила по-розрядного додавання в Д-кодах

Для визначення формальних правил по-розрядного додавання чисел,

представлених у Д-кодi, розглянемо тi особливостi, що властивi цим кодам.

1. Наявнiсть дозволених i заборонених комбiнацiй. Поява забороненої комбiнацiї при виконаннi якихось дiй над числами свiдчить про виникнення помилки або ж про необхіднiсть увести коректування результату.

2. При додаваннi тетрад виникає вихiдний мiжтетрадний перенос, тобто перенос 1 до старшої тетради, замість десяткового мiжрозрядного переносу.

Уведемо правила додавання Д-кодiв. При додаваннi чисел у Д-кодi можуть виникнути наступнi випадки.

1. Якщо при додаваннi двох тетрад у даному розрядi числа утвориться сума менше  $10_{10}$ , корекцiя результату не потрібна . Наприклад: додати двi тетради  $a = 0100_2$ ,  $b = 0101_2$  за умови, що немає вхiдного мiжтетрадного переносу з молодшої тетради:

$$c = 0100_2 + 0101_2 = 1001_2 = 9_{10}. \text{ Результат менше } 10_{10}.$$

2. Свiдченням того, що результат сумування неправильний, є або поява забороненої комбiнацiї, якщо  $15 \geq c \geq 10$ , або поява вихiдного мiжтетрадного переносу  $p_o = 1$ , коли  $c \geq 16$ , що перевищує умову виникнення десяткового переносу на 6. Отже, потрібна корекцiя результату в даній тетрадi введенням виправлення, рiвного  $+0110_2 = 6_{10}$ .

Наприклад:

1) Скласти  $a = 0101_2$ ,  $b = 1001_2$ , при наявностi переносу з молодшої тетради  $p_i = 1$ :

$c_{заб} = a + b + p_i = 1111_2$  - заборонена комбiнацiя i потрібне виправлення, корекцiя тетради  $c = c_{заб} + 0110_2 = 1111 + 0110 = [1]0101_2$ , тобто, результат у даній тетрадi дорiвнює  $c = 0101_2$  i утворився вихiдний мiжтетрадний перенос  $[1]$  ( $p_o = 1$ ) до старшої тетради.

2) Додати 2 тетради:  $a = 0111_2$ ,  $b = 1001_2$ ,  $p_i = 1$ .

$c_{заб} = a + b + p_i = [1]0001_2$ , з'явився вихiдний мiжтетрадний перенос  $p_o = 1$  до старшої тетради i потрібна корекцiя результату:  $c = c_{заб} + 0110_2 = 0001_2 + 0110_2 = 0111_2$ .

3) Додати  $a = 219_{10} = 0010 \quad 0111 \quad 1001_d$ ,  $b = 581_{10} = 0101 \quad 1000 \quad 0001_d$

$$\begin{array}{r}
 219 \\
 + 581 \\
 \hline
 800
 \end{array}
 \begin{array}{r}
 + 0010 \ 0001 \ 1001 \\
 + 0101 \ 1000 \ 0001 \\
 \hline
 0111 \ 1001 \ 1010 \\
 \phantom{0111} + 0110 \\
 \hline
 0111 + 1010 \leftarrow 0000 \\
 \phantom{0111} + 0110 \\
 \hline
 1000 \leftarrow 0000 \ 0000 \\
 8 \quad 0 \quad 0
 \end{array}
 \begin{array}{l}
 - \text{незкоректований результат} \\
 - \text{корекція молодшої тетради} \\
 - \text{корекція наступної тетради} \\
 - \text{відкоректований результат}
 \end{array}$$

Тут стрілка вказує передачу 1 десятикового переносу.

Як видно, корекція тетрад здійснюється послідовно, починаючи з молодшої. В наслідок корекції може виникати перенос до наступної тетради, при цьому вона може набувати забороненого значення і її також необхідно буде коректувати.

Результат:  $c = 1000 \ 0000 \ 0000_{\text{д}} = 800_{10}$ .

Суматор однієї тетради, призначений для формування суматорів чисел, представлених у Д-коді прямого заміщення, може бути реалізований у такий спосіб (рис. 5.1).

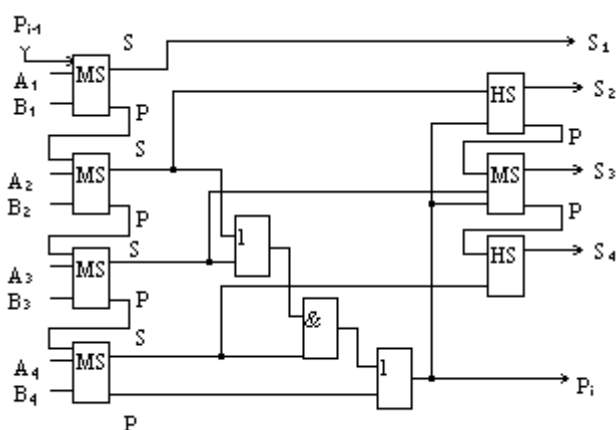


Рис. 5.1. Суматор однієї тетради

### 5.3 Представлення від'ємних чисел у Д-кодах

Представлення Д-коду в розрядній сітці машини може здійснюватися у формі або з фіксованою, або з рухомою комою. При цьому від'ємні числа можуть представлятися в прямому, оберненому або доповняльному коді.

Для розглянутого коду 8421 не можна одержати обернений або доповняльний код простим інвертуванням, тому що інвертування набору тетрад означає одержання доповнення до  $2^4 - 1 = 15$ , при цьому можуть виникати заборонені коди. Отже,

необхідно їх усувати. Один з використовуваних при цьому прийомів полягає в тому, що до усіх цифрових тетрад числа в кодї 8421 додається 0110 і після цього провадиться інвертування набору. Отримане зображення являє собою обернений код числа. А доповняльний код утворюється, як звичайно, додаванням 1 до молодшого розряду молодшої тетради. Розглянемо приклади:

1) представити число  $A_{ПК} = -256_{10} = 1.0010\ 0101\ 0110_{Д}$  в оберненому кодї 8421:

$$\begin{array}{r} 0010\ 0101\ 0110 \\ +\ 0110\ 0110\ 0110 \\ \hline 1000\ 1011\ 1100 \end{array} \quad \text{- до усіх тетрад додали 0110,}$$

після інвертування маємо  $A_{ОК} = 1.0111\ 0100\ 0011_{Д} = -743_{10}$ .

2) представити число  $A_{ПК} = -398_{10}$  у доповняльному кодї 8421:

$$\begin{array}{r} 0011\ 1001\ 1000 \\ +\ 0110\ 0110\ 0110 \\ \hline 1001\ 1111\ 1110 \end{array} \quad \text{- до усіх тетрад додали 0110,}$$

після інвертування маємо  $A_{ОК} = 1.0110\ 0000\ 0001_{Д} = -601_{10}$ .

Додаємо +0001 до молодшої тетради і отримуємо результат  $A_{ДК} = 1.0110\ 0000\ 0010_{Д} = -602_{10}$ .

#### 5.4 Виконання операцій додавання і віднімання в Д-кодах

Операції додавання і віднімання в Д-кодах виконуються над операндами за формальними правилами десяткової арифметики, описаними раніше:

1. При додаванні двох додатніх чисел переведення в доповняльний код виключається. У випадку появи по-тетрадного переносу або результату більшого 9 провадиться корекція результату додаванням 6.

2. Додавання від'ємних чисел виконується аналогічно додаванню додатніх з тою лише різницею, що результатіві присвоюється від'ємний знак.

3. При відніманні чисел здійснюється попереднє переведення в доповняльний код, а потім додавання чисел.

Розглянемо приклад, коли одне число від'ємне, а інше додатнє:

1) Додати два числа в кодї 8421:

$$A_{ПК} = -825_{10} = -1000\ 0010\ 0101_{Д}, \quad B_{ПК} = +946_{10} = 1001\ 0100\ 0110_{Д},$$

$$A_{ДК} = -175_{10} = 1.0001\ 0111\ 0101_{Д}, \quad B_{ДК} = +946_{10} = 0\ 1001\ 0100\ 0110_{Д},$$

$$\begin{array}{r} 1\ 0001\ 0111\ 0101 \\ +\ 0\ 1001\ 0100\ 0110 \\ \hline 1\ 1010\ 1011\ 1011 \end{array}, \text{ результат додавання містить заборонені комбінації}$$

тетрад, які необхідно коректувати:

$$\begin{array}{r} 1\ 1010\ 1011\ 1011 \\ +\ 0\ 0110\ 0110\ 0110 \\ \hline 0\ 0001\ 0010\ 0001 \end{array}.$$

Результат додатній  $C_{ДК} = C_{ПК} = 1.0001\ 0010\ 0001_{Д} = 121_{10}$ .

### 5.5 Множення чисел у Д-кодах

Виконання операцій множення в Д-кодах принципово провадиться за класичною схемою. Множення чисел зводиться до послідовного сумування часткових добутків, отриманих при множенні множеного на чергову цифру множника. Тому що кожна цифра множника представляється тетрадою, те множення супроводжується розшифровкою значення чергової тетради множника і зсувом на 4 розряди відразу. Розшифровку можна здійснити різними способами. Найпростішим прикладом є послідовне віднімання 1 зі значення тетради до одержання 0 і відповідно додавання множеного. Треба враховувати обов'язково проміжні переповнення.

### 5.6 Ділення чисел у Д-кодах

Ділення десяткових чисел у Д-кодах виконується методом послідовного віднімання дільника з діленого на першому кроці і з залишків - на наступних кроках. Віднімання на кожному кроці провадиться доти, поки не вийде від'ємний залишок. Щораз при одержанні додатнього залишку додається 1 у спеціальний лічильник, де накопичується чергова цифра частки. Потім здійснюється зсув на 4 двійкових розряди і додавання дільника доти, поки не вийде додатній залишок. Кількість додавань (без останнього) є доповненням відповідної цифри частки до 9, що заноситься в лічильник чергової цифри частки.

Таким чином, процес ділення складається з послідовного ряду циклів додавання і віднімання, що чергуються, і зсувів. Знак частки утворюється як логічна сума за модулем 2 знаків чисел.

## 5.7 Переведення чисел з Д-коду в двійковий і з двійкового в Д-код

Послідовність переведення чисел з Д-коду в двійковий збігається з послідовністю переведення десяткового числа до шістнадцяткової системи числення, коли дії виконуються у шістнадцятковій системі числення.

Число  $X_D = (T_n T_{n-1} \dots T_1 T_0)$ , записане за допомогою коду 8421, до шістнадцяткової системи числення переводиться у відповідності до формули:

$$X = T_n A^n + T_{n-1} A^{n-1} + \dots + T_1 A^1 + T_0 A^0, \text{ де } T_i - \text{тетради};$$

$$A_{16} = 10_{10} \text{ (десять у десятковій системі числення).}$$

Усі операції виконуються у шістнадцятковій системі числення, результат виходить так само у шістнадцятковій системі числення. Після чого перехід до двійкової системи числення здійснюється заміною кожної шістнадцяткової цифри на чотири двійкових розряди.

Переведення з двійкової системи в код 8421 може здійснюватися різними способами. Зокрема, для ряду послідовних операцій над двійковим зображенням числа може бути використана процедура ділення вихідного числа на  $1010_2$ , як цілих двійкових чисел. Десяткові цифри утворюються послідовно одна за одною, починаючи зі старшого десяткового розряду. При дробових числах ця операція видозмінюється таким чином, щоб при множенні на число  $1010_2$  можна було одержати відповідні цифри десяткових дробів.

Є більш простий спосіб переведення - це зсув ліворуч двійкового числа стільки разів, скільки розрядів у двійковому числі. Необхідно передбачити корекцію в тих тетрадах, значення яких перевищить  $1010_2$  або з яких відбудеться міжтетрадний перенос.

Наприклад,  $110001_2 = 49_{10}$  переведемо в код 8421 (таблиця 5.).

**Таблиця 5.3. Переведення з двійкової системи в код Д<sub>1</sub>**

									110001
1) Зсув								1	10001
2) Зсув						1	1		0001
3) Зсув					1	1	0		001
4) Зсув і корекція				1	1	0	0		01

+0110				0	1	1	0	
Результат після корекції			1	0	0	1	0	01
5) Зсув		1	0	0	1	0	0	1
6) Зсув	1	0	0	1	0	0	1	

Результат  $01001001_D = 49_{10}$  .

Алгоритми переведення чисел із двійкової системи числення в Д-код і навпаки можуть бути реалізовані схемними або програмними способами.



## 6 КОНТРОЛЬ РОБОТИ ЦИФРОВОГО АВТОМАТА

### 6.1 Основні поняття теорії кодування

У цифровому автоматі можуть відбутися ті або інші збої, що приводять до перекручування інформації. Тому, при проектуванні цифрових автоматів повинні бути передбачені засоби, що дозволяють контролювати, виявляти і виправляти виникаючі помилки. Рішення всіх задач контролю стає можливим тільки при наявності певної надлишковості інформації, що супроводжує основну інформацію. Інакше кажучи, при представленні числа в якому-небудь коді, тобто під час кодування інформації, необхідно передбачити в цьому коді додаткові, так звані, контрольні розряди.

Систематичний код - код, що містить у собі, крім інформаційних, контрольні розряди.

У контрольні розряди записується деяка інформація про вихідне число. Тому можна говорити, що систематичний код має надлишковість.

### 6.2 Контроль методом парності-непарності

Якщо в математичному коді виділений один контрольний розряд ( $k=1$ ), то до кожного двійкового числа додається один надлишковий розряд і в нього записується 1 або 0 з такою умовою, щоб сума цифр у кожному числі була рівна 0 за модулем 2 для випадку парності або 1 для випадку непарності. Поява помилки в кодуванні виявиться за порушенням парності (непарності). При такому кодуванні допускається, що може виникнути тільки одна помилка. Приклад реалізації методу парності представлений нижче (таблиця 6.).

**Таблиця 6.1. Приклади контролю за парністю**

Число	Контрольний розряд	Перевірка (кількість 1 за модулем 2)
10101011	1	0
11001010	0	0
10010001	1	0
11001011	0	1 - порушення

Можна представити і трохи видозмінений спосіб контролю за методом парності-непарності. Довге слово розбивається на групи, кожна з яких містить  $n$  розрядів.

Контрольні розряди  $k$  виділяються всім групам по рядках і по стовпцях відповідно до наступної схеми (таблиця 6.).

**Таблиця 6.2. Вдосконалений контроль за парністю**

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$k_1$
$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$k_2$
$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$	$k_3$
$a_{16}$	$a_{17}$	$a_{18}$	$a_{19}$	$a_{20}$	$k_4$
$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$	$k_5$
$k_6$	$k_7$	$k_8$	$k_9$	$k_{10}$	

Збільшення надлишковості інформації приводить до того, що з'являється можливість не тільки знайти помилку, але і виправити її (таблиця 6.).

**Таблиця 6.3. Приклад вдосконаленого контролю за парністю**

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

Перевірка показує, що помилка виникла в інформації другого рядка і другого стовпця ліворуч. Отже, розряд, що містить помилкову інформацію, знаходиться на перетинанні другого рядка і другого стовпця, тобто в другому рядку треба другу 1 ліворуч замінити на 0.

### 6.3 Коди Геммінга

Коди, запропоновані американським ученим Р. Геммінгом, дозволяють не тільки знайти, але і виправити одиночні помилки. Ці коди - систематичні. Геммінг уперше ввів поняття кодової відстані.

Кодовою відстанню між двома словами називається число розрядів, у яких символи слів не збігаються.

Мінімальною кодовою відстанню ( $d_{min}$ ) даного коду називається мінімальна відстань між двома будь-якими словами в цьому коді. Якщо довжина слова  $n$ , то кодова відстань може приймати значення від 1 до  $n$ . Якщо є хоч одна пара слів, що відрізняються в одному розряді, то мінімальна кодова відстань дорівнює 1. Для

систематичних кодів  $d_{min} > 1$ . У загальному випадку, щоб код дозволяв виявляти помилки кратністю  $r$ , повинне виконуватися умова  $d_{min} \geq r + 1$ .

При пересиланні інформації до  $i$  інформаційних розрядів додається  $k$  перевірочних розрядів так, що загальна довжина  $n$  слова, яке пересилається, становить  $n = i + k$  розрядів. Співвідношення  $n$ ,  $i$  та  $k$  обирається з умови  $n + 1 = i + k + 1 \leq 2^k$ , тоді  $i \leq 2^k - k - 1$  або  $2^k \geq 1 + n$ ,  $2^{n-i} \geq 1 + n$ ,  $2^n \geq 2^i(1 + n)$ ,  $2^n / (1 + n) \geq 2^i$ .

Для різних значень  $n$  обчислені значення  $i$  та  $k$  наведені нижче (таблиця 6.).

**Таблиця 6.4. Кількість інформаційних та контрольних розрядів**

$n$	$i$	$k$
7	4	3
15	11	4
31	26	5

Формули, за допомогою яких формуються перевірочні розряди  $k$ , виводяться за допомогою перевірконої матриці (таблиця 6.), де позначено:

$i$  - інформаційні розряди, які необхідно переслати лінією зв'язку;

$k$  - перевірочні розряди, які додаються до інформаційних передавачем інформації;

$n_1 \dots n_{15}$  - розряди слова, яке передається (код Геммінга).

**Таблиця 6.5. Перевірочна матриця**

$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$k_1$	$k_2$	$i_3$	$k_4$	$i_5$	$i_6$	$i_7$	$k_8$	$i_9$	$i_{10}$	$i_{11}$	$i_{12}$	$i_{13}$	$i_{14}$	$i_{15}$

Номер розряду, який передається, записаний у таблиці в стовпчик у двійковому коді (старший розряд - верхній). Перевірочним розрядам відповідають ті графі таблиці, двійковий код яких має тільки одну 1. Перевірочні розряди знаходяться додаванням за модулем 2 тих інформаційних розрядів, які мають у своїй графі 1 на тому самому місці, що і відповідний перевірочний розряд:

$$k_1 = i_3 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} \oplus i_{13} \oplus i_{15};$$

$$k_2 = i_3 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} \oplus i_{14} \oplus i_{15};$$

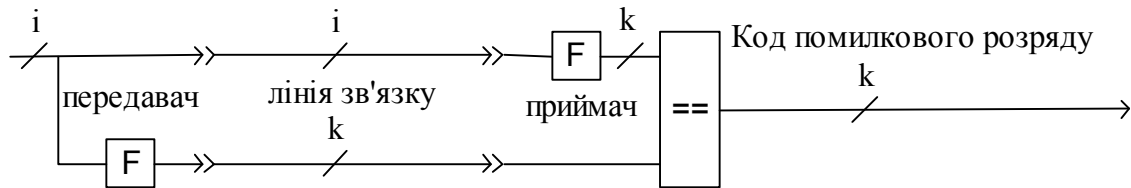
$$k_4 = i_5 \oplus i_6 \oplus i_7 \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15};$$

$$k_8 = i_9 \oplus i_{10} \oplus i_{11} \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15};$$

$\oplus$  - позначення операції додавання за модулем 2.

Для більшої кількості розрядів таблиці складаються аналогічно.

Рис. 6.1 містить схему пересилання інформації з використанням коду Геммінга.



F - вузол формування контрольних розрядів коду Геммінга

Рис. 6.1. Використання коду Геммінга

Нахилені риси (рис. 6.1) і позначення над ними вказують на кількість двійкових розрядів, що передаються відповідними лініями зв'язку.

На виході передавача і на вході приймача на інформаційних розрядах стоять однакові схеми формування коду Геммінга  $F$ . Їхні виходи порівнюються і при відсутності помилок повинні збігатися. Незбіжність - ознака помилки. Одночасно двійковий код на виходах схеми порівняння вказує, який інформаційний розряд помилковий, тобто є можливість його виправити.

Наприклад, сформувані код Геммінга для двійкового коду  $i = 010\ 0110\ 0011$  і перевірити, як він розпізнає помилку в розряді  $i_{13}$ .

Передавач сформує перевіірочні розряди ( $k$ ) коду Геммінга, як це показує таблиця 6.. Тобто, приймач повинен отримати код  $n = 110\ 0100\ 0110\ 0011$ .

$$k_1 = i_3 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} \oplus i_{13} \oplus i_{15} = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1;$$

$$k_2 = i_3 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} \oplus i_{14} \oplus i_{15} = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1;$$

$$k_4 = i_5 \oplus i_6 \oplus i_7 \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15} = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1;$$

$$k_8 = i_9 \oplus i_{10} \oplus i_{11} \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 0$$

Припустимо, внаслідок помилки під час передачі інформації змінився розряд  $n_{13}$ , тобто приймач отримав код  $110\ 0100\ 0110\ 0(1)11$  (розряд, який змінився, взято у дужки).

Таблиця 6.6. Перевірочна матриця передавача

$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$k_1$	$k_2$	$i_3$	$k_4$	$i_5$	$i_6$	$i_7$	$k_8$	$i_9$	$i_{10}$	$i_{11}$	$i_{12}$	$i_{13}$	$i_{14}$	$i_{15}$
$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$
1	1	0	1	1	0	0	0	1	1	0	0	0	1	1

Тоді приймач сформує перевірочні розряди (К) коду Геммінг, як це показує таблиця 6..

Таблиця 6.7. Перевірочна матриця приймача

$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$k_1$	$k_2$	$i_3$	$k_4$	$i_5$	$i_6$	$i_7$	$k_8$	$i_9$	$i_{10}$	$i_{11}$	$i_{12}$	$i_{13}$	$i_{14}$	$i_{15}$
$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$
0	1	0	1	1	0	0	1	1	1	0	0	(1)	1	1

$$K_1 = i_3 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} \oplus i_{13} \oplus i_{15} = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus (1) \oplus 1 = 0;$$

$$K_2 = i_3 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} \oplus i_{14} \oplus i_{15} = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1;$$

$$K_4 = i_5 \oplus i_6 \oplus i_7 \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15} = 1 \oplus 0 \oplus 0 \oplus 0 \oplus (1) \oplus 1 \oplus 1 = 0;$$

$$K_8 = i_9 \oplus i_{10} \oplus i_{11} \oplus i_{12} \oplus i_{13} \oplus i_{14} \oplus i_{15} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus (1) \oplus 1 \oplus 1 = 0.$$

Схема порівняння порозрядно порівнює коди  $K$  і  $k$  за допомогою операції додавання за модулем 2.

$$K \oplus k = (K_8 \oplus k_8)(K_4 \oplus k_4)(K_2 \oplus k_2)(K_1 \oplus k_1) = (1 \oplus 0)(0 \oplus 1)(1 \oplus 1)(0 \oplus 1) = 1101.$$

Якщо отриманий чотирирозрядний код дорівнює 0000, то інформація передалася без помилок. Будь-який інший код указує на номер розряду, який передався з помилкою (у прикладі  $1101_2 = 13_{10}$ ).

#### 6.4 Контроль за модулем

Контроль виконання арифметичних і логічних операцій можна здійснювати за

допомогою контрольних кодів, що представляють собою залишки від ділення чисел на деякий модуль. Такий контроль називається контролем за модулем. Для двійкових чисел цей модуль звичайно дорівнює або більше 3. Розрізняють числовий і цифровий контроль за модулем.

При числовому методі код заданого числа визначається як найменший додатній залишок від ділення числа на обраний модуль.

Наприклад: визначити контрольний код чисел  $A = 125$  і  $B = 89$  за модулем 11.

1)  $125 : 11 = 12$  (4), тобто контрольний код дорівнює 4.

2)  $89 : 11 = 8$  (1), тобто контрольний код дорівнює 1.

Тут числа у дужках – залишки від ділення.

При цифровому методі контролю контрольний код числа утвориться діленням суми цифр числа на обраний модуль. У даному варіанті можливі два шляхи одержання контрольного коду:

1) безпосереднє ділення суми цифр на модуль;

2) просте сумування цифр за обраним модулем.

Наприклад: визначити контрольний код для чисел 153 і 41 за модулем 3. Сума цифр 153 дорівнює 9, а сума цифр 41 дорівнює 5. Розділивши ці суми на 3 одержимо відповідно контрольні коди 0 і 2.

### 6.5 Контроль арифметичних операцій

Арифметичні операції можна представити у вигляді послідовності наступних елементарних операцій: передача слова, зсув, узяття оберненого коду, додавання.

Операцію зсуву можна представити як передачу слова з  $i$ -того розряду до  $(i+x)$ -го або  $(i-x)$ -го розряду. Тому, контроль зсуву можна здійснити за методом парності-непарності.

Контроль виконання арифметичних операцій (додавання, віднімання, множення) можна здійснити методом контролю за модулем. Одночасно з виконанням операції над числами та ж операція виробляється над їхніми контрольними кодами, і контрольний код результату основної операції порівнюється з результатом операції над контрольними кодами вихідних чисел. При розбіжності фіксується помилка. Тобто,

$$(A * B)(mod m) = A (mod m) * B (mod m),$$

якщо символ \* позначає операції додавання, віднімання, множення.

Структурна схема, яка пояснює контроль виконання арифметичних операцій, дуже подібна до схеми, яка пояснює використання коду Геммінга. При цьому замість лінії передачі задіяний вузол, який виконує ту чи іншу арифметичну операцію (рис. 6.2).

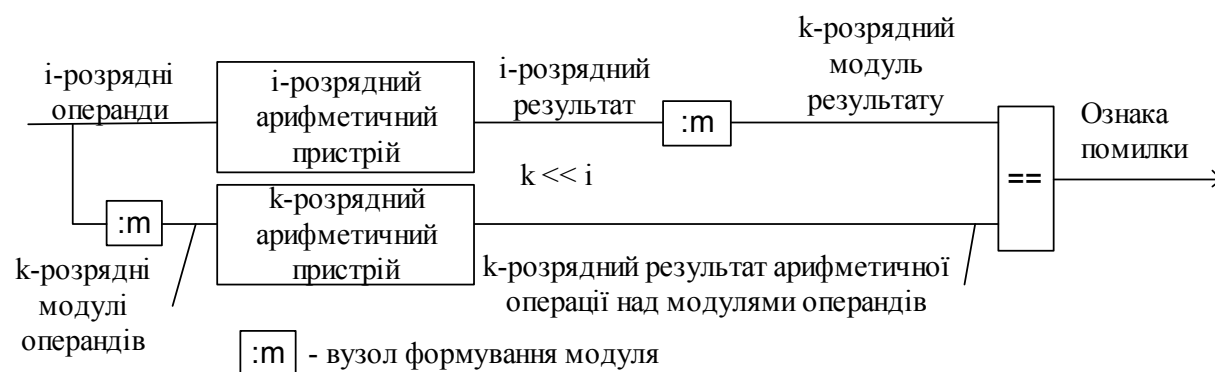


Рис. 6.2. Контроль за модулем

Наприклад:

1) Числовий контроль. Знайти контрольні коди чисел  $A = 571_{10} = 1000111011_2$  і  $B = 329_{10} = 0101001001_2$  і контрольний код їхньої суми методом контролю за модулем 3. При діленні А на 3 маємо залишок  $01_2 = 1_{10}$ , а при діленні В на 3 - залишок дорівнює  $10_2 = 2_{10}$ . Просумуємо ці числа та їхні залишки:

A =	571 = 1000111011	01
+	B =	329 = 0101001001
	A + B =	900 = 1110000100
		11

Якщо розділити суму  $A + B = 900_{10}$  на 3, залишок буде 0, і якщо розділити суму контрольних кодів на 3, залишок також буде 0. Оскільки залишки рівні, операція пройшла без збою.

2) Цифровий контроль.

На тому ж прикладі розглянемо цифровий контроль. Сума двійкових цифр А дорівнює 6, отже контрольний код дорівнює 0. Сума двійкових цифр В дорівнює 4, отже контрольний код цього числа є 1. Тобто, числа А і В з урахуванням контрольних кодів запишуться так:

$$\begin{array}{r} \hline A = \quad 571 = 1000111011 \quad 00 \\ + \quad B = \quad 329 = 0101001001 \quad 01 \\ \hline A + B = \quad 900 = 1110000100 \quad 01 \end{array}$$

Якщо додати двійкові цифри результату  $A + B$ , одержимо 4, отже контрольний код буде 1 і сума залишків теж 1 – помилки немає.



## 7 ОСНОВИ АЛГЕБРИ ЛОГІКИ

### 7.1 Основні поняття алгебри логіки

Логічною основою цифрових автоматів або комп'ютерів є алгебра логіки (булева алгебра) - одна з основних частин математичної логіки. Теорія логічних основ цифрових автоматів надзвичайно насичена досить специфічними термінами, поняттями. Тому в цій главі та у всіх наступних буде наведене багато визначень цих понять. Причому, нерідко для того самого поняття буде використано кілька визначень для більш чіткого розуміння його сутності і взаємозв'язку з іншими специфічними поняттями теорії цифрових автоматів. Отже, наведемо перший варіант визначення понять логічної змінної і логічної функції.

Функція  $f(a, b, \dots, z)$  називається логічною (перемикальною), або булевою, якщо вона, так само як і її аргументи  $a, b, \dots, z$ , можуть приймати тільки два значення: 0 або 1.

Логічна (булева) змінна ця така величина  $x$ , що може приймати тільки два значення: 0 або 1.

Таким чином, логічні функції, їхні аргументи і просто логічні змінні можуть приймати тільки два значення 0 або 1. Причому, у цих випадках цифри 0 і 1 є символами стану, а не числами. Алгебра логіки є алгеброю станів, а не алгеброю чисел, тому цю алгебру називають також алгеброю висловлень.

Висловленням називається твердження, про яке можна безперечно сказати, істинне воно або хибне. Якщо висловлення істинне, то говорять, що його значення істинності дорівнює одиниці. Якщо ж висловлення хибне, - його значення істинності дорівнює нулеві. Висловлень одночасно істинних і хибних не буває.

Висловлення бувають простими і складними. Прості окремі висловлення - це логічні змінні, їх прийнято позначати літерами латинського алфавіту. Наприклад, якщо просте висловлення  $X$  істинне, то  $X = 1$ , якщо ж хибно, то  $X = 0$ .

Висловлення з різним змістом позначаються різними літерами і вважаються різними. Два висловлення називаються еквівалентними, якщо істинності їх однакові. Еквівалентність висловлень позначається знаком рівності ( $=$ ) або тотожності ( $\equiv$ ) або ж знаком ( $\sim$ ). Наприклад, запис  $a = b$  означає, що  $a$  і  $b$  або істинні або хибні одночасно.

Очевидно, що складне висловлення - це та або інша логічна функція деяких логічних змінних, тобто аргументів. Знаки, що поєднують логічні змінні в складні висловлення, тобто в логічні функції, є знаками логічних дій, точніше логічних зв'язувань, а не математичних дій.

Сукупність значень аргументів логічної функції називається набором і може позначатися, зокрема, як  $x_0x_1\dots x_n$ , де змінна  $x_i$  дорівнює нулеві або одиниці ( $i = 1, 2, \dots, n$ ). Очевидно, що набір значень аргументів фактично являє собою деяке двійкове число. Кожному наборові значень аргументів приписується номер, який дорівнює двійковому числу, що відповідає значенню даного набору. Наприклад, для чотирьох аргументів:

0000 - нульовий набір;

0001 - перший набір;

0010 - другий набір;

1010 - десятий набір і т.д.

Таким чином, логічна функція (функція алгебри логіки, ФАЛ) це функція  $f(a, b, \dots, x)$  яка приймає значення 0 або 1 на наборі логічних перемінних  $a, b, \dots, x$ . Кожній логічній функції даного набору аргументів також прийнято приписувати номер: 0, 1, 2, ... .

Будь-яка логічна функція  $n$  аргументів визначена на  $2^n$  наборах, тобто може мати  $2^n$  наборів. Неповністю визначена логічна функція  $n$  змінних, це функція, задана на числі наборів меншому  $2^n$ . Для наборів, на яких функція не визначена, її значення можна прийняти довільним.

Число різних логічних функцій  $n$  аргументів скінчене і дорівнює  $2^{2^n}$ . Як видно (таблиця 7.) кількість ФАЛ дуже велика і прямує до нескінченості із зростанням кількості змінних.

**Таблиця 7.1. Змінні, набори і функції алгебри логіки**

кількість змінних	кількість наборів	кількість ФАЛ
0	1	2
1	2	4
2	4	16
3	8	256

4	16	65536
...	...	...
n	$2^n$	$2^{2^n}$

Фактична кількість ФАЛ, які залежать від  $n$  змінних, може бути менша їхньої формальної кількості  $2^{2^n}$ . Наприклад, функція  $f(a,b,c) = ab$  формально залежить від трьох змінних  $(a,b,c)$ , а фактично – тільки від двох ( $= ab$ ).

Фактична кількість ФАЛ, які залежать від  $n$  змінних визначається за допомогою ітераційної формули:

$$A_n = 2^{2^n} - C_n^{n-1} A_{n-1} - C_n^{n-2} A_{n-2} - \dots - C_n^1 A_1 - A_0,$$

де  $C_h^l = \frac{h!}{l!(h-l)!}$  - кількість сполучень по  $l$  елементів з  $h$ .

Користуючись методами функціональної декомпозиції і перестановки можна звести функції, які залежать від великої кількості змінних, до функцій меншої кількості змінних. Це дозволяє не вивчати усі логічні функції, а обмежитися вивченням властивостей невеликої кількості простих логічних функцій (базисом або функціонально-повною системою функцій алгебри логіки, ФПС ФАЛ). Тоді для оволодіння усіма логічними функціями треба вирішити дві задачі:

вибір базису;

представлення потрібної логічної функції за допомогою функцій, які входять до базису.

Функції алгебри логіки в цифрових автоматах технічно реалізуються за допомогою різного типу логічних електронних елементів - вентилів. Вибір базису для представлення усіх функцій алгебри логіки однозначно приводить до вибору елементів цифрової техніки, які будуть реалізовувати ці функції.

При переході від логічних функцій до логічних схем звичайно приймається, що логічній 1 відповідає імпульсний сигнал стандартної амплітуди, наприклад, високого рівня, а логічному 0 - низького рівня і звичайно фіксованої тривалості. Причому, усі вхідні сигнали повинні надходити на кожен вентиль одночасною. Інколи не зовсім правильно говорять, що символіві стану 1 відповідає наявність деякого сигналу, зокрема електричного, а символіві стану 0 - відсутність цього сигналу. Інакше кажучи,

якщо є деяке джерело цифрової інформації, що розглядається як визначаюча логічна змінна, наприклад, змінна  $A$ , то при наявності сигналу від цього джерела  $A = 1$ , а при відсутності  $A = 0$ . Насправді мають на увазі, що сигнал „є”, якщо його напруга більша деякого порогового значення (наприклад, 2 В, при цьому вважають, що  $A = 1$ ), і сигналу „немає”, коли його напруга менше деякого порогового значення (наприклад, 0,8 В, при цьому вважають, що  $A = 0$ ). Насправді, сигнал завжди є, тільки з різною амплітудою або іншими характеристиками.

## 7.2 Функції алгебри логіки нуля змінної (ФАЛ0)

Для випадку відсутності аргументу (кількість аргументів рівна 0) маємо  $2^0$  (тобто, 1) набір і  $2^{2^0} = 2$  (тобто, 2) логічні функції від нуля змінних (таблиця 7.). При цьому формальна кількість ФАЛ0 збігається з фактичною кількістю, тобто  $A_0 = 2$ .

**Таблиця 7.2. Функції алгебри логіки нуля змінних**

$f_0$	$f_1$
0	1

У зв'язку з тим, що функція  $f_1$  завжди дорівнює одиниці, то ця функція називається абсолютно істинною (константа одиниці).

Аналітичний запис  $f_1 = 1$ .

Інші назви функції – ТАК, одиниця.

Технічна реалізація – безпосереднє з'єднання з джерелом живлення (рис. 1.4, VCC – поширене в системах автоматизованого проектування позначення джерела напруги, утворилося від англійського виразу Voltage Collector Collector – напруга живлення колектора біполярного транзистора. Вираз вживається зараз навіть у схемах де немає біполярних транзисторів. Загалом напруга позначається літерою V з наступними індексами, які уточнюють її призначення. Для напруг живлення індекси, як правило, подвоюються).



*Рис. 7.1. Константа одиниці*

Функція  $f_0$  - абсолютно хибна функція (константа нуля).

Аналітичний запис  $f_0 = 0$ .

Інші назви функції – нуль.

Технічна реалізація – безпосереднє з'єднання з землею (рис. 7.2, GND – поширене в системах автоматизованого проектування позначення напруги 0 Вольт - землі, утворилося від англійського Ground - земля).

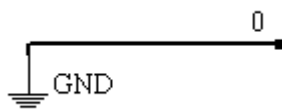


Рис. 7.2. Константа нуля

### 7.3 Функції алгебри логіки однієї змінної (ФАЛ1)

Для одного аргументу маємо  $2^1 = 2$  набори і  $2^2 = 4$  чотири логічні функції від однієї змінної (таблиця 7.).

Фактична кількість ФАЛ1 дорівнює  $A_1 = 2^{2^1 - A_0} = 4 - 2 = 2$ . У таблиці (таблиця 7.) відмічено, які з ФАЛ1 ( $f_0(a)$  і  $f_3(a)$ ) фактично є функціями алгебри логіки нуля змінних (ФАЛ0).

Таблиця 7.3. Функції алгебри логіки однієї змінної

a	$f_0(a)$	$f_1(a)$	$f_2(a)$	$f_3(a)$
0	0	0	1	1
1	0	1	0	1
Фактично є	ФАЛ0			ФАЛ0

Функція  $f_1(a)$ , що повторює абсолютне значення логічної змінної, - функція тотожності.

Варіанти аналітичного запису  $f_1(a) = a$ ;  $f_1(a) \equiv a$ .

Наведена вище формула читається так:  $f_1$  дорівнює a.

Інші назви для функція тотожності – функція повторення.

Умовні графічні позначення (символ) пристрою (елемента цифрової техгніки), який реалізує функцію тотожності (елемента повторювача) наведені нижче (рис. 7.3), де (і далі) позначено:

а – умовне графічне позначення згідно з стандартом ГОСТ 2.743-91;

б - умовне графічне позначення згідно з міжнародним стандартом.

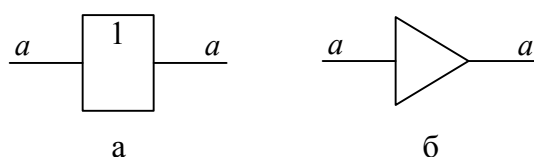


Рис. 7.3. Повторювач

Функція  $f_2(a)$ , що приймає значення, обернене значенню  $a$ , - логічне заперечення (інверсія), або функція НЕ (NOT), що позначається одним з наступних способів:

$$f_2(a) = \bar{a} = \neg a = /a = !a = ^a = \text{not } a.$$

Наведена вище формула читається так:  $f_2$  дорівнює не  $a$ .

Логічне заперечення або інверсія (від латинського Inversio - перестановка) деякої логічної змінної, наприклад, змінної  $a$ , це також логічна змінна, яка приймає значення, обернене значенню змінної  $a$ . Це позначається як  $\bar{a}$ . Якщо  $a = 1$ , то  $\bar{a} = 0$ , якщо ж  $a = 0$ , то  $\bar{a} = 1$ . Але потрібно врахувати, що часто за допомогою заперечення, тобто інверсії змінної просто позначають випадок, коли її значення дорівнює нулеві.

Графічна ілюстрація повторення і інверсії наведена нижче (рис. 7.4). Припустимо, що через  $a$  позначена така умова, як навчання на кафедрі ЕОМ. Тоді студентів, які мають цю властивість можна „зібрати” у коло. Тоді всі, хто знаходиться у колі, навчаються на кафедрі ЕОМ, для них умова виконується, для них  $a$  – істинне, для них  $a=1$ , ТАК, вони навчаються на кафедрі ЕОМ.

Для всіх інших, хто знаходиться поза колом, умова не виконуються, вони НЕ навчаються на кафедрі ЕОМ, для них умова  $a$  – хибна, для них правильно буде НЕ  $a$ ,  $\text{not } a$ ,  $a=0$ .

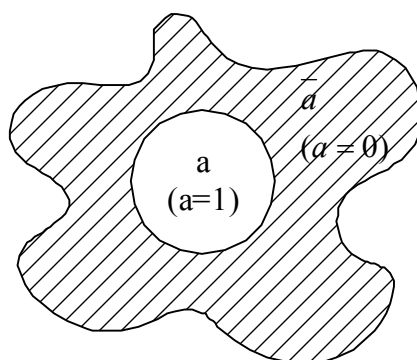


Рис. 7.4. Графічна ілюстрація інверсії

Символи елемента, який реалізує функцію інверсії (інвертора) наведено нижче (Рис. 7.5).

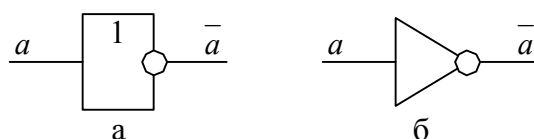


Рис. 7.5. Інвертор (елемент НЕ)

#### 7.4 Функції алгебри логіки двох змінних (ФАЛ2)

Нижче (таблиця 7.) наведений повний перелік усіх шістнадцяти елементарних логічних функцій від двох змінних  $a$  і  $b$ .

Фактична кількість ФАЛ2 дорівнює

$$A_2 = 2^{2^2} - C_2^1 A_1 - A_0 = 16 - 2 \cdot 2 - 2 = 10.$$

У таблиці (таблиця 7.) відмічено, які з ФАЛ2 фактично є функціями алгебри логіки нуля або однієї змінних (ФАЛ0 або ФАЛ1).

Таблиця 7.4 - Функції алгебри логіки двох змінних

Функція	$ab$				Аналітичний вираз	Назва	Фактично є
	00	01	10	11			
$f_0$	0	0	0	0	0	константа нуль	ФАЛ0
$f_1$	0	0	0	1	$ab$	кон'юнкція	
$f_2$	0	0	1	0	$\overline{a \rightarrow b}$	заперечення імплікації прямої	
$f_3$	0	0	1	1	$a$	змінна $a$	ФАЛ1
$f_4$	0	1	0	0	$\overline{a \leftarrow b}$	заперечення імплікації зворотної	
$f_5$	0	1	0	1	$b$	змінна $b$	ФАЛ1
$f_6$	0	1	1	0	$a \oplus b$	додавання за модулем 2	
$f_7$	0	1	1	1	$a \vee b$	диз'юнкція	
$f_8$	1	0	0	0	$a \downarrow b$	функція Пірса (АБО-НЕ)	
$f_9$	1	0	0	1	$a \equiv b$	рівнозначність	
$f_{10}$	1	0	1	0	$\bar{b}$	інверсія $b$	ФАЛ1
$f_{11}$	1	0	1	1	$a \leftarrow b$	імплікація зворотна	
$f_{12}$	1	1	0	0	$\bar{a}$	інверсія $a$	ФАЛ1

$f_{13}$	1	1	0	1	$a \rightarrow b$	імплікація пряма	
$f_{14}$	1	1	1	0	$a b$	функція Шеффера (І-НЕ)	
$f_{15}$	1	1	1	1	1	константа одиниця	ФАЛО

Диз'юнкція (від латинського *disjunctio* - роз'єднання), логічне додавання або функція АБО (ИЛИ, OR) - це функція  $f_7(a,b)$ , що істинна тоді, коли істинна хоча б одна з її змінних (таблиця 7.).

**Таблиця 7.5. Таблиця істинності диз'юнкції (АБО)**

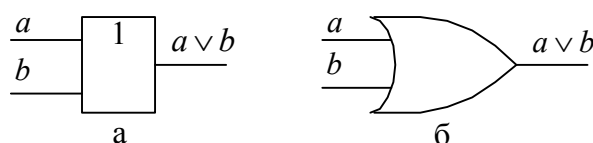
Змінні (аргументи)		Функція
$a$	$b$	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Варіанти аналітичного запису цієї функції:  $f_7(a,b) = a \vee b$  і зараз вже дуже рідко  $a + b$ .

Це читається в такий спосіб:  $f_7$  дорівнює  $a$  або  $b$ .

Це значить, що функція істинна, тобто дорівнює одиниці, коли аргумент  $a = 1$ , тобто істинний, або аргумент  $b = 1$ , або ж обидва аргументи істинні одночасно.

Символи елемента, який реалізує функцію диз'юнкції (диз'юнктора, елемента АБО) наведено нижче (рис. 7.6).



*Рис. 7.6. Диз'юнктор (елемент АБО)*

Графічна ілюстрація функції АБО наведена нижче (рис. 7.7).



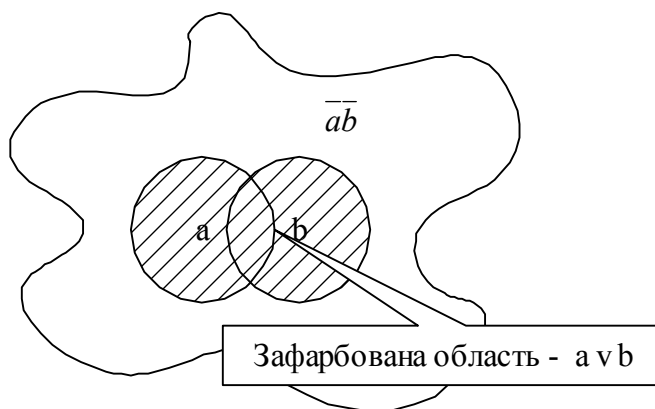


Рис. 7.7. Графічна ілюстрація диз'юнкції

Кон'юнкція (від латинського conjunctio – сполучник, зв'язок), логічне множення або функція І (И, AND) - це функція  $f_i(a, b)$ , що істинна тоді, коли всі її змінні одночасно істинні (таблиця 7.).

Таблиця 7.6. Таблиця істинності кон'юнкції (І)

Змінні (аргументи)		Функція
$a$	$b$	$a \& b$
0	0	0
0	1	0
1	0	0
1	1	1

Варіанти аналітичного запису цієї функції: .

Це читається в такий спосіб:  $f_i$  дорівнює  $ab$  (можна читати і так:  $f_i$  дорівнює  $a$  і  $b$ ).

Це значить, що функція істинна, тобто дорівнює одиниці, коли обидва аргументи одночасно істинні, тобто дорівнюють одиниці.

Символи елемента, який реалізує функцію кон'юнкції (кон'юнктора, елемента І) наведені нижче (рис. 7.8).

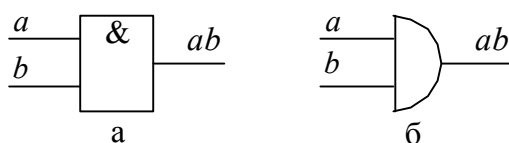


Рис. 7.8. Кон'юнктор (елемент І)

Графічна ілюстрація функції І наведена нижче (рис. 7.9).

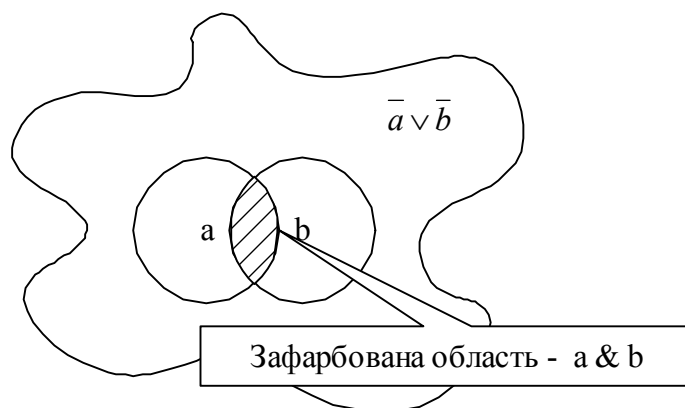


Рис. 7.9. Графічна ілюстрація кон'юнкції

Функція (штрих) Шеффера або функція І-НЕ (NOT AND, NAND, И-НЕ) - це функція  $f_{14}(a, b)$ , що хибна тоді, коли всі змінні істинні (таблиця 7.).

Таблиця 7.7. Таблиця істинності функції Шеффера (І-НЕ)

Змінні (аргументи)		Функція
$a$	$b$	$\overline{a \& b}$
0	0	1
0	1	1
1	0	1
1	1	0

Умовна позначка цієї функції:

$$f_{14}(a, b) = \overline{a \& b} = \overline{ab} = \overline{a \cdot b} = \overline{a \wedge b} = a | b.$$

Це читається в такий спосіб:  $f_{14}$  дорівнює не  $ab$ .

Ця функція хибна, тобто дорівнює 0, коли обидва аргументи одночасно істинні, тобто дорівнюють одиниці, і функція істинна, тобто дорівнює одиниці, коли або обидва аргументи одночасно хибні, або ж хоча б один з них хибний.

Символи елемента, який реалізує функцію Шеффера (І-НЕ) наведені нижче (рис. 7.10).

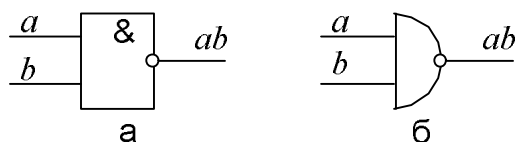


Рис. 7.10. Елемент Шеффера (І-НЕ)

Графічна ілюстрація функції Шеффера (І-НЕ) наведена нижче (рис. 7.11).

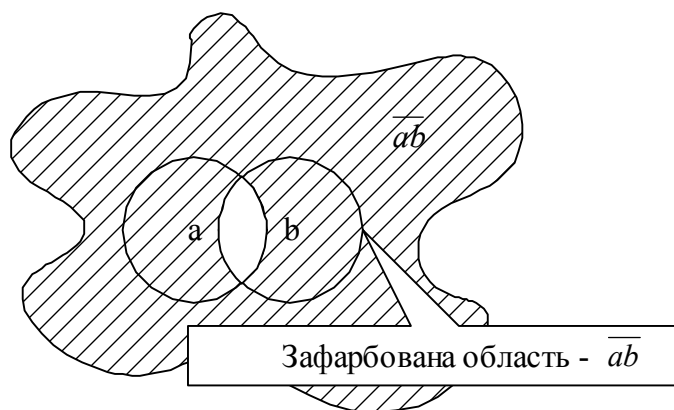


Рис. 7.11. Графічна ілюстрація функції Шеффера (І-НЕ)

Функція (стрілка) Пірса (Вебба) або функція АБО-НЕ (ИЛИ-НЕ, NOT OR, NOR) - це функція  $f_8(a, b)$ , що істинна тільки тоді, коли всі змінні хибні (таблиця 7.). (Пірс Чарлз Сандерс (1839-1914) - американський філософ, логік, математик і натураліст. Родоначальник прагматизму. Висунув принцип, відповідно до якого зміст поняття цілком вичерпується представленням про його можливі наслідки. Засновник семіотики. Роботи з математичної логіки).

Умовна позначка цієї функції:

$$f_8(a, b) = \overline{a \vee b} = \overline{a + b} = a \downarrow b.$$

Це читається в такий спосіб:  $f_8$  дорівнює не ( $a$  або  $b$ ).

Ця функція хибна, тобто дорівнює 0, коли хоча б один з її аргументів істинний, або ж обоє одночасно істинні, тобто дорівнюють одиниці, і функція істинна, тобто дорівнює одиниці, коли обидва аргументи одночасно хибні.

Таблиця 7.8. Таблиця істинності функції Пірса (АБО-НЕ)

Змінні (аргументи)		Функція
$a$	$b$	$\overline{a \vee b}$
0	0	1

0	1	0
1	0	0
1	1	0

Символи елемента, який реалізує функцію Пірса (АБО-НЕ) наведені нижче (рис. 7.12).

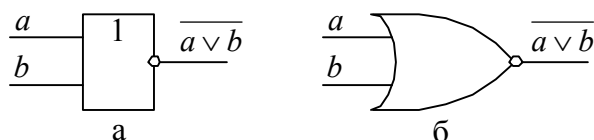


Рис. 7.12. Елемент Пірса (АБО-НЕ)

Графічна ілюстрація функції Пірса (АБО-НЕ) наведена нижче (рис. 7.13).

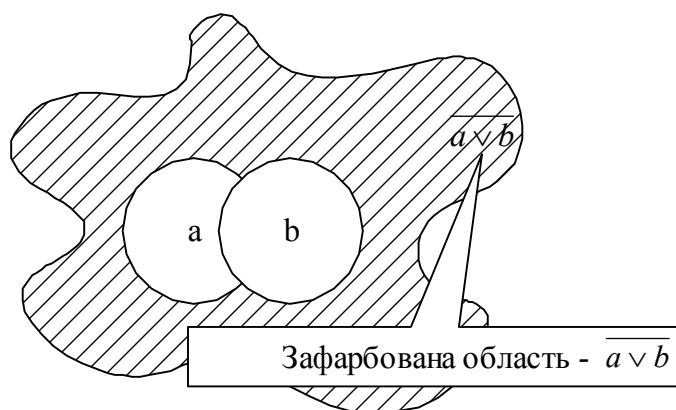


Рис. 7.13. Графічна ілюстрація функції Пірса (АБО-НЕ)

Виключне АБО (XOR) - це функція  $f_6(a,b)$ , що позначається знаком  $\oplus$ . Ця операція реалізує функцію нерівнозначності, тобто фактично реалізується процедура додавання за модулем 2 (таблиця 7.).

Таблиця 7.9. Таблиця істинності функції виключне АБО

Змінні (аргументи)		Функція
$a$	$b$	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

$f_6(a,b) = a \oplus b = a \neq b = a \# b$  (останнє позначення не є загальновизнаним, про його

значення треба попереджати заздалегіть).

Це читається в такий спосіб:  $f_6$  дорівнює  $a$  плюс  $b$  за модулем 2,  $f_6$  дорівнює  $a$  ксор  $b$  (від англійського XOR – читається як «ксор»).

Ця функція хибна, тобто дорівнює 0, коли її аргументи мають однакові значення, і функція істинна, тобто дорівнює одиниці, коли аргументи мають різні значення (істинне значення має виключно один з аргументів).

Символи елемента, який реалізує функцію нерівнозначності (виключне АБО) наведені нижче (рис. 7.14).

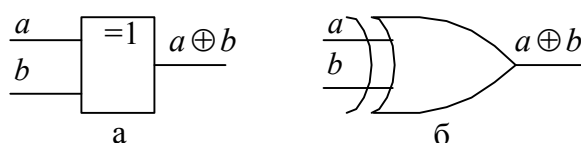


Рис. 7.14. Елемент виключне АБО

Графічна ілюстрація функції виключне АБО наведена нижче (рис. 7.15).

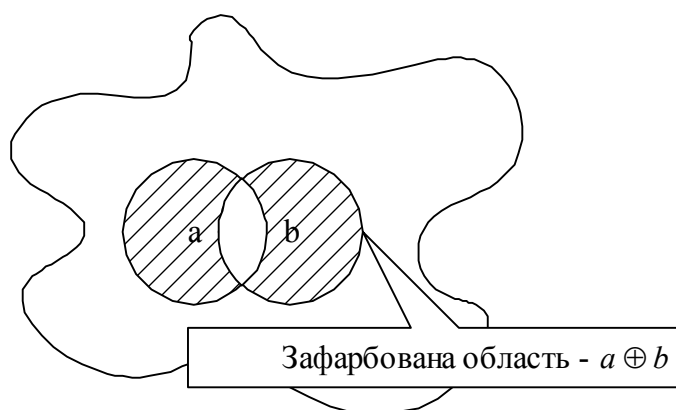


Рис. 7.15. Графічна ілюстрація функції виключне АБО

Рівнозначність (еквівалентність) - це функція  $f_7(a, b)$  (таблиця 7.).

Таблиця 7.10. Таблиця істинності функції рівнозначності

Змінні (аргументи)		Функція
$a$	$b$	$a \equiv b$
0	0	1
0	1	0
1	0	0
1	1	1

$$f_9(a, b) = a \equiv b = a \sim b = a == b.$$

Це читається в такий спосіб:  $f_9$  істина, коли  $a$  дорівнює  $b$ .

Ця функція хибна, тобто дорівнює 0, коли її аргументи мають різні значення, і функція істинна, тобто дорівнює одиниці, коли аргументи мають однакові значення.

Символи елемента, який реалізує функцію рівнозначності (еквівалентності) наведені нижче (рис. 7.16).

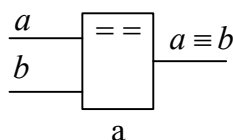


Рис. 7.16. Елемент рівнозначності (еквівалентності)

Графічна ілюстрація функції рівнозначності (еквівалентності) наведена нижче (рис. 7.17).

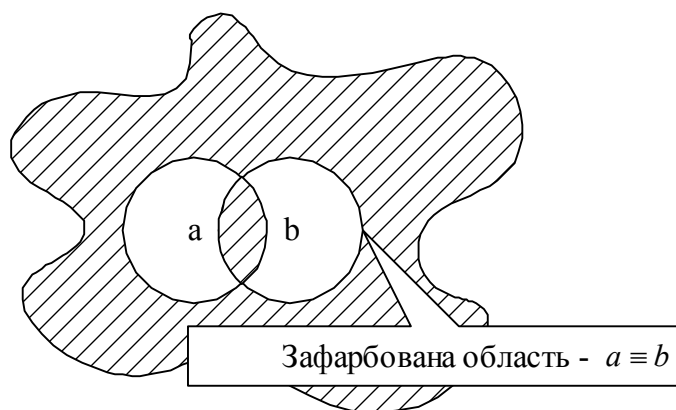


Рис. 7.17. Графічна ілюстрація функції рівнозначності (еквівалентності)

Імплікація (пряма) від  $a$  до  $b$  або функція ЯКЩО-ТО (IF-THEN) це функція  $f_{13}(a, b)$ , що хибна тоді і тільки тоді, коли  $a$  істинно і  $b$  хибно. Слово „імплікація” утворилося від латинського *Implicare* - тісно зв'язую.

Говорять, що  $a$  включене в  $b$ ,  $a$  менше або дорівнює  $b$  ( $a \leq b$ ). Графічну ілюстрацію прямої імплікації наведено нижче (рис. 7.18). Таблицю істинності прямої імплікації наведено нижче (таблиця 7.). У коментарях вважається, якщо  $X$  потрапляє в коло  $a$  ( $b$ ), то  $a=1$  ( $b=1$ ), в іншому випадку  $a=0$  ( $b=0$ ). Якщо висновок правильний, то це правда – функція дорівнює 1, якщо неправда – функція дорівнює 0.

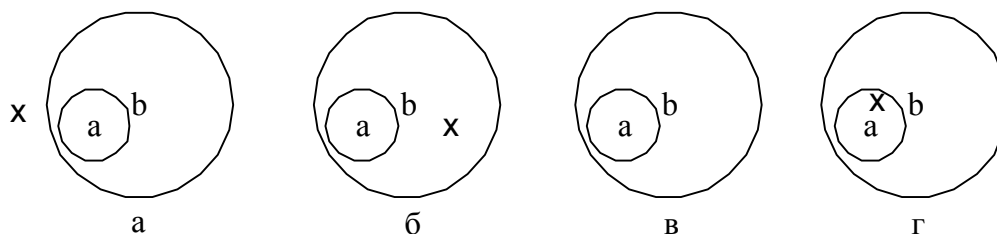


Рис. 7.18. Графічна ілюстрація прямої імплікації

Таблиця 7.11. Таблиця істинності імплікації

Змінні (аргументи)		Примітки	Функція	Рис. 7.18
$a$	$b$	$a$ включено в $b$	$f = a \rightarrow b$	
0	0	Якщо $X$ не потрапляє в $a$ , то $X$ може не потрапити в $b$	1 (правда)	а
0	1	Якщо $X$ не потрапляє в $a$ , то $X$ може потрапити в $b$	1 (правда)	б
1	0	Якщо $X$ потрапляє в $a$ , то $X$ може не потрапити в $b$	0 (неправда)	в
1	1	Якщо $X$ потрапляє в $a$ , то $X$ може потрапити в $b$	1 (правда)	г

Аргумент  $a$  називається посилкою, а  $b$  - наслідком. Умовна позначка прямої імплікації  $f_{13}(a, b) = a \rightarrow b$  (функція дорівнює імплікації від  $a$  до  $b$ ).

Імплікація зворотна дуже подібна до імплікації прямої, це також функція типу ЯКЩО-ТО (IF-THEN). Це функція  $f_{11}(a, b)$ , що хибна тоді і тільки тоді, коли  $b$  істинне і  $a$  хибне. Це імплікація від  $b$  до  $a$ . Говорять, що  $b$  включене в  $a$ ,  $b$  менше або дорівнює  $a$  ( $b \leq a$ ). Таблиця істинності зворотної імплікації наведена нижче (таблиця 7.). Графічна ілюстрація прямої імплікації наведена нижче (рис. 7.19).

Таблиця 7.12. Таблиця істинності зворотної імплікації

Змінні (аргументи)		Примітки	Функція	Рис. 7.19
$a$	$b$	$b$ включене в $a$	$f = a \leftarrow b$	

0	0	Якщо $X$ не потрапляє в $b$ , то $X$ може не потрапити в $a$	1 (правда)	а
0	1	Якщо $X$ потрапляє в $b$ , то $X$ може не потрапити в $a$	0 (неправда)	в
1	0	Якщо $X$ не потрапляє в $b$ , то $X$ може потрапити в $a$	1 (правда)	б
1	1	Якщо $X$ потрапляє в $b$ , то $X$ може потрапити в $a$	1 (правда)	г

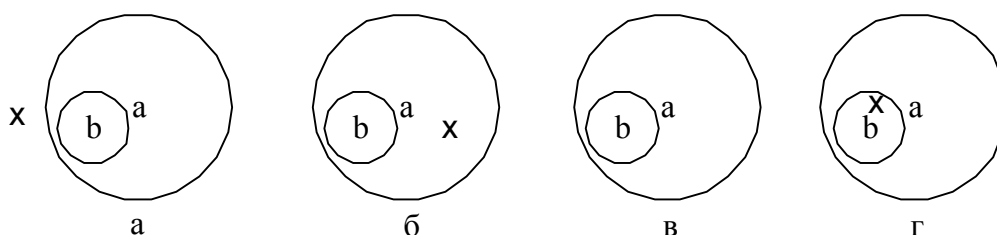


Рис. 7.19. Графічна ілюстрація зворотної імплікації

Умовна позначка зворотної імплікації  $f_{11}(a, b) = a \leftarrow b$  (функція дорівнює імплікації від  $b$  до  $a$ ).

Заперечення імплікації (прямої) це функція  $f_2(a, b)$ . Говорять,  $a$  більше  $b$  ( $a > b$ ). Таблиця істинності функції заперечення прямої імплікації наведена нижче (таблиця 7.).

Таблиця 7.13. Таблиця істинності функції заперечення імплікації

Змінні (аргументи)		Функція
$a$	$b$	$f = \overline{a \rightarrow b}$
0	0	0
0	1	0
1	0	1
1	1	0

Умовна позначка прямої імплікації  $f_2(a, b) = \overline{a \rightarrow b}$  (функція дорівнює запереченню імплікації від  $a$  до  $b$ ).

Заперечення зворотної імплікації це функція  $f_4(a, b)$ . Говорять,  $b$  більше  $a$  ( $b > a$ ). Таблиця істинності функції заперечення зворотної імплікації наведена нижче (таблиця 7.).



Таблиця 7.14. Таблиця істинності функції заперечення зворотної імплікації

Змінні (аргументи)		Функція
$a$	$b$	$f = \overline{a \leftarrow b}$
0	0	0
0	1	1
1	0	0
1	1	0

Умовна позначка прямої імплікації  $f_4(a, b) = \overline{a \leftarrow b}$  (функція дорівнює запереченню імплікації від  $a$  до  $b$ ).

З усіх наведених вище визначень ясно, що в алгебрі логіки всі знаки дій:  $\wedge$  або  $\&$ ,  $\vee$  або  $+$ ,  $\rightarrow$ ,  $\oplus$  і т.д, у відмінності від звичайної алгебри, є знаками логічних зв'язок, тобто логічних дій, а не знаками арифметичних дій.

Уведемо ще три специфічних поняття алгебри логіки.

Дві функції вважаються рівносильними одна одній, якщо вони приймають на всіх можливих наборах своїх аргументів ті самі значення.

Логічна змінна  $x$  дійсна, якщо значення логічної функції  $f(a, b, \dots, x, \dots, z)$  змінюється при зміні  $x$ . У протилежному випадку ця змінна для даної функції фіктивна, тобто фактично не є її аргументом.

Необхідність введення двох останніх понять виникла з наступної причини. При аналізі деякої невідомої логічної функції (логічної схеми), для якої необхідно сформулювати аналітичний вираз, не всі використовувані змінні можуть бути аргументами цієї функції, що і виявляється в підсумку проведеного аналізу.

Як буде показано надалі, будь-яку логічну операцію над логічними змінними можна звести до певної сукупності простіших логічних функцій, наприклад, таких як І, АБО, НЕ та виключне АБО.

Логічні операції в комп'ютерах виконуються також над двійковими числами і виконуються по-розрядно. Розглянемо кілька простих прикладів виконання логічних операцій над двома двійковими числами (функції І, АБО, виключне АБО – це функції двох змінних, тому в наведених нижче прикладах в них приймають участь два операнди, це операції над двома числами. А операція НЕ – це функція однієї змінної, тому у прикладі приймає участь один операнд, це операція над одним числом):

<i>I</i>	<i>АБО</i>	<i>ВиключнеАБО</i>	<i>НЕ</i>
01011010	01011010	01011010	01011010
11110000	11110000	11110000	
$\overline{01010000}$	$\overline{1111010}$	$\overline{1010010}$	$\overline{10100101}$

Підстановка в логічну функцію замість її аргументів інших логічних функцій називається суперпозицією.

Система логічних функцій називається функціонально повною або базисом, якщо за допомогою функцій, що входять у цю систему, застосовуючи операції суперпозиції і підстановки, можна одержати будь-яку як завгодно складну логічну функцію.

Як будь-яку логічну функцію можна реалізувати за допомогою відповідної комбінації елементарних логічних функцій, так і будь-яку логічну схему можна сформулювати за допомогою відповідної комбінації логічних вентилів. Усе це буде ще раз обговорюватися більш докладно в Розділі 9.

### 7.5 Основні правила виконання операцій у базисі Буля.

(Буль (Boole) Джордж (1815-1864) - англійський математик і логік, один з засновників математичної логіки. Розробив алгебру логіки („Дослідження законів мислення”, 1854)).

Базис Буля утворюють три функції алгебри логіки: *I*, *АБО*, *НЕ* (кон’юнкція, диз’юнкція, інверсія). Розділ алгебра логіки, який займається вивченням властивостей саме цих трьох функцій називають алгеброю Буля.

Розглянемо основні закони, аксіоми і теореми алгебри Буля.

Деякі закони звичайної алгебри застосовні і до алгебри Буля. Але алгебра Буля має и ряд специфічних аксіом і теорем, основні з яких, необхідні для аналізу і синтезу логічних виразів або схем, наведені нижче (таблиця 7.).

**Таблиця 7.15. Основні правила алгебри Буля**

Для диз’юнкції	Для кон’юнкції	Примітки
$a \vee b = b \vee a$	$ab = ba$	Закон комутативності
$a \vee (b \vee c) = (a \vee b) \vee c$	$a(bc) = (ab)c$	Закон асоціативності
$a \vee bc = (a \vee b)(a \vee c)$	$a(b \vee c) = ab \vee ac$	Закон дистрибутивності
$a = 1$ , якщо $a \neq 0$ ;	$a = 0$ , якщо $a \neq 1$ ;	

Якщо $a=0$ , то $\bar{a}=1$	Якщо $a=1$ , то $\bar{a}=0$	
$0 \vee 0 = 0$	$0 \cdot 0 = 0$ ;	
$0 \vee 1 = 1$	$1 \cdot 0 = 0$ ;	
$1 \vee 1 = 1$	$1 \cdot 1 = 1$ ;	
$\bar{0} = 1$	$\bar{1} = 0$	
$a \vee 0 = a$	$a \cdot 1 = a$ ;	
$a \vee 1 = 1$	$a \cdot 0 = 0$ ;	
$a \vee a = a$	$a \cdot a = a$ ;	
$\overline{\bar{a}} = a$	$\overline{\bar{a}} = a$	
$a \vee \bar{a} = 1$	$a \cdot \bar{a} = 0$	
$\overline{a \vee b \vee c} = \bar{a} \cdot \bar{b} \cdot \bar{c}$	$\overline{abc} = \bar{a} \vee \bar{b} \vee \bar{c}$	Правило де Моргана
$a(b \vee c) = a$	$a \vee ab = a$	(закон поглинання)
$a \vee \bar{a}b = a \vee b$	$a(\bar{a} \vee b) = ab$	(закон поглинання)
$ab \vee \bar{a}b = b$	$(a \vee b)(\bar{a} \vee b) = b$	(закон склеювання)

Аксіоми і теореми для диз'юнкцій (кон'юнкцій) називаються двоїстими аксіомам і теоремам для кон'юнкцій (диз'юнкцій).

Двоїстість визначається як зміна всіх знаків операції I на знаки операції АБО, усіх знаків операції АБО на знаки операції I, усіх нулів на одиниці і всіх одиниць на нулі.

Двоїстість є одним з основних властивостей алгебри Буля й означає, що якщо  $f(a, b, c)$  і  $f'(a, b, c)$  - двоїсті функції, то  $\overline{f(a,b,c)} = f'(\bar{a}, \bar{b}, \bar{c})$ .

Закони де Моргана є однієї з ілюстрацій двоїстості. З законів де Моргана випливають наслідки:

$$a \cdot b \cdot c = \overline{\overline{a \vee b \vee c}},$$

$$a \vee b \vee c = \overline{\overline{a \cdot b \cdot c}}.$$

Отже, з'являється можливість виражати кон'юнкцію через диз'юнкцію і заперечення, або диз'юнкцію - через кон'юнкцію і заперечення. Закони де Моргана і наслідки з них справедливі для будь-якої кількості змінних (Морган (De Morgan) Огастес (Августус) (1806-1871) - шотландський математик і логік. Праці з алгебри, теорії рядів. Незалежно від Дж. Буля прийшов до основних ідей математичної логіки).

## 7.6 Основні правила виконання операцій у базисі Жегалкіна.

(Жегалкін Іван Іванович (1869 – 1947) – російський математик. Роботи з теорії множин, алгебри логіки, проблеми розв’язності).

Базис Жегалкіна складається з трьох функцій алгебри логіки – І, виключне АБО, константи одиниці. Розділ алгебра логіки, який займається вивченням властивостей саме цих трьох функцій, називають алгеброю Жегалкіна.

Функція додавання за модулем 2 представляється в базисі Буля в такий спосіб:

$$a \oplus b = a\bar{b} \vee \bar{a}b = (a \vee \bar{b})(\bar{a} \vee b).$$

Для цієї функції справедливі наступні аксіоми:

$$a \oplus a = 0; a \oplus a \oplus a = a; a \oplus \bar{a} = 1; a \oplus 1 = \bar{a}; a \oplus 0 = a.$$

На підставі розглянутих аксіом і властивостей елементарних логічних функцій можна, наприклад, вивести правила представлення функцій І, АБО, НЕ через функцію додавання за модулем 2 і навпаки:

$$\bar{a} = a \oplus 1; a \vee b = a \oplus b \oplus ab; ab = (a \oplus b) \oplus (a \vee b).$$

## 7.7 Основні правила виконання операцій у монобазисі І-НЕ.

Для функції Шеффера (штрих Шеффера) – “|”), що може бути виражена співвідношенням  $a|b = \overline{ab} = \bar{a} \vee \bar{b}$ , характерні аксіоми, наведені нижче (таблиця 7.).

Вирази функцій базису Буля (НЕ, І, АБО) через функцію Шеффера наведені також (таблиця 7.).

**Таблиця 7.16. Основні аксіоми і правила операцій у монобазисах**

Штрих Шеффера	Стрілка Пірса	Примітки
$a a = \bar{a}$	$a \downarrow a = \bar{a}$	Аксіома
$a \bar{a} = 1$	$a \downarrow \bar{a} = 0$	Аксіома
$a 0 = 1$	$a \downarrow 0 = \bar{a}$	Аксіома
$a 1 =$	$a \downarrow 1 = 0$	Аксіома
$\bar{a} 0 = 1$	$\bar{a} \downarrow 0 = a$	Аксіома
$\bar{a} 1 = a$	$\bar{a} \downarrow 1 = 0$	Аксіома
$\bar{a} = a a$	$\bar{a} = a \downarrow a$	НЕ з базису Буля
$ab = \overline{a b} = (a b) (a b)$	$ab = (a \downarrow a) \downarrow (b \downarrow b)$	І з базису Буля
$a \vee b = \overline{(a a) (b b)}$	$a \vee b = \overline{a \downarrow b} = (a \downarrow b) \downarrow (a \downarrow b) a$	АБО з базису Буля

## 7.8 Основні правила виконання операцій у монобазисі АБО-НЕ.

Функція Пірса (Вебба – « $\downarrow$ ») може описуватися виразами

$$a \downarrow b = \overline{a \vee b} = \bar{a} \cdot \bar{b}. \text{ Характерні для неї аксіоми містить таблиця 7..}$$

Вирази функцій базису Буля (НЕ, І, АБО) через функцію Пірса (Вебба) наведені там же (таблиця 7.).

На закінчення огляду основних властивостей логічних функцій підкреслимо, що логічні вирази, які містять операції диз'юнкції і кон'юнкції можна перетворювати (розкривати дужки, виносити загальний множник, переставляти місцями члени і т.д.) за правилами алгебри, вважаючи формально диз'юнкцію операцією додавання, а кон'юнкцію - операцією множення. Але потрібно завжди чітко пам'ятати, що в алгебрі логіки, на відміну від звичайної алгебри, знак  $+$  або знак  $\vee$  означають логічне зв'язування АБО (OR), а знак множення " $\cdot$ " або знаки  $\wedge$ , і  $\&$ , означають логічне зв'язування І (AND).

## 7.9 Аналітичне представлення функцій алгебри логіки

Існують різні способи представлення логічних функцій.

Представлення (опис) функції на словах. Наприклад: функція трьох аргументів приймає значення 1, якщо два будь-яких аргументи або всі три рівні 1. В всіх інших випадках функція рівна 0.

Табличний спосіб. Для представлення логічної функції можна використовувати, так званий, табличний спосіб, коли функція представляється своєю таблицею істинності. Приведемо приклад такої таблиці (таблиця 7.) для деякої логічної функції трьох аргументів  $f(a, b, c)$ .

**Таблиця 7.17. Приклад таблиці істинності**

№ набору	Набір			Значення функції $f$
	$a$	$b$	$c$	
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0

6	1	1	0	1
7	1	1	1	0

Звичайно в таблиці істинності стовпець з номером набору не наводиться.

Тепер розглянемо приклад простої таблиці істинності функцій двох змінних: І -  $f_1(a,b)$ , АБО -  $f_7(a,b)$ , виключне АБО -  $f_6(a,b)$ , які входять до складу базисів Буля та Жегалкіна (таблиця 7.).

**Таблиця 7.18. Таблиці істинності ФАЛ2 базисів Буля та Жегалкіна**

$a$	$b$	І, $f_1(a,b)$	АБО, $f_7(a,b)$	виключне АБО, $f_6(a,b)$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Алгебраїчний або аналітичний спосіб. Табличний спосіб максимально наочний, але у випадку складних функцій алгебри логіки (ФАЛ) досить некомпактний. Простіше виглядає аналітичний запис у вигляді формул. До розгляду аналітичної форми представлення ФАЛ уведемо кілька нових понять.

Змінні і їхн інверсії часто називають літералами.

Терм - це група літерал і констант, об'єднаних тим самим знаком логічного зв'язування: логічного додавання або ж логічного множення. У термі кожен літерал і кожна константа зустрічається тільки один раз, тобто в терм може входити або змінна, або її заперечення.

Диз'юнктивний терм (макстерм, елементарна диз'юнкція) - це логічна функція, що зв'язує всі літерали знаком диз'юнкції.

Наприклад:  $f_1 = a \vee \bar{b} \vee c \vee d$ ;  $f_2 = a \vee b$ .

Макстерм називають також конституентою нуля, тому що ця логічна функція дорівнює 0 тільки тоді, коли всі її літерали рівні 0 одночасно.

Кон'юнктивний терм (мінтерм, елементарна кон'юнкція) - це логічна функція, що зв'язує літерали знаком кон'юнкції.

Наприклад:  $f_1 = \bar{a}\bar{b}cd$ ;  $f_2 = bd$ .

Мінтерм називають також конституентою одиниці, тому що ця функція

дорівнює 1 тільки тоді, коли всі її літерали одночасно дорівнюють одиниці.

Ранг терму -  $r$ , визначається кількістю літерал, що входять у даний терм.

Наприклад, для мінтерму  $f = \bar{a}bcde$  ранг  $r = 5$ , а для макстерму  $f = a \vee b \vee \bar{c}$  ранг  $r = 3$ .

Будь-яка таблично задана ФАЛ може бути представлена аналітично у вигляді диз'юнкції скінченного числа мінтермів, на кожному з яких функція дорівнює одиниці (диз'юнктивна нормальна форма, ДНФ):

$$f(a,b,\dots,z) = F_1 \vee F_2 \vee \dots \vee F_n = \bigvee_{i=1}^n F_i, \quad (7.1)$$

де  $i$  - номери наборів, на яких функція дорівнює 1.

Приклад: записати в аналітичному вигляді функцію, задану таблично (таблиця 7.).

**Таблиця 7.19. Таблиця істинності функції трьох змінних**

№ набору	$a$	$b$	$c$	$f(a, b, c)$	Терми	
0	0	0	0	1	$F_0$	$\bar{a} \cdot \bar{b} \cdot \bar{c}$
1	0	0	1	0	$\Phi_1$	$a \vee b \vee \bar{c}$
2	0	1	0	0	$\Phi_2$	$a \vee \bar{b} \vee c$
3	0	1	1	1	$F_3$	$\bar{a} \cdot b \cdot c$
4	1	0	0	1	$F_4$	$a \cdot \bar{b} \cdot \bar{c}$
5	1	0	1	0	$\Phi_5$	$\bar{a} \vee b \vee \bar{c}$
6	1	1	0	0	$\Phi_6$	$\bar{a} \vee \bar{b} \vee c$
7	1	1	1	0	$\Phi_7$	$\bar{a} \vee \bar{b} \vee \bar{c}$

Згідно (7.1)

$$f(a,b,c) = F_0 \vee F_3 \vee F_4 = \bar{a} \cdot \bar{b} \cdot \bar{c} \vee \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot \bar{c}.$$

Тут символом  $F$  позначені елементарні кон'юнкції, а індексами вказані номери наборів, на яких функція приймає значення 1.

Треба відзначити, що функція, представлена таблицею істинності, може бути визначена не тільки її одиничними значеннями, але і нульовими.

Наприклад, у випадку попередньої функції для її інверсного значення маємо:

$$\overline{f(a,b,c)} = F_1 \vee F_2 \vee F_5 \vee F_6 \vee F_7.$$

Будь-яка таблично задана ФАЛ може бути задана аналітично у вигляді кон'юнкції скінченного числа макстермів, на кожному з яких функція дорівнює нулеві

(кон'юнктивна нормальна форма, КНФ):

$$f(a,b,\dots,z) = \Phi_1 \wedge \Phi_2 \wedge \dots \wedge \Phi_m = \bigwedge_{i=1}^m \Phi_i.$$

Наприклад, використовуючи правила де Моргана, результат попереднього рішення можна представити в наступному вигляді:

$$f(a,b,\dots,z) = \Phi_1 \wedge \Phi_2 \wedge \Phi_5 \wedge \Phi_6 \wedge \Phi_7 = (a \vee b \vee \bar{c})(a \vee \bar{b} \vee c)(\bar{a} \vee b \vee \bar{c})(\bar{a} \vee \bar{b} \vee c)(\bar{a} \vee \bar{b} \vee \bar{c}).$$

Тут символом  $\Phi$  позначені елементарні диз'юнкції, а індексами вказані номери наборів, на яких функція приймає значення 0.

Диз'юнктивна нормальна форма (ДНФ) це диз'юнктивне об'єднання мінтермів різних рангів, включаючи ранг, що дорівнює одиниці.

$$\text{Наприклад: } f(a,b,c) = c \vee a \cdot \bar{b} \vee b \cdot c \vee \bar{a} \cdot \bar{b} \cdot c.$$

Кон'юнктивна нормальна форма (КНФ) це кон'юнктивне об'єднання макстермів, що включає в себе макстерми різних рангів, включаючи ранг, що дорівнює одиниці.

$$\text{Наприклад: } f(a,b,c) = (a \vee b) \cdot (\bar{a} \vee b \vee \bar{c}) \cdot c.$$

Кожна логічна функція в загальному випадку може мати декілька ДНФ або КНФ.

Розрізняють також мінімальні ДНФ і КНФ логічних функцій. Зокрема, ДНФ заданої функції називається мінімальною, якщо кількість літер (літерал), що вона містить, буде не більше, ніж у будь-який інший ДНФ тієї же функції. Саме літер, а не змінних. Наприклад, ДНФ  $ab \vee a\bar{b}c \vee \bar{a} \cdot \bar{c}$  містить сім літер або літерал, але три змінні:  $a$ ,  $b$ ,  $c$ .

Мінімальна форма представлення ФАЛ це така форма, що містить мінімальну кількість термів, що мають мінімальні ранги.

Раніше ми розглянули процедуру формування аналітичної форми ФАЛ за її таблицею істинності. Тепер розглянемо зворотну процедуру: побудова таблиці істинності за логічним виразом.

Якщо в логічному виразі  $n$  аргументів, то для них у таблиці передбачається  $n$  колонок і один стовпчик для значень функції. Для  $2^n$  наборів аргументів виділяється така ж кількість рядків. Після цього, починаючи з нульового набору, кожен набір вписується в таблицю і для нього обчислюється значення функції, що також



заноситься до таблиці.

### 7.10 Досконалі нормальні форми

Нормальні кон'юнктивна і диз'юнктивна форми не дають однозначного представлення функції. Таке представлення виходить тільки при досконалих нормальних формах. Досконалу нормальну форму називають також стандартною або ж канонічною нормальною формою.

Досконала нормальна форма відрізняється від нормальної форми тим, що завжди містить терми тільки максимального рангу і дає однозначне представлення функції.

Кожна ФАЛ, крім абсолютно істинної функції, може бути представлена в єдиній досконалій кон'юнктивній нормальній формі (ДКНФ) або в єдиній досконалій диз'юнктивній нормальній формі (ДДНФ).

Отже, ДКНФ - це стандартний або канонічний логічний добуток макстермів максимального рангу даної функції, а ДДНФ - стандартна або канонічна логічна сума мінтермів максимального рангу даної функції.

ДДНФ функції знаходять у такий спосіб:

випикується ряд добутоків, тобто мінтермів, всіх аргументів і з'єднують їх знаками диз'юнкції; кількість добутоків повинна дорівнювати числу наборів, на яких задана функція дорівнює одиниці. У кожному мінтермі над аргументом, значення якого в даному наборі дорівнювало нулеві, ставиться знак заперечення.

ДКНФ функції формують у такий спосіб:

формується добуток диз'юнкцій, тобто макстермів, всіх аргументів з кількістю співмножників, рівним числу наборів, на яких задана функція обертається в нуль. У кожному макстермі над аргументом, рівним одиниці в даному наборі, ставиться знак заперечення.

Приклад: представити в ДДНФ функцію, задану таблично (таблиця 7.).

**Таблиця 7.20. Таблиця істинності функції трьох змінних**

№ набору	$a$	$b$	$c$	$f$	Терми	
0	0	0	0	0	$\Phi_0$	$a \vee b \vee c$
1	0	0	1	0	$\Phi_1$	$a \vee b \vee \bar{c}$
2	0	1	0	0	$\Phi_2$	$a \vee \bar{b} \vee c$

3	0	1	1	1	$F_3$	$\bar{a} \cdot b \cdot c$
4	1	0	0	1	$F_4$	$a \cdot \bar{b} \cdot \bar{c}$
5	1	0	1	0	$\Phi_5$	$\bar{a} \vee b \vee \bar{c}$
6	1	1	0	1	$F_6$	$a \cdot b \cdot \bar{c}$
7	1	1	1	0	$\Phi_7$	$\bar{a} \vee \bar{b} \vee \bar{c}$

Як мінтерми, як і в попередніх прикладах, відбираються рядки із значеннями аргументів, на яких функція дорівнює одиниці. У результаті одержуємо ДДНФ даної функції:

$$f(a,b,c) = \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot b \cdot \bar{c}.$$

Спосіб перетворення ДНФ у ДДНФ.

Довільна ДНФ переводиться в ДДНФ у такий спосіб.

Нехай  $f_{\text{днф}} = F_j$  - деякий мінтерм функції, представленої в ДНФ. Тоді цей мінтерм перетвориться в такий спосіб:

$$f_{\text{сднф}} = F_j (x \vee \bar{x}) = F_j x \vee F_j \bar{x},$$

де  $x$  - змінна, котра не входить у даний мінтерм  $F_j$ .

Якщо максимальний ранг для функції дорівнює  $r$ , а мінімальний ранг  $j$ -го мінтерму  $F_j$  дорівнює  $k$ , то попереднє перетворення необхідно застосувати до  $j$ -го мінтерму  $(r - k)$  разів кожного разу стосовно нової змінної.

Приклад. Логічну функцію  $f(a, b, c, d)$ , задану в ДНФ, перетворити в ДДНФ.

$$f(a,b,c,d) = a \cdot \bar{b} \vee b \cdot \bar{c} \cdot d \vee \bar{a} \cdot \bar{c} \cdot d \vee a \cdot b \cdot c \cdot d = F_1 \vee F_2 \vee F_3 \vee F_4.$$

Тому що терм  $F_4$ , як ми бачимо, має максимальний ранг і в нього входять усі змінні функції, то скористаємося прийомом перетворення  $f_{\text{сднф}} = F_j (x \vee \bar{x}) = F_j x \vee F_j \bar{x}$  по черзі до інших трьох термів:

$$F_1 = a \cdot \bar{b} \cdot (c \vee \bar{c}) = a \cdot \bar{b} \cdot c \vee a \cdot \bar{b} \cdot \bar{c}.$$

Обидва члени отриманого виразу помножимо на  $(d \vee \bar{d})$ .

У результаті одержимо:

$$F_1 = a \cdot \bar{b} \cdot c \cdot d \vee a \cdot \bar{b} \cdot \bar{c} \cdot d \vee a \cdot \bar{b} \cdot c \cdot \bar{d} \vee a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}.$$

Аналогічно,

$$F_2 = b \cdot \bar{c} \cdot d \cdot (a \vee \bar{a}) = a \cdot b \cdot \bar{c} \cdot d \vee \bar{a} \cdot b \cdot \bar{c} \cdot d.$$

$$F_3 = \bar{a} \cdot \bar{c} \cdot d \cdot (b \vee \bar{b}) = \bar{a} \cdot b \cdot \bar{c} \cdot d \vee \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d.$$

Після приведення подібних членів визначений ДДНФ даної функції  $f(a, b, c, d) = a \cdot \bar{b} \cdot c \cdot d \vee a \cdot \bar{b} \cdot \bar{c} \cdot d \vee a \cdot \bar{b} \cdot c \cdot \bar{d} \vee a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \vee a \cdot b \cdot \bar{c} \cdot d \vee \bar{a} \cdot b \cdot \bar{c} \cdot d \vee \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d \vee a \cdot b \cdot c \cdot d$ .

Довільна КНФ переводиться в ДКНФ шляхом наступного перетворення. Нехай макстерм  $f_{\text{КНФ}} = \Phi_j$ . Тоді  $f_{\text{СКНФ}} = \Phi_j \vee x \cdot \bar{x} = (\Phi_j \vee x) \cdot (\Phi_j \vee \bar{x})$ .

Приклад. Перетворити в ДКНФ логічну функцію:

$$f(a, b, c) = (a \vee b)(b \vee \bar{c})(a \vee b \vee c) = \Phi_1 \cdot \Phi_2 \cdot \Phi_3.$$

Застосовуємо вищезгадане правило перетворень по черзі до макстермів  $\Phi_1$  і  $\Phi_2$ , тому що макстерм  $\Phi_3$ , як ми бачимо, має максимальний ранг:

$$\Phi_1 = (a \vee b) \vee c \bar{c} = (a \vee b \vee c)(a \vee b \vee \bar{c}),$$

$$\Phi_2 = (b \vee \bar{c}) \vee a \bar{a} = (a \vee b \vee \bar{c})(\bar{a} \vee b \vee \bar{c}).$$

Після спрощень ДКНФ функції прийме остаточний вид:

$$f(a, b, c) = (a \vee b \vee c)(a \vee b \vee \bar{c})(\bar{a} \vee b \vee \bar{c}).$$

Нарешті, наведемо ДКНФ і ДДНФ для елементарних логічних функцій двох аргументів (таблиця 7.).

**Таблиця 7.21. Досконалі форми для ФАЛ2**

ФАЛ2	ab				Позначення функції	ДДНФ функції	ДКНФ функції
	00	01	10	11			
$f_0$	0	0	0	0	0	немає	$(a \vee b)(a \vee \bar{b})(\bar{a} \vee b)(\bar{a} \vee \bar{b})$
$f_1$	0	0	0	1	$ab$	$a \cdot b$	$(a \vee b)(a \vee \bar{b})(\bar{a} \vee b)$
$f_2$	0	0	1	0	$\overline{a \rightarrow b}$	$a \cdot \bar{b}$	$(a \vee b)(a \vee \bar{b})(\bar{a} \vee \bar{b})$
$f_3$	0	0	1	1	$a$	$a \cdot \bar{b} \vee a \cdot b$	$(a \vee b)(a \vee \bar{b})$
$f_4$	0	1	0	0	$\overline{a \leftarrow b}$	$\bar{a} \cdot b$	$(a \vee b)(\bar{a} \vee b)(\bar{a} \vee \bar{b})$
$f_5$	0	1	0	1	$b$	$\bar{a} \cdot b \vee a \cdot b$	$(a \vee b)(\bar{a} \vee b)$
$f_6$	0	1	1	0	$a \oplus b$	$\bar{a} \cdot b \vee a \cdot \bar{b}$	$(a \vee b)(\bar{a} \vee \bar{b})$
$f_7$	0	1	1	1	$a \vee b$	$\bar{a} \cdot b \vee a \cdot \bar{b} \vee a \cdot b$	$(a \vee b)$
$f_8$	1	0	0	0	$a \downarrow b$	$\bar{a} \cdot \bar{b}$	$(a \vee \bar{b})(\bar{a} \vee b)(\bar{a} \vee \bar{b})$
$f_9$	1	0	0	1	$a \equiv b$	$\bar{a} \cdot \bar{b} \vee a \cdot b$	$(a \vee \bar{b})(\bar{a} \vee b)$
$f_{10}$	1	0	1	0	$\bar{b}$	$\bar{a} \cdot \bar{b} \vee a \cdot \bar{b}$	$(a \vee \bar{b})(\bar{a} \vee \bar{b})$
$f_{11}$	1	0	1	1	$a \leftarrow b$	$\bar{a} \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot b$	$(a \vee \bar{b})$

$f_{12}$	1	1	0	0	$\bar{a}$	$\bar{a} \cdot \bar{b} \vee \bar{a} \cdot b$	$(\bar{a} \vee b)(\bar{a} \vee \bar{b})$
$f_{13}$	1	1	0	1	$a \rightarrow b$	$\bar{a} \cdot \bar{b} \vee \bar{a} \cdot b \vee a \cdot b$	$(\bar{a} \vee b)$
$f_{14}$	1	1	1	0	$a b$	$\bar{a} \cdot \bar{b} \vee \bar{a} \cdot b \vee \bar{a} \cdot b$	$(\bar{a} \vee \bar{b})$
$f_{15}$	1	1	1	1	1	$\bar{a} \cdot \bar{b} \vee \bar{a} \cdot b \vee \bar{a} \cdot b \vee a \cdot b$	немає

### 7.11 Поліном Жегалкіна

Будь-яку логічну функція можна представити багаточленом (поліномом) Жегалкіна виду:

$$a_0 \oplus a_1 \cdot x_1 \oplus a_2 \cdot x_2 \oplus \dots \oplus a_{n-1} \cdot x_{n-1} \oplus a_1 \cdot x_1 \cdot x_1 \oplus \dots \oplus a_1 \cdot x_1 \cdot x_{n-1} \oplus \dots \oplus a_1 \cdot x_1 \cdot x_2 \cdot \dots \cdot x_{n-1},$$

де  $a_0, a_1, \dots, a_{n-1}$ , - деякі константи, рівні нулеві або одиниці; знак  $\oplus$ , як відомо, означає операцію додавання за модулем два.

При записі конкретної логічної функції у вигляді полінома Жегалкіна коефіцієнти  $a_0, a_1, \dots, a_{n-1}$ , випадають, тому що члени, при яких ці коефіцієнти дорівнюють нулеві, можна опустити, а коефіцієнти, рівні одиниці, не писати.

Щоб логічну функцію, задану таблицею її значень, представити у вигляді багаточлена, потрібно цю функцію записати у вигляді суми конститuent одиниці, рівних одиниці на тих же наборах, на яких дорівнює одиниці задана функція. Потім всі аргументи, що входять в отриманий вираз із запереченням, замінити за допомогою співвідношення  $\bar{x} = x \oplus 1$ , розкрити дужки і привести подібні члени з урахуванням, що в даному випадку  $x \oplus x \oplus \dots \oplus x = x$ , якщо кількість  $x$  непарна і  $x \oplus x \oplus \dots \oplus x = 0$ , якщо кількість  $x$  парна.

Приклад. Представити функцію  $f_{14}(a,b)$  (таблиця 7.) у вигляді полінома Жегалкіна.

Ця функція дорівнює одиниці на нульовому, першому і другому наборах, тому

$$f_{14}(a,b) = F_0 \oplus F_1 \oplus F_2 = \bar{a} \cdot \bar{b} \oplus \bar{a} \cdot b \oplus a \cdot \bar{b}.$$

Використовуючи співвідношення  $a \oplus 1 = \bar{a}$  і  $a(b \oplus c) = a \cdot b \oplus a \cdot c$ , після всіх перетворень одержимо  $f_{14}(a,b) = 1 \oplus a \oplus a \oplus b \oplus b \oplus ab \oplus ab \oplus ab$ .

Після зведення подібних членів, остаточно одержуємо:  $f_{14}(a,b) = 1 \oplus ab$ .

### 7.12 Класи функцій алгебри логіки

Клас функцій алгебри логіки – це набір таких функцій алгебри логіки, з яких

шляхом суперпозиції можна одержати тільки функції цього ж набору.

Існує 5 класів ФАЛ, що важливі для логічного аналізу.

Клас лінійних функцій (Кл). Логічна функція називається лінійною, якщо вона представляється поліномом Жегалкіна першого степеня:

$$f(x_0, x_1, \dots, x_{n-1}) = a_0 \oplus a_1 \cdot x_1 \oplus a_2 \cdot x_2 \oplus \dots \oplus a_{n-1} \cdot x_{n-1},$$

де коефіцієнти  $a_i$  дорівнюють нулеві або одиниці.

Клас функцій, що зберігають нуль (К0). Якщо функція на нульовому наборі змінних дорівнює нулеві, то говорять, що функція зберігає нуль:

$$f(0, 0, \dots, 0) = 0.$$

Клас функцій, що зберігають одиницю (К1). Якщо функція на одиничному наборі змінних дорівнює одиниці, то говорять, що така функція зберігає одиницю:

$$f(1, 1, \dots, 1) = 1.$$

Клас монотонних функцій (Км). Для початку введемо необхідні визначення.

Якщо значення кожного аргументу одного набору більше або дорівнює значенню того ж аргументу другого набору, то говорять, що перший набір не менше другого. Такі набори називаються порівнянними.

Якщо ж деякі зі значень аргументів першого набору більше або рівні, а інші менше значень тих же аргументів другого набору, то такі набори називаються непорівнянними.

Приклад порівнянних наборів:

$$1,1,0,1 > 1,1,0,0; 0,1,1,0,1 > 0,1,0,0,0; 1,0,1,1 > 1,0,0,0.$$

Приклади непорівнянних наборів: 0,1 і 1,0; 1,0,0 і 0,0,1; 1,1,1,0 і 1,0,0,1.

Функція називається монотонною, якщо при будь-якому зростанні сусідніх наборів значення цієї функції не зменшуються.

Клас самодвоїстих функцій (Кс).

Спершу дамо визначення протилежних наборів.

Два набори називаються протилежними, якщо всі значення аргументів одного набору протилежні значенням аргументів іншого набору. Наприклад: 1,0,1,1,0,0 і 0,1,0,0,1,1; 1,1,1,1 і 0,0,0,0, тобто, для того, щоб одержати протилежний набір, досить замінити в даному наборі нулі одиницями, а одиниці - нулями.

Логічна функція називається самодвоїстою, якщо на кожній парі протилежних наборів вона приймає протилежні значення, тобто

$$f(a, b, \dots, z) = \overline{f(\overline{a}, \overline{b}, \dots, \overline{z})}.$$

Усі перераховані класи логічних функцій мають наступну властивість: будь-яка логічна функція, отримана за допомогою суперпозиції і підстановки з функцій одного класу, обов'язково буде належати цьому ж класові.

Функції  $f_8(a, b)$  (функція Пірса) і  $f_{14}(a, b)$  (функція Шеффера) не належать жодному з зазначених класів.

### 7.13 Функціонально повні системи функцій алгебри логіки

Базисом (або функціонально повною системою функцій алгебри логіки) називається набір функцій алгебри логіки, з функцій, що входять до складу якого, методом суперпозиції можна представити будь-яку ФАЛ.

Для того щоб система ФАЛ була функціонально повною необхідно і досить, щоб вона містила хоча б одну функцію, яка:

- не зберігає нуль,
- не зберігає одиницю,
- не є лінійною,
- не є монотонною,
- не є самодвоїстою.

До базисів відноситься система функцій I, АБО, НЕ (базис 1). Базисами є також системи, що містять функції I, НЕ (базис 2), АБО, НЕ (базис 3), що складаються з функції Шеффера (I-НЕ) (базис 4) або функції Пірса (Вебба) (АБО-НЕ) (базис 5). Це перерахування показує, що базиси можуть бути надлишковими (базис 1) і мінімальними (базиси 4 і 5). Базис є мінімальним, якщо видалення хоча б однієї функції перетворює систему ФАЛ у неповну.

Проблема найпростішого представлення логічних функцій зводиться до вибору не тільки базису, але і форми найбільш ощадливого представлення цих функцій.

Як уже відзначалося, кожна з логічних функцій, що входить до складу базисів, апаратно реалізується за допомогою певних електронних логічних елементів.

Всі інші логічні функції можна одержати на основі відомих правил алгебри логіки з функцій, що входять у базис. Отже, з електронних логічних елементів, що реалізують функції базисів, можна побудувати будь-який логічний пристрій (комп'ютер).

Наприклад, досить мати один тип логічного елемента І-НІ або ІЛІ-НІ, щоб на його основі побудувати все різноманіття логічних схем. Однак для реалізації складних ФАЛ такий спосіб може зажадати великого числа корпусів логічних елементів. Тому на практиці, при необхідності, використовують і інші логічні елементи, що реалізують більш складні логічні функції.

#### 7.14 Числове та геометричне подання логічних функцій

Як уже відзначалося, для спрощення запису логічних функцій замість повного перерахування термів часто використовують тільки номери наборів, для яких функція приймає одиничне значення. Наприклад, якщо деяка, задана таблично, функція  $f(a, b, c)$  приймає значення одиниці на наборах з номерами 0, 3, 4 і 6, то її можна представити в такий спосіб:

$$f(a, b, c) = \vee F(0, 3, 4, 6).$$

Якщо ця ж функція на наборах 1, 2, 5, 7 - приймає значення 0, то її можна представити так:

$$f(a, b, c) = \wedge F(1, 2, 5, 7).$$

Таку форму запису називають числовою.

Перший вид такої форми використовують, коли функція представляється в ДДНФ, а другий - коли в ДКНФ.

Багато перетворень логічних функцій, іноді зручно виконувати з використанням їхніх геометричних представлень. Наприклад, функцію двох змінних можна інтерпретувати як деяку площину, задану в системі координат  $a, b$ . Якщо відкласти на кожній осі одиничні відрізки  $a$  і  $b$ , то вийде квадрат, вершини якого відповідають комбінаціям змінних (рис. 7.20).

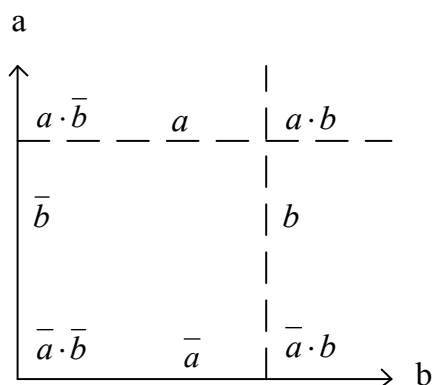


Рис. 7.20. Геометричне представлення ФАЛ

Звідси випливає, що дві вершини, які належать тому самому ребру і які називаються сусідніми, "склеюються" за змінною, яка змінює своє значення вздовж цього ребра. Наприклад,

$$\bar{a} \cdot \bar{b} \vee a \cdot \bar{b} = \bar{b},$$

так як нам уже відомо із властивостей логічних функцій, що

$$\bar{a} \cdot \bar{b} \vee a \cdot \bar{b} = (\bar{a} \vee a) \bar{b} = 1 \cdot \bar{b} = \bar{b}.$$

Для функцій трьох змінних геометричне представлення виконують у вигляді тривимірного куба. Ребра куба поглинають вершини. Грані куба поглинають свої ребра і, отже, вершини.

У випадку ж чотирьох перемінних - "чотиривимірного" куба. У геометричному значенні кожний набір змінних  $x_0, x_1, \dots, x_{n-1}$  можна розглядати як  $n$ -мірний вектор, що визначає точку  $n$ -мірного простору. Тому вся безліч наборів, на яких визначена функція  $n$  змінних, представляється у вигляді вершин  $n$ -мірного куба. Координати вершин куба вказуються в порядку, що відповідає порядкові перерахування змінних у записі функції. Відзначаючи точками вершини, у яких функція приймає значення, рівне одиниці, одержуємо геометричне представлення ФАЛ.



## 8 СПРОЩЕННЯ І МІНІМІЗАЦІЯ ЛОГІЧНИХ ФУНКЦІЙ

### 8.1 Задача мінімізації

Як уже відзначалося, складність логічного виразу визначається числом букв, що входять у цей вираз, тобто числом змінних і їхніх інверсій. Нагадаємо, що змінні і їхньої інверсії часто називають літералами. Зокрема, вираз у формі логічної суми мінтермів з мінімальним числом літерал називають мінімальною сумою, тобто мінімальною ДНФ (МДНФ). Мінімальним числом літерал повинні відрізнятися і мінімальні КНФ (МКНФ), тобто мінімальні добутки макстермів.

Очевидно, що складність логічної функції, а звідси складність і вартість реалізуючої її схеми пропорційні числу логічних операцій і числу входжень змінних або їхніх заперечень. У принципі будь-яка логічна функція може бути спрощена безпосередньо за допомогою аксіом і теорем логіки, але, як правило, такі перетворення вимагають громіздких викладень. До того ж процес спрощення булевих виразів не є алгоритмічним. Тому більш доцільно використовувати спеціальні алгоритмічні методи мінімізації, що дозволяють проводити спрощення функції більш просто, швидко і безпомилково. До таких методу відносяться, наприклад, метод Квайну, метод карт Карно (діаграми Вейча), метод Квайна-Мак-Класкі й ін. Ці методи найбільш придатні для звичайної технічної практики, особливо мінімізація логічної функції з використанням карт Карно. Метод карт Карно зберігає наочність при числі змінних не більш шести. При досить рідких випадках, коли число аргументів більше шести, звичайно використовують метод Квайна-Мак-Класкі. У зв'язку з тим, що для звичайної практики метод Карно найбільш оптимальний, надалі, під час обговорення методів мінімізації логічних функцій, в основному, буде описуватися і застосовуватися цей метод.

У процесі мінімізації тої або іншої логічної функції, звичайно, враховується в якому базисі ефективніше буде реалізувати її мінімальну форму за допомогою електронних схем.

Розглянемо деякі визначення, що будуть використовуватися надалі.

Добуток декількох змінних, узятих із запереченнями або без них, тобто мінтерм, називається також елементарним добутком, а диз'юнкція змінних, частина з яких може

мати заперечення, тобто макстерм, називається елементарною диз'юнкцією.

Тоді ДНФ функції є логічна сума будь-яких елементарних добутоків, тобто мінтермів, а КНФ - добуток будь-яких елементарних диз'юнкцій, тобто макстермів.

Два доданки функції, представлені в ДДНФ, або взагалі два будь-яких терми, що відрізняються тільки однею змінною (в одному вона має заперечення, а в іншому - немає) називаються сусідніми.

Тепер уведемо поняття накриття для логічних функцій.

Нехай на якому-небудь наборі аргументів функція  $f$  приймає значення  $a_1$ , а функція  $\varphi$  на цьому ж наборі приймає значення  $a_2$ . Тоді говорять, що функція  $f$  на даному наборі накриває значення  $a_2$  функції  $\varphi$  своїм значенням  $a_1$ .

Так, наприклад, у ДДНФ кожна одиниця заданої логічної функції накривається одиницею тільки одного мінтерму. Тому кількість мінтермів, що входять у ДДНФ, дорівнює числу наборів, на яких функція дорівнює одиниці. Наприклад, ДДНФ функції  $f_{11}(a,b)$  (таблиця 7.) має вигляд:  $f_{11}(a,b) = \bar{a} \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot b$ . Незавжди побачити, що кожен мінтерм, що входить у цю форму, накриває тільки одну одиницю функції.

Якщо деяка логічна функція  $\varphi$  (в окремому випадку - елементарний добуток) дорівнює нулеві на тих же наборах, на яких дорівнює нулеві інша функція  $f$ , то говорять, що функція  $\varphi$  входить до функції  $f$ . Іншими словами, функція  $\varphi$  входить до функції  $f$  тоді, коли вона накриває нулями всі нулі функції  $f$ , а одиниці функції  $f$  можуть бути накриті як нулями, так і одиницями функції  $\varphi$ . Вхідження позначається:  $\varphi \subset f$ . Наприклад, у функцію  $f_6(a,b)$  (таблиця 7.) будуть входити всі функції, що приймають нульові значення на наборах 0,0 і 1,1, тобто функції  $f_4(a,b) = \bar{a} \cdot b$ ,  $f_2(a,b) = a \cdot \bar{b}$  і  $f_0(a,b) = 0$ . Очевидно, що константа нуль входить в усі функції, а в константу одиниця входять усі функції.

Функцію  $\varphi$ , що входить у дану функцію  $f$ , називають її імплікантою.

Простими імплікантами логічної функції  $f$  називають такі елементарні, які самі входять до даної функції, але ніяка власна частина цих добутоків не входить до функції  $f$ .

Власною частиною називають добуток, отриманий шляхом виключення з даного

добутку одного або декількох співмножників. Наприклад, добуток  $a \cdot \bar{b} \cdot c$  має такі власні частини:  $a \cdot \bar{b}$ ,  $\bar{b} \cdot c$ ,  $ac$ ,  $a$ ,  $\bar{b}$ ,  $c$ .

Прості імпліканти являють собою найбільш короткі елементарні добутки, що входять до даної логічної функції.

Для того, щоб знайти прості імпліканти логічної функції треба виписати всі елементарні добутки, що входять до даної функції, і вибрати з їхнього числа ті, власні частини яких у цю функцію не входять.

Приклад. Розглянемо логічну функцію трьох аргументів (таблиця 8.).

**Таблиця 8.1. Логічна функція трьох аргументів**

$a$	$b$	$c$	$f(a,b,c)$	Елементарні добутки
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{a} \cdot b \cdot c$
1	0	0	1	$a \cdot \bar{b} \cdot \bar{c}$
1	0	1	1	$a \cdot \bar{b} \cdot c$
1	1	0	0	
1	1	1	0	

У цю функцію входять чотири елементарних добутки:  $\bar{a} \cdot b \cdot c$ ,  $a \cdot \bar{b} \cdot \bar{c}$ ,  $a \cdot \bar{b} \cdot c$ ,  $a \cdot \bar{b}$ . Простою підстановкою значень аргументів неважко перевірити, що ці добутки звертаються в нуль на всіх наборах, на яких дорівнює нулеві задана функція. Простими імплікантами будуть тільки два добутки:  $\bar{a} \cdot b \cdot c$  і  $a \cdot \bar{b}$ . А власна частина  $a \cdot \bar{b}$  добутків, що залишилися, входить у задану функції, тому вони не є елементарними.

Будь-яка логічна функція дорівнює диз'юнкції усіх своїх простих імплікант. Диз'юнкція всіх простих імплікант називається скороченою диз'юнктивною нормальною формою логічної функції.

Прості імпліканти накривають одиницями не одну, а кілька одиниць заданої функції. Наприклад, проста імпліканта  $a$  функції  $f_{11}(a,b)$  обертається разом з цією функцією в одиницю на наборах 1,0 і 1,1, а проста імпліканта  $\bar{b}$  - на наборах 0,0 і 1,0. Тому прості імпліканти  $a$  і  $\bar{b}$  спільно накривають одиницями всі одиниці функції  $f_{11}(a,b)$ , що внаслідок цього може бути представлена у формі  $f_{11}(a,b) = a \vee \bar{b}$ , що є

скороченої ДНФ функції  $f_{11}(a,b)$ .

Нижче наведено прості імпліканти і скорочені ДНФ логічних функцій двох аргументів (таблиця 8.).

**Таблиця 8.2. Прості імпліканти логічних функцій двох аргументів**

Логічні функції	Елементарні добутки, що входять у функцію	Прості імпліканти	Скорочена диз'юнктивна форма
$f_0$	Немає	Немає	0
$f_1$	$ab$	$ab$	$ab$
$f_2$	$a \cdot \bar{b}$	$a \cdot \bar{b}$	$a \cdot \bar{b}$
$f_3$	$a, a \cdot \bar{b}, ab$	$a$	$a$
$f_4$	$\bar{a} \cdot b$	$\bar{a} \cdot b$	$\bar{a} \cdot b$
$f_5$	$\bar{a} \cdot b, ab, b$	$b$	$b$
$f_6$	$\bar{a} \cdot b, a \cdot \bar{b}$	$\bar{a} \cdot b, a \cdot \bar{b}$	$\bar{a} \cdot b \vee a \cdot \bar{b}$
$f_7$	$\bar{a} \cdot b, a \cdot \bar{b}, ab, a, b$	$a, b$	$a \vee b$
$f_8$	$\bar{a} \cdot \bar{b}$	$\bar{a} \cdot \bar{b}$	$\bar{a} \cdot \bar{b}$
$f_9$	$\bar{a} \cdot \bar{b}, ab$	$\bar{a} \cdot \bar{b}, ab$	$\bar{a} \cdot \bar{b} \vee a \cdot b$
$f_{10}$	$\bar{a} \cdot \bar{b}, a \cdot \bar{b}, \bar{b}$	$\bar{b}$	$\bar{b}$
$f_{11}$	$\bar{b}, a, ab, \bar{a} \cdot \bar{b}, a \cdot \bar{b}$	$a, \bar{b}$	$a \vee \bar{b}$
$f_{12}$	$\bar{a} \cdot \bar{b}, \bar{a} \cdot b, \bar{a}$	$\bar{a}$	$\bar{a}$
$f_{13}$	$\bar{a}, \bar{a} \cdot \bar{b}, b, \bar{a} \cdot b, ab$	$b, \bar{a}$	$\bar{a} \vee b$
$f_{14}$	$\bar{a}, \bar{b}, \bar{a} \cdot \bar{b}, \bar{a} \cdot b, a \cdot \bar{b}$	$\bar{a}, \bar{b}$	$\bar{a} \vee \bar{b}$
$f_{15}$	$\bar{a}, \bar{b}, a, b, \bar{a} \cdot b, a \cdot \bar{b}, \bar{a} \cdot \bar{b}, ab$	$a, b, \bar{a}, \bar{b}$	немає

Тепер розглянемо процес спрощення логічних виразів з використанням теорем і правил алгебри логіки.

Найбільш ефективними для цього процесу є закон поглинання і закон склеювання. Нагадаємо, що в застосуванні до функції в ДДНФ ці закони мають вигляд:  $A \cdot B \vee \bar{A} \cdot B = B$ ,  $A \vee A \cdot B = A$ , де  $A$  і  $B$  - будь-які функції.

Крім того, застосовуються правила де Моргана, винесення спільних членів за дужки, метод іспиту членів і т. ін.

Розглянемо два простих приклади.

Дано деяку функцію в ДДНФ:

$$f(a,b,c) = \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot c \vee a \cdot b \cdot \bar{c} \vee a \cdot b \cdot c.$$

Четвертий доданок є сусіднім з будь-яким з перших трьох, тому є зміст додати

цей доданок ще два рази. Тоді одержимо:

$$f(a, b, c) = \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot c \vee a \cdot b \cdot \bar{c} \vee a \cdot b \cdot c \vee a \cdot b \cdot c \vee a \cdot b \cdot c.$$

Шляхом попарного склеювання одержимо:

$$f(a, b, c) = a \cdot b \vee b \cdot c \vee a \cdot c.$$

Це вираз є формою глухого кута, тому що не має надлишкових доданків.

Розглянемо другий приклад.

Спростимо вираз:

$$f(a, b, c) = \bar{a} \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot b \vee b \cdot c.$$

Спрощення провадиться в такий спосіб:

$$\begin{aligned} f(a, b, c) &= \bar{a} \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot b \vee b \cdot c = \\ &= \bar{a} \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot b \vee b \cdot c \quad (\text{за правилом: } x \vee x = x) = \\ &= \bar{a} \cdot \bar{b} \vee \bar{b} \cdot a \vee a \cdot \bar{b} \vee a \cdot b \vee b \cdot c \quad (\text{за правилом: } xy = yx) = \\ &= \bar{b}(\bar{a} \vee a) \vee a(\bar{b} \vee b) \vee b \cdot c \quad (\text{за правилом: } x(y \vee z) = xy \vee xz) = \\ &= \bar{b}(a \vee \bar{a}) \vee a(b \vee \bar{b}) \vee b \cdot c \quad (\text{за правилом: } x \vee y = y \vee x) = \\ &= \bar{b} \cdot 1 \vee a \cdot 1 \vee b \cdot c \quad (\text{за правилом: } x \vee \bar{x} = 1) = \\ &= \bar{b} \vee a \vee b \cdot c \quad (\text{за правилом: } x \cdot 1 = x) = \\ &= a \vee \bar{b} \vee b \cdot c \quad (\text{за правилом: } x \vee y = y \vee x) = \\ &= a \vee \bar{b} \vee c \quad (\text{за правилом: } x \vee \bar{x} \cdot y = x \vee y). \end{aligned}$$

У такий спосіб одержуємо:  $f(a, b, c) = \bar{a} \cdot \bar{b} \vee a \cdot \bar{b} \vee a \cdot b \vee b \cdot c = a \vee \bar{b} \vee c$ .

У функції, представлений в ДДНФ, відшукуються всі сусідні доданки і провадиться їхнє склеювання. Можна застосувати і групове склеювання, тобто у виразі відшукуються доданки, що мають спільні змінні, котрі виносяться за дужки, а в дужках залишається група доданків більш низького рангу. Якщо ця група складається з  $2^k$  доданків  $k$ -го рангу, то вона являє собою повну досконалу форму для  $k$  змінних і, отже, дорівнює одиниці. У результаті залишається один доданок, що містить змінні, винесені за дужки. Для зручності склеювання, у тому числі і групового, якщо це необхідно, додають доданки, що вже є у вихідному виразі.

Після того, як можливості склеювання вичерпані, робиться спроба виключити надлишкові доданки застосуванням закону поглинання і, при необхідності, методу

іспиту членів, що полягає в наступному. Попередньо виконуються всі можливі згадані процедури склеювання і поглинання і тим самим утворюється скорочена ДНФ. Далі виконується сам метод іспиту кожного члена отриманого виразу. Щоб випробувати деякий член варто виключити його зі скороченої ДНФ і підставити у вираз, що залишився, такі значення перемінних, при яких виключений член обертається в одиницю. Якщо при цьому вираз теж тотожно дорівнює одиниці, то випробовуваний член є зайвим. Після того, як будуть відкинуті в такий спосіб усі зайві доданки, утворюється ДНФ глухого кута.

Приклад. Знайти форми глухого кута логічної функції, заданої скороченою ДНФ

$$f(a, b, c) = \bar{a} \cdot \bar{b} \vee a \cdot c \vee \bar{b} \cdot c.$$

1) Випробуємо член  $\bar{a} \cdot \bar{b}$ , для чого підставимо у вираз  $a \cdot c \vee \bar{b} \cdot c$  значення перемінних  $a = 0, b = 0$  (при цьому  $\bar{a} \cdot \bar{b} = \bar{0} \cdot \bar{0} = 1$ ):

$$0 \cdot c \vee \bar{0} \cdot c = c.$$

Тому що цей вираз не дорівнює тотожно 1, то член  $\bar{a} \cdot \bar{b}$  виключати не можна.

2) Випробуємо член  $ac$ . Підставляючи у вираз  $\bar{a} \cdot \bar{b} \vee \bar{b} \cdot c$  значення  $a = 1$  і  $c = 1$  ( $ac = 1 \cdot 1 = 1$ ), переконуємося, що його виключити не можна:

$$\bar{1} \cdot \bar{b} \vee \bar{b} \cdot 1 = \bar{b}.$$

3) Випробуємо член  $\bar{b} \cdot c$ , для чого підставимо у вираз  $\bar{a} \cdot \bar{b} \vee a \cdot c$  значення перемінних  $b = 0, c = 1$  ( $\bar{b} \cdot c = \bar{0} \cdot 1 = 1 \cdot 1 = 1$ ):

$$\bar{a} \cdot \bar{0} \vee a \cdot 1 = \bar{a} \cdot 1 \vee a \cdot 1 = \bar{a} \vee a = 1 \quad \bar{a} \cdot \bar{0} \vee a \cdot 1 = \bar{a} \cdot 1 \vee a \cdot 1 = \bar{a} \vee a = 1.$$

Отриманий вираз тотожно дорівнює 1, тому член  $\bar{b} \cdot c$  можна виключити.

Відповідь. Задана функція має одну форму глухого кута, як є мінімальною:

$$f(a, b, c) = \bar{a} \cdot \bar{b} \vee a \cdot c.$$

Виключення надлишкових доданків може бути зроблено різними способами. Тому для однієї і тої ж функції може існувати кілька форм глухого кута. Якщо усі вони знайдені, то серед них обирається мінімальна.

## 8.2 Метод Квайна й імплікантні матриці

Метод одержання скороченої диз'юнктивної нормальної форми логічної функції називається методом Квайна.

При мінімізації за методом Квайна в базисі Буля передбачається, що вихідна функція задана в ДДНФ.

Нагадаємо, що імпліканта функції - це деяка логічна функція, що звертається в нуль при наборі змінних, на яких сама функція також дорівнює нулеві.

Тому будь-який мінтерм у складі ДДНФ, або група мінтермів, з'єднаних знаками диз'юнкції, є імплікантами вихідної ДНФ.

Первинна або проста імпліканта функції - це імпліканта типу елементарної кон'юнкції деяких перемінних, ніяка частина якої вже не є імплікантою даної функції.

Диз'юнкція простих імплікант, ні одну з яких виключити не можна, називається ДНФ глухого кута заданої функції. Деякі функції мають кілька форм глухого кута. Форми глухого кута, що містять найменшу кількість букв, будуть мінімальними.

Задача мінімізації за методом Квайна складається в попарному порівнянні всіх імплікант, що входять у ДДНФ, з метою виявлення можливості поглинання якоїсь змінної:

$$F \cdot x_i \vee F \cdot \bar{x}_i = F.$$

Таким чином, удається знизити ранг термів. Ця процедура проводиться доти, поки не залишиться жодного члена, що допускає поглинання з яким-небудь іншим термом. Терми, які поглинулися, відзначаються. Невідмічені терми являють собою первинні імпліканти.

Отриманий логічний вираз не завжди виявляється мінімальним. Тому досліджується можливість подальшого спрощення. Для цього складається таблиця, у рядках якої записуються знайдені первинні імпліканти, а в стовпцях указуються терми вихідного рівняння. Клітки цієї таблиці відзначаються у випадку, якщо первинна імпліканта входить до складу якого-небудь терму. Після цього задача спрощення зводиться до того, щоб знайти таку мінімальну кількість первинних імплікант, що покривають усі стовпці.

У цьому методі використовуються операції неповного склеювання (повним склеюванням, як відомо, буде:  $a \cdot b \vee a \cdot \bar{b} = a$ ) і поглинання ( $a \vee a \cdot b = a$ ). Застосовувана в методі Квайна операція неповного склеювання визначається формулою:  $a \cdot b \vee a \cdot \bar{b} = a \vee a \cdot b \vee a \cdot \bar{b}$ . Помітимо, що в правій частині рівності, крім члена  $a$ ,

отриманого в результаті повного склеювання, залишаються обидва члени, що беруть участь у склеюванні.

Теорема Квайна. Якщо в досконалій диз'юнктивній нормальній формі логічної функції провести всі операції неповного склеювання і потім всі операції поглинання, то в результаті виходить скорочена диз'юнктивна нормальна форма цієї функції, тобто диз'юнкція всіх її простих імплікант.

Метод Квайна виконується в кілька етапів і скорочену ДНФ зручно знаходити в наступній послідовності.

Провести в ДДНФ функції всі можливі операції склеювання конститuent одиниці (які складаються з  $n$  літер). У результаті цього утворяться добутки, що містять  $n - 1$  літеру. Підкреслимо, що склеюватися можуть тільки добутки з однаковим числом літер. Тому після цієї процедури провадиться операція поглинання, а потім виконуються всі можливі склеювання членів з  $n - 1$  літерою.

Після цього проводиться поглинання членів з  $n - 1$  літерою і знову виконується операція склеювання членів з числом літер, рівним  $n - 2$ , і т.д.

Приклад. Знайти скорочену диз'юнктивну нормальну форму логічної функції (таблиця 8.).

Представимо функцію в досконалій диз'юнктивній нормальній формі

$$f(a, b, c) = \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot \bar{b} \cdot c.$$

У правій частині отриманого виразу можна виконати тільки одне склеювання: другого члена з третім за змінною  $c$ . При цьому одержимо

$$f(a, b, c) = a \cdot \bar{b} \vee \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot \bar{b} \cdot c.$$

Добуток  $a \cdot \bar{b}$  поглинає члени  $a \cdot \bar{b} \cdot \bar{c}$  та  $a \cdot \bar{b} \cdot c$ :

$$f(a, b, c) = a \cdot \bar{b} \vee \bar{a} \cdot b \cdot c.$$

Цей вираз є скороченою диз'юнктивною нормальною формою заданої логічної функції, тому що подальше застосування склеювання і поглинання неможливе.

Тепер розглянемо процедуру мінімізації більш складної функції з використанням імплікантних матриць.

Припустимо, що потрібно знайти мінімальну ДНФ логічної функції, ДДНФ якої визначається виразом



$$f(a, b, c, d) = \bar{a} \cdot b \cdot c \cdot d \vee \bar{a} \cdot \bar{b} \cdot c \cdot d \vee a \cdot \bar{b} \cdot c \cdot d \vee a \cdot b \cdot \bar{c} \cdot d \vee a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} \vee \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}.$$

Побудуємо для цієї функції імплікантну матрицю, що представляє собою таблицю, у вертикальні і горизонтальні входи якої записуються всі конституенти одиниці, тобто всі мінтерми, і всі прості імпліканти заданої функції відповідно.

Для знаходження простих імплікант ми повинні провести спочатку процедуру неповного склеювання. Щоб швидше знаходити члени, що склеюються один з одним, нагадаємо, що склеюватися можуть тільки сусідні члени, тобто такі члени, у яких число змінних із запереченнями відрізняється на одиницю. Проведемо операції склеювання в наступному порядку:

виконаємо всі можливі склеювання 1-го члена з іншими;

виконаємо всі можливі склеювання 2-го члена з іншими, крім 1-го;

виконаємо всі можливі склеювання 3-го члена з іншими, крім 1-го і 2-го і т.д.

Запишемо результат:

$$1 - 2 = \bar{a} \cdot c \cdot d (\bar{b} \vee b) = \bar{a} \cdot c \cdot d \text{ (за } b);$$

$$2 - 3 = \bar{b} \cdot c \cdot d \text{ (за } a);$$

$$3 - 4 = a \cdot \bar{b} \cdot d \text{ (за } c);$$

$$4 - 5 = a \cdot \bar{b} \cdot \bar{c} \text{ (за } d);$$

$$5 - 6 = \bar{b} \cdot \bar{c} \cdot \bar{d} \text{ (за } a).$$

Одержали прості імпліканти. Після процедури неповного склеювання вираз прийме наступний вигляд:

$$f(a, b, c, d) = \bar{a} \cdot b \cdot c \cdot d \vee \bar{a} \cdot c \cdot d \vee \bar{a} \cdot \bar{b} \cdot c \cdot d \vee \bar{b} \cdot c \cdot d \vee \underline{a \cdot \bar{b} \cdot c \cdot d} \vee a \cdot \bar{b} \cdot d \vee \underline{a \cdot \bar{b} \cdot \bar{c} \cdot d} \vee a \cdot \bar{b} \cdot \bar{c} \vee \underline{a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}} \vee \bar{b} \cdot \bar{c} \cdot \bar{d} \vee \underline{a \cdot \bar{b} \cdot c \cdot d}.$$

Підкреслено ті члени, що поглинаються добутками, які утворилися після склеювання. Тепер проведемо операцію поглинання. Для кожної імпліканти знайдемо конституенти одиниці, тобто мінтерми, що нею поглинаються, тобто ті конституенти, власною частиною яких є дана імпліканта. Наприклад, імпліканта  $\bar{a} \cdot c \cdot d$  поглинає конституенти  $\bar{a} \cdot b \cdot c \cdot d$  та  $\bar{a} \cdot \bar{b} \cdot c \cdot d$ , імпліканта  $\bar{b} \cdot c \cdot d$  - конституенти  $\bar{a} \cdot \bar{b} \cdot c \cdot d$  та  $a \cdot \bar{b} \cdot c \cdot d$  і т.д.

$$\underline{\bar{a} \cdot b \cdot c \cdot d} \vee \bar{a} \cdot c \cdot d \vee \underline{\bar{a} \cdot \bar{b} \cdot c \cdot d} = \bar{a} \cdot c \cdot d (b \vee 1 \vee \bar{b}) = \bar{a} \cdot c \cdot d;$$

$$\underline{\bar{a} \cdot \bar{b} \cdot c \cdot d} \vee \bar{b} \cdot c \cdot d \vee \underline{a \cdot \bar{b} \cdot c \cdot d} = \bar{b} \cdot c \cdot d (\bar{a} \vee 1 \vee a) = \bar{b} \cdot c \cdot d;$$

$$\underline{a \cdot \bar{b} \cdot c \cdot d} \vee \underline{a \cdot \bar{b} \cdot d} \vee \underline{a \cdot \bar{b} \cdot \bar{c} \cdot d} = a \cdot \bar{b} \cdot d (c \vee 1 \vee \bar{c}) = a \cdot \bar{b} \cdot d;$$

$$\underline{a \cdot \bar{b} \cdot \bar{c} \cdot d} \vee \underline{a \cdot \bar{b} \cdot \bar{c}} \vee \underline{a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}} = a \cdot \bar{b} \cdot \bar{c} (d \vee 1 \vee \bar{d}) = a \cdot \bar{b} \cdot \bar{c};$$

$$\underline{a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}} \vee \underline{\bar{b} \cdot \bar{c} \cdot \bar{d}} \vee \underline{\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}} = \bar{b} \cdot \bar{c} \cdot \bar{d} (a \vee 1 \vee \bar{a}) = \bar{b} \cdot \bar{c} \cdot \bar{d}.$$

Тепер розглянемо принцип побудови і використання імплікантної матриці на прикладі матриці (таблиця 8.).

Клітки імплікантної матриці, утворені перетинанням рядків з імплікантами і колонок з поглиненими ними конститuentами, відзначимо хрестиками

**Таблиця 8.3. Імплікантна матриця**

№ п/п	Прості імпліканти	Конститuentи одиниці					
		$\bar{a} \cdot b \cdot c \cdot d$	$\bar{a} \cdot \bar{b} \cdot c \cdot d$	$a \cdot \bar{b} \cdot c \cdot d$	$a \cdot \bar{b} \cdot \bar{c} \cdot d$	$a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$	$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$
		1	2	3	4	5	6
1	$\bar{a} \cdot c \cdot d$	+	+				
2	$\bar{b} \cdot c \cdot d$		+	+			
3	$a \cdot \bar{b} \cdot d$			+	+		
4	$a \cdot \bar{b} \cdot \bar{c}$				+	+	
5	$\bar{b} \cdot \bar{c} \cdot \bar{d}$					+	+

Щоб одержати мінімальну ДНФ заданої функції, досить знайти мінімальне число імплікант, що спільно накривають хрестиками всі стовпчики імплікантної матриці.

З таблиці випливає, що в мінімальну форму обов'язково повинні ввійти імпліканти  $\bar{a} \cdot c \cdot d$  і  $\bar{b} \cdot \bar{c} \cdot \bar{d}$ , тому що тільки вони накривають хрестиками перший і шостий стовпчики таблиці.

Крім того, імпліканта  $\bar{a} \cdot c \cdot d$  накриває другий, а імпліканта  $\bar{b} \cdot \bar{c} \cdot \bar{d}$  - п'ятий стовпчики. Тому залишається накрити тільки третій і четвертий стовпчики таблиці. Для цього можна вибрати пари імплікант  $\bar{b} \cdot c \cdot d$  і  $a \cdot \bar{b} \cdot d$ ,  $a \cdot \bar{b} \cdot d$  і  $a \cdot \bar{b} \cdot \bar{c}$  або одну імпліканту  $a \cdot \bar{b} \cdot d$ . Якщо вибрати  $a \cdot \bar{b} \cdot d$ , члени  $\bar{b} \cdot c \cdot d$  і  $a \cdot \bar{b} \cdot \bar{c}$  виявляються зайвими, тому що імпліканта  $a \cdot \bar{b} \cdot d$  одна накриває третій і четвертий стовпчики таблиці. Таким чином, вибравши імпліканту  $a \cdot \bar{b} \cdot d$ , одержимо мінімальну ДНФ заданої функції:

$$f(a, b, c, d) = \bar{a} \cdot c \cdot d \vee a \cdot \bar{b} \cdot d \vee \bar{b} \cdot \bar{c} \cdot \bar{d}.$$

На підставі вищевикладеного сформулюємо алгоритм одержання мінімальних ДНФ логічної функції.

1. Логічну функцію представляють у зробленій ДНФ, застосовуючи або запис "за

одинацями" функції, якщо функція задана таблично, або застосовуючи операції розгортання, правила де Моргана й інші формул алгебри логіки, якщо функція задана в довільній аналітичній формі.

2. В отриманій досконалій ДНФ проводять всі операції неповного склеювання і поглинання. У результаті утворюється скорочена ДНФ заданої функції.

3. Знаходять мінімальні ДНФ з імплікантної матриці. Якщо кількість членів у скороченій ДНФ невелике, можна знайти форми глухого кута методом перебору членів і обрання серед них мінімальних.

Помітимо, що в ряді випадків мінімальна ДНФ збігається зі скороченою. Наприклад, скорочена ДНФ будь-якої логічної функції двох аргументів збігається з мінімальною, у чому неважко переконатися, провівши іспит членів скороченої ДНФ будь-якій ФАЛ2 (таблиця 2.).

Для того щоб знайти вираз заданої логічної функції, найбільш зручний для синтезу логічної схеми, слід, крім МДНФ функції, одержати також її мінімальну КНФ і вибрати з них ту, при технічній реалізації якої буде потрібна найменша кількість логічних елементів.

Один з алгоритмів одержання МКНФ функції аналогічний описаному вище для МДНФ. У цьому випадку також спочатку тим або іншим способом формується ДКНФ функції. Далі знаходиться скорочена КНФ, з якої визначаються КНФ глухого кута. Нарешті, за ними знаходиться МКНФ заданої функції. Якщо функція задана таблично, то ДКНФ формується "за нулями" функції. В усіх же подальших процедурах мінімізації враховується, що в цьому випадку використовуються макстерми, тобто елементарні суми, а не елементарні добутки. Імпліканта, у тому числі і проста, - також елементарна сума.

Операція неповного склеювання і поглинання для кон'юнктивної форми визначається відповідно наступними співвідношеннями:

$$(a \vee b)(a \vee \bar{b}) = a(a \vee b)(a \vee \bar{b}),$$

$$a(a \vee \bar{b}) = a,$$

а формули розгортання мають вигляд:

$$a = (a \vee b)(a \vee \bar{b}),$$

$$(a \vee b) = (a \vee b \vee c)(a \vee b \vee \bar{c}).$$

При іспиті членів скороченої КНФ виключають випробовуваний член і у вираз, що залишився, підставляють такі значення змінних, котрі звертають виключений член у нуль. Якщо при цьому вираз, що залишився, також буде дорівнює нулеві, то випробовуваний член є зайвим. У такий спосіб одержують КНФ глухого кута, з яких вибирають МКНФ. Якщо отримана скорочена КНФ містить велике число членів, то МКНФ одержують за допомогою імплікантних матриць за методикою, аналогічної використовуваний для одержання МДНФ.

Але можна використовувати й інший спосіб одержання МКНФ. Попередньо знаходиться МДНФ функції. Потім від отриманої МДНФ береться заперечення і після перетворення за правилами де Моргана одержують МКНФ заданої функції.

### 8.3 Мінімізація функцій методом Квайна-МакКласкі-Петрика

Метод складається з двох частин.

#### 8.3.1 Побудова простих імплікант (метод Квайна-МакКласкі).

Логічна функція  $g(a, b, \dots, z)$  називається імплікантою логічної функції  $f(a, b, \dots, z)$ , якщо

$$f(a, b, \dots, z) \& g(a, b, \dots, z) = g(a, b, \dots, z).$$

Імпліканта називається простою, якщо вона є кон'юнкцією змінних, і будь-яка кон'юнкція, отримана з неї шляхом викреслювання будь-яких змінних, не є імплікантою. Побудова простих імплікант ілюструється наступним прикладом.

Нехай функція  $f(a, b, c, d)$  задана таблично (таблиця 8.). Випишемо до графі I (таблиця 8.) усі набори, на яких функція  $f$  обертається в 1. Для виконання алгоритму їх зручно вписати розбитими на групи у відповідності з кількістю одиничних компонент у наборах (таблиця 8., колонка К у графі I). Оскільки мінімізуються (склеюються) лише набори, які відрізняються в одній компоненті, то для того, щоб провести всі склеювання за однією змінною, досить продивитися всі можливі пари наборів, які входять до двох сусідніх груп. Результати склеювання наборів із графі I розмістимо у графі II. Набори із графі I, які приймали участь у склеюваннях, позначимо знаком +. У графі II набори вже автоматично розбиваються на групи за кількістю одиниць (при склеюванні наборів графі I з груп із  $j-1$  одиницями і з  $j$  одиницями отримуються

набори графи II із  $j-1$  одиницями).

**Таблиця 8.4. ФАЛ4**

$abcd$	$f$
0000	1
0001	1
0010	0
0011	1
0100	1
0101	0
0110	1
0111	0
1000	1
1001	0
1010	1
1011	1
1100	1
1101	1
1110	0
1111	0

До створених наборів знову застосовуємо операцію склеювання (клеяться пари наборів, які мають риску на однакових місцях і відрізняються однією компонентою). При цьому треба знову переглянути всі пари наборів із сусідніх груп. Набори, до яких застосована операція, позначені знаком +. Результати склеювання заносимо до графи III таблиці. У графі III знову намагаємося виконати склеювання, але цього зробити не вдається. На цьому процедура завершується.

**Таблиця 8.5. Хід мінімізації**

I			II				III	
К	П	У	С	К	П	У	С	К
0000	$a_0$	+	$a_0b_0$	000-	$e_0$	+	$e_0f_3$	-00-
0001	$b_0$	+	$a_0b_1$	0-00	$e_1$	+	$e_1f_5$	--00
0100	$b_1$	+	$a_0b_2$	-000	$e_2$	+	$e_2f_4$	--00
1000	$b_2$	+	$b_0c_0$	00-1	$f_0$	+	$e_2f_2$	-00-
0011	$c_0$	+	$b_1c_1$	01-0	$f_1$		$f_2g_0$	-0-1
0110	$c_1$	+	$b_0c_2$	-001	$f_2$	+	$f_0g_1$	-0-1
1001	$c_2$	+	$b_2c_2$	100-	$f_3$	+	$f_3g_3$	1-0-
1100	$c_3$	+	$b_1c_3$	-100	$f_4$	+	$f_5g_2$	1-0-

1011	$d_0$	+	$b_2c_3$	1-00	$f_5$	+
1101	$d_1$	+	$c_0d_0$	-011	$g_0$	+
			$c_2d_0$	10-1	$g_1$	+
			$c_2d_1$	1-01	$g_2$	+
			$c_3d_1$	110-	$g_3$	+

Таблиця 8. має такі графи:

К - код набору;

С - склеюванням яких наборів цей код утворився;

П - умовне позначення набору;

У - позначка про участь набору в склеюванні.

В отриманій таблиці знаходяться всі імпліканти функції  $f$ , які мають вигляд кон'юнкцій. Простими будуть лише ті з них, які не мають позначки + (із яких не можна викреслити жодної змінної, інакше може бути застосована операція склеювання). В розглянутому прикладі простими імплікантами є кон'юнкції

$$\bar{a} \cdot \bar{b} \cdot \bar{d}, \bar{b} \cdot \bar{c}, \bar{c} \cdot \bar{d}, \bar{b} \cdot d, a \cdot \bar{c},$$

які відповідають наборам

01-0, -00-, --00, -0-1, 1-0-.

### 8.3.2 Формування мінімальної диз'юнктивної нормальної форми (метод Петрика).

Позначимо через  $I_1, I_2, \dots, I_s$  всі прості імпліканти функції  $f$ . Будемо говорити, що кон'юнкція  $K$  покриває набір  $n$ , якщо на наборі  $n$  вона дорівнює 1. Побудуємо імплікантну таблицю (таблицю покриття) для функції  $f$ . Її рядки відповідають одиничним наборам функції  $f$ , а графи - простим імплікантам. На схрещенні рядка  $n$  і графа  $I_j$  проставляється \*, якщо імпліканта  $I_j$  покриває набір  $n$  (у протилежному випадку не ставиться нічого).

Імплікантна таблиця 8. відповідає функції (таблиця 8.), і імплікантам, знайденим у процесі мінімізації (таблиця 8.).

З імплікантою  $I_j$  будемо пов'язувати логічну змінну  $i_j$ . Кожній множині імплікант приписується набір значень змінних  $i_j$ : якщо  $I_j$  входить у множину, то  $i_j = 1$ , якщо ні, то  $i_j = 0$ . Розглянемо рядок таблиці покриття, який відповідає якомусь набору  $n$ . Нехай у цьому рядку символи \* знаходяться у графах  $I_1, I_2, \dots, I_w$ . Рядок  $n$  буде

покритий тоді і лише тоді, коли до множини буде введена хоча б одна з імплікант  $I_1, I_2, \dots, I_w$ , тобто коли диз'юнкція  $i_1 \vee i_2 \vee \dots \vee i_w = 1$ . Складемо таку диз'юнкцію для кожного рядка імплікантної таблиці (таблиця 8.) і візьмемо їхній добуток за всіма рядками. Отримаємо функцію  $F$ :

$$F = (i_2 \vee i_3)(i_2 \vee i_4)i_4(i_1 \vee i_3)i_1(i_2 \vee i_3 \vee i_5)(i_2 \vee i_4 \vee i_5)i_4(i_3 \vee i_5)i_5.$$

**Таблиця 8.6. Імплікантна таблиця**

Одиничні набори	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$
	$\bar{a} \cdot b \cdot \bar{d}$	$\bar{b} \cdot \bar{c}$	$\bar{c} \cdot \bar{d}$	$\bar{b} \cdot d$	$a \cdot \bar{c}$
$abcd$	01-0	-00-	--00	-0-1	1-0-
0000		*	*		
0001		*		*	
0011				*	
0100	*		*		
0110	*				
1000		*	*		*
1001		*		*	*
1011				*	
1100			*		*
1101					*
Літер в імплікації	3	2	2	2	2

Після спрощення  $F = i_1 \cdot i_2 \cdot i_4 \cdot i_5 \vee i_1 \cdot i_3 \cdot i_4 \cdot i_5$ . Тепер функція  $F$  вказує на 2 ненадлишкових покриття:  $\{I_1, I_2, I_4, I_5\}$  та  $\{I_1, I_3, I_4, I_5\}$ . Їм відповідають дві ДНФ, кожна з яких складаються з 9 літер, тому кожна з них є мінімальною. Таким чином, дана функція  $f$  має дві мінімальні ДНФ:

$$f = I_1 \vee I_2 \vee I_4 \vee I_5 = \bar{a} \cdot b \cdot \bar{d} \vee \bar{b} \cdot \bar{c} \vee \bar{b} \cdot d \vee a \cdot \bar{c},$$

$$f = I_1 \vee I_3 \vee I_4 \vee I_5 = \bar{a} \cdot b \cdot \bar{d} \vee \bar{c} \cdot \bar{d} \vee \bar{b} \cdot d \vee a \cdot \bar{c}.$$

При спрощенні слід брати до уваги:

в першу чергу, вирази вигляду  $\dots(i_1 \vee i_3)i_1\dots$  спрощуються до  $\dots i_1 \dots$ , за правилом,  $(a \vee b)a = a$ ;

по-друге, якщо у дужках є декілька імплікацій з різною кількістю літер, то в результаті спрощення у дужці залишаються лише імпліканти з мінімальною кількістю літер. Наприклад, у наведеному вище прикладі дужка  $(i_1 \vee i_3)$ , до якої входять

імпліканти з кількістю літер відповідно 3 і 2, спрощується до виразу  $i_3$  (кількість літер - 2).

#### 8.4 Мінімізація не повністю визначених функцій методом Квайна-МакКласкі-Петрика.

Кожний рядок таблиці істинності функції алгебри логіки відповідає певному значенню логічної функції для відповідної комбінації значень аргументів, тобто для відповідного набору. У деяких випадках відомо, що якийсь набір з'явитися не може або ж, якщо він з'явиться, то значення функції на ньому, з тих або інших причин, несуттєве. Для таких випадків немає необхідності визначати таке значення функції за таблицею істинності. У таких випадках говорять про невизначені умови.

У таблиці істинності невизначена умова позначається спеціальним символом (X) у відповідній комірці графі значень функції. Такі комірки можуть довільним чином включатися в групи при побудові мінімальних макстермів або мінтермів. Будь-яку з них можна включити як у групу одиничних, так і в групу нульових комірок, або ж узагалі нікуди не включати.

Під час мінімізації логічних функцій з невизначеними значення всі дії відбуваються, як описано вище. Лише на першому етапі до табл. 8.5 як початкові набори вводяться додатково і всі набори, на яких функція невизначена, а на другому – до табл. 8.6, як і раніше, тільки набори, на яких функція приймає значення 1.

Нариклад, функція задана таблично (таблиця 8.), X – невизначене значення.

**Таблиця 8.7. ФАЛ4 з X**

<i>abcd</i>	<i>f</i>
0000	0
0001	0
0010	0
0011	0
0100	0
0101	1
0110	0
0111	X
1000	0
1001	0



1010	X
1011	X
1100	0
1101	X
1110	X
1111	1

Знаходження простих імплікант показує таблиця 8..

Знаходження мінімального покриття показує таблиця 8..

**Таблиця 8.8. Хід мінімізації ФАЛ з X**

I			II			III		
К	П	У	С	К	П	У	С	К
0101	$a_0$	+	$a_0b_0$	01- 1	$d_0$	+	$d_0e_2$	-1-1
1010	$a_1$	+	$a_0b_2$	- 101	$d_1$	+	$d_1e_0$	-1-1
0111	$b_0$	+	$a_1b_1$	101 -	$d_2$	+	$d_2e_3$	1-1-
1011	$b_1$	+	$a_1b_3$	1- 10	$d_3$	+	$d_3e_1$	1-1-
1101	$b_2$	+	$b_0c_0$	- 111	$e_0$	+		
1110	$b_3$	+	$b_1c_0$	1- 11	$e_1$	+		
1111	$c_0$	+	$b_2c_0$	11- 1	$e_2$	+		
			$b_3c_0$	111 -	$e_3$	+		

**Таблиця 8.9. Результат**

Одиничні набори	$I_1$	$I_2$
	$bd$	$ac$
$abcd$	-1-1	1-1-
0101	*	
1111	*	*
Літер в імплікації	3	2

## 8.5 Метод карт Карно (діаграм Вейча)

Процес спрощення булевих, тобто логічних, виразів, як уже відзначалося, не є алгоритмічним. Далеко не очевидно, яку з тотожностей варто застосувати на тому або іншому кроці. Мистецтво приходить тільки з досвідом. Тому для спрощення булевих виразів були розроблені алгоритмічні методи. Ще одним з них є метод карт Карно.

Кarti Карно (їхнім різновидом є карти Вейча, що будуються як розгорнення куба на площині), є графічним представленням таблиць істинності. Тому вони будуються або за таблицею істинності аналізованої функції, або ж за її ДДНФ.

Нагадаємо, що кожен рядок таблиці істинності, для якого функція дорівнює одиниці, відповідає конкретному мінтермові функції, представленої в ДДНФ. Рядок, для якого функція дорівнює нулеві є визначеним макстермом функції, представленої в ДКНФ.

Карта Карно являє собою прямокутник, розбитий на клітинки, число яких дорівнює загальному числу наборів для даної функції  $n$  змінних, тобто воно дорівнює  $2^n$ . Так, для функції чотирьох змінних клітинок буде 16, для п'яти змінних - 32 і т.д. Кожна клітинка відповідає певному наборові або термові, причому набори розташовуються так, щоб сусідні набори або терми, як по горизонталі, так і по вертикалі, відрізнялися б тільки значенням однієї змінної: в одній клітинці вона з інверсією, а в іншій, сусідній - без. Функцію в ДДНФ наносять на карту, відзначаючи, наприклад, знаком 1 клітинки, що відповідають тим наборам, на яких функція дорівнює одиниці, тобто в ДДНФ функції є відповідний мінтерм. Інші клітинки відзначаються знаком 0. Іноді в куті клітинки ставлять номер набору. Він використовується, якщо функція задана числовим способом, але це незручнє для процедури мінімізації.

Розглянемо приклади побудови карт Карно за таблицями істинності для 2-х, 3-х і 4-х змінних.

1) Логічна функція двох змінних (рис. 8.1), де а - таблиця істинності; б - карта Карно.

a	b	f(a,b)
0	0	f(0,0)

0	1	f(0,1)
1	0	f(1,0)
1	1	f(1,1)

а

	b = 0	b = 1
a = 0	f(0,0)	f(0,1)
a = 1	f(1,0)	f(1,1)

б

Рис. 8.1. Таблиця істинності (а) і карта Карно (б) для ФАЛ двох змінних

2) Логічна функція трьох змінних (рис. 8.2).

a	b	c	f(a,b,c)
0	0	0	f(0,0,0)
0	0	1	f(0,0,1)
0	1	0	f(0,1,0)
0	1	1	f(0,1,1)
1	0	0	f(1,0,0)
1	0	1	f(1,0,1)
1	1	0	f(1,1,0)
1	1	1	f(1,1,1)

а

	bc = 00	bc = 01	bc = 11	bc = 10
a = 0	f(0,0,0)	f(0,0,1)	f(0,1,1)	f(0,1,0)
a = 1	f(1,0,0)	f(1,0,1)	f(1,1,1)	f(1,1,0)

б

Рис. 8.2. Таблиця істинності (а) і карта Карно (б) для ФАЛ трьох змінних

3) Логічна функція чотирьох змінних (рис. 8.3).

a	b	c	d	f(a,b,c,d)
0	0	0	0	f(0,0,0,0)
0	0	0	1	f(0,0,0,1)
0	0	1	0	f(0,0,1,0)
0	0	1	1	f(0,0,1,1)
0	1	0	0	f(0,1,0,0)
0	1	0	1	f(0,1,0,1)
0	1	1	0	f(0,1,1,0)
0	1	1	1	f(0,1,1,1)
1	0	0	0	f(1,0,0,0)
1	0	0	1	f(1,0,0,1)
1	0	1	0	f(1,0,1,0)
1	0	1	1	f(1,0,1,1)
1	1	0	0	f(1,1,0,0)
1	1	0	1	f(1,1,0,1)
1	1	1	0	f(1,1,1,0)
1	1	1	1	f(1,1,1,1)

а

	cd = 00	cd = 01	cd = 11	cd = 10
ab = 00	f(0,0,0,0)	f(0,0,0,1)	f(0,0,1,1)	f(0,0,1,0)
ab = 01	f(0,1,0,0)	f(0,1,0,1)	f(0,1,1,1)	f(0,1,1,0)
ab = 11	f(1,1,0,0)	f(1,1,0,1)	f(1,1,1,1)	f(1,1,1,0)
ab = 10	f(1,0,0,0)	f(1,0,0,1)	f(1,0,1,1)	f(1,0,1,0)

б

Рис. 8.3. Таблиця істинності (а) і карта Карно (б) для ФАЛ чотирьох змінних

Карту Карно для функції, наприклад, чотирьох змінних  $a, b, c, d$  можна

представити й у такий спосіб (рис. 8.4).

		$cd$			
		00	01	11	10
$ab$	00				
	01				
	11		*		
	10				

Рис. 8.4. Карта Карно для ФАЛ чотирьох змінних

Наприклад, у клітці із зірочкою записується значення ФАЛ на наборі 1101, тобто на наборі  $a \cdot b \cdot \bar{c} \cdot d$ .

Аналогічно карту Карно для п'яти аргументів  $a, b, c, d, e$  можна представити в наступному виді (рис. 8.5).

		$a=1$				$a=0$			
		$de$				$de$			
		00	01	11	10	00	01	10	11
$bc$	00								
	01								
	11						*		
	10								

Рис. 8.5. Карта Карно для ФАЛ п'яти змінних

Наприклад, у клітці із зірочкою записується значення ФАЛ на наборі 01101, тобто на наборі  $\bar{a} \cdot b \cdot c \cdot \bar{d} \cdot e$ .

Нарешті, карта Карно для шести аргументів  $a, b, c, d, e, g$  може виглядати, наприклад, у такий спосіб (рис. 8.6).

		$a=0$				$a=1$			
		$eg$				$eg$			
		00	01	11	10	00	01	10	11
$b=0$	$cd$	00							
		01							
		11							
		10							
$b=1$	$cd$	00							
		01							
		11		*					
		10							

Рис. 8.6. Карта Карно для ФАЛ шести змінних

Наприклад, у клітці із зірочкою записується значення ФАЛ на наборі 011101, тобто на наборі  $\bar{a}bcd \bar{e} g \bar{a} \cdot b \cdot c \cdot d \cdot \bar{e} \cdot g$ .

Карту Карно для ФАЛ зокрема п'яти аргументів  $abcde$ , представляють разом з номерами наборів змінних (у кутках клітинок) і в такий спосіб (рис. 8.7). Номери клітинок розташовані у кутах клітинок, а змінні позначені дугами на краях карти. При цьому дуга охоплює ті клітинки карти, у яких відповідна змінна має значення 1. У всіх інших клітинках карти Карно ця змінна буде мати значення 0. У такий спосіб кожна змінна розбиває карту Карно навпіл, але кожна по-своєму.

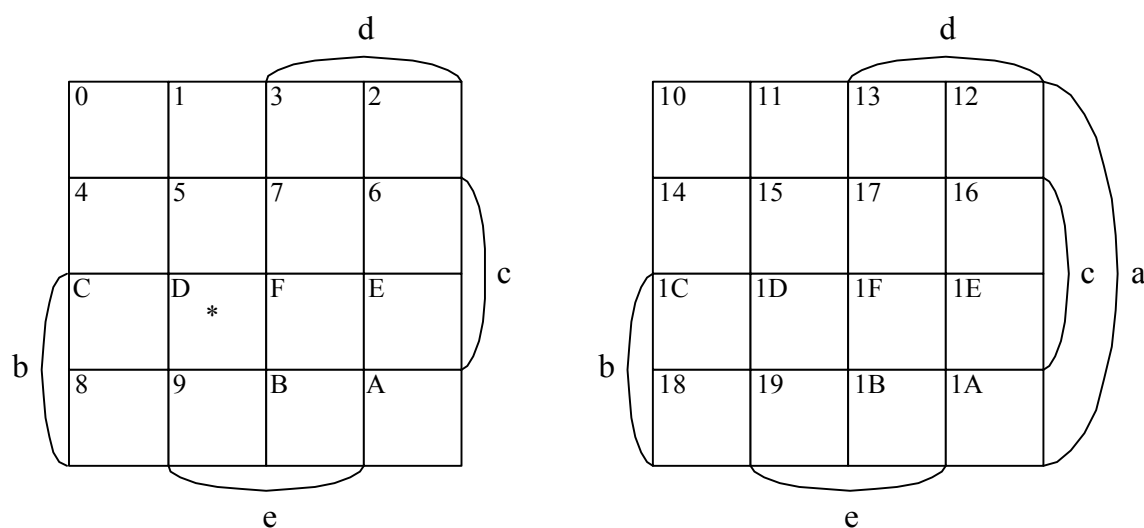


Рис. 8.7. Карта Карно для ФАЛ п'яти змінних

Наприклад, в клітці із зірочкою записується значення ФАЛ на наборі 011101, тобто на наборі  $\bar{a} \cdot b \cdot c \cdot \bar{d} \cdot e$ .

У підсумку карту Карно можна оформляти будь-яким способом, але тільки так, щоб кожна клітинка відповідала певному набору або терму і щоб вони, як уже відзначалося, розташовувалися б таким чином, щоб сусідні набори або терми, як по горизонталі, так і по вертикалі, відрізнялися б тільки значенням одні змінної: в одній клітинці вона з інверсією, а в іншій, сусідній - без. Причому, треба врахувати, що клітинки, розташовані на протилежних кінцях кожного рядка або стовпця, також є сусідніми.

Перерахуємо послідовність дій, що виконуються для мінімізації функції методом карт Карно.

1) Вихідна функція, що підлягає мінімізації, повинна бути представлена в ДНФ.

Потім її треба представити в ДДНФ. Або ж складається таблиця істинності функції, що підлягає мінімізації.

Як уже відзначалося, між рядками таблиці істинності і клітками на карті Карно існує взаємно однозначна відповідність. Коли карта Карно складається за ДДНФ функції, що підлягає мінімізації, то кожна змінна без заперечення замінюється значенням 1, а із запереченням - 0.

2) Потім будується карта Карно за принципом, описаним раніше. Представимо систему координат, у якій, наприклад, для функції двох аргументів по горизонтальній осі відкладаються значення одного аргументу, а по вертикалі - іншого. На перетинанні відповідних координат одержуємо комірку, куди записується значення функції (0 або 1), що відповідає даному наборові. Якщо функція представлена в ДДНФ, то в комірці відповідному існуючому мінтермові записується 1, а в комірці неіснуючого мінтерму - 0.

3) Після цього поєднуються в групи суміжні комірки, у яких записані одиниці, у такий спосіб: поєднується обов'язково  $2^n$  сусідніх комірок з одиницями як по вертикалі, так і по горизонталі. Причому, кожна комірка з 1 може потрапити одночасно в декілька груп, отриманих в результаті об'єднання одиниць як по вертикалі, так і по горизонталі.

4) Кожній такій групі ставиться у відповідність новий мінтерм для зображення вихідної функції у формі мінімальної ДНФ.

5) Зображення кожного нового мінтерму формується за наступним алгоритмом:

а) змінна, котра в кожній комірці утвореної групи має значення тільки 0, зображується її інверсією;

б) змінна, котра в кожній комірці групи має значення тільки 1, зображується без інверсії;

в) змінна, котра в межах утвореної групи змінює своє значення, не зображується, тобто скорочується, відкидається.

Таким чином, карту Карно можна розглядати як графічне представлення сукупності усіх (існуючих і не існуючих) мінтермів функції в ДДНФ даного числа логічних перемінних.

На підставі закону дистрибутивності і теорем  $a \vee \bar{a} = 1$  і  $a \cdot \bar{a} = 0$ , а також  $a \vee 0 = a$  і  $a \cdot 0 = 0$ , два мінтерми, що знаходяться в сусідніх клітках, можуть бути замінені одним новим мінтермом, що містить на одну перемінну менше. Якщо сусідніми є дві пари мінтермів, то така група з 4 мінтермів може бути замінена новим мінтермом, що містить на дві змінні менше і т.д. У загальному випадку наявність одиниць у  $2^n$  сусідніх клітках дозволяє виключити  $n$  змінних.

При формуванні на базі карти Карно нових мінтермів для представлення функції в мінімальної ДНФ, або ж у мінімальної КНФ, значення відповідних змінних, рівних 1, замінюються зображенням змінних без заперечення, а при значеннях рівних 0 - із запереченням.

Розглянемо процес мінімізації на прикладі функції, заданої наступним логічним рівнянням:

$$f(a, b, c, d) = b \cdot c \cdot d \vee \bar{a} \cdot b \cdot d \vee \bar{b} \cdot c \cdot d \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot \bar{c} \cdot d \vee \bar{b} \cdot \bar{c} \cdot d \vee \bar{a} \cdot b \cdot c \vee \bar{a} \cdot \bar{b} \cdot \bar{c}.$$

Представимо цю функцію в ДДНФ:

$$\begin{aligned} f(a, b, c, d) &= (a \vee \bar{a}) \cdot b \cdot c \cdot d \vee \bar{a} \cdot b \cdot (c \vee \bar{c}) \cdot d \vee (a \vee \bar{a}) \cdot \bar{b} \cdot c \cdot d \vee a \cdot \bar{b} \cdot \bar{c} \cdot (d \vee \bar{d}) \vee \\ &\vee a \cdot (b \vee \bar{b}) \cdot \bar{c} \cdot d \vee (a \vee \bar{a}) \cdot \bar{b} \cdot \bar{c} \cdot d \vee \bar{a} \cdot b \cdot c \cdot (d \vee \bar{d}) \vee \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot (d \vee \bar{d}) = \\ &= abcd \vee \bar{a}bcd \vee \bar{a}\bar{b}cd \vee \bar{a}b\bar{c}d \vee \bar{a}\bar{b}\bar{c}d \vee \bar{a}bcd \vee \bar{a}\bar{b}cd \vee \bar{a}bcd \vee \bar{a}\bar{b}cd \vee \bar{a}\bar{b}\bar{c}d. \end{aligned}$$

Нижче зображена карта Карно, що відповідає розглянутій функції (рис. 8.8).

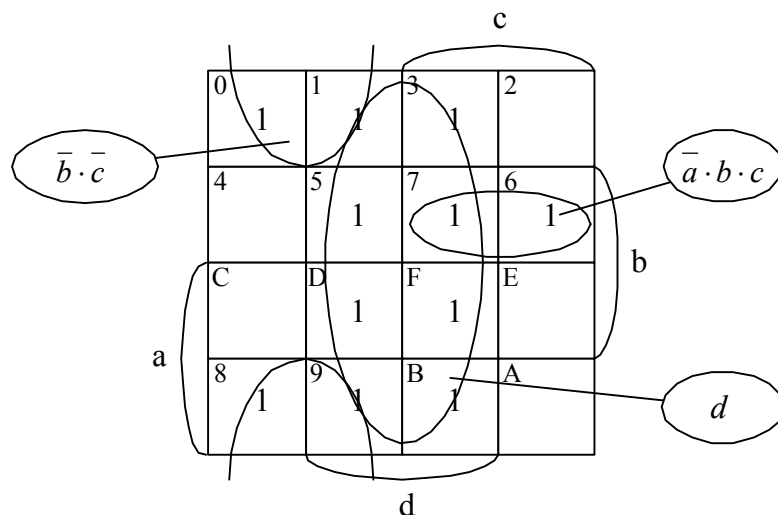


Рис. 8.8. Мінімізація ФАЛІ за допомогою Карти Карно

Мінтерми функції утворюють у карті три групи:

- 1)  $\bar{b} \cdot \bar{c}$ ; 2)  $d$ ; 3)  $\bar{a} \cdot b \cdot c$ .

Отже,  $f = \bar{b} \cdot \bar{c} \vee d \vee \bar{a} \cdot b \cdot c$ .

Але треба мати на увазі, що в загальному випадку функція може мати кілька мінімальних форм.

Хоча звичайно карти Карно для функцій 3-х і 4-х перемінних зображуються на площині, але як відзначалося вище, з погляду формування прямокутних груп карту потрібно вважати багатовимірної.

Для карт із трьома змінними карту варто розглядати як циліндр зі склеєними або по вертикалі, або по горизонталі, або одночасно і по вертикалі, і по горизонталі. Оскільки прямокутні групи формуються на такому циліндрі, на плоскому малюнку та або інша група може виявитися розірваною.

Розірвані прямокутні групи можуть з'являтися і на картах з чотирма змінними. На таких картах потрібно вважати склеєними не тільки краї краї, але, при необхідності, також верхню та нижню площини (рис. 8.8).

При одержанні мінімальної форми для ДКНФ функція задається термами, що приймають нульове значення на відповідних наборах, тобто макстермами. Тому в клітках карт Карно пишуть нулі, що групуються за описаною раніше методикою. При цьому одержують вирази для інверсного значення шуканої функції. Для одержання прямого значення функції необхідно потім застосувати правила де Моргана.

Як уже відзначалося, для того щоб знайти вираз заданої логічної функції, найбільш зручний для синтезу логічної схеми, необхідно, крім мінімальних диз'юнктивних нормальних форм, одержати також мінімальні кон'юнктивні нормальні форми і вибрати з них ту, при технічній реалізації якої буде потрібна найменша кількість елементів. Існує кілька методів одержання мінімальних кон'юнктивних нормальних форм.

На закінчення огляду методів мінімізації логічних функцій потрібно відзначити, що в даний час розроблено багато програм, що дозволяють робити процедуру мінімізації на комп'ютерах різними методами.



## 9 АНАЛІЗ ТА СИНТЕЗ ЛОГІЧНИХ ЕЛЕКТРОННИХ СХЕМ

### 9.1 Логічні оператори електронних схем

Логічною схемою називається сукупність логічних електронних елементів, з'єднаних між собою таким чином, щоб виконувався заданий закон функціонування схеми, інакше кажучи, виконувалася задана логічна функція.

У залежності вихідного сигналу від вхідного всі електронні логічні схеми можна умовно розбити на:

схеми першого роду, тобто комбінаційні схеми, схеми без зворотніх зв'язків, вихідний сигнал яких у момент часу  $t$  залежить тільки від стану вхідних сигналів у цей же момент часу  $t$ ;

схеми другого роду або схеми із зворотніми зв'язками (схеми послідовнісні), що елементи пам'яті, вихідний сигнал яких залежить як від значення вхідних сигналів у момент часу  $t$ , так і від їхніх значень у попередні моменти часу.

За кількістю входів і виходів схеми бувають:

- з одним входом і одним виходом,
- з декількома входами й одним виходом,
- з одним входом і декількома виходами,
- з декількома входами і виходами.

За способом здійснення синхронізації схеми бувають:

- із зовнішньою синхронізацією (синхронні автомати),
- з внутрішньою синхронізацією (асинхронні автомати є їхнім окремим випадком).

Практично будь-який цифровий автомат (і комп'ютер в тому числі) складається з комбінації схем першого і другого родів різної складності. Таким чином, основою будь-якого цифрового автомата, що обробляє цифрову інформацію, є електронні елементи двох типів:

- логічні або комбінаційні,
- запам'ятовуючі або послідовнісні.

Логічні елементи виконують найпростіші логічні операції над цифровою інформацією, а запам'ятовуючі служать для її збереження. Як ми вже знаємо, логічна операція це перетворення за визначеними правилами вхідної цифрової інформації у

вихідну.

Як уже відзначалися функції алгебри логіки апаратно реалізуються за допомогою відповідних комбінаційних електронних логічних елементів - вентилів. Можна вважати, що ці елементарні логічні функції є логічними операторами згаданих електронних елементів, тобто схем. Кожна така схема позначається певним графічним символом, що може бути орієнтований або ж неорієнтований.

Приклади графічного позначення цих схем було наведено вище у Розділі 7.

При аналізі або синтезі логічних ланцюгів треба враховувати наступну обставину.

Здійснити логічні функції на практиці дозволяють різні, так називані логічні (цифрові) напівпровідникові схеми - вентиля, вихідні сигнали яких однозначно визначаються комбінаціями рівнів сигналів на входах цих схем. Причому, як вхідні, так і вихідні сигнали цих вентилів можуть бути імпульсними або потенційними і мають два фіксованих значення: логічний „0” і логічна „1”.

У потенційній системі, якщо логічний "1" відповідає один рівень, тоді логічному "0" – інший. Якщо логічний "1" відповідає наявність сигналу високого рівня, то в такому випадку говорять, що логічні схеми працюють у додатній логіці. Якщо ж логічний "1" відповідає наявність сигналу низького рівня, то вважається, що схеми працюють у від'ємній логіці. Існує також і змішана логіка, тобто коли в розглянутому електронному вузлі одні вентиля працюють у додатній логіці, а інші - у від'ємній.

Звідси випливає, що, згідно уже відомому нам властивості двоїстості логічних функцій (правила де Моргана), той самий вентиль, що у додатній логіці реалізує якусь функцію, у від'ємній логіці буде виконувати іншу логічну функцію (таблиця 9.).

**Таблиця 9.1. Деякі двоїсті функції алгебри логіки**

ФАЛ у додатній логіці	ФАЛ у від'ємній логіці	Правило де Моргана	Ілюстрація
I-НЕ	НЕ-АБО	$\overline{ab} = \overline{a} \vee \overline{b}$	Рис. 9.1
АБО-НЕ	НЕ-I	$\overline{a \vee b} = \overline{a} \cdot \overline{b}$	Рис. 9.2
I	НЕ-АБО-НЕ	$ab = \overline{\overline{a \vee b}}$	Рис. 9.3
АБО	НЕ-I-НЕ	$a \vee b = \overline{\overline{a \cdot b}}$	Рис. 9.4

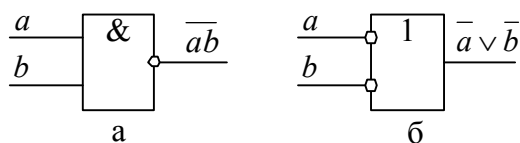


Рис. 9.1. Символи двоїстих функції І-НЕ та НЕ-АБО

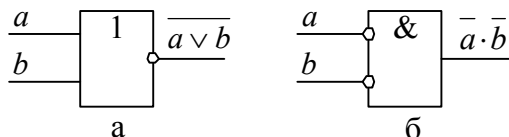


Рис. 9.2. Символи двоїстих функції АБО-НЕ та НЕ-І

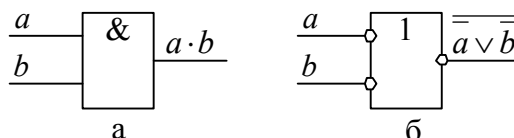


Рис. 9.3. Символи двоїстих функції І та НЕ-АБО-НЕ

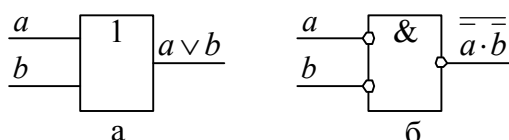


Рис. 9.4. Символи двоїстих функції АБО та НЕ- І-НЕ

З цих креслень видно, що там, де логічної "1" відповідає відсутність сигналу, а логічному "0" - наявність сигналу, на схемі прийнято креслити коло. Інакше кажучи, можна вважати, що коло означає, що на відповідний вхід логічного елемента подається інверсія логічної змінної, а з виходу - виводиться інверсне значення змінної або логічної функції.

Усе це дозволяє абстрагуватися від фізичних характеристик сигналів і вважати, що на входах і виходах вентилів та й інших компонентів логічної схеми з'являються безпосередньо логічні значення. Тому можна прийти до висновку, що різні вентиля або їхні комбінації на логічному рівні можуть бути описані за допомогою логічних операторів. Такий операторний опис вентилів дозволяє відриватися (абстрагуватися) від фізичної природи конкретних вентилів і інших електронних елементів і здійснювати аналіз відповідних електронних схем. При цьому виявляється, що для аналізу зовсім не обов'язково мати саму схему. Для того, щоб одержати значення функції на виході якої-небудь схеми, досить записати цю залежність у вигляді

логічних операторів, зв'язаних між собою відповідно до виконуваної функції.

## 9.2 Задачі аналізу і синтезу електронних схем

Для аналізу електронних схем за допомогою апарата алгебри логіки потрібно знайти логічну функцію, що описує роботу заданої схеми. При цьому виходять з того, що кожному функціональному елементові електронної схеми можна поставити у відповідність логічний оператор. Цим самим встановлюється однозначна відповідність між елементами схеми і її математичним описом.

Аналіз електронної схеми проводиться в три етапи:

- 1) із принципової схеми видаляються всі несуттєві допоміжні елементи, що не впливають на логіку роботи схеми;
- 2) через логічні оператори виражають всі електронні елементи, дістаючи логічне рівняння, що є моделлю функції, виконуваною заданою схемою;
- 3) будь-яким способом одержують МКНФ і МДНФ цього рівняння і тим самим виявляють зайві частини аналізованої схеми, якщо, звичайно, вони в ній є.

Вилучення зайвих частин спрощує схему і тим самим поліпшує її технічні характеристики.

Задачу синтезу електронних схем можна сформулювати в такий спосіб:

при заданих вхідних змінних і відомій вихідній функції спроектувати логічний пристрій, що реалізує цю функцію. Отже, у результаті рішення задачі синтезу виникає логічна схема, що відтворює задану функцію.

Звичайно, вирішуючи задачі аналізу і синтезу, використовують повні базиси логічних функцій. При цьому кожен логічний функцію, що входить у базис, зіставляють з деяким фізичним електронним елементом, що дозволяє логічну схему замінити структурною схемою, яка складається з електронних елементів.

Таким чином, удається з'єднати математичну задачу синтезу логічної схеми з інженерною задачею проектування електронної схеми. При розробці електронної схеми за основні критерії приймають:

- мінімум апаратури,
- мінімум типів застосовуваних елементів,
- максимум швидкодії,

максимум надійності.

З погляду математичної логіки задача синтезу вирішується при забезпеченні: мінімального числа логічних операторів, мінімальної кількості типів логічних операторів.

Можна сформулювати послідовно розв'язувані задачі при синтезі електронної схеми:

складання математичного опису (системи логічних рівнянь), що адекватно відображає процеси, що відбуваються в схемі;

аналіз логічних рівнянь і одержання мінімальної форми для кожного з них у заданому базисі;

перехід від логічних рівнянь до логічної (структурної) схеми за допомогою застосування логічних операторів.

Таким чином, синтез електронних схем починається з завдання вхідних змінних і вихідної функції або вихідних функцій. Очевидно, що все це може бути задане або відповідною таблицею істинності, або ж аналітично. У будь-якому випадку наступним етапом синтезу є процедура одержання будь-яким методом МДНФ і МКНФ відповідного логічного рівняння або рівнянь

Далі на основі отриманих МДНФ і МКНФ синтезуються структурні або функціональні схеми кількох варіантів електронної схеми з використанням різних базисів. Після цього вибирається як остаточний варіант та структурна схема, що має мінімальне число логічних електронних елементів, мінімальна кількість типів цих елементів і мінімальну кількість каскадів синтезованої схеми.

Під кількістю каскадів у даному випадку розуміється число логічних елементів на шляху сигналу від входу в схему до виходу. Мінімальна глибина забезпечує мінімальну затримку реакції схеми на зміну сигналів на її входах при обраному типі (серії) електронних елементів синтезованої схеми.

Задача синтезу, як правило, має різні рішення в залежності від обраної системи логічних елементів. Однак для будь-якої заданої функції алгебри логіки майже завжди можна синтезувати схему, що відповідає цієї функції. Одержання схеми з мінімальною кількістю логічних зв'язувань вимагає знаходження мінімальної форми для логічної

функції.

### 9.3 Синтез логічних схем з одним виходом у базисі Буля на елементах з довільною кількістю входів

Базис Буля (базис І, АБО, НЕ) складається з трьох функцій алгебри логіки (ФАЛ):

функція І (кон'юнкція, логічне множення, AND, в аналітичному запису -  $\&$ ,  $\cdot$ ), кількість входів – більше 1;

функція АБО (диз'юнкція, логічне додавання, OR, в аналітичному запису – “ $\vee$ ”, «+»), кількість входів – більше 1;

функція НЕ (інверсія, в аналітичному запису – риска над символом, або “/” перед символом, або “-“ перед символом) , кількість входів – 1.

Рис. 9.5 містить умовні графічні позначення елементів І, АБО, НЕ.

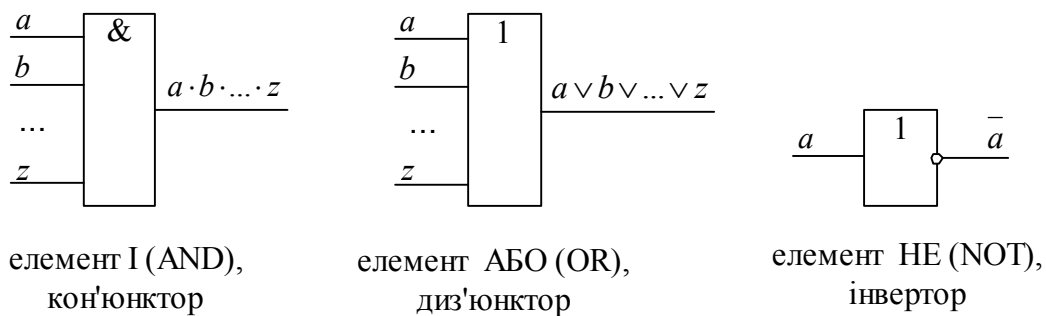


Рис. 9.5. Символи елементів базису Буля

Таблиця 9.2. Кон'юнкція та диз'юнкція багатьох змінних

Номер набору	Аргументи				Функція	
	$a$	$b$	$\dots$	$z$	І	АБО
0	0	0	$\dots$	0	0	0
1	0	0	$\dots$	1	0	1
	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$2^n - 2$	1	1	$\dots$	0	0	1
$2^n - 1$	1	1	$\dots$	1	1	1

На виході елемента І буде одиниця тільки тоді, коли на всіх його входах  $a, b, c, \dots, z$  є одиниця (таблиця 9.).

На виході елемента АБО буде одиниця тоді, коли хоча б на одному з його входів  $a, b, c, \dots, z$  є одиниця (таблиця 9.).

На виході елемента НЕ буде одиниця тоді, коли на його вході  $a$  є нуль

(таблиця 9.).

**Таблиця 9.3. Функція НЕ**

Номер набору	Аргумент	Функція
	$a$	НЕ
0	0	1
1	1	0

Таблиця істиності задає значення ФАЛ при всіх можливих станах аргументів (змінних) цієї функції.

Базис Буля (тобто, функціонально повна система функцій алгебри логіки, яка складається з трьох функцій – І, АБО, НЕ). За допомогою функції базису Буля можна реалізувати будь-яку іншу функцію алгебри логіки.

Досконалі диз'юнктивні нормальні форми (ДДНФ) це аналітичний вираз ФАЛ у вигляді диз'юнкції кон'юнкцій. При цьому кожна кон'юнкція містить усі змінні (аргументи) ФАЛ, а кількість термів дорівнює кількості одиничних значень функції у її таблиці істиності. ДДНФ визначає, коли дана ФАЛ набуває значення одиниці.

Для того, щоб утворити з таблиці істиності ДДНФ необхідно:

визначити з таблиці істиності, скільки разів ( $N$ ) функція набуває значення одиниці;

написати рівняння

$$F = abc\dots z \vee \bar{a}bc\dots z \vee abc\dots \bar{z} \vee \dots \vee abc\dots z,$$

**Таблиця 9.4. Довільна ФАЛЗ**

Номер набору	Аргументи			Функція	Терм
	$a$	$b$	$c$	$f$	
0	0	0	0	0	
1	0	0	1	0	
2	0	1	0	1	$\bar{a} \cdot b \cdot \bar{c}$
3	0	1	1	1	$\bar{a} \cdot b \cdot c$
4	1	0	0	1	$a \cdot \bar{b} \cdot \bar{c}$
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	1	$a \cdot b \cdot c$

де добуток (терм)  $abc\dots z$  містить усі аргументи функції і повторюється  $N$  разів.

При цьому перший терм рівняння відповідає першому набору таблиці істиності, на

якому ФАЛ набуває значення одиниці, другий – другому, і так далі, а останній, відповідно, останньому;

розставити позначки інверсії змінних в кожному термі згідно із значенням цих змінних у відповідних наборах таблиці істиності:

якщо змінна у відповідному наборі таблиці істиності має значення 0, то у термі вона повинна мати позначку інверсії (інверсне значення),

якщо у наборі змінна має значення 1, то у термі змінна повинна бути без інверсії (пряме значення).

Наприклад, ФАЛ задана таблицею істиності (таблиця 9.). Скласти ДДНФ для даної функції.

Рішення:  $f = \bar{a} \cdot b \cdot \bar{c} \vee \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot b \cdot c$ .

Функція набуває значення одиниці чотири рази, тому ДДНФ містить чотири терми.

Функціональна схема, яка відповідає ДДНФ даної ФАЛ у базисі Буля (рис. 9.6) складається з трьох послідовно з'єднаних частин:

матриці інверторів, яка реалізує інверсні значення змінних;

матриці елементів І, яка реалізує окремі терми ФАЛ;

елемента АБО, на виході якого власне і реалізується ФАЛ.

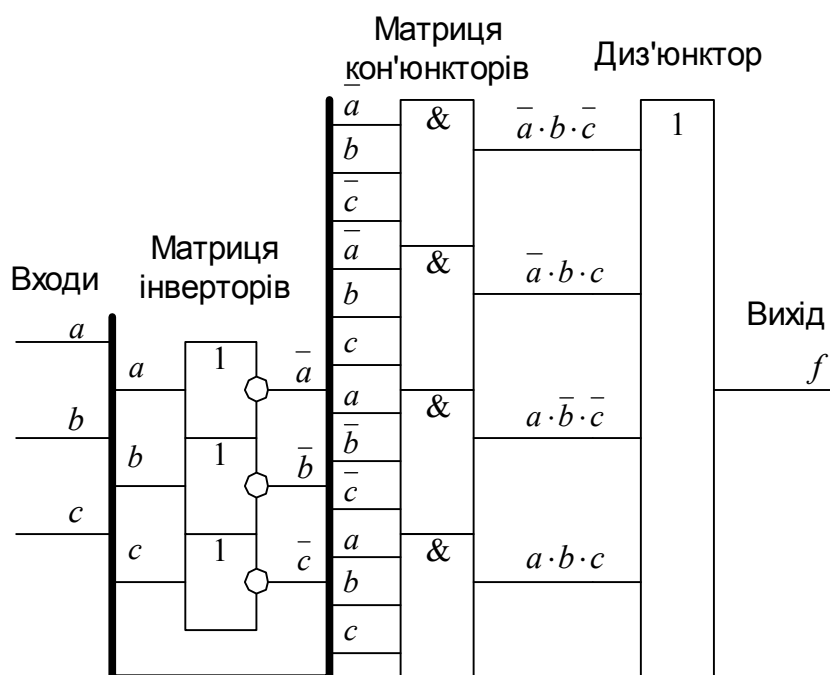


Рис. 9.6. Схема на елементах базису Буля



## 9.4 Синтез логічних схем з одним виходом у базисі Буля на елементах з обмеженою кількістю входів

Таблиця 9.5. Функції І та АБО на 2 входи

Номер набору	Аргументи		Функція	
	$a$	$b$	І	АБО
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	1

На практиці дуже часто необхідно синтезувати функціональну схему, яка реалізує задану функцію, з використанням елементів І та АБО, які мають менше входів ніж потрібно. Наприклад, елементи мають тільки по 2 входи (елементи НЕ завжди мають тільки 1 вхід). Таблиця 9. – це таблиця істинності двовходових елементів І та АБО. Рис. 9.7 містить умовні графічні позначення відповідних елементів.

Рис. 9.8 показує каскадну реалізацію функцій 3І ( $f = abc$ ) та 4І ( $f = abcd$ ) на елементах І з двома входами. Рис. 9.9 показує каскадну реалізацію функцій 3АБО ( $f = a \vee b \vee c$ ) та 4АБО ( $f = a \vee b \vee c \vee d$ ) на елементах АБО з двома входами. Приклад реалізації функції  $f = abc \vee deh \vee ijk$  показує рис. 9.10.

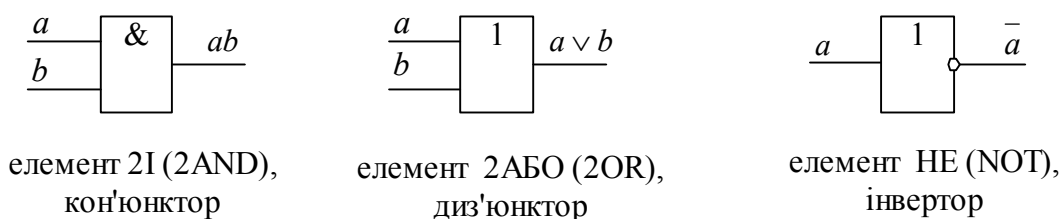


Рис. 9.7. Двовходові елементи І та АБО базису Буля і елемент НЕ

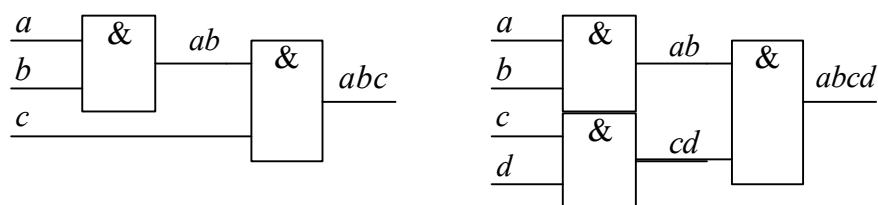


Рис. 9.8. Багатовходові схеми І

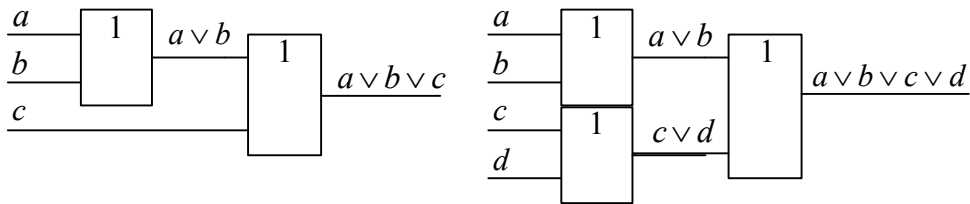


Рис. 9.9. Багатовходові схеми АБО

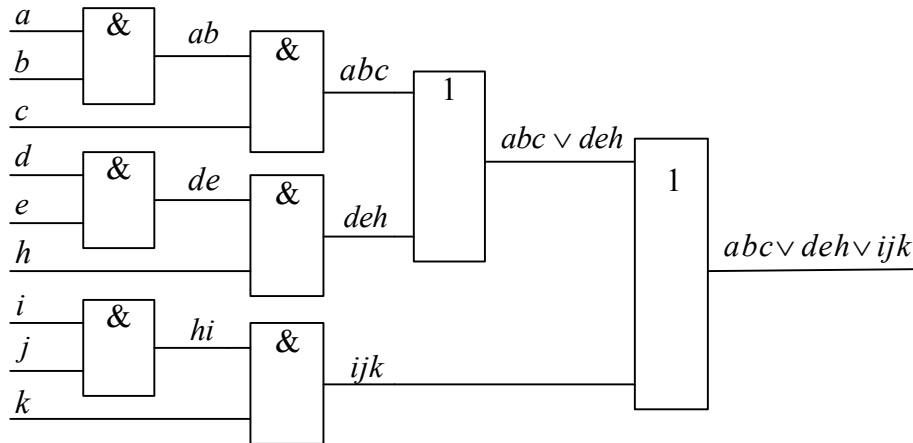


Рис. 9.10. Схема на двовходових елементах І та АБО

### 9.5 Синтез логічних схем з одним виходом у монобазисі І-НЕ

Функція алгебри логіки І-НЕ утворює монобазис (тобто, функціонально повну систему функцій алгебри логіки, яка складається з однієї функції). За допомогою функції І-НЕ можна реалізувати будь-яку іншу функцію алгебри логіки.

Елементи монобазиса І-НЕ повинні мати кількість входів не менше 2. При одному вході елемент І-НЕ перетворюється на інвертор. Відома інша назва цієї функції - заперечення кон'юнкції. Багатовходовий елемент  $n$ І-НЕ (символ  $n$  у назві вказує на кількість входів елемента) реалізує функцію  $f = \overline{a \cdot b \cdot \dots \cdot z}$  - функцію І-НЕ  $n$  змінних. Згідно з правилом Моргана  $f = \overline{a \cdot b \cdot \dots \cdot z} = \bar{a} \vee \bar{b} \vee \dots \vee \bar{z}$ . Тобто, можуть існувати 2 абсолютно рівноправні умовні графічні позначення (УГП, символи) цього елемента: перше - як елемента  $n$ І-НЕ, друге - як елемента НЕ- $n$ АБО (рис. 9.11).

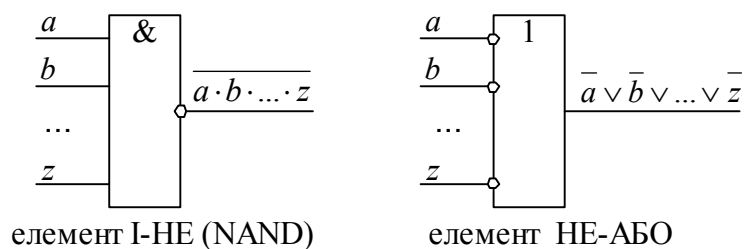


Рис. 9.11 - Елементи монобазису І-НЕ

На виході елемента І-НЕ буде ноль тільки тоді, коли на всіх його входах  $a, b, c, \dots, z$  є одиниця (таблиця 9.).

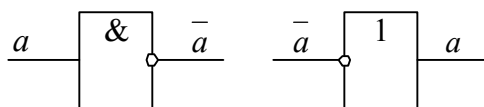
**Таблиця 9.6. Інверсія кон'юнкції багатьох змінних**

Аргументи				Функція І-НЕ
$a$	$b$	...	$z$	$\overline{ab\dots z} = \bar{a} \vee \bar{b} \vee \dots \vee \bar{z}$
0	x	...	x	1
x	0	...	x	1
...	...	...	...	...
x	x	...	0	1
1	1	...	1	0

На елементах І-НЕ можна реалізувати функції, які утворюють базис Буля (І, АБО, НЕ).

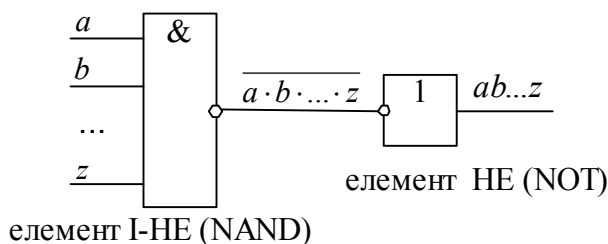
Реалізація функції інверсії у монобазисі І-НЕ відбувається відповідно до формули:

$f = \bar{a}$  на елементі, який має 1 вхід (на інверторі, рис. 9.12).



*Рис. 9.12. Інвертор у монобазисі І-НЕ*

Реалізація функції І  $n$  змінних у монобазисі  $n$ І-НЕ відбувається відповідно до формули  $f = ab\dots z = \overline{\overline{ab\dots z}}$ , як послідовність операцій  $n$ І-НЕ та інверсії. Схемна реалізація наведена на рис. 9.13.



*Рис. 9.13. Схема І на елементах монобазиса І-НЕ*

Реалізація функції АБО  $n$  змінних у монобазисі  $n$ І-НЕ відбувається відповідно до формули  $f = \bar{a} \vee \bar{b} \vee \dots \vee \bar{z}$ , як послідовність операцій інверсії кожної змінної та НЕ- $n$ АБО. Схемна реалізація наведена нижче (рис. 9.14).

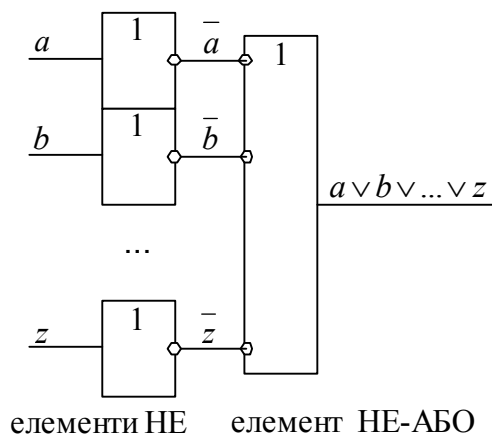


Рис. 9.14. Схема АБО на елементах монобазиса І-НЕ

Реалізація функції І-АБО в монобазисі І-НЕ відбувається як послідовність операцій І-НЕ та НЕ-АБО з потрібною кількістю входів. Наприклад, потрібно реалізувати задану у вигляді диз'юнктивної нормальної форми (ДНФ) функцію  $f = abc \vee deh \vee ijk$ . Результат реалізації показує рис. 9.15. Доречно звернути увагу, на те, що на всіх наведених рисунках прямі виходи з'єднуються із прямими входами, а інверсні — з інверсними.

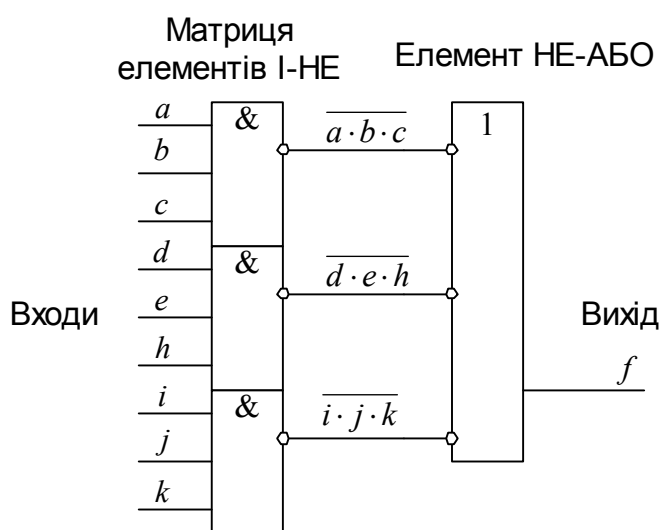


Рис. 9.15. Схема реалізації довільної ФАЛ на елементах монобазиса І-НЕ

Таблиця 9.7. Монобазис І-НЕ

Номер набору	Аргументи		Функція
	$a$	$b$	І-НЕ
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

## 9.6 Синтез логічних схем з одним виходом у монобазисі 2І-НЕ (Шеффера)

Елементи монобазиса 2І-НЕ мають два входи. Відомі інші назви цієї функції:  
заперечення кон'юнкції,  
елемент Шеффера,  
штрих Шеффера.

Двовходовий елемент 2І-НЕ (символ 2 у назві вказує на кількість входів елемента) реалізує функцію  $f = \overline{ab}$  - функцію І-НЕ двох змінних. Згідно з правилом Моргана  $\overline{ab} = \overline{a} \vee \overline{b}$ . Тобто можуть існувати 2 абсолютно рівноправні умовні графічні позначення (УГП, символи) цього елемента: перше, як елемента 2І-НЕ, друге - як елемента НЕ-2АБО (рис. 9.1).

Таблиця 9. містить таблицю істинності елемента 2І-НЕ.

На елементах 2І-НЕ можна реалізувати функції, які утворюють базис Буля (І, АБО, НЕ).

Реалізація функції інверсії у монобазисі 2І-НЕ відбувається відповідно до формули  $f = \overline{aa} = \overline{a}$ . Два варіанти схемної реалізації інверсії показує рис. 9.16: на обидва входи елемента треба подати сигнал, який потрібно інвертувати.

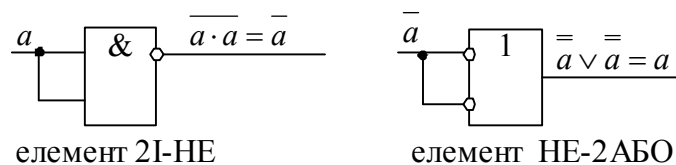


Рис. 9.16. Схема реалізації інверсії на елементах 2І-НЕ

Реалізація функції І в монобазисі 2І-НЕ відбувається відповідно до формули  $f = ab = \overline{\overline{ab}}$ , як послідовність операцій 2І-НЕ та інверсії. Схемну реалізацію показує рис. 9.17.

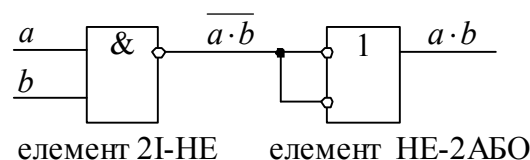


Рис. 9.17. Схема реалізації кон'юнкції на елементах 2І-НЕ

Реалізація функції АБО в монобазисі 2І-НЕ відбувається відповідно до формули  $f = \overline{\overline{a} \vee \overline{b}}$ , як послідовність операцій інверсії кожної змінної та НЕ-2АБО. Схемну

реалізацію показує рис. 9.18.

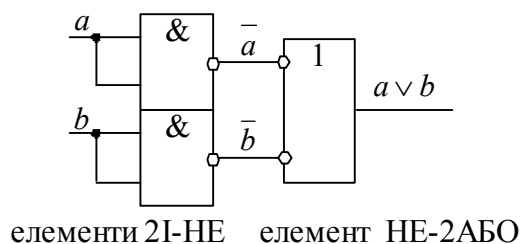


Рис. 9.18. Схема реалізації диз'юнкції на елементах 2І-НЕ

Реалізація функції І-АБО в монобазисі 2І-НЕ відбувається відповідно до формули  $f = \overline{\overline{ab} \vee \overline{cd}}$  як послідовність операцій 2І-НЕ та НЕ-2АБО. Схемну реалізацію показує рис. 9.19.

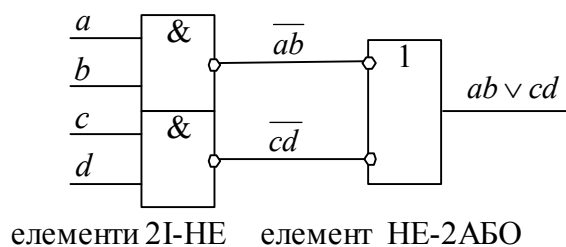


Рис. 9.19. Схема реалізації кон'юнкції та диз'юнкції на елементах 2І-НЕ

Рис. 9.20 показує каскадну реалізацію функції 3І-НЕ  $f = \overline{abc}$ .

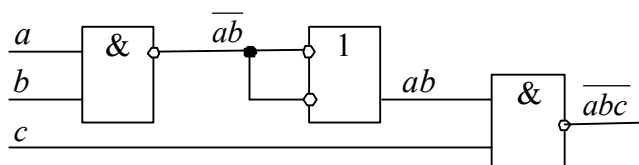


Рис. 9.20. Схема 3І-НЕ на елементах 2І-НЕ

Рис. 9.21 показує каскадну реалізацію функції 4І-НЕ  $f = \overline{abcd}$ .

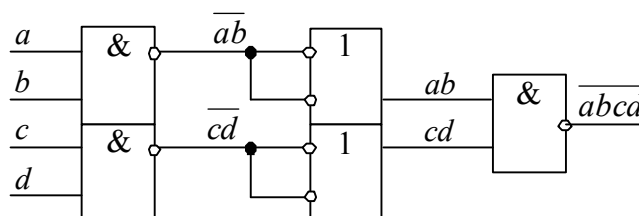


Рис. 9.21. Схема 4І-НЕ на елементах 2І-НЕ

Рис. 9.22 показує каскадну реалізацію функції 3АБО  $f = a \vee b \vee c$ .

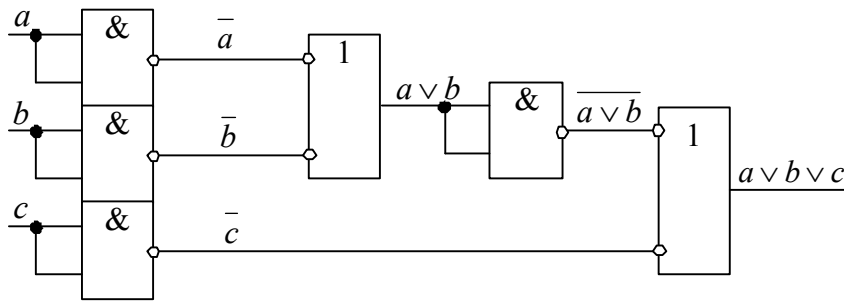


Рис. 9.22. Схема 3АБО на елементах 2I-НЕ

Рис. 9.23 показує каскадну реалізацію функції 4АБО  $f = a \vee b \vee c \vee d$ .

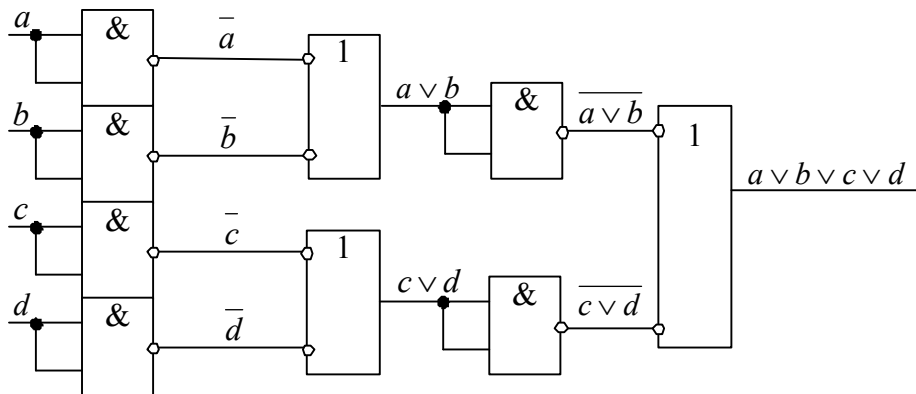


Рис. 9.23. Схема 4АБО на елементах 2I-НЕ

Результат реалізації функції  $f = abc \vee deh \vee ijk$  показує рис. 9.24. Доречно звернути увагу, на те, що на всіх наведених рисунках прямі виходи з'єднуються із прямими входами, а інверсні — з інверсними.

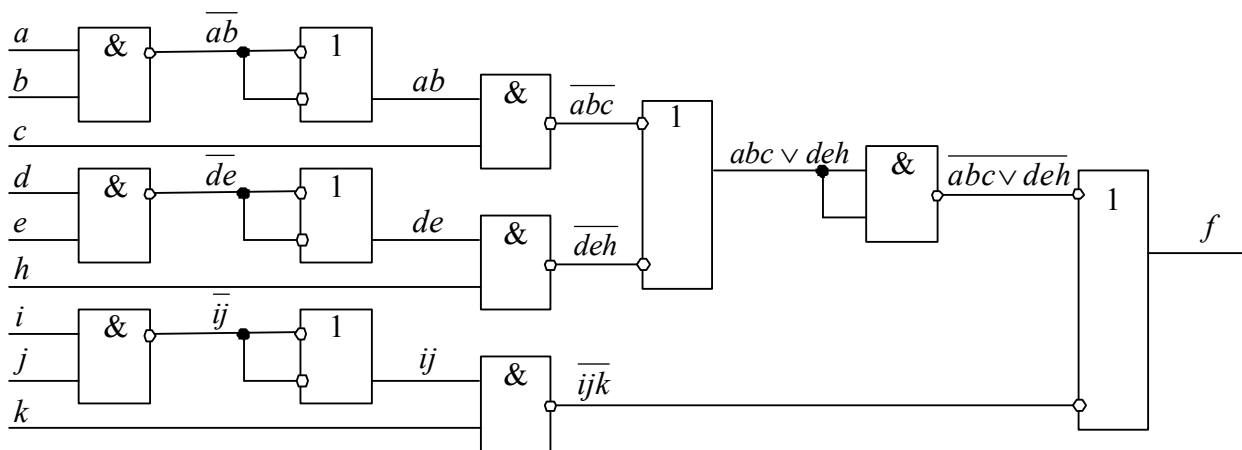


Рис. 9.24. Схема ФАЛ на елементах 2I-НЕ

### 9.7 Синтез логічних схем з одним виходом у монобазисі АБО-НЕ

Функція алгебри логіки АБО-НЕ утворює монобазис (тобто, функціонально повну систему функцій алгебри логіки, яка складається з однієї функції). За

допомогою функції АБО-НЕ можна реалізувати будь-яку іншу функцію алгебри логіки.

Елементи монобазиса АБО-НЕ повинні мати кількість входів не менше 2. При одному вході елемент АБО-НЕ перетворюється на інвертор. Відома інша назва цієї функції - заперечення диз'юнкції. Багатовходовий елемент  $n$ АБО-НЕ (символ  $n$  у назві вказує на кількість входів елемента) реалізує функцію  $f = \overline{a \vee b \vee \dots \vee z}$  - функцію АБО-НЕ  $n$  змінних. Згідно з правилом Моргана  $f = \overline{a \vee b \vee \dots \vee z} = \bar{a} \cdot \bar{b} \cdot \dots \cdot \bar{z}$ . Тобто можуть існувати 2 абсолютно рівноправні умовні графічні позначення (УГП) цього елемента: перше - як елемента  $n$ АБО-НЕ, друге - як елемента НЕ- $n$ I (рис. 9.25).

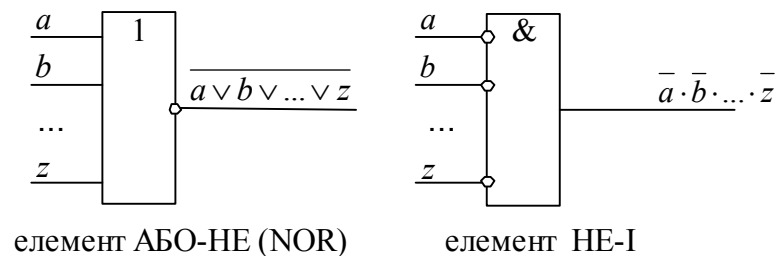


Рис. 9.25. Елементи монобазиса АБО-НЕ

На виході елемента АБО-НЕ буде одиниця тільки тоді, коли на всіх його входах  $a, b, c, \dots, z$  є нулі (таблиця 9.).

**Таблиця 9.8. Інверсія диз'юнкції багатьох змінних**

Аргументи				Функція АБО-НЕ
$a$	$b$	...	$z$	$\overline{ab\dots z}$
0	0	...	0	1
x	x	...	1	0
...	...	...	...	...
x	1	...	x	0
1	x	...	x	0

На елементах АБО-НЕ можна реалізувати функції, які утворюють базис Буля (І, АБО, НЕ).

Реалізація функції інверсії у монобазисі АБО-НЕ відбувається відповідно до формули:  $f = \bar{a}$  на елементі, який має 1 вхід (на інверторі, рис. 9.26).



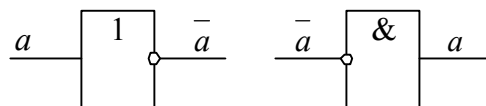


Рис. 9.26. Інвертор у монобазисі АБО-НЕ

Реалізація функції АБО  $n$  змінних у монобазисі  $n$ АБО—НЕ відбувається відповідно до формули  $f = a \vee b \vee \dots \vee z = \overline{\overline{a \vee b \vee \dots \vee z}}$ , як послідовність операцій  $n$ АБО-НЕ та інверсії. Схемну реалізацію показує рис. 9.27.

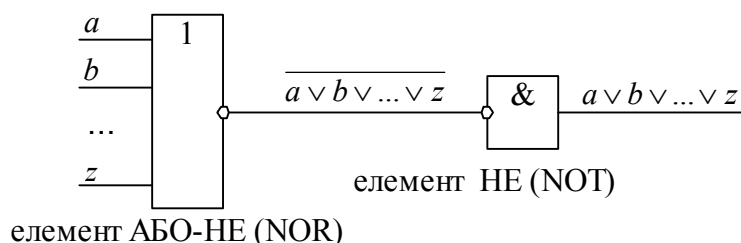


Рис. 9.27. Схеми АБО на елементах монобазиса АБО-НЕ

Реалізація функції І  $n$  змінних у монобазисі  $n$ АБО-НЕ відбувається відповідно до формули  $f = \overline{\overline{a \cdot b \cdot \dots \cdot z}}$ , як послідовність операцій інверсії кожної змінної та НЕ-І. Схемна реалізація наведена нижче (рис. 9.28).

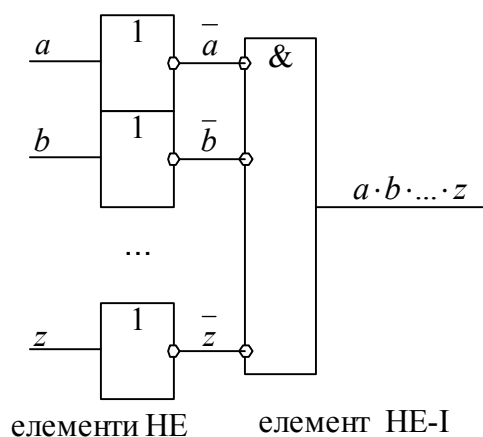


Рис. 9.28. Схеми І на елементах монобазиса АБО-НЕ

Реалізація функції АБО-І в монобазисі АБО-НЕ відбувається як послідовність операцій АБО-НЕ та НЕ-АБО з потрібною кількістю входів. Наприклад, реалізувати задану у вигляді кон'юнктивної нормальної форми (КНФ) функцію  $f = (a \vee b \vee c) \& (d \vee e \vee h) \& (i \vee j \vee k)$ . Результат реалізації показує рис. 9.29. Доречно звернути увагу, на те, що на всіх наведених рисунках прямі виходи з'єднуються із прямими входами, а інверсні — з інверсними.

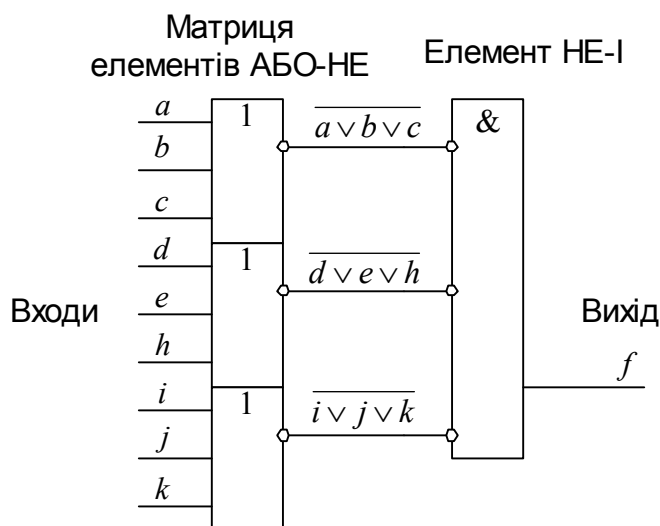


Рис. 9.29. Схема реалізації довільної ФАЛ на елементах монобазиса АБО-НЕ

Таблиця 9.9. Монобазис АБО-НЕ

Номер набору	Аргументи		Функція
	$a$	$b$	АБО-НЕ
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

### 9.8 Синтез логічних схем з одним виходом у монобазисі 2АБО-НЕ (Пірса)

Елементи монобазиса 2АБО-НЕ мають 2 входи. Відомі інші назви функції 2АБО-НЕ:

заперечення диз'юнкції,

елемент Пірса,

стрілка Пірса.

Двовходовий елемент 2АБО-НЕ (символ 2 у назві вказує на кількість входів елемента) реалізує функцію  $f = \overline{a \vee b}$  - функцію АБО-НЕ двох змінних. Згідно з правилом Моргана  $\overline{a \vee b} = \bar{a} \cdot \bar{b}$ . Тобто, можуть існувати 2 абсолютно рівноправні умовні графічні позначення (УГП) цього елемента (рис. 9.2):

перше, як елемента 2АБО-НЕ,

друге - як елемента НЕ-2І.

Таблиця 9. істинності елемента 2АБО-НЕ пояснює його роботу.

На елементах 2АБО-НЕ можна реалізувати функції, які утворюють базис Буля (І,

АБО, НЕ).

Реалізація функції інверсії в монобазисі 2АБО-НЕ відбувається відповідно до формули  $\overline{a \vee a} = \bar{a}$ . Схемну реалізацію інверсії показує рис. 9.30: на обидва входи елемента треба подати сигнал, який потрібно інвертувати.

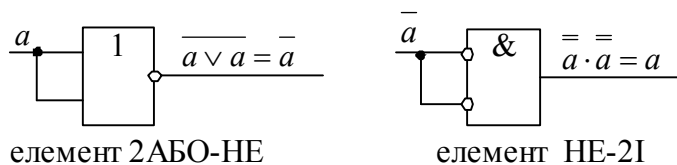


Рис. 9.30. Схема реалізації інверсії на елементах 2АБО-НЕ

Реалізація функції АБО в монобазисі 2АБО-НЕ відбувається відповідно до формули  $f = \overline{\overline{a \vee b}}$  як послідовність операцій 2АБО-НЕ та інверсії (рис. 9.31).

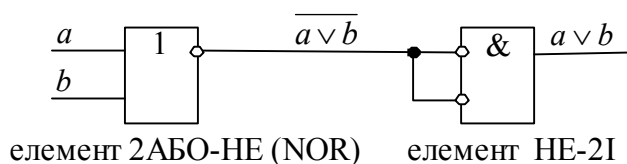


Рис. 9.31. Схема реалізації диз'юнкції на елементах 2АБО-НЕ

Реалізація функції І в монобазисі 2АБО-НЕ відбувається відповідно до формули  $f = \overline{\overline{a \vee b}}$  як послідовність операцій інверсії кожної змінної та НЕ-2І (рис. 9.32).

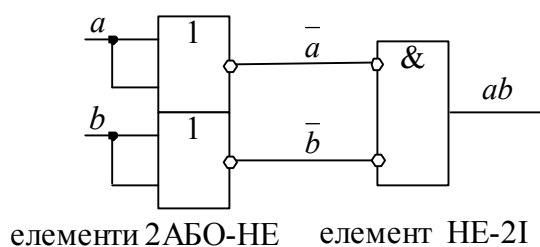


Рис. 9.32. Схема реалізації кон'юнкції на елементах 2АБО-НЕ

Реалізація функції АБО-І в монобазисі 2АБО-НЕ відбувається відповідно до формули  $f = \overline{\overline{a \vee b} \cdot \overline{c \vee d}}$  як послідовність операцій 2АБО-НЕ та НЕ-2І (рис. 9.33).

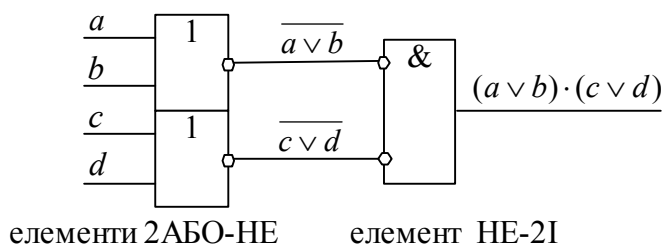


Рис. 9.33. Схема реалізації диз'юнкції та кон'юнкції на елементах 2АБО-НЕ

Реалізація функції 3АБО-НЕ  $f = \overline{a \vee b \vee c}$  каскадна (рис. 9.34).

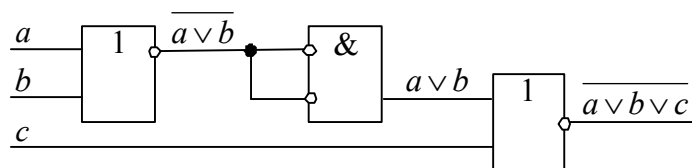


Рис. 9.34. Схема 3АБО-НЕ на елементах 2АБО-НЕ

Реалізація функції 4АБО-НЕ  $f = \overline{a \vee b \vee c \vee d}$  каскадна (рис. 9.35).

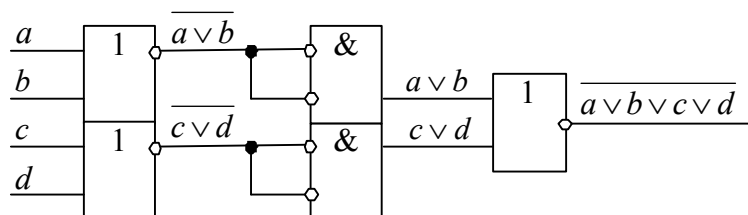


Рис. 9.35. Схема 4АБО-НЕ на елементах 2АБО-НЕ

Реалізація функції 3І  $f = abc$  каскадна (рис. 9.36).

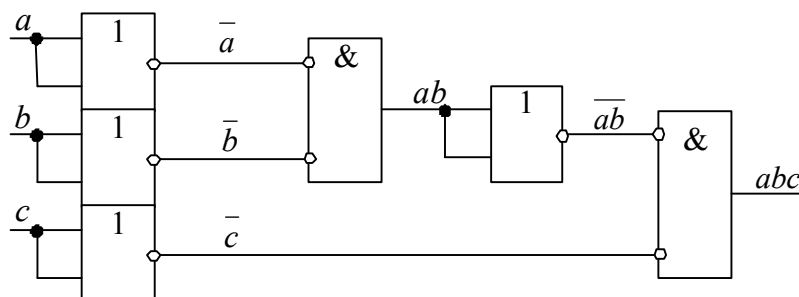


Рис. 9.36. Схема 3І на елементах 2АБО-НЕ

Реалізація функції 4І  $f = abcd$  каскадна (рис. 9.37).

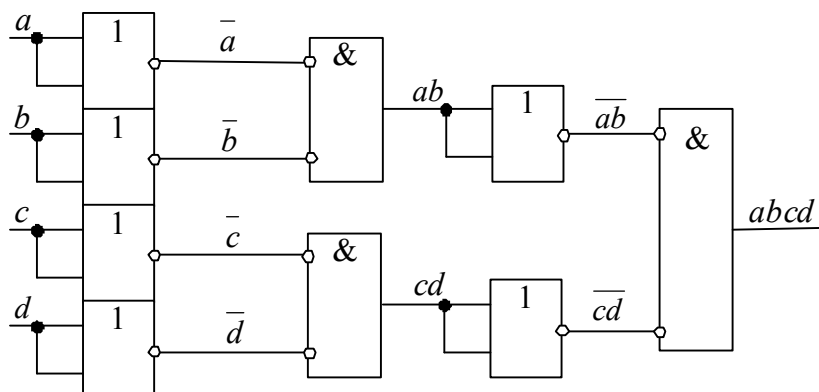


Рис. 9.37. Схема 4І на елементах 2АБО-НЕ

Рис. 9.38 показує реалізацію функції  $f = (a \vee b \vee c)(d \vee e \vee h)(i \vee j \vee k)$ . Доречно звернути увагу, що на всіх наведених рисунках прямі виходи з'єднуються із прямими входами, а інверсні - з інверсними.

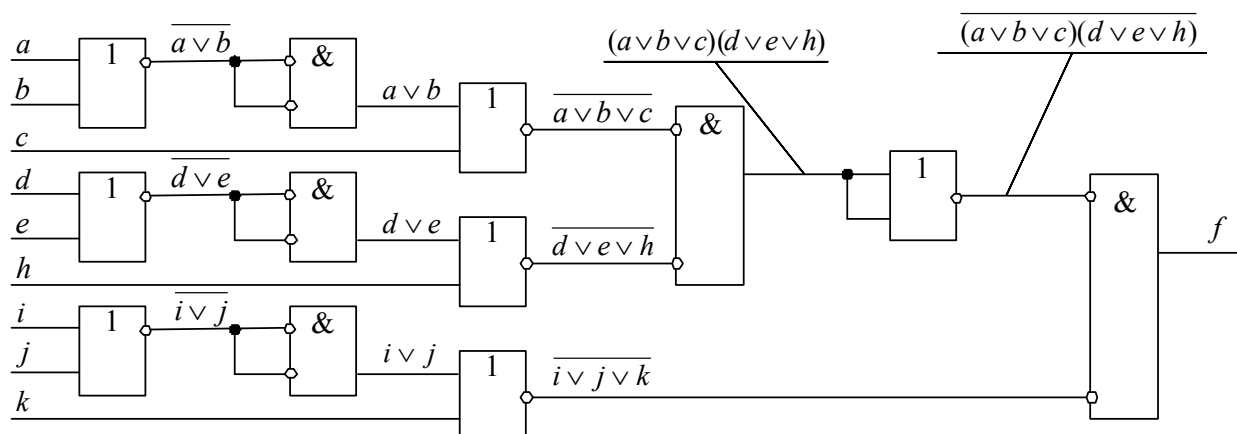


Рис. 9.38. Схема ФАЛ на елементах 2АБО-НЕ

### 9.9 Порівняння варіантів синтезу комбінаційних логічних схем

Схеми з одним виходом і декількома входами відносяться до найбільш простих схем. Основна складність при синтезі цих схем полягає в тому, щоб знайти вираз для вихідної функції в заданому базисі.

Розглянемо деякі прості приклади переходу від логічних рівнянь до логічних схем, тобто приклади синтезу простих логічних схем. Зокрема, розглянемо перехід від представлення функції в ДНФ до її реалізації на елементах І-НЕ і АБО-НЕ.

Нехай задана функція 4-х змінних у ДНФ:

$$f = \bar{c} \cdot d \vee \bar{a} \cdot \bar{b} \cdot \bar{d} \vee a \cdot b \cdot \bar{d} \vee \bar{a} \cdot b \cdot \bar{c} \vee a \cdot \bar{c} \cdot \bar{d}.$$

Проведемо її мінімізацію за допомогою карти Карно (рис. 9.39).

Терму  $\bar{c} \cdot d$  відповідають клітинки 1, 5, D, 9;

терму  $\bar{a} \cdot \bar{b} \cdot \bar{d}$  - клітинки 0, 2;

терму  $a \cdot b \cdot \bar{d}$  - клітинки C, E;

терму  $\bar{a} \cdot b \cdot \bar{c}$  - клітинки 4, 5;

терму  $a \cdot \bar{c} \cdot \bar{d}$  - клітинки C, 8.

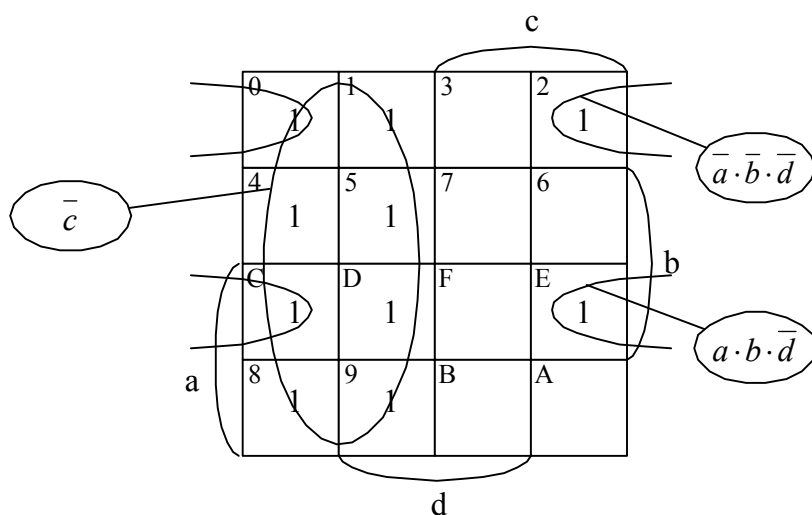


Рис. 9.39. Мінімізація ФАЛ за одиницями

Мінімальна ДНФ має вигляд:  $f = \bar{a} \cdot \bar{b} \cdot \bar{d} \vee a \cdot b \cdot \bar{d} \vee \bar{c}$ .

З карти Карно може бути побудована таблиця 9. істинності. Також користуючись картою Карно можна проводити мінімізацію і в базисі Жегалкіна. Відмінність в порівнянні з базисом Буля полягає в тому, що ніяка клітинка з одиничним значенням не повинна покриватися парною кількістю склеювань (рис. 9.40). Результатом мінімізації є поліном  $f = \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} \oplus a \cdot b \cdot c \cdot \bar{d} \oplus \bar{c} = (1 \oplus a)(1 \oplus b)c(1 \oplus d) \oplus a \cdot b \cdot c \cdot (1 \oplus d) \oplus (1 \oplus c)$  (цей результат утворюється після застосування правила  $\bar{x} = 1 \oplus x$ ).

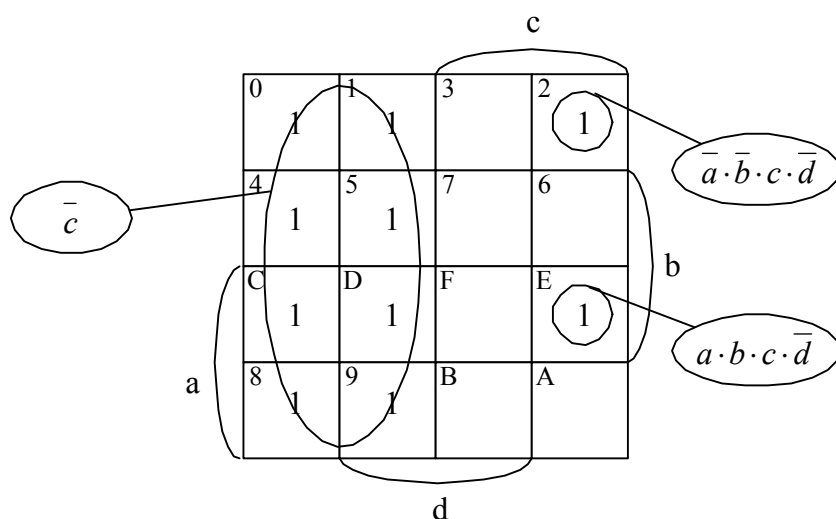


Рис. 9.40. Мінімізація ФАЛ для базису Жегалкіна

Таблиця 9.10. Таблиця істинності ФАЛ

Номер набору	a	b	c	d	f	Елементарні кон'юнкції

0	0	0	0	0	1	$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$
1	0	0	0	1	1	$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d$
2	0	0	1	0	1	$\bar{a} \cdot \bar{b} \cdot c \cdot \bar{d}$
3	0	0	1	1	0	
4	0	1	0	0	1	$\bar{a} \cdot b \cdot \bar{c} \cdot \bar{d}$
5	0	1	0	1	1	$\bar{a} \cdot b \cdot \bar{c} \cdot d$
6	0	1	1	0	0	
7	0	1	1	1	0	
8	1	0	0	0	1	$a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$
9	1	0	0	1	1	$a \cdot \bar{b} \cdot \bar{c} \cdot d$
A	1	0	1	0	0	
B	1	0	1	1	0	
C	1	1	0	0	1	$a \cdot b \cdot \bar{c} \cdot \bar{d}$
D	1	1	0	1	1	$a \cdot b \cdot \bar{c} \cdot d$
E	1	1	1	0	1	$a \cdot b \cdot c \cdot \bar{d}$
F	1	1	1	1	0	

Розглянемо реалізацію рівняння  $f = \bar{a} \cdot \bar{b} \cdot \bar{d} \vee a \cdot b \cdot \bar{d} \vee \bar{c}$  за допомогою елементів І-НЕ. У загальному випадку на елементах І-НЕ ДНФ реалізується за допомогою двох ступіней (каскадів) логіки. На першій ступіні виходять інверсні значення логічних добуток. На другій ступіні виконуються операції І-НЕ, тобто НЕ-АБО, над отриманими інверсіями. Схема, що відповідає даному рівнянню, наведена нижче (рис. 9.41).

Цю ж мінімальну ДНФ можна реалізувати і на елементах АБО-НЕ. На елементах АБО-НЕ ДНФ функції реалізується за допомогою трьох ступіней (рис. 9.42). На першій ступіні за допомогою операції НЕ-І над інверсними значеннями змінних, вхідних у ДНФ, утворюються логічні добутки. На другій ступіні виконується операція АБО-НЕ над логічними добутками й однобуквеними членами (якщо вони є), тим самим утвориться інверсне значення функції. На третій ступіні виконується інверсія і утворюється шукана функція.

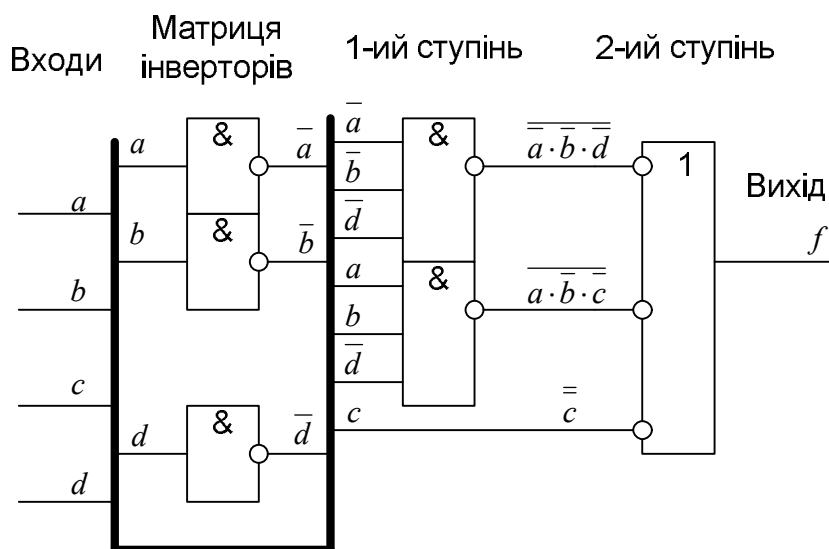


Рис. 9.41. Схема ДНФ ФАЛ на елементах І-НЕ

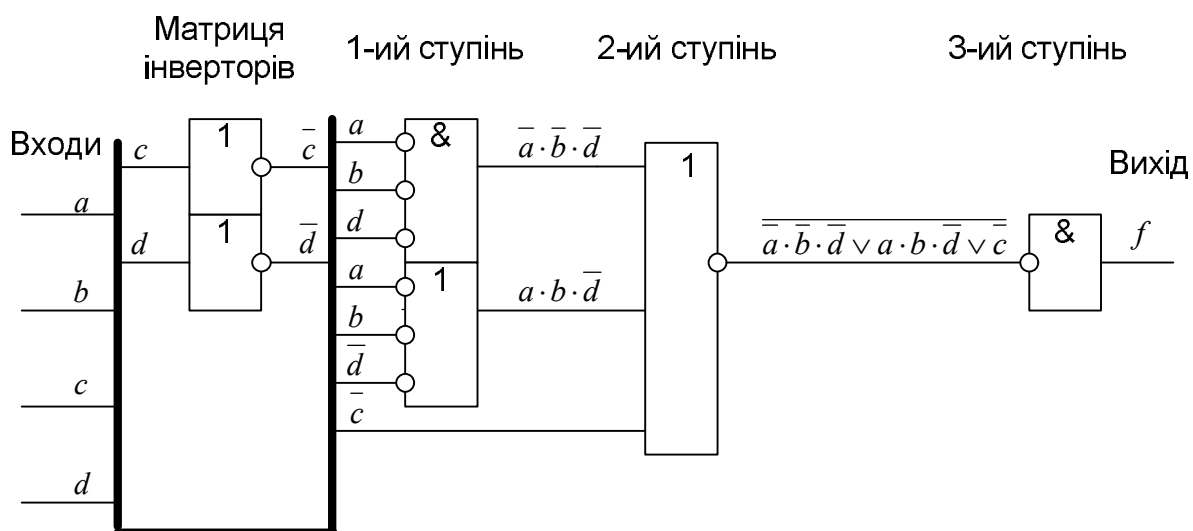


Рис. 9.42. Схема ДНФ ФАЛ на елементах АБО-НЕ

У наведеній схемі для елементів першої і другої ступіні застосовані різні, але еквівалентні умовні позначки. При реалізації ДНФ функції за допомогою елементів І-НЕ такий прийом дозволяє вести проектування схем, користуючись операціями І, АБО і НЕ.

За розглянутими раніше правилами з вищенаведеної карти Карно, може бути знайдена мінімальна КНФ заданої функції (рис. 9.43).



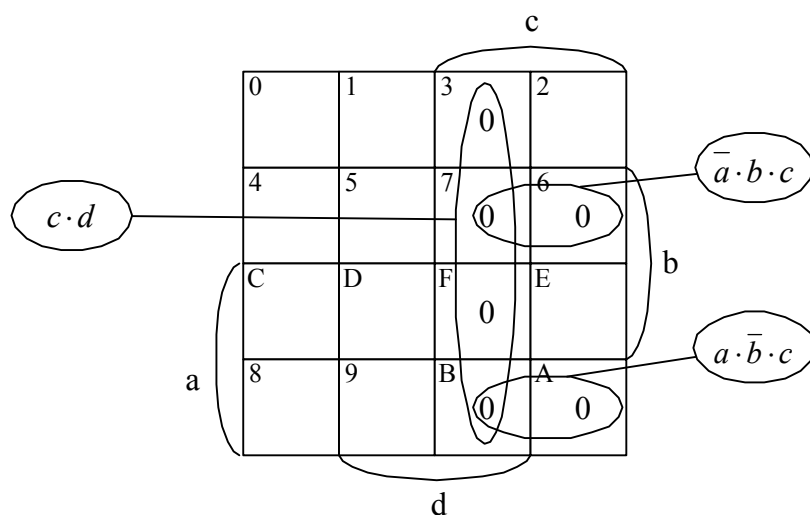


Рис. 9.43. Мінімізація ФАЛ за нулями

Оскільки мінімізація йде за нулями, то спочатку отримуємо інверсне значення функції:

$$\bar{f} = c \cdot d \vee \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot c.$$

Далі за правилами де Моргана отримуємо КНФ даної функції:

$$f = \overline{c \cdot d \vee \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot c} = (\bar{c} \vee \bar{d}) \cdot (a \vee \bar{b} \vee \bar{c}) \cdot (\bar{a} \vee b \vee \bar{c}).$$

На елементах І-НЕ КНФ функції реалізується за допомогою трьох ступіней (рис. 9.44). На першій ступіні за допомогою операції НЕ-АБО над інверсними значеннями змінних, входних у КНФ, утворюються логічні суми. На другій ступіні виконується операція І-НЕ над логічними сумами й однобуквеними членами (якщо вони є), тим самим утвориться інверсне значення функції. На третій ступіні виконується інверсія і утворюється шукана функція.

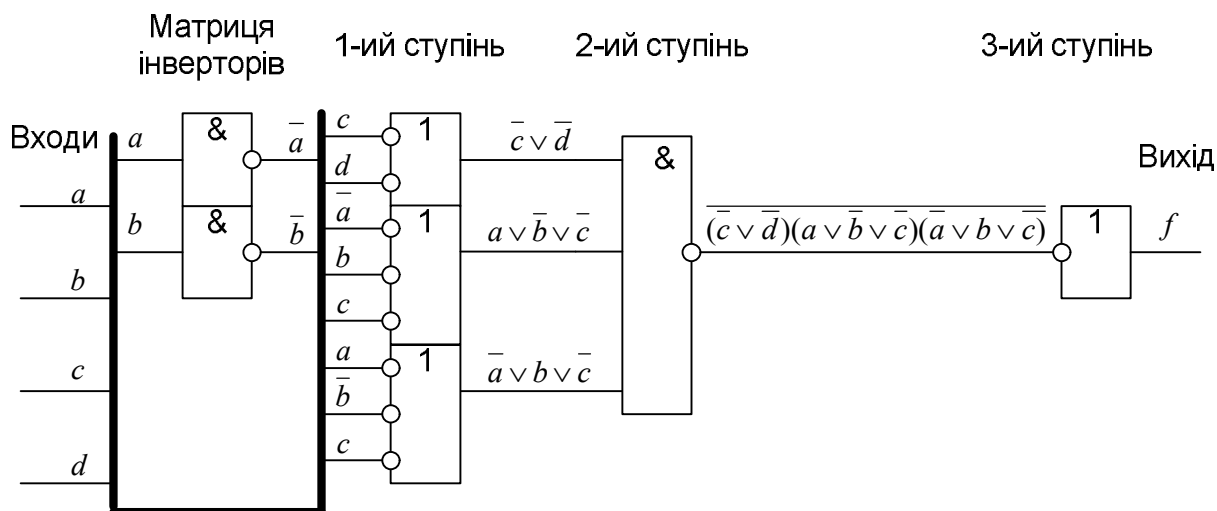


Рис. 9.44. Схема КНФ ФАЛ на елементах І-НЕ

На елементах АБО-НЕ КНФ функції реалізується за допомогою двох ступіней (рис. 9.45). На першій ступіні за допомогою операції АБО-НЕ над значеннями змінних, входних у КНФ, утворюються логічні суми. На другій ступіні виконується операція НЕ-І над логічними сумами й однобуквеними членами (якщо вони є), тим самим утвориться пряме значення функції.

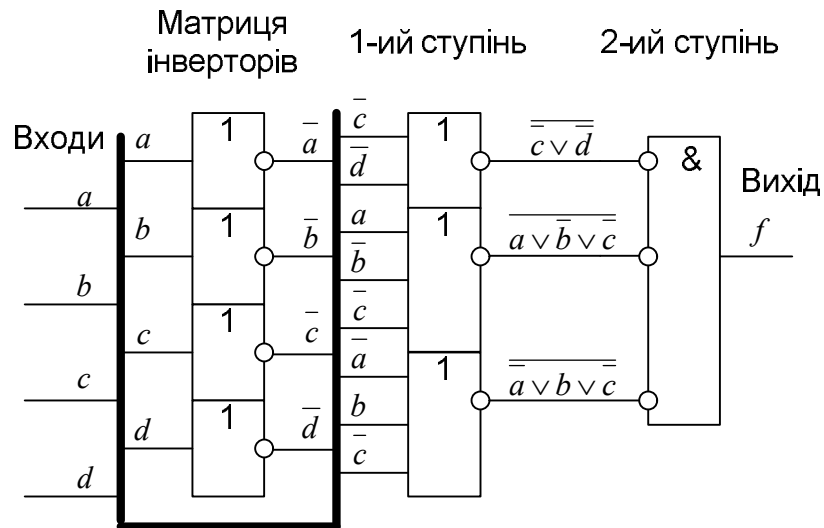


Рис. 9.45. Схема КНФ ФАЛ на елементах АБО-НЕ

При мінімізації логічних функцій для логічних схем, що передбачається будувати на базі елементів І-НЕ або АБО-НЕ, необхідно крім власне мінімізації прагнути також до того, щоб структурна формула була представлена у вигляді комбінації з елементів І-НЕ (НЕ-АБО) або АБО-НЕ (НЕ-І). Тоді перехід від структурної формули до функціональної схеми не буде складним. У будь-якому випадку при побудові логічної схеми в базисі І-НЕ на основі логічної функції, представленої в МДНФ, необхідно скрізь замість елементів І і АБО ставити елементи І-НЕ та НЕ-АБО. При побудові логічної схеми в базисі АБО-НЕ на основі логічної функції, представленої в МКНФ, необхідно скрізь замість елементів І і АБО ставити елементи НЕ-І та АБО-НЕ.

Коли треба обирати для використання якийсь один монобазис, то, як видно з наведених рисунків, ДНФ краще реалізовувати на елементах монобазису І-НЕ, а КНФ – на елементах монобазису АБО-НЕ. При такому підході кількість ступенів буде мінімальна.

Базис Буля з цієї точки зору є універсальним: і ДНФ (наприклад,

$f = \bar{a} \cdot \bar{b} \cdot \bar{d} \vee a \cdot b \cdot \bar{d} \vee \bar{c}$ , рис. 9.46), і КНФ (наприклад,  $f = (\bar{c} \vee \bar{d}) \cdot (a \vee \bar{b} \vee \bar{c}) \cdot (\bar{a} \vee b \vee \bar{c})$ , рис. 9.47) на ньому реалізуються на двох супенях.

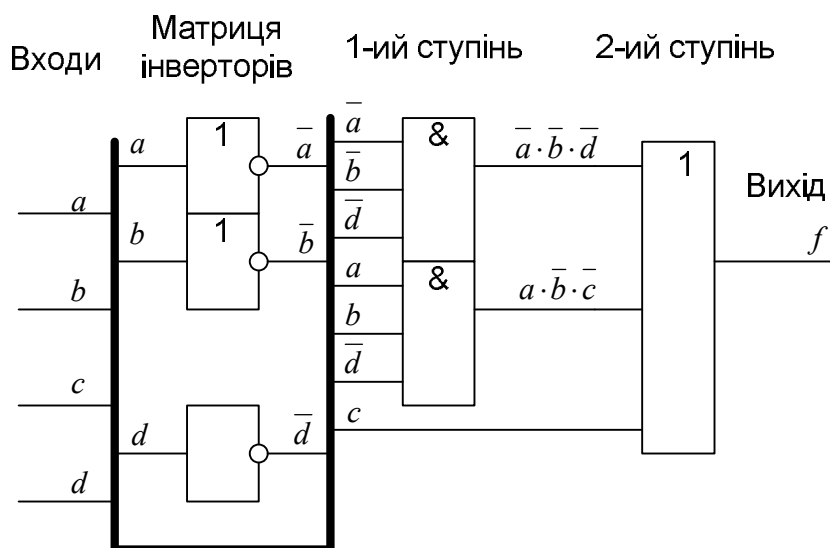


Рис. 9.46. Схема ДНФ ФАЛ на елементах базису Буля

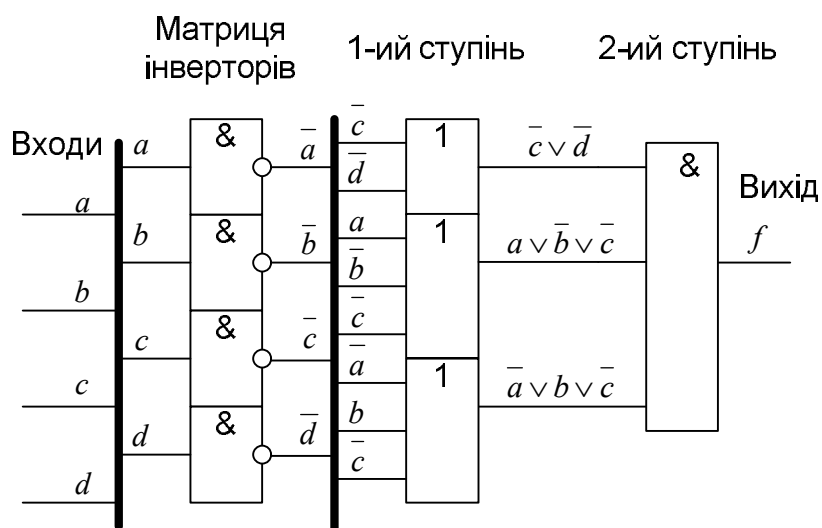


Рис. 9.47. Схема КНФ ФАЛ на елементах базису Буля

### 9.10 Синтез логічних схем з одним виходом у базисі Жегалкіна

Тепер розглянемо способи формування схеми, що реалізує функцію сумування за модулем 2 (функція  $f_6$ ), у різних базисах. Логічна функція  $f_6$  (таблиця 9.), як відомо, в аналітичному виді представляється у ДНФ як  $f = a\bar{b} \vee \bar{a}b$ , а у КНФ як  $f = (\bar{a} \vee \bar{b})(a \vee b)$ .

Таблиця 9.11. Сума за модулем 2

$a$	$b$	$f$
-----	-----	-----

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

У базисі І, АБО, НЕ схема, що реалізує ДНФ  $f_6$ , має вигляд (рис. 9.48).

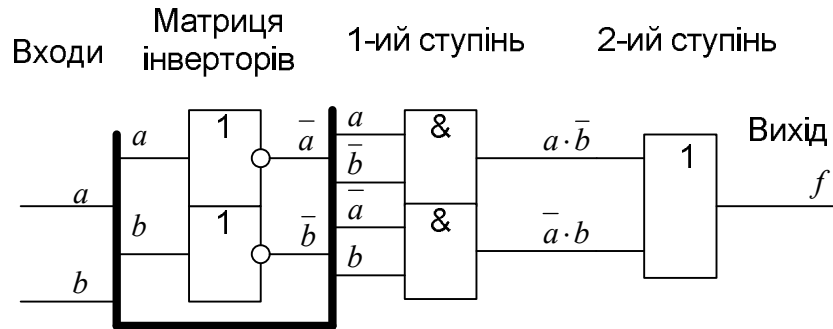


Рис. 9.48. Суматор за модулем 2 на елементах базису Буля

У базисі І-НЕ схема, що реалізує ДНФ  $f_6$ , має вигляд (рис. 9.49).

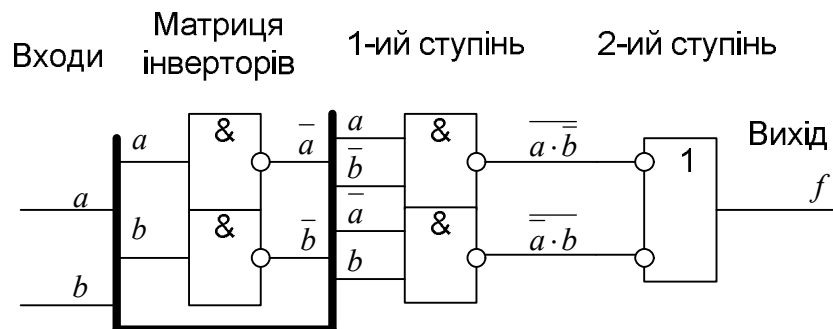


Рис. 9.49. Суматор за модулем 2 на елементах базису І-НЕ

У базисі АБО-НЕ схема, що реалізує КНФ  $f_6$ , має вигляд (рис. 9.50).

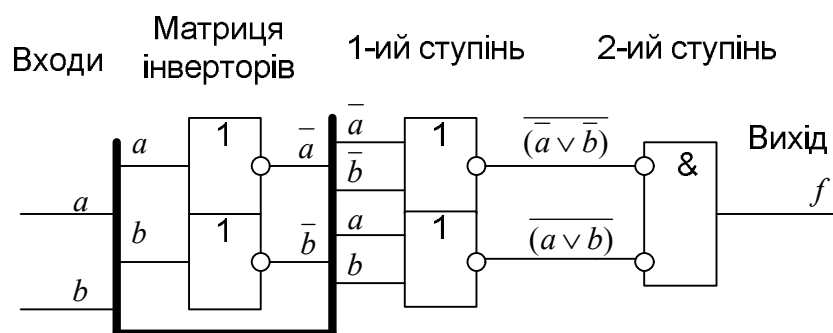


Рис. 9.50. Суматор за модулем 2 на елементах базису АБО-НЕ

Релізація функції  $f = (1 \oplus a)(1 \oplus b)c(1 \oplus d) \oplus a \cdot b \cdot c \cdot (1 \oplus d) \oplus (1 \oplus c)$ , яку задає карта Карно (рис. 9.40), показує рис. 9.51.

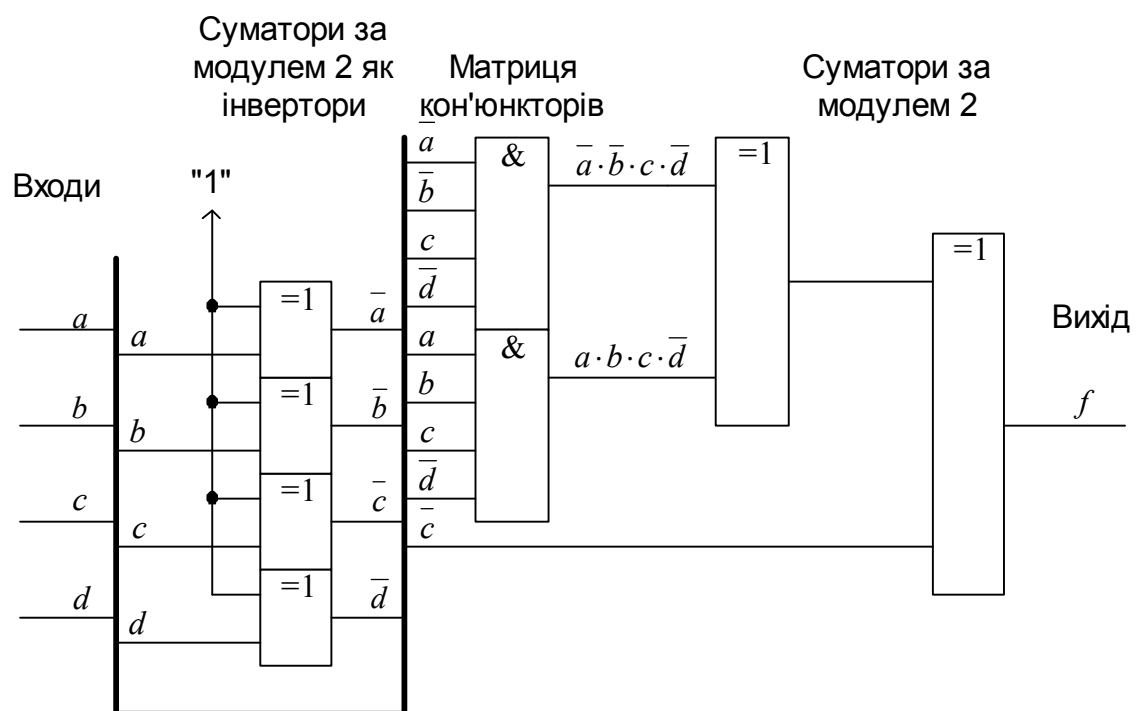


Рис. 9.51. Схема, яка реалізує поліном Жегалкіна

## 10 СИНТЕЗ ЛОГІЧНИХ СХЕМ З ДЕКІЛЬКОМА ВИХОДАМИ

Більш складні схеми, що мають кілька виходів, можуть бути зведені в окремому випадку до набору схем з одним виходом. Тоді синтез здійснюється шляхом декомпозиції для кожної виділюваної схеми.

Задача синтезу схеми з  $n$  входами і  $k$  виходами відрізняється від задачі синтезу  $k$  схем з  $n$  входами й одним виходом тим, що при рішенні необхідно виключити дублювання в  $k$  схемах синтезованих функцій.

### 10.1 Базові комбінаційні вузли

До базових комбінаційних вузлів належать:

- суматори і напівсуматори;
- вузли порівняння;
- дешифратори і демультимплексори;
- шифратори;
- мультимплексори;
- перетворювачі кодів;
- постійні запам'ятовуючі пристрої;
- програмовані логічні матриці.

### 10.2 Однорозрядний двійковий суматор

Розглянемо для прикладу синтез заданого таблично (таблиця 10.) однорозрядного двійкового суматора методом декомпозиції.

**Таблиця 10.1. Повний двійковий однорозрядний суматор**

$A$	$B$	$C_i$	$S$	$C_0$	$S$	$AB = 00$	$AB = 01$	$AB = 11$	$AB = 10$
0	0	0	0	0	$C_i = 0$	0	1	0	1
0	0	1	1	0	$C_i = 1$	1	0	1	0
0	1	0	1	0					
0	1	1	0	1					
1	0	0	1	0					
1	0	1	0	1	$C_0$	$AB = 00$	$AB = 01$	$AB = 11$	$AB = 10$
1	1	0	0	1	$C_i = 0$	0	0	1	0
1	1	1	1	1	$C_i = 1$	0	1	1	1

На виході  $S$  є результат додавання двох однорозрядних чисел ( $A$  і  $B$ ) і переносу

$C_i$  попереднього суматора. Перенос, що при цьому виникає, подається на вихід  $C_o$ . Синтезовану схему можна розглядати як схему, до складу якої входять дві частини:

схема для одержання по-розрядної суми  $S$ ;

схема для одержання переносу  $C_o$ .

З карт Карно видно, що можливо мінімізувати тільки функцію  $C_o$ . Для складання схеми суматора можуть бути використані частково мінімізовані функції і після різних алгебраїчних перетворень можна одержати наступні диз'юнктивні нормальні форми для функцій  $S$  і  $C_o$ , а також їхні анормальні форми:

$$\begin{aligned} C_o &= AB \vee C_i A \vee C_i B = AB \vee C_i (A \vee B) = \\ &= AB \vee C_i (\overline{AB} \vee AB \vee \overline{AB} \vee AB) = AB \vee C_i (AB \vee \overline{AB}) \vee C_i (\overline{AB} \vee AB) = \\ &= AB \vee C_i AB \vee C_i (\overline{AB} \vee \overline{AB}) = AB(1 \vee C_i) \vee C_i (\overline{AB} \vee \overline{AB}) = AB \vee C_i (\overline{AB} \vee \overline{AB}) = \\ &= AB \vee C_i (A \oplus B); \end{aligned}$$

$$\begin{aligned} \overline{C_o} &= \overline{AB} \vee \overline{BC_i} \vee \overline{AC_i} = \overline{AB} \vee \overline{C_i} (\overline{B} \vee \overline{A}) = \overline{AB} \vee \overline{C_i} (\overline{AB} \vee \overline{AB} \vee \overline{AB} \vee \overline{AB}) = \\ &= \overline{AB} \vee \overline{C_i} (\overline{AB} \vee \overline{AB}) \vee \overline{C_i} (\overline{AB} \vee \overline{AB}) = \overline{AB} \vee \overline{C_i} \overline{AB} \vee \overline{C_i} (\overline{AB} \vee \overline{AB}) = \\ &= \overline{AB} (1 \vee \overline{C_i}) \vee \overline{C_i} (\overline{AB} \vee \overline{AB}) = \overline{AB} \vee \overline{C_i} (\overline{AB} \vee \overline{AB}) = \overline{AB} \vee \overline{C_i} (A \oplus B); \end{aligned}$$

$$S = \overline{A} \cdot \overline{B} \cdot C_i \vee A \cdot \overline{B} \cdot \overline{C_i} \vee \overline{A} \cdot B \cdot \overline{C_i} \vee A \cdot B \cdot C_i = (A \vee B \vee C_i) \overline{C_o} \vee ABC_i \quad (\text{розряд суми})$$

формується, коли на якомусь вході є одиничка і немає вихідного переносу або коли на всіх входах є одинички).

На підставі цих рівнянь можна побудувати схему повного суматора на елементах І, АБО і НЕ.

Уведемо наступні позначення:  $E = A \oplus B$ ;  $D = AB$ ;  $T = A \vee B$ .

Тоді  $S = E \oplus C_i$ ;  $C_o = D \vee EC_i$ ;  $\overline{C_o} = \overline{T} \vee \overline{EC_i}$ .

Схему повного двійкового однорозрядного суматора, побудованого на елементах І-НЕ, відповідно до останніх рівнянь показує рис. 10.1.

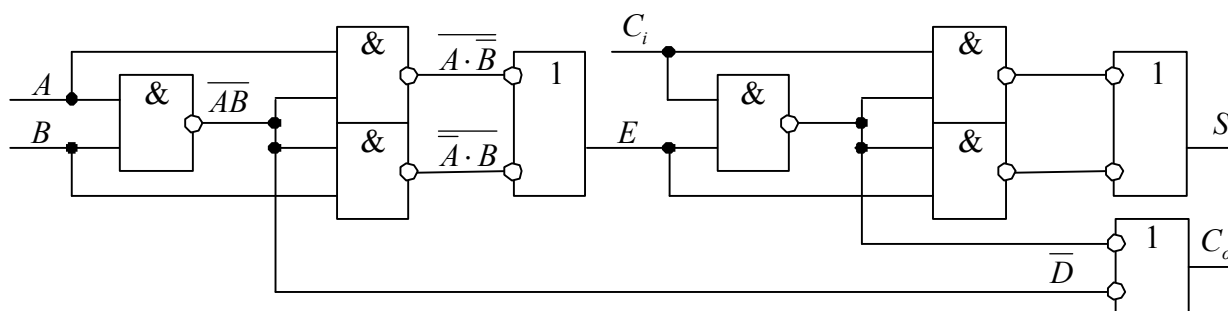


Рис. 10.1. Повний однорозрядний двійковий суматор на елементах І-НЕ

### 10.3 Однорозрядний двійковий напівсуматор

Схему повного суматора можна також сформувати за допомогою двох напівсуматорів (рис. 10.2).

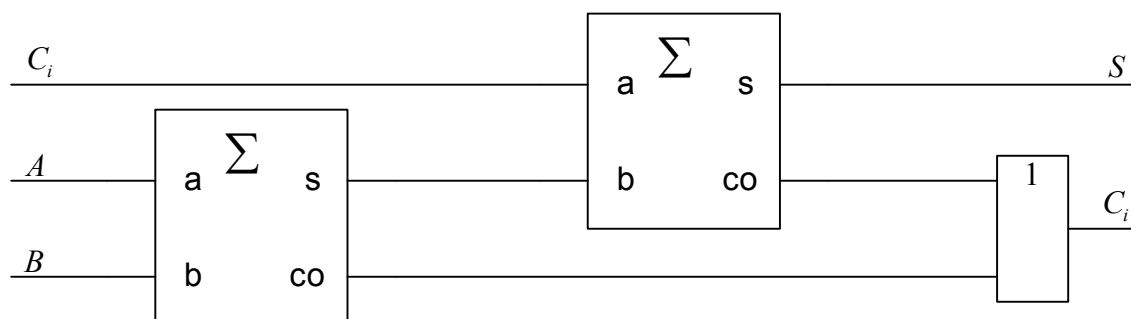


Рис. 10.2. Повний однорозрядний двійковий суматор на напівсуматорах

Напівсуматор не має вхідного переносу і для нього справедливі наступні рівняння:  $c_o = ab$ ;  $s = ab \vee \bar{a}\bar{b} = (a \vee b)\bar{c}_o$ .

рис. 10.3 показує схему напівсуматора в базисі Буля (І, АБО, НЕ).

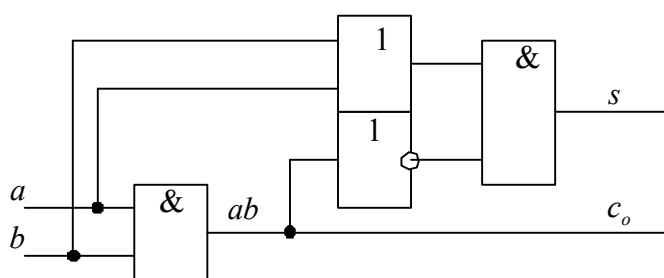


Рис. 10.3. Однорозрядний двійковий напівсуматор

### 10.4 Дешифратор

Дешифратор - це схема, яка має  $n$  інформаційних входів і до  $2^n$  виходів і використовується для перетворення  $n$ -розрядного двійкового коду на вході в унарний код на виході. Унарний код складається з набору нулів і лише однієї одиниці (або



навпаки). В комп'ютерах дешифратори найчастіше використовуються для перетворення  $n$ -розрядного двійкового коду в унарний код на  $2^n$  виходах (кожному  $n$ -розрядному двійковому коду на вході відповідає сигнал 1 тільки на одному з  $2^n$  виходів). Для прикладу наводиться таблиця 10. істинності дешифратора на 3 входи, його внутрішня схема на елементах базису Буля (рис. 10.4, а), його повні умовні графічні (б, в) і стисле (групове) умовне графічне позначення (г). Інформаційні входи дешифратора позначаються на символі (б) вагою розрядів двійкового коду, а на символі (в) – своїми порядковими номерами з першою літерою *I* (Input – вхід). Виходи дешифратора позначаються порядковими номерами (можуть бути з першою літерою *O* (Output – вихід)), які відповідають номеру набору на інформаційних входах. Виходи дешифраторів можуть бути інверсними. Додатковий вхід керування *CS* (Chip Select - вибір кристалу) дозволяє або забороняє формування сигналів на виходах дешифратора.

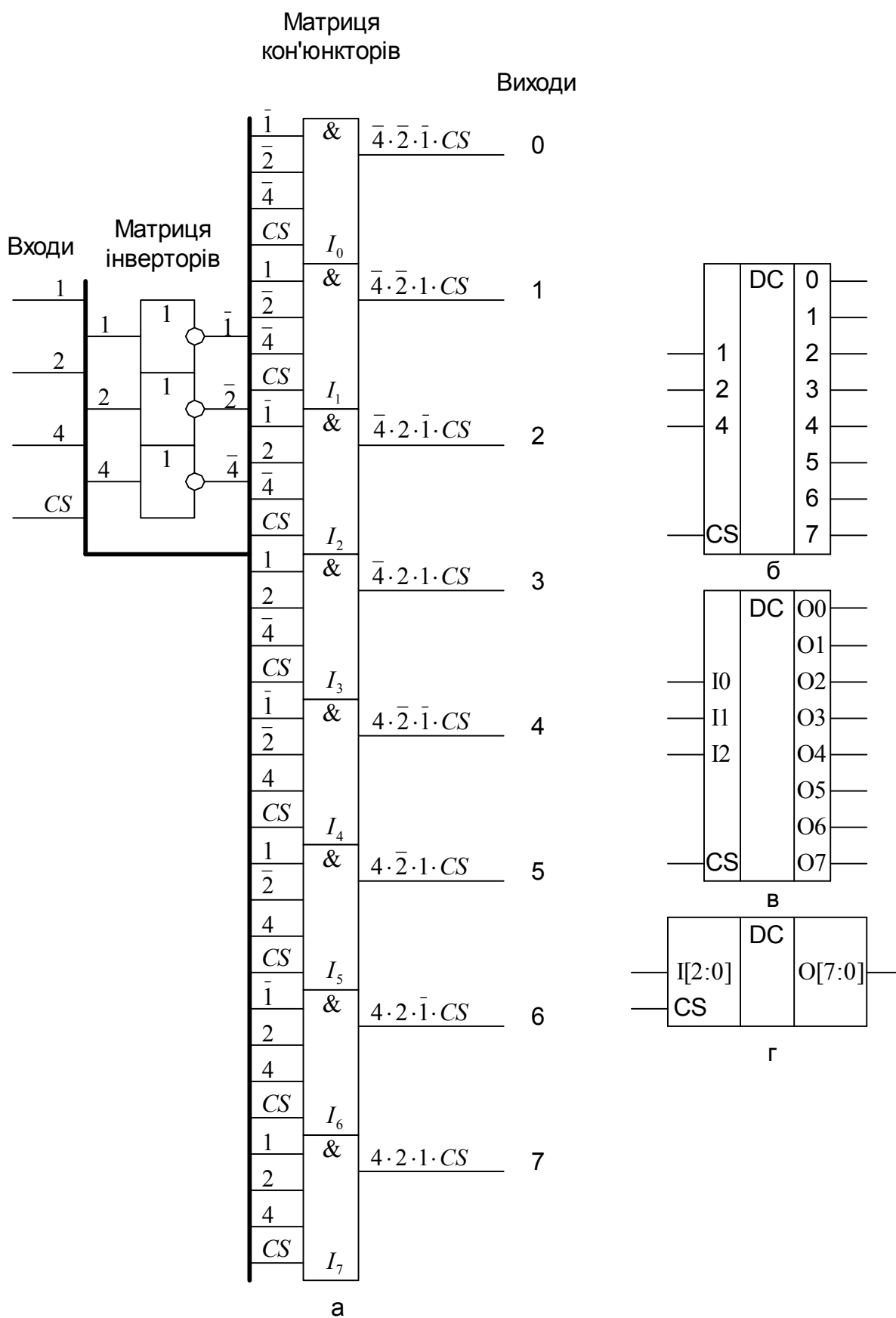


Рис. 10.4. Дешифратор „3 у 8”

**Таблиця 10.2. Дешифратор „3 у 8”**

Номер набору	Входи				Виходи							
	4	2	1	CS	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	1	0
2	0	1	0	1	0	0	0	0	0	1	0	0
3	0	1	1	1	0	0	0	0	1	0	0	0
4	1	0	0	1	0	0	0	1	0	0	0	0
5	1	0	1	1	0	0	1	0	0	0	0	0
6	1	1	0	1	0	1	0	0	0	0	0	0
7	1	1	1	1	1	0	0	0	0	0	0	0
X	x	x	x	0	0	0	0	0	0	0	0	0

Відповідність позначень виводів дешифратора на різних символах показує таблиця 10..

**Таблиця 10.3. Відповідність позначень виводів дешифратора**

Рис. 10.4	Входи			Виходи							
	4	2	1	7	6	5	4	3	2	1	0
Символ (б)	4	2	1	7	6	5	4	3	2	1	0
Символ (в)	<i>I2</i>	<i>I1</i>	<i>I0</i>	<i>O7</i>	<i>O6</i>	<i>O5</i>	<i>O4</i>	<i>O3</i>	<i>O2</i>	<i>O1</i>	<i>O0</i>
Символ (г)	<i>I(2)</i>	<i>I(1)</i>	<i>I(0)</i>	<i>O(7)</i>	<i>O(6)</i>	<i>O(5)</i>	<i>O(4)</i>	<i>O(3)</i>	<i>O(2)</i>	<i>O(1)</i>	<i>O(0)</i>

Спрощенну (матричну) функціональну схема дешифратора „3 у 8” показує рис. 10.5.

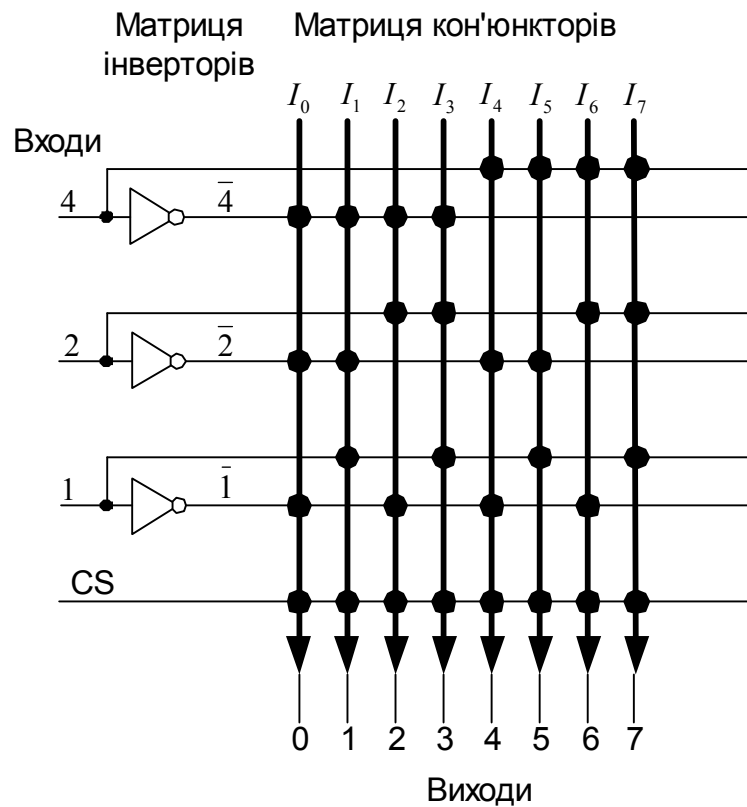


Рис. 10.5. Спрощена функціональна схема дешифратора „3 у 8”

Описані вище дешифратори називаються повними, оскільки кількість їхніх виходів дорівнює  $2^n$ . Дешифратори, які мають меншу кількість виходів, називаються неповними.

Нижче наведений опис на мові VHDL дешифратора, якому відповідає рис. 10.40, г.

---

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity DC is
    port ( O : out STD_LOGIC_VECTOR (7 downto 0);
          I : in  STD_LOGIC_VECTOR (2 downto 0);
          CS : in  STD_LOGIC);
end entity;

architecture DC_arch of DC is
begin
    O(0) <= CS when (I = 0) else '0';
    O(1) <= CS when (I = 1) else '0';

```

```

O(2) <= CS when (I = 2) else '0';
O(3) <= CS when (I = 3) else '0';
O(4) <= CS when (I = 4) else '0';
O(5) <= CS when (I = 5) else '0';
O(6) <= CS when (I = 6) else '0';
O(7) <= CS when (I = 7) else '0';

```

*end architecture;*

На дешифраторах зручно реалізовувати ДДНФ, оскільки ДДНФ безпосередньо вказує на ті з  $2^n$  наборів, на яких функція приймає значення "1". Для схемної реалізації такої функції необхідно на інформаційні входи дешифратора подати вхідні змінні і з'єднати виходи дешифратора, які відповідають номерам набору, на яких функція приймає значення "1", з входами додаткового елемента АБО. На виході елемента АБО буде сформована потрібна функція.

Наприклад, реалізувати функцію, яку задає таблиця 10..

**Таблиця 10.4. Таблиця істинності ФАЛЗ**

Номер набору	<i>a</i>	<i>b</i>	<i>c</i>	<i>f</i>	Елементарні кон'юнкції
0	0	0	0	1	$\bar{a} \cdot \bar{b} \cdot \bar{c}$
1	0	0	1	0	
2	0	1	0	1	$\bar{a} \cdot b \cdot \bar{c}$
3	0	1	1	1	$\bar{a} \cdot b \cdot c$
4	1	0	0	0	
5	1	0	1	0	
6	1	1	0	1	$a \cdot b \cdot \bar{c}$
7	1	1	1	1	$a \cdot b \cdot c$
Вага входу	4	2	1		

Схему реалізації показує рис. 10.60, там же неведено спрощену (матричну) функціональну схему.

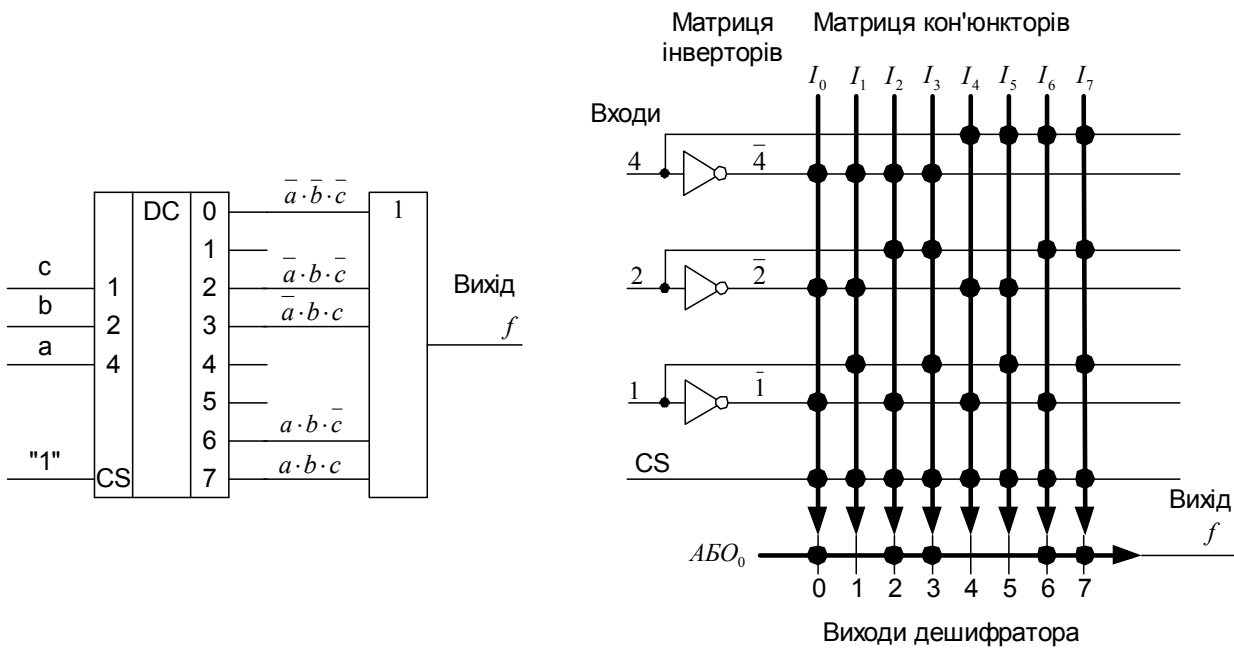


Рис. 10.6. Реалізація ФДЛЗ на дешифраторі „3 у 8”

Розширення за входами схем, які реалізовано на дешифраторах.

Якщо потрібно дешифрувати більше сигналів, ніж є інформаційних входів у дешифратора, то необхідно використовувати вхід керування  $CS$  і заводити на нього результат дешифрації тих розрядів, які безпосередньо не подані на інформаційні входи дешифратора. Приклад дешифрації 4 розрядів на дешифраторах з трьома інформаційними входами демонструє рис. 10.70, там же показано символ відповідного дешифратора „4 у 16”.

Коли сигнал  $a=0$  - працює верхній дешифратор  $D2$ , на його виходах формуються сигнали, які відповідають наборам 0, ..., 7 вхідних змінних  $a$ ,  $b$ ,  $c$ ,  $d$ . Коли  $a=1$  - працює нижній дешифратор  $D3$ , на його виходах формуються сигнали, які відповідають десятковим наборам 8, ..., 15 вхідних змінних  $a$ ,  $b$ ,  $c$ ,  $d$ . Таким чином, вхід  $CS$  у дешифраторах має найбільшу вагу, у наведеному прикладі вага входу  $CS$  у дешифраторів „3 у 8”  $D2$  і  $D3$  дорівнює 8.

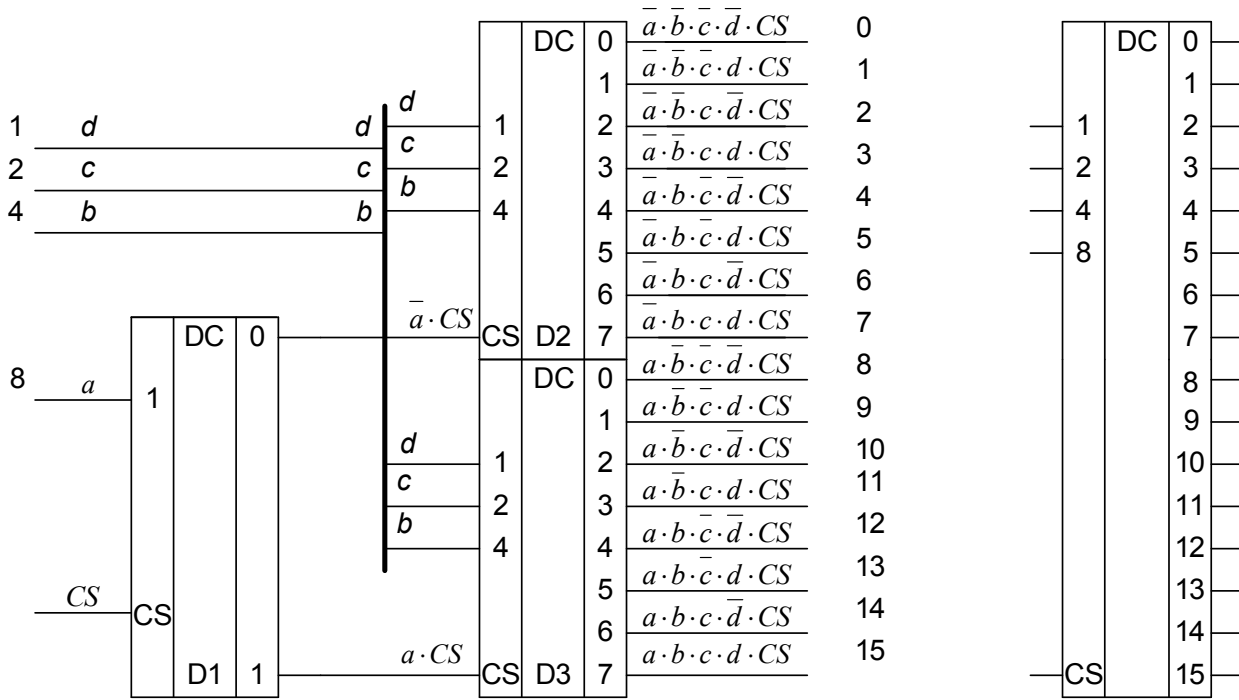


Рис. 10.7. Реалізація дешифратора „4 у 16” на дешифраторах „3 у 8”

### 10.5 Мультиплексори

Мультиплексор - це комбінаційна багатовходова схема з одним виходом  $f$ . Входи мультиплексора поділяються на інформаційні ( $k$  входів) та керування ( $n$  входів). Загалом,  $k \leq 2^n$ . Код, який надходить на входи керування, визначає один з інформаційних входів, сигнал з якого передається на вихід  $f$ . Таблиця 10. істинності пояснює, як працює мультиплексор з трьома входами керування і 8 інформаційними входами, на рис. 10.8 показано умовне графічне позначення такого мультиплексора. Входи керування мультиплексора позначаються вагою розрядів двійкового коду. Інформаційні входи мультиплексора позначаються порядковими номерами, які відповідають номеру набору на вход керування.

Таблиця 10.5. Мультиплексор „8 у 1”

Номер набору	Входи керування			Інформаційний вхід, з якого буде передаватися інформація на вихід
	4	2	1	
0	0	0	0	0
1	0	0	1	1

2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7

Рис. 10.8 показує символи мультиплектора (а, б, в) та його спрощену (матричну) функціональну схему (г).

На мультиплекторах зручно реалізовувати ДДНФ, оскільки ДДНФ безпосередньо вказує на ті з  $2^n$  наборів, на яких функція приймає значення "1". Для схемної реалізації такої функції необхідно подати на входи керування мультиплектора змінні-аргументи у відповідності з їхньою вагою і з'єднати входи мультиплектора, номери яких відповідають наборам, на яких функція приймає значення "1", з логічною "1". На решту входів треба подати логічний "0". На виході мультиплектора буде сформована потрібна функція.



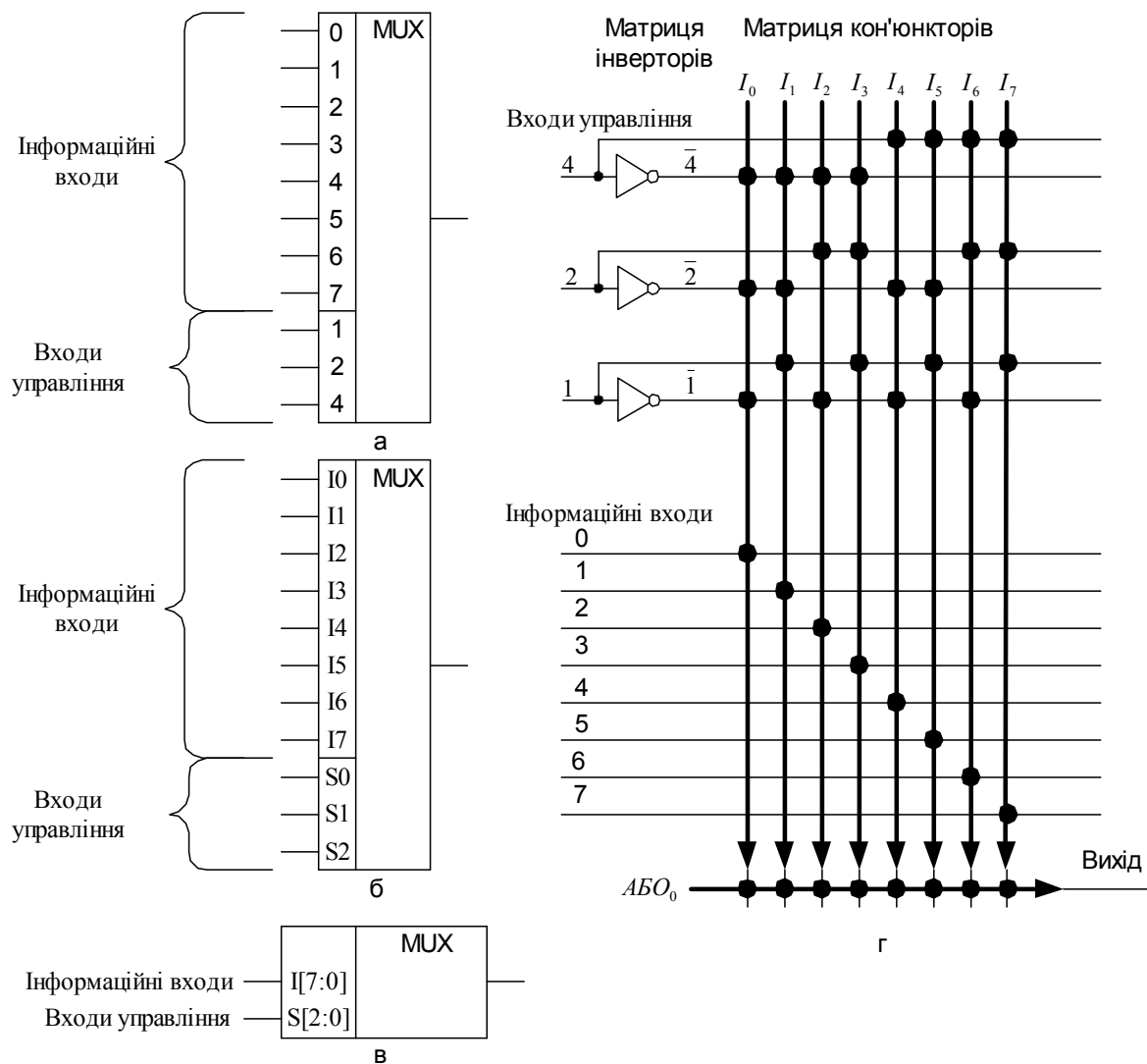


Рис. 10.8. Мультиплексор „8 у 1”

Відповідність позначень виводів мультиплексора на різних символах показує таблиця 10..

Таблиця 10.6. Відповідність позначень виводів мультиплексора

Рис. 10.8	Входи керування			Інформаційні входи							
Символ (а)	4	2	1	7	6	5	4	3	2	1	0
Символ (б)	S2	S1	S0	I7	I6	I5	I4	I3	I2	I1	I0
Символ (в)	S(2)	S(1)	S(0)	I(7)	I(6)	I(5)	I(4)	I(3)	I(2)	I(1)	I(0)

Нижче наведений опис на мові VHDL дешифратора, якому відповідає рис. 10.8, в.

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

```

```

entity mux is
    port ( I : in std_logic_vector (7 downto 0);
          S : in std_logic_vector (2 downto 0);
          O : out std_logic);
end entity;

architecture mux_arch of mux is
begin
    O <= I(CONV_INTEGER(S));
end architecture;

```

На схемі (рис. 10.9) показано приклад реалізації функції алгебри логіки, яку задає таблиця 10..

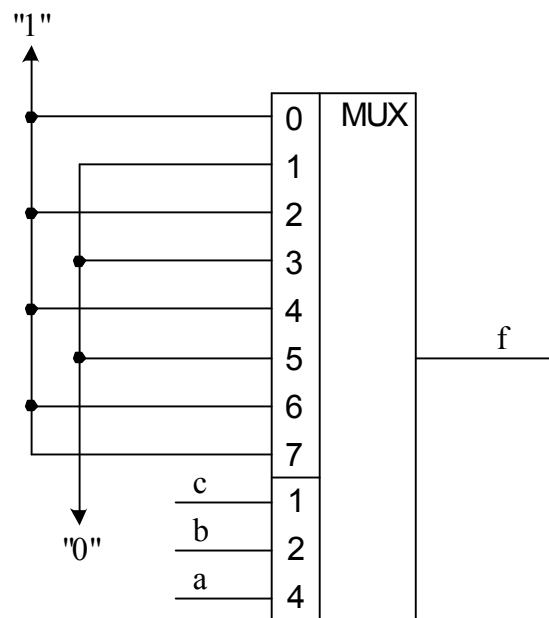


Рис. 10.9. Реалізація ФАЛ на мультиплексорі „8 у 1”

Розширення за входами схем, які реалізовано на мультиплексорах.

Коли кількість змінних перевищує кількість входів керування мультиплексора, необхідно використовувати каскадне з'єднання мультиплексорів.

Приклад мультиплексора на 16 входів, який побудований на базі 8-входового і 2-входового мультиплексорів подає рис. 10.10, де наведено схему мультиплексора та його символ.

Коли на вході керування  $8 \in 0$  на вихід 2-входового мультиплексора  $D3$  проходить сигнал з верхнього мультиплексора  $D1$ , тобто з інформаційних

входів 0, ..., 7 схеми.

Коли на вході керування 8 є 1 на вихід 2-входового мультиплексора  $D3$  проходить сигнал з нижнього мультиплексора  $D2$ , тобто з інформаційних входів 8, ..., 15 схеми.

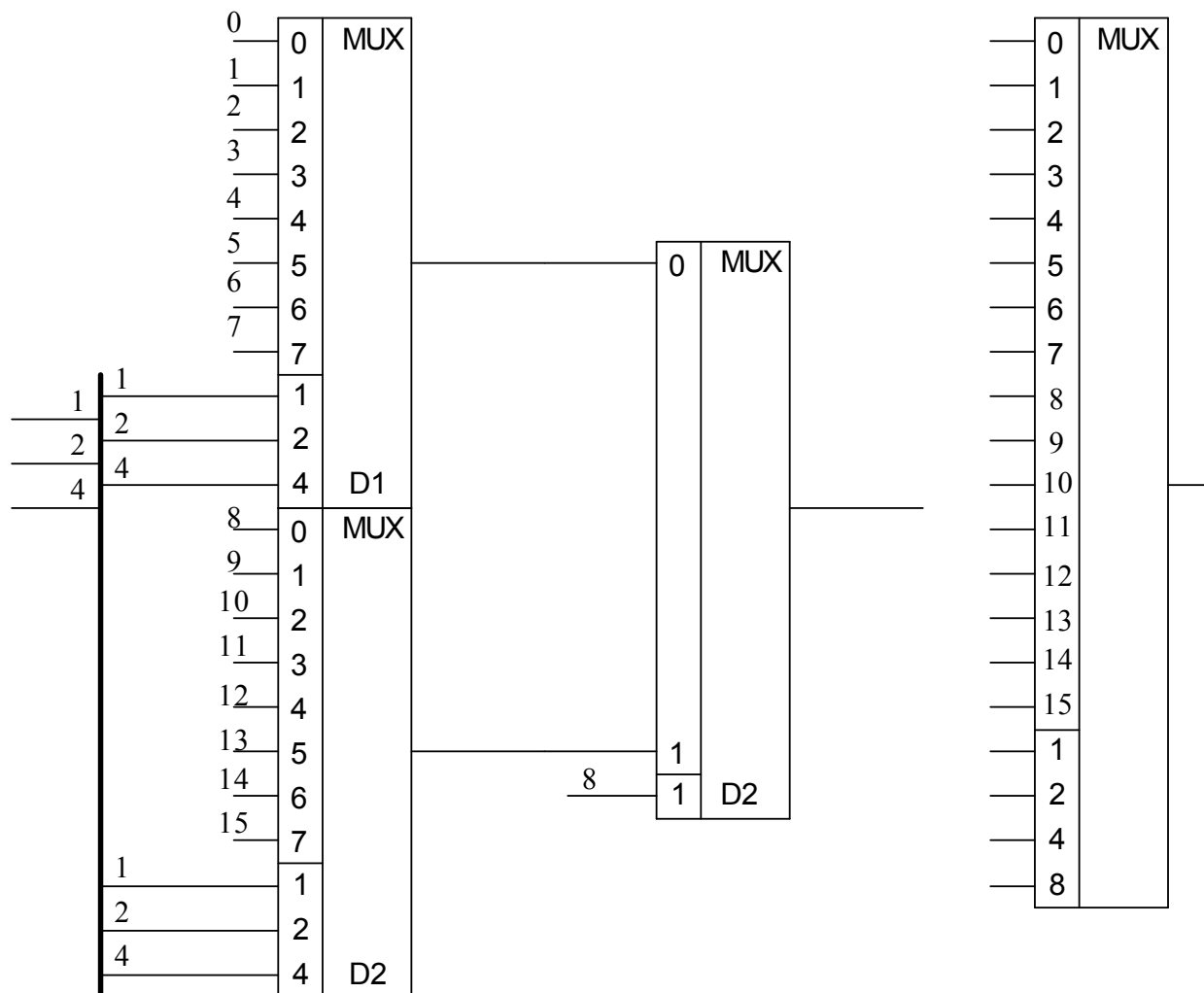


Рис. 10.10. Реалізація мультиплексора „16 у 1”

### 10.6 Постійний запам'ятовуючий пристрій

Постійний запам'ятовуючий пристрій (ПЗП) - це комбінаційна багатовходова схема з одним або кількома виходами  $f_0, \dots, f_{k-1}$ . На входи подаються набори, які називаються адресами і позначаються  $A_0, A_1, \dots, A_{n-1}$ , а з виходів знімаються набори, які називаються даними і позначаються  $D_0, D_1, \dots, D_{k-1}$ . Рис. 10.11 містить символ ПЗП (а, б) і його спрощену (матричну) функціональну схему (в).

Кожній адресі відповідають свої дані, які записані в ПЗП або в процесі виготовлення, або користувачем перед встановленням на плату (комірку) чи вже на

самій платі. Для занесення інформації в ПЗП необхідно скласти таблицю істинності ПЗП - таблицю прошиття, яка встановлює відповідність між адресами і даними. Занесення інформації до ПЗП здійснюється користувачем за допомогою пристрою, який називається програматором.

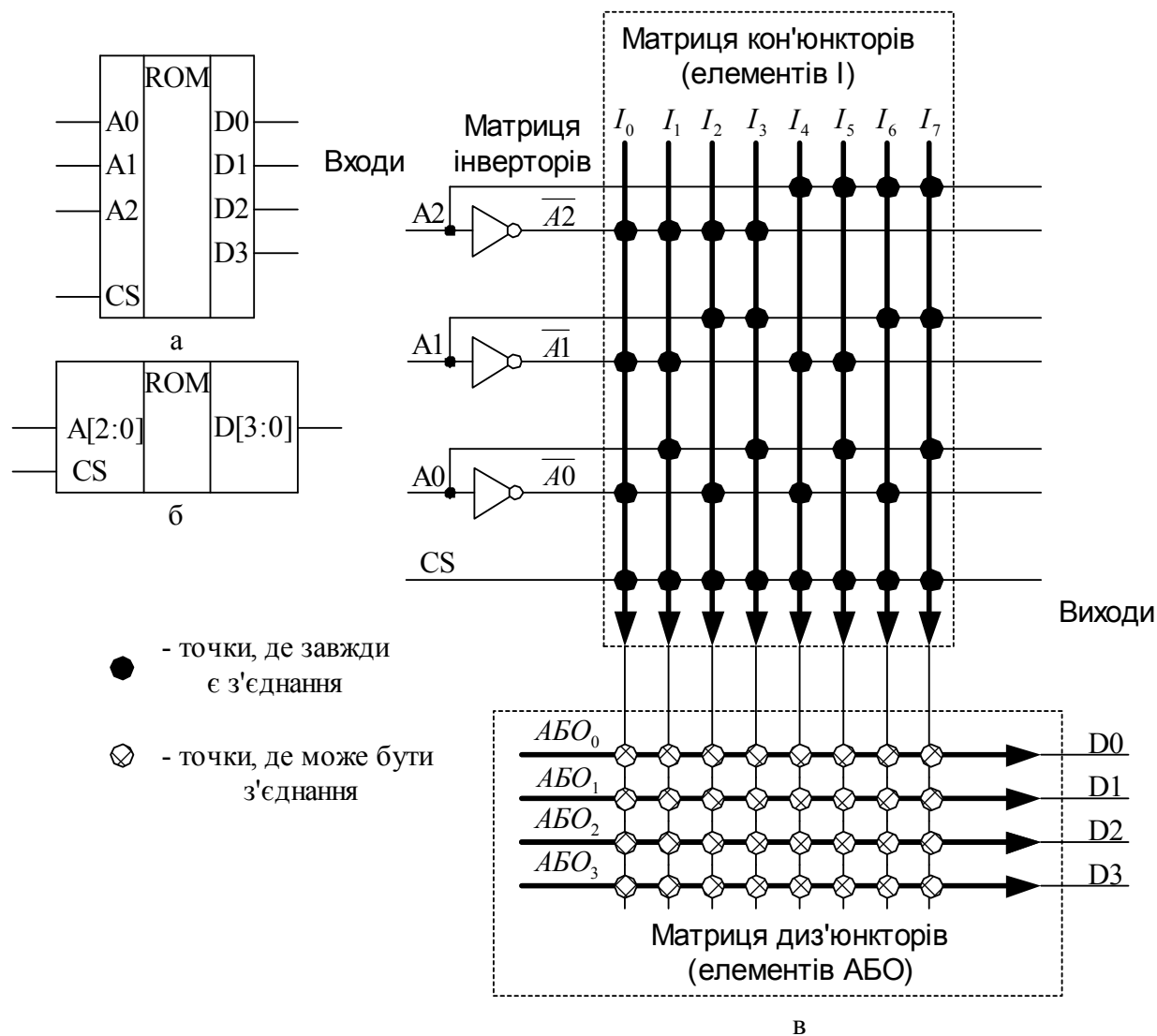


Рис. 10.11. Незапрограмований ПЗП

Як видно, ПЗП складається з двох матриць:

вже запрограмованої матриці елементів I;

матриці елементів АБО, яку користувач може запрограмувати, зробивши відповідні є'днання виходів елементів I з входами елементів АБО.

Якщо кількість розрядів адреси дорівнює  $n$ , то в ПЗП зберігається  $2^n$  наборів даних (слів) розрядністю  $k$ , де  $k$  - кількість виходів ПЗП, тобто, об'єм ПЗП дорівнює  $2^n \cdot k$  біт.

На ПЗП зручно реалізовувати ДДНФ набору функцій, оскільки ДДНФ безпосередньо вказує на ті набори з  $2^n$ , на яких функція приймає значення "1". Для реалізації таких функцій необхідно завести на входи ПЗП усі змінні, з яких формуються функції, кожній з функцій поставити у відповідність один з виходів ПЗП і скласти таблицю прошиття.

Рис. 10.12 ілюструє приклад реалізації за допомогою ПЗП об'ємом  $256 \times 4$  трьох функцій алгебри логіки, які задає таблиця 10.. На цьому ж рисунку показана спрощена (матрична) функціональна схема запрограмованого ПЗП, при цьому взято до уваги лише 8 з 256 елементів матриці I.

Кількість входів адреси ПЗП дорівнює 8 (оскільки  $256 = 2^8$ ). Кількість виходів ПЗП - 4.

Розряди адрес  $A3...A7$  і даних  $D3$  - не використовуються. Таблиця 10. також містить і таблицю прошиття ПЗП.

У даному випадку використовується тільки 8 (0...7) з 256 можливих адрес (3 розряди адресних входів з 7) і тільки 3 розряди даних з 4.

**Таблиця 10.7. Реалізація ФАЛ на ПЗП**

Сигнали	$c$	$b$	$a$	$f2$	$f1$	$f0$	Таблиця прошиття	
Виводи ПЗП	$A2$	$A1$	$A0$	$D2$	$D1$	$D0$		
Вага входу	4	2	1					
Номер набору:	Адреса			Дані			Адреса	Дані
0	0	0	0	1	0	0	0	4
1	0	0	1	0	1	0	1	2
2	0	1	0	1	1	1	2	7
3	0	1	1	1	0	0	3	4
4	1	0	0	0	0	1	4	1
5	1	0	1	0	1	1	5	3
6	1	1	0	1	0	1	6	5
7	1	1	1	1	0	1	7	5
							8...FF	0

Нижче наведено опис на мові VHDL ПЗП, якому відповідає рис. 10.12.

```
-----
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
```

*entity rom is*

```
port ( CS : in STD_LOGIC;  
      A : in STD_LOGIC_VECTOR(2 downto 0);  
      D : out STD_LOGIC_VECTOR(3 downto 0));
```

*end entity;*

*architecture rom\_arch of rom is*

*begin*

```
process(A, CS)
```

```
begin
```

```
  if (CS = '1') then
```

```
    case (A) is
```

```
      when "000" => D <= "0100";
```

```
      when "001" => D <= "0010";
```

```
      when "010" => D <= "0111";
```

```
      when "011" => D <= "0100";
```

```
      when "100" => D <= "0001";
```

```
      when "101" => D <= "0011";
```

```
      when "110" => D <= "0101";
```

```
      when "111" => D <= "0101";
```

```
      when others => D <= "0000";
```

```
    end case;
```

```
  else
```

```
    D <= "0000";
```

```
  end if;
```

```
end process;
```

*end architecture;*

---

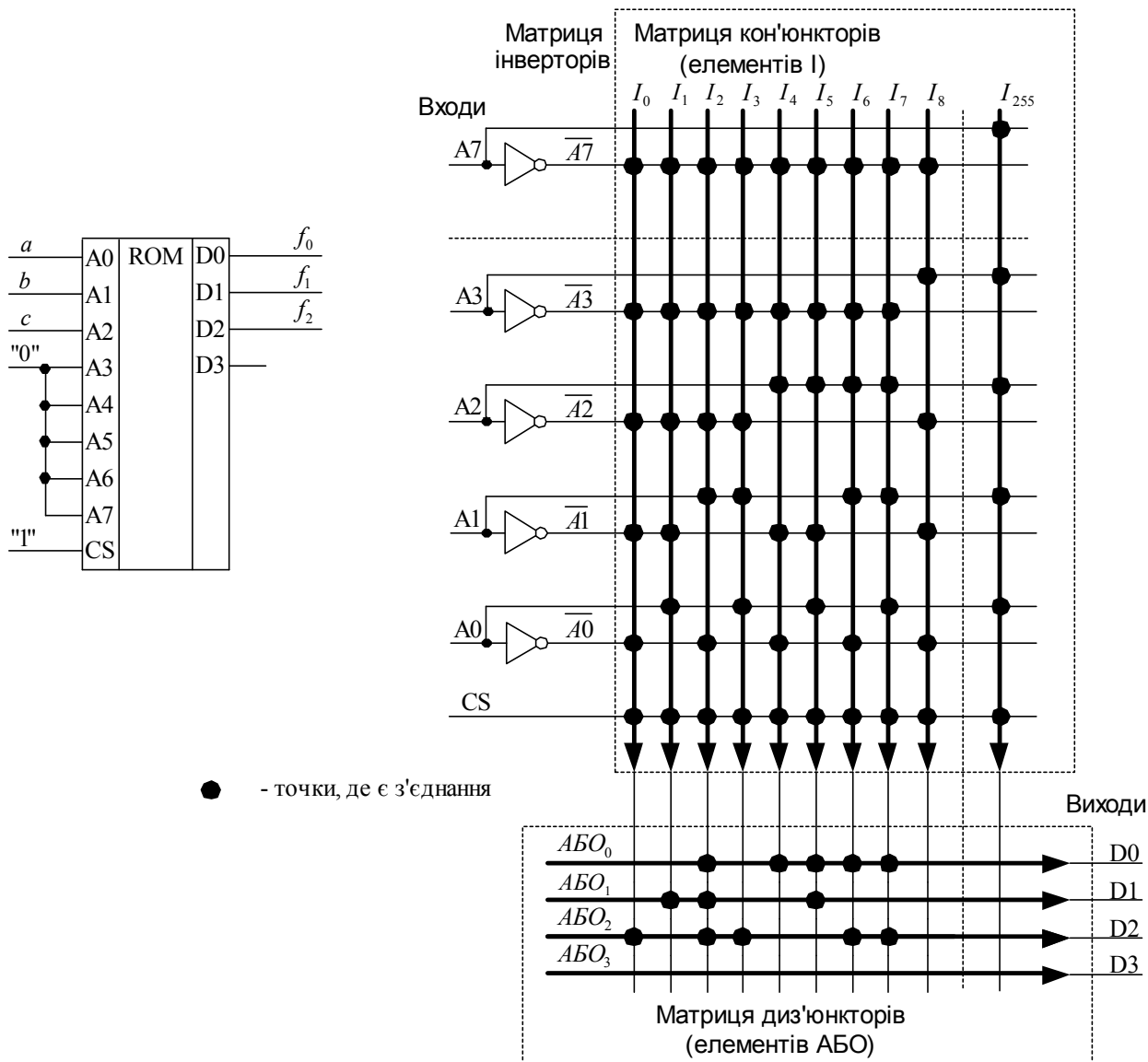


Рис. 10.12. Реалізація ФАЛ на ПЗП (запрограмований ПЗП)

Розширення за входами схем, реалізованих на ПЗП, здійснюється, коли кількість вхідних змінних перевищує кількість адресних входів ПЗП. Розширення виконується з використанням входів вибору кристалу ( $CS$ ) так само, як розширення за входами дешифраторів. Але, додатково, здійснюється об'єднання одноіменних виходів ПЗП за допомогою елементів АБО (для ПЗП, в яких вхід  $CS$  переводить виходи в так званий „третій”, високоімпедансний стан, замість елементів АБО використовується монтажне АБО).

## 10.7 Програмовані логічні матриці

Програмована логічна матриця (ПЛМ, PLA - Programmable Logic Array) - це комбінаційна багатовходова схема з одним або кількома виходами. На входи

подаються набори, які називаються адресами, а з виходів знімаються набори, які називаються даними. Кожній адресі відповідають свої дані, які записані в ПЛІМ або в процесі виготовлення, або користувачем перед встановленням на плату (комірку) чи вже на самій платі. Для занесення інформації в ПЛІМ необхідно скласти таблицю прошиття, яка встановлює відповідність між адресами і даними. Занесення користувачем інформації в ПЛІМ здійснюється за допомогою пристрою, який називається програматором.

На відміну від ПЗП, у ПЛІМ зберігається тільки частина з  $2^n$  наборів даних (слів) розрядністю  $k$ , де  $n$  - кількість адресних входів,  $k$  - кількість виходів ПЛІМ. Яка саме частина - визначається з таблиці прошиття.

Рис. 10.13 показує символи (а, б) та структуру (в) ПЛІМ типу PLA (Programmable Logic Array).

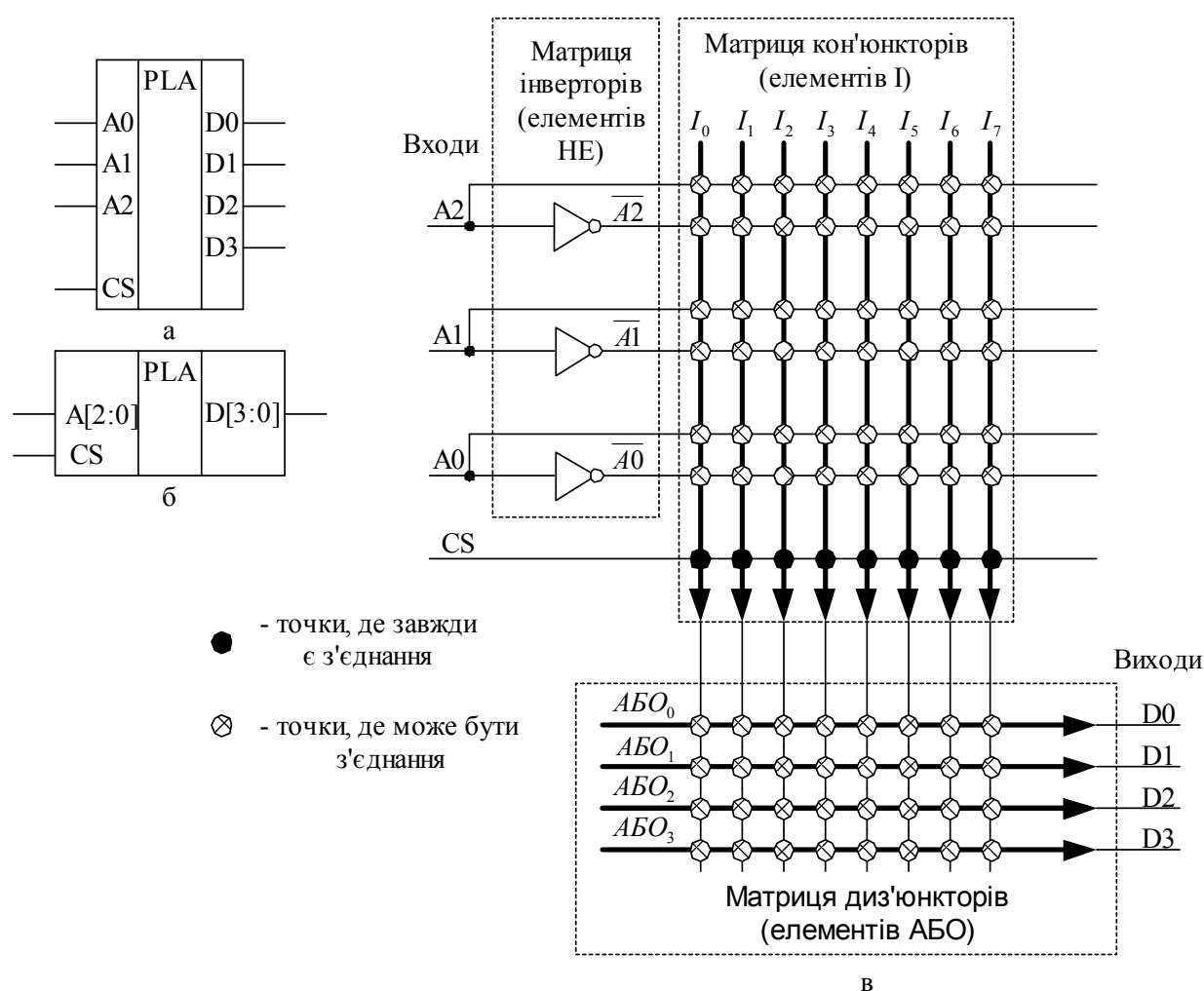


Рис. 10.13. Програмована логічна матриця (незапрограмована)

ПЛІМ складається з трьох основних частин:



набору (матриці) інверторів вхідних сигналів;

набору (матриці) програмованих елементів І (рис. 10.13 умовно показує 8 елементів  $I_0...I_7$  вертикальними стрілками, місця перетину стрілок і ліній вхідних сигналів є входами елементів І, продовженням стрілки є зв'язок, на якому формується вихідний сигнал даного елемента І);

набору (матриці) програмованих елементів АБО (рис. 10.13 умовно показує 4 елементи АБО0...АБО3 горизонтальними стрілками, місця перетину стрілок і ліній вхідних сигналів елементів І є входами елементів АБО, продовженням стрілки є зв'язок, на якому формується вихідний сигнал даного елемента АБО).

Кількість елементів у матриці І залежить від типу ПЛМ (може бути 48, 64 і т.д.). На кожний елемент І заведені сигнали з кожного адресного входу в прямому і інверсному коді, тобто входів у елемента І вдвічі більше, ніж входів у ПЛМ.

Кількість елементів у матриці АБО залежить від кількості виходів ПЛМ (може бути 8, 14, 16 і т.д.). У ПЛМ типу PLA на кожний елемент АБО заведені виходи всіх елементів І. ПЛМ (рис. 10.13) має 3 адресних входи, 8 елементів І, 4 елементи АБО, 4 виходи.

Під час програмування ПЛМ користувач має можливість забирати зайві сигнали з входів елементів І та АБО.

На ПЛМ зручно реалізовувати мінімізовані ДНФ набору функцій. Для їхньої реалізації необхідно завести на входи ПЛМ всі змінні, з яких формуються функції, кожній з функцій поставити у відповідність один з виходів ПЛМ і скласти таблицю прощиття.

Рис. 10.14 демонструє приклад реалізації на ПЛМ (рис. 10.13) функцій

$$f_0 = b \cdot c \vee \bar{a} \cdot \bar{c} \vee a \cdot \bar{b};$$

$$f_1 = a \cdot \bar{b} \vee \bar{a} \cdot b \cdot \bar{c};$$

$$f_2 = c \vee \bar{a} \cdot b \vee a \cdot \bar{b}.$$

Які саме елементи І та АБО будуть використовуватися визначає користувач. Під'єднання незадіяних елементів не показані.

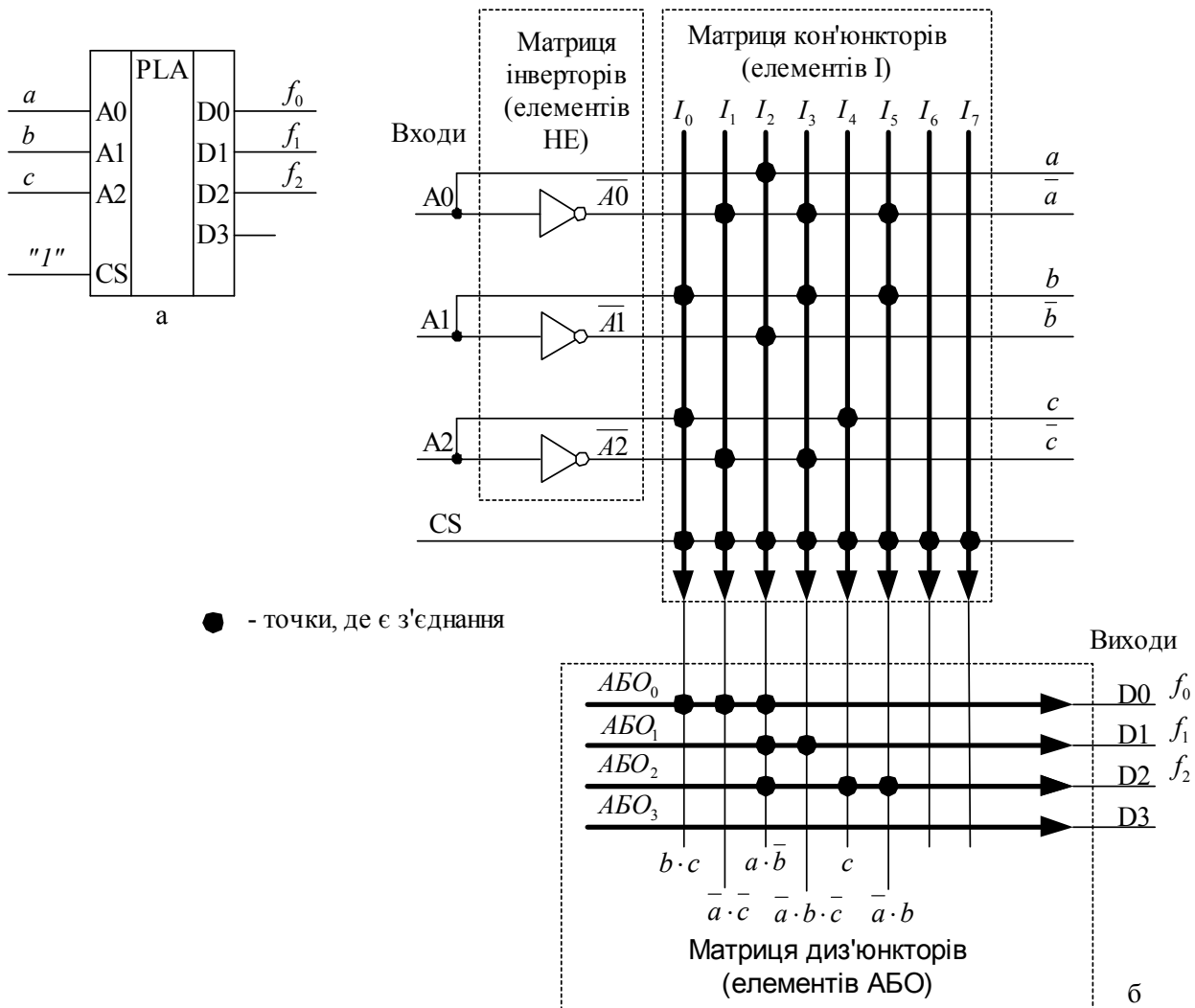


Рис. 10.14. Програмована логічна матриця (запрограмована)

Терм  $b \cdot c$  реалізовано на елементі  $I_0$ ;

терм  $\overline{a} \cdot \overline{c}$  реалізовано на елементі  $I_1$ ;

терм  $a \cdot \overline{b}$  реалізовано на елементі  $I_2$  один раз для всіх функцій;

терм  $\overline{a} \cdot b \cdot \overline{c}$  реалізовано на елементі  $I_3$ ;

терм  $c$  реалізовано на елементі  $I_4$ ;

терм  $\overline{a} \cdot b$  реалізовано на елементі  $I_5$ ;

функцію  $f_0$  реалізовано на елементі АБО0 (формується на виході  $D0$ );

функцію  $f_1$  реалізовано на елементі АБО1 (формується на виході  $D1$ );

функцію  $f_2$  реалізовано на елементі АБО2 (формується на виході  $D2$ ).

Розряд даних  $D3$  - не використовуються. Таблиця прошиття ПЛМ виглядає як таблиця 10.. У даному випадку використовуються 6 ( $I_0, \dots, I_5$ ) елементів І (кон'юнкторів) з 8, 3 розряди адресних входів з 4 і 3 виходи даних з 4.

**Таблиця 10.8. Таблиця прошиття ПЛМ**

Сигнали	<i>c</i>	<i>b</i>	<i>a</i>	<i>f2</i>	<i>f1</i>	<i>f0</i>
Виводи ПЗП	<i>A2</i>	<i>A1</i>	<i>A0</i>	<i>D2</i>	<i>D1</i>	<i>D0</i>
Елемент матриці I	Адреса (вхід)			Дані		
<i>I0</i>	H	H	-	-	-	A
<i>I1</i>	L	-	L	-	-	A
<i>I2</i>	-	L	H	A	A	A
<i>I3</i>	L	H	L	-	A	-
<i>I4</i>	H	-	-	A	-	-
<i>I5</i>	-	H	L	A	-	-
<i>I6</i>	0	0	0	-	-	-
<i>I7</i>	0	0	0	-	-	-

Таблиця 10. містить такі умовні позначення:

на перетині рядка з позначенням елемента I та графи з номером розряду адреси можуть стояти такі символи:

"H" - сигнал з даного входу заведений на відповідний елемент I в прямому коді;

"L" - сигнал з даного входу заведений на відповідний елемент I в інверсному коді;

"-" - сигнал з даного входу від'єднаний від відповідного елемента I;

"0" - сигнал з даного входу заведений на відповідний елемент I одночасно в прямому та інверсному кодах (це початковий стан ПЛМ, даний елемент I не використовується).

На перетині рядка з номером елемента I та графи з номером розряду даних (тобто, номером елемента АБО) можуть стояти такі символи:

"A" - сигнал з даного елемента I заведений на вхід відповідного елемента АБО і проходить через елемент АБО на відповідний вихід ПЛМ (це початковий стан незапрограмованої ПЛМ);

"-" - сигнал з даного елемента I від'єднаний від входу відповідного елемента АБО і не проходить на відповідний вихід ПЛМ.

Літери А, які стоять в одній графі, вказують на елементи I, які об'єднуються за АБО.

Розширення за входами схем, які реалізовано на ПЛМ, здійснюється, коли

кількість вхідних змінних перевищує кількість адресних входів ПЛМ.

Розширення виконується так само, як розширення за входами ПЗП з використанням сигналу "Вибір кристалу" (ВК) ПЛМ. ПЛМ у цьому випадку повинна мати цей додатковий вхід ВК.

### **10.8 Програмовані матриці логіки**

На відміну від програмованих логічних матриць (ПЛМ, PLA) у програмованих матриць логіки (ПМЛ, PAL - Programmable Array Logic) на кожний елемент АБО заведено виходи не всіх елементів І. Найчастіше елементи І розділені нарівно між елементами АБО, входи елементів АБО запрограмовані в процесі виробництва. Рис. 10.15 показує символи (а, б) і структуру (в) ПМЛ типу PAL.

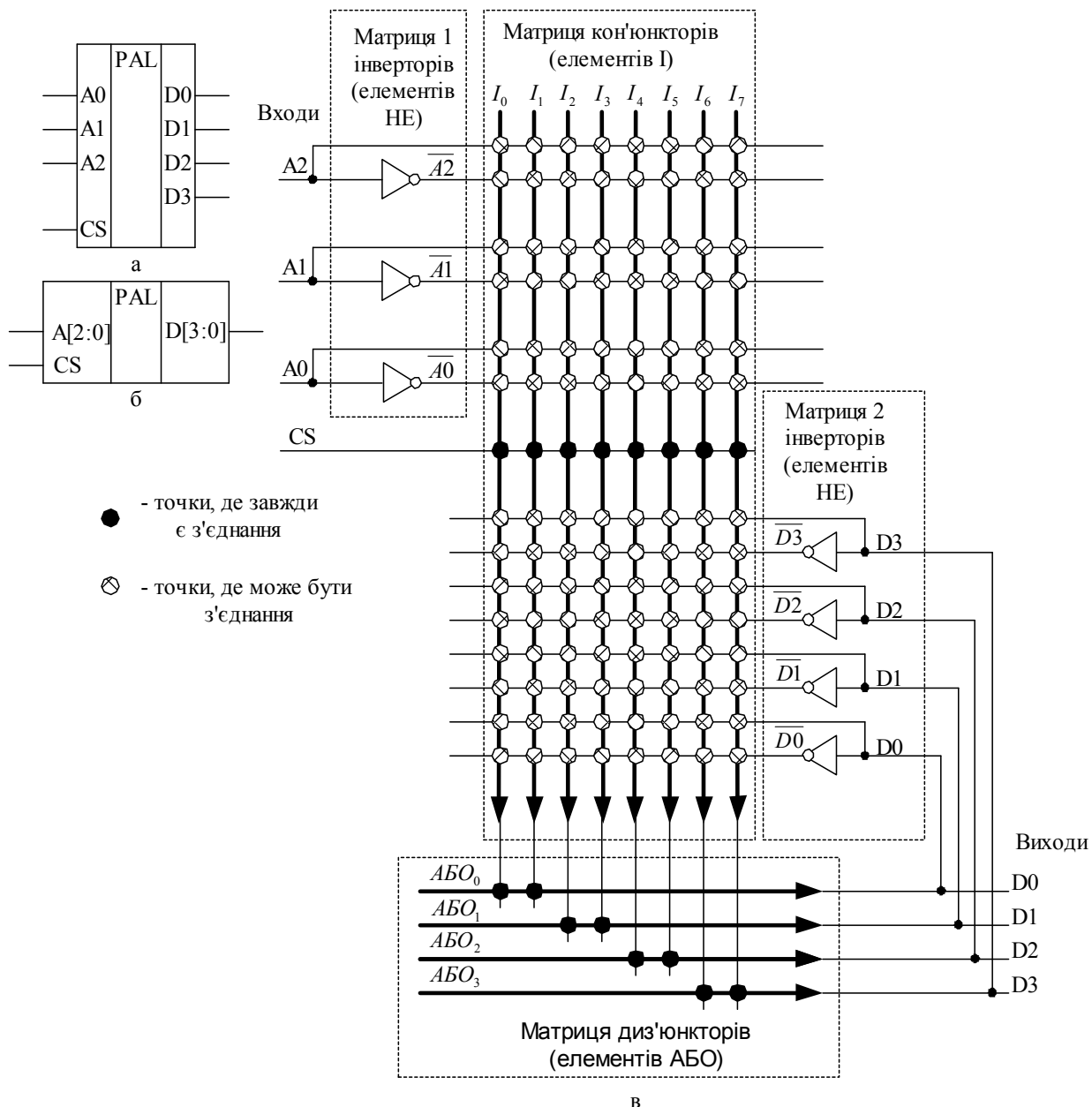


Рис. 10.15. Програмована матриця логіки (незапрограмована)

ПМЛ має 3 адресних входи ( $A0, \dots, A2$ ), 8 елементів І на 14 входів кожний, 4 елементи АБО на 2 входи кожний, 8 зворотних зв'язків ( $D0, \dots, D3$ ). Як видно, програмується тільки матриця елементів І.

ПМЛ складається з чотирьох основних частин:

набору (матриці 1) інверторів вхідних сигналів;

набору (матриці) програмованих елементів І (рис. 10.15 умовно показує 8 елементів  $I_0, \dots, I_7$  вертикальними стрілками, місця перетину стрілок і ліній вхідних сигналів є входами елементів І, продовженням стрілки є зв'язок, на якому формується вихідний сигнал даного елемента І);

набору (матриці) непрограмованих елементів АБО (рис. 10.15 умовно показує 4 елементи АБО0...АБО3 горизонтальними стрілками, місця перетину стрілок і ліній вихідних сигналів елементів І є входами елементів АБО, продовженням стрілки є зв'язок, на якому формується вихідний сигнал даного елемента АБО);

набору (матриці 2) інверторів зворотніх зв'язків.

Кількість елементів у матриці І залежить від типу ПЛМ(може бути 48, 64 і т.д.). На кожний елемент І заведені сигнали з кожного адресного входу в прямому і інверсному коді, а також зворотні зв'язки з виходів матриці. Тобто, входів у елемента І приблизно вдвічі більше, ніж разом входів і виходів у ПЛМ.

Кількість елементів у матриці АБО залежить від кількості виходів ПЛМ (може бути 8, 14, 16 і т.д.). У ПМЛ типу PAL на кожний елемент АБО заведені виходи тільки деяких елементів І. ПМЛ (0) має 3 адресних входи, 8 елементів І, 4 елементи АБО, 4 виходи. Елементи І розподілені по 2 на кожний елемент АБО.

Під час програмування ПЛМ користувач має можливість забирати зайві сигнали тільки з входів елементів І.

На ПМЛ зручно реалізовувати мінімізовані за "1" ДНФ набору функцій. Для їхньої реалізації необхідно завести на входи ПЛМ усі змінні, з яких формуються функції, кожній з функцій поставити у відповідність один з виходів ПЛМ і скласти таблицю прощиття.

Рис. 10.15 демонструє приклад реалізації на ПЛМ функцій

$$f_0 = b \cdot c \vee \bar{a} \cdot \bar{c} \vee a \cdot \bar{b};$$

$$f_1 = c \vee \bar{a} \cdot b \vee a \cdot \bar{b}.$$

Рис. 10.16 показує символ (а) і структуру (б) запрограмованої ПМЛ типу PAL.

Які саме елементи І та АБО будуть використовуватися визначає користувач.

Під'єднання незадіяних елементів не показані.

Таблиця прощиття виглядає як таблиця 10..

Терм  $b \cdot c$  реалізований на елементі І0;

терм  $\bar{a} \cdot \bar{c}$  реалізований на елементі І1;

терм  $a \cdot \bar{b}$  реалізований два рази - на елементі І3 для функції  $f_0$  та на елементі І7

для функції  $f_1$ ;

терм  $c$  реалізований на елементі  $I4$ ;

терм  $\bar{a} \cdot b$  реалізований на елементі  $I5$ ;

функцію  $f_0$  послідовно реалізовано на елементах АБО0 і АБО1 (формується на виході  $D1$ );

функцію  $f_1$  послідовно реалізовано на елементах АБО2 і АБО3 (формується на виході  $D3$ ).

У даному випадку використовується усі елементи  $I0, \dots, I7$ , усі елементи АБО0...АБО3, 3 розряди адресних входів  $A0, \dots, A3$ , 4 розряди вихідних даних  $D0, \dots, D3$  і два зворотніх зв'язки:

$D0$  – використовується при формуванні функції  $f_0$ ;

$D3$  – використовується при формуванні функції  $f_1$ .

Таблиця 10. прошиття ПМЛ має такі позначення:

на перетині рядка з позначенням елемента  $I$  та граф з номером входу і номеру зворотнього зв'язку можуть стояти такі символи:

"H" - сигнал з даного входу заведений на відповідний елемент  $I$  в прямому коді;

"L" - сигнал з даного входу заведений на відповідний елемент  $I$  в інверсному коді;

"-" - сигнал з даного входу від'єднаний від відповідного елемента  $I$ ;

"0" - сигнал з даного входу заведений на відповідний елемент  $I$  одночасно в прямому та інверсному кодах (початковий стан ПЛМ, даний елемент  $I$  не використовується).

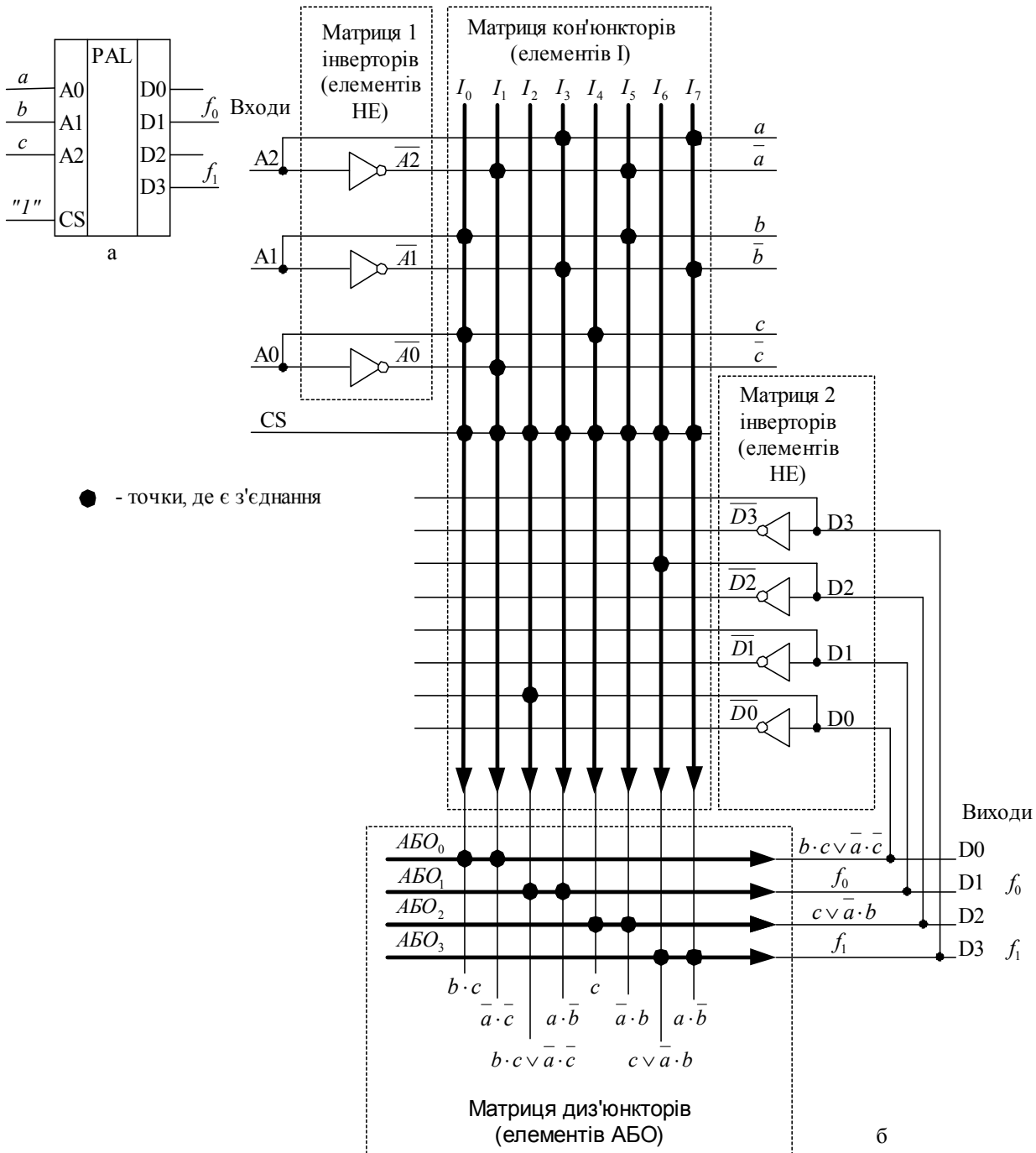


Рис. 10.16. Програмована матриця логіки (запрограмована)

Розширення за входами схем, які реалізовано на ПМЛ, здійснюється, коли кількість вхідних змінних перевищує кількість адресних входів ПЛМ. Розширення виконується так само, як розширення за входами ПЗП з використанням сигналу "Вибір кристалу" (ВК) ПМЛ. ПМЛ у цьому випадку повинна мати додатковий вхід ВК.

Таблиця 10.9. Таблиця прошивки ПМЛ

Сигнали	c	b	a	f1	f0		
Виводи ПЗП	A2	A1	A0	D3	D2	D1 D0	Примітки



Елемент матриці АБО	Дані (вихід)	Елемент матриці I	Адреса (вхід)			Дані (зворотний зв'язок)				
			H	L	-	-	-	-	-	-
АБО0	$D0$	$I0$	H	H	-	-	-	-	-	$b \cdot c$
		$I1$	L	-	L	-	-	-	-	$\bar{a} \cdot \bar{c}$
АБО1	$D1$	$I2$	-	-	-	-	-	-	H	$D0$
		$I3$	-	L	H	-	-	-	-	$a \cdot \bar{b}$
АБО2	$D2$	$I4$	H	-	-	-	-	-	-	$c$
		$I5$	-	H	L	-	-	-	-	$\bar{a} \cdot b$
АБО3	$D3$	$I6$	-	-	-	-	H	-	-	$D1$
		$I7$	-	L	H	-	-	-	-	$a \cdot \bar{b}$





	числення	
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,1 до 10-кової системи числення	0
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,2 до 10-кової системи числення	1
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,3 до 10-кової системи числення	1
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,4 до 10-кової системи числення	2
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,5 до 10-кової системи числення	3
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,6 до 10-кової системи числення	3
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,7 до 10-кової системи числення	4
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,8 до 10-кової системи числення	5
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення шістнадцяткового числа 0,9 до 10-кової системи числення	5
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на парність, для повідомлення, яке складається з 8 молодших біт шістнадцяткового числа 38	1
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на парність, для повідомлення, яке складається з 8 молодших біт шістнадцяткового числа 08	1
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на парність, для повідомлення, яке складається з 8 молодших біт шістнадцяткового числа 68	1
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на парність, для повідомлення, яке складається з 8 молодших біт шістнадцяткового числа 90	0

































	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа 83	0
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа C8	0
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа 56	1
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа 5C	1
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа 15	0
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа 26	0
	Визначити і підкреслити, яким буде контрольний розряд при здійсненні контролю на непарність, для повідомлення, яке складається з 8 молодших біт шістнадцятькового числа E4	1
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 88	7
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 50	6
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 200	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 216	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 185	8























	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 110	7
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 123	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 177	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 35	6
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 185	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 215	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 125	8
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 59	7
	Визначити і підкреслити, скільки (мінімальна кількість у 16-ковій системі числення) контрольних розрядів повинен містити код Хеммінга, який виправляє 1 помилку, якщо кількість інформаційних біт у повідомленні дорівнює 4	2
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 098 та 026	6
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 003 та	4

125		
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1EE та 02E		3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 050 та 0A9		6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 11F та 1DE		3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1D4 та 1DC		1
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1AD та 155		5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1B8 та 1C6		6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 140 та 1C9		3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1CC та 1B1		6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 078 та 075		3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 173 та 17C		4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 145 та 178		5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 15F та 17C		3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 068 та 16F		4

Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0B8 та 19F	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 184 та 159	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 09B та 1A4	7
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 013 та 0EA	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1CC та 128	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0D1 та 1ED	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 148 та 090	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 068 та 0C7	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0C7 та 04B	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1FB та 1EF	2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 143 та 14B	1
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 184 та 1FE	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 06F та 1D0	8
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 053 та	4



1F2		
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1F1 та 016		7
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 009 та 088		2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0F9 та 04E		6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 151 та 159		1
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1B8 та 1BD		2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 03E та 09E		2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 061 та 032		4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1AD та 12A		4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 11F та 096		4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1AE та 04E		4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1E0 та 117		7
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 15D та 0FB		5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1E3 та 16F		3

Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1E4 та 174	2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 141 та 1B9	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 01F та 0EF	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0DA та 15D	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 152 та 134	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0D9 та 180	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 035 та 172	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 17A та 04D	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 12B та 17E	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 01F та 089	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 051 та 15D	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1EA та 1B8	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1D2 та 1DA	1
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1F6 та	4

0B5	
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1C8 та 099	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 036 та 162	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 006 та 0DF	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 064 та 1E9	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 15D та 119	2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0B1 та 097	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0EA та 031	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1BE та 1F2	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 131 та 151	2
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 151 та 174	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0C5 та 08C	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 19F та 070	8
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1A2 та 16C	5

Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 007 та 09E	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1CB та 1C0	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0AF та 037	3
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 183 та 084	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 01A та 0C2	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 12D та 1E7	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1A6 та 035	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 024 та 0CA	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1CF та 131	7
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 125 та 033	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 076 та 0B5	4
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0EB та 1D5	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 07A та 18A	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 13A та	6

	0B5	
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 05A та 172	3
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 1AE та 137	4
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 04B та 1C8	4
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 19D та 108	4
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0A6 та 1A0	3
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0BA та 02E	3
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0DD та 1EE	5
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 10A та 05F	5
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 156 та 0B0	6
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 095 та 131	4
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0CA та 084	4
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0FC та 1EF	4
	Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 128 та 19C	4

Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 079 та 1BE	6
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0A5 та 199	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 0EF та 1DA	5
Визначити і підкреслити (у 16-ковій системі числення), чому дорівнює кодова відстань між шістнадцятьковими числами 018 та 0BC	3
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент HE. Число перед назвою елемента вказує на кількість входів.	2
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 2I-HE. Число перед назвою елемента вказує на кількість входів.	4
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 3I-HE. Число перед назвою елемента вказує на кількість входів.	6
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 4I-HE. Число перед назвою елемента вказує на кількість входів.	8
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 5I-HE. Число перед назвою елемента вказує на кількість входів.	A
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 6I-HE. Число перед назвою елемента вказує на кількість входів.	C
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 7I-HE. Число перед назвою елемента вказує на кількість входів.	E
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 8I-HE. Число перед назвою елемента вказує на кількість входів.	10
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 2ABO-HE. Число перед назвою елемента вказує на кількість входів.	4
Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 3ABO-HE. Число перед назвою	6

	елемента вказує на кількість входів.	
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 4АБО-НЕ. Число перед назвою елемента вказує на кількість входів.	8
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 5АБО-НЕ. Число перед назвою елемента вказує на кількість входів.	А
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 6АБО-НЕ. Число перед назвою елемента вказує на кількість входів.	С
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 7АБО-НЕ. Число перед назвою елемента вказує на кількість входів.	Е
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 8АБО-НЕ. Число перед назвою елемента вказує на кількість входів.	10
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 3	1
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення мажоритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 7	1





	двійкового еквіваленту десяткового числа 5	
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 11	1
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення міноритарної функції, якщо її аргументами є 3 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 2АБО. Число перед назвою елемента вказує на кількість входів.	6
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 3АБО. Число перед назвою елемента вказує на кількість входів.	8
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 4АБО. Число перед назвою елемента вказує на кількість входів.	А



	елемента вказує на кількість входів.	
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 7АБО-НЕ. Число перед назвою елемента вказує на кількість входів.	Е
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 2І-НЕ. Число перед назвою елемента вказує на кількість входів.	4
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 3І-НЕ. Число перед назвою елемента вказує на кількість входів.	6
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 4І-НЕ. Число перед назвою елемента вказує на кількість входів.	8
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 5І-НЕ. Число перед назвою елемента вказує на кількість входів.	А
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 6І-НЕ. Число перед назвою елемента вказує на кількість входів.	С
	Визначити і підкреслити, скільки КМОН-транзисторів (у 16-ковій системі числення) містить елемент 7І-НЕ. Число перед назвою елемента вказує на кількість входів.	Е
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг	1



	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=0$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 3	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового	1



	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 3	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1



	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $>T$ ", поріг дорівнює $T=4$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового	1



	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\geq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\theta < T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1



	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $< T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=1$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 3	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=2$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 0	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 1	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 2	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 3	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 4	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 5	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 6	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 7	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 8	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 9	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 10	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового	1

	еквіваленту десяткового числа 11	
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 12	0
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 13	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 14	1
	Визначити і підкреслити, яким буде результат обчислення порогової функції з одиничною вагою усіх її аргументів, якщо її одиничне значення формується при виконанні умови " $\leq T$ ", поріг дорівнює $T=3$ , а її аргументами є 4 молодших розряди двійкового еквіваленту десяткового числа 15	1
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить дешифратор "2 у 4"	4
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить дешифратор "3 у 8"	8
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить дешифратор "4 у 16"	10
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить мультиплексор "2 у 1".	2
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить мультиплексор "4 у 1".	4
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить мультиплексор "8 у 1".	8
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить мультиплексор "16 у 1".	10
	Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить демультиплексор "1 у 4"	4

Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить демультимплексор "1 у 8"	8
Визначити і підкреслити, скільки (у 16-ковій системі числення) елементів I містить демультимплексор "1 у 16"	10
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має мультимплексор "2 у 1"	1
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має мультимплексор "4 у 1"	2
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має мультимплексор "8 у 1"	3
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має мультимплексор "16 у 1"	4
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має демультимплексор "1 у 2"	1
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має демультимплексор "1 у 4"	2
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має демультимплексор "1 у 8"	3
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів керування має демультимплексор "1 у 16"	4
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має шифратор з 2 виходами	4
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має шифратор з 3 виходами	8
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має шифратор з 4 виходами	10
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має ПЗП з організацією 4 сл. x 16 розрядів	2
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має ПЗП з організацією 8 сл. x 16 розрядів	3
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має ПЗП з організацією 16 сл. x 16 розрядів	4
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має ПЗП з організацією 32 сл. x 16 розрядів	5
Визначити і підкреслити, скільки (у 16-ковій системі числення) входів має ПЗП з організацією 64 сл. x 16 розрядів	6









Визначити і підкреслити, скільки (у 16-ковій системі числення) виходів має ПЗП з організацією 64К сл. x 8 розрядів	8
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1М x 10 розрядів, ніж ПЗП з організацією 1К x А	А
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2М x 11 розрядів, ніж ПЗП з організацією 1К x В	В
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4М x 12 розрядів, ніж ПЗП з організацією 1К x С	С
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8М x 13 розрядів, ніж ПЗП з організацією 1К x D	D
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16М x 14 розрядів, ніж ПЗП з організацією 1К x E	E
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32М x 15 розрядів, ніж ПЗП з організацією 1К x F	F
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64М x 16 розрядів, ніж ПЗП з організацією 1К x 10	10
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1М x 9 розрядів, ніж ПЗП з організацією 2К x 9	9
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2М x 10 розрядів, ніж ПЗП з організацією 2К x А	А
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4М x 11 розрядів, ніж ПЗП з організацією 2К x В	В
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8М x 12 розрядів, ніж ПЗП з організацією 2К x С	С
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16М x 13 розрядів, ніж ПЗП з організацією 2К x D	D
Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32М x 14 розрядів, ніж ПЗП з	E

	організацією 2К x E	
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64М x 15 розрядів, ніж ПЗП з організацією 2К x F	F
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1М x 8 розрядів, ніж ПЗП з організацією 4К x 8	8
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2М x 9 розрядів, ніж ПЗП з організацією 4К x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4М x 10 розрядів, ніж ПЗП з організацією 4К x A	A
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8М x 11 розрядів, ніж ПЗП з організацією 4К x B	B
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16М x 12 розрядів, ніж ПЗП з організацією 4К x C	C
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32М x 13 розрядів, ніж ПЗП з організацією 4К x D	D
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64М x 14 розрядів, ніж ПЗП з організацією 4К x E	E
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1М x 7 розрядів, ніж ПЗП з організацією 8К x 7	7
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2М x 8 розрядів, ніж ПЗП з організацією 8К x 8	8
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4М x 9 розрядів, ніж ПЗП з організацією 8К x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8М x 10 розрядів, ніж ПЗП з організацією 8К x A	A
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16М x 11 розрядів, ніж ПЗП з	B

	організацією 8К x B	
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32М x 12 розрядів, ніж ПЗП з організацією 8К x C	C
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64М x 13 розрядів, ніж ПЗП з організацією 8К x D	D
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1М x 6 розрядів, ніж ПЗП з організацією 16К x 6	6
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2М x 7 розрядів, ніж ПЗП з організацією 16К x 7	7
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4М x 8 розрядів, ніж ПЗП з організацією 16К x 8	8
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8М x 9 розрядів, ніж ПЗП з організацією 16К x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16М x 10 розрядів, ніж ПЗП з організацією 16К x A	A
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32М x 11 розрядів, ніж ПЗП з організацією 16К x B	B
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64М x 12 розрядів, ніж ПЗП з організацією 16К x C	C
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1Г x 10 розрядів, ніж ПЗП з організацією 1М x A	A
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2Г x 11 розрядів, ніж ПЗП з організацією 1М x B	B
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4Г x 12 розрядів, ніж ПЗП з організацією 1М x C	C
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8Г x 13 розрядів, ніж ПЗП з	D

	організацією 1М x D	
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16Г x 14 розрядів, ніж ПЗП з організацією 1М x E	E
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32Г x 15 розрядів, ніж ПЗП з організацією 1М x F	F
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64Г x 16 розрядів, ніж ПЗП з організацією 1М x 10	10
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1Г x 9 розрядів, ніж ПЗП з організацією 2М x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2Г x 10 розрядів, ніж ПЗП з організацією 2М x A	A
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4Г x 11 розрядів, ніж ПЗП з організацією 2М x B	B
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8Г x 12 розрядів, ніж ПЗП з організацією 2М x C	C
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16Г x 13 розрядів, ніж ПЗП з організацією 2М x D	D
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32Г x 14 розрядів, ніж ПЗП з організацією 2М x E	E
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64Г x 15 розрядів, ніж ПЗП з організацією 2М x F	F
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1Г x 8 розрядів, ніж ПЗП з організацією 4М x 8	8
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2Г x 9 розрядів, ніж ПЗП з організацією 4М x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4Г x 10 розрядів, ніж ПЗП з	A

	організацією 4М x А	
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8Г x 11 розрядів, ніж ПЗП з організацією 4М x В	В
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16Г x 12 розрядів, ніж ПЗП з організацією 4М x С	С
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32Г x 13 розрядів, ніж ПЗП з організацією 4М x D	D
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64Г x 14 розрядів, ніж ПЗП з організацією 4М x E	E
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1Г x 7 розрядів, ніж ПЗП з організацією 8М x 7	7
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2Г x 8 розрядів, ніж ПЗП з організацією 8М x 8	8
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4Г x 9 розрядів, ніж ПЗП з організацією 8М x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8Г x 10 розрядів, ніж ПЗП з організацією 8М x А	А
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16Г x 11 розрядів, ніж ПЗП з організацією 8М x В	В
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32Г x 12 розрядів, ніж ПЗП з організацією 8М x С	С
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64Г x 13 розрядів, ніж ПЗП з організацією 8М x D	D
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 1Г x 6 розрядів, ніж ПЗП з організацією 16М x 6	6
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 2Г x 7 розрядів, ніж ПЗП з	7



	організацією 16М x 7	
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 4Г x 8 розрядів, ніж ПЗП з організацією 16М x 8	8
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 8Г x 9 розрядів, ніж ПЗП з організацією 16М x 9	9
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 16Г x 10 розрядів, ніж ПЗП з організацією 16М x А	А
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 32Г x 11 розрядів, ніж ПЗП з організацією 16М x В	В
	Визначити і підкреслити, на скільки (у 16-ковій системі числення) більше входів має ПЗП з організацією 64Г x 12 розрядів, ніж ПЗП з організацією 16М x С	С
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 7679. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	8
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 5740. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	F
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій	В

	<p>позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 4467. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 69DA. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	8
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 1898. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	8
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 6781. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	0
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого</p>	C

	(наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 20FD. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 0586. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	9
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 44BE. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	2
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 34D7. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	2
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій	5

	позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 57СА. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 2А33. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	0
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 2С23. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	8
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 49А4. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	D
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які	4

	дорівнюють степеням 2. Приймач отримав повідомлення 2621. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 0228. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	0
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 07E3. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	E
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 591E. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	5
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 0D42. У	1

	якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 5AE7. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	7
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 0EA6. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	6
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 4A2A. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	B
	При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 4FF6. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?	2

	0)?	
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 4343. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	8
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 7B70. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	D
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 0CCD. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	E
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 3365. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	1

	0)?	
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 6948. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	5
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 39B6. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	8
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 1E00. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	5
	<p>При пересиланні повідомлення використовується код Хеммінга, який може виправляти одну помилку. Довжина повідомлення разом з контрольними розрядами - 15 двійкових розрядів. Старший (лівий) двійковий розряд вважається 1-им (знаходиться на 1-ій позиції), за ним іде 2-ий і так далі до крайнього правого (наймолодшого), який вважається 15-им (знаходиться на 15-ій позиції). 4 контрольні розряди розташовано на позиціях, які дорівнюють степеням 2. Приймач отримав повідомлення 5F7E. У якому розряді помилка (відсутність помилки - це помилка у розряді 0)?</p>	5









	вказують на її немонотонність? (якщо 0 пар - функція монотонна).	
	ФАЛЗ $f(a,b,c)$ , має на своїх наборах (7-му, ..., 0-му) значення, які відповідають 8-ми молодшим бітам (7-му, ..., 0-му) шістнадцяткового числа D8. Скільки пар сусідніх наборів цієї ФАЛ вказують на її немонотонність? (якщо 0 пар - функція монотонна).	2
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 14:9 у простому полі Галуа GF(17)? Кількість елементів у полі вказано в 10-ій системі числення.	B
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 15:11 у простому полі Галуа GF(17)? Кількість елементів у полі вказано в 10-ій системі числення.	6
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 6+6 у простому полі Галуа GF(7)? Кількість елементів у полі вказано в 10-ій системі числення.	5
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 7-8 у простому полі Галуа GF(11)? Кількість елементів у полі вказано в 10-ій системі числення.	A
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 11:8 у простому полі Галуа GF(17)? Кількість елементів у полі вказано в 10-ій системі числення.	C
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 4:15 у простому полі Галуа GF(17)? Кількість елементів у полі вказано в 10-ій системі числення.	F
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 12:3 у простому полі Галуа GF(17)? Кількість елементів у полі вказано в 10-ій системі числення.	4
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 1-3 у простому полі Галуа GF(11)? Кількість елементів у полі вказано в 10-ій системі числення.	9
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 10-9 у простому полі Галуа GF(11)? Кількість елементів у полі вказано в 10-ій системі числення.	1
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 9-4 у простому полі Галуа GF(11)? Кількість елементів у полі вказано в 10-ій системі числення.	5
	Яким буде шістнадцятьковий результат виконання опреції над десятковими числами 4x8 у простому полі Галуа GF(13)? Кількість елементів у полі вказано в 10-ій системі числення.	6



Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $4+4$ у простому полі Галуа $GF(7)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	1
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $9-1$ у простому полі Галуа $GF(11)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	8
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $9 \times 8$ у простому полі Галуа $GF(13)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	7
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $5 \times 11$ у простому полі Галуа $GF(13)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	3
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $3+6$ у простому полі Галуа $GF(7)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	2
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $1 \times 11$ у простому полі Галуа $GF(13)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	В
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $0:10$ у простому полі Галуа $GF(17)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	0
Яким буде шістнадцятьковий результат виконання опреції над десятковими числами $3+4$ у простому полі Галуа $GF(7)$ ? Кількість елементів у полі вказано в 10-ій системі числення.	0
Скільки термів 4 змінних покриває імпліканта $ac$ ?	4
Скільки термів 4 змінних покриває імпліканта $abd$ ?	2
Скільки термів 3 змінних покриває імпліканта $c$ ?	4
Скільки термів 4 змінних покриває імпліканта $a$ ?	8
Скільки термів 5 змінних покриває імпліканта $ae$ ?	8
Скільки термів 3 змінних покриває імпліканта $abc$ ?	1
Скільки термів 5 змінних покриває імпліканта $abce$ ?	2

	Скільки термів 5 змінних покриває імпліканта be?	8
	Скільки термів 3 змінних покриває імпліканта bc?	2
	Скільки термів 3 змінних покриває імпліканта a?	4
	Скільки термів 5 змінних покриває імпліканта c?	10
	Скільки термів 5 змінних покриває імпліканта a?	10
	Скільки термів 3 змінних покриває імпліканта a?	4
	Скільки термів 5 змінних покриває імпліканта acd?	4
	Скільки термів 4 змінних покриває імпліканта ab?	4
	Скільки термів 5 змінних покриває імпліканта a?	10
	Скільки термів 3 змінних покриває імпліканта ac?	2
	Скільки термів 4 змінних покриває імпліканта ad?	4
	Скільки термів 3 змінних покриває імпліканта abc?	1
	Скільки термів 3 змінних покриває імпліканта b?	4
	Скільки термів 3 змінних покриває імпліканта ac?	2
	Скільки термів 4 змінних покриває імпліканта a?	8
	Скільки термів 5 змінних покриває імпліканта bce?	4
	Скільки термів 3 змінних покриває імпліканта b?	4
	Скільки термів 4 змінних покриває імпліканта acd?	2

	Скільки термів 4 змінних покриває імпліканта $ac$ ?	4
	Скільки термів 4 змінних покриває імпліканта $bd$ ?	4
	Скільки термів 4 змінних покриває імпліканта $abcd$ ?	1
	Скільки термів 3 змінних покриває імпліканта $bc$ ?	2
	Скільки термів 3 змінних покриває імпліканта $b$ ?	4
	Скільки клітинок на карті Карно для 4 змінних покриває імпліканта $abcd$ ?	1
	Скільки клітинок на карті Карно для 5 змінних покриває імпліканта $acde$ ?	2
	Скільки клітинок на карті Карно для 4 змінних покриває імпліканта $c$ ?	8
	Скільки клітинок на карті Карно для 3 змінних покриває імпліканта $b$ ?	4
	Скільки клітинок на карті Карно для 4 змінних покриває імпліканта $bd$ ?	4
	Скільки клітинок на карті Карно для 5 змінних покриває імпліканта $abcde$ ?	1
	Скільки клітинок на карті Карно для 5 змінних покриває імпліканта $ace$ ?	4
	Скільки клітинок на карті Карно для 4 змінних покриває імпліканта $bd$ ?	4
	Скільки клітинок на карті Карно для 5 змінних покриває імпліканта $ace$ ?	4
	Скільки клітинок на карті Карно для 5 змінних покриває імпліканта $ad$ ?	8
	Скільки клітинок на карті Карно для 3 змінних покриває імпліканта $ab$ ?	2
	Скільки клітинок на карті Карно для 3 змінних покриває імпліканта $bc$ ?	2
	Скільки клітинок на карті Карно для 3 змінних покриває імпліканта $abc$ ?	1



Скільки клітинок на карті Карно для 5 змінних покриває імпліканта bcd?	4
Скільки клітинок на карті Карно для 4 змінних покриває імпліканта abcd?	1
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта a?	4
Скільки клітинок на карті Карно для 5 змінних покриває імпліканта abcd?	2
Скільки клітинок на карті Карно для 5 змінних покриває імпліканта abce?	2
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта a?	4
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта ac?	2
Скільки клітинок на карті Карно для 4 змінних покриває імпліканта ab?	4
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта ac?	2
Скільки клітинок на карті Карно для 4 змінних покриває імпліканта ab?	4
Скільки клітинок на карті Карно для 5 змінних покриває імпліканта ce?	8
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта b?	4
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта c?	4
Скільки клітинок на карті Карно для 5 змінних покриває імпліканта abcd?	2
Скільки клітинок на карті Карно для 4 змінних покриває імпліканта ab?	4
Скільки клітинок на карті Карно для 3 змінних покриває імпліканта ac?	2
Скільки клітинок на карті Карно для 5 змінних покриває імпліканта a?	10
Визначити і підкреслити, яким буде результат обчислення логічної функції NOR, якщо її аргументами є 5 молодших біти двійкового еквіваленту десяткового числа 0	1











































Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 4 молодших біти двійкового еквіваленту десяткового числа 248	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 7 молодших біти двійкового еквіваленту десяткового числа 96	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 7 молодших біти двійкового еквіваленту десяткового числа 226	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 7 молодших біти двійкового еквіваленту десяткового числа 219	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 6 молодших біти двійкового еквіваленту десяткового числа 77	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 8 молодших біти двійкового еквіваленту десяткового числа 167	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 4 молодших біти двійкового еквіваленту десяткового числа 81	1
Визначити і підкреслити, яким буде результат логічного додавання, якщо його аргументами є 5 молодших біти двійкового еквіваленту десяткового числа 192	0
Визначити і підкреслити мінімальну розрядність зворотного зв'язку (кількість біт у 16-ковій системі числення) в автоматі Мура, якщо кількість станів у яких він може знаходитися, дорівнює 674	A
Визначити і підкреслити мінімальну розрядність зворотного зв'язку (кількість біт у 16-ковій системі числення) в автоматі Мура, якщо кількість станів у яких він може знаходитися, дорівнює 2	1
Визначити і підкреслити мінімальну розрядність зворотного зв'язку (кількість біт у 16-ковій системі числення) в автоматі Мура, якщо кількість станів у яких він може знаходитися, дорівнює 39	6
Визначити і підкреслити мінімальну розрядність зворотного зв'язку (кількість біт у 16-ковій системі числення) в автоматі Мура, якщо кількість станів у яких він може знаходитися, дорівнює 23772	F
Визначити і підкреслити мінімальну розрядність зворотного зв'язку (кількість біт у 16-ковій системі числення) в автоматі Мура, якщо кількість станів у яких він може знаходитися, дорівнює 3	2



















Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 4	2
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 2540	C
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 2	1
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 7714	D
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 6	3
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 17480	F
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 9737	E
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 36	6
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 912	A
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 2266	C
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 10	4
Визначити і підкреслити мінімальну розрядність (кількість біт у 16-ковій системі числення) коду вершин графа автомата Мура, якщо кількість станів у яких автомат може знаходитися, дорівнює 12	4
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=bcd$ , сформує значення 0 на своєму виході?	E
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=abc$ , сформує значення 0 на своєму виході?	E



На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=bcd$ , сформує значення 0 на своєму виході?	Е
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=ac$ , сформує значення 0 на своєму виході?	С
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c)=bc$ , сформує значення 0 на своєму виході?	6
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d,e)=ae$ , сформує значення 1 на своєму виході?	8
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=b$ , сформує значення 1 на своєму виході?	8
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c)=ab$ , сформує значення 0 на своєму виході?	6
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=ac$ , сформує значення 0 на своєму виході?	С
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=ad$ , сформує значення 1 на своєму виході?	4
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d,e)=ab$ , сформує значення 1 на своєму виході?	8
На скількох наборах змінних елемент, який реалізує функцію $f(a,b,c,d)=ad$ , сформує значення 1 на своєму виході?	4
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=b$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=ac(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	4
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	1
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim a)(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	4
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=bc(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	4





Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim a)bde$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	5
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=de$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=ab(\sim c)(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	6
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=b(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	3
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim b)de$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	4
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=acde$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	4
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim a)(\sim b)c(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	7
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim a)(\sim b)(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	6
Визначити найменшу кількість входів у елементів базису Буля, які реалізують ФАЛ $f=(\sim a)(\sim c)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	5
Скільки літер додає до скороченої ДНФ функції 2 змінних склеювання 2 клітинок на карті Карно?	1
Скільки літер додає до скороченої ДНФ функції 5 змінних склеювання 32 клітинок на карті Карно?	0
Скільки літер додає до скороченої ДНФ функції 11 змінних склеювання 128 клітинок на карті Карно?	4
Скільки літер додає до скороченої ДНФ функції 11 змінних склеювання 128 клітинок на карті Карно?	4
Скільки літер додає до скороченої ДНФ функції 7 змінних склеювання 128 клітинок на карті Карно?	0



Скільки літер додає до скороченої ДНФ функції 3 змінних склеювання 1 клітинок на карті Карно?	3
Скільки літер додає до скороченої ДНФ функції 8 змінних склеювання 32 клітинок на карті Карно?	3
Скільки літер додає до скороченої ДНФ функції 4 змінних склеювання 4 клітинок на карті Карно?	2
Скільки літер додає до скороченої ДНФ функції 6 змінних склеювання 32 клітинок на карті Карно?	1
Скільки літер додає до скороченої ДНФ функції 9 змінних склеювання 64 клітинок на карті Карно?	3
Скільки літер додає до скороченої ДНФ функції 12 змінних склеювання 256 клітинок на карті Карно?	4
Скільки літер додає до скороченої ДНФ функції 8 змінних склеювання 64 клітинок на карті Карно?	2
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim a)b(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ .	4
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim b)$ , якщо $(\sim x)$ позначає інверсію $x$ .	1
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=c$ , якщо $(\sim x)$ позначає інверсію $x$ .	0
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim d)e$ , якщо $(\sim x)$ позначає інверсію $x$ .	3
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim a)(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ .	4
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim b)(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ .	4
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ .	0
Визначити найменшу кількість виходів у елементів базису "І-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=b(\sim c)(\sim d)e$ , якщо $(\sim x)$ позначає інверсію $x$ .	4



















Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=ac(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ .	3
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim a)(\sim c)d(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=c(\sim d)(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim d)(\sim e)$ , якщо $(\sim x)$ позначає інверсію $x$ .	1
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=a(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=(\sim a)b(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=e$ , якщо $(\sim x)$ позначає інверсію $x$ .	0
Визначити найменшу кількість виходів у елементів базису "АБО-НЕ" з довільною кількістю входів, які реалізують ФАЛ $f=b(\sim d)e$ , якщо $(\sim x)$ позначає інверсію $x$ .	3
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=b$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=c$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim a)b(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	С
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=bc(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	А



Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim a)bc$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim a)c$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim b)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim b)(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim b)cd$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=ac$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	6
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2

Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=acd$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim a)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису Буля, яка реалізує ФАЛ $f=(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "I, HE", яка реалізує ФАЛ $f=a(\sim b)(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	C
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "I, HE", яка реалізує ФАЛ $f=(\sim a)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "I, HE", яка реалізує ФАЛ $f=b$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "I, HE", яка реалізує ФАЛ $f=a(\sim c)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A



Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=ac(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim b)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)bc$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim b)cd$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	E
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim b)c$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=bc$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	6
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=ac$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	6
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=b(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2

Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)b(\sim c)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	Е
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=abc$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim c)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim b)(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	Е
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)b(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	С
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=b(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	А
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=cd$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	6
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2

Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=c(\sim d)$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=a(\sim b)(\sim c)(\sim d)$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	10
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=a$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=d$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim c)(\sim d)$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	Е
Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim b)c$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	С
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim a)bc$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	Е
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim b)$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim c)(\sim d)$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	8
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim a)$ , якщо ( $\sim x$ ) позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2

'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim a)(\sim b)c$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	C
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a(\sim b)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	E
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=ab(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	E
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=b$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=abc(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	12

'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim b)cd$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	E
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim a)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim c)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim b)(\sim c)(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim a)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=b(\sim c)(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	C

'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=abd$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	10
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=(\sim c)d$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=b(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	A
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО, НЕ", яка реалізує ФАЛ $f=a(\sim b)(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , а багатовходові елементи мають необмежену кількість входів.	C
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=(\sim d)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=d$ , якщо $(\sim x)$ позначає інверсію $x$ .	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=(\sim b)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=d$ , якщо $(\sim x)$ позначає інверсію $x$ .	0
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=cd$ , якщо $(\sim x)$ позначає інверсію $x$ .	6
'Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "І-НЕ" з довільною кількістю входів, яка реалізує	0











	реалізує ФАЛ $f=bd$ , якщо $(\sim x)$ позначає інверсію $x$ .	
	Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=(\sim a)$ , якщо $(\sim x)$ позначає інверсію $x$ .	2
	Скільки КМОН-транзисторів необхідно для створення схеми з елементів базису "АБО-НЕ" з довільною кількістю входів, яка реалізує ФАЛ $f=a$ , якщо $(\sim x)$ позначає інверсію $x$ .	0
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 8, 0, 6, 6	1
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 5, 1, 2, 0	1
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 14, 12, 7, 4	5
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 6, 4, 0, 4	1
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 11, 3, 0, 3	1
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 4, 3, 3, 3	0
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 11, 1, 2, 9	2
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 13, 8, 4, 1	2
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 7, 1, 4, 3	2
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ , $a$ , $b$ , $c$ мають значення, відповідно, 16, 14, 11, 4	5
	За правилами $N$ -значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо $(\sim x)$ позначає інверсію $x$ , $N$ ,	4



	a, b, c мають значення, відповідно, 16, 8, 8, 3	
	За правилами N-значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо ( $\sim x$ ) позначає інверсію x, N, a, b, c мають значення, відповідно, 13, 7, 5, 11	6
	За правилами N-значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо ( $\sim x$ ) позначає інверсію x, N, a, b, c мають значення, відповідно, 5, 0, 2, 3	1
	За правилами N-значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо ( $\sim x$ ) позначає інверсію x, N, a, b, c мають значення, відповідно, 5, 3, 2, 3	3
	За правилами N-значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо ( $\sim x$ ) позначає інверсію x, N, a, b, c мають значення, відповідно, 14, 12, 7, 5	6
	За правилами N-значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо ( $\sim x$ ) позначає інверсію x, N, a, b, c мають значення, відповідно, 14, 1, 12, 6	2
	За правилами N-значної алгебри логіки Поста визначити значення ФАЛ $f=(\sim a)bc \vee a(\sim b)c \vee ab(\sim c)$ , якщо ( $\sim x$ ) позначає інверсію x, N, a, b, c мають значення, відповідно, 10, 5, 9, 5	5
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=bc$ ?	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b)=b$ ?	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b)=b$ ?	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=b$ ?	4
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b)=a$ ?	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію	2



	$f(a,b)=a?$	
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=ab?$	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=a?$	4
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=abc?$	0
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c,d)=bd?$	4
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b)=a?$	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=bc?$	2
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c,d)=ab?$	4
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c,d)=b?$	8
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c,d)=abcd?$	0
	Скільки виходів дешифратора з входами для всіх змінних необхідно з'єднати з елементом АБО у схемі, яка реалізує функцію $f(a,b,c)=bc?$	2
	Яку найменшу кількість входів повинен мати елемент АБО у схемі на основі дешифратора з входами для всіх змінних, яка реалізує функцію $f(a,b,c)=b?$	4
	Яку найменшу кількість входів повинен мати елемент АБО у схемі на основі дешифратора з входами для всіх змінних, яка реалізує функцію $f(a,b,c)=a?$	4
	Яку найменшу кількість входів повинен мати елемент АБО у схемі на основі дешифратора з входами для всіх змінних, яка реалізує	2







	функцію $f(a,b,c)=bc$ ?	
	Яку найменшу кількість входів повинен мати елемент АБО у схемі на основі дешифратора з входами для всіх змінних, яка реалізує функцію $f(a,b,c,d)=ab$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c,d)=bc$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c,d)=acd$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b)=b$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=ac$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=bc$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c,d)=abcd$ ?	1
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b)=b$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b)=a$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c,d)=a$ ?	8
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка	1



	реалізує функцію $f(a,b)=a$ ?	
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c,d)=b$ ?	8
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=ac$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b)=ab$ ?	1
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=ab$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b,c)=bc$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 1 у схемі, яка реалізує функцію $f(a,b)=b$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c,d)=bd$ ?	C
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=ab$ ?	6
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c,d)=abd$ ?	E
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b)=a$ ?	2
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=abc$ ?	7
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка	2



	реалізує функцію $f(a,b)=a$ ?	
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=ac$ ?	6
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c,d)=ac$ ?	C
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=ac$ ?	6
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=a$ ?	4
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c)=bc$ ?	6
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c,d)=acd$ ?	E
	На скільки інформаційних входів мультиплексора з входами керування для всіх змінних необхідно завести 0 у схемі, яка реалізує функцію $f(a,b,c,d)=ac$ ?	C
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: ABE5 та 8FC7?	A
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: DFDD та 4269?	D
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-	2

	кових чисел: CCB8 та 7346?	
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: AE0A та A7DF?	9
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: E2B2 та BD6A?	C
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 9616 та CFF2?	C
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: BE6C та 8712?	6
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 2D4A та B0EB?	6
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 259B та 046D?	A
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: F933 та DE3E?	A
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: ADBA та DB31?	A
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: E622 та E535?	5
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-	D

	кових чисел: 2B8C та E7FD?	
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: D6F5 та C363?	B
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: BE28 та 3776?	7
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: E96F та 755C?	9
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 436E та 7E34?	B
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: D004 та A769?	1
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 32E5 та 978D?	9
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 84AF та A3E6?	A
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: DB4D та 8804?	5
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 19C0 та E533?	1
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-	1



	кових чисел: 5905 та 8076?	
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: C808 та DF6E?	5
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 889E та 4CD8?	4
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: D2D5 та 8F3C?	C
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 52AE та D122?	7
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 4371 та 6908?	3
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: A859 та EF09?	7
	Скільки рівних 1 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 513C та AC7A?	4
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: F95B та F5CB?	5
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 9AFD та FBFF?	1
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-	9

	кових чисел: BEAA та 1394?	
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 57F6 та 4324?	8
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 0FF1 та D217?	4
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 5893 та 2411?	D
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 72C8 та E5A2?	9
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 287F та 0982?	7
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 4271 та 6786?	D
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 94B4 та F1F6?	6
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 92DD та 905A?	A
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 96AF та CD22?	8
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-	8

	кових чисел: 094F та 3929?	
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: C4DD та B0D0?	C
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 836C та 6DB3?	9
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: A50F та D2C4?	D
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 1503 та FCEF?	6
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 07FE та 0DC7?	5
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 52EE та 65A9?	8
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 18B0 та B06E?	B
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: EAE2 та 7C73?	7
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 5DF9 та B58C?	4
	Скільки рівних 0 двійкових переносів сформуєть повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-	B

	кових чисел: 4095 та D95E?	
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: BF9A та 4A8B?	4
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: AEA2 та 5BA2?	5
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 2472 та B029?	D
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: A56F та 9C00?	B
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 3EAC та F95D?	2
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: 4EF9 та 28B8?	A
	Скільки рівних 0 двійкових переносів сформує повні однорозрядні двійкові суматори, які утворюють 16-бітний суматор з наскрізним переносом, у якому виконується додавання двох 16-кових чисел: A3A0 та 88B6?	B
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 89E4 та 2145?	4
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 9E89 та 6B92?	9
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	4

	кових чисел: 0149 та 7D0B?	
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: EAC1 та 4967?	A
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: F344 та 8423?	1
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 3C53 та 8374?	3
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: FE59 та 6124?	3
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: A16D та 544D?	4
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: E494 та 27FD?	C
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: B02E та 23BE?	6
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 63B7 та FCB9?	F
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 6A34 та F2F2?	8
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	9

	кових чисел: F7E4 та DB82?	
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: B058 та 940C?	5
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 1245 та B92E?	4
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 2615 та B342?	3
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 21A6 та 7933?	6
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 603D та 2686?	6
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 8954 та 4592?	2
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 35D6 та E872?	A
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 7FFA та 4C67?	E
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 7AED та 6B45?	B
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	4

	кових чисел: F29C та 8293?	
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: DB5B та 323D?	C
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 3217 та 1561?	5
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: AA3A та 5DA9?	8
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: BEA9 та 0030?	1
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: F95E та C1F4?	9
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 915B та 3D88?	5
	Скільки рівних 1 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 760F та C749?	8
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: F0E6 та E6E2?	8
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 0DF5 та FF04?	7
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	B

	кових чисел:7870 та D803?	
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:5EDB та 524B?	5
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:8F0D та 8FF2?	B
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:97F2 та 1302?	B
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:FDBC та 772C?	4
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:AD02 та CDDC?	C
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:E9DB та F65C?	4
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:F0C0 та 0456?	E
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:E7FC та 2455?	4
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:7001 та 698F?	A
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	2



	кових чисел:6F7D та C2D7?	
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:A23C та 5338?	C
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:72DE та CССА?	8
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:50E6 та 0ED5?	D
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:029D та 934E?	C
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:D414 та F241?	C
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:9518 та 3CDE?	A
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:6BCB та DFB2?	6
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:6361 та A84B?	A
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:65A2 та 2A0E?	B
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	3

	кових чисел:7FF3 та 5D25?	
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:08D8 та 25F0?	В
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:9D06 та FCD4?	9
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:DA92 та BE48?	9
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:731A та 1565?	А
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:772E та D250?	А
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:DA03 та 9A83?	А
	Скільки рівних 0 двійкових переносів сформує 16-бітний вузол прискорення переносів, який працює в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел:8AC8 та 1A82?	С
	Скільки рівних 1 ознак G генерації переносів сформує однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 03CB та 695D?	4
	Скільки рівних 1 ознак G генерації переносів сформує однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 904D та 86DF?	5
	Скільки рівних 1 ознак G генерації переносів сформує однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	6

	кових чисел: 687E та 4B6B?	
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 6826 та 260D?	2
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: A916 та A2D3?	4
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 98BA та 099F?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: B4B7 та B381?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: EE0A та 49ED?	3
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 1856 та AFB5?	3
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 6108 та 6429?	3
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 0B78 та 11EF?	4
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: A733 та CF65?	6
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	1

	кових чисел: 005A та 8660?	
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 05DF та E68C?	4
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 99FF та BCAA?	7
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: B0AA та 0A69?	2
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 008D та F969?	2
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: BA72 та 0606?	2
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 8B60 та 677D?	4
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: B737 та 392A?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: C3C1 та CFB0?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: D940 та AF04?	3
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	4

	кових чисел: 4C1C та AEF4?	
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: A755 та E5AB?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: FB57 та 0F15?	6
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 59B4 та CC7E?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: B3E9 та 4E2E?	3
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 1381 та AFFE?	3
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 66C9 та B6C0?	5
	Скільки рівних 1 ознак G генерації переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 5677 та AC4D?	4
	Скільки рівних 1 ознак P проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 3CDF та 03D8?	D
	Скільки рівних 1 ознак P проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: C323 та D302?	8
	Скільки рівних 1 ознак P проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	B

	кових чисел: 76B0 та A6A3?	
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 0381 та FA62?	C
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 812F та 9DCD?	C
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: CF43 та 230B?	B
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 5C29 та A37D?	E
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 3789 та 434E?	C
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 9A4E та 3EAA?	C
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 2D82 та 3000?	7
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 53FB та 6BB5?	E
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 3BFD та 1773?	E
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	C

	кових чисел: D2CA та 6506?	
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 103B та 4EA3?	B
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: F43E та FF31?	E
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 8707 та 418A?	A
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: CC89 та 4665?	B
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: CB96 та EBE3?	D
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 2CDE та 3598?	B
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 23FF та 7646?	E
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 2717 та 73B9?	D
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 50DA та B370?	C
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-	9

	кових чисел: 42D2 та 0A9C?	
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: A548 та 5B56?	D
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: E5C6 та C837?	D
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 2F4C та 1BF1?	D
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 2126 та B5BB?	C
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: 04D4 та 6E1C?	A
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: C1EE та C9A3?	B
	Скільки рівних 1 ознак Р проходження переносів сформуєть однобітні суматори, які працюють в складі 16-бітного суматора з паралельним переносом, у якому виконується додавання двох 16-кових чисел: A759 та DA95?	E
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням М і 4-бітним множником N. Множення починається з молодших розрядів множника із зсувом часткових сум S праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми S. Якою буде 2-га часткова сума S, якщо $M = 60$ , $N = A$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	C



	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 92</math>, <math>N = 9</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	В
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 98</math>, <math>N = 0</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	0
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 2C</math>, <math>N = C</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	В
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = A5</math>, <math>N = 6</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	0
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий</p>	С

	і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 18$ , $N = 3$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = A9$ , $N = 8$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	0
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = D6$ , $N = 4$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	1
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = BD$ , $N = 5$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	0
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 67$ , $N = 3$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	9

	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 4E</math>, <math>N = 8</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	0
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 92</math>, <math>N = 5</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	A
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = BA</math>, <math>N = 0</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	0
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 6F</math>, <math>N = D</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	0
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий</p>	9

	і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 08$ , $N = 3$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = AC$ , $N = E$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	C
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 32$ , $N = A$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	A
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = E5$ , $N = 8$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	0
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = F4$ , $N = 1$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	4

	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множителем <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 33</math>, <math>N = 6</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	3
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множителем <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 30</math>, <math>N = 6</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	3
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множителем <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 84</math>, <math>N = 6</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	6
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множителем <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 62</math>, <math>N = 4</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	2
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множителем <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий</p>	2

	і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 17$ , $N = 4$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 8B$ , $N = 0$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	0
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = AA$ , $N = 1$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	5
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 91$ , $N = D$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	5
	Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням $M$ і 4-бітним множником $N$ . Множення починається з молодших розрядів множника із зсувом часткових сум $S$ праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми $S$ . Якою буде 2-га часткова сума $S$ , якщо $M = 06$ , $N = B$ ? На початку $S=0$ . Результат записати як $S \bmod 15$ .	3

	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 93</math>, <math>N = C</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	3
	<p>Помножувач здійснює операцію над 16-ковими цілими і додатніми 8 бітним множенням <math>M</math> і 4-бітним множенням <math>N</math>. Множення починається з молодших розрядів множника із зсувом часткових сум <math>S</math> праворуч. 1-а часткова сума утворюється додаванням першого і другого часткових добутоків (які утворюються множенням на 0-ий і 1-ий розряди множника, відповідно). Після знаходження кожного наступного часткового добутку обраховується значення наступної часткової суми <math>S</math>. Якою буде 2-га часткова сума <math>S</math>, якщо <math>M = 35</math>, <math>N = A</math>? На початку <math>S=0</math>. Результат записати як <math>S \bmod 15</math>.</p>	1
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код <math>K</math> десяткового числа 2,875? Як відповідь записати число <math>K \bmod 15</math>. При неможливості утворити код <math>K</math> як результат записати 16-кове 10.</p>	D
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код <math>K</math> десяткового числа -0,34375? Як відповідь записати число <math>K \bmod 15</math>. При неможливості утворити код <math>K</math> як результат записати 16-кове 10.</p>	2
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код <math>K</math> десяткового числа -0,71875? Як відповідь записати число <math>K \bmod 15</math>. При неможливості утворити код <math>K</math> як результат записати 16-кове 10.</p>	4
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди</p>	3

	мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К десяткового числа -0,1875? Як вiдповiдь записати число Kmod15. При неможливостi утворити код К як результат записати 16-кове 10.	
	Формат чисел з рухомою комою складеться з 8 бiт: 1 бiт (старший) - знак, далi 3 бiта змiщений на 4 порядок i 4 бiта - молодшi розряди мантиси, старшi розряди мантиси (0,1 у двiйовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К десяткового числа 6,25? Як вiдповiдь записати число Kmod15. При неможливостi утворити код К як результат записати 16-кове 10.	1
	Формат чисел з рухомою комою складеться з 8 бiт: 1 бiт (старший) - знак, далi 3 бiта змiщений на 4 порядок i 4 бiта - молодшi розряди мантиси, старшi розряди мантиси (0,1 у двiйовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К десяткового числа 6? Як вiдповiдь записати число Kmod15. При неможливостi утворити код К як результат записати 16-кове 10.	0
	Формат чисел з рухомою комою складеться з 8 бiт: 1 бiт (старший) - знак, далi 3 бiта змiщений на 4 порядок i 4 бiта - молодшi розряди мантиси, старшi розряди мантиси (0,1 у двiйовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К десяткового числа 1,25? Як вiдповiдь записати число Kmod15. При неможливостi утворити код К як результат записати 16-кове 10.	9
	Формат чисел з рухомою комою складеться з 8 бiт: 1 бiт (старший) - знак, далi 3 бiта змiщений на 4 порядок i 4 бiта - молодшi розряди мантиси, старшi розряди мантиси (0,1 у двiйовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К десяткового числа 0,078125? Як вiдповiдь записати число Kmod15. При неможливостi утворити код К як результат записати 16-кове 10.	5
	Формат чисел з рухомою комою складеться з 8 бiт: 1 бiт (старший) - знак, далi 3 бiта змiщений на 4 порядок i 4 бiта - молодшi розряди мантиси, старшi розряди мантиси (0,1 у двiйовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К десяткового числа -3,875? Як вiдповiдь записати число Kmod15. При неможливостi утворити код К як результат записати 16-кове 10.	E
	Формат чисел з рухомою комою складеться з 8 бiт: 1 бiт (старший) - знак, далi 3 бiта змiщений на 4 порядок i 4 бiта - молодшi розряди мантиси, старшi розряди мантиси (0,1 у двiйовому кодi) у форматi не зберiгаються. Яким буде в описаному форматi код К	B



	десятькового числа 0,375? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десятикового числа -0,65625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	2
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десятикового числа -0,1640625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	0
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десятикового числа 0,265625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	4
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десятикового числа 0,84375? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	0
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десятикового числа -0,09765625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	3
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi	1

	не зберігаються. Яким буде в описаному форматі код К десяткового числа 0,234375? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код К десяткового числа 0,15625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	6
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код К десяткового числа 0? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	0
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код К десяткового числа 2,75? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	C
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код К десяткового числа 5,75? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	E
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код К десяткового числа 0? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.	0
	Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому коді) у форматі не зберігаються. Яким буде в описаному форматі код К десяткового числа -0,078125? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове	D

10.		
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа -1,625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.</p>	8
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа 0? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.</p>	0
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа 0,0625? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.</p>	1
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа -0,53125? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.</p>	D
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа -3,5? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове 10.</p>	B
	<p>Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа -1,8125? Як відповідь записати число Kmod15. При неможливості утворити код К як результат записати 16-кове</p>	B

10.		
Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа 0,1953125? Як відповідь записати число $K \bmod 15$ . При неможливості утворити код К як результат записати 16-кове 10.		В
Формат чисел з рухомою комою складеться з 8 біт: 1 біт (старший) - знак, далі 3 біта зміщений на 4 порядок і 4 біта - молодші розряди мантиси, старші розряди мантиси (0,1 у двійовому кодi) у форматi не зберігаються. Яким буде в описаному форматi код К десяткового числа -5,75? Як відповідь записати число $K \bmod 15$ . При неможливості утворити код К як результат записати 16-кове 10.		7
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 30,5 до 10-кової системи числення		5
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 70,4 до 10-кової системи числення		4
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 37,1 до 10-кової системи числення		1
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 58,0 до 10-кової системи числення		0
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 11,5 до 10-кової системи числення		5
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 07,8 до 10-кової системи числення		8
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 27,8 до 10-кової системи числення		8
Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системi числення з основою 9 числа 83,7 до 10-кової системи числення		7



































	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 3 числа 12,2 до 10-кової системи числення	6
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 53,5 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	3
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 11,1 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	7
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 7 числа 32,6 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	8
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 13,4 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	9
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 5 числа 31,1 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	1
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 7 числа 32,3 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	8
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 13,3 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	9
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 4 числа 30,1 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	C
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 3 числа 01,0 до 10-кової системи числення (результат представити у вигляді 16-кового числа $X \bmod 15$ )	1
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 3 числа 02,2 до 10-кової системи	2



	Xmod15)	
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 40,1 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	9
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 41,2 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	A
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 4 числа 22,1 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	A
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 32,1 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	5
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 9 числа 34,8 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	1
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 5 числа 34,3 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	4
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 7 числа 04,2 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	4
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 9 числа 38,3 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	5
	Якою буде ціла частина $X$ результату переведення записаного у системі числення з основою 6 числа 13,0 до 10-кової системи числення (результат представити у вигляді 16-кового числа Xmod15)	9
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 10 числа 23,0 до системи числення з основою	0









	цифри та перші літери латинської абетки.	
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 10 числа 26,6 до системи числення з основою 16. Для запису розрядів результату використовувати десяткові цифри та перші літери латинської абетки.	9
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 10 числа 20,6 до системи числення з основою 16. Для запису розрядів результату використовувати десяткові цифри та перші літери латинської абетки.	9
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 10 числа 77,8 до системи числення з основою 16. Для запису розрядів результату використовувати десяткові цифри та перші літери латинської абетки.	C
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 10 числа 61,6 до системи числення з основою 16. Для запису розрядів результату використовувати десяткові цифри та перші літери латинської абетки.	9
	Визначити і підкреслити, якою буде друга цифра після коми результату переведення без заокруглення записаного у системі числення з основою 10 числа 04,9 до системи числення з основою 16. Для запису розрядів результату використовувати десяткові цифри та перші літери латинської абетки.	6
	Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-8$ , а $N=-21$ ?	3
	Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-5$ , а $N=-57$ ?	3
	Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-6$ , а $N=-5$ ?	1
	Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-7$ , а $N=66$ ?	3
	Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-3$ , а $N=41$ ?	2
	Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=5$ , а $N=-63$ ?	7

Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=7$ , а $N=12$ ?	C
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-3$ , а $N=-78$ ?	3
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=4$ , а $N=24$ ?	4
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=7$ , а $N=-87$ ?	B
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-5$ , а $N=86$ ?	1
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=8$ , а $N=99$ ?	B
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=3$ , а $N=18$ ?	3
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-7$ , а $N=-32$ ?	3
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=6$ , а $N=76$ ?	A
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-5$ , а $N=-98$ ?	2
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=3$ , а $N=66$ ?	3
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=4$ , а $N=-42$ ?	6
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-5$ , а $N=-34$ ?	1
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=6$ , а $N=-73$ ?	B
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=6$ , а $N=54$ ?	6
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=8$ , а $N=41$ ?	9
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-7$ , а $N=-17$ ?	4
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-6$ , а $N=-73$ ?	5

Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-5$ , а $N=59$ ?	4
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-4$ , а $N=-49$ ?	3
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-5$ , а $N=-72$ ?	3
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=8$ , а $N=66$ ?	A
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=5$ , а $N=26$ ?	6
Яким буде результат $X$ обчислення виразу $X=abs(M)+NmodM$ , якщо $M=-8$ , а $N=-45$ ?	3
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=685$ кГц і амплітудою 1 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	C
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=628$ МГц і амплітудою 86 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	1
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=158$ кГц і амплітудою 30 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	7
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=823$ кГц і амплітудою 58 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	3
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=869$ кГц і амплітудою 31 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	4
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=758$ МГц і амплітудою 20 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	0
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=386$ МГц і амплітудою 63 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $Tmod15$ .	5

Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=419$ МГц і амплітудою 33 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	8
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=829$ МГц і амплітудою 75 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	3
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=645$ кГц і амплітудою 98 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	9
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=999$ кГц і амплітудою 80 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	В
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=807$ кГц і амплітудою 15 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	4
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=38$ кГц і амплітудою 48 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	0
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=427$ кГц і амплітудою 76 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	0
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=975$ кГц і амплітудою 99 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	1
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=360$ МГц і амплітудою 38 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	9
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=196$ МГц і амплітудою 74 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	1

Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=338$ МГц і амплітудою 38 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	9
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=746$ МГц і амплітудою 93 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	A
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=25$ кГц і амплітудою 16 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	5
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=211$ МГц і амплітудою 92 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	0
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=628$ кГц і амплітудою 43 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	8
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=853$ кГц і амплітудою 54 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	B
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=682$ кГц і амплітудою 13 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	D
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=844$ кГц і амплітудою 62 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	7
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=860$ кГц і амплітудою 85 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	A
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=149$ МГц і амплітудою 22 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	B

Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=60$ кГц і амплітудою 70 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	8
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=185$ МГц і амплітудою 82 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	3
Яким, за теоремою Котельнікова, в пс, може бути найбільший період $T$ вимірювання синусоїдального сигналу з частотою $F=416$ кГц і амплітудою 6 В? Період $T$ заокруглити до цілого значення в пс, відповідь дати у вигляді $T \bmod 15$ .	3
Яка найменша кількість двійкових розрядів потрібна для кодування 58737 градацій синього кольору?	10
Яка найменша кількість двійкових розрядів потрібна для кодування 23234 градацій синього кольору?	F
Яка найменша кількість двійкових розрядів потрібна для кодування 15360 градацій синього кольору?	E
Яка найменша кількість двійкових розрядів потрібна для кодування 16813 градацій синього кольору?	F
Яка найменша кількість двійкових розрядів потрібна для кодування 51379 градацій синього кольору?	10
Яка найменша кількість двійкових розрядів потрібна для кодування 30577 градацій синього кольору?	F
Яка найменша кількість двійкових розрядів потрібна для кодування 26755 градацій синього кольору?	F
Яка найменша кількість двійкових розрядів потрібна для кодування 19537 градацій синього кольору?	F
Яка найменша кількість двійкових розрядів потрібна для кодування 39426 градацій синього кольору?	10
Яка найменша кількість двійкових розрядів потрібна для кодування 17885 градацій синього кольору?	F
Яка найменша кількість двійкових розрядів потрібна для кодування 459 градацій червоного кольору?	9
Яка найменша кількість двійкових розрядів потрібна для кодування 1124 градацій червоного кольору?	B
Яка найменша кількість двійкових розрядів потрібна для кодування 3249 градацій червоного кольору?	C

Яка найменша кількість двійкових розрядів потрібна для кодування 3687 градацій червоного кольору?	С
Яка найменша кількість двійкових розрядів потрібна для кодування 4006 градацій червоного кольору?	С
Яка найменша кількість двійкових розрядів потрібна для кодування 1330 градацій червоного кольору?	В
Яка найменша кількість двійкових розрядів потрібна для кодування 1348 градацій червоного кольору?	В
Яка найменша кількість двійкових розрядів потрібна для кодування 2636 градацій червоного кольору?	С
Яка найменша кількість двійкових розрядів потрібна для кодування 1364 градацій червоного кольору?	В
Яка найменша кількість двійкових розрядів потрібна для кодування 3348 градацій червоного кольору?	С
Яка найменша кількість двійкових розрядів потрібна для кодування 76 градацій жовтого кольору?	7
Яка найменша кількість двійкових розрядів потрібна для кодування 158 градацій жовтого кольору?	8
Яка найменша кількість двійкових розрядів потрібна для кодування 215 градацій жовтого кольору?	8
Яка найменша кількість двійкових розрядів потрібна для кодування 241 градацій жовтого кольору?	8
Яка найменша кількість двійкових розрядів потрібна для кодування 15 градацій жовтого кольору?	4
Яка найменша кількість двійкових розрядів потрібна для кодування 27 градацій жовтого кольору?	5
Яка найменша кількість двійкових розрядів потрібна для кодування 37 градацій жовтого кольору?	6
Яка найменша кількість двійкових розрядів потрібна для кодування 6 градацій жовтого кольору?	3
Яка найменша кількість двійкових розрядів потрібна для кодування 235 градацій жовтого кольору?	8
Яка найменша кількість двійкових розрядів потрібна для кодування 43 градацій жовтого кольору?	6
Скільки інформації І містить запис 10-кового числа 983К? Відповідь дати у вигляді $I \bmod 15$ .	5



Скільки інформації I містить запис 10-кового числа 77К? Відповідь дати у вигляді I mod 15.	2
Скільки інформації I містить запис 10-кового числа 572М? Відповідь дати у вигляді I mod 15.	0
Скільки інформації I містить запис 10-кового числа 812К? Відповідь дати у вигляді I mod 15.	5
Скільки інформації I містить запис 10-кового числа 249Г? Відповідь дати у вигляді I mod 15.	8
Скільки інформації I містить запис 10-кового числа 778К? Відповідь дати у вигляді I mod 15.	5
Скільки інформації I містить запис 10-кового числа 852Г? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 619Г? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 697 ? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 228К? Відповідь дати у вигляді I mod 15.	3
Скільки інформації I містить запис 10-кового числа 311Г? Відповідь дати у вигляді I mod 15.	9
Скільки інформації I містить запис 10-кового числа 813 ? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 846 ? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 832Г? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 906М? Відповідь дати у вигляді I mod 15.	0
Скільки інформації I містить запис 10-кового числа 560М? Відповідь дати у вигляді I mod 15.	0
Скільки інформації I містить запис 10-кового числа 472 ? Відповідь дати у вигляді I mod 15.	9
Скільки інформації I містить запис 10-кового числа 995Г? Відповідь дати у вигляді I mod 15.	A
Скільки інформації I містить запис 10-кового числа 658 ? Відповідь дати у вигляді I mod 15.	A

Скільки інформації I містить запис 10-кового числа 272K? Відповідь дати у вигляді $I \bmod 15$ .	4
Скільки інформації I містить запис 10-кового числа 268Г? Відповідь дати у вигляді $I \bmod 15$ .	9
Скільки інформації I містить запис 10-кового числа 374K? Відповідь дати у вигляді $I \bmod 15$ .	4
Скільки інформації I містить запис 10-кового числа 410K? Відповідь дати у вигляді $I \bmod 15$ .	4
Скільки інформації I містить запис 10-кового числа 551 ? Відповідь дати у вигляді $I \bmod 15$ .	A
Скільки інформації I містить запис 10-кового числа 415M? Відповідь дати у вигляді $I \bmod 15$ .	E
Скільки інформації I містить запис 10-кового числа 273K? Відповідь дати у вигляді $I \bmod 15$ .	4
Скільки інформації I містить запис 10-кового числа 521Г? Відповідь дати у вигляді $I \bmod 15$ .	A
Скільки інформації I містить запис 10-кового числа 629Г? Відповідь дати у вигляді $I \bmod 15$ .	A
Скільки інформації I містить запис 10-кового числа 1023K? Відповідь дати у вигляді $I \bmod 15$ .	5
Скільки інформації I містить запис 10-кового числа 849 ? Відповідь дати у вигляді $I \bmod 15$ .	A
Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 167 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	4
Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 109 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	4
Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 18 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 206 ПЗП, який використовується для обчислення результату	5

	$S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 163 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	3
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 192 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	0
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 117 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 53 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 17 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	1
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 119 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	4
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 199 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	4
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 92 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких	3

	$m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 181 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 169 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	3
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 107 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	5
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 166 ПЗП, який використовується для обчислення результату $S = 1M + 1N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	3
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 85 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	3
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 50 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 79 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	9
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 99 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди	3

	- молодші	
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 102 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	4
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 10 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	6
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 163 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	3
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 212 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 31 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	9
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 33 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	1
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 172 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	6
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 84 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди	2

	- молодші	
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 2 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	2
	Визначити і підкреслити 16-ий код, що буде читатися з десяткової адреси 128 ПЗП, який використовується для обчислення результату $S = 1M + 2N$ , де $M$ і $N$ - дворозрядні двійкові числа, розряди яких $m_0, m_1, n_0, n_1$ заведені на адресні входи $a_0, \dots, a_3$ ПЗП і 0-ві розряди - молодші	0
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 1, якщо на вхід переносу подається 0	2
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 2, якщо на вхід переносу подається 0	4
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 3, якщо на вхід переносу подається 0	6
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 4, якщо на вхід переносу подається 0	8
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 5, якщо на вхід переносу подається 0	A
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 6, якщо на вхід переносу подається 0	C
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 7, якщо на вхід переносу подається 0	E
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 8, якщо на вхід переносу подається 0	0
	Визначити і підкреслити, яким буде 16-ковий результат логічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 9, якщо на вхід переносу подається 0	2

























	Визначити і підкреслити, яким буде 16-ковий результат циклічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 14	D
	Визначити і підкреслити, яким буде 16-ковий результат циклічного зсуву ліворуч 4-бітного числа, 10-ковий код якого дорівнює 15	F

... Далі буде.

## 12 КОРОТКИЙ ТЛУМАЧЕВИЙ СЛОВНИК

Алгоритм	Скінчена послідовність команд, направлених на розв'язання деякого завдання.
Алфавіт	Впорядкований набір символів деякої природної мови.
Алфавіт мови програмування	Символьний набір, заданий описом конкретної мови програмування для подання вихідних програм та даних.
Байт	Група з 8 біт, використовується як міра інформації.
Біт	Двійкова одиниця кількості інформації, яка відповідає тій інформації, яка одержується при прийомі одного з двох рівно імовірних повідомлень.
Буфер	Поле пам'яті або пристрій, що використовується для узгодження між швидкостями обміну, розмірами блоків даних та моментами виникнення подій при обміні даними.
Величина	Об'єкт, з яким пов'язується певна множина значень;
Вираз	Формульний запис в програмі, що визначає операції над даними, виконуючи які, процесор одержує значення; правила запису виразів визначаються конкретною мовою програмування.
Дані	Інформація, подана у вигляді, зручному для формального опрацювання автоматичними пристроями або людиною.
Діаграма	Графічне зображення співвідношення між величинами, об'єктами.
Довжина рядка	Кількість символів, які дійсно містяться в рядковій величині на час її опрацювання.
Елемент	Складова частина складного цілого.
ЕОМ	Електронно-обчислювальна машина (див. Комп'ютер) – цифрова обчислювальна машина, основні вузли якої реалізовані засобами електроніки.
Залежність	Відношення між функцією та її аргументами.
Запам'ятовуючий пристрій	Пристрій для фіксації даних таким чином, щоб вони потім могли бути автоматично відтворені.
Змінна	Тріада: об'єкт, що може змінювати значення в ході виконання програми; ім'я змінної; значення змінної.
Значення	Смисл, який приписується об'єкту людиною.
Інформатика	Технічна наука, що систематизує прийоми створення збереження, накопичення, опрацювання та передачі даних засобами обчислювальної техніки, а також принципи функціонування цих засобів і методи керування ними.

Інформація	Продукт взаємодії даних та методів, який розглядається в контексті цієї взаємодії
Кілобайт	Одиниця виміру інформації, містить 1024 байт.
Код	Набір знаків в сукупності зі схемою кодування для подання інформації у вигляді даних.
Кодування даних	Процес перетворення даних до єдиної форми подання
Комп'ютер	Універсальний електронний пристрій, призначений для автоматизації накопичення, збереження, опрацювання, передачі та відтворення даних
Константа	Величина, що не змінює свого значення за час виконання програми.
Контролер	Пристрій, призначений для керування зовнішніми пристроями
Лінійний алгоритм	Алгоритм, в якому оператори виконуються в порядку їх слідування.
Масив	Структура даних, що являє собою впорядкований набір однотипних даних.
Мегабайт	Одиниця виміру інформації, містить 1024 кілобайти або приблизно 1000000 байт.
Множина	Сукупність деяких об'єктів, які мають характерну властивість для всіх об'єктів.
Модем (модулятор – демодулятор)	Пристрій для перетворення даних для передачі їх по певним лініям зв'язку.
Обчислювальна техніка	Сукупність пристроїв, призначених для автоматизації опрацювання даних.
Обчислювальна система	Конкретний набір програмно-апаратних пристроїв, призначених для виконання певного класу завдань
Операнд	Аргумент операції.
Оперативна пам'ять	Набір мікросхем, призначених для зберігання даних під час їх безпосереднього опрацювання;
Оператор	Грамотична конструкція, визначення якої уточнюється конкретною мовою програмування і яка виражає певну закінчену послідовність машинних операцій; інтерпретується як вказівка виконати ці команди;
Оператор вводу	Оператор, призначений для вводу даних з клавіатури або файлу.
Оператор виводу	Оператор, призначений для виводу даних на екран дисплею, принтер або у файл.
Оператор	Оператор, що обчислює значення виразу та присвоює

присвоювання	його змінній.
Оператор умовного переходу	Оператор, що змінює природну послідовність виконання операторів в залежності від наведеної в операторі умови.
Оператор безумовного переходу	Оператор, що дає значенням мітку наступного опертора, що має виконуватись.
Оператор вибору	Оператор з переліком альтернатив, в яких задані дії відповідно до наведених умов.
Оператор переходу	Оператор, що змінює природну послідовність вибірки операторів програми при її виконанні.
Оператор присвоювання	Оператор, в результаті виконання якого змінній надається значення відповідно до заданого виразу.
Оператор циклу	Оператор, що визначає тіло циклу, параметри циклу та/або умови повтору виконання тіла циклу.
Операція	Деяка закінчена послідовність дій людини або машини.
Опис	Декларативне речення мови програмування, що оголошує об'єкти, їх імена, типи та області існування.
Організація	Внутрішня впорядкованість, узгодженість та взаємодія частин цілого, обумовлена його будовою.
Пам'ять	Загальна назва для будь-яких засобів фіксації та збереження інформації, довільного реального або абстрактного механізму, що забезпечує можливість фіксувати сигнали.
Параметр	Змінна величина, від якої залежать значення інших величин.
Постійна пам'ять	Мікросхема, яка призначена для постійного зберігання інформації, у тому числі і при вимкненому живленні
Пристрій	Сукупність деталей, вузлів, елементів, якій властива конструктивна та функціональна єдність.
Процесор (центральний процесор)	Пристрій опрацювання даних в комп'ютері; виконує функції арифметичної та логічної обробки даних та загального керування роботою інших пристроїв комп'ютера .
Розгалужений алгоритм	Алгоритм, у якому здійснюється вибір між декількома групами операторів.
Символ	Літерний ланцюжок з одного або декількох літер для позначення чого-небудь.
Система	1. Сукупність абстрактних або матеріальних об'єктів разом з відомими або заданими властивостями і відношеннями, які утворюють в певному сенсі єдине ціле.

	2. Сукупність матеріальних або абстрактних органів, пов'язаних загальною функцією, метою, призначенням.
Система числення	Спосіб вираження та позначення чисел.
Структура	1. Сукупність стійких зв'язків між частинами об'єкта, що забезпечують його цілісність та тотожність самому собі, тобто, збереження його характеристичних властивостей при внутрішніх та зовнішніх змінах. 2. Екземпляр моделі даних.
Циклічний алгоритм	Алгоритм, що являє собою послідовність операторів, яка повинна виконатись декілька раз.
Швидкодія	Параметр, що характеризує продуктивність обчислювальної системи кількістю операцій, які виконуються за певний проміжок часу.
Шина	Набір провідників для передачі даних та сигналів керування.

## БІБЛІОТЕЧНИЙ СПИСОК

Основна використана література.

- ГОСТ 2.743-91. Обозначения условные графические в схемах. - М.: 1991.
- Баранов С.И. Синтез микропрограммных автоматов. Л.: Энергия. 1979.
- Бровков В.Г., Богатова О.О. Конспект лекцій з курсу “ПТЦА” Одеса ОНПУ 2001
- Будинский Я. Логические цепи в цифровой технике. SNTL. Praha. 1977.
- Глушков В.М. Синтез цифровых автоматов. — М.: ГИФМЛ, 1962. — 476 с.
- Ерош И.Л. Дискретная математика. Булева алгебра, комбинационные схемы, преобразования двоичных последовательностей: Учебное пособие. - СПб.: ГУАП, 2001. - 30 с.
- Жабін В. І., Жуков І. А., Клименко І.А., Ткаченко В.В. «Прикладна теорія цифрових автоматів». НАУ-Друк. 2009
- Н. Г. Захаров, В. Н. Рогов. Синтез цифровых автоматов: Учебное пособие / Н. Г. Захаров, В. Н. Рогов. –Ульяновск: УлГТУ, 2003.
- Зілінський Ю.В. Конспект лекцій з навчальної дисципліни "Обчислювальна техніка" для студентів денної форми навчання за напрямом 6.050102 - "Комп'ютерна інженерія". Кременчуцький Національний університет ім. М.Остроградського. Кременчук 2010.
- Карцев М.А. Арифметика цифровых машин. 1969.
- Лысиков Е.Г. Арифметические и логические основы цифровых автоматов: Учебник для вузов по специальности ЭВМ. - 2-е изд., перераб. и доп.- Минск: Вышэйшая школа, 1980.
- Матвієнко М. П. Комп'ютерна логіка. Навчальний посібник. — К.:Видавництво Ліра-К, 2012. — 288 с.
- Миллер Р.Теория переключательных схем.1970.Т1.Комбинационные схемы.
- Миллер Р.Теория переключательных схем.1971.Т2.Последовательностные схемы и машины.
- Савельев А.Я. Прикладная теория цифровых автоматов: Учебник для вузов по специальности ЭВМ. - М.: Высшая школа, 1987.
- Поспелов Д.А.Арифметические основы вычислительных машин дискретного

действия.1970.

Савельев А.Я. Арифметические и логические основы цифровых автоматов. 1980.

Фридман А. Теория и проектирование переключательных схем. 1978.

Шауман А.М. Основы машинной арифметики. 1979.

Arithmetic and Logic in Computer Systems. Mi Lu. Texas A&M University. Copyright © 2004 by John Wiley & Sons, Inc. All rights reserved.

Digital Arithmetic, M.D. Ercegovac and T. Lang, Morgan Kaufmann Publishers - An Imprint of Elsevier Science, 2004

DIGITAL DESIGN by M.MORRIS MANO

Дунець Р.Б., Кудрявцев О.Т. Арифметичні основи комп'ютерної техніки. Навчальний посібник. Інститут підприємництва та перспективних технологій при Національному університеті «Львівська політехніка». Львів. 2006. – 142 с.

Лукащук Л.О. Схемотехніка логічних та послідовнісних схем. – Львів. Видавництво Національного університету «Львівська політехніка», 2004. – 116 с.

“Прикладная теория цифровых автоматов” К.Г. Самофалов, Ю.С. Каневский, М.М. Пиневиц: Учебник для вузов Украины по спец. ЭВМ.-К.:Вища школа, 1987. 375 с.

“Цифровые ЭВМ: Теория и проектирование” Самофалов К.Г., Корнейчук В.И., Тарасенко В.П.: Учебник для вузов Украины по спец. “Вычислительные машины, комплексы, системы и сети”. 3-е изд. Вища школа, 1989.

Мельник А.О. Архітектура комп'ютера. Підручник. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.

“Цифрова техніка” Рицар Б.Є.: Учбовий посібник.-К.: НМК ВО, 1991.

“Теория информации и кодирования” Цимбал В.П.: Учебник.-К.: Вища школа, 1992.

“Основы теории информации и кодирования” Кузьмин И.Б., Кедрус В.А.: Учебник для вузов по спец. “Автоматика и телемеханика” и “Прикладная математика” 2-е изд. Вища школа, 1986.

“Цифровые устройства на программируемых БИС с матричной структурой” Баранов С.И., Складов В.А.: -М: Радио и связь, 1986.

В.С. Князьков, Т.В. Волченская. Введение в теорию автоматов.  
<http://www.intuit.ru/department/algorithms/intavth/>

В.О. Чуканов, В.В. Гуров. Логические и арифметические основы и принципы работы ЭВМ. <http://www.intuit.ru/department/hardware/archsys/>

Ю.В. Новиков. Введение в цифровую схемотехнику. <http://www.intuit.ru/department/hardware/digs/>

М.И. Дехтярь. Введение в схемы, автоматы и алгоритмы. <http://www.intuit.ru/department/ds/introsaa/>

Література, рекомендована для поглибленого вивчення курсу.

Методичні вказівки до курсової роботи “Проектування цифрових структур” з дисципліни "Прикладна теорія цифрових автоматів" для студентів бакалаврату 6.0915 "Комп'ютерна інженерія", видання друге, / Укл. В.С.Глухов, В.А.Голембо. Львів: НУ"ЛП", 2003.-109 с.

Дунець Р.Б., Кудрявцев О.Т. Арифметичні основи комп'ютерної техніки. Навчальний посібник. Інститут підприємництва та перспективних технологій при Національному університеті «Львівська політехніка». Львів. 2006. – 142 с.

Лукащук Л.О. Схемотехніка логічних та послідовнісних схем. – Львів. Видавництво Національного університету «Львівська політехніка», 2004. – 116 с.

Рицар Б.Є. Цифрова техніка: Учбовий посібник для студентів радіотехнічних спеціальностей - К.: НМК ВО, 1991.

Прикладная теория цифровых автоматов /К.Г.Самофалов, А.М.Романкевич, В.Н.Валуйский, Ю.С.Каневский, М.М.Пиневиц: Учебник для вузов Украины по специальности ЭВМ.-К.: Вища шк. Головное изд-во, 1987.

Савельев А.Я. Прикладная теория цифровых автоматов: Учебник для вузов по специальности ЭВМ. - М.: Высшая школа, 1987.

Лысыков Е.Г. Арифметические и логические основы цифровых автоматов: Учебник для вузов по специальности ЭВМ. - 2-е изд., перераб. и доп.- Минск: Вышэйшая школа, 1980.

Мельник А.О. Архітектура комп'ютера. Підручник. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.

Самофалов К.Г., Корнейчук В.И., Тарасенко В.П. Цифровые ЭВМ: Теория и проектирование : Учебник для вузов по специальности "Вычислительные машины,



комплексы, системы и сети" / Под общ. ред. К.Г.Самофалова - 3-е изд. перераб. и дополн. - К.: Выща школа, Главное изд-во, 1989.

Угрюмов Е.П. Проектирование элементов и узлов ЭВМ.- М.: Высшая школа, 1987.

Скаржепа В.А. Сенько В.И. Электроника и микросхемотехника. ч.1. Электронные устройства информационной автоматики /Под общ. ред. А.А.Краснопрошиной - К.: Выща школа, Головн. изд-во, 1989.

Скаржепа В.А., Луценко А.Н. Электроника и микросхемотехника. ч.3. Сборник задач /Под общ. ред. А.А.Краснопрошиной - К.: Выща школа, Головн. изд-во, 1989.

Арсеньев Ю.Н. Журавлев В.М. Проектирование систем логического управления на микропроцессорных средствах: Учебное пособие для вузов по спец. "Вычисл. машины, комплексы, системы и сети". - М.: Высшая шк., 1991.

Преснухин Л.Н., Воробьев Н.В., Шишкевич А.А. Расчет элементов цифровых устройств: Учебн. пособие / Под ред. Л.Н.Преснухина - 2-е изд. перераб. и доп. - М.: Высшая шк. 1991.

Лихтциндер Б.Я., Кузнецов В.Н. Микропроцессоры и вычислительные устройства в радиотехнике: Учебное пособие.-К.:Выща шк. Головное изд-во, 1988.

Электронные вычислительные машины. Лабораторный практикум: Учебное пособие /Под ред. Г.Н.Соловьева - М.: Высшая школа, 1987.

ГОСТ 2.743-91. Обозначения условные графические в схемах. - М.: 1991.

Справочник по Единой системе конструкторской документации. Под редакцией Ю.И.Степанова. Издание третье, переработанное и дополненное. Харьков «Прапор» 1981.

Графическое изображение электрорадиосхем: Справочник / Усатенко С.Т., Каченюк Т.К., Терехова М.В. – К.: Техніка, 1986. – 120 с., ил.

В.С. Князьков, Т.В. Волченская. Введение в теорию автоматов.  
<http://www.intuit.ru/department/algorithms/intavth/>

В.О. Чуканов, В.В. Гуров. Логические и арифметические основы и принципы работы ЭВМ. <http://www.intuit.ru/department/hardware/archsys/>

Ю.В. Новиков. Введение в цифровую схемотехнику.  
<http://www.intuit.ru/department/hardware/digs/>

М.И. Дехтярь. Введение в схемы, автоматы и алгоритмы.

<http://www.intuit.ru/department/ds/introsaa/>

Література до курсового проектування

Методичні вказівки до курсової роботи “Проектування цифрових структур” з дисципліни "Прикладна теорія цифрових автоматів" для студентів бакалаврату 6.0915 "Комп'ютерна інженерія", видання друге, /Укл. В.С.Глухов, В.А.Голембо. Львів: НУ"ЛП", 2003.-109

Мельник А.О. Архітектура комп'ютера. Підручник. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.

Дунець Р.Б., Кудрявцев О.Т. Арифметичні основи комп'ютерної техніки. Навчальний посібник. Інститут підприємництва та перспективних технологій при Національному університеті «Львівська політехніка». Львів. 2006. – 142 с.

Лукащук Л.О. Схемотехніка логічних та послідовнісних схем. – Львів. Видавництво Національного університету «Львівська політехніка», 2004. – 116 с.

Рицар Б.Є. Цифрова техніка: Учбовий посібник для студентів радіотехнічних спеціальностей - К.: НМК ВО, 1991.

ГОСТ 2.743-91. Обозначения условные графические в схемах. - М.: 1991.

Справочник по Единой системе конструкторской документации. Под редакцией Ю.И.Степанова. Издание третье, переработанное и дополненное. Харьков «Прапор» 1981.

Графическое изображение электрорадиосхем: Справочник / Усатенко С.Т., Каченюк Т.К., Терехова М.В. – К.: Техніка, 1986. – 120 с., ил.

В.С. Князьков, Т.В. Волченская. Введение в теорию автоматов. <http://www.intuit.ru/department/algorithms/intavth/>

В.О. Чуканов, В.В. Гуров. Логические и арифметические основы и принципы работы ЭВМ. <http://www.intuit.ru/department/hardware/archsys/>

Ю.В. Новиков. Введение в цифровую схемотехнику. <http://www.intuit.ru/department/hardware/digs/>

М.И. Дехтярь. Введение в схемы, автоматы и алгоритмы. <http://www.intuit.ru/department/ds/introsaa/>

Література до теоретичного курсу.

1. «Логіка роботи комбінаційних цифрових вузлів». Навчальний посібник з дисципліни «Комп'ютерна логіка» для студентів усіх форм навчання напряму 6.050102 «Комп'ютерна інженерія», обсяг 102 стор, укладач — д.т.н. Глухов В.С., доц. каф. ЕОМ. Львів: НУЛП, 2014.

2. Дунець Р.Б., Кудрявцев О.Т. Арифметичні основи комп'ютерної техніки. Навчальний посібник. Інститут підприємництва та перспективних технологій при Національному університеті «Львівська політехніка». Львів. 2006. – 142 с.

3. Лукащук Л.О. Схемотехніка логічних та послідовнісних схем. – Львів. Видавництво Національного університету «Львівська політехніка», 2004. – 116 с.

Література до практичних (семінарських) занять.

4. Методичні вказівки «Приклади використання теорії цифрових автоматів та комп'ютерної логіки. Послідовнісні схеми» до практичних занять з дисциплін «Прикладна теорія цифрових автоматів» та «Комп'ютерна логіка» для студентів бакалаврату 6.050102 «Комп'ютерна інженерія» /Укл. В.С.Глухов. Львів: НУ"ЛП", 2010.

5. Методичні вказівки до практичних занять «Логіка роботи і структури цифрових вузлів» з дисципліни «Комп'ютерна логіка» для студентів усіх форм навчання напряму 6.050102 «Комп'ютерна інженерія», обсяг 20 стор, укладач — д.т.н. Глухов В.С., доц. каф. ЕОМ. Львів: НУ"ЛП", 2014.

Література до виконання РГР

6. Методичні вказівки до курсової роботи “Арифметичні та логічні основи комп'ютерних технологій” з дисципліни "Комп'ютерна логіка" базового напрямку 6.050102 "Комп'ютерна інженерія" /Укл. В.С.Глухов, В.А.Голембо. Львів: НУ"ЛП", 2014.-96 с.

7. «Логіка роботи комбінаційних цифрових вузлів». Навчальний посібник з дисципліни «Комп'ютерна логіка» для студентів усіх форм навчання напряму 6.050102 «Комп'ютерна інженерія», обсяг 102 стор, укладач — д.т.н. Глухов В.С., доц. каф. ЕОМ. Львів: НУЛП, 2014.

### 13 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

#### Базова

8. Матвієнко М. П. Комп'ютерна логіка. Навчальний посібник. — К.: Видавництво Ліра-К, 2012. — 288 с.

#### Допоміжна

9. Мельник А.О. Архітектура комп'ютера. Підручник. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.

10. ГОСТ 2.743-91. Обозначения условные графические в схемах. - М.: 1991.

11. Справочник по Единой системе конструкторской документации. Под редакцией Ю.И.Степанова. Издание третье, переработанное и дополненное. Харьков «Прапор» 1981.

12. Графическое изображение электрорадиосхем: Справочник / Усатенко С.Т., Каченюк Т.К., Терехова М.В. – К.: Техніка, 1986. – 120 с., ил.

## 14 ІНФОРМАЦІЙНІ РЕСУРСИ

13. Комп'ютерна логіка. <https://vns.lpnu.ua/course/view.php?id=4592>
14. В.С. Князьков, Т.В. Волченская. Введение в теорию автоматов. <http://www.intuit.ru/department/algorithms/intavth/>
15. В.О. Чуканов, В.В. Гуров. Логические и арифметические основы и принципы работы ЭВМ. <http://www.intuit.ru/department/hardware/archsys/>
16. Ю.В. Новиков. Введение в цифровую схемотехнику. <http://www.intuit.ru/department/hardware/digs/>
17. М.И. Дехтярь. Введение в схемы, автоматы и алгоритмы. <http://www.intuit.ru/department/ds/introsaa/>

## ДОДАТОК А ГРЕЦЬКА АБЕТКА

Грецькі літери, великі	Грецькі літери, малі	Назва літери, Українською мовою
Α	α	Альфа
Β	β	Бета
Γ	γ	Гамма
Δ	δ	Дельта
Ε	ε	Епсилон
Ζ	ζ	Дзета
Η	η	Ета
Θ	θ	Тета
Ι	ι	Йота
Κ	κ	Каппа
Λ	λ	Ламбда
Μ	μ	Мі
Ν	ν	Ні
Ξ	ξ	Ксі
Ο	ο	Омікрон
Π	π	Пі
Ρ	ρ	Ро
Σ	σς	Сигма
Τ	τ	Тау
Υ	υ	Іпсилон
Φ	φ	Фі
Χ	χ	Хі
Ψ	ψ	Псі
Ω	ω	Омега

## НАВЧАЛЬНЕ ВИДАННЯ

«Комп'ютерна логіка комбінаційних схем».

Навчальний посібник

з дисципліни «Комп'ютерна логіка» для студентів першого (бакалаврського) рівня  
вищої освіти спеціальності 123 «Комп'ютерна інженерія»

Укладач проф. каф. ЕОМ, д.т.н., проф. Глухов В. С.,

Редактор

Комп'ютерне складання

Підписано до друку \_\_.\_\_.\_\_\_\_ р.

Формат 70 x 100 <sup>1</sup>/<sub>16</sub>. Папір офсетний.

Друк на різнографі. Умовн. друк. арк. \_\_ Обл.-вид. арк. \_\_\_\_.

Наклад \_\_\_\_ прим. Зам. \_\_\_\_.

Поліграфічний центр

Видавництва Національного університету “Львівська політехніка”

Вул. Колесси, 2, 79000, Львів