МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ТЕОРІЯ ІНФОРМАЦІЇ ТА КОДУВАННЯ В ПРИКЛАДАХ І ЗАДАЧАХ

А.В. Івашко, В.А. Крилова

Навчальний посібник

для студентів спеціальностей «Автоматизація
та комп'ютерно-інтегровані технології»,
«Телекомунікація та радіотехніка»
усіх форм навчання вищих навчальних закладів

Рекомендовано
редакційно–видавничою
радою університету,
протокол № 3 від 26.10.2022

Харків 2022

І 19
ББК 32.988-5
УДК 681.5 (004)

Рецензенти:

М.А. Мірошник, д-р техн. наук, проф. «Харківський національний університет ім. В.Н. Каразіна» м. Харків

А.А. Коваленко, д-р техн. наук, проф. «Харківський національний університет радіоелектроніки» м. Харків

Івашко А.В.

І 19 Івашко А.В., Крилова В.А. Теорія інформації та кодування в прикладах і задачах: навч.-метод. посіб. Харків : НТУ «ХПІ», 2022. 317 с.

У навчальному посібнику викладено основні поняття та положення теорії інформації, визначення кількості інформації та ентропії повідомлення, основні принципи ефективного кодування та стиснення даних. Розглянуто методи завадостійкого кодування, алгоритми кодування та декодування циклічними, БЧХ та Ріда-Соломона кодами. Наведені приклади та розв’язання типових задач, завдання для самоконтролю, що сприяє кращому засвоєнню матеріалу.

Призначено для студентів денної та заочної форм навчання вищих навчальних закладів, які вивчають теорію інформації та кодування.

Табл.18. Іл.23. Бібліогр. 22

ББК 32.988-5
УДК 681.5 (004)

ISBN 978-617-8072-62-9

ЗМІСТ

ПЕРЕДМОВА

Теорією інформації та кодування називають науку, що вивчає кількісні закономірності, пов'язані з отриманням, передачею, обробкою та зберіганням інформації. Теорія інформації – наука з відомою датою народження. Її появу на світ зазвичай пов'язують із публікацією роботи Клода Шеннона «Математична теорія зв'язку» (1948), що містить ряд важливих теоретичних результатів, які на багато років вперед визначили принципи побудови систем зв'язку. Однак, виникнувши з суто зв'язних завдань, теорія інформації та кодування в даний час є не тільки теоретичною основою, а й прикладним математичним апаратом при проектуванні систем управління, передачі та обробки даних.

До найбільш актуальних завдань теорії інформації належить відшукання найбільш економних методів кодування, що дозволяють передавати заданий обсяг інформації за допомогою мінімального числа символів як за наявності перешкод, так і за їх відсутності. З іншого боку, розробка на основі методів теорії інформації ефективних кодів, що виправляють помилки, присутні під час передачі та зберігання даних дозволило створити сучасні системи мобільної телефонії, комп'ютерні мережі, системи супутникового зв’язку та навігації.

Важливість зазначених завдань обумовила той факт, що дисципліна «Теорія інформації та кодування» є однією з основних ІТ-дисциплін, вивчаємих студентами, що спеціалізуються на вивченні автоматизованих систем, програмно-технічних комплексів, засобів телекомунікацій та радіотехніки. Однак, незважаючи на наявність значної кількості підручників та навчальних посібників з теорії інформації, попит на літературу з цієї проблематики повністю не задоволений. Особливо затребуваними виявляються збірники прикладів та завдань з теорії інформації та кодування, що дозволяють спростити застосування теоретичних методів до завдань, що виникають на практиці. Частково ліквідувати цю лакуну дозволяє пропонована читачеві книга.

Матеріал книги поділено на шість розділів. У кожному розділі крім прикладів викладається необхідний мінімум теоретичних відомостей. Докладніше теоретичний матеріал наводиться, якщо у загальнодоступній літературі він висвітлений недостатньо або мало доступно

Зауваження та пропозиції щодо змісту книги автори просять надсилати на адресу Viktoriia.Krylova@khpi.edu.ua.

ГЛАВА 1

МАТЕМАТИЧНІ ОСНОВИ ТЕОРІЇ ІНФОРМАЦІЇ

1.1 Подання числової інформації в різних системах числення

Кодування інформації – це її представлення у формі, зручної для зберігання, обробки і передачі каналами зв'язку. При цьому широкий розвиток цифрових технологій обумовив представлення повідомлень як множини обраних із заданого набору (алфавіту) дискретних символів.

Для однозначного представлення чисел слід попередньо вибрати ту чи іншу *систему числення* – сукупність правил для запису чисел дискретними символами. Вибрана система числення повинна забезпечувати виконання таких вимог:

– можливість представлення будь-якого числа у заданому діапазоні;

– однозначність представлення;

– стисліcть запису чисел.

Системи числення діляться на *непозиційні* та *позиційні*.

Непозиційними називаються такі системи числення, в котрих кількісний еквівалент будь-якого символу постійний і залежить тільки від його значення, але не від позиції в числі. Прикладом непозиційної системи числення є римська система. У римській системі числення використовуються такі символи: I – одиниця, V – п'ять, X – десять, L – п'ятдесят, C – сто, D – п'ятсот, M – тисяча. При цьому, якщо більша цифра стоїть перед меншою, то вони складаються (принцип додавання), якщо ж менша стоїть перед більшою, то менша віднімається від більшої (принцип віднімання).

Приклад 1.1. Перевести число MCDXCII з римської системи в десяткову.

Рішення. Складаємо значення символів числа в порядку зменшення: M= 1000, D = 500, C = 100, двічі I = 1. Віднімаємо символи з меншими значеннями, що стоять перед символами з більшими значеннями: C = 100, X = 10. Тоді A = 1000+500+100+1+1–100–10=1492. ■

Приклад 1.2. Перевести число 1948 з десяткової системи в римську.

Рішення. Числу 1000 відповідає римський знак M, число 900 представляється як 1000 – 100=CM, число 40 як 50 – 10 = XL, число

8 = 5+3=5+1+1+1 = VIII. Об'єднуючи символи, отримуємо 1948 = MCMXLVIII. ■

Непозиційні системи незручні для запису як малих, так і великих чисел, а також для виконання арифметичних операцій, тому найчастіше застосовуються позиційні системи числення.

Позиційними називаються такі системи, алфавіт яких містить обмежену кількість символів, кількісний еквівалент котрих залежить не тільки від їх значення, а й від позиції в числі.

Число A у позиційній системі може бути представлене у вигляді

$$A = a_n p_n + a_{n-1} p_{n-1} + \ldots + a_i p_i + \ldots + a_2 p_2 + a_1 p_1, \qquad (1.1)$$

де p_i – вага розряду, a_i – цифра i-го розряду числа, $a_i < p_i$. Позиційні системи, у свою чергу, поділяються на *неоднорідні* і *однорідні* системи числення. У неоднорідних (змішаних) системах числення p_i не залежать один від одного і можуть приймати будь-які значення. Прикладом неоднорідної позиційної системи числення може служити система вимірювання часу, для якої p_1=1 с, p_2=60 с, p_3=60 мин=60·60=3600с. Наприклад, час у 17 годин, 23 хвилини та 12 секунд, виражений в одиницях молодшого розряду визначається з (1.1) як A=17·3600+23·60+12·1 = 62592с.

Іншим прикладом неоднорідної позиційної системи числення є так звана Фібоначчієва система (на ім'я середньовічного італійського математика Леонардо Пізанського, відомого на прізвисько Фібоначчі). Фібоначчієва система числення застосовується при побудові коригуючих кодів, для стиснення інформації, у високонадійних аналого-цифрових перетворювачах та в інших галузях обробки та передачі. Як ваги розрядів у Фібоначчієвій системі числення використовуються числа Фібоначчі, що визначаються лінійним рекурентним співвідношенням

$$F_0=0,\ F_1=1,\ F_i=F_{i-2}+F_{i-1}. \qquad (1.2)$$

Наприклад, $F_2=F_0+F_1=0+1=1$, $F_3=F_1+F_2=1+1=2$, $F_4=F_2+F_3=1+2=3$, $F_5=F_3+F_4=2+3=5$ і т.д.

Таким чином, формується числова послідовність

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610…

Як ваги розрядів в (1.2) обираються значення чисел Фібоначчі $p_i=F_{i+1}$, коефіцієнти подання a_i приймають значення 0 або 1, а число представляється як

$$A = a_n F_{n+1} + a_{n-1} F_n + \ldots + a_i F_{i+1} + \ldots + a_2 F_3 + a_1 F_2 . \qquad (1.3)$$

Оскільки при такому поданні відповідно до (1.2) можлива неоднозначність, то дві одиниці, що стоять поруч, замінюються одиницею в наступному за старшинством розряді. Отже, алгоритм переведення цілого числа A до Фібоначчієвої системи числення можна сформулювати наступним чином:

1. Знайдемо максимальне число Фібоначчі F_m, що не перевищує число A, і зафіксуємо його зменшений на одиницю номер $n = m - 1$. Шуканий запис числа A в системі Фібоначчі буде містити n розрядів. У першому наближенні Фібоначчієвого представлення числа значення старшого розряду a_n=1, значення решти розрядів $a_1 \ldots a_{n-1}$ =0.

2. Обчислимо різницю $A = A - F$.

3. Якщо отримане значення A дорівнює нулю, то перехід до п. 6.

4. Знову знайдемо максимальне число Фібоначчі F_j, яке не перевищує число A, і зафіксуємо його зменшений на одиницю номер $k=j-1$. Надамо a_k значення 1.

5. Перехід до п. 2.

6. Читання числа $A = \{a_n, a_{n-1}, \ldots, a_2, a_1\}$. Кінець алгоритму.

Приклад 1.3. Перевести число $A_{\text{фіб}}$=1001010001001 з Фібоначчієвої системи до десяткової.

Рішення. Число розрядів у перетворюваному числі n=13. Згідно (1.3)

$A_{\text{фіб}} = a_{13}F_{14} + a_{12} F_{13} + a_{11} F_{12} + \ldots + a_3 F_4 + a_2 F_3 + a_1 F_2 =$

$=1\cdot377+0\cdot233+0\cdot144+1\cdot89+0\cdot55+1\cdot34+0\cdot21+0\cdot13+0\cdot8+1\cdot5+0\cdot3+0\cdot2+1\cdot1 = 506.$

■

Приклад 1.4. Перевести число A=209 з десяткової системи в Фібоначчієву.

Рішення. Знаходимо максимальне число Фібоначчі F, що не перевищує число A = 209 і зафіксуємо його зменшений на одиницю номер n. Тоді $F = F_{12} = 144$, $n = 12 - 1 = 11$. Надамо a_{11}=1, $a_1 \ldots a_{10}$=0. Запишемо перше наближення Фібоначчієвого представлення $A_{\text{фіб1}}$ = 10000000000.

Обчислимо різницю $A = A - F = 209 - 144 = 65$.

Знайдемо максимальне число Фібоначчі F, яке не перевищує число A, і зафіксуємо його зменшений на одиницю номер k. Тоді $F = F_{10} = 55$, n=10–1=9. Надамо a_9=1. Друге наближення Фібоначчієвого представлення $A_{\text{фіб2}}$=10100000000.

Обчислимо різницю $A = A - F = 65 - 55 = 10$.

Знайдемо максимальне число Фібоначчі F, що не перевищує число A, і зафіксуємо його зменшений на одиницю номер k. Тоді $F = F_6 = 8$, $k = 6 - 1 = 5$. Надамо a_5=1. Третє наближення Фібоначчієвого представлення $A_{\text{фіб3}}$= 10100010000.

Обчислимо різницю $A = A - F = 10 - 8 = 2$.

Знайдемо максимальне число Фібоначчі F, що не перевищує число A, і зафіксуємо його зменшений на одиницю номер k. Тоді $F = F_3 = 2$, k=3– 1 = 2. Надамо $a_3 = 1$. Четверте наближення Фібоначчієва представлення $A_{\text{фіб3}}$=10100010010.

Обчислимо різницю $A = A - F = 2 - 2 = 0$.

Оскільки отримане значення A дорівнює нулю, переходимо на кінець алгоритму та записуємо відповідь: $A_{\text{фіб}} = 10100010010$. ■

У однорідних позиційних системах числення ваги розрядів є ступенями деякого числа p, званого *основою системи числення*. Таким чином, число подається у вигляді

$$A = a_n p^n + a_{n-1} p^{n-1} + \ldots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + \ldots + a_{-(m-1)} p^{-(m-1)} + a_{-m} p^{-m}. \quad (1.4)$$

У випадку, якщо A – ціле, сума (1.4) містить тільки додатні ступеня, тобто $m = 0$. Якщо A – правильний дріб, який не містить цілої частини, сума (1.4) містить лише від'ємні ступені, тобто $n = -1$.

При обробці інформації часто буває необхідно перетворити число з однієї однорідної позиційної системи числення в іншу. Існує універсальний алгоритм перекладу, придатний для будь-якої основи p. У системі з основою p ціле число подається у вигляді

$$A = a_n p^n + a_{n-1} p^{n-1} + a_{n-2} p^{n-2} + \ldots + a_2 p^2 + a_1 p^1 + p_0 . \quad (1.5)$$

Для знаходження невідомих коефіцієнтів a_0 ... a_n розкладемо (1.5) за схемою Горнера

$$A = p(a_n p^{n-1} + a_{n-1} p^{n-2} + a_{n-2} p^{n-3} + \ldots + a_2 p^1 + a_1 p^0) + a_0 . \qquad (1.6)$$

Праву частину виразу (1.6) розділимо на основу системи числення p. В результаті визначимо перший залишок a_0 – молодший p-ічний розряд числа та цілу частину

$$a_n p^{n-1} + a_{n-1} p^{n-2} + a_{n-2} p^{n-3} + \ldots + a_2 p^1 + a_1 p^0 .$$

Розділивши цілу частину на p, отримуємо наступний залишок a_1. Повторюючи процес ділення n разів, отримуємо останню цілу частку a_n, яка за умовою менша за основу системи і є старшою цифрою числа, представленого в системі з основою p.

У разі представлення дробового числа, що не містить цілої частини (правильного дробу), алгоритм перекладу дещо коригується. Нехай

$$A = a_{-1} p^{-1} + a_{-2} p^{-2} + \ldots + a_{-(m-1)} p^{-(m-1)} + a_{-m} p^{-m} . \qquad (1.7)$$

Розкладемо (1.7) за схемою Горнера

$$A = p^{-1}(a_{-1} + a_{-2} p^{-1} + \ldots + a_{-(m-1)} p^{-m} + a_{-m} p^{-(m+1)}) . \qquad (1.8)$$

Праву частину виразу (1.8) помножимо на основу системи числення p. В результаті визначимо цілу частину a_0 – перший після роздільної точки p-ічний розряд числа і залишок

$$(a_{-1} + a_{-2} p^{-1} + \ldots + a_{-(m-1)} p^{-m} + a_{-m} p^{-(m+1)}) .$$

Помноживши залишок на p, отримуємо наступну цілу частину a_{-2}. Повторюючи процес множення m разів, отримуємо останній залишок a_{-m}. На практиці число дробових розрядів визначається залежно від вимог до необхідної точності представлення (на відміну від перетворення цілих чисел, коли число розрядів визначається величиною числа).

При перетворенні числа, що містить як цілу, так і дробову частини (неправильного дробу) доцільно перевести обидві частини окремо за відповідними алгоритмами та об'єднати результати.

Найчастіше для представлення інформації використовується двійкова система числення – однорідна позиційна система числення з основою $p = 2$. У двійковій системі числення записуються за допомогою

двох символів (0 та 1). Переваги двійкової системи числення полягають у простоті реалізації процесів зберігання, передачі та обробки інформації на комп'ютері. Це зумовлено тим, що будь-який сучасний цифровий пристрій складається з логічних елементів І - АБО - НЕ, що мають два стійкі стани.

Число в двійковій системі може бути представлене як

$$A = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_1 2 + a_0 2^0 + a_{-1} 2^{-1} + \ldots + a_{-(m-1)} 2^{-(m-1)} + a_{-m} 2^{-m}. \quad (1.9)$$

Розглянемо приклади перетворення чисел у двійковій системі. Щоб розрізняти, в якій системі числення представлено число, його доповнюють вказівником праворуч внизу. Наприклад, число в десятковій системі 13_{10} може бути записано як 1101_2.

Приклад 1.5. Перевести ціле число A=100110001011_2 із двійкової системи до десяткової.

Рішення. Відповідно до (1.9) ціле число в двійковій системі може бути представлене як

$$A_2 = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_1 2^1 + a_0 2^0. \quad (1.10)$$

Визначаємо кількість розрядів двійкового представлення числа A. Число розрядів L =12, тоді показник ступеня старшого розряду n=L–1 = 11. Число A виражається як

$$A_2 = a_{11}2^{11} + a_{10}2^{10} + a_9 2^9 + a_8 2^8 + a_7 2^7 + a_6 2^6 + a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0 \quad (1.11)$$

Значення a_i одержуємо з вихідних даних

a_{11} = 1, a_{10} = 0, a_9 = 0… a_1 = 1, a_0 = 1.

Тоді $A_2 = 1\cdot2^{11} + 0\cdot2^{10} + 0\cdot2^9 + 1\cdot2^8 + 1\cdot2^7 + 0\cdot2^6 + 0\cdot2^5 + 0\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$ $= 2048 + 256 + 128 + 8 + 2 + 1 = 2443_{10}$. ■

Приклад 1.6. Перетворити ціле число A=1358_{10} із десяткової системи на двійкову.

Рішення. Перекладаємо число A=1358_{10} у двійковий код. Для цього ділимо число, що перетворюється, на два і записуємо залишки, починаючи з молодшого розряду:

1358| 2
679| 0
339| 1
169| 1
84| 1
42| 0
21| 0
10| 1
5| 0
2| 1
1| 0
0| 1

$1358_{10}=10101001110_2$.

Перевіримо перетворення:

$10101001110_2=1\cdot2^{10}+0\cdot2^9+1\cdot2^8+0\cdot2^7+1\cdot2^6+0\cdot2^5+0\cdot2^4+1\cdot2^3+1\cdot2^2+1\cdot2^1+0\cdot2^0=$

$=1024+256+64+8+4+2=1358.$■

Приклад 1.7. Перевести дробове число $A=0{,}0100110001_2$ із двійкової системи до десяткової.

Рішення. Відповідно до (1.9) дробове число в двійковій системі може бути представлене як

$$A= a_{-1}2^{-1}+ a_{-2}2^{-2}+\ldots+ a_{-(m-1)}2^{-(m-1)}+ a_{-m}2^{-m}. \quad (1.12)$$

Визначаємо кількість розрядів двійкового представлення числа A. Число розрядів $m = 10$

$$A_2=a_{-1}2^{-1}+a_{-2}2^{-2}+a_{-3}2^{-3}+a_{-4}2^{-4}+a_{-5}2^{-5}+a_{-6}2^{-6}+a_{-7}2^{-7}+a_{-8}2^{-8}+a_{-9}2^{-9}+a_{-10}2^{-10} \quad (1.13)$$

Значення a_i одержуємо з вихідних даних:

$a_{-1} = 0$, $a_{-2} = 1$, $a_{-3} = 0\ldots$ $a_{-9} = 0$, $a_{-10} = 1$.

Тоді

$$A_2=0\cdot2^{-1}+1\cdot2^{-2}+0\cdot2^{-3}+0\cdot2^{-4}+1\cdot2^{-5}+1\cdot2^{-6}+0\cdot2^{-7}+0\cdot2^{-8}+0\cdot2^{-9}+1\cdot2^{-10} \quad (1.14)$$ ■

Приклад 1.8. Перевести дробове число $A=0{,}0438_{10}$ з десяткової системи до двійкової, передбачивши 12 двійкових розрядів для дробової частини.

Рішення. Перекладаємо число 0,0438 у двійковий код. Для цього множимо перетворюване число на два і записуємо цілі частини (0 або 1), починаючи з першого розряду після коми, поки не буде забезпечено 12 розрядів. Якщо отримана ціла частина виявляється рівною одиниці, вона при наступному множенні відкидається:

0,0438 | ×2
0,0876 | 0
0,1752 | 0
0,3504 | 0
0,7008 | 0
1,4016 | 1
0,8032 | 0
1,6064 | 1
1,2128 | 1
0,4256 | 0
0,8512 | 0
1,7024 | 1
1,4048 | 1

$0{,}0438_{10}=0{,}000010110011_2$.

Перевіримо перетворення:

$$0{,}000010110011_2=0\cdot2^{-1}+0\cdot2^{-2}+0\cdot2^{-3}+0\cdot2^{-4}+1\cdot2^{-5}+0\cdot2^{-6}+1\cdot2^{-7}+1\cdot2^{-8}+0\cdot2^{-9}+0\cdot2^{-10}+1\cdot2^{-11}+1\cdot2^{-12}=$$

$$=\frac{1}{32}+\frac{1}{128}+\frac{1}{256}+\frac{1}{2048}+\frac{1}{4096}=\frac{128+32+16+2+1}{4096}=\frac{179}{4096}=0{,}043701171\approx0{,}0438.\ \blacksquare$$

Приклад 1.9. Перевести число $A=79{,}293_{10}$ з десяткової системи до двійкової, передбачивши 8 двійкових розрядів для дробової частини.

Рішення. При перетворенні цілої частини використовуємо алгоритм, розглянутий у прикладі 1.6, при перетворенні дробової частини – у прикладі 1.8. Перетворимо цілу частину

79| 2
39| 1
19| 1
 9| 1
 4| 1
 2| 0
 1| 0
 0| 1

$79_{10}=1001111_2$.

Перевіримо перетворення цілої частини:

$1001111_2=1\cdot2^6+0\cdot2^5+0\cdot2^4+1\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0=64+8+4+2+1=79$.

Перетворимо дробову частину.

0,293 | ×2
0,586 | 0
1,172 | 1
0,344 | 0
0,688 | 0
1,376 | 1
0,752 | 0
1,504 | 1
1,008 | 1

$0{,}293_{10}=0{,}01001011_2$.

Перевіримо перетворення:

$0{,}01001011_2=0\cdot2^{-1}+1\cdot2^{-2}+0\cdot2^{-3}+0\cdot2^{-4}+1\cdot2^{-5}+0\cdot2^{-6}+1\cdot2^{-7}+1\cdot2^{-8}=$

$$=\frac{1}{4}+\frac{1}{32}+\frac{1}{128}+\frac{1}{256}=\frac{64+8+2+1}{256}=\frac{75}{256}=0{,}29296875\approx0{,}293$$

Об'єднуємо цілу та дробову частини $79{,}293_{10}=1001111{,}01001011_2$.■

Крім двійкової системи для відображення інформації використовуються *вісімкова* і *шістнадцяткова* системи числення. Головна їх перевага полягає в тому, що переведення чисел з двійкової системи в систему числення з основою 2^k і зворотне перетворення дуже просте. З іншого боку при аналізі двійкової інформації буває доцільно виконати проміжне конвертування на систему числення з основою 2^k, що дозволяє підвищити наочність представлення даних.

Правило переведення чисел в систему числення з основою 2^k виглядає наступним чином: вихідне число переводиться в двійковий код, потім двійкове число ділиться на групи по k розрядів, починаючи від коми (праворуч і ліворуч). Кожна група з k двійкових розрядів замінюється однією цифрою (оскільки число, представлене групою з k двійкових розрядів, міститься в межах від 0 до 2^k-1, то загальна кількість різних символів становить 2^k). Природно, що можливе переведення у вісімкову та шістнадцяткову системи числення традиційним методом відповідно до рівнянь (1.5) – (1.8).

Вісімкова система числення – позиційна система числення з основою вісім. Для представлення чисел у ній використовуються цифри від 0 до 7. Характеризується легким переведенням вісімкових чисел у двійкові та назад шляхом заміни вісімкових чисел на триплети (трійки) двійкових

відповідно до таблиці 1.1. При цьому до неповних груп ліворуч дописуються нулі.

Таблиця 1.1 – Відповідність між вісімковими та двійковими числами

Вісімкове число	0	1	2	3	4	5	6	7
Двійкове число	000	001	010	011	100	101	110	111

Приклад 1.10. Перетворити ціле число $A=2735_8$ з вісімкової системи в двійкову та десяткову.

Рішення. Для переведення в двійковий код замість кожної вісімкової цифри записуємо триплет двійкових відповідно до табл.1.1.

$2735_8 = 010\ 111\ 011\ 101_2$.

У десятковий код вісімкове число переводимо відповідно до формули (1.5).

$2735_8=2\cdot8^3+7\cdot8^2+3\cdot8^1+5\cdot8^0=2\cdot512+7\cdot64+3\cdot8+5\cdot1=1024+448+24+5=1501_{10}$.■

Приклад 1.11. Перетворити ціле число $A=810_{10}$ із десяткової системи у двійкову та вісімкову.

Рішення. Перекладаємо число 810 у двійковий код, використовуючи алгоритм, розглянутий у прикладі 1.6.

$810_{10}=1100101010_2$.

Двійкове представлення числа розбиваємо праворуч наліво на триплети, дописуючи при необхідності на початку незначні нулі, і надаючи одне із значень 0..7 кожній з трійок: $810_{10}=001\ 100\ 101\ 010_2=1452_8$.

Переведемо число 810 у вісімкову систему шляхом ділення на основу системи числення – число 8.

810 | 8
101| 2
12| 5
8| 4
0| 1

$810_{10}=1452_8$,

тобто результати отримані двома способами, збігаються. ■

Шістнадцяткова система числення - позиційна система числення з основою 16. Як символи цієї системи числення зазвичай використовуються

цифри від 0 до 9 і латинські літери від A до F. Букви A, B, C, D, E, F мають значення 10_{10}, 11_{10}, 12_{10}, 13_{10}, 14_{10}, 15_{10} відповідно.

Система широко використовується в низькорівневому програмуванні та комп'ютерній графіці, оскільки мінімальною одиницею комп'ютерної пам'яті, що адресується, є восьмирозрядний байт, значення якого зручно записувати двома шістнадцятковими цифрами (напівбайтами).

Для переведення двійкового числа в шістнадцяткову систему потрібно розбити його на тетради (четвірки) праворуч наліво, дописуючи при необхідності в неповні групи зліва нулі та замінити кожну тетраду шістнадцятковою цифрою відповідно до табл. 1.2. Для переведення числа із шістнадцяткової системи в двійкову потрібно замінити кожну його цифру на відповідну тетраду.

Таблиця 1.2 – Відповідність між шістнадцятковими, двійковими та десятковими числами

Десяткове число	Шістнадцяткове число	Двійкове число
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Приклад 1.12. Перетворити ціле число A=3D6A$_{16}$ з шістнадцяткової системи в двійкову і десяткову.

Рішення. Для переведення в двійковий код замість кожної шістнадцяткової цифри записуємо тетраду двійкових відповідно до табл.1.2.

$3D6A_{16}=0011\ 1101\ 0110\ 1010_2$.

У десятковий код шістнадцяткове число переводимо у відповідності з формулою (1.5).

$3D6A_{16}=3\cdot16^3+13\cdot16^2+6\cdot16^1+10\cdot16^0=3\cdot4096+13\cdot256+6\cdot16+10\cdot1=$

$=12288+3328+96+10=15722_{10}$. ■

Приклад 1.13. Перетворити ціле число $A=1470_{10}$ з десяткової системи у двійкову та шістнадцяткову.

Рішення. Перекладаємо число 1471 у двійковий код, використовуючи алгоритм, розглянутий у прикладі 6.

$1470_{10}=10110111110_2$.

Для переведення в шістнадцятковий код двійкове представлення числа розбиваємо праворуч наліво на тетради, дописуючи при необхідності на початку незначні нулі, і привласнюючи одне зі значень 0...F кожній з четвірок:

$1470_{10}=0101\ 1011\ 1110_2=5BE_{16}$.

Переведемо число 1470 у шістнадцяткову систему шляхом ділення на основу системи числення – число 16.

```
 1470 | 16
   91 | 14
    5 | 11
    0 | 5
```

$1470_{10}=5BE_{16}$, тобто результати, отримані двома способами, збігаються. ■

При виведенні цифрової інформації на індикацію, введенні з клавіатури, в телефонному зв'язку використовується так званий двійково-десятковий код (англ. *binary-coded decimal*, *BCD*). У цьому коді кожна цифра десяткового коду записується чотирирозрядним двійковим кодом відповідно до табл. 1.3. Цей код називають іноді кодом «8, 4, 2, 1» (відповідно до ваги двійкових розрядів).

Таблиця 1.3 – Відповідність між десятковими цифрами та двійковими кодами

Десяткова цифра	0	1	2	3	4	5	6	7	8	9
Двійковий код	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Приклад 1.14. Перетворити ціле число $A=3698_{10}$ у двійково-десятковий код.

Рішення. Ставимо у відповідність кожної з десяткових цифр четвірку (тетраду) двійкових:

$3698_{10}=0011\ 0110\ 1001\ 1000_{2\text{-}10}$. ■

Увага! Переводити вихідне число *A* в двійковий код **не потрібно**.

Приклад 1.15. Перетворити двійково-десяткове число $A=00010111010010000011_{2\text{-}10}$ у десятковий код.

Рішення. Розбиваємо двійково-десятковий код на четвірки і ставимо у відповідність кожній з тетрад десяткову цифру:

$A=0001\ 0111\ 0100\ 1000\ 0011_{2\text{-}10} = 17483_{10}$ ■

1.2 Подання чисел зі знаками

При обробці інформації в цифрових обчислювальних пристроях буває необхідно робити арифметичні операції як з додатними, так і з від’ємними числами. Для відображення чисел зі знаками застосовуються три основні коди.

При записі числа у *прямому* коді старший розряд представляє знак числа та називається *знаковим розрядом*. Якщо знаковий розряд дорівнює 0, число додатне, якщо дорівнює 1 від’ємне. У інших (*інформаційних*) розрядах записується двійкове представлення модуля числа. Таким чином, для представлення числа у прямому коді достатньо до двійкового запису його абсолютної величини приписати нуль для додатних чисел або одиницю для від’ємних.

Приклад 1.16. Перетворити числа +88 та –88 у прямий код.

Рішення. Перекладаємо число 88 у двійковий код, використовуючи алгоритм, розглянутий у прикладі 1.6.

$88_{10}=1011000_2$

Для кодування додатного числа приписуємо зліва до двійкового подання модуля числа нуль, відповідний знаку +

$+88_{пр}=01011000.$

Для кодування від'ємного числа приписуємо зліва до двійкового представлення одиницю, відповідну знаку –

$-88_{пр}=11011000.$■

Зворотний код додатного числа збігається із прямим кодом цього числа, тобто складається з нульового знакового розряду та двійкового представлення абсолютної величини числа. Зворотний двійковий код від'ємного числа складається з однорозрядного коду знака (двійкової цифри 1), за яким слідує інвертоване представлення модуля числа (при інвертуванні числа нулі замінюються одиницями, а одиниці – нулями). Таким чином, для зміни знака числа у зворотному коді достатньо проінвертувати всі його розряди, не звертаючи уваги, чи це знаковий розряд чи інформаційний.

Приклад 1.17. Перетворити числа +88 та –88 у зворотний код.

Рішення. Відповідно до результатів прикладу 1.6

$88_{10}=1011000_2$, $+88_{пр}=01011000.$

Зворотний код додатного числа збігається із прямим кодом цього числа

$+88_{звор} = +88_{пр} = 01011000.$

Для кодування від'ємного числа приписуємо ліворуч до двійкового подання одиницю, відповідну знаку – та інвертуємо всі інформаційні розряди

$-88_{звор} =10100111.$ ■

Додатковий код – найпоширеніший спосіб представлення чисел зі знаком у цифрових пристроях. Він дозволяє замінити операцію віднімання операцією додавання і зробити операції додавання та віднімання однаковими для знакових і беззнакових чисел, що спрощує структуру арифметично-логічного пристрою.

Додатковий код додатного числа збігається з прямим кодом цього числа, тобто складається з нульового знакового розряду та двійкового представлення абсолютної величини числа. Для від'ємного числа додатковий код утворюється шляхом одержання зворотного коду (інвертуванням його двійкового модуля та приписуванням одиничного

знакового розряду) та додаванням до молодшого розряду інверсії одиниці. При цьому слід врахувати можливий при складанні ланцюжок переносів.

У [4,5] показано, що переклад із додаткового коду в прямий здійснюється за тими самими правилами, що і з прямого в додатковий – інвертування розрядів і додаванням одиниці в молодший розряд.

Приклад 1.18. Перетворити числа +88 та –88 на додатковий код.

Рішення. Відповідно до результатів прикладу 16

$88_{10}=1011000_2$, $+88_{пр}=01011000$.

Додатковий код додатного числа збігається із прямим кодом цього числа $+88_{дод} =+88_{пр}=01011000$.

Для отримання додаткового коду від'ємного числа до молодшого розряду зворотного коду додається одиниця:

$-A_{дод} =-A_{звор} +1$.

$$\begin{array}{r} 10100111 \\ +\quad\quad 1 \\ \hline 10101000 \end{array}$$

$-88_{дод} = 10101000$. ■

Як зазначалося вище, у додатковому коді можна проводити операції складання чисел зі знаком без будь-якої корекції результату (що, взагалі кажучи, несправедливо для прямого і зворотного кодів). При цьому використовується звичайний дворозрядний двійковий суматор, таблиця істинності одного розряду якого наведена в табл. 1.4.

Таблиця 1.4 – Таблиця істинності розряду багаторозрядного двійкового суматора

Вхідні сигнали			Вихідні сигнали	
Розряд першого доданку a_i	Розряд другого доданку b_i	Вхідне перенесення c_i	Розряд суми S_i	Вихідне перенесення w_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Приклад 1.19. Для чисел E=420 і F=171 виконати такі дії. Перевести додатковий код числа ± E та ± F. Обчислити в додатковому коді $+E+F$, $+E-F$, $-E+F$, $-E-F$, перевірити результати обчислень.

Рішення. Перекладемо ± E та ± F у додатковий код, використовуючи алгоритм, розглянутий у прикладі 1.18.

$E=420_{10}=110100100_2$; $+E_{\text{дод}}=0110100100$; $-E_{\text{дод}}=1001011100$;

$F=171_{10}=10101011$.

Враховуючи, що в подальшому належить виконувати операцію додавання, вирівнюємо число розрядів двійкового подання в обох числах шляхом дописування до числа з меншою розрядністю необхідного числа нулів, в даному випадку одного: $171_{10}=010101011_2$.

$+F_{\text{дод}}=0010101011$; $-F_{\text{дод}}=1101010101$.

Звертаємо увагу, що приписаний зліва нуль після інверсії перетворився на одиницю. Складаємо у додатковому коді.

Обчислимо $+420_{\text{дод}}-171_{\text{дод}}$.

$$\begin{array}{r} 0110100100 \\ +1101010101 \\ \hline 0011111001 \end{array}$$

Додавання у кожному розряді проводиться відповідно до табл. 1.4. Перенесення зі старшого інформаційного розряду в знаковий провадиться за загальними правилами, перенесення із знакового розряду ігнорується.

Перевіримо правильність результату. Знаковий розряд дорівнює нулю, отже, результат є додатним і може бути легко переведений у прямий код, а потім і в десяткову систему.

$G=+E-F=0011111001_{\text{дод}}=0011111001_{\text{пр}}=+11111001_2=$

$=1\cdot2^7+1\cdot2^6+1\cdot2^5+1\cdot2^4+1\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0=249_{10}$.

Дійсно, +420–171=249, тобто обчислення зроблено правильно.

Обчислимо $-420_{\text{дод}}+171_{\text{дод}}$.

$$\begin{array}{r} 1001011100 \\ +0010101011 \\ \hline 1100000111 \end{array}$$

Перевіримо правильність результату. Знаковий розряд дорівнює одиниці, отже, результат від'ємний. Для визначення його чисельного значення доцільно перевести результат у прямий код, а потім у десяткову систему.

$G = -E+F = 1100000111_{\text{дод}}$.

Перекладемо результат у прямий код. Для цього проінвертуємо всі інформаційні розряди та додамо одиницю до молодшого розряду.

$G = 1011111001_{\text{пр}} = -11111001_2 =$

$= -(1\cdot2^7+1\cdot2^6+1\cdot2^5+1\cdot2^4+1\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0) = -249_{10}$.

Дійсно, –420+171= –249, тобто обчислення вірні.

Обчислимо $+420_{\text{дод}}+171_{\text{дод}}$

0110100100

+0010101011

1001001111

Перевіримо правильність результату. Знаковий розряд дорівнює одиниці, отже, результат має бути від'ємним, що неможливо під час підсумовування двох додатних чисел. Очевидно, під час підсумовування відбулося переповнення розрядної сітки, тобто результат перевищив за абсолютною величиною $2^9-1=511$, максимальне значення, яке може бути представлене дев'ятьма інформаційними розрядами. Методи визначення переповнення будуть розглянуті нижче.

Обчислимо $-420_{\text{доп}}-171_{\text{доп}}$

1001011100

+1101010101

0110110001

Перевіримо правильність результату. Знаковий розряд дорівнює нулю, отже, результат має бути додатним, що неможливо під час підсумовування двох від'ємних чисел. Очевидно, під час підсумовування відбулося переповнення розрядної сітки. ■

При проведенні операції додавання, щоб не отримувати невірних результатів, необхідно виявляти переповнення, тобто ситуацію, коли сума за абсолютною величиною перевищує $A_{\max}=2^k-1$, де k– число інформаційних розрядів. Найбільш поширеним методом виявлення переповнення є аналіз переносів, що виникають під час додавання. В [4,5] показано, що за значеннями перенесення зі старшого інформаційного розряду в знаковий і зі знакового можна індикувати переповнення (табл. 1.5).

Таблиця 1.5 – Співвідношення між сигналами перенесення та результатом підсумовування

Перенесення зі старшого до знакового *CH*	Перенесення зі знакового *CS*	Результат
0	0	Коректне число
0	1	Занадто велике за модулем від'ємне число
1	0	Занадто велике за модулем додатне число
1	1	Коректне число

Таким чином, прапор (розряд стану) OF, що вказує на переповнення розрядної сітки, у більшості мікропроцесорів формується як OF=CH ⊕ CS, де ⊕ – логічна функція ВИКЛЮЧНЕ АБО (сума за модулем два).

Іншим, найбільш наочним методом визначення переповнення, є застосування так званого *модифікованого додаткового коду*. У модифікованому додатковому коді під знак числа відводиться не один, а два розряди: 00 відповідає знаку плюс, 11 – знаку мінус. Будь-яка інша комбінація (01 або 10), що може виникнути у знакових розрядах, служить ознакою переповнення розрядної сітки. Додавання чисел у модифікованому коді нічим не відрізняється від додавання в звичайному додатковому коді.

Приклад 1.20. Для чисел E = 219 і F = 58 виконати такі дії. Перевести до модифікованого додаткового коду числа ±E та ±F. Обчислити у модифікованому додатковому коді +E+F, +E–F, –E+F, –E–F, перевірити результати обчислень.

Рішення. Перекладемо ± E та ± F у додатковий код, використовуючи алгоритм, розглянутий у прикладі 1.17.

$E=219_{10} = 11011011_2$; $+E_{\text{дод}} = 011011011$; $-E_{\text{дод}} = 100100101$;

$F=58_{10} = 111010$.

Враховуючи, що надалі належить виконувати операцію складання, вирівнюємо число розрядів двійкового подання в обох числах шляхом дописування до числа з меншою розрядністю двох нулів:

$58_{10} = 00111010_2$.

$+F_{\text{дод}} = 000111010$; $-F_{\text{дод}} = 111000110$.

Звертаємо увагу, що приписані зліва нулі після інверсії перетворилися на одиниці. Перекладаємо в модифікований додатковий код, для чого подвоюємо знакові розряди.

$+E_{\text{мдод}} = 0011011011$; $-E_{\text{мдод}} = 1100100101$;

$+F_{\text{мдод}} = 0000111010$; $-F_{\text{мдод}} = 1111000110$.

Складаємо у модифікованому додатковому коді

$+219_{\text{мдод}}+58_{\text{мдод}}$.

0011011011

+0000111010

0100010101.

Оскільки в знакових розрядах суми (виділені) розташовуються два різні двійкові значення 01, то очевидно, що в результаті додавання відбулося переповнення (занадто велике додатне число), результат не може бути направлений на подальшу обробку і підлягає корекції.

$+219_{\text{мдод}}-58_{\text{мдод}}$.

0011011011

+1111000110

0010100001.

Оскільки в знакових розрядах суми розташовуються два однакові двійкові значення 00, то очевидно, що в результаті складання переповнення не відбулося, отримано додатне число і результат може бути направлений в подальшу обробку без корекції. Перевіримо правильність результату. Результат є додатним і може бути легко переведений у прямий код, а потім і в десяткову систему.

$G=+E-F=0010100001_{\text{мдоп}} = +10100001_2=$

$=1\cdot2^7+0\cdot2^6+1\cdot2^5+0\cdot2^4+0\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0=128+32+1 = 161_{10}$.

Дійсно, +219−58=161, тобто обчислення зроблено вірно.

$-219_{\text{мдод}}+58_{\text{мдод}}$.

1100100101

+0000111010

1101011111.

Оскільки в знакових розрядах суми розташовуються два однакові двійкові значення 11, то очевидно, що в результаті складання

переповнення не відбулося, отримано від'ємне число і результат може бути направлений в подальшу обробку без корекції. Для визначення його чисельного значення доцільно перевести результат у прямий код, а потім у десяткову систему.

$G= -E+F=1101011111_{мдод}$.

Перекладемо результат у прямий код. Для цього проінвертуємо всі інформаційні розряди та додамо одиницю до молодшого розряду.

$G= 101011111_{дод} = 110100001_{пр}= -10100001_2=$

$= -(1\cdot2^7+0\cdot2^6+1\cdot2^5+0\cdot2^4+0\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0)= -161_{10}$.

Дійсно, –219+58= –161, тобто обчислення вірні.

$-219_{мдоп}-58_{мдоп}$.

1100100101

+1111000110

1011101011.

Оскільки в знакових розрядах суми (виділені) розташовуються два різні двійкові значення 10, то очевидно, що в результаті додавання відбулося переповнення (занадто велике за модулем від'ємне число), результат не може бути направлений на подальшу обробку і підлягає корекції.

1.3 Подання чисел у форматі з плаваючою комою

При зберіганні та обробці чисел у цифрових пристроях зазвичай важливіша не абсолютна, а відносна точність представлення. Підвищити точність і значно розширити діапазон збережених чисел дозволяє формат представлення даних з плаваючою (рухомою) комою (плаваючою точкою).

Для представлення чисел з плаваючою комою використовується напівлогарифмічна форма запису числа

$$A_{пк} = \pm m \cdot p^{\pm E}, \qquad (1.15)$$

де p – основа системи числення, E – порядок числа, m – мантиса числа A.

Положення коми визначається значенням порядку E. Зі зміною порядку точка переміщається (плаває) вліво або вправо. Наприклад,

$27_{10} = 270\cdot10^{-1} = 27\cdot10^0 = 2{,}7\cdot10^1 = 0{,}27\cdot10^2 = 0{,}027\cdot10^3$=...

Для встановлення однозначності при запису чисел прийнято вживати нормалізовану форму запису числа, тобто визначати мінімальне та максимальне значення мантиси. Зазвичай мантиса нормалізованого числа може змінюватись у діапазоні: $p^{-1} \le |m| < 1$, однак, іноді приймаються й інші домовленості. Таким чином, у нормалізованих двійкових числах мантиса являє собою правильний дріб $\frac{1}{2} \le |m| < 1$, в якому цифра після точки повинна бути значною (більшою за нуль). Зокрема, у двійкового нормалізованого числа у старшому розряді мантиси завжди стоїть 1. При записі числа у форматі з плаваючою комою виділяються такі поля: знак мантиси (вказує на від'ємність чи додатність числа), мантиса, знак порядку, порядок.

Приклад 1.21. Визначити значення двійкових чисел *A* і *B*, представлених у форматі з плаваючою комою. Число двійкових розрядів виділених для запису числа наступне: знак мантиси – один розряд, мантиса – вісім, знак порядку – один, порядок чотири розряди. Мантиса та порядок представлені у прямих кодах.

$A_{\text{пк}}$ = 0 10010111 0 0101;

$B_{\text{пк}}$ = 1 10100100 1 0011.

Рішення. Мантиса числа *A* дорівнює

$$m_A = +10010111_2 = \frac{1}{2} + \frac{1}{16} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} = \frac{128+16+4+2+1}{256} = \frac{151}{256},$$

(неявно передбачається, що кома стоїть перед старшим розрядом мантиси). Порядок числа *A*, в свою чергу,

$E_A = +0101_2 = +5$.

Згідно (1.15) $A = \frac{151}{256} \cdot 2^5 = \frac{151}{256} \cdot 32 = \frac{151}{8} = 18\frac{7}{8} = 18{,}875$.

Мантиса числа *B* дорівнює

$$m_B = -10100100_2 = -\left(\frac{1}{2} + \frac{1}{8} + \frac{1}{64}\right) = -\frac{32+8+1}{64} = -\frac{41}{64},$$

порядок числа *B* дорівнює

$E_B = -0011_2 = -3$.

Згідно (1.15) $B = -\frac{41}{64} \cdot 2^{-3} = -\frac{41}{64} \cdot \frac{1}{8} = -\frac{41}{512} = -0{,}080078125$. ■

Розглянемо алгоритм перекладу чисел із формату з фіксованою комою у формат із плаваючою. Доцільно при цьому виділити три випадки.

Перший випадок: $0{,}5 \le |A| < 1$. Порядок при цьому дорівнює нулю.

$E_A = 0_{10} = 0\ 00\ldots000_2$.

Мантиса збігається з числом, що перетворюється.

Приклад 1.22. Перетворити на формат з плаваючою комою числа $A = 0{,}738$ та $B = -0{,}61$. Число двійкових розрядів виділених для запису числа наступне: знак мантиси – один розряд, мантиса – вісім, знак порядку – один, порядок чотири розряди. Мантису та порядок подати у прямих кодах.

Рішення. Перетворимо число A у двійкову систему числення, зберігши вісім розрядів після коми, та у прямий код.

$A = 0{,}738_{10} = 0{,}10111101_2 = 0\ 10111101_{\text{пр}}$.

Мантиса збігається з числом, що перетворюється. $m_A = 0\ 10111101$ (не-явно передбачається, що кома стоїть перед старшим розрядом мантиси); Порядок дорівнює нулю $E_A = 0_{10} = 0\ 0000$.

Об'єднуємо мантису та порядок:

$A_{\text{пк}} = 0\ 10111101\ 0\ 0000$.

Виконаємо аналогічні операції з числом B.

$B = -0{,}61 = -0{,}10011100_2 = 1\ 10011100_{\text{пр}}$.

Мантисса збігається з числом, що перетворюється. $m_B = 1\ 10011100$. Порядок дорівнює нулю $E_B = 0_{10} = 0\ 0000$.

$B_{\text{пк}} = 1\ 10011100\ 0\ 0000$.■

Другий випадок перетворення має місце, коли $|A| \ge 1$.. Порядок при цьому більший за нуль. $E_A = 0\ \times\times\ldots\times\times\times_2$.

Для визначення порядку слід подумки рухати кому вліво, доки вона стане перед першою одиницею в двійковому поданні числа. Порядок дорівнюватиме кількості зрушень, а мантиса – отриманому після зсуву коми правильному дробу.

Приклад 1.23. Перетворити на формат з плаваючою комою числа A=37,28 та B=–1291. Число двійкових розрядів виділених для запису числа наступне: знак мантиси – один розряд, мантиса – вісім, знак порядку – один, порядок чотири розряди. Мантису та порядок подати у прямих кодах.

Рішення. Перетворимо число A на двійкову систему числення і в прямий код, зберігши вісім розрядів сумарно в цілій і дробовій частині.

$A = 37{,}28_{10} = 100101{,}010_2 = 0\ 100101{,}01_{\text{пр}}$.

Зсуваємо кому вліво, поки вона не стане перед першою одиницею в інформаційної частини числа. Для цього знадобилося шість зрушень, тому порядок дорівнює шести. $E_A = 6_{10} = 0\ 0110$.

Мантиса, що отримана після зсуву коми вліво: $m_A = 0\ 10010101$.

Об'єднуємо мантису та порядок:

$A_{\text{пк}} = 0\ 10010101\ 0\ 0110$.

Перевіримо перетворення. Мантиса числа A дорівнює

$$m_A = 0\ 10010101_2 = \frac{1}{2} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256} = \frac{128+16+4+1}{256} = \frac{149}{256},$$

порядок числа A дорівнює $E_A = 0\ 0110_{\text{пр}} = +6$.

Згідно (1.15) $A = \frac{149}{256} \cdot 2^6 = \frac{149}{256} \cdot 64 = \frac{149}{4} = 37{,}25 \approx 37{,}28.$

.

Виконаємо аналогічні операції з числом B. Перетворимо число B у двійкову систему числення та у прямий код.

$B = -1289_{10} = -10100001001_2 = 1\ 10100001001{,}00\ldots_{\text{пр}}$.

Зсуваємо кому вліво, поки вона стане перед першою одиницею в інформаційної частини числа. Для цього знадобилося одинадцять зсувів, тому порядок дорівнює одинадцяти. $E_B = 11_{10} = 0\ 1011$.

Мантиса, що отримана після зсуву коми вліво та збереження восьми знаків: $m_B = 1\ 10100001$. Об'єднуємо мантису та порядок:

$B_{\text{пк}} = 1\ 10100001\ 0\ 1011$.

Перевіримо перетворення. Мантиса числа B дорівнює

$$m_B = 1\ 10100001_2 = -\left(\frac{1}{2} + \frac{1}{8} + \frac{1}{256}\right) = -\frac{128+32+1}{256} = -\frac{161}{256},$$

порядок числа B дорівнює

$E_B = 0\ 1011_{\text{пр}} = +11$.

Згідно (1.15) $A = -\frac{161}{256} \cdot 2^{11} = -\frac{161}{256} \cdot 2048 = -161 \cdot 8 = -1288 \approx -1291.$ ■

Третій випадок $|A| < 0{,}5$. Порядок при цьому менший за нуль.

$E_A = 1\ \times\times\ldots\times\times\times_2$.

Для визначення порядку слід подумки рухати кому вправо, поки вона стане перед першою одиницею в двійковому поданні числа. Порядок буде від’ємним і дорівнює за абсолютною величиною числу зрушень, а мантиса – отриманому після зсуву коми правильному дробу.

Приклад 1.24. Перетворити на формат з плаваючою комою числа $A = 0{,}0166$ та $B = -0{,}1037$. Число двійкових розрядів виділених для запису числа наступне: знак мантиси - один розряд, мантиса - вісім, знак порядку - один, порядок чотири розряди. Мантису та порядок подати у прямих кодах.

Рішення. Перетворимо число A в двійкову систему числення і в прямий код, зберігши вісім значних розрядів у дрібній частині, відраховуючи від старшої одиниці.

$A = 0{,}0166_{10} = 0{,}00000100010000_2 = 0\ 0000010001000_{\text{пр}}$.

Зсуваємо кому вправо, поки вона не стане перед першою одиницею в інформаційній частині числа. Для цього потрібно п'ять зрушень, тому порядок дорівнює мінус п'яти. $E_A = -5_{10} = 1\ 0101_{\text{пр}}$.

Мантиса, що отримана після зсуву коми праворуч та збереження восьми знаків: $m_A = 0\ 10001000$.

Об'єднуємо мантису та порядок: $A_{\text{пк}} = 0\ 10001000\ 1\ 0101$.

Перевіримо перетворення. Мантиса числа A дорівнює

$$m_A = 0\ 10001000_2 = \frac{1}{2} + \frac{1}{32} = \frac{16+1}{32} = \frac{17}{32},$$

порядок числа A дорівнює

$E_A = 1\ 0101_{\text{пр}} = -5$.

Згідно (1.15) $A = \frac{17}{32} \cdot 2^{-5} = \frac{17}{32} \cdot \frac{1}{32} = \frac{17}{1024} = 0{,}0166015625 \approx 0{,}0166$.

Виконаємо аналогічні операції з числом B. Перетворимо число B у двійкову систему числення та у прямий код.

$B = -0{,}104_{10} = -0{,}00011010101_2 = 1\ 00011010101_{\text{пр}}$.

Зрушуємо кому вправо, поки вона не стане перед першою одиницею в інформаційній частині числа. Для цього потрібно три зрушення, тому порядок дорівнює мінус трьом. $E_B = -3_{10} = 1\ 0011_{\text{пр}}$.

Мантиса, що отримана після зсуву коми праворуч та збереження восьми знаків: $m_B = 1\ 11010101$.

Об'єднуємо мантису та порядок:

$B_{\text{пк}} = 1\ 11010101\ 1\ 0011$.

Перевіримо перетворення. Мантиса числа A дорівнює

$$m_B = 1\ 11010101_2 = -\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256}\right) = -\frac{128+64+16+4+1}{256} = -\frac{213}{256},$$

порядок числа B дорівнює $E_B = 1\ 0011_{\text{пр}} = -3$.

Згідно (1.15) $B = -\frac{213}{256}\cdot 2^{-3} = -\frac{213}{256}\cdot\frac{1}{8} = -\frac{213}{2048} = -0{,}104003906 \approx -0{,}104$.■

В багатьох програмних (компілятори різних мов програмування) і апаратних (мікропроцесори загального та спеціального призначення) реалізаціях арифметичних операцій використовується стандарт IEEE 754 (*IEEE - Institute of Electrical and Electronics Engineers, Інститут інженерів з електротехніки та електроніки*), що описує формат представлення чисел з плаваючою комою.

Формат запису тридцятидворозрядних чисел одинарної точності за стандартом IEEE 754 передбачає три поля: S (1 розряд) для зберігання знака, Es (8 розрядів) для зміщеної експоненти, mr (23 розряди) для залишку мантиси.

S – біт знака, якщо $S = 0$, число що перетворюється додатне, $S = 1$ – від'ємне число. Es – зміщена на 127 експонента (завжди невід'ємна). Порядок числа визначається як $E = Es - 127$ і може приймати значення в межах −127…+128. Залишок мантиси mr є дробовою частиною мантиси, при цьому мантиса утворюється шляхом додавання одиниці до залишку $m = mr+1$ і може приймати значення $1 \le m < 2$. Число, таким чином, представляється у вигляді

$$A = (-1)^S \cdot (1 + mr / 2^{23}) \cdot 2^{(Es-127)} \qquad (1.16).$$

Приклад 1.25. Визначити значення числа, поданого у форматі IEEE 754. $A_{\text{ieee754}} = 0\ 10000101\ 01110011010000000000000$.

Рішення. Оскільки S=0, аналізоване число більше від нуля. Зміщена експонента $Es = 10000101_2 = 133_{10}$. Порядок $E = Es - 127 = 133 - 127 = 6$.

Залишок мантиси $mr = 01110011010000000000000_2$. Мантиса

$m = mr+1 = 1,01110011010000000000000_2 =$

$$= 1 + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{128} + \frac{1}{256} + \frac{1}{2048} = \frac{2048+512+256+128+16+8+1}{2048} = \frac{2969}{2048}.$$

Згідно з (1.15), (1.16)

$$A = 1\cdot\frac{2969}{2048}\cdot 2^6 = \frac{2969}{2048}\cdot 64 = \frac{2969}{32} = 92{,}78125 .■$$

1.4 Основні комбінаторні співвідношення теорії інформації

В теорії інформації часто доводиться зустрічатися з елементами комбінаторики – розділу математики, у якому вивчаються завдання вибору елементів із заданої множини та їх розташування за заданими правилами. Розглянемо деякі співвідношення, що використовуються при оцінюванні обсягу збереженої та переданої інформації.

Розміщеннями з n елементів по k ($n \geq k$) називають комбінації, кожна з яких складається з k елементів, взятих з n елементів даної множини і які відрізняються між собою або самими елементами, або їх розташування.

Число розміщень обчислюється за формулою

$$A_n^k = \frac{n!}{(n-k)!}. \qquad (1.17)$$

де $n! = 1 \cdot 2 \cdot 3 \ldots (n-1) \cdot n$.

Приклад 1.26. Нехай інформація передається за допомогою чотирипозиційної фазової модуляції (QPSK), що використовується, наприклад, у стандарті цифрового супутникового мовлення DVB-S2. У цьому випадку кожен символ може набувати одного з чотирьох значень. Визначити, скільки можна скласти пар із двох різних символів.

Рішення. У даному разі $n=4$, $k=2$. Число різних варіантів (розміщень) згідно з (1.17) складе:

$$A_4^2 = \frac{4!}{(4-2)!} = \frac{4!}{2!} = \frac{4 \cdot 3 \cdot 2 \cdot 1}{2 \cdot 1} = 4 \cdot 3 = 12.$$

Справді, можливі такі пари символів: (1,2), (1,3), (1,4), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,1), (3,2), (4,3). ■

Приклад 1.27. Нехай інформація передається за допомогою восьмипозиційної фазової модуляції (8-PSK). У цьому випадку кожен символ може набувати одного з восьми значень. Визначити, скільки можна скласти трисимвольних слів з символів, що не повторюються.

Рішення. В цьому випадку $n=8$, $k=3$. Число різних розміщень складе:

$$A_8^3 = \frac{8!}{(8-3)!} = \frac{8!}{5!} = \frac{8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 8 \cdot 7 \cdot 6 = 336. \ ■$$

Якщо вибір елементів множини відбувається з поверненням (тобто кожен елемент може бути обраний кілька разів), то число розміщень позначається як B_n^k (кількість *розміщень із повтореннями*) і визначається за такою формулою:

$$B_n^k = n^k. \tag{1.18}$$

Приклад 1.28. Нехай кожен символ послідовності, що передається, може приймати одне з восьми значень. Визначити, скільки можна скласти трисимвольних слів, якщо символи можуть повторюватися.

Рішення. У даному разі n=8, k=3. Число різних розміщень із повтореннями складе:

$$B_8^3 = 8^3 = 512. \blacksquare$$

У практично важливому випадку двійкового алфавіту (n=2) формула (1.18) набуває вигляду

$$B_2^k = 2^k \tag{1.19}$$

Цей вираз визначає загальну кількість можливих k –розрядних двійкових чисел, які можуть набувати значення від 0 до $2^k - 1$.

Приклад 1.29. Так званий код Бодо був розроблений французьким інженером Емілем Бодо в 1870 для винайденого ним телеграфа. Код вводився з клавіатури, що складається з п'яти клавіш (за кількістю пальців), натискання або ненатискання клавіші відповідало одиничному або нульовому значенню одного двійкового розряду (біта) у п'ятибітному коді. Визначити максимально можливу кількість різних символів, що передаються кодом Бодо.

Рішення. Завдання зводиться до визначення кількості можливих п'ятирозрядних бінарних слів. Згідно (1.19)

$$B_2^5 = 2^5 = 32. \blacksquare$$

Приклад 1.30. Літери азбуки Морзе складаються з двох символів – точка та тире. Скільки літер можна отримати, якщо вимагати, щоб кожна літера складалася не менше ніж з одного і не більше ніж з п'яти двійкових символів?

Рішення. Загальна кількість букв азбуки Морзе визначається як сума одно-, двох-, трьох-, чотири- та п'яти символьних комбінацій.

$$N_{морзе} = N_1 + N_2 + N_3 + N_4 + N_5 = B_2^1 + B_2^2 + B_2^3 + B_2^4 + B_2^5 = 2^1 + 2^2 + 2^3 + 2^4 + 2^5 =$$
$$= 2 + 4 + 8 + 16 + 32 = 62$$ ■

Обернення співвідношень (1.18), (1.19) призводить до важливого практичного висновку. Необхідне для запису N різних n-ічних кодів число символів становить

$$k=\log_n N \qquad (1.20)$$

У окремому двійковому випадку (n =2)

$$k=\log_2 N \qquad (1.21)$$

Оскільки зазвичай потрібно, щоб число розрядів було цілим, отримане число часто округляється до найближчого більшого чи рівного

$$k=\lceil \log_2 N \rceil, \qquad (1.22)$$

де $\lceil\ \rceil$ – позначення функції «стеля» (англ. *ceiling*) — округлення числа до найближчого цілого у більший бік. На практиці в таких випадках зазвичай не обчислюють логарифм, а просто підбирають таке мінімальне ціле число, щоб виконувалось співвідношення $2^k \geq N$.

Приклад 1.31. Скільки потрібно п'ятіркових символів, щоб закодувати 2500 різних повідомлень.

Рішення. Оскільки n=5, N=2500, $k=\log_5 2500=4{,}2920$. При необхідності округлити до найближчого цілого числа. $k=\lceil 4{,}2920 \rceil = 5$. ■

Приклад 1.32. Скільки потрібно двійкових розрядів, щоб представити числа від 0 до 1999.

Рішення. У даному разі n=2, N=2000, $k=\log_2 2000=10{,}9658$. При необхідності округлити до найближчого цілого числа. $k=\lceil 10{,}9658 \rceil = 11$.

Цей результат можна отримати наочніше. Оскільки

$2^{10}=1024< N$, а $2^{11}=2048> N$, то число розрядів k = 11. ■

Якщо в (1.17) довжина послідовності збігається з числом різних символів n, тобто $n=k$, то мають місце *перестановки* елементів – комбінації, що складаються з одних і тих же n різних елементів і розрізняються між собою тільки порядком розташування елементів. Число перестановок з n елементів знаходиться за формулою

$$P_n=n!\ . \qquad (1.23).$$

Приклад 1.33. Сім підключених до каналу зв'язку джерел у довільному порядку по черзі відправляють з каналу по одному символу, що відповідає своєму номеру. Визначити можливу кількість різних комбінацій, що передаються в канал.

Рішення. Відповідно до (1.18)

$$P_7=7!=1\cdot2\cdot3\cdot4\cdot5\cdot6\cdot7=5040. \blacksquare$$

Сполученнями з n елементів по k ($n\geq k$) називають комбінації, кожна з яких складається з k елементів, взятих з n даних елементів і що відрізняються між собою хоча б одним елементом. Число сполучень обчислюється за формулою

$$C_n^k=\frac{n!}{(n-k)!k!}, \tag{1.24}$$

причому величини C_n^k називаються також біноміальними коефіцієнтами.

Приклад 1.34. Відомо, що під час передачі шестирозрядної двійкової кодової комбінації сталося дві помилки. Визначити кількість можливих варіантів розташування помилок.

Рішення. Як вибрані елементи служать номери розрядів, в яких відбулися помилки. Кількість різних варіантів визначається як число сполучень при n =6, k=2.

$$C_6^2=\frac{6!}{(6-2)!2!}=\frac{6\cdot5\cdot4\cdot3\cdot2\cdot1}{(4\cdot3\cdot2\cdot1)\cdot(2\cdot1)}=\frac{6\cdot5}{2\cdot1}=15.$$

Дійсно, можливі місця розташування двох помилок у шестирозрядному слові визначаються списком: (1,2), (1,3), (1,4), (1,5), (1,6) (2,3), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6), (5,6). ■

Приклад 1.35. При передачі інформації часто використовуються так звані коди з постійною вагою, що містять однакове число одиниць в кожній кодовій комбінації. Визначити, скільки існує десятирозрядних кодів із вагою чотири.

Рішення. Як вибрані елементи служать номери розрядів, в яких розміщені одиниці. Таким чином, n =10, k=4.

$$C_{10}^4=\frac{10!}{(10-4)!4!}=\frac{10\cdot9\cdot8\cdot7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1}{(6\cdot5\cdot4\cdot3\cdot2\cdot1)\cdot(4\cdot3\cdot2\cdot1)}=\frac{10\cdot9\cdot8\cdot7}{4\cdot3\cdot2\cdot1}=210. \blacksquare$$

1.5 Основні поняття теорії ймовірностей та математичної статистики

Оскільки при передачі та зберіганні масивів інформації можуть відбуватися неконтрольовані та заздалегідь непередбачувані події, при вирішенні завдань теорії інформації широко використовується апарат теорії ймовірностей та математичної статистики. Тож нагадаємо деякі основні поняття та визначення цього розділу математики.

Випадковою подією називається подія, яка при здійсненні деяких умов може статися або не статися. Наприклад, під час передачі каналом зв'язку символ може спотворитися або залишитися незмінним.

Якщо провести N дослідів, то в частині їх випадкова подія відбудеться (сприятливий результат досвіду). Нехай M – кількість спостережень, у яких подія сталася, N – загальна кількість спостережень. Тоді величину W_A

$$W_A = \frac{M}{N}, \tag{1.25}$$

називають частотою реалізації події A.

Границя, до якої прагне відношення M/N при необмеженому зростанні числа дослідів N, називається *ймовірністю* $P(A)$ випадкової події A

$$P(A) = \lim_{N\to\infty} \frac{M}{N} \tag{1.26}$$

Ймовірність має такі властивості.

1. Ймовірність достовірної події дорівнює 1.
2. Ймовірність неможливої події дорівнює 0.
3. Ймовірність випадкової події А задовольняє подвійній нерівності

$$0 \le P(A) \le 1. \tag{1.27}$$

Повною групою подій називається система випадкових подій така, що в результаті здійсненого випадкового експерименту неодмінно відбудеться

одне і лише одне з них. Сума ймовірностей всіх подій у групі завжди дорівнює одиниці.

Протилежною (або додатковою) до події A називається подія $\overline{A}$, що полягає в тому, що A в результаті експерименту не відбулося. Якщо ймовірність події A становить $P(A)$, то ймовірність протилежної події дорівнюватиме $P(\overline{A}) = 1 - P(A)$.

Умовною ймовірністю події називається ймовірність наступу події B за умови, що подія A вже настала. Позначається умовна ймовірність: $P(B/A)$ або $P_A(B)$.

Розрахунки ймовірності складної події через ймовірності найпростіших подій базуються на використанні основних теорем теорії ймовірностей.

Теорема складання ймовірностей несумісних подій. Ймовірність наступу однієї з двох несумісних подій дорівнює сумі їх ймовірностей, тобто

$$P(A+B)=P(A)+P(B). \qquad (1.28)$$

Приклад 1.36. Джерело передає до каналу символи, що відповідають числам від одного до п'яти. Ймовірності появи символів такі:

$P(X=1) = 0{,}1$; $P(X=2)=0{,}2$; $P(X=3)=0{,}3$; $P(X=4)=0{,}3$; $P(X=5)=0{,}1$.

Визначити ймовірність того, що передане в канал значення більше двох.

Рішення. У канал може одночасно передаватися лише один символ, тому події є несумісними. Можливі значення, більші за два, – це три, чотири та п'ять. Тоді

$P(X>2) = P(X=3)+P(X=4)+P(X=5) = 0{,}3+0{,}3+0{,}1=0{,}7$. ■

Теорема множення ймовірностей. Ймовірність спільного наступу подій A і B, дорівнює добутку ймовірності одного з них на умовну ймовірність іншого

$$P(AB)=P(A)\cdot P(B/A) \text{ или } P(AB)=P(B)\cdot P(A/B). \qquad (1.29)$$

Наслідок. Ймовірність добутку двох незалежних подій дорівнює добутку їх ймовірностей

$$P(AB)=P(A)\cdot P(B). \qquad (1.30)$$

Приклад 1.37. Джерело передає в канал незалежні один від одного символи 1 та 0 з ймовірностями відповідно 0,6 та 0,4. Визначити можливість передачі в канал чотирьох одиниць поспіль.

Рішення. Нехай A – подія, що полягає у появі одиниці в каналі. Оскільки події незалежні, для обчислення ймовірностей доцільно застосувати формулу (1.30). Ймовірність появи двох одиниць $P(AA)=P(A)\cdot P(A)= P(A)^2$. Аналогічно, оскільки для появи трьох одиниць поспіль після двох одиниць має з'явитися ще одна, то ймовірність цієї події $P(AAA) = P(AA)\cdot P(A) = P(A)^3$.

Вочевидь, що можливість появи n одиниць поспіль $P(nA) = P(A)^n$.

Зокрема, ймовірність появи чотирьох одиниць поспіль

$P(AAAA) = P(AAA)\cdot P(A) = P(A)^4. = 0{,}6^4 = 0{,}1296$. ■

Приклад 1.38. Джерело передає в канал незалежні один від одного символи 1 та 0 з ймовірностями відповідно 0,3 та 0,7. Визначити можливість передачі в канал комбінації 101.

Рішення. Нехай A – подія, що полягає у появі одиниці в каналі, B – подія, що полягає у появі в каналі нуля. Відповідно до формули (1.30).

$P(A\,B\,A)=P(A)\cdot P(B)\cdot P(A)= 0{,}3\cdot 0{,}7\cdot 0{,}3 = 0{,}063$. ■

Приклад 1.39. Джерело передає в канал незалежні один від одного посилки 111 та 000 з ймовірностями відповідно 0,7 та 0,3. При передачі каналом зв'язку кожен розряд незалежно від інших може змінювати своє значення на протилежне з ймовірністю 0,2. Визначити ймовірність прийому каналу комбінації 100.

Рішення. Нехай A – подія, що складається в передачі в канал комбінації 111, B – подія, що складається в передачі в канал комбінації 000, C – подія, що полягає в спотворенні одного окремого біта, відповідно $\overline{C}$ – протилежна подія, що полягає в тому, що біт залишиться неспотвореним. D – подія, яка полягає в тому, що прийнято комбінацію 100. Ймовірність подій згідно з умовою становить:

$P(A) = 0{,}7$; $P(B) = 0{,}3$; $P(C) = 0{,}2$; $P(\overline{C}) = 1 - P(C) = 0{,}8$.

Комбінація 100 на приймальній стороні може бути отримана у двох випадках: була передана комбінація 000 і спотворений перший біт (другий

і третій не змінилися) або була передана комбінація 111 і спотворені другий і третій біти, а перший залишився незмінним.

Згідно з формулами (1.29), (1.30) умовна ймовірність першого випадку

$$P_1 = P(B)\cdot P(D/B) = P(B)\cdot P(C)\cdot P(\overline{C})\cdot P(\overline{C}) = P(B)\cdot P(C)\cdot P(\overline{C})^2 =$$

$$= 0{,}3\cdot 0{,}2\cdot 0{,}8^2 = 0{,}0384.$$

Ймовірність другого випадку

$$P_2 = P(A)\cdot P(D/A) = P(A)\cdot P(C)\cdot P(C)\cdot P(\overline{C}) = P(B)\cdot P(C)^2\cdot P(\overline{C}) =$$

$$= 0{,}7\cdot 0{,}2^2\cdot 0{,}8 = 0{,}0224.$$

Загальна ймовірність складе

$P_{100} = P_1 + P_2 = 0{,}0384+0{,}0224 = 0{,}0608.$ ■

Приклад 1.40. У лінію зв'язку передаються семирозрядні кодові комбінації. Можливість спотворення окремого біта p=0,2. Знайти можливість безпомилкової передачі всієї комбінації, можливість передачі з однією, двома і трьома помилками.

Рішення. Подія, що полягає у неспотвореній передачі окремого розряду, є протилежною по відношенню до спотворення розряду, тому її ймовірність q=1–p = 1–0,2 = 0,8. Згідно прикладу 1.37, ймовірність того, що жоден розряд не буде спотворений, становить $P_0 = q^N$, де N – довжина кодової комбінації. У цьому випадку $P_0 = 0{,}8^7 = 0{,}2097$.

При розрахунку ймовірності k-кратної помилки слід врахувати, що ця ймовірність дорівнюватиме добутку ймовірності спотворення k біт на ймовірності безпомилкової передачі інших N–k розрядів і коефіцієнт, що враховує число можливих варіантів появи k спотворених біт у N-розрядному слові і дорівнює кількості *сполучень* з N по k

$$P_k = C_N^k \cdot p^k \cdot q^{(N-k)}. \qquad (1.31)$$

Зокрема, ймовірність спотворення олного розряду:

$$P_1 = C_7^1 \cdot 0{,}2^1 \cdot 0{,}8^{(7-1)} = 7\cdot 0{,}2\cdot 0{,}2621 = 0{,}3669.$$

Ймовірність спотворення двох та трьох розрядів відповідно:

$$P_2 = C_7^2 \cdot 0{,}2^2 \cdot 0{,}8^{(7-2)} = \frac{7!}{5!\cdot 2!}\cdot 0{,}2^2\cdot 0{,}8^5 = 21\cdot 0{,}04\cdot 0{,}3277 = 0{,}2753.$$

$$P_3 = C_7^3 \cdot 0{,}2^3 \cdot 0{,}8^{(7-3)} = \frac{7!}{4!3!} \cdot 0{,}2^3 \cdot 0{,}8^4 = 35 \cdot 0{,}008 \cdot 0{,}4096 = 0{,}1147. \blacksquare$$

Приклад 1.41. Джерело має передати в канал у довільному порядку чотири одиниці та три нулі. Визначити ймовірність того, що перші два символи в каналі будуть 10.

Рішення. Нехай A – подія, що полягає у появі одиниці в каналі, B – подія, що полягає у появі в каналі нуля.

Оскільки загальна кількість символів, що передаються сім, серед яких чотири одиниці, то ймовірність одиничного значення першого символу $P(A) = \frac{4}{7.}$ Після першого символу ситуація змінюється: для передачі залишається три нулі і три одиниці. Тоді умовна ймовірність появи на другій позиції нуля $P(B/A) = \frac{3}{6.}$

Згідно (1.29) $P(AB) = P(A) \cdot P(B/A) = \frac{4}{7} \cdot \frac{3}{6} = \frac{2}{7}. \blacksquare$

Теорема складання ймовірностей сумісних подій. Ймовірність наступу хоча б однієї з двох спільних подій дорівнює сумі їх ймовірностей мінус ймовірність їхньої спільної появи, тобто

$$P(A+B)=P(A)+P(B)-P(AB). \qquad (1.32)$$

Якщо події незалежні, (1.32) набуває вигляду

$$P(A+B)=P(A)+P(B)-P(A)\cdot P(B). \qquad (1.33)$$

Приклад 1.42. Два джерела незалежно один від одного передають у канал по одному бінарному символу. Для першого джерела ймовірність передачі канал одиниці p_1=0,7, для другого p_2=0,6. Визначити ймовірність того, що в канал буде передана хоча б одна одиниця.

Рішення. Відповідно до (1.33) ймовірність передачі хоча б однієї одиниці

$p_{12} = p_1+p_2-p_1\cdot p_2$=0,7+0,6–0,7·0,6 = 0,88.

Якщо має місце кілька джерел подій, зручніше визначати ймовірність появи хоча б однієї з незалежних подій A_1, A_2,...A_n, як різницю між одиницею і добутком ймовірностей протилежних подій

$$P(A) = 1 - P(\bar{A}_1) \cdot P(\bar{A}_1) \cdot \ldots \cdot P(\bar{A}_n) = 1 - (1 - P(A_1)) \cdot (1 - P(A_2)) \cdot \ldots \cdot (1 - P(A_n)) \qquad (1.34)$$

Приклад 1.43. Повідомлення проходить через канал зв'язку, що складається із трьох ділянок. Ймовірність помилки на першій ділянці p_1 = 0,1, на другій p_2 = 0,2, на третій p_3 = 0,15. Визначити ймовірність помилки в каналі та безпомилкового проходження повідомлення.

Рішення. Помилка матиме місце, якщо відбудеться спотворення на будь-якій дільниці каналу зв'язку. Тоді згідно (2.19):

$P_{\text{пом}} = 1–(1–p_1)\cdot(1–p_2)\cdot(1–p_2) = 1–0{,}9\cdot0{,}8\cdot0{,}85 = 1–0{,}612 = 0{,}388.$

Ймовірність безпомилкової передачі $P_{\text{бп}} = 1–0{,}388 = 0{,}612$. ■

Випадкова величина – величина, яка в результаті випробування може приймати деяке числове значення, що залежить від випадкових факторів і наперед непередбачуване.

Випадкова величина може бути дискретною чи неперервною.

Дискретна випадкова величина – випадкова величина, яка може приймати значення тільки з кінцевої чи лічильної множини дійсних чисел.

Неперервна випадкова величина – випадкова величина, яка може набувати будь-яких значень з кінцевого або нескінченного інтервалу.

Основною функцією кожної випадкової величини, з якої випливають усі основні характеристики, є закон розподілу випадкової величини – функція, що встановлює зв'язок між можливими значеннями випадкових величин та відповідними їм ймовірностями.

Закон розподілу дискретної випадкової величини задається рядом розподілу, тобто таблицею

x_i	x_1	x_2	…	x_{N-1}	x_N
$P(x_i)$	p_1	p_2	…	p_{N-1}	p_N

у якій x_1, x_2,... x_{N-1}, x_N – значення дискретної випадкової величини X, а p_1, p_2,... p_{N-1}, p_N – відповідні цим значенням ймовірності. При цьому має виконуватись умова $\sum_{i=1}^{N} p_i = 1.$

Математичне сподівання – середнє (зважене ймовірно можливих значень) значення випадкової величини. Для дискретних величин воно обчислюється за такою формулою

$$M(X) = \sum_{i=1}^{N} x_i \cdot p_i = x_1 \cdot p_1 + x_2 \cdot p_2 + ... + x_N \cdot p_N, \quad (1.35)$$

де x_1, x_2 ,..., x_N – можливі значення випадкової величини, p_1, p_2,...p_N – їх ймовірності.

Дисперсія характеризує міру розсіювання можливих значень випадкової величини навколо її математичного сподівання. Для дискретних величин дисперсія визначається як

$$D(X) = \sum_{i=1}^{N} p_i \cdot (x_i - M(X))^2. \quad (1.36)$$

Невід'ємний корінь із дисперсії випадкової величини називається *середньоквадратичним відхиленням*

$$\sigma_x = \sqrt{D(X)}. \quad (1.37)$$

Приклад 1.44 З каналу приймаються символи 0 та 1. Після першої появи символу 0 прийом припиняється. Ймовірність появи одиниці при прийомі p_1=0,4. Приймається не більше п'яти символів. Визначити закон розподілу випадкової величини, що дорівнює кількості переданих символів. Обчислити математичне сподівання $M(X)$, дисперсію $D(X)$ та середньоквадратичне відхилення $\sigma(X)$ величини X.

Рішення. Ймовірність появи нуля при прийомі p_0=1–p_1=0,6.

Відповідно до умов завдання з каналу може бути прийнято шість варіантів послідовностей s.

1. s=0. В цьому випадку довжина послідовності складе X=1;
2. s=10, X=2;
3. s=110, X=3;
4. s=1110, X=4;
5. s=11110, X=5;
6. s=11111, X=5;

Розрахуємо ймовірності можливих результатів:

$P(X=1)= p_0$=0,6; $P(X=2) = p_1 \cdot p_0$=0,4·0,6=0,24;

$P(X=3) = p_1 \cdot p_1 \cdot p_0$=0,4·0,4·0,6=0,096;

$P(X=4) = p_1 \cdot p_1 \cdot p_1 \cdot p_0$=0,4·0,4·0,4·0,6=0,0384.

Довжина послідовності п’ять може бути досягнена в двох випадках, тому розглядаємо кожен випадок окремо і підсумовуємо ймовірність.

$P(X=5) = p_1 \cdot p_1 \cdot p_1 \cdot p_1 \cdot p_0 + p_1 \cdot p_1 \cdot p_1 \cdot p_1 \cdot p_1 = p_1 \cdot p_1 \cdot p_1 \cdot p_1 \cdot (p_0 + p_1) =$

$= p_1 \cdot p_1 \cdot p_1 \cdot p_1 \cdot 1 = p_1 \cdot p_1 \cdot p_1 \cdot p_1 = 0{,}4 \cdot 0{,}4 \cdot 0{,}4 \cdot 0{,}4 = 0{,}0256.$

Розподіл ймовірностей можна подати у вигляді табл.1.6.

Таблиця 1.6 – Розподіл ймовірностей довжин послідовностей X

x_i	1	2	3	4	5
$P(x_i)$	0,6	0,24	0,096	0,0384	0,0256

Перевірка: $\sum_{i=1}^{5} x_i = 0{,}6 + 0{,}24 + 0{,}096 + 0{,}0384 + 0{,}0256 = 1.$

Згідно формулам (2.20)-(2.22):

$$M(X) = \sum_{i=1}^{5} x_i \cdot p_i = 1 \cdot 0{,}6 + 2 \cdot 0{,}24 + 3 \cdot 0{,}096 + 4 \cdot 0{,}0384 + 5 \cdot 0{,}0256 = 1{,}6496.$$

$$D(X) = \sum_{i=1}^{5} p_i \cdot (x_i - M(X))^2 = 0{,}6 \cdot (1 - 1{,}6496)^2 + 0{,}24 \cdot (2 - 1{,}6496)^2 +$$

$$+ 0{,}096 \cdot (3 - 1{,}6496)^2 + 0{,}0384 \cdot (4 - 1{,}6496)^2 + 0{,}0256 \cdot (5 - 1{,}6496)^2 = 0{,}9572.$$

$\sigma_x = \sqrt{D(X)} = \sqrt{0{,}9572} = 0{,}9784.$ ■

Приклад 1.45 У канал зв'язку передаються числа від нуля до трьох, закодовані двобітними посилками. Перший біт у кожній посилці приймає одиничне значення з ймовірністю 0,6, нульове з ймовірністю 0,4. Другий біт збігається з першим із ймовірністю 0,7, не збігається – з ймовірністю 0,3. Визначити закон розподілу випадкової величини, що відповідає двобитному числу, що передається. Обчислити математичне сподівання $M(X)$, дисперсію $D(X)$ та середньоквадратичне відхилення $\sigma(X)$ величини X.

Рішення. Нехай A – подія, що складається в передачі першим одиничного символу, $\overline{A}$ – нульового символу, B – подія, що складається у передачі другим символу, що збігається з першим, $\overline{B}$ – у передачі другим символу, відмінного від першого. Розрахуємо ймовірність всіх двобітових посилок. Згідно (1.29):

$$P_{00} = P(\overline{A}) \cdot P(B) = 0{,}4 \cdot 0{,}7 = 0{,}28;$$

$$P_{01} = P(\overline{A}) \cdot P(\overline{B}) = 0{,}4 \cdot 0{,}3 = 0{,}12;$$

$$P_{10} = P(A) \cdot P(\overline{B}) = 0{,}6 \cdot 0{,}3 = 0{,}18;$$

$$P_{10} = P(A) \cdot P(B) = 0{,}6 \cdot 0{,}7 = 0{,}42.$$

Розподіл ймовірностей можна подати у вигляді табл.1.7.

Таблиця 1.7 – Розподіл ймовірностей двобітових посилок

Код	00	01	10	11
x_i	0	1	2	3
$P(x_i)$	0,28	0,12	0,18	0,42

Згідно формулам (1.35)-(1.37):

$$M(X) = \sum_{i=1}^{4} x_i \cdot p_i = 0 \cdot 0{,}28 + 1 \cdot 0{,}12 + 2 \cdot 0{,}18 + 3 \cdot 0{,}42 = 1{,}74.$$

$$D(X) = \sum_{i=1}^{4} p_i \cdot (x_i - M(X))^2 = 0{,}28 \cdot (0 - 1{,}74)^2 + 0{,}12 \cdot (1 - 1{,}74)^2 +$$

$$+ 0{,}18 \cdot (2 - 1{,}74)^2 + 0{,}42 \cdot (3 - 1{,}74)^2 = 1{,}5924.$$

$$\sigma_x = \sqrt{D(X)} = \sqrt{1{,}5924} = 1{,}2619. \blacksquare$$

При аналізі безперервних випадкових величин використовують два типи законів розподілу: *інтегральний* (*функція розподілу*) та *диференціальний* (*щільність розподілу*).

Функція розподілу $F(x)$ – функція, що визначає для всіх дійсних x ймовірність те, що випадкова величина X приймає значення не більше, ніж x

$$F(x) = P(X \le x). \qquad (1.38)$$

Функція розподілу $F(x)$ має такі властивості (рис. 1.1, *а*).

1. Її ордината, що відповідає довільній точці x_1, являє собою ймовірність того, що випадкова величина X буде меншою, ніж x_1, тобто $F(x_1) = P(X \le x_1)$.

2. Функція розподілу приймає значення, укладене між нулем та одиницею: $0 \le F(x) \le 1)$

3. Функція розподілу прагне до нуля при необмеженому зменшенні x і прагне до одиниці при необмеженому зростанні x, тобто $F(-\infty)=0; F(\infty)=1.$

4. Функція розподілу є монотонно зростаючою кривою, тобто якщо $x_2>x_1$, то $F(x_2)>F(x_1)$.

5. Її збільшення на довільному відрізку (x_1; x_2) дорівнює ймовірності того, що випадкова величина X потрапить в даний інтервал: $F(x_2)-F(x_1)=P(X\le x_2)-P(X\le x_1)=P(x_1<X\le x_2)$.

Функція розподілу дискретної випадкової величини є ступінчастою функцією зі стрибками в точках x_1, x_2,…x_n (рис 1.1, *б*).

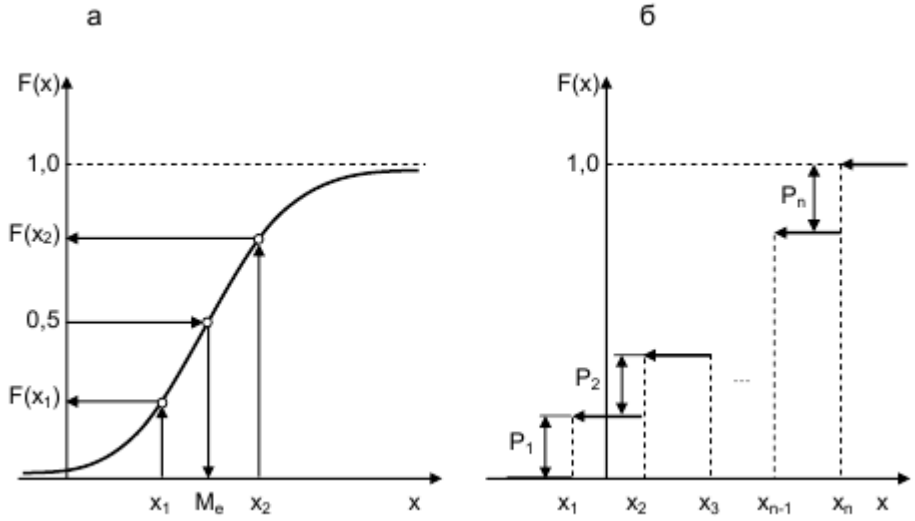


Рисунок 1.1 – Інтегральний закон розподілу:
а – неперервної випадкової величини;
б – дискретної випадкової величини

Щільність розподілу $f(x)$ – перша похідна (якщо існує) функції розподілу

$$f(x)=\frac{dF(x)}{dx}. \qquad (1.39)$$

Щільність розподілу $f(x)$ має такі властивості (рис. 1.2)

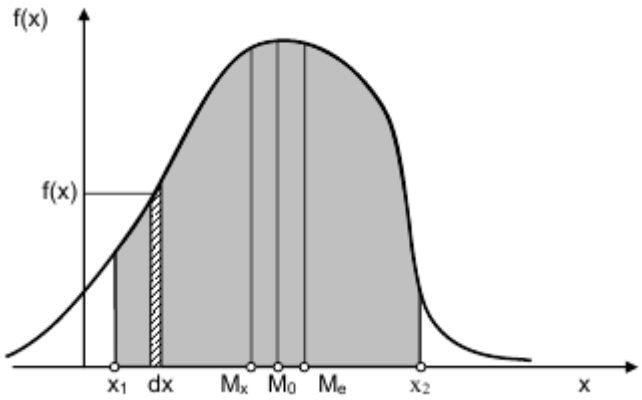


Рисунок 1.2 – Диференціальний закон розподілу $f(x)$

1. Щільність розподілу ймовірностей є невід'ємною функцією, тобто $f(x) \geq 0$. Ця властивість справедлива, оскільки $F(x)$ є незменшною функцією.

2. Функція розподілу випадкової величини X дорівнює визначеному інтегралу від щільності розподілу ймовірностей у межах $(-\infty, x)$, $F(x) = \int_{-\infty}^{x} f(x)dx$.

3. Ймовірність події, яка полягає в тому, що випадкова величина X набуде значення, укладеного в напівінтервалі $[x_1, x_2]$, дорівнює визначеному інтегралу від щільності розподілу ймовірностей на цьому напівінтервалі

$$P(x_1 < X < x_2) = F(x_2) - F(x_1) = \int_{x_1}^{x_2} f(x)dx. \qquad (1.40)$$

4. Інтеграл щільності розподілу в нескінченно великому інтервалі $(-\infty, +\infty)$ дорівнює одиниці

$$\int_{-\infty}^{+\infty} f(x)dx = P(-\infty < X \leq +\infty) = 1,$$

оскільки попадання випадкової величини в інтервал $-\infty < X < +\infty$ є достовірною подією.

Відомі співвідношення для визначення параметрів розподілу безперервної випадкової величини. Так, математичне сподівання випадкової величини визначається інтегралом

$$M(X) = \int_{-\infty}^{+\infty} x \cdot f(x)dx, \tag{1.41}$$

де x – значення неперервної випадкової величини X; $f(x)$ – щільність розподілу безперервної випадкової величини.

Дисперсія безперервної випадкової величини визначається виразом

$$D(X) = \sigma_x^2 = \int_{-\infty}^{+\infty} (x - M(X))^2 f(x)dx\,. \tag{1.42}$$

Як і у випадку дискретних випадкових величин, невід'ємне значення квадратного кореня з дисперсії називається середнім квадратичним відхиленням $\sigma_x = \sqrt{D(X)}$.

Існує значна кількість теоретично визначених законів розподілу, але найбільш поширені *нормальний* (*розподіл Гауса*), *рівномірний* та *показовий* (*експоненційний*) закони.

Нормальний розподіл грає важливу роль у багатьох галузях знань. Як випливає з центральної граничної теореми теорії ймовірностей, випадкова величина підпорядковується нормальному закону розподілу, коли вона піддається впливу великої кількості випадкових факторів, що є типовою ситуацією в аналізі даних.

Нормальний закон розподілу випадкової величини визначається щільністю розподілу

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x-M(X))^2}{2\sigma_x^2}}, \tag{1.43}$$

де $M(X)$ – математичне сподівання, σ_x^2 – дисперсія.

Рівномірний розподіл – розподіл випадкової величини, що приймає значення, в деякому проміжку кінцевої довжини. Для рівномірного розподілу характерно, що щільність ймовірності на цьому проміжку постійна

$$f(x) = \begin{cases} \dfrac{1}{b-a}, x \in [a,b] \\ 0, x \notin [a,b] \end{cases}. \tag{1.44}$$

Числові характеристики рівномірного розподілу визначаються співвідношеннями: $M(X)=\frac{a+b}{2}; D(X)=\frac{(b-a)^2}{12}$.

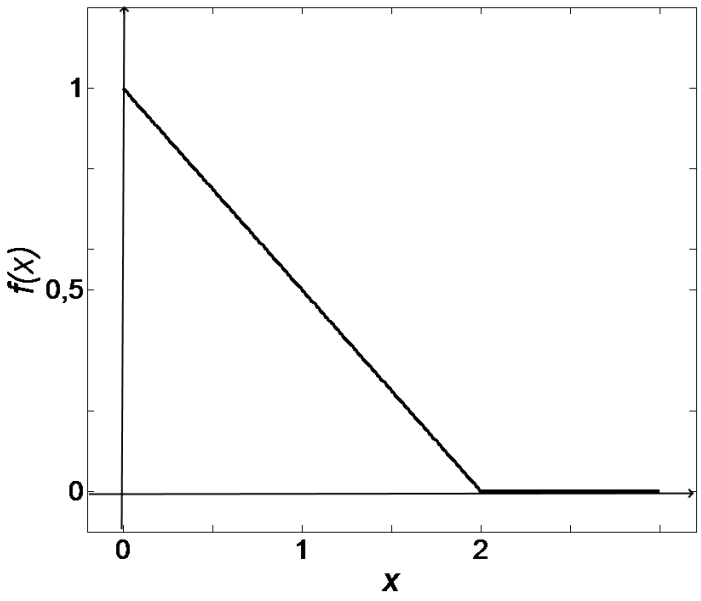
Безперервна випадкова величина X має експоненційний (показовий) закон розподілу з параметром λ, якщо її щільність ймовірності має вигляд

$$f(x)=\begin{cases} 0, x<0, \\ \lambda e^{-\lambda x}, x\geq 0. \end{cases} \tag{1.45}$$

Математичне сподівання експоненційно розподіленої величини становить $M(X)=\frac{1}{\lambda}$, дисперсія $D(X)=\frac{1}{\lambda^2}$.

Експоненційний розподіл відіграє важливу роль у завданнях телекомунікацій, оскільки дозволяє моделювати інтервали часу між настанням подій, наприклад надходженням пакетів інформації в канал зв'язку.

Приклад 1.46. Щільність розподілу $f(x)$ випадкової величини задано графіком (рис 1.3). Знайти аналітичний вираз для щільності та функції розподілу $F(x)$. Визначити математичне сподівання $M(X)$, дисперсію $D(X)$ та середньоквадратичне відхилення $\sigma(X)$ величини X.



Риснок 1.3 – Щільність розподілу $f(x)$ випадкової величини X

Рішення. З графіка видно, що при $x \in [0,2]$ щільність розподілу $f(x)$ являє собой відрізок прямої, що проходить через дві точки з координатами (0, 1) і (2, 0). Використовуючи рівняння такої прямої, $\frac{x-x_1}{x_2-x_1}=\frac{y-y_1}{y_2-y_1}$, отримаємо: $\frac{x-0}{2-0}=\frac{y-1}{0-1}, y=1-\frac{x}{2}$

Звідси випливає

$$f(x)=\begin{cases}0, x<0\\ 1-\frac{x}{2},\ 0\le x\le 2\\ 0, x>2.\end{cases}$$

Визначимо функцію розподілу $F(x)$.

$F(x)=\int_{-\infty}^{x} f(x)dx=\int_{0}^{x}(1-\frac{x}{2})dx=x-\frac{x^2}{4}$ Таким чином функція розподілу випадкової величини X:

$$F(x)=\begin{cases}0, x<0\\ x-\frac{x^2}{4},\ 0\le x\le 2.\\ 1, x>2.\end{cases}$$

Графік функції $F(x)$ зображено на рис.1.4.

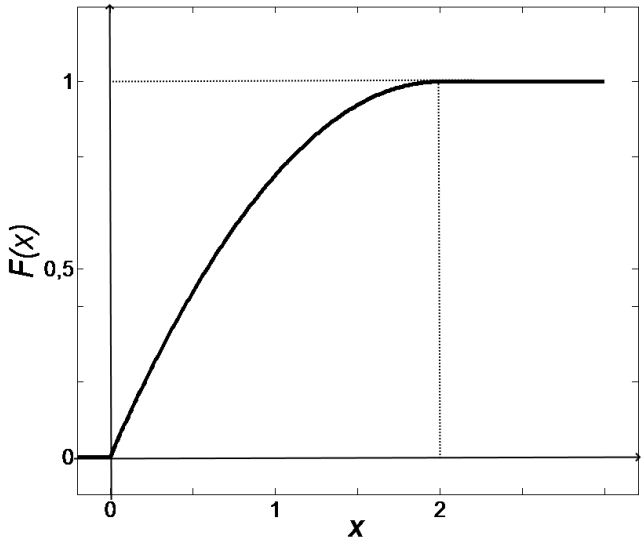


Рисунок 1.4 – Функція розподілу (інтегральний закон розподілу) випадкової величини X

Визначимо математичне сподівання $M(X)$), дисперсію $D(X)$, та середньоквадратичне відхилення $\sigma(X)$ випадкової величини X.

Згідно (1.40), (1.41)

$$M(X)=\int_{-\infty}^{+\infty} x\cdot f(x)dx=\int_{0}^{2} x\cdot(1-\frac{x}{2})dx=\int_{0}^{2}\left(x-\frac{x^2}{2}\right)dx=\left(\frac{x^2}{2}-\frac{x^3}{6}\right)\Bigg|_0^2=2-\frac{8}{6}=\frac{2}{3}.$$

$$D(X)=\int_{-\infty}^{+\infty}(x-M(X))^2 f(x)dx=\int_{0}^{2}\left(x-\frac{2}{3}\right)^2\cdot\left(1-\frac{x}{2}\right)dx=\int_{0}^{2}\left(x^2-\frac{4}{3}x+\frac{4}{9}\right)\cdot\left(1-\frac{x}{2}\right)dx=$$

$$=\int_{0}^{2}\left(-\frac{x^3}{2}+\frac{5}{3}x^2-\frac{14}{9}x+\frac{4}{9}\right)dx=\left(-\frac{x^4}{8}+\frac{5}{9}x^3-\frac{7}{9}x^2+\frac{4}{9}x\right)\Bigg|_0^2=$$

$$=-\frac{16}{8}+\frac{40}{9}-\frac{28}{9}+\frac{8}{9}=\frac{2}{9}=0{,}2222$$

$$\sigma_x=\sqrt{D(X)}=\sqrt{\frac{2}{9}}=\frac{\sqrt{2}}{3}=0{,}4714.\ \blacksquare$$

Завдання для самостійного вирішення

1. Перевести число MMDCCLIX з римської системи до десяткової.

2. Перевести числа 479, 1371, 3439 із десяткової системи в римську.

3. Перевести числа $A_{\text{фіб}}$=100010100000101, $B_{\text{фіб}}$=101000101010100101 з Фібоначчієвої системи до десяткової.

4. Перевести числа 442, 503, 1122 з десяткової системи в Фібоначчієву.

5. Перевести числа A=100110001011_2, B=101101100101100_2, C=1110011100010111102 із двійкової системи до десяткової.

6. Перевести числа 442, 1321, 5148 з десяткової системи в двійкову, вісімкову, шістнадцяткову.

7. Перевести числа 341, 1277, 3070 з десяткової системи до трійкової та систем з основою п’ять та сім.

8. Перевести числа 573, 2365, 4036 з вісімкової системи до двійкової, шістнадцяткової, десяткової.

9. Перевести числа E86, 7A3D, 4B0FC з шістнадцяткової системи в двійкову, вісімкову, десяткову.

10. Перевести числа 1947, 23488, 560112 з десяткової системи до двійково-десяткової.

11. Перевести числа A=0101100110000011$_{2\text{-}10}$, B=011000111001011001011000$_{2\text{-}10}$, C=10000111000010011000010000011$_{2\text{-}10}$ з двійково-десяткової системи до десятковою.

12. Для чисел D = 186, E = 311, F = 3768 перевести в прямий, зворотний, додатковий і модифікований додатковий коди числа ± D, ± E, ± F.

13. Обчислити в модифікованому додатковому коді +E+D, +E–D, –E+D, –E-D, +E+F, +E–F, –E+F, –E–F з попереднього завдання, перевірити результати обчислень.

14. Визначити значення двійкових чисел A=0 1011000101 0 00100$_{\text{пк}}$, B=0 1001001110 1 00011$_{\text{пк}}$, C=1 1010001010 1 00111$_{\text{пк}}$, представлених у форматі з плаваючою комою. Число двійкових розрядів виділених для запису числа наступне: знак мантиси – один розряд, мантиса – десять, знак порядку – один, порядок п'ять розрядів. Мантиса та порядок представлені у прямих кодах.

15. Перетворити на формат з плаваючою комою числа +326; +78,19; –921; +0,892; +0,207; +0,0031; –0, 073. Число двійкових розрядів виділених для запису числа наступне: знак мантиси – один розряд, мантиса – вісім, знак порядку – один, порядок чотири розряди. Мантису та порядок подати у прямих кодах.

16. Визначити значення чисел A та B, представлених у форматі IEEE 754.

A_{ieee754} = 0 10001101 01010011100000000000000;

B_{ieee754} = 0 01111011 11010011010000000000000.

17. Скільки можна скласти слів із чотирьох різних п'ятіркових символів, що приймають значення від 0 до 4?

18. Скільки можна скласти слів з чотирьох п'ятіркових символів за умови, що символи можуть повторюватися?

19. Скільки існує різних шістнадцятирозрядних двійкових чисел?

20. Скільки потрібно двійкових розрядів, щоб закодувати числа від 0 до 2999?

21. Скільки потрібно п'ятіркових розрядів, щоб закодувати числа від 0 до 3999?

22. Дев'ять підключених до каналу зв'язку джерел у довільному порядку по черзі відправляють із канал по одному символу, що відповідає

своєму номеру. Визначити можливу кількість різних комбінацій, що передаються в канал.

23. Відомо, що під час передачі десятирозрядної двійкової кодової комбінації сталося чотири помилки. Визначити кількість можливих варіантів розташування помилок.

24. Джерело передає до каналу символи, що відповідають числам від одного до шости. Ймовірності появи символів такі: $P(X=1)=0,2$; $P(X=2)=0,1$; $P(X=3)=0,3$; $P(X=4)=0,1$; $P(X=5)=0,1$; $P(X=6)=0,2$.

Визначити ймовірність того, що число, передане до каналу парне.

25. Джерело передає в канал незалежні один від одного символи 1 та 0 з ймовірностями відповідно 0,3 та 0,7. Визначити ймовірність передачі в канал п'яти одиниць поспіль, за якими слідує нуль.

26. Джерело передає в канал незалежні один від одного посилки 1111 та 0000 з ймовірностями відповідно 0,6 та 0,4. При передачі каналом зв'язку кожен розряд незалежно від інших може змінювати своє значення з нуля на одиницю з ймовірністю 0,3, з одиниці на нуль з ймовірністю 0,2. Визначити ймовірність прийому каналу комбінації 1011.

27. По каналу зв'язку з перешкодами передається одна з двох комбінацій 10101 та 01010, ймовірності передачі цих комбінацій відповідно дорівнюють 0,6 та 0,4. Ймовірність правильного прийому кожного із символів 0 та 1 дорівнює 0,7. Символи спотворюються перешкодами незалежно один від одного. На виході каналу маємо кодову комбінацію 10110. Визначити ймовірність передачі комбінацій 10101 та 01010.

28. По лінії зв'язку посилаються сигнали 1 та 0 з ймовірностями відповідно $p_1=0,4$, $p_0=0,6$. Якщо посилається сигнал 1, то з ймовірностями $r_{11} = 0,8$, $r_{10} =0,2$ приймаються відповідно сигнали 1 і 0. Якщо передається сигнал 0, то з ймовірностями $r_{01}=0,3$, $r_{00}=0,7$ приймаються відповідно сигнали 1 і 0. Яка ймовірність того, що буде прийнята одиниця? Якою є умовна ймовірність того, що посилається сигнал 1 за умови, що приймається сигнал 1?

29. До лінії зв'язку передаються десятирозрядні кодові комбінації. Можливість спотворення окремого біта $p=0,2$. Знайти можливість безпомилкової передачі всієї комбінації, можливість передачі не більш, ніж із чотирма помилками.

30. Джерело має передати в канал у довільному порядку шість одиниць та чотири нулі. Визначити ймовірність того, що перші три символи в каналі будуть 101.

31. Три джерела незалежно один від одного передають у канал по одному двійковому символу. Для першого джерела ймовірність передачі в канал одиниці p_1=0,4, для другого p_2=0,5, для третього p_3=0,7. Визначити ймовірність того, що в канал буде передана хоча б одна одиниця.

32. З каналу приймаються символи 0 і 1. Після появи двох однакових символів поспіль прийом припиняється. Ймовірність появи одиниці p_1=0,6, нуля p_0=0,4. Приймається не більше шести символів. Визначити закон розподілу випадкової величини, що дорівнює кількості переданих символів. Обчислити математичне сподівання $M(X)$, дисперсію $D(X)$ та середньоквадратичне відхилення $\sigma(X)$ величини X

33. До каналу зв'язку передаються числа від нуля до трьох, закодовані двобітними посилками. Кожний біт у посилці приймає одиничне значення з ймовірністю 0,6, нульове з ймовірністю 0,4. У каналі з ймовірністю 0,3 одиниця може перейти у нуль, з ймовірністю 0,2 нуль може перейти в одиницю. Визначити закон розподілу випадкової величини, що відповідає прийнятому двобитному числу. Обчислити математичне сподівання $M(X)$, дисперсію $D(X)$ та середньоквадратичне відхилення $\sigma(X)$ величини X.

34. Закон розподілу (інтегральний закон) $F(x)$ випадкової величини X заданий виразом.

$$F(x)=\begin{cases} 0, x<1 \\ \dfrac{x-1}{3},\ 1\le x\le 4. \\ 1, x>4. \end{cases}$$

35. Знайти аналітичний вираз для щільності ймовірности $f(x)$ та функції розподілу $F(x)$. Обчислити математичне сподівання $M(X)$, дисперсію $D(X)$ та середньоквадратичне відхилення $\sigma(X)$ величини X.

ГЛАВА 2

КІЛЬКІСНІ ХАРАКТЕРИСТИКИ ІНФОРМАЦІЇ. ЕНТРОПІЯ ДЖЕРЕЛА ІНФОРМАЦІЇ

2.1 Критерії вибору міри інформації

Для практичного використання поняття інформації потрібно навчитися її вимірювати. Це можна зробити за аналогією з методикою чисельного виміру інших фундаментальних понять, таких як маса, довжина, сила струму тощо. Для цього потрібно встановити, що приймається за міру кількісної оцінки інформації та як вибрати одиницю виміру цієї міри.

Починаючи з робіт американського вченого Клода Шеннона прийнято вважати, що поняття інформації складається з трьох аспектів: *синтаксичного*, *семантичного* та *прагматичного*. Прагматичний аспект торкається питань впливу інформації на поведінку людей. Семантичний має відношення до сенсу та значення істинності повідомлень. Теорія семантичної інформації досліджує галузь людських знань і є складовою розробки штучного інтелекту. При оцінюванні семантичної інформації зазвичай аналізується *тезаурус* – сукупність відомостей, які має користувач або система.

При побудові технічних засобів зберігання та передачі інформації зазвичай застосовується синтаксична міра інформації. Ця міра використовується для кількісного вираження знеособленої інформації, що не виражає смислового ставлення до об'єктів.

Структурний підхід до вимірювання інформації розглядає будову масивів інформації та їх вимір простим підрахунком інформаційних елементів чи комбінаторним методом. Носіями інформації при цьому виступають *повідомлення* – кінцеві послідовності символів. Символи, зібрані у групи, називаються *словами*. *Алфавітом* джерела повідомлень називається вся множина різних символів, що генеруються джерелом повідомлень. *Об'єм алфавіту* джерела повідомлень – кількість різних символів, що генеруються джерелом повідомлень.

Міра кількісної оцінки інформації I має задовольняти наступним умовам:

1. Повідомленню більшої довжини (при тому самому обсязі алфавіту) відповідає більша кількість інформації.

2. Більша кількість інформації міститься в тих повідомленнях (однакової довжини), які складені із символів більшого алфавіту.

3. Функція $I(X)$ для будь-яких повідомлень X повинна мати властивість адитивності

$$I(X_1) + I(X_2) = I(X_1 + X_2). \qquad (2.1)$$

4. Кількість інформації I повинна залежати від ймовірностей тих чи інших повідомлень. Зокрема, інформація про вчинення достовірної події має дорівнювати нулю.

Геометрична міра інформації передбачає вимірювання параметра геометричної моделі інформаційного повідомлення (довжини, площі, обсягу тощо) у дискретних одиницях. Наприклад, геометричною моделлю інформації як однорозрядного слова може бути лінія певної довжини, дворозрядного слова – прямокутник, трирозрядного – паралелепіпед. Зокрема, у разі застосування двійкового алфавіту, коли символи набувають значення 0 або 1, однорозрядне слово представляється лінією одиничної довжини (рис 2.1, *а*), дворозрядне слово – квадратом (рис. 2.1, *б*), трирозрядне слово – кубом (рис 2.1, *в*).

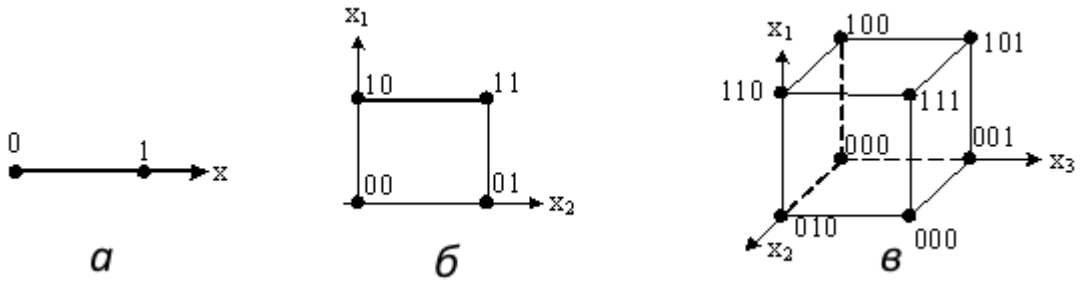


Рисунок 2.1 – Геометрична міра інформації:
а – відображення однорозрядного слова, *б* – дворозрядного, *в* – трьохрозрядного

Максимально можлива кількість інформації при геометричній мірі визначається як сума дискретних значень з усіх вимірів (координат).

Комбінаторна міра. В комбінаторній мірі кількість інформації обчислюється як кількість різних комбінацій елементів, що містяться в повідомленні. При цьому можуть використовуватися різні комбінації, розглянуті в розділі 1.4 – розміщення, перестановки, сполучення.

Геометрична та комбінаторна міри малопридатні для оцінювання значних обсягів інформації, крім того для них не завжди виконується умова адитивності (2.1). Тому було введено міру, частково позбавлену цих недоліків.

2.2 Адитивна логарифмічна міра Гартлі

Нехай число можливих випадків події дорівнює N, причому всі випадки рівноймовірні. Тоді отримання конкретного повідомлення рівнозначно випадковому вибору одного з повідомлень N з ймовірністю $1/N$. При цьому, зрозуміло що чим більше N, тим більший ступінь невизначеності характеризує цей вибір і тим більш інформативним вважатимуться повідомлення.

Теоретично мірою інформації могло б служити число N. Однак необхідно наділити цю міру властивістю адитивності (2.1), тобто визначити її так, щоб вона була пропорційна довжині повідомлення. Для забезпечення адитивності як міру невизначеності для джерела з рівноймовірними станами приймають логарифм кількості станів

$$I = \log_a N . \qquad (2.2)$$

Зазначена адитивна міра була запропонована Р. Гартлі в 1928 році для кількісної оцінки здатності системи зберігати або передавати інформацію. Основа логарифму не має принципового значення і визначає лише масштаб чи одиницю невизначеності. Оскільки сучасна інформаційна техніка будується на елементах що мають два стійких стана, зазвичай вибирають основу логарифма, рівну 2

$$I = \log_2 N . \qquad (2.3)$$

Тоді одиницю кількості інформації на один елемент повідомлення називають *двійковою одиницею* чи *бітом*. При цьому одиниця невизначеності (двійкова одиниця чи біт) передбачає можливість вибору з двох подій «0» чи «1» (*bit* – скорочення від англ. *binary digit* – двійкова одиниця, також гра слів: *bit* – шматочок, частка).

Найчастіше алфавіт джерела повідомлень складається з n знаків, кожен із яких може бути елементом повідомлення. Кількість N можливих повідомлень довжини k дорівнює кількості перестановок з необмеженими повтореннями

$$N = n^k . \tag{2.4}$$

Тоді (2.3) набуває вигляду

$$I = \log_2 N = \log_2 n^k = k \log_2 n. \tag{2.5}$$

Теоретично оптимальну щільність зберігання інформації забезпечує система числення з основою e = 2,71828 (число Ейлера). Якщо в якості основи логарифму вибирають e, то інформація вимірюється в натуральних одиницях або *натах*, 1 нат дорівнює $\log_2 e \approx 1{,}4427$ біт.

Оскільки найближчим до числа e цілим є трійка, то робилися спроби створення комп'ютерів на основі троїчної системи числення. Як одиниця зберігання інформації в таких комп'ютерах використовувався *трит* (англ. ***tr**inary dig**it***) – трійковий розряд, що приймає значення 0, +1, –1.

1 трит = $\log_2 3$біт ≈ 1,5850 біт.

Мінімальною адресованою одиницею пам'яті в трійкових комп’ютерах був обраний *трайт*, що дорівнює шести тритам.

Якщо основою логарифму вибрати 10, то кількість інформації вимірюється в *дитах* (інша назва – *гартлі*), 1 дит = $\log_2 10$ біт ≈ 3,3219 біт. 1дит – це кількість інформації, яку можна передати у повідомленні, що складається з одного десяткового знака (арабської цифри).

Для кожної основної одиниці виміру інформації існують похідні більші одиниці виміру. Оскільки найчастіше ми будемо використовувати як основну одиницю біт, розглянемо похідні одиниці виміру для біта. Найбільш популярною одиницею, що використовується на практиці, є

байт. Слово *byte* було обрано як навмисне спотворене *bite*, що вимовляється так само (англ. *bite* – «шматок», «частина чогось, відокремлена за один укус»).

1 байт (1В) = 8 біт. Саме до байта (а не до біта) зазвичай наводяться всі більші обсяги інформації, що обробляються сучасними комп'ютерами.

Кількість станів (кількість різних двійкових чисел), що може приймати один байт, визначається згідно з (2.4)

$$N_B = 2^8 = 256. \qquad (2.6)$$

Слід зазначити, що історично байт визначався для конкретного комп'ютера як мінімальний крок адресації пам'яті, який на деяких комп'ютерах, що існували, не дорівнював 8 бітам (а пам'ять не обов'язково складалася з бітів, наприклад, в трійкових комп'ютерах – з тритів і трайтів). Однак у сучасній традиції байт вважають рівним восьми бітам.

Особливу назву мають чотири двійкові розряди (4 біти) – *тетрада, напівбайт, ніббл* (від англ. *nibble* – «покусувати»), які вміщують кількість інформації, що міститься в одній шістнадцятковій цифрі.

Для вимірювання великих ємностей запам'ятовуючих пристроїв і великих обсягів інформації, служать більші одиниці. При цьому, однак, виникає деяка неоднозначність у використанні термінів. Так, одиниця виміру інформації кілобайт (скор. Кбайт, міжнародне Kbyte, КВ) може позначати як 10^3 (1000), так і 2^{10} (1024) байт. Міжнародна система одиниць СІ та Міжнародна електротехнічна комісія (МЕК) визначили поняття кілобайт як 10^3 байт. При цьому у разі необхідності позначення 2^{10} байт МЕК запропонувала використовувати термін «*кібібайт*».

На практиці найменування «кібібайт» і аналогічні йому не отримали широкого поширення. Для вказівки обсягу пам'яті з двійковою адресацією, такою, як оперативна та флеш-пам'ять; в операційних системах сімейства Windows термін кілобайт використовується переважно у значенні 1024 (2^{10}) байта. Останнім часом, однак, багато виробників флеш-пам'яті та жорстких дисків стали вказувати саме десяткові значення приставок, що дозволяє їм значно економити. Наприклад, один двійковий терабайт (тебібайт) дорівнює 1 ТіВ = 2^{40} = 1 099 511 627 776 байт = 1,0995 ТВ і об'єм, виражений через десяткову приставку, майже на 10% менше вираженого через двійкову.

Таким чином, існує дві лінійки похідних одиниць для байта – лінійка десяткових приставок та лінійка двійкових приставок. У разі десяткових приставок кожна наступна одиниця виміру дорівнює 1000 попередніх одиниць. Позначаються десяткові приставки латинськими літерами (літера префікса із системи СІ та велика «В», що означає «байт»).

При використанні двійкових приставок кожна наступна одиниця вимірювання дорівнює 1024 попередніх одиниць. Прийнято позначати двійкові приставки, записуючи префікс великою літерою і після нього слово «байт» цілком. У стандартах МЕК для позначення двійкових приставок між префіксом і В додається маленька буква *і* (від слова *binary*). Крім того, всі префікси записуються великими літерами. У неофіційній мові замість назви одиниці виміру іноді використовується лише перша літера – 1К замість 1Кбайт, 1М замість 1Мбайт тощо.

У табл. 2.1 зведені одиниці вимірювання великих обсягів інформації, що використовуються.

Таблиця 2.1 – Одиниці виміру інформації

Двійкова приставка			Десяткова приставка	Стандарт МЕК		
Назва	Позначення	Ступінь	Ступінь	Назва	Позначення	Ступінь
байт	Байт, В	2^0	10^0	байт	В	2^0
кілобайт	Кбайт, КВ	2^{10}	10^3	кібібайт	КіВ	2^{10}
мегабайт	Мбайт, МВ	2^{20}	10^6	мебібайт	МіВ	2^{20}
гігабайт	Гбайт, GB	2^{30}	10^9	гібібайт	GiB	2^{30}
терабайт	Тбайт, ТВ	2^{40}	10^{12}	тебібайт	ТіВ	2^{40}
петабайт	Пбайт, РВ	2^{50}	10^{15}	пебібайт	РіВ	2^{50}
ексабайт	Ебайт, ЕВ	2^{60}	10^{18}	ексбібайт	ЕіВ	2^{60}
зетабайт	Збайт, ZB	2^{70}	10^{21}	зебібайт	ZiB	2^{70}
йотабайт	Йбайт, YB	2^{80}	10^{24}	йобібайт	YiB	2^{80}

Природно, що всі зазначені приставки можуть використовуватись і для оцінювання обсягів інформації, виражених у бітах (кілобіт, мегабіт тощо). Термін «кілобіт» найчастіше використовують при вимірі швидкості передачі цифрових ліній зв'язку, як кілобіт на секунду (кбіт/с). Іноді швидкість передачі визначають також у *бодах* – одиницях, що визначають кількість змін інформаційного параметра в секунду. Бод названий на ім'я

Еміля Бодо, винахідника коду Бодо — кодування символів для телетайпів. У бодах висловлюють повну ємність каналу, включаючи службові символи, якщо вони є. У деяких випадках швидкості в бод та біт/с можуть збігатися, але найчастіше це не так.

Приклад 2.1. Загадано число *A* в діапазоні від 1 до 400. Яку мінімальну кількість запитань потрібно поставити, щоби вгадати число. На запитання можна відповідати лише «так» та «ні».

Рішення. Правильна стратегія полягає в тому, щоб щоразу зменшувати кількість можливих варіантів вдвічі, тобто ділити діапазон, що залишається для розгляду, навпіл. Оскільки після кожної відповіді невизначеність зменшується вдвічі, максимально необхідне число питань (ціле) становитиме $N_{max} = \lceil \log_2 400 \rceil = \lceil 8{,}6439 \rceil = 9$.

Нехай, наприклад, вказано число 117.

Питання 1. A> 200? Ні.

Питання 2. A> 100? Так.

Питання 3. A> 150? Ні.

Питання 4. A> 125? Ні.

Питання 5. A> 113? Так.

Питання 6. A> 119? Ні.

Питання 7. A> 116? Так.

Питання 8. A> 118? Ні.

Питання 9. A> 117? Ні.

Таким чином, число, що шукається, більше 116, але не перевищує 117. Цій умові задовольняє тільки число 117.■

Приклад 2.2. Джерело може передавати символи, кожен з яких представляє число в діапазоні від 1 до 3000. Визначити обсяг інформації, що міститься в символі, виражений у бітах, натах, тритах та дитах.

Рішення. Загальна кількість можливих різних символів $N = 3000$.

Згідно (2.2)

I (біт) = $\log_2 3000 = 11{,}5507$ біт;

I (нат) = $\ln 3000 = 8{,}0064$ нат;

I (трит) = $\log_3 3000 = 7{,}2877$ трит;

I (дит) = $\lg 3000 = 3{,}4771$ дит. ■

Приклад 2.3. Нехай інформація передається за допомогою восьмипозиційної фазової модуляції. Визначити, скільки біт інформації міститься в повідомленні з п'яти символів, що не повторюються.

Рішення. В даному випадку число різних повідомлень визначається числом розміщень без повторень з n елементів по k, де n=8, k=5. Відповідно до (1.17) число різних розміщень складе:

$$N = A_8^5 = \frac{8!}{(8-5)!} = \frac{8!}{3!} = \frac{8\cdot 7\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{3\cdot 2\cdot 1} = 8\cdot 7\cdot 6\cdot 5\cdot 4 = 6720.$$

$I = \log_2 N = \log_2 6720 = 12{,}7142$ біт. ■

Приклад 2.4. Визначити, скільки біт інформації міститься в повідомленні з п'яти вісімкових символів, якщо символи можуть повторюватися.

Рішення. У даному разі n =8, k=5. Число різних розміщень із повтореннями складе:

$I = \log_2 N = \log_2 2^{3\cdot 5} = 3\cdot 5\cdot \log_2 2 = 3\cdot 5 = 15$ біт. ■

Приклад 2.5. Визначити, скільки біт інформації міститься у повідомленні із шести п'ятіркових символів

Рішення. У даному разі n =5, k=6.

Загальна кількість можливих різних символів:

$N = 5^6 = 12625.$

$I = \log_2 N = \log_2 15625 = 13{,}9216$ біт. ■

Приклад 2.6. У канал зв'язку передаються дванадцятирозрядні двійкові слова, у кожному з яких міститься рівно три одиниці, що знаходяться на довільних місцях. Визначити, скільки бітів інформації можна передати таким кодом.

Рішення. Кількість різних варіантів визначається числом сполучень при n =12, k=3.

$$N = C_{12}^3 = \frac{12!}{(12-3)!3!} = \frac{12\cdot 11\cdot 10\cdot 9\cdot 8\cdot 7\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{(9\cdot 8\cdot 7\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1)\cdot(3\cdot 2\cdot 1)} = \frac{12\cdot 11\cdot 10}{3\cdot 2\cdot 1} = 220.$$

$I = \log_2 N = \log_2 220 = 7{,}7814$ біт. ■

Приклад 2.7. Три радіолокатори передають інформацію про ціль у полярних координатах. Перший радіолокатор посилає на приймальну станцію інформацію про дальність – ціле число в інтервалі від одиниці до 10000, другий про азимут – ціле число від 0 до 359, третій про кут місця – ціле число від 0 до 89. Скільки інформації міститься у відомостях про координати мети?

Рішення. Кількість варіантів координати дальності становить N_1=10000, азимута N_2=360, кута місця N_3=90. Загальна кількість варіантів координати визначається як $N= N_1 \cdot N_2 \cdot N_3 = 10000 \cdot 360 \cdot 90 = 324000000$.

Кількість інформації $I = \log_2 N = \log_2 324000000 = 28{,}2714$ біт.

Можна оцінити цю кількість з урахуванням властивості адитивності міри Гартлі $I=I_1+I_2+I_3$, де інформація відповідно про дальність, азимут і вугілля місця.

$I_1 = \log_2 10000 = 13{,}2877$ біт;

$I_2 = \log_2 360 = 8{,}4919$ біт;

$I_3 = \log_2 90 = 6{,}4918$ біт.

$I=I_1+I_2+I_3 = 13{,}2877+8{,}4919+6{,}4918 = 28{,}27$ біт, що збігається з результатом, отриманим попереднім методом.

Оскільки повідомлення, що передаються, містять зазвичай ціле число біт, то доцільно округлити результати до найближчого більшого цілого:

$I_ц=I_{1ц}+I_{2ц}+I_{3ц} = \lceil 13{,}2877 \rceil + \lceil 8{,}4919 \rceil + \lceil 6{,}4918 \rceil = 14+9+7 = 30$ біт. ■

Приклад 2.8. Яка кількість інформації міститься у друкованому документі обсягом N_1=250 сторінок. На сторінці тексту можна надрукувати N_2=30 рядків з N_3=62 символами. Кожен символ кодується одним байтом.

Рішення. Число символів на сторінці:

$N_{стор}= N_2 \cdot N_3 = 30 \cdot 62 = 1860$.

Оскільки кожен символ кодується одним байтом, обсяг інформації у байтах збігається із загальним числом знаків у документі:

$I=N=N_1 . N_{стор}$=250·1860 байт=465000 байт=465000/1024=454,1016 Кбайт.■

Приклад 2.9. Яка кількість інформації буде у растровом зображенні формату BMP розміром N_1=1024 на N_2=600 пікселів. Кожен піксель представляється у форматі TrueColor з використанням одного байта (256) рівнів для кожної з трьох компонентів моделі RGB: червоного (R), зеленого (G) та синього (B), що в результаті дає 16777216, тобто 2^{24} різних кольорів. Об'ємом службової інформації, що міститься у файлі BMP, знехтувати.

Рішення. Число пікселів у зображенні

$N_{пікс} = N_1 \cdot N_2 = 1024 \cdot 600 = 614400$ пікселів.

Оскільки кожен символ кодується трьома байтами, обсяг інформації у зображенні становитиме:

$I=3 \cdot N_{пікс}$=3·614400 байт=1843200 байт=1843200/1024=

=1800 Кбайт=1800/1024=1,7578 Мбайт. ■

2.3 Статистична міра Шеннона

Розглянута вище оцінка інформації, так звана міра Гартлі, заснована на припущенні про рівноймовірність всіх знаків алфавіту і не враховує того факту, що ймовірності можуть бути різними. У реальних дискретних повідомленнях символи часто з'являються з різними ймовірностями і, більше того, часто існують статистичні залежності між символами, що характеризуються умовною ймовірністю $P(B/A)$, яка визначає ймовірність здійснення події B за умови, що подія A вже настала.

Наприклад, у осмисленому тексті українською мовою ймовірність появи різних символів (літер) різна [2]. У середньому, у тексті з 1000 літер пробіл між словами зустрічається 138 разів, літера О з'являється 86 разів, раз, Н – 68, А – 64, И – 55. У той же час літера Щ зустрічається 4 рази, Ф – 3, Ґ менше одного разу. Крім того, існують статистичні зв'язки між літерами, скажімо, після голосних літер не може з'явитись Ь.

Оскільки у більшості практичних випадків ступінь невизначеності стану джерела інформації залежить не тільки від кількості станів, а й від ймовірностей цих станів, виникла потреба у пошуку міри інформації, що враховує різноймовірність станів. Інтуїтивно зрозуміло, що часті чи очікувані події несуть мало інформації і, навпаки, рідкісні, тобто несподівані події, мають високий інформаційний зміст. Отже, інформація та ймовірність перебувають у зворотно пропорційній залежності.

Цей підхід до визначення кількості інформації в повідомленнях, що враховує нерівноймовірну появу символів повідомлення та їх статистичний зв'язок, було запропоновано К.Шенноном у 1948 році у роботі «Математична теорія зв'язку». Згідно із запропонованою Шенноном мірою виражена в бітах кількість інформації про подію A, що відбувається з ймовірністю $P(A)$, дорівнює

$$I(A) = \log_2 \frac{1}{P(A)} = -\log_2 P(A). \qquad (2.7)$$

Логарифмічна міра має властивість адитивності. Визначимо кількість інформації, що міститься в складній події, що складається з одночасного здійснення незалежних подій A і B. Відповідно до виразу $P(AB)=P(A)\cdot P(B)$. Тоді

$$I(AB) = \log_2 \frac{1}{P(A)\cdot P(B)} = -\left(\log_2 P(A) + \log_2 P(B)\right), \qquad (2.8)$$

тобто загальна кількість інформації дорівнює сумі інформації, що міститься у кожному з повідомлень.

Приклад 2.10. Джерело передає в канал двійкові символи, причому ймовірність появи нуля становить p_0=0,6, ймовірність появи одиниці p_1=0,4. Визначити кількість інформації в нульовому та одиничному повідомленнях. Визначити кількість інформації у повідомленні 1100.

Рішення. Згідно (2.7)

$$I_0 = \log_2 \frac{1}{p_0} = -\log_2 p_0 = -\log_2 0{,}6 = -(-0{,}7370) = 0{,}7370 \text{ біт};$$

$$I_1 = \log_2 \frac{1}{p_1} = -\log_2 p_1 = -\log_2 0{,}4 = -(-1{,}3219) = 1{,}3219 \text{ біт}.$$

Відповідно до формули (1.30) ймовірність появи повідомлення 1100

$P_{1100} = p_1 \cdot p_1 \cdot p_0 \cdot p_0 = 0{,}4 \cdot 0{,}4 \cdot 0{,}6 \cdot 0{,}6 = 0{,}0576$.

Кількість інформації у повідомленні 1100:

$$I_{1100} = \log_2 \frac{1}{P_{1100}} = -\log_2 P_{1100} = -\log_2 0{,}0576 = -(-4{,}1178) = 4{,}1178 \text{ біт}. \blacksquare$$

Приклад 2.11. Джерело має передати в канал у довільному порядку сім одиниць та три нулі. Визначити кількість інформації в повідомленні, що перші три символи в каналі будуть 110.

Рішення. Оскільки загальна кількість символів, що передаються десять, серед яких сім одиниць, то ймовірність одиничного значення першого символу $P(1) = \frac{7}{10.}$ Після першого символу ситуація змінюється: для передачі залишається шість одиниць і три нулі. Тоді умовна ймовірність появи на другій позиції одиниці $P(1/1) = \frac{6}{9}$. Після другого символу для передачі залишається п'ять одиниць та три нулі. Умовна ймовірність появи на третій позиції нуля $P(0/11) = \frac{3}{8}$.

Згідно (1.29) $P_{110} = P(1) \cdot P(1/1) \cdot P(0/11) = \frac{7}{10} \cdot \frac{6}{9} \cdot \frac{3}{8} = \frac{7}{40} = 0{,}175.$

Кількість інформації у повідомленні 110:

$$I_{110} = \log_2 \frac{1}{P_{110}} = -\log_2 P_{110} = -\log_2 0{,}175 = -(-2{,}5146) = 2{,}5146 \text{ біт.} \blacksquare$$

Приклад 2.12. Визначити кількість інформації, що міститься у повідомленні, що у восьмибітному слові міститься дві помилки. Ймовірність бітової помилки у каналі $p_{пом}$=0,1.

Рішення. Ймовірність того, що в N-бітному двійковому слові міститься k помилок згідно (2.16) визначається як $P_k = C_N^k \cdot p^k \cdot q^{(N-k)}$.

У даному разі $N = 8$, $k = 2$, $p = 0{,}1$ ймовірність безпомилкового прийому символу q=0,9.

Ймовірність спотворення двох розрядів

$$P_2 = C_8^2 \cdot 0{,}1^2 \cdot 0{,}9^6 = 28 \cdot 0{,}01 \cdot 0{,}5314 = 0{,}1488.$$

Згідно з ймовірнісною мірою Шеннона кількість інформації в повідомленні про наявність двох помилок у восьмибітному слові:

$$I_2 = \log_2 \frac{1}{P_2} = -\log_2 P_2 = -\log_2 0{,}1488 = -(-2{,}7486) = 2{,}74866 \text{ біт.} \blacksquare$$

Для дискретного джерела інформації однією з ключових характеристик є середня кількість інформації, що передається в одному символі повідомлення. Нехай джерело може генерувати один із n символів $x_1,\ldots, x_n$ з ймовірностями появи символів відповідно p_2, ... p_n. Кількість інформації про подію, що відповідає появі кожного з символів, позначимо як I_1, I_2, ..I_n.

Нехай елементи $\{x_1,\ldots, x_n\}$ утворюють ансамбль – повну систему випадкових подій з ймовірностями їх появи, що становлять у сумі одиницю

$$\sum_{i=1}^{n} p_i = 1. \qquad (2.9)$$

Середня кількість інформації на один символ інформаційного повідомлення від дискретного джерела складе

$$I_{сер} = \sum_{i=1}^{n} p_i \cdot I_i. \qquad (2.10)$$

Підставляючи (2.7) в (2.10), отримуємо

$$I_{сер} = \sum_{i=1}^{n} p_i \cdot \log_2\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i. \qquad (2.11)$$

Отримане співвідношення К. Шеннон назвав *ентропією джерела* та позначив літерою *H*.

Ентропія найпростішого джерела без пам'яті *X* з алфавітом $\{x_1,\ldots, x_n\}$ та відповідними ймовірностями $p(x_1), p(x_2), \ldots, p(x_n)$ дорівнює

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i). \qquad (2.12)$$

Ентропія – це величина, яка виражає середню на одну подію (повідомлення) кількість інформації. Фізично ентропія *H* висловлює середню невизначеність стану джерела повідомлень та є об'єктивною інформаційною характеристикою джерела.

Для окремих незалежних елементів множини як міра може використовуватися часткова невизначеність

$$H(x_i) = -\log_2 p(x_i). \qquad (2.13)$$

Ентропія, як кількісна міра інформативності джерела має такі властивості.

1. Ентропія є величина дійсна, обмежена та невід'ємна.

2. Ентропія детермінованих повідомлень (заздалегідь відомих) дорівнює нулю, тобто $H = 0$ якщо хоча б одне з повідомлень має ймовірність рівну одиниці.

3. Ентропія максимальна, якщо повідомлення рівноймовірні

$$p(x_1) = p(x_2) = \ldots = p(x_m) = \frac{1}{n}, \qquad (2.14)$$

$$H(x) = -\sum_{i=1}^{n} \frac{1}{n}\log_2 \frac{1}{n} = \log_2 n. \qquad (2.15)$$

У цьому випадку статистична міра інформації Шеннона (2.11) збігається зі структурною мірою Гартлі (2.3)

Найбільш широке застосування в дискретних системах передачі отримали дискретні двійкові джерела, які характеризуються передачею лише двох можливих повідомлень, нуля і одиниці. Причому якщо ймовірність одного з них, наприклад одиниці дорівнює $p(1)$, то ймовірність передачі іншого, тобто нуля $p(0) = 1 - p(1)$. Визначимо ентропію двійкового джерела

$$H(x) = -\sum_{i=1}^{2} p(x_i)\log_2 p(x_i) = -p(0)\log_2 p(0) - p(1)\log_2 p(1) =$$

$$= -p(0)\log_2 p(0) - (1 - p(0))\log_2 (1 - p(0)).$$

Графік залежності ентропії двійкового джерела $H(x)$ від ймовірності символу $p(x)$ представлений на рис. 2.2.

Як випливає з рис. 2.2, ентропія двійкового джерела змінюється в границях від 0 до 1. Ентропія дорівнює 0, коли ймовірність передачі одного з символів дорівнює 0 чи 1, тобто передається лише одне повідомлення. Ентропія двійкового джерела буде максимальна, якщо існує найбільша невизначеність $p(x_1) = p(x_2) = 0{,}5$, тоді $H_{\max}(x) = \log_2 0{,}5 = 1$.

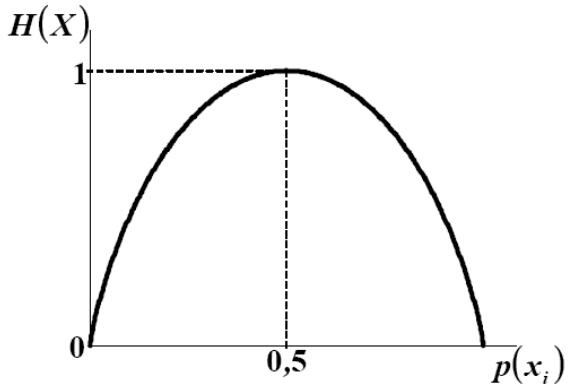


Рисунок 2.2 – Залежність ентропії від ймовірності символів

Приклад 2.13. Джерело повідомлень видає символи з алфавіту $\{x_1, \ldots, x_6\}$ зі ймовірностями відповідно до табл. 2.2. Визначити кількість інформації в повідомленні за Гартлі та Шенноном.

Таблиця 2.2 – Ймовірності появи символів

x_i	1	2	3	4	5	6
$P(x_i)$	0,31	0,14	0,22	0,07	0,16	0,10

Рішення. Кількість інформації відповідно до структурної адитивної міри Гартлі обчислимо за формулою (2.3).

Кількість інформації відповідно до статистичної міри Шеннона розрахуємо за формулою (2.11).

Для спрощення обчислень при розв'язанні задач в додатку А наведено таблицю значень величин $-p\log_2 p$ та таблицю двійкових логарифмів деяких цілих чисел. При використанні калькулятора, що не має функції двійкового логарифму, доцільно використовувати співвідношення

$$\log_2 N = \frac{\lg N}{\lg 2} = \frac{\lg N}{0{,}301} = 3{,}3219 \lg N.$$

Таким чином:

$I_\Gamma = \log_2 N = \log_2 6 = 2{,}5850$ бит.

$$I_{Ш} = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i = -(0{,}31 \cdot \log_2 0{,}31 + 0{,}14 \cdot \log_2 0{,}14 + 0{,}22 \cdot \log_2 0{,}22 +$$

$$+ 0{,}07 \cdot \log_2 0{,}07 + 0{,}16 \cdot \log_2 0{,}16 + 0{,}10 \cdot \log_2 0{,}10) = -(-0{,}5238 - 0{,}3971 -$$

$$- 0{,}4806 - 0{,}2686 - 0{,}4230 + 0{,}3322) = 2{,}4253 \text{ біт}.$$

Як і слід очікувати, кількість інформації за Гартлі більше, ніж за Шенноном. ■

Приклад 2.14 До каналу зв'язку передаються числа, закодовані двобітними посилками. Перший біт у кожній посилці приймає одиничне значення з ймовірністю 0,3, нульове з ймовірністю 0,7. Другий біт збігається з першим із ймовірністю 0,8, не збігається – зі ймовірністю 0,2. Визначити середню кількість інформації в повідомленні за Гартлі та Шенноном.

Рішення. Нехай A – подія, що складається в передачі першим одиничного символу, $\overline{A}$ – нульового символу, B – подія, що складається в передачі другим символу, що збігається з першим, $\overline{B}$ – у передачі другим символу, відмінного від першого. Розрахуємо ймовірність всіх двобітових посилок

$$P_{00} = P(\overline{A}) \cdot P(B) = 0{,}7 \cdot 0{,}8 = 0{,}56;$$

$$P_{01} = P(\overline{A}) \cdot P(\overline{B}) = 0{,}7 \cdot 0{,}2 = 0{,}14;$$

$$P_{10} = P(A) \cdot P(\overline{B}) = 0{,}3 \cdot 0{,}2 = 0{,}06;$$

$$P_{10} = P(A) \cdot P(B) = 0{,}3 \cdot 0{,}8 = 0{,}24.$$

Розподіл ймовірностей відображено в табл. 2.3.

Таблиця 2.3 – Розподіл ймовірностей двобітових посилок

Код	00	01	10	11
$P(x_i)$	0,56	0,14	0,06	0,24

Загальна кількість двобітових посилок N=4.

Згідно з формулами (2.3), (2.11)

$I_Г = \log_2 N = \log_2 4 = 2$ бита.

$$I_{Ш} = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i = -(0{,}56 \cdot \log_2 0{,}56 + 0{,}14 \cdot \log_2 0{,}14 + 0{,}06 \cdot \log_2 0{,}06 +$$

$$+ 0{,}24 \cdot \log_2 0{,}24) = -(-0{,}4684 - 0{,}3971 - 0{,}2435 - 0{,}4941) = 1.6031 \text{біт}$$

$I_X > I_Ш$. ■

Приклад 2.15 Джерело передає в канал незалежні один від одного посилки 11 та 00 з ймовірностями відповідно 0,6 та 0,4. При передачі каналом зв'язку кожен розряд незалежно від інших може змінювати своє значення на протилежне з ймовірністю 0,1. Визначити ентропію повідомлення, що приймається.

Рішення. Нехай A – подія, що полягає в передачі в канал комбінації 11, B – подія, що складається в передачі в канал комбінації 00, C – подія, що полягає в спотворенні одного окремого біта, відповідно $\overline{C}$ – протилежна подія, що полягає в тому, що біт залишиться неспотвореним. Ймовірність подій згідно з умовою становить:

$P(A) = 0{,}6$; $P(B) = 0{,}4$; $P(C) = 0{,}1$; $P(\overline{C}) = 1 - P(C) = 0{,}9$.

Розглянемо ймовірнсть прийому кожної з комбінацій. Комбінація 00 може бути прийнята у двох випадках: якщо була передана комбінація 11 і обидва біти спотворилися і, якщо була передана комбінація 00 і обидва біти залишилися незмінними

$$P_{00} = P(A) \cdot P(C) \cdot P(C) + P(B) \cdot P(\overline{C}) \cdot P(\overline{C}) = 0{,}6 \cdot 0{,}1 \cdot 0{,}1 + 0{,}4 \cdot 0{,}9 \cdot 0{,}9 =$$

$$= 0{,}006 + 0{,}324 = 0{,}33.$$

Аналогічно

$$P_{01} = P(A) \cdot P(C) \cdot P(\overline{C}) + P(B) \cdot P(\overline{C}) \cdot P(C) = 0{,}6 \cdot 0{,}1 \cdot 0{,}9 + 0{,}4 \cdot 0{,}9 \cdot 0{,}1 =$$

$$= 0{,}054 + 0{,}036 = 0{,}09.$$

$$P_{10} = P(A)\cdot P(\overline{C})\cdot P(C) + P(B)\cdot P(C)\cdot P(\overline{C}) = 0{,}6\cdot 0{,}9\cdot 0{,}1 + 0{,}4\cdot 0{,}1\cdot 0{,}9 =$$
$$= 0{,}054 + 0{,}036 = 0{,}09.$$

$$P_{11} = P(A)\cdot P(\overline{C})\cdot P(\overline{C}) + P(B)\cdot P(C)\cdot P(C) = 0{,}6\cdot 0{,}9\cdot 0{,}9 + 0{,}4\cdot 0{,}1\cdot 0{,}1 =$$
$$= 0{,}486 + 0{,}004 = 0{,}49.$$

Згідно (2.12) значення ентропії

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) = -(0{,}33\cdot\log_2 0{,}33 + 0{,}09\cdot\log_2 0{,}09 +$$
$$+ 0{,}09\cdot\log_2 0{,}09 + 0{,}49\cdot\log_2 0{,}49) = -(0{,}5278 + 0{,}3127 + 0{,}3127 + 0{,}5043) = 1{,}6575 \text{ біт.}$$

■

Приклад 2.16. Ймовірнсть бітової помилки у каналі p=0.2. Визначити кількість інформації, у повідомленні про кількість помилок у п'ятибітовому слові.

Рішення. Кількість помилок у п'ятибітовому слові може змінюватись у межах від нуля до п'яти, тобто можливо шість різних результатів досвіду.

Відповідно до адитивної міри Гартлі у даному разі $I=\log_2 6= 2{,}5850$ біт.

Визначимо кількість інформації з урахуванням нерівної ймовірності повідомлень (мера Шеннона). Розрахуємо ймовірність кожного з можливих чисел помилок у повідомленні.

Згідно з формулою (1.31)

$$P_0 = C_5^0\cdot 0{,}2^0\cdot 0{,}8^5 = 1\cdot 1\cdot 0{,}32768 = 0{,}32768;$$

$$P_1 = C_5^1\cdot 0{,}2^1\cdot 0{,}8^4 = 5\cdot 0{,}2\cdot 0{,}4096 = 0{,}4096;$$

$$P_2 = C_5^2\cdot 0{,}2^2\cdot 0{,}8^3 = 10\cdot 0{,}04\cdot 0{,}512 = 0{,}2048;$$

$$P_3 = C_5^3\cdot 0{,}2^3\cdot 0{,}8^2 = 10\cdot 0{,}008\cdot 0{,}64 = 0{,}0512;$$

$$P_4 = C_5^4\cdot 0{,}2^4\cdot 0{,}8^1 = 5\cdot 0{,}0016\cdot 0{,}8 = 0{,}0064;$$

$$P_5 = C_5^5\cdot 0{,}2^5\cdot 0{,}8^0 = 1\cdot 0{,}00032\cdot 1 = 0{,}00032.$$

З метою перевірки підсумуємо ймовірність подій, що утворюють повну групу.

P_0+P_1+P_2+ P_3+ P_4+P_5=0,32768+0,4096+0,2048+0,0512+0,0064+
+0,00032=1.

Визначимо середню кількість інформації (ентропію), що припадає на елементарне повідомлення. У даному випадку елементарним повідомленням є інформація про кількість помилок у слові (0..5).

Згідно з формулами (2.3), (2.11)

$I_X = \log_2 N = \log_2 6 = 2{,}5850$ бит.

$$I_{Ш} = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i = -(0{,}32768 \cdot \log_2 0{,}32768 + 0{,}4096 \cdot \log_2 0{,}4096 +$$

$$0{,}2048 \cdot \log_2 0{,}2048 + 0{,}0512 \cdot \log_2 0{,}0512 + 0{,}0064 \cdot \log_2 0{,}0064 +$$

$$+ 0{,}00032 \cdot \log_2 0{,}00032) = -(-0{,}5275 - 0{,}5275 - 0{,}4685 - 0{,}2195 - 0{,}0466 -$$

$$- 0{,}0037) = 1.7933 \text{ бит}.$$

Як і в попередніх випадках, $I_X > I_Ш$.

2.4 Визначення ентропії об'єднаних повідомлень

Деякі задачі теорії інформації вимагають одночасного розгляду двох або більшої кількості джерел повідомлень. Наприклад, при вимірюванні вологості психрометричним методом можуть одночасно зніматися і передаватися в канал значення температур, що вимірюються сухим і мокрим термометрами. В більшості випадків для всіх джерел задаються алфавіти та набори ймовірностей появи символів.

Ансамблем називається повна сукупність станів із ймовірностями їх появ, що становлять у сумі одиницю

x	x_1	x_2	x_3	x_4	…	x_m
$p(x)$	$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	…	$p(x_m)$

причому $\sum_{i=1}^{m} p(x_i) = 1$.

Об'єднанням ансамблів називається сукупність двох і більше взаємозалежних або незалежних ансамблів джерел повідомлень.

Розглянемо об'єднання, що складається з двох ансамблів X та Y. Повідомлення першого джерела X приймають значення x_1, x_2, x_3, … x_n з ймовірностями $p(x_1)$, $p(x_2)$, … $p(x_n)$, повідомлення другого джерела Y приймають значення y_1, y_2, y_3, … y_m з ймовірностями $p(y_1)$, $p(y_2)$, … $p(y_m)$. Об'єднання ансамблів характеризується матрицею $P(X, Y)$ ймовірностей $p(x_i, y_j)$ всіх можливих комбінацій спільної появи елементів x_i та y_j.

$$P(X,Y)=\begin{vmatrix} p(x_1,y_1)\ldots p(x_i,y_1)\ldots p(x_n,y_1) \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ p(x_1,y_j)\ldots p(x_i,y_j)\ldots p(x_n,y_j) \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ p(x_1,y_m)\ldots p(x_i,y_m)\ldots p(x_n,y_m) \end{vmatrix} \qquad (2.16)$$

Підсумовуючи стовпці та рядки матриці (2.16) отримаємо інформацію про ансамблі *X* та *Y* вихідних джерел

x	x_1	x_2	x_3	x_4	…	x_n
$p(x)$	$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	…	$p(x_n)$

y	y_1	y_2	y_3	y_4	…	y_m
$p(y)$	$p(y_1)$	$p(y_2)$	$p(y_3)$	$p(y_4)$	…	$p(y_m)$

$$p(x_i)=\sum_{j=1}^{m} p(x_i,y_j) \quad \text{та} \quad p(y_j)=\sum_{i=1}^{n} p(x_i,y_j)\,. \qquad (2.17)$$

Якщо обидва джерела якимось чином пов'язані між собою, слід очікувати, що повідомлення, що надходять з одного джерела дозволяють робити певне припущення про стан іншого. Таким чином невизначеність другого джерела знижується, тобто джерела обмінюються взаємною інформацією.

Спільна ентропія двох джерел визначається як математичне сподівання кількості інформації всіх пар подій.

$$H(X,Y)=-\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i,y_j)\log_2 p(x_i,y_j)\,. \qquad (2.18)$$

Для знаходження спільної ентропії *H* (*X, Y*) об'єднаного ансамблю джерел повідомлень розглянемо два види об'єднання ансамблів:

– *статистично незалежних* ансамблів *X* та *Y*;

– *статистично залежних* ансамблів *X* та *Y*.

У разі *статистичної незалежності* об'єднаних ансамблів X та Y спільна ймовірність появи елементів x_i та y_j дорівнює

$$p(x_i, y_j) = p(x_i) \cdot p(y_j). \tag{2.19}$$

Оскільки при цьому $\log_2 p(x_i,y_j) = \log_2 p(x_i) + \log_2 p(y_j)$, то спільна ентропія об'єднаного ансамблю

$$\begin{aligned} H(X,Y) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log_2 p(x_i, y_j) = \\ &= -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i)p(y_j)\log_2 (p(x_i)p(y_j)) = \\ &= -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)\cdot\sum_{j=1}^{m} p(y_j) - \sum_{j=1}^{m} p(y_j)\log_2 p(y_j)\cdot\sum_{i-1}^{n} p(x_i). \end{aligned} \tag{2.20}$$

Враховуючи, що, $\sum_{j=1}^{m} p(y_j) = 1;$ $\sum_{i=1}^{n} p(x_i) = 1,$ отримаємо

$$H(X,Y) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) - \sum_{j=1}^{m} p(y_j)\log_2 p(y_j) = H(X) + H(Y). \tag{2.21}$$

Таким чином, спільна ентропія незалежних джерел дорівнює сумі *безумовних ентропій* ансамблів джерел повідомлень

$$H(X,Y) = H(X) + H(Y). \tag{2.22}$$

У разі *статистичної залежності* об'єднаних ансамблів X та Y спільна ймовірність появи елементів x_i та y_j дорівнює

$$p(x_i, y_j) = p(x_i) \cdot p(y_j / x_i) = p(y_j)\, p(x_i / y_j), \tag{2.23}$$

де $p(x_i / y_j)$ – *умовна ймовірність* реалізації стану x_i за умови, що реалізувався стан y_j; $p(y_j / x_i)$ – *умовна ймовірність* реалізації стану y_j за умови, що реалізувався стан x_i.

Тоді вираз для ентропії об'єднання відповідно до (2.13) набуває вигляду

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log_2 p(x_i, y_j) =$$

$$-\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i)p(x_i / y_j)\log_2 p(x_i)p(x_i / y_j) = \qquad . \qquad (2.24)$$

$$= -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)\sum_{j=1}^{m} p(y_j / x_i) - \sum_{i=1}^{n} p(x_i)\sum_{j=1}^{m} p(y_j / x_i)\log_2 p(y_j / x_i)$$

З урахуванням того, що $\sum_{j=1}^{m} p(y_j / x_i) = 1$, перший доданок – це ентропія $H(X)$ ансамблю X. Таким чином, отримаємо

$$H(X,Y) = H(X) - \sum_{i=1}^{n} p(x_i)\sum_{j=1}^{m} p(y_j / x_i)\log_2 p(y_j / x_i), \qquad (2.25)$$

де $-\sum_{j=1}^{m} p(y_j / x_i)\log_2 p(y_j / x_i)$ – *часткова умовна ентропія*, яка характеризує невизначеність, що припадає на один стан ансамблю Y за умови, що реалізувався конкретний стан x_i ансамблю X. Позначається *часткова умовна ентропія* як $H(Y/x_i)$

$$H(Y / x_i) = -\sum_{j=1}^{m} p(y_j / x_i)\log_2 p(y_j / x_i). \qquad (2.26)$$

При усередненні по всіх станах ансамблю X отримаємо середню невизначеність, що припадає на стан ансамблю Y при відомих станах ансамблю X

$$H(Y / X) = -\sum_{i=1}^{n} p(x_i)H(Y / x_i) = -\sum_{i=1}^{n} p(x_i)\sum_{j=1}^{m} p(y_j / x_i)\log_2 p(y_j / x_i). \qquad (2.27)$$

де $H(Y/X)$ – *повна умовна ентропія* ансамблю Y стосовно ансамблю X, яка показує яку ентропію дають повідомлення джерела Y, коли відома ентропія повідомлення джерела X.

Таким чином, з урахуванням формули (3.8) *повна умовна ентропія* двох статистично залежних ансамблів

$$H(Y/X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_j, y_j)\log_2 p(y_j/x_i), \tag{2.28}$$

$$H(X/Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_j, y_j)\log_2 p(x_i/y_j). \tag{2.29}$$

Підставляючи (2.27) в (2.24) отримуємо

$$H(X,Y) = H(X) + H(Y/X) = H(Y) + H(X/Y), \tag{2.30}$$

де

$$H(X/Y) = \sum_{j=1}^{m} p(y_j)H(X/y_j) = -\sum_{j=1}^{m} p(y_j)\sum_{i=1}^{n} p(x_i/y_j)\log_2 p(x_i/y_j),$$

$$H(X/y_j) = -\sum_{i=1}^{n} p(x_i/y_j)\log_2 p(x_i/y_j).$$

Таким чином, ентропія об'єднання двох статистично пов'язаних ансамблів *X* та *Y* дорівнює безумовної ентропії одного ансамблю *H*(*X*) або *H*(*Y*) плюс умовна ентропія іншого ансамблю відносно першого *H*(*X*/*Y*) або *H*(*Y*/*X*).

Основні властивості спільної та умовної ентропії.

1. При статистично незалежних повідомленнях ансамблів *X* та *Y* спільна ентропія дорівнює сумі ентропій ансамблів

$$H(X,Y) = H(X) + H(Y). \tag{2.31}$$

2. При повній статистичній залежності ансамблів $\{X, p(x_i)\}$ і $\{Y, p(y_j)\}$ (при цьому обов'язково *n*=*m*) спільна ентропія дорівнює безумовної ентропії одного з ансамблів.

3. Для спільної ентропії завжди справедливе співвідношення

$$H(X,Y) \le H(X) + H(Y). \tag{2.32}$$

4. Умовна ентропія може змінюватись у межах

$$0 \le H(Y/X) \le H(Y). \tag{2.33}$$

Приклад 2.16. Ймовірності спільного здійснення подій $P(x_i, y_j)$, що поєднує два статистично залежні джерела ансамблів задані в таблиці 2.4.

Таблиця 2.4 – Ймовірності спільного здійснення подій X_i та Y_j

	X_1	X_2
Y_1	0.20	0.35
Y_2	0.40	0.05

Визначити:

1. Умовну ентропію ансамблів $H_x(Y) \rightarrow H(Y/X)$.

2. Ентропію об'єднаного ансамблю X та $Y \rightarrow H(X,Y)$ для незалежних подій.

Рішення

1. Визначаємо *закони розподілу* дискретних випадкових величин X та Y. Число рядків та стовбців матриці n=2.

$$p(x_i) = \sum_{j=1}^{n} P_{ij} \qquad p(y_j) = \sum_{i=1}^{n} P_{ij}$$

$p(x_1)= P_{11} + P_{12}=p(x_1,y_1) + p(x_1,y_2) = 0.20+0.40=0.6;$
$p(x_2)= P_{21} + P_{22}= p(x_2,y_1) + p(x_2,y_2) = 0.35+0.05=0.4;$

$p(y_1)= P_{11} + P_{21}= p(x_1,y_1) + p(x_2,y_1) = 0.20 + 0.35=0.55;$
$p(y_2)= P_{12} + P_{22}= p(x_1,y_2) + p(x_2,y_2) = 0.40 + 0.05=0.45.$

Знаходимо *умовний закон розподілу X*. Для цього знаходимо умовні ймовірності можливих значень x_i за умови, що величина Y набуває значення y_j:

$$p(x_i/y_j)=p(x_i,y_j)/\ p(y_j).$$

$p(x_1/y_1)=p(x_1,y_1)/\ p(y_1) = 0.20/0.55 = 0,33;$
$p(x_2/y_1)=p(x_2,y_1)/\ p(y_1) = 0.35/0.55 = 0,87;$
$p(x_1/y_2)=p(x_1,y_2)/\ p(y_2) = 0.40/0.45 = 0.67;$
$p(x_2/y_2)=p(x_2,y_2)/\ p(y_2) = 0.05/0.45 = 0.13.$

Знаходимо *умовний закон розподілу Y*. Для цього знаходимо умовні ймовірності можливих значень y_j за умови, що величина X набуває значення x_i:

$$p(y_j/x_i)=p(x_i,y_j)/\ p(x_i).$$

$p(y_1/x_1)=p(x_1,y_1)/\ p(x_1) = 0.20/0.6 = 0.33;$

$p(y_2/x_1)=p(x_1,y_2)/\ p(x_1) = 0.40/0.6 = 0.66;$

$p(y_1/x_2)=p(x_2,y_1)/\ p(x_2) = 0.35/0.4 = 0.875;$

$p(y_2/x_2)=p(x_2,y_2)/\ p(x_2) = 0.05/0.4 = 0.125.$

Знаходимо *часткові умовні ентропії* ансамблю Y, які характеризують невизначеність, що припадає на один стан ансамблю Y за умови, що реалізувався конкретний стан y_j ансамблю X:

$$H(Y / x_i) = -\sum_{j=1}^{n} p(y_j / x_i) \log_2\ p(y_j / x_i)$$

$H(Y/x_1) = -(p(y_1/x_1)\log_2 p(y_1/x_1) + p(y_2/x_1)\log_2 p(y_2/x_1))=$

$=-(0.33 \log_2 0.33 + 0.66 \log_2 0.66)= 0.53 + 0.39 = 0.92.$

$H(Y/x_2) = -(p(y_1/x_2)\log_2 p(y_1/x_2) + p(y_2/x_2)\log_2 p(y_2/x_2))=$

$=-(0.875 \log_2 0.875+ 0.125 \log_2 0.125)= 0.17 + 0.38 = 0.$

Визначаємо *повну умовну ентропію* ансамблю Y стосовно ансамблю X:

$$H(Y / X) = \sum_{i=1}^{n} p(x_i) H(Y / x_i)\ .$$

$H(Y/X) = p(x_1)\ H(Y/\ x_1) + p(x_2)\ H(Y/\ x_2) = 0.6 \bullet 0.92 + 0.4 \bullet 0.55\ = 0.77.$

2. Визначимо ентропію об’єднання двох незалежних подій

$$H(X,Y) = H(X) + H(Y) = -\sum_{i=1}^{2} p(x_i) \log p(x_i) - \sum_{j=1}^{2} p(y_j) \log p(y_j) =$$

$$-(0.6\log 0.6 + 0.4\log 0.4) - (0.55\log 0.55 + 0.45\log 0.45) =$$

$$= 0.44 + 0.53 + 0.47 + 0.52 = 1.96$$

■

Приклад 2.17. Закон розподілу ймовірностей системи, що поєднує залежні джерела інформації, X та Y заданий за допомогою таблиці 2.5.

Таблиця 2.5 – Матриця ймовірностей P_{ij} спільного здійснення подій X_i та Y_j

	Y_1	Y_2
X_1	P_{11}=0.15	P_{12}=0.05
X_2	P_{21} =0.3	P_{22}=0.12
X_3	P_{31}=0.35	P_{32}=0.03

Визначити:

1. Ентропію ансамблів X та $Y \rightarrow H(X)$, $H(Y)$.

2. Умовну ентропію ансамблів $H_x(Y) \rightarrow H(Y/X)$, $H_y(X) \rightarrow H(X/Y)$.

3. Ентропію об'єднаного ансамблю X та $Y \rightarrow H(X,Y)$.

Рішення

1. Визначаємо *закони розподілу* дискретних випадкових величин X та Y. Число рядків матриці n=3. Число стовпців матриці m=2

$$p(x_i) = \sum_{j=1}^{m} P_{ij} \qquad p(y_j) = \sum_{i=1}^{n} P_{ij}$$

$p(x_1)= P_{11} + P_{12}=p(x_1,y_1) + p(x_1,y_2) = 0.15+0.05=0.2;$

$p(x_2)= P_{21} + P_{22}= p(x_2,y_1) + p(x_2,y_2) = 0.3+0.12=0.42;$

$p(x_3)= P_{31} + P_{32}= p(x_3,y_1) + p(x_3,y_2) = 0.35+0.03=0.38;$

$p(y_1)=P_{11}+P_{21}+ P_{31}= p(x_1,y_1) + p(x_2,y_1) + p(x_3,y_1) = 0.15 + 0.3 + 0.35=0.8;$

$p(y_2)=P_{12}+P_{22}+P_{32}=p(x_1,y_2) + p(x_2,y_2) + p(x_3,y_2) = 0.05 + 0.12 + 0.03=0.2.$

2. Знаходимо *часткові безумовні ентропії* ансамблів X та Y.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \qquad H(Y) = -\sum_{i=1}^{n} p(y_j)\log_2 p(y_j)$$

$H(X) =-(p(x_1)\log_2 p(x_1) + p(x_2)\log_2 p(x_2) + p(x_3)\log_2 p(x_3))=$

$=-(0.2 \log_2 0.2 + 0.42 \log_2 0.42 + 0.38 \log_2 0.38)=$

$0.4644+0.5256+0.5305 = 1.5205.$

$H(Y) =-(p(y_1)\log_2 p(y_1) + p(y_2)\log_2 p(y_2) =-(0.8 \log_2 0.8 + 0.2 \log_2 0.2)=$

$0.2575 + 0.4644 = 0.7219.$

3. Знаходимо *умовний закон розподілу X*. Для цього знаходимо умовні ймовірності можливих значень x_i за умови, що величина Y набуває значення y_j:

$p(x_i/y_j)=p(x_i,y_j)/ p(y_j).$

$p(x_1/y_1)=p(x_1,y_1)/ p(y_1) = 0.15/0.8 = 0.1875;$

$p(x_2/y_1)=p(x_2,y_1)/ p(y_1) = 0.3/0.8 = 0.375;$

$p(x_3/y_1)=p(x_3,y_1)/ p(y_1) = 0.35/0.8 = 0.4375;$

$p(x_1/y_2)=p(x_1,y_2)/ p(y_2) = 0.05/0.2 = 0.25;$

$p(x_2/y_2)=p(x_2,y_2)/ p(y_2) = 0.12/0.2 = 0.6;$

$p(x_3/y_2)=p(x_3,y_2)/ p(y_2) = 0.03/0.2 = 0.15.$

4. Знаходимо *умовний закон розподілу Y*. Для цього знаходимо умовні ймовірності можливих значень y_j за умови, що величина X набуває значення x_i:

$p(y_j/x_i)=p(x_i,y_j)/\ p(x_i)$.

$p(y_1 / x_1)=p(x_1,y_1)/\ p(x_1) = 0.15/0.2 = 0.75$;

$p(y_2/x_1)=p(x_1,y_2)/\ p(x_1) = 0.05/0.2 = 0.25$;

$p(y_1/x_2)=p(x_2,y_1)/\ p(x_2) = 0.3/0.42 = 0.7143$;

$p(y_2/x_2)=p(x_2,y_2)/\ p(x_2) = 0.12/0.42 = 0.2857$;

$p(y_1/x_3)=p(x_3,y_1)/\ p(x_3) = 0.35/0.38 = 0.9211$;

$p(y_2/x_3)=p(x_3,y_2)/\ p(x_3) = 0.03/0.38 = 0.0789$;

5. Знаходимо *часткові умовні ентропії* ансамблю X, які характеризують невизначеність, що припадає на один стан ансамблю X за умови, що реалізувався конкретний стан y_j ансамблю Y:

$$H(X / y_j) = -\sum_{i=1}^{n} p(x_i / y_j) \log_2\ p(x_i / y_j)$$

$H(X/y_1) = -\ (p(x_1/y_1)\log_2 p(x_1/y_1) + p(x_2/y_1)\log_2 p(x_2/y_1) + p(x_3/y_1)\log_2 p(x_3/y_1))=$

$=-\ (0.1875 \log_2 0.1875 + 0.375 \log_2 0.375 + 0.4375 \log_2 0.4375) =$

$= 0.4528+0.5307+0.5218 = 1.5053.$

$H(X/y_2) = -\ (p(x_1/y_2)\log_2 p(x_1/y_2) + p(x_2/y_2)\log_2 p(x_2/y_2) + p(x_3/y_2)\log_2 p(x_3/y_2))=$

$=-(0.25 \log_2 0.25 + 0.6 \log_2 0.5 + 0.15 \log_2 0.15)=$

$= 0.5+0.4422+0.4105 = 1.3527.$

6. Знаходимо *часткові умовні ентропії* ансамблю Y, які характеризують невизначеність, що припадає на один стан ансамблю Y за умови, що реалізувався конкретний стан y_j ансамблю X:

$$H(Y / x_i) = -\sum_{j=1}^{m} p(y_j / x_i) \log_2\ p(y_j / x_i)$$

$H(Y/x_1) = -(p(y_1/x_1)\log_2 p(y_1/x_1) + p(y_2/x_1)\log_2 p(y_2/x_1))=$

$=-(0.75 \log_2 0.75 + 0.25 \log_2 0.25)= 0.3113 + 0.5 = 0.8113.$

$H(Y/x_2) = -(p(y_1/x_2)\log_2 p(y_1/x_2) + p(y_2/x_2)\log_2 p(y_2/x_2))=$

$=-(0.7143 \log_2 0.7143+ 0.2857 \log_2 0.\ 2857)= 0.3467 + 0.5164 = 0.8631.$

$H(Y/x_3) = -(p(y_1/x_3)\log_2 p(y_1/x_3) + p(y_2/x_3)\log_2 p(y_2/x_3))=$

$=-(0.9211 \log_2 0.9211 + 0.\ 0789 \log_2 0.\ 0.0789)=$

$= 0.1093 + 0.2892 = 0.3985.$

7. Визначаємо *повну умовну ентропію* ансамблю X стосовно ансамблю Y:

$$H(X/Y) = \sum_{j=1}^{m} p(y_j) H(X/y_j)$$

$H(X/Y) = p(y_1)\, H(X/y_1) + p(y_2)\, H(X/y_2) = 0.8 \bullet 1.5053 + 0.2 \bullet 1.3527 =$
$= 1.4748.$

8. Визначаємо *повну умовну ентропію* ансамблю Y стосовно ансамблю X:

$$H(Y/X) = \sum_{i=1}^{n} p(x_i) H(Y/x_i)$$

$H(Y/X) = p(x_1)\, H(Y/x_1) + p(x_2)\, H(Y/x_2) + p(x_3)\, H(Y/x_3) =$
$= 0.2 \bullet 0.8113 + 0.42 \bullet 0.8631 + 0.38 \bullet 0.3985 = 0.6762.$

9. Визначаємо трьома способами ентропію об'єднання двох статистично пов'язаних ансамблів

$$H_1(X,Y) = H(X) + H(Y/X)$$

$H_1(X,Y) = 1.5205 + 0.6762 = 2.1967$

$$H_2(X,Y) = H(Y) + H(X/Y)$$

$H_2(X,Y) = 0.7219 + 1.4747 = 2.1967$

$$H_3(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j) \log_2 p(x_i, y_j)$$

$H_3(X,Y) = -(p(x_1,y_1) \log_2 p(x_1,y_1) + p(x_1,y_2) \log_2 p(x_1,y_2) +$
$+p(x_2,y_1) \log_2 p(x_2,y_1) + p(x_2,y_2) \log_2 p(x_2,y_2) + p(x_3,y_1) \log_2 p(x_3,y_1) +$
$+p(x_3,y_2) \log_2 p(x_3,y_2)) = -(0.15 \log_2 0.15 + 0.05 \log_2 0.05 + 0.3 \log_2 0.3 +$
$+ 0.12 \log_2 0.12 + 0.35 \log_2 0.35 + 0.03 \log_2 0.03) =$
$= 0.4105 + 0.2161 + 0.5211 + 0.3671 + 0.5301 + 0.1518 = 2.1967.$

Можна бачити, що обчислені трьома способами ентропії об'єднання двох статистично пов'язаних ансамблів $H_1(X,Y)$, $H_2(X,Y)$, $H_3(X,Y)$ співпадають. ■

2.5 Розрахунок кількості інформації неперервних повідомлень. Дискретизація та квантування сигналів

До цього часу нами розглядалися дискретні за рівнем випадкові повідомлення, які мають кінцеву кількість можливих станів. Узагальнимо

тепер розглянутий матеріал на випадок, коли множина станів джерела неперервна. Джерела інформації, у яких множина можливих станів становить континуум, називають неперервними (аналоговими). Оскільки більшість сучасних пристроїв обробки інформації є цифровими, тобто оперують із послідовностями двійкових чисел, результати вимірювань безперервної величини зазвичай піддаються *квантуванню за рівнем*.

Операція квантування за рівнем полягає у тому, що безперервна множина значень, які може набувати безперервна величина X, замінюється кінцевою множиною *рівнів квантування* $X_{кв}^{k}$, k=1…m. Операція квантування може здійснюватися із заокругленням до найближчого більшого значення, найближчого меншого та до середини інтервалу квантування (у більшості випадків).

Різниця між двома суміжними рівнями квантування $\Delta X_{кв}^{k+1} = X_{кв}^{k+1} - X_{кв}^{k}$ називається *кроком квантування*. Розрізняють *рівномірне* та *нерівномірне квантування*. При рівномірному квантуванні крок береться незмінним. При нерівномірному квантуванні крок $\Delta X_{кв}^{k}$ є змінним та залежить від номера рівня k.

Найчастіше передбачається, що величина X розподілена рівномірно в деякому інтервалі від X_{min} до X_{max}. При цьому зазвичай задається необхідна точність вимірювання величини, яка визначає максимально можливий крок квантування ΔX.

Тоді число рівнів, на які ділиться квантована величина X, становитиме

$$N = \frac{X_{\max} - X_{\min}}{\Delta X}. \tag{2.34}$$

Кількість інформації, що міститься в повідомленні про результат вимірювання величини дорівнює

$$I = \log_2 N = \log_2 \frac{X_{\max} - X_{\min}}{\Delta X}. \tag{2.35}$$

Приклад 2.18. Визначити кількість інформації, що міститься в результаті виміру випадкової величини, рівномірно розподіленої в діапазоні 0 ... 250 В. Необхідна точність виміру 0,1 В.

Рішення. Число рівнів, на які ділиться квантована величина X, складе

$$N = \frac{X_{\max} - X_{\min}}{\Delta X} = \frac{250 - 0}{0{,}1} = 2500.$$

Визначимо кількість інформації, що міститься у повідомленні про результат вимірювання величини $I = \log_2 N = \log_2 2500 = 11{,}2877$ біт.

Оскільки повідомлення, що передаються, містять зазвичай ціле число біт, то доцільно округлити результати до найближчого більшого цілого $I_{\text{ц}} = \lceil 11{,}2877 \rceil = 12$ біт. ■

Приклад 2.18. Визначити кількість інформації, що міститься в результаті виміру випадкової величини, рівномірно розподіленої в діапазоні –5…+10 В. Необхідна точність виміру 1 мВ.

Рішення. Число рівнів, на які ділиться квантована величина *X*, складе

$$N = \frac{X_{\max} - X_{\min}}{\Delta X} = \frac{+10 - (-5)}{0{,}001} = 15000.$$

Визначимо кількість інформації, що міститься у повідомленні про результат вимірювання величини $I = \log_2 N = \log_2 15000 = 13{,}8727$ біт.

Округлимо результати до найближчого цілого $I_{\text{ц}} = \lceil 13{,}8727 \rceil = 14$біт. ■

У разі, якщо вимірювана величина розподілена згідно із законом, який відрізняється від рівномірного, доцільно дослідити її статистичні характеристики. Відомо, що неперервне повідомлення як випадкова величина *X* характеризується диференціальним законом розподілу ймовірностей (щільністю розподілу) $f(x)$ (1.39). Формулу для ентропії джерела безперервних повідомлень можна отримати з формули (2.12) для ентропії дискретного джерела. Для цього розіб'ємо діапазон зміни безперервної випадкової величини *x* на кінцеве число *m* малих інтервалів шириною Δx і розглянемо множину можливих значень $\{x_1, \ldots, x_m\}$, що збігаються зі значеннями рівнів квантування.

Відповідно до (1.40) ймовірність попадання випадкової величини *X* в інтервал $[x_i, x_i + \Delta x]$ складе

$$p(x_i \le x \le x_i + \Delta x) = \int_{x_i}^{x_i + \Delta x} f(x)dx. \qquad (2.36)$$

Оскільки Δx мале, можна вважати, що значення $f(x)$ у межах інтервалу $[x_i, x_i + \Delta x]$ приблизно постійне і дорівнює $f(x_i)$. Тоді (2.36) набуває вигляду

$$p(x_i \le x \le x_i + \Delta x) = \int_{x_i}^{x_i+\Delta x} f(x)dx \approx f(x_i)\Delta x. \qquad (2.37)$$

Враховуючи, що число інтервалів Δx утворює дискретну множину, можна визначити ентропію безперервної випадкової величини X за аналогією з ентропією дискретного джерела у вигляді

$$H_{дискр}(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) = -\sum_{i=1}^{n} f(x_i)\Delta x \log_2 f(x_i)\Delta x =$$
$$= -\sum_{i=1}^{n} f(x_i)\Delta x \log_2 f(x_i) - \sum_{i=1}^{n} f(x_i)\Delta x \log_2 \Delta x \qquad . \qquad (2.38)$$

З інтегралу щільності розподілу в нескінченно великому інтервалі випливає, що $\sum_{i=1}^{n} f(x_i)\Delta x = 1$. Тоді

$$H_{дискр}(X) = -\sum_{i=1}^{n} f(x_i)\Delta x \log_2 f(x_i) - \log_2 \Delta x \quad . \qquad (2.39)$$

По мірі зменшення Δx ймовірність $p(x_i \le x \le x_i+\Delta x)$ дедалі більше наближається до ймовірності $p(x_i)$, а властивості дискретної величини – до властивостей безперервної випадкової величини. В результаті граничного переходу, з урахуванням $\lim_{\Delta x \to 0} \sum_{i=1}^{n} f(x_i)\Delta x = \int_{-\infty}^{\infty} f(x)dx = 1$ отримуємо

$$H_{квант}(X) = \lim_{\Delta x \to 0} H_{дискр}(X) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx - \log_2 \Delta x. \qquad (2.40)$$

Перший доданок в (2.39) має кінцеве значення, яке залежить тільки від закону розподілу безперервної випадкової величини X і не залежить від кроку квантування Δx. Величина $\log_2 \Delta x$ залежить тільки від обраного інтервалу квантування Δx , тобто від точності вимірювання і при Δx = const вона також постійна.

Приклад 2.19. Визначимо кількість інформації, що міститься в одноразовому вимірі, рівномірно розподіленої в інтервалі [a, b] випадкової величини, квантованої з кроком Δx.

Рішення. Відповідно до (1.44) щільність ймовірності рівномірно розподіленої випадкової величини $f(x)=\frac{1}{b-a}=\frac{1}{\varepsilon}$. Тоді (2.40) набуває вигляду:

$$H_{квант}(X)=-\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx-\log_2\Delta x=-\int_{a}^{b}\frac{1}{b-a}\log_2\frac{1}{b-a}dx-\log_2\Delta x=$$

$$-\frac{b-a}{b-a}\cdot\log_2\frac{1}{b-a}-\log_2\Delta x=-\left(\log_2\frac{1}{b-a}+\log_2\Delta x\right)=\log_2\frac{b-a}{\Delta x},$$

що збігається з (2.35). ■

Приклад 2.20. Визначимо кількість інформації, що міститься в одноразовому вимірі квантованої з кроком Δx=0,1 нормально розподіленої випадкової величини з математичним сподіванням $M(X)$=0 дисперсією $D(X)=\sigma_x^2$=4.

Рішення. Відповідно до (1.42) щільність розподілу нормального закону з нульовим математичним сподіванням задається формулою

$$f(x)=\frac{1}{\sigma_x\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma_x^2}}.$$

Підставляючи (1.42) в (2.40), отримуємо:

$$\begin{aligned}
&H_{квант}(X)=-\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx-\log_2\Delta x=\\
&-\frac{1}{\ln 2}\int_{-\infty}^{\infty} f(x)\cdot\ln f(x)dx-\log_2\Delta x=\\
&=-\frac{1}{\ln 2}\int_{-\infty}^{\infty} f(x)\cdot\left[\ln\frac{1}{\sigma_x\sqrt{2\pi}}-\frac{x^2}{2\sigma_x^2}\right]dx-\log_2\Delta x= \qquad (2.41)\\
&=-\frac{1}{\ln 2}\cdot\left[\ln\frac{1}{\sigma_x\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)dx-\int_{-\infty}^{\infty} f(x)\cdot\frac{x^2}{2\sigma_x^2}dx\right]-\log_2\Delta x
\end{aligned}$$

Оскільки $\int_{-\infty}^{\infty} f(x)dx=1$, а згідно (1.41) $\int_{-\infty}^{+\infty} x^2 f(x)dx=D(X)=\sigma_x^2$, то (2.41) набуде вигляду

$$H_{квант}(X)=-\frac{1}{\ln 2}\cdot\ln\sqrt{2\pi e\sigma_x^2}-\log_2\Delta x=\log_2\frac{\sqrt{2\pi e\sigma_x^2}}{\Delta x}$$

Підставивши чисельні значення, отримуємо

$$H_{квант}(X) = \log_2 \frac{\sqrt{2\pi e \cdot 4}}{0{,}1} = \log_2 170{,}0795 = 7{,}4161 \text{ біт}. \blacksquare$$

Приклад 2.21. Джерело передає до лінії зв'язку у випадковий час пакети інформації. Інтервал між пакетами розподілений за експоненціальним законом із щільністю розподілу $f(x)=e^{-\lambda x}$, де $\lambda=2$мс$^{-1}$. Точність виміру інтервалу $\Delta x=10$мкс$=0{,}01$мс. Визначити кількість інформації, що міститься у одноразовому вимірі часу між пакетами.

Рішення. Підставляючи вираз (1.45) для щільності експоненціального розподілу в (2.40), отримуємо:

$$H_{квант}(X) = -\int_0^\infty f(x)\log_2 f(x)dx - \log_2 \Delta x = -\frac{1}{\ln 2}\int_0^\infty f(x)\cdot \ln f(x)dx - \log_2 \Delta x =$$

$$= -\frac{1}{\ln 2}\int_0^\infty e^{-\lambda x}\cdot \ln e^{-\lambda x}dx - \log_2 \Delta x = \frac{1}{\ln 2}\int_0^\infty e^{-\lambda x}\cdot(-\lambda x)dx - \log_2 \Delta x =$$

$$= -\frac{\lambda}{\ln 2}\cdot e^{-\lambda x}\left[\frac{x}{-\lambda} - \frac{1}{\lambda^2}\right]\Bigg|_0^\infty - \log_2 \Delta x = \frac{1}{\lambda \ln 2} - \log_2 \Delta x = \frac{\ln e}{\lambda \ln 2} - \log_2 \Delta x =$$

$$= \frac{\log_2 e}{\lambda} - \log_2 \Delta x = \log_2 \frac{e}{\lambda \Delta x}.$$

Підставивши чисельні значення, маємо

$$H_{квант}(X) = \log_2 \frac{e}{2\cdot 0{,}01} = \log_2 135{,}9141 = 7{,}0866 \text{ біт}. \blacksquare$$

2.6 Ентропія безперервного джерела. Диференціальна ентропія

Крім ситуацій, коли аналогові величини перетворюються на дискретні за допомогою використання пристроїв дискретизації та квантування існує чимало таких систем, в яких інформація передається і перетворюється у формі безперервних сигналів. Прикладами можуть бути системи телевимірювань з частотним поділом сигналів. Спробуємо визначити ентропію безперервного джерела. В результаті граничного переходу в (2.40) при $\Delta x \to 0$ отримуємо

$$H_{безпер}(x) = \lim_{\Delta x\to 0} H_{квант}(X) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx - \lim_{\Delta x\to 0}\log_2 \Delta x. \qquad (2.42)$$

Перший член виразу (2.42) залежить тільки від закону розподілу безперервної випадкової величини X і має таку структуру як ентропія дискретного джерела. Другий член виразу (2.42) при $\Delta x \to 0$ прагне нескінченності. Отже, ентропія безперервного повідомлення має дорівнювати нескінченності, що повністю відповідає інтуїтивному уявленню про те, що невизначеність вибору нескінченно великої кількості можливих станів нескінченно велика.

Щоб все ж таки кількісно оцінити ентропію безперервного повідомлення, доцільно ввести відносну міру невизначеності досліджуваної безперервної випадкової величини X по відношенню до заданої. Як задану величину X_0 візьмемо безперервну випадкову величину, рівномірно розподілену на інтервалі $[a, b]$ с шириною $\varepsilon = b - a$. Відповідно до (1.44) її щільність ймовірності $f(x)=1/\varepsilon$, а ентропія

$$H(X_0) = -\int_a^b \frac{1}{\varepsilon} \log_2 \frac{1}{\varepsilon} dx - \lim_{\Delta x \to 0} \log_2 \Delta x = \log_2 \varepsilon - \lim_{\Delta x \to 0} \log_2 \Delta x . \qquad (2.43)$$

Поклавши для простоти $\varepsilon = 1$, визначимо величину

$$H_\Delta(X) = H(X) - H(X_0) = -\int_{-\infty}^{\infty} f(x) \log_2 f(x) dx , \qquad (2.44)$$

яка показує, наскільки невизначеність безперервної випадкової величини X із щільністю розподілу $f(x)$ більша або менша за невизначеність випадкової величини, розподіленої рівномірно на інтервалі $\varepsilon = 1$. Тому величину

$$H_\Delta(x) = -\int_{-\infty}^{\infty} f(x) \log_2 f(x) dx \qquad (2.45)$$

називають *диференціальною ентропією* безперервного джерела інформації.

На відміну від ентропії джерел дискретних повідомлень диференціальна ентропія може набувати позитивних, негативних і нульових значень і має сенс середньої невизначеності вибору випадкової величини з довільним законом розподілу за вирахуванням невизначеності випадкової величини, рівномірно розподіленої в одиничному інтервалі.

Приклад 2.22. Вимірювана величина змінюється в межах від a=10 до b=70 ампер та розподілена за рівномірним законом. Визначити диференціальну ентропію цієї випадкової величини.

Рішення. Підставляючи вираз для щільності рівномірного розподілу (1.44), отримуємо

$$H_{\Delta}(X) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx = -\int_{a}^{b} \frac{1}{b-a}\log_2 \frac{1}{b-a}dx = -\frac{b-a}{b-a}\cdot\log_2\frac{1}{b-a}$$

$$= -\left(\log_2 \frac{1}{b-a}\right) = \log_2(b-a) = \log_2(70-10) = \log_2 60 = 5{,}9069. \blacksquare$$

Приклад 2.23. Амплітуда сигналу розподілена за нормальним законом дисперсією $D(X)=\sigma_x^2$=12. Визначити диференціальну ентропію амплітуди сигналу.

Рішення. Відповідно до результатів прикладу 2.20.

$$H_{\Delta}(X) = -\frac{1}{\ln 2}\cdot\ln\sqrt{2\pi e\sigma_x^2} = \log_2\sqrt{2\pi e\sigma_x^2} = \log_2\sqrt{204{,}9536} = \log_2 14{,}3162 = 3{,}8395. \blacksquare$$

Приклад 2.24. Визначити диференціальну та повну ентропію безперервного джерела інформації з диференціальним законом розподілу

$$f(x) = \begin{cases} 0, x < 0 \\ \dfrac{x}{2},\ 0 \le x \le 2 \\ 0, x > 2 \end{cases}.$$

Точність виміру Δx=0,1.

Рішення. Підставимо в (2.45) вираз для щільності ймовірності рівномірного розподілу $f(x)=x/2$ і визначимо диференціальну ентропію.

$$H_{\Delta}(x) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx = -\int_{0}^{2}\frac{x}{2}\cdot\log_2\frac{x}{2}dx = -\frac{1}{\ln 2}\int_{0}^{2}\frac{x}{2}\cdot\ln\frac{x}{2}dx =$$

$$= -\frac{1}{\ln 2}\cdot\left(\left(2\ln\frac{x}{2}-1\right)\cdot\frac{x^2}{8}\right)\Bigg|_0^2 = \frac{1}{2\cdot\ln 2} = 0{,}7213.$$

Згідно (2.40)

$$H_{повн}(X) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx - \log_2\Delta x = \frac{1}{2\cdot\ln 2} - \log_2 0{,}2 = 0{,}7213 + 3{,}3219 = 4{,}0432.$$

2.7 Визначення кількості інформації, що міститься в дискретних сигналах

У багатьох випадках буває необхідним зберігати і обробляти не просто окремі повідомлення, а фізичні величини, що змінюються в часі - сигнали.

Більшість сигналів, що використовуються в автоматиці, зв'язку, побутовій техніці описується неперервною або кусочно-неперервною функцією $x(t)$, причому і аргумент t і сам сигнал можуть набувати будь-яких значень в деяких інтервалах $t_{min} \le t \le t_{max}$, $x_{min} \le x \le x_{max}$. Такі сигнали називаються *неперервними* чи *аналоговими*. До аналогових сигналів відносяться сигнали, які знімаються з датчиків температури, тиску і т.д. в системах управління, звукові сигнали, що надходять з мікрофонів, телевізійні, електрокардіографічні та інші.

У той самий час, дедалі ширше застосування знаходять цифрові системи, які обробляють послідовності двійкових кодів. Для зберігання і обробки у таких системах сигнали повинні бути піддані *дискретизації за часом* і *квантуванню за рівнем*.

Дискретні сигнали визначаються в дискретні моменти часу і є послідовністю чисел (часовим рядом) $x(iT_д)$. Кожен елемент послідовності може набувати будь-яких значень у певному інтервалі, у той час як номер відліку сигналу i приймає цілі дискретні значення $i = 0, 1, ...$ (приклад дискретного ситналу зображений на рис 2.3, *б*). При цьому проміжок часу $T_д$, через який беруться відліки сигналу x називається *інтервалом дискретизації*. Величина $f_д=1/T_д$ – *частота дискретизації*.

При *квантуванні за рівнем* вся область значень сигналу розбивається на дискретні рівні, причому відстань між суміжними рівнями називається кроком квантування Δx. Кожен із рівнів квантування кодується двійковим кодом, що складається з n двійкових розрядів. Число розрядів вибирається відповідно до співвідношення $n \ge \log_2 N$, де N – число рівнів квантування. *Квантований сигнал* представлений на рис. 2.3, *в*.

До *цифрових* (*дискретно-квантованих*) сигналів (рис 2.3, *г*) відносять ті, у яких дискретними є як незалежна змінна (наприклад, час), так і рівень сигналу. Для того, щоб представити аналоговий сигнал послідовністю чисел кінцевої розрядності, його слід перетворити на дискретний сигнал, а

потім піддати квантуванню і записати відліки у вигляді двійкових кодів, тобто виконати *аналого-цифрове перетворення*. При цьому для кодування значень відліків можуть використовуватися прямі, зворотні або додаткові коди (див. розділ 1), або якісь спеціалізовані коди. Отримана послідовність двійкових кодів може оброблятися, зберігатися в пам'яті або передаватися по каналу зв'язку за допомогою *імпульсно-кодової модуляції*. Процес відновлення сигналу за його цифровими відліками називається *цифро-аналоговим перетворенням*.

Приклад 2.25. Обчислити значення відліків сигналу $x(t)$, заданого у межах t =0…1.

$$x(t)=A_1\sin(2\pi f_1 t+\varphi_1)+A_2\sin(2\pi f_2 t+\varphi_2)+A_3\sin(2\pi f_3 t+\varphi_3) = 0{,}4\sin(2\pi\cdot 1\cdot t+\pi/12) + \\ +0{,}6\sin(2\pi\cdot 2t+5\pi/12)+0{,}7\sin(2\pi\cdot 3t+\pi/2).$$

Частота дискретизації $f_д$ =16. Відліки кодуються прямим кодом, один двійковий розряд виділяється на знак, два на цілу частину, два на дробову.

Рішення. Визначимо величину інтервалу дискретизації $T_д=1/f_д=1/16=0{,}0625$. Значення дискретних відліків сигналу $x(iT_д)$ виходять шляхом підстановки величин $iT_д$ замість t в (5.13), тобто

$$x_{дискр}(t)=A_1\sin(2\pi f_1 T_д+\varphi_1)+A_2\sin(2\pi f_2 T_д+\varphi_2)+A_3\sin(2\pi f_3 T_д+\varphi_3)= \\ =0{,}4\sin(2\pi\cdot i\cdot(1/16)+\pi/12)+0{,}6\sin(2\pi\cdot i\cdot(2/16)+5\pi/12)+0{,}7\sin(2\pi\cdot i\cdot(3/16)+\pi/2).$$

Цифрові відліки $x_{цифр}(iT_д)$ можна обчислити шляхом заокруглення значень $x(iT_д)$ до найближчого значення, кратного 0,25, оскільки два розряди дробової частини дозволяють виражати числа з дробовими частинами 0, 0,25, 0,5, 0,75. Відліки $x_{цифр}(iT_д)$ записуються у двійковому коді. Значення відліків зведено в табл.2.6, графічно результати перетворення представлені на рис.2.3.

Таблиця 2.6 – Результати перетворення аналогового сигналу на цифровий

i	$iT_д$	$x(iT_д)$	$x_{цифр}(iT_д)$	
			десятковий код	двійковий код
0	0	1.3831	1.50	00110
1	0.0625	1.0310	1	00100
2	0.125	0.0067	0	00000
3	0.1875	–0.5501	–0.5	10010
4	0.25	–0.1932	–0.25	10001
5	0.3125	0.4444	0.5	00010
6	0.375	0.5397	0. 5	00010
7	0.4375	0.0843	0	00000
8	0.5	–0.2240	–0.25	10001
9	0.5625	0.0082	0	00000
10	0.625	0.3039	0.25	00001
11	0.6875	–0.0499	0	00000
12	0.75	–0.9659	–1	10100
13	0.8125	–1.4837	–1.5	10110
14	0.875	–0.8503	–0.75	10011
15	0.9375	0.5157	0.5	00010
16	1.0	1.3831	1.5	00110

Важливим питанням при побудові аналого-цифрових та цифро-аналогових перетворювачів є вибір частоти дискретизації, що забезпечує, з одного боку рийнятну якість відновлення сигналів, з іншого – допустимі програмні та апаратні витрати.

Для класу сигналів з обмеженим спектром відновлення сигналу найбільш ефективно здійснюється на основі *теореми відліків* , що повязується з іменами *В. Котельникова*, *Г. Найквіста та К. Шеннона*) [18]. Теорема відліків свідчить: якщо аналоговий сигнал $x(t)$ має спектр, обмежений верхньою граничною частотою f_{max}, то сигнал може бути однозначно відновлений за послідовністю дискретних відліків $x(iT_д)$, взятих через інтервали часу $T_д \le \frac{1}{2f_{max}}$. Частота $f_д=2f_{max}$ часто називається *частотою Найквіста*

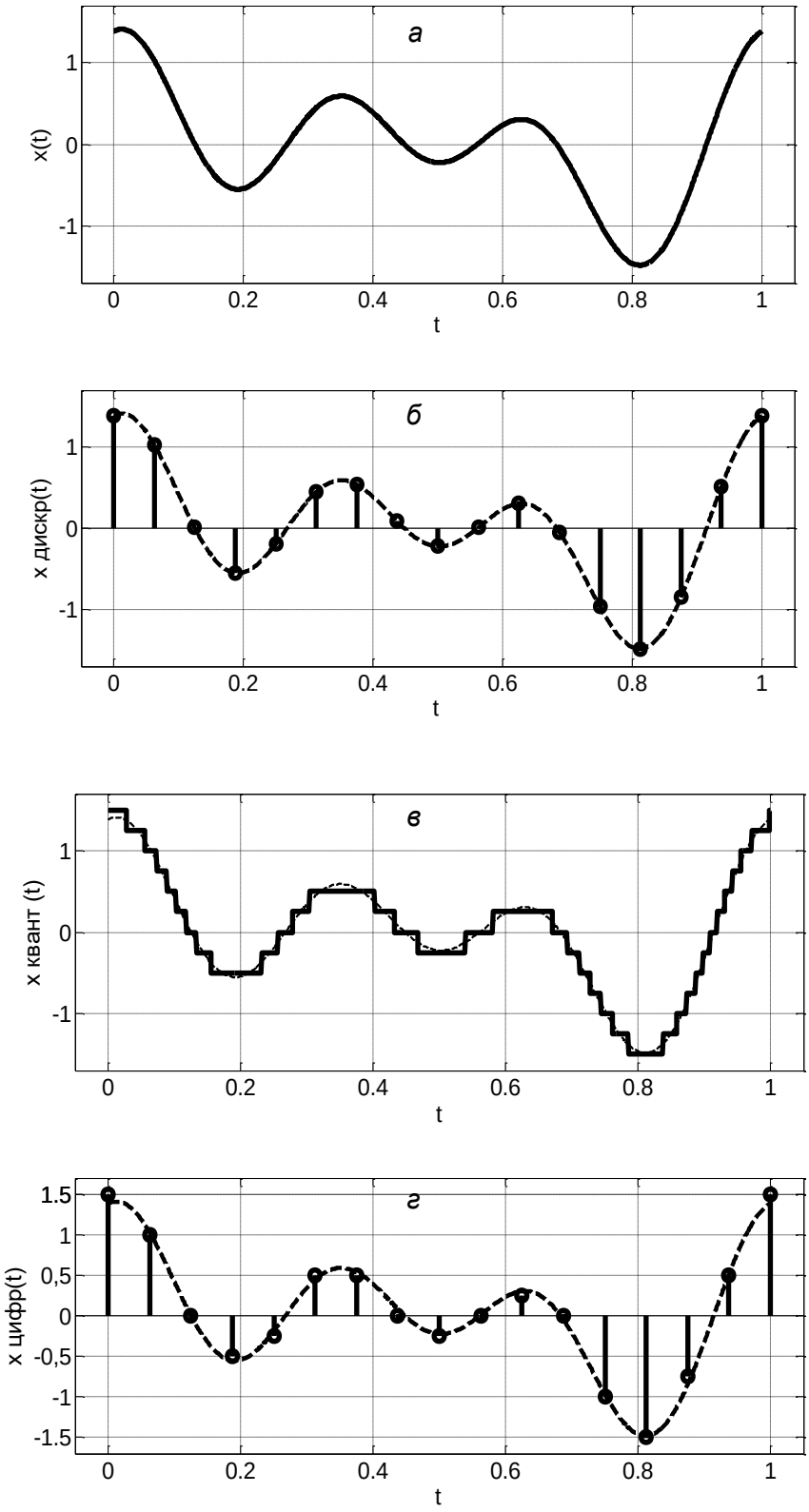


Рисунок 2.3 – Аналого-цифрове перетворення сигналу (*а* – неперервний сигнал, *б* – дискретний сигнал, *в* – квантований сигнал, *г* – цифровий сигнал)

Можна показати [18], що неперервний сигнал відновлюється за його дискретними відліками, якщо на вхід ідеального фільтра нижніх частот зі смугою пропускання $0...f_{max}$ подати послідовність δ-функцій $\delta(t-iT_д)$, помножених на значення відліків $x(iT_д)$. Однак, ані сигнал у вигляді δ-функції, ані ідеальний фільтр нижніх частот фізично не реалізуються. Тому на практиці замість δ-функцій використовують короткі імпульси, а замість ідеального фільтра – реальний фільтр нижніх частот, що природно призводить до похибки відновлення. Крім того, всі реальні сигнали кінцеві за часом і мають необмежений за частотою спектр, що також викликає похибки.

Тим не менше, теорема відліків має велике практичне значення під час вибору частоти дискретизації. Зазвичай $f_д$ визначають за наближеною формулою

$$f_д \approx k \cdot 2 f_{max} , \qquad (2.46)$$

де k – коефіцієнт, що залежить від заданої точності і методу відновлення і що лежить, зазвичай, в межах 1.1...2.5, а іноді і більше.

Приклад 2.26. Визначити частоту дискретизації телефонних сигналів. Спектр мовного сигналу перебуває у діапазоні частот 0,3...3,4 кГц.

Рішення. Частота Найквіста для мовних сигналів становитиме $f_н = 2f_{max} = 2 \cdot 3,4 = 6,8$ кГц.

Відповідно до стандарту G.711 Міжнародної спілки електрозв'язку для дискретизації телефонних сигналів обрано частоту $f_д$ =8кГц. Таким чином, коефіцієнт запасу дорівнює $k = f_д / f_н = 8/6,8 = 1,176$. ■

Приклад 2.27. Визначити частоту дискретизації аудіосигналів. Діапазон частот, що розрізняються людиною на слух, знаходиться в межах 20 Гц – 20 кГц.

Рішення. Частота Найквіста для звукових сигналів становитиме $f_н = 2f_{max} = 2 \cdot 20 = 40$ кГц.

Формат звукозапису DAT передбачає частоту $f_д$ =48кГц. Таким чином, коефіцієнт запасу дорівнює $k = f_д / f_н = 48/40 = 1,2$.■

Таким чином, в інженерній практиці при виборі параметрів аналого-цифрового перетворення розрядність АЦП визначається відповідно до співвідношення (2.34), частота дискретизації – на основі (2.46).

Швидкість передачі лініями зв'язку перетвореної в послідовності двійкових кодів інформації визначається як

$$C=n\cdot f_д \text{ біт/с}, \tag{2.47}$$

де n – число біт, що припадають на кожен відлік інформації, $f_д$ – частота дискретизації.

Обсяг переданої інформації, який визначає ємність пам'яті, необхідної для запису оцифрованого сигналу

$$V = C\cdot t= n\cdot f_д\cdot t \text{ біт}, \tag{2.48}$$

де t – час передачі.

Приклад 2.28. Визначити швидкість передачі сигналу цифрової телефонії каналом зв'язку. Розрядність кодів, що передаються n=8, частота дискретизації $f_д$ =8кГц.

Рішення. Відповідно до (2.47) швидкість передачі сигналів цифрової телефонії за допомогою імпульсно-кодової модуляції складе

$C=n\cdot f_д$ = 8·8 = 64000 біт/с ≈ 64кбіт/с. ■

Слід зазначити, що вихідна послідовність відліків мовного сигналу зі швидкістю 64 кбіт/с утворює так званий основний цифровий сигнал (канал), передбачений стандартом G.711, який потім, як правило, піддається спеціальному кодуванню. У реальних мережах мобільної телефонії проводиться стиснення мовного сигналу, що дозволяє значно скоротити цифровий потік. Так, стандарт G.729, що використовується в стільниковому зв'язку, передбачає передачу мовного сигналу зі швидкістю 6,4 Кбіт/сек, тобто забезпечує стиснення в десять разів, стандарт кодування мови Міністерства оборони США MELP (англ. *Mixed-excitation linear prediction, лінійне передбачення зі змішаним збудженням*) – зі швидкістю 2400 біт/с і менше, тобто стиснення в 26,7 і більше разів.

Приклад 2.29. Визначити потрібний об’єм пам'яті для запису оцифрованого музичного твору. Число бітів на відлік n=16, частота дискретизації $f_д$=44,1 кГц, тривалість твору 74 хвилини (саме стільки звучить Симфонія Бетховена №9, на основі якої вибиралися параметри перших CD дисків).

Рішення. Відповідно до (2.48) визначимо необхідний об’єм пам'яті для запису оцифрованого сигналу.

Перекладемо 74 хвилини на секунди: $t = 74 \cdot 60 = 4440$ с.

$V = n \cdot f_д \cdot t$ біт = 16·44100·4440 = 3132864000 біт = 3409920000/1024 = 30594375 Кбит = 30594375 /1024 = 2987,7319 Мбіт = 373,4665 Мбайт. ■

Приклад 2.30. Система холтерівського моніторингу здійснює добовий запис оцифрованої електрокардіограми. Визначити потрібний об'єм пам'яті для запису ЕКГ. Число бітів на відлік n=12, частота дискретизації $f_д$ =400 Гц.

Рішення. Перекладемо час запису на секунди. $t = 24 \cdot 3600 = 86400$ с. Відповідно до (5.16) необхідний об'єм пам'яті для запису оцифрованого сигналу:

$V = n \cdot f_д \cdot t$ бит = 12·400·86400 = 414720000 біт = 414720000/1024 = 405000 Кбіт = 405000 /1024 = 395,5078 Мбіт = 49,4385 Мбайт. ■

Приклад 2.31. Визначити потрібний об'єм пам'яті для запису оцифрованого відео. Кожен піксель кодується одним байтом для кожної з трьох компонентів моделі RGB, тобто трьома байтами. Розмір кадру 1280×720 пікселів, частота кадрів 50 Гц, тривалість фільму дві години.

Рішення. Необхідний об'єм пам'яті для запису оцифрованого сигналу $V = N_{пикс} \cdot n \cdot f_д \cdot t$

Перекладемо час запису за секунди. $t = 2 \cdot 3600 = 7200$ с.

Число пікселів у зображенні

$N_{пикс} = N_1 \cdot N_2 = 1280 \cdot 720 = 921600$ пікселів.

Підставивши чисельні значення, визначаємо:

$V = 921600 \cdot 3 \cdot 50 \cdot 7200 = 995328000000$ байт = $9{,}9533 \cdot 10^{11}$ байт = =995328000000/1024=972000000 Кбайт=972000000/1024=949218,75 Мбайт = 949218,75 /1024 = 926,9714 Гбайт. ■

Невідповідність отриманого результату життєвому досвіду, згідно з яким фільм займає одиниці, в крайньому випадку, 10-30 гігабайт, пояснюється тим, що як між суміжними пікселями, так і між суміжними кадрами існує статистична залежність, тобто відеоінформація має значну надмірність. Тому при передачі та зберіганні відеосигналів застосовуються методи стиснення, найбільш відомим з яких є MPEG-4 – міжнародний стандарт, що використовується переважно для стиснення цифрового аудіо та відео, і дозволяє стискати інформацію в десятки, а іноді й у сотні разів.

Для оцінювання необхідної пропускної здатності каналу та обсягу пам'яті для зберігання оцифрованого сигналу можуть бути корисними формули (2.47) та (2.48).

Приклад 2.32. Визначити об’єм пам'яті, потрібний для запису оцифрованого сигналу. Діапазон зміни сигналу –100…+200 В, необхідна точність вимірювання Δ_x=0,02В, тривалість сигналу 30 хвилин, верхня гранична частота f_{max} = 10кГц, коефіцієнт запасу за частотою k=1,5.

Рішення. Загальний обсяг інформації, що міститься в оцифрованому сигналі, відповідно до (2.48) визначається за формулою

$I=n_{біт} \cdot N_{від}$,

де $n_{біт}$ – число біт, якими кодується кожен відлік сигналу, N – число відліків сигналу.

$n_{біт} = \lceil \log_2 N_{рівн} \rceil$,

де $N_{рівн}$ – число рівнів, до яких округляються відліки сигналу

Число рівнів, на які ділиться квантована величина X, складе

$$N_{рівн} = \frac{X_{max} - X_{min}}{\Delta X} = \frac{200-(-100)}{0{,}02} = 15000.$$

Визначимо мінімальне число біт, потрібне для кодування кожного відліку $n = \log_2 N_{рівн} = \log_2 15000 = 13{,}8727$ біт.

Оскільки повідомлення, що передаються, містять зазвичай ціле число біт, то доцільно округлити результати до найближчого більшого цілого:

$n_{біт} = \lceil 13{,}8727 \rceil = 14$ біт.

Число відліків, що містяться в сигналі, визначається як $N_{від}=f_д \cdot t$,

де $f_д$ – частота дискретизації сигналу, t – тривалість сигналу.

$f_д$ визначається з теореми відліків $f_д = k \cdot 2 \cdot f_{max}$,

де k – коефіцієнт запасу, що вибирається в залежності від вимог до якості відтворення сигналу.

У даному прикладі f_{max} =10 Кгц, коефіцієнт запасу k=1,5,

t=30 хвилин = 1800 с.

$f_д = k \cdot 2 \cdot f_{max} = 1{,}5 \cdot 2 \cdot 10 = 30$ КГц.

Число відліків у сигналі

$N_{від} = f_д \cdot t = 30000 \cdot 1800 = 54000000$.

Загальний обсяг інформації

$I = n_{біт} \cdot N_{від} = 14 \cdot 54000000 = 756000000$ біт = 756000000/8 =

=94500000 байт =94500000/1024 = 92285,15625 Кбайт =

= 92285,15625/1024 = 90,1222 Мбайт. ■

Завдання для самостійного вирішення

1. Джерело може передавати символи, кожен з яких представляє число в діапазоні від 1 до 10000. Визначити обсяг інформації, що міститься в символі, виражений у бітах, натах, тритах та дитах.

2. Інформація передається за допомогою чотирипозиційної фазової модуляції. Визначити, скільки біт інформації міститься у повідомленні з десяти символів, що не повторюються.

3. Визначити, скільки біт інформації міститься у повідомленні з десяти четвіркових символів, якщо символи можуть повторюватися.

4. Визначити, скільки біт інформації міститься у повідомленні із трьох трайтів троїчних символів.

5. До каналу зв'язку передаються двадцятирозрядні двійкові слова, у кожному з яких міститься рівно чотири одиниці, що знаходяться на довільних місцях. Визначити, скільки бітів інформації можна передати таким кодом.

6. У канал зв'язку передається інформація про параметри синусоїдального сигналу. Амплітуда сигналу – ціле число в інтервалі від одиниці до 1000, частота - ціле число від 0 до 3999, фаза – ціле число від 0 до 179. Скільки інформації міститься у відомостях про сигнал?

7. Яка кількість інформації буде у друкованому документі обсягом N_1=400 сторінок. На сторінці друкованого тексту міститься N_2=40 рядків за N_3=72 символами. Кожен символ кодується одним байтом.

8. Яка кількість інформації міститься в растровом зображенні розміром N_1=1024 на N_2=800 пікселів. Кожен піксель представляється у форматі HighColor з використанням п'яти біт для кодування червоного кольору (R), п'ять біт для синього (B) та шести для зеленого (G).

9. Вісім підключених до каналу зв'язку джерел у довільному порядку по черзі відправляють в канал по одному символу, що відповідає своєму номеру. Визначити кількість інформації в повідомленні про порядок передачі номерів у канал.

10. Джерело передає в канал двійкові символи, причому ймовірність появи нуля становить p_0=0,7, ймовірність появи одиниці p_1=0,3. Визначити кількість інформації в нульовому та одиничному повідомленнях. Визначити кількість інформації у повідомленні 10101.

11. Джерело має передати в канал у довільному порядку вісім одиниць та чотири нуля. Визначити кількість інформації в повідомленні, що перші чотири символи в каналі будуть 1010.

12. Визначити кількість інформації, що міститься у повідомленні, що у десятибітному слові міститься три помилки. Ймовірність бітової помилки у каналі $p_{пом}$=0,15.

13. Джерело передає до каналу символи, що відповідають числам від одного до чотирьох. Ймовірності появи символів такі:

$P(X=1) = 0,4$; $P(X=2) = 0,1$; $P(X=3) = 0,3$; $P(X=4) = 0,2$;

Визначити кількість інформації у повідомленні, що число, передане в канал непарне.

14. Джерело передає в канал незалежні один від одного посилки 1111 та 0000 з ймовірностями відповідно 0,6 та 0,4. При передачі каналом зв'язку кожен розряд незалежно від інших може змінювати своє значення з нуля на одиницю з ймовірністю 0,3, з одиниці на нуль з ймовірністю 0,2. Визначити кількість інформації у повідомленні про прийом із каналу комбінації 1011.

15. По лінії зв'язку посилаються сигнали 1 та 0 з ймовірностями відповідно p_1=0,4, p_0=0,6. Якщо посилається сигнал 1, то з ймовірностями r_{11} = 0,8, r_{10} =0,2 приймаються відповідно сигнали 1 і 0. Якщо передається сигнал 0, то з ймовірностями r_{01} = 0,3, r_{00} =0,7 приймаються відповідно сигнали 1 і 0. Визначити кількість інформації у повідомленнях про прийом нуля та одиниці.

16. Три джерела незалежно один від одного передають у канал по одному бінарному символу. Для першого джерела ймовірність передачі канал одиниці p_1=0,3, для другого p_2=0,4, для третього p_3=0,8. Визначити кількість інформації в повідомленні, що в канал була передана хоча б одна одиниця.

17. Джерело повідомлень видає символи з алфавіту $\{x_1,\ldots,x_5\}$ із ймовірностями p_1=0,25; p_2=0,15; p_3=0,1; p_1=0,28; p_2=0,22. Визначити кількість інформації в повідомленні по Гартлі та Шеннону.

18. Джерело передає в канал незалежні одна від одної посилки 101 та 010 з ймовірностями відповідно 0,6 та 0,4. При передачі каналом зв'язку кожен розряд незалежно від інших може змінювати своє значення протилежне з ймовірністю 0,2. Визначити ентропію повідомлення, що прийнято посилку 111.

19. Визначити ентропії $H(X)$, $H(Y)$, $H(X/Y)$, $H(X, Y)$, якщо задана матриця ймовірностей станів системи, що поєднує джерела X та Y

$$P(x_i, y_j) = \begin{vmatrix} 0.4 & 0.1 & 0 \\ 0 & 0.2 & 0.1 \\ 0 & 0 & 0.2 \end{vmatrix}$$

20. Ансамблі подій X та Y об'єднані. Імовірності спільних подій (x_i, y_j) описуються матрицею

	Y_1	Y_2	Y_3
X_1	0.1	0.2	0.3
X_2	0.25	0	0.15

Визначити

1) ентропію ансамблів X та Y;

2) ентропію об'єднаного ансамблю (X, Y);

3) умовні ентропії ансамблів;

4) кількість інформації, що міститься в подіях y щодо x.

21. Визначити кількість інформації, що міститься в результаті вимірювання випадкової величини, рівномірно розподіленої в діапазоні ±10В. Необхідна точність вимірювання 1 мВ.

22. Визначити кількість інформації, що міститься в одноразовому вимірі величини, квантованої з кроком Δx=0,1, нормально розподіленої з математичним сподіванням $M(X)$=0, дисперсією $D(X)=\sigma_x^2$ =12.

23. Визначити кількість інформації, що міститься в одноразовому вимірі величини, квантованої з кроком Δx=0,1, рівномірно розподіленої з математичним сподіванням $M(X)$=0, дисперсією $D(X)=\sigma_x^2$ =12.

24. Визначити кількість інформації, що міститься в одноразовому вимірі величини, квантованої з кроком Δx=0,1, експоненційно розподіленої з дисперсією $D(X)=\sigma_x^2$ =16.

25. Інформація передається за допомогою частотно-модульованих сигналів, робоча частота f яких змінюється з рівною ймовірністю в межах від f_1 = 20 МГц до f_2 = 40 МГц. Визначити ентропію частоти, якщо точність виміру частоти Δf = 5кГц.

26. Амплітуда сигналу розподілена за нормальним законом з дисперсією $D(X)=\sigma_x^2$ =25. Визначити диференціальну ентропію амплітуди сигналу.

27. Визначити диференціальну та повну ентропію безперервного джерела інформації з диференціальним законом розподілу.

$$f(x)=\begin{cases}0, x<0\\ 1-\dfrac{x}{4},\ 0\le x\le 4\\ 0, x>4\end{cases}$$

Точність виміру Δx=0,2.

28. Визначити диференціальну та повну ентропію безперервного джерела інформації з експоненційним законом розподілу та математичним очікуванням $M(X)$=0,25. Точність виміру Δx=0,02.

29. Визначити значення перших десяти відліків сигналу $x(t)$.

$x(t)$ = 5sin(2π·1000Гц·t+π/6) + 8sin(2π·2500t+π/6).

Частота дискретизації $f_д$ = 10кГц. Відліки кодуються додатковим кодом, один двійковий розряд виділяється на знак, чотири на цілу частину, три на дробову.

30. Верхня гранична частота спектра сигналу f_{max} = 120 кГц, частота дискретизації $f_д$= 300 кГц. Визначити коефіцієнт запасу за частотою.

31. Визначити потрібний об’єм пам'яті для запису електроенцефалограми. Число біт на відлік n = 16, частота дискретизації $f_д$=512 Гц, час запису 12 годин.

32. Одна хвилина запису цифрового аудіофайлу займає на диску 6,3086 Мбайт, розрядність відліків 20 біт. З якою частотою дискретизації записано звук?

33. Дві хвилини запису цифрового аудіофайлу займають на диску 16,4794 Мбайт. Частота дискретизації 48000 Гц. Яка розрядність аудіоадаптера?

34. До лінії зв'язку передається оцифрований відеосигнал. Кожен піксель кодується трьома байтами. Розмір кадру 1920×1080 пікселів, частота кадрів 50Гц. Після стиснення передача канал зв'язку відбувається зі швидкістю 25 Мбіт/с. Визначити ступінь стиснення відео.

35. Визначити об’єм пам'яті, потрібний для запису оцифрованого сигналу з датчику. Діапазон зміни сигналу ±20 В, необхідна точність вимірювання Δx=5 мВ, тривалість сигналу 1 година, верхня гранична частота f_{max} = 50кГц, коефіцієнт запасу за частотою k=1,2.

ГЛАВА 3

МЕТОДИ ЕФЕКТИВНОГО КОДУВАННЯ ТА СТИСНЕННЯ ДАНИХ

3.1 Принципи побудови ефективних кодів

Повідомлення, що передаються з використанням систем зв'язку (мова, музика, телевізійні зображення і т.д.) здебільшого призначені для безпосереднього сприйняття органами почуттів людини і у більшості випадків погано пристосовані для ефективної передачі каналами зв'язку. При цьому символи, що входять до повідомлення, зазвичай розподілені нерівномірно і можуть бути статистично пов'язані, що дозволяє не передавати частину повідомлення, відновлюючи його на прийомі за відомими статистичними властивостями.

Надмірність призводить до того, що за заданий проміжок часу буде передано менше повідомлень, і менш ефективно використовуватиметься канал передачі дискретних повідомлень. Завдання усунення надмірності на передачі в канал виконує *кодер джерела*.

Кодування повідомлень з метою усунення надмірності, тобто скорочення обсягу інформації та підвищення швидкості її передачі називають ентропійним, безнадлишковим або *ефективним кодуванням*, а також стисненням даних. Кодуванням повідомлень ансамблю $\{x_i,\ p(x_i)\}$ називається відображення множини повідомлень $A\{x_i\}$ у множину кодових слів $B\{x_i\}$. Символи за допомогою яких записане повідомлення становлять первинний алфавіт. Символи, за допомогою яких повідомлення трансформується в код, складають вторинний алфавіт. Кількість можливих символів у кодовому слові визначає значність коду n (наприклад, n=2).

Коди поділяються на рівномірні та нерівномірні. Коди, у яких повідомлення представлені комбінаціями з нерівною кількістю символом, називаються *нерівномірними* (кодами *змінної довжини*), і з рівною кількістю символом – *рівномірними* (кодами *фіксованої довжини*).

Для побудови рівномірного коду достатньо пронумерувати літери вихідного алфавіту і записати їх як m-розрядні числа в n-ичній системі числення, тобто кожній букві відповідає однакове число розрядів. Наприклад, при двійковому кодуванні 32 букв українського алфавіту використовується $m=\log_2 32=5$ розрядів.

Якщо елементи повідомлення з'являються на виході джерела з рівними ймовірностями $p(x_i)=1/N$ і незалежно один від одного, тоді надмірність дорівнює нулю

$$D = \frac{H_{\max}(X) - H(X)}{H_{\max}(X)} = \frac{\log_2 N - \log_2 N}{\log_2 m} = 0. \qquad (3.1)$$

Однак якщо літери вихідного алфавіту з'являються з різними ймовірностями, то рівномірний код є *надлишковим*. Оскільки ентропія (за умови, що всі літери алфавіту рівноймовірні) $H_{\max}=\log_2 m$ завжди більше ентропії $H(X)$ (отриманої з урахуванням нерівномірності появи різних літер алфавіту) то інформаційні можливості цього коду використовуються не повністю.

Приклад 3.1. Джерело має алфавіт із чотирьох символів А, Б, В, Г із ймовірностями p(А)=0,5; p(Б)=0,25; p(В)=p(Г)=0,125. Визначити надмірність повідомлень, складених із цього алфавіту.

Рішення. Ентропія джерела

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) =$$

$$= -0{,}5\log_2 0{,}5 - 0{,}25\log_2 0{,}25 - 0{,}125\log_2 0{,}125 - 0{,}125\log_2 0{,}125 =$$

$$= 0{,}5 + 0{,}5 + 0{,}375 + 0{,}375 = 1{,}75 \quad \textit{біт/символ}$$

Максимальна ентропія $H_{\max}= \log_2 4=2$.

Надмірність повідомлень складе

$$D = \frac{H_{\max}(X) - H(X)}{H_{\max}(X)} = \frac{2 - 1{,}75}{2} = 0{,}125. \blacksquare$$

Приклад 3.2. Відносні частоти появи букв українського алфавіту у зв'язному тексті відповідно до [5] зведені в табл. 3.1.

Таблиця 3.1 – Відносні частоти вживання букв української алфавіту

Літера	Частота	Літера	Частота	Літера	Частота	Літера	Частота
-	0,138	Е	0,042	Я	0,019	Ю	0,008
О	0,086	С	0,037	Ь	0,016	Ж	0,007
Н	0,068	К	0,033	Б	0,013	Ш	0,005
А	0,064	М	0,029	Г	0,013	Є	0,005
И	0,055	У	0,027	Ч	0,011	Щ	0,004
В	0,046	Д	0,027	Х	0,011	Ф	0,003
Т	0,045	Л	0,027	Ї	0,010	Ґ	<0,001
І	0,044	П	0,025	Ц	0,010		
Р	0,043	З	0,020	Й	0,009		

Визначити ентропію та середню надмірність тексту українською мовою.

Рішення. Ентропія текстового повідомлення:

$$H(X) = -\sum_{i=1}^{34} p(x_i)\log_2 p(x_i) =$$

$$= -0{,}138\log_2 0{,}138 - 0{,}086\log_2 0{,}086 - \ldots - 0{,}003\log_2 0{,}003 =$$

$$= 4{,}5356 \quad \textit{біт/символ}$$

Максимальна ентропія $H_{\max} = \log_2 34 = 5{,}0874$.

Надмірність повідомлень складе

$$D = \frac{H_{\max}(X) - H(X)}{H_{\max}(X)} = \frac{5{,}0874 - 4{,}5356}{5{,}0874} = 0{,}1085. \blacksquare$$

Усунення надмірності може досягатися використанням саме нерівномірних кодів, які враховують ймовірність появи літер у повідомленні. Літери, що мають велику ймовірність, кодуються більш короткими кодовими послідовностями, а більш довгі комбінації присвоюються рідкісними літерами.

Приклад 3.3. Розглянемо джерело з прикладу 3.1, що має алфавіт з чотирьох символів А, Б, В, Г з ймовірностями p(А)=0,5; p(Б)=0,25; p(В)=p(Г)=0,125. Для передачі каналом використовується нерівномірний код А→0, Б→10, В→110, Г→111. Визначити середню кількість двійкових символів, що припадають на один символ джерела. Порівняти з ентропією джерела.

Рішення. Зведемо вихідні дані у табл. 3.2.

Таблиця 3.2 – Характеристики нерівномірного коду

Символ x_i	x_1 =А	x_2=Б	x_3 =В	x_4 =Г
Ймовірність p_i	0,5	0,25	0,125	0,125
Код	0	10	110	111
Число біт n_i	1	2	3	3

Середня кількість двійкових символів, що припадають на один символ джерела:

$$n_{сер} = \sum_{i=1}^{4} p_i \cdot n_i = 0{,}5 \cdot 1 + 0{,}25 \cdot 2 + 0{,}125 \cdot 3 + 0{,}125 \cdot 3 = 1{,}75 \; біт .$$

Таким чином, середня кількість двійкових розрядів, що припадають на один символ, що передається в канал, дорівнює ентропії джерела $H(X)=n_{ср}=1{,}75$, тобто. для даного джерела нерівномірний код виявляється більш економічним, ніж рівномірний і оптимальним. При цьому коефіцієнт стиснення

$$\mu = \frac{\log_2 N}{n_{ср}} = \frac{\log_2 4}{1{,}75} = \frac{2}{1{,}75} = 1{,}1429 ,$$

тобто швидкість передачі по каналу зв'язку при використанні нерівномірного кодування може бути в 1,1429 разів більше ніж при рівномірному кодуванні. ■

Приклад 3.4. Символи українського алфавіту закодовані відповідно до табл. 3.3. (принципи побудови кодової таблиці будуть розглянуті нижче). Визначити середню кількість двійкових символів, що припадають на один символ джерела. Порівняти з ентропією джерела.

Таблиця 3.3 – Нерівномірне кодування букв українського алфавіту

Літера	Ймовірність p_i	Код	Число біт n_i	Літера	Ймовірність p_i	Код	Число біт n_i
-	0,138	101	3	З	0,020	110111	6
О	0,086	1111	4	Я	0,019	110100	6
Н	0,068	1001	4	Ь	0,016	011110	6
А	0,064	1000	4	Б	0,013	010100	6
И	0,055	0110	4	Г	0,013	001001	6
В	0,046	0011	4	Ч	0,011	001000	6
Т	0,045	0001	4	Х	0,011	1101101	7
І	0,044	0000	4	Ї	0,010	1101100	7
Р	0,043	11101	5	Ц	0,010	1101011	7
Е	0,042	11100	5	Й	0,009	0111111	7
С	0,037	11001	5	Ю	0,008	0111110	7
К	0,033	11000	5	Ж	0,007	0101011	7
М	0,029	01110	5	Ш	0,005	11010101	8
У	0,027	01001	5	Є	0,005	11010100	8
Д	0,027	01000	5	Щ	0,004	01010101	8
Л	0,027	01011	5	Ф	0,003	010101001	9
П	0,025	00101	5	Ґ	0,000	010101000	9

Рішення. Середня кількість двійкових символів, що припадають на один символ джерела

$$n_{сер} = \sum_{i=1}^{34} p_i \cdot n_i = 0{,}138 \cdot 3 + 0{,}086 \cdot 4 + ... + 0{,}002 \cdot 9 + 0{,}000 \cdot 9 = 4{,}5720.$$

Таким чином, середня кількість двійкових розрядів на один символ, що передається в канал, $n_{сер}$=4,572, що дещо більше, ніж визначена в прикладі 3.2. ентропії джерела $H(X)$=4,5356. Однак $n_{сер}$ менше максимальної ентропії $H_{max}=\log_2 34=5{,}0874$, тобто для зазначеного джерела нерівномірний код виявляється більш економічним, ніж рівномірний. При цьому коефіцієнт стиснення

$$\mu = \frac{\log_2 N}{n_{ср}} = \frac{\log_2 34}{4{,}572} = \frac{5{,}0875}{4{,}572} = 1{,}1128,$$

тобто швидкість передачі по каналу зв'язку при використанні нерівномірного кодування може бути в 1,1128 разів більше ніж при рівномірному кодуванні.

При передачі за допомогою нерівномірних кодів виникає проблема пошуку границь між суміжними кодовими комбінаціями. Нехай, наприклад, код складається зі слів 0, 10, 01 та 101. Тоді повідомлення 01010101 можна трактувати декількома способами.

0 101 0 101

01 01 01 01

0 10 10 101

Щоб уникнути проблеми неоднозначності кодування, код повинен конструюватися таким чином, щоб короткі кодові слова не були початковою частиною (префіксом) довших кодових слів. Такі коди, в яких жодне кодове слово не є початком іншого (умова Фано), називаються префіксними. Префіксні коди дозволяють однозначно декодувати послідовність повідомлень.

Наприклад, код, що складається зі слів 0, 10 та 11, є префіксним, і повідомлення 0110101101110 можна розбити на слова єдиним чином:

0 11 0 10 11 0 11 10. ■

Приклад 3.5. Визначити, які з наведених у таблиці 8.4 нерівномірних кодів є префіксними.

Таблиця 4.4 – Приклади нерівномірних кодів

Символ	x_1	x_2	x_3	x_4
Код 1	0	1	00	11
Код 2	0	10	110	111
Код 3	0	10	101	111

Рішення. Для коду 1 символ x_1=0 є префіксом коду x_3=00, для коду 3 символ x_2=10 є префіксом коду x_3=101. Для коду 3 таких проблем немає, тому він є префиксным.

Для декодування послідовності символів, закодованих префіксними кодами, може бути запропонований такий алгоритм:

1. Виділити у поточному повідомленні крайній лівий символ, приєднати праворуч до робочого кодового слова.

2. Порівняти робоче кодове слово із кодовою таблицею; якщо збігу немає, перейти до 1.

3. Декодувати робоче кодове слово, очистити його.

4. Перевірити, чи є ще символи в повідомленні, якщо так, перейти до 1. ■

Приклад 3.6. Декодувати повідомлення 101100111011010 закодоване кодом, з прикладу 8.3. А→0, Б→10, В→110, Г→111.

Рішення. Процес декодування зведемо у табл. 3.5.

Таблиця 3.5 – Процес декодування префіксного коду

Крок	Робоче слово	Повідомлення	Розпізнаний символ	Декодована послідовність
1		101100111011010		
2	1	01100111011010		
3	10	1100111011010	Б	Б
4	1	1100111011010		Б
5	11	100111011010		Б
6	110	00111011010	В	БВ
7	0	0111011010	А	БВА
8	1	111011010		БВА
9	11	11011010		БВА
10	111	1011010	Г	БВАГ
11	0	011010	А	БВАГА
12	1	11010		БВАГА
13	11	1010		БВАГА
14	110	010	В	БВАГАВ
15	1	10		БВАГАВ
16	0	0	Б	БВАГАВБ

Таким чином, забезпечилося однозначне розбиття послідовності

10 110 0 111 0 110 10 та її декодування БВАГАВБ.

Питання знаходження найекономічнішого коду, у якого середня кількість $n_{сер}$ двійкових символів на букву алфавіту було б мінімальним, можливий і відповідь на нього дає теорема Фано у наступному формулюванні. ■

Якщо дано джерело з обсягом алфавіту N та ентропією $H(X)$, то, застосовуючи код з основою m, можливо забезпечити середнє число двійкових символів на літеру алфавіту джерела, що лежить у межах

$$\frac{H(X)}{\log_2 N} \le n_{сер} < \frac{H(X)}{\log_2 N} + 1. \qquad (3.2)$$

Зокрема, при N=2 (3.2) набуває вигляду

$$H(X) \le n_{сер} < H(X) + 1. \qquad (3.3)$$

Іншими словами, середня кількість двійкових елементів коду, що відображає символи алфавіту джерела, принаймні дорівнює ентропії джерела і може перевищувати її на одиницю. Якщо ймовірність кодових слів $p(x_k)$ задовольняє умові

$$p(x_k) = N^{-n(x_k)}. \qquad (3.4)$$

можливо побудувати оптимальний код, у якого

$$n_{сер} = \frac{H(X)}{\log_2 N}. \qquad (3.5)$$

Так, оптимальним є код, розглянутий у прикладі 3.3. Справді,

$p_1 = 2^{-n_1} = 2^{-1} = 0{,}5;\ p_2 = 2^{-n_2} = 2^{-2} = 0{,}25;\ p_3 = p_4 = 2^{-n_3} = 2^{-3} = 0{,}125.$

Як показано у прикладі, для цього коду $H(X)$=$n_{сер}$. Якщо умова (3.4) не виконується, то оптимальний код побудувати неможливо.

3.2 Коди Шеннона-Фано та Гаффмана

Найбільш відомими представниками нерівномірних двійкових кодів є *код Шеннона-Фано та код Гаффмана.*

Згідно з методикою Шеннона-Фано побудова оптимального коду зводиться до наступного.

1. Символи алфавіту повідомлень розташовуються у порядку зменшення ймовірностей.

2. Алфавіт символів, що кодуються, розбивається на дві підгрупи, таким чином, щоб сумарні ймовірності символів обох груп були по можливості рівні. Якщо абсолютно рівної ймовірності в підгрупах не можна досягти, то їх ділять так, щоб у верхній частині (у верхній підгрупі) залишалися символи, сумарна ймовірність яких дещо більша від сумарної ймовірності символів у нижній частині (у нижній групі).

3. Першому розряду кодів першої групи надається значення 1, другій групі – значення 0.

4. Кожну з утворених груп знов ділять на дві частини таким чином, щоб сумарні ймовірності новостворених підгруп були по можливості рівні.

5. Першим групам кожної з підгруп знову надається значення 1, а другим – 0. Таким чином, отримуємо другі розряди коду. Потім кожна із чотирьох груп знову ділиться на рівні (з погляду сумарної ймовірності) частини тощо. Процес повторюється до тих пір, поки в кожній підгрупі не залишиться по одному символу.

Приклад 3.7. Нехай ймовірністі появи кожного з дев'яти повідомлень зведено в табл. 3.6. Побудувати код Шеннона-Фано, визначити максимальну ентропію $H_{\max}$, ентропію джерела $H(X)$, середнє число біт на символ $n_{\text{сер}}$, коефіцієнт стиснення μ.

Рішення. Побудову коду зведемо у табл. 3.6.

Таблиця 3.6 – Побудова коду Шеннона-Фано

Символ x_i	$p(x_i)$	Розбиття повідомлень на групи					Код	$n(x_i)$	$n(x_k)\cdot p(x_i)$
x_1	0,35	1	1				11	2	0,7
x_2	0,15	1	0				10	2	0,3
x_3	0,13	0	0	1			001	3	0,39
x_4	0,09	0	0	0			000	3	0,27
x_5	0,09	0	1	1	1		0111	4	0,36
x_6	0,08	0	1	1	0		0110	4	0,32
x_7	0,05	0	1	0	0		0100	4	0,2
x_8	0,04	0	1	0	1	1	01011	5	0,2
x_9	0,02	0	1	0	1	0	01010	5	0,1

На першому кроці множина X можливих повідомлень x_i розбивається на два підмножини. У першу підмножину включаються повідомлення x_1, x_2 з сумарною ймовірністю $p_{12}= p_1+p_2 = 0,35+0,15=0,50$, у другу – $\{x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$ з сумарною ймовірністю $p_{3456789}=p_3+p_4+p_5+p_6+p_7+p_8+p_8=0,13+0,09+0,09+0,08+0,05+0,04+0,02=0,50$. Сумарні ймовірності обох груп рівні, як перший розряд кодового слова першої підмножини вибирається одиниця, другої – нуль.

Потім перша підмножина $\{x_1, x_2\}$ знову розбивається на дві підмножини. У першу підмножину включається повідомлення x_1 з ймовірністю $p_1=0,35$, у другу x_2 з ймовірністю $p_2=0,15$. Оскільки ймовірність подій першої групи більша, ніж у другої, значенню другого розряду кодового слова першої підмножини $\{x_1\}$ присвоюється одиниця, другої підмножини $\{x_2\}$ – нуль.

Підмножина $\{x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$ у свою чергу розбивається на дві підмножини з приблизно рівною сумарною ймовірністю. У першу підмножину включається повідомлення $\{x_3, x_4\}$ із сумарною ймовірністю $p_{34}=0,13+0,09=0,22$, у друге повідомлення $\{x_5, x_6, x_7, x_8, x_9\}$ із сумарною ймовірністю $p_{56789}=0,09+0,08+0,05+0,04+0,02=0,28$. Оскільки $p_{34}<p_{56789}$, як другий розряд кодового слова першої підмножини вибирається нуль, другої підмножини – одиниця.

Процес, відбитий у табл. 3.6, повторюється до того часу, поки у кожній підгрупі залишиться по одному повідомленню. Таким чином, символи можуть бути закодовані наступним чином:

$x_1 \to 11$; $x_2 \to 10$; $x_3 \to 001$; $x_4 \to 000$; $x_5 \to 0111$;

$x_6 \to 0110$; $x_7 \to 0100$; $x_8 \to 01011$; $x_9 \to 01011$.

Звернемо увагу, що отриманий код є префіксним і послідовність символів на виході кодера може бути декодована однозначно за допомогою розглянутого в прикладі 3.6 алгоритму. Знайдений код близький до оптимального. Справді, ентропія повідомлень

$$H(X) = -\sum_{i=1}^{9} p(x_i)\log_2 p(x_i) = -(0,35\log_2 0,35 + 0,15\log_2 0,15 + 0,13\log_2 0,13 +$$

$$+ 0,09\log_2 0,09 + 0,09\log_2 0,09 + 0,08\log_2 0,08 + 0,05\log_2 0,05 +$$

$$+ 0,04\log_2 0,04 + 0,02\log_2 0,02) = 2,75 \quad \textit{біт / повідомл.}$$

Середня довжина кодового слова або середня кількість біт на символ

$$n_{ср} = \sum_{i=1}^{9} p(x_i)n(x_i) = 0{,}7 + 0{,}3 + 0{,}39 + 0{,}27 + 0{,}36 + 0{,}32 + 0{,}2 + 0{,}2 + 0.1 = 2{,}84 \text{ бит/сим.}$$

$H(X) \approx n_{ср}$, отже, отриманий код задовольняє умові Фано (3.3) і дуже близький до оптимального, проте не досягає лівої межі (3.3).

Максимальна ентропія $H_{max} = \log_2 N = \log_2 9 = 3{,}1699$.

Коефіцієнт стиснення $\mu = \dfrac{\log_2 N}{n_{ср}} = \dfrac{\log_2 9}{2{,}84} = \dfrac{3{,}1699}{2{,}84} = 1{,}1162.$ ■

Приклад 3.8. Нехай алфавіт джерела містить шість елементів: $\{x_1, x_2, x_3, x_4, x_5, x_6\}$, що з'являються із ймовірностями:

$p(x_1)=0{,}25$; $p(x_2)=0{,}25$; $p(x_3)=0{,}15$; $p(x_4)=0{,}13$; $p(x_5)=0{,}12$; $p(x_6)=0{,}1$.

Побудувати код Шеннона-Фано для цього джерела. Визначити максимальну ентропію H_{max} ентропію джерела $H(X)$, середнє число біт на символ $n_{сер}$ коефіцієнт стиснення μ.

Рішення. Побудову коду зведемо у табл. 3.7.

Таблиця 3.7 – Побудова коду Шеннона-Фано

Символ x_i	Ймовірність $p(x_i)$	Розбиття повідомлень на підгрупи			Код	Число біт $n(x_i)$
x_1	0,25	1	1		1 1	2
x_2	0,25	1	0		1 0	2
x_3	0,15	0	1	1	0 1 1	3
x_4	0,13	0	1	0	0 1 0	3
x_5	0,12	0	0	1	0 0 1	3
x_6	0,1	0	0	0	0 0 0	3

Розрахуємо ентропію джерела

$$H(X) = -(0{,}25\log_2 0{,}25 + 0{,}25\log_2 0{,}25 + 0{,}15\log_2 0{,}15 + 0{,}13\log_2 0{,}13 +$$
$$+ 0{,}12\log_2 0{,}12 + 0{,}1\log_2 0{,}1) = 2{,}4925 \quad \textit{біт / повідомл.}$$

Середня кількість символів на літеру

$$n_{сер} = \sum_{i=1}^{6} p(x_i)n(x_k) = 0{,}25 \cdot 2 + 0{,}25 \cdot 2 + 0{,}15 \cdot 3 + 0{,}13 \cdot 3 + 0{,}12 \cdot 3 + 0{,}1 \cdot 3 = 2{,}5$$

$H(X) \approx n_{сер}$. Максимальна ентропія $H_{max} = \log_2 N = \log_2 6 = 2{,}5850$.

Коефіцієнт стиснення $\mu = \dfrac{\log_2 N}{n_{сер}} = \dfrac{\log_2 6}{2{,}4925} = \dfrac{2{,}5850}{2{,}4925} = 1{,}0371.$ ■

Недоліком коду Шеннона-Фано є необхідність довільного розбиття ймовірностей на дві групи на кожному кроці алгоритму, що ускладнює його програмну реалізацію та знижує ефективність стиснення. Частково вільний від цих недоліків код Гаффмана – алгоритм оптимального префіксного кодування алфавіту з мінімальною надмірністю. Метод кодування був розроблений в 1952 аспірантом Массачусетського технологічного інституту Девідом Гаффманом при написанні ним курсової роботи (цікаво, що керівником Гаффмана був Р.Фано). В даний час код Гаффмана використовується в багатьох програмах стиснення даних – при кодуванні фото- та відеозображень (JPEG, MPEG), в популярних архіваторах (PKZIP, LZH), протоколах передачі даних HTTP (Deflate), MNP5 і MNP7 та інших.

Для отримання коду Гаффмана всі повідомлення виписуються в порядку зменшення ймовірностей. Дві найменші ймовірності поєднують в один допоміжний символ, якому приписують сумарну ймовірність. Ймовірності символів, що не брали участі в об'єднанні, і отримана сумарна ймовірність знову розташовуються в порядку зменшення ймовірностей у додатковому стовпці, а дві останні об'єднуються. Процес триває до тих пір, поки не отримаємо єдиний символ з ймовірністю рівній одиниці.

На основі таблиці будується кодове дерево. З точки, що відповідає ймовірності 1, направляються дві гілки причому гілці з більшою ймовірністю присвоюється символ 1, а з меншою 0 і так доти не дійдемо до ймовірності останнього символу. Саме таким чином було отримано коди букв українського алфавіту у прикладі 3.4.

Приклад 3.9. Нехай алфавіт джерела містить шість елементів: x_1, x_2, x_3, x_4, x_5, x_6, , що з'являються з ймовірностями:

$p(x_1)=0{,}25$; $p(x_2)=0{,}25$; $p(x_3)=0{,}15$; $p(x_4)=0{,}13$; $p(x_5)=0{,}12$; $p(x_6)=0{,}1$.

Побудувати код Гаффмана для цього джерела.

Рішення. Побудову коду зведемо у табл. 3.8.

Таблиця 3.8 – Побудова коду Гаффмана

Крок 1		2		3		4		5		
x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	
x_1	0,25	x_1	0,25	x_{34}	**0,28**	x_{256}	**0,47**	x_{256}	**0,53**	1
x_2	0,25	x_2	0,25	x_1	0,25	x_{34}	0,28	x_{134}	0,47	
x_3	0,15	x_{56}	**0,22**	x_2	0,25	x_1	0,25			
x_4	0,13	x_3	0,15	x_{56}	0,22					
x_5	0,12	x_4	0,13							
x_6	0,1									

На першому етапі поєднуються дві події x_5 та x_6 і утворюється допоміжний символ x_{56} із сумарною ймовірністю допоміжний символ x_{56} із сумарною ймовірністю p_{56} На другому кроці знову поєднуємо символи з мінімальними ймовірностями – x_3 та x_4. Так об'єднуємо символи, доки не отримуємо єдиний символ x_{123456} із сумарною ймовірністю, що дорівнює одиниці. На підставі отриманої таблиці збудуємо кодове дерево рис 3.1.

З точки, що відповідає ймовірності 1, спрямовуємо дві гілки, причому гілці з більшою ймовірністю привласнюємо символ 1, а з меншою 0. Таке послідовне розгалуження продовжуємо доти, доки не дійдемо до ймовірності кожного символу. Рухаючись по кодовому дереву зверху вниз, можна записати для кожного символу відповідну кодову комбінацію.

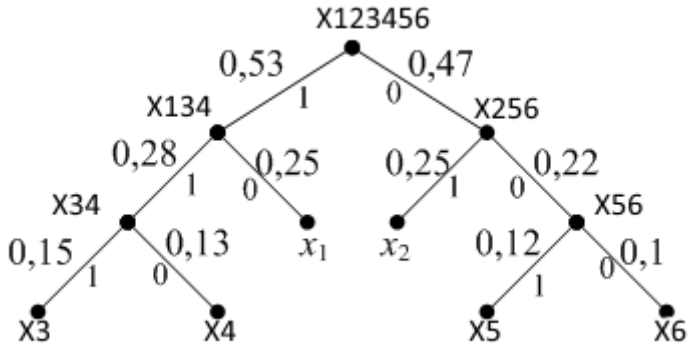


Рисунок 3.1 – Кодове дерево, побудоване методом Гаффмана

Кодові символи джерела повідомлень

x_1	x_2	x_3	x_4	x_5	x_6
10	01	111	110	001	000

Приклад 3.10. Нехай алфавіт джерела містить вісім елементів: А, Б, В, Р, Д, Е, Ж, З, що з'являються з ймовірностями: p(А)=0,16; p(Б)=0,11; p(В)=0,09; p(Г)=0,03; p(Ґ)=0,39; p(Д)=0,13; p(Е)=0,03; p(Є)=0,06. Побудувати код Шеннона-Фано і Гаффмана, визначити максимальну ентропію H_{max}, ентропію джерела $H(X)$, середнє число біт на символ $n_{сер}$, коефіцієнт стиснення μ. Порівняти ефективність кодів Шеннона-Фано та Гаффмана.

Рішення. Розташуємо літери алфавіту повідомлень у порядку зменшення ймовірностей і надамо їм для зручності позначення x_1, … x_8.

x_1 відповідає літері Ґ, x_2 ~А, x_3 ~Д, x_4 ~Б, x_5 ~В, x_6 ~ Є, x_7 ~Е, x_8~Г. Побудову коду Шеннона-Фано зведемо у табл. 3.9.

Таблиця 3.9 – Побудова коду Шеннона-Фано

x_i	$p(x_i)$	Розбиття повідомлень на підгрупи					Код	$n(x_i)$	$n(x_k)\cdot p(x_i)$
x_1	0,39	1	1				1 1	2	0,78
x_2	0,16	1	0				1 0	2	0,32
x_3	0,13	0	1	1			0 1 1	3	0,39
x_4	0,11	0	1	0			0 1 0	3	0,33
x_5	0,09	0	0	0			0 0 0	3	0,27
x_6	0,06	0	0	1	1		0 0 1 1	4	0,24
x_7	0,03	0	0	1	0	1	0 0 1 0 1	5	0,15
x_8	0,03	0	0	1	0	0	0 0 1 0 0	5	0,15

Ентропія повідомлень

$$H(X) = -\sum_{i=1}^{8} p(x_i)\log_2 p(x_i) = -(0{,}39\log_2 0{,}39 + 0{,}16\log_2 0{,}16 + 0{,}13\log_2 0{,}13 +$$

$$+ 0{,}11\log_2 0{,}11 + 0{,}09\log_2 0{,}09 + 0{,}06\log_2 0{,}06 + 0{,}03\log_2 0{,}03 + 0{,}03\log_2 0{,}03 =$$

$$= 2{,}5455 \quad \textit{біт / повідомл.}$$

Середня кількість біт на символ

$$n_{серш} = \sum_{i=1}^{8} p(x_i)n(x_i) = 0{,}78 + 0{,}32 + 0{,}39 + 0{,}33 + 0{,}27 + 0{,}15 + 0{,}15 = 2{,}63 \text{ біт/сим.}$$

Максимальна ентропія $H_{\max} = \log_2 N = \log_2 8 = 3$.

Коефіцієнт стиснення $\mu = \dfrac{\log_2 N}{n_{ср}} = \dfrac{\log_2 8}{2{,}63} = \dfrac{3,}{2{,}63} = 1{,}1407.$

Побудову коду Гаффмана зведемо в табл. 3.10.

Таблиця 3.10 – Побудова кода Гаффмана

Крок 1		2		3		4		5		6		7	
x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$	x_i	$p(x_i)$
x_1	0,39	x_1	0,39	x_1	0,39	x_1	0,39	x_1	0,39	x_1	0,39	$x_{2453678}$	**0,61**
x_2	0,16	x_2	0,16	x_2	0,16	x_{45}	**0,20**	x_{3678}	**0,25**	x_{245}	**0,36**	x_1	0,39
x_3	0,13	x_3	0,13	x_3	0,13	x_2	0,16	x_{45}	0,20	x_{3678}	0,25		
x_4	0,11	x_4	0,11	x_{678}	**0,12**	x_3	0,13	x_2	0,16				
x_5	0,09	x_5	0,09	x_4	0,11	x_{678}	0,12						
x_6	0,06	x_6	0,06	x_5	0,09								
x_7	0,03	x_{78}	**0,06**										
x_8	0,03												

На підставі отриманої таблиці збудуємо кодове дерево рис 3.2.

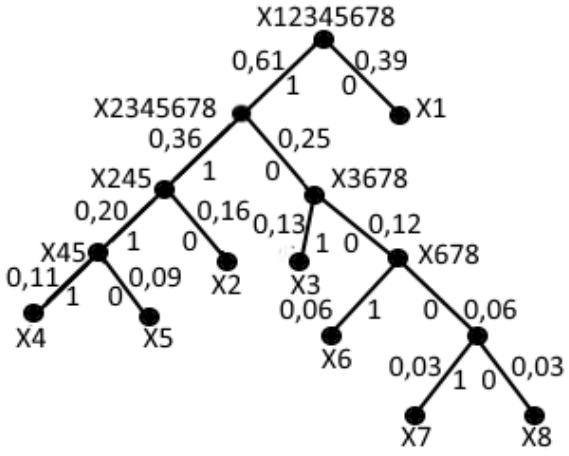


Рисунок 3.2 – Кодове дерево, побудоване методом Гаффмана

Кодові символи, одержані з табл. 3.10 та рис 3.2 мають такий вигляд

Літера	Ґ	А	Д	Б	В	Є	Е	Г
x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Код	0	110	101	1111	1110	1001	10001	10000

Для коду Гаффмана середня кількість біт на символ:

$$n_{серх} = \sum_{i=1}^{8} p(x_i)n(x_i) = 0{,}39 + 0{,}48 + 0{,}39 + 0{,}44 + 0{,}36 + 0{,}24 + 0{,}15 + 0{,}15 = 2{,}60 \text{ біт/сим.}$$

Коефіцієнт стиснення $\mu = \dfrac{\log_2 N}{n_{ср}} = \dfrac{\log_2 8}{2{,}60} = \dfrac{3}{2{,}60} = 1{,}1538.$

Аналіз отриманих результатів показує, що відоме з теорії співвідношення $H(X) \le n_{серх} \le n_{серш} \le H_{\max}$ виконується.

3.3 Кодування цілих чисел. Коди Левенштейна, Еліаса, Голомба, Райса, Фібоначчі, Івен-Роде

Недоліком алгоритмів стиснення Гаффмана та Шеннона-Фано є необхідність точного знання ймовірностей кожного з можливих повідомлень, що практично рідко можна досягти. Тому було розроблено ряд алгоритмів кодування чисел, які не потребують знання всіх ймовірностей. У всіх цих алгоритмах передбачається, що малі значення кодованих величин вірогідніші, ніж великі і кодуються більш короткими послідовностями.

Методи цієї групи є трансформуючими та потоковими, тобто можуть застосовуватися навіть у тому випадку, коли обсяг вхідних даних наперед не відомий. Коди є префіксними, тобто, послідовність кодів без додаткових розділювачів може однозначно декодуватися. Перевага кодів, що розглядаються, в тому, що кодування даними методами не вимагає великої кількості пам'яті, а також однаково швидко виконується як кодування, так і декодування інформації. Наведені коди зазвичай застосовуються для ефективного кодування джерел без пам'яті.

Унарний код

Унарне кодування – це ентропійне кодування, що представляє число *A* у вигляді *A* одиниць із замикаючим нулем (прямий унарний код) або у вигляді *A* нулів із замикаючою одиницею (інверсний код). Надалі

використовуватимемо позначення. α(*A*) – унарне представлення числа *A*, тобто *A* поспіль одиниць із роздільним нулем, β(*A*) – звичайне представлення числа *A* у двійковій системі числення.

Приклад 3.11. Закодувати унарним двійковим кодом число *A*=13.

Рішення. Записуємо двійкову послідовність з тринадцятьма поспіль одиницями, що закінчуються розділовим нулем: $A_{ун}$=11111111111110.

Можливий варіант із 13 поспіль нулів, що закінчуються контрольною одиницею: $A_{ун}$=00000000000001. ■

Приклад 3.12. Декодувати записане в унарному коді $A_{ун}$=11111111110.

Рішення. Підраховуємо число одиниць в унарному коді до нульового біта. *A*=10. ■

Приклад 3.13. Декодувати записану в унарному коді послідовність чисел $A_{ун}$=1110101111011111110110.

Рішення. Виключаємо розділові нулі та підраховуємо числа одиниць між ними, які відповідатимуть елементам закодованої послідовності.

A = 111 1 1111 1111111 11 = 3, 1, 4, 7, 2. ■

Гамма-код Левенштейна

Гамма код Левенштейна для числа *A* утворюється шляхом звернення (переписування справа наліво) послідовності бітів у двійковому записі цього числа та додавання перед кожним бітом, крім останнього, прапорного біта 0. Останнім прапорним бітом є біт 1, який збігається з найстаршим бітом у вихідному запису числа А.

Приклад 3.14. Закодувати двійковим гамма-кодом Левенштейна число *A*=53.

Рішення. Перетворимо число *A*=53 у двійковий код $A_{дв}$=110101. Записуємо біти числа у зворотному порядку: $A_{двзвор}$=101011. Додаємо перед кожним бітом, крім останнього нулі (підкреслені): $A_{лев}$=0100100011. ■

Приклад 3.15. Декодувати записане в коді Левенштейна число $A_{лев}$=010001010001011.

Рішення. Виключаємо з двійкового коду $A_{лев}$=010001010001011 нульові біти, що стоять на непарних позиціях: $A_{двзвор}$ = 10110111.

Записуємо біти числа у зворотному порядку: $A_{дв}$=11101101. Перекладаємо з двійкового коду до десяткового: *A* = 237. ■

Приклад 3.16. Декодувати записану в гамма-коді Левенштейна послідовність чисел $A_{лев}$ = 0100011000001011010000001101011.

Рішення. Виділяємо одиниці, що стоять на непарних позиціях, що маркують кінець символу (виділені жирним) і ділимо послідовність на ряд окремих кодів. $A_{лев}$ = 0100011 000001011 0100000011 01011.

Виключаємо з кожного з кодів нульові біти, що стоять на непарних позиціях: $A_{двзвор}$=1011 00111 100011 111.

Записуємо біти чисел у зворотному порядку:

$A_{дв}$ = 1101 11100 110001 111.

Перекладаємо з двійкових кодів до десяткових: A=13, 28, 49, 7. ■

Гамма-код Еліаса

Гамма-код Еліаса – двійковий префіксний код для представлення натуральних чисел. Число кодується у два етапи. На першому етапі число перетворюється на двійковий код, визначається кількість двійкових розрядів m і число $r = m-1$ кодується за допомогою інверсного унарного коду. На другому етапі до отриманого унарного коду праворуч у незмінному вигляді приписуються $m-1$ молодших розрядів вихідного числа (старший одиничний розряд, таким чином, опускається).

Іншими словами, код є числом у двійковому поданні, доповненим зліва нулями в кількості на одиницю менше розрядності кодованого числа. У сумі довжина коду повинна дорівнювати $2m - 1$. Таким чином, можна записати представлення гамма-коду Еліаса як $\gamma(A) = \alpha(A) : \beta(A-2^k)$.

Приклад 3.17. Закодувати двійковим гамма-кодом Еліаса число A=173.

Рішення. Перетворимо число 173 у двійковий код: $A_{дв}$=10101101. Підраховуємо число розрядів у коді: m=8, r=m–1=7.

Приписуємо до двійкового коду $A_{дв}$ ліворуч сім нулів:

$A_{гамел}$ = 000000010101101.

Очевидно, що отриманий код складається з інверсного коду унарного числа розрядів m=8 і двійкового подання числа 173 без старшої одиниці

$A_{гамел}$ =00000001 0101101. ■

Приклад 3.18. Декодувати записане в гамма-коді Еліаса число $A_{гамел}$=0000001101001.

Рішення. Підраховуємо число нулів у коді $A_{гамел}$=0000001101001, що стоять ліворуч перед старшою одиницею: r=6, кількість двійкових розрядів у декодованому числі $m=r+1=6+1=7$.

Виділяємо наступні сім розрядів, починаючи з першої зліва одиниці: $A_{дв}$=1101001.

Перекладаємо з двійкового коду до десяткового A=105. ■

Приклад 3.19. Декодувати записану в гамма-коді Еліаса послідовність чисел $A_{гамел}$ =000110100100000000110010.

Рішення. Підраховуємо в послідовності ліворуч нулі до першої значущої одиниці і отримуємо число r=3, $m=r+1=3+1=4$, після чого зчитуємо наступні m розрядів, виділяючи перше закодоване гамма-кодом Еліаса число. Повторюємо до кінця послідовності, розбиваючи цим послідовність на окремі коди. Після цього декодуємо кожен код.

r_1=3, $m_1 = r_1+1=3+1=4$, $A_{гамел1}$=0001101; $A_1=1101_2=13$;

r_2=2, $m_2 = r_2+1=2+1=3$, $A_{гамел2}$=00100; $A_2=100_2=4$;

r_1=5, $m_3 = r_3+1=5+1=6$, $A_{гамел3}$=00000110010; $A_3=110010_2=50$.

A =13, 4, 50. ■

Дельта-код Еліаса

Дельта-код Еліаса є похідним від гамма-коду. Дельта-коди Еліаса відрізняються від гамма-кодів тим, що в них унарна частина також закодована гамма-кодами. Таким чином, для отримання дельта-коду з гамма-коду Еліаса потрібно застосувати гамма-кодування до унарної частини: $\delta(A)=\gamma(\alpha(A)) : \beta(A-2^k)$. Спочатку з допомогою гамма-кодування записується кількість значних двійкових розрядів у числі, після чого кодуються всі значні розряди, крім старшої одиниці.

Приклад 3.20. Закодувати двійковим дельта-кодом Еліаса число A=612.

Рішення. Перетворимо число 612 у двійковий код: $A_{дв}$=1001100100. Підраховуємо кількість розрядів у коді: m=10. Кодуємо число m за допомогою гамма-коду Еліаса. Для цього перетворимо число m=10 у двійковий код: $m_{дв}$=1010.

Підраховуємо число розрядів у коді $m_{дв}$: q=4, $s=q-1=3$.

Приписуємо до двійкового коду mдв зліва три нулі: $m_{гамел}$=0001010. Приписуємо до отриманого коду праворуч код $A_{дв}$ без старшої одиниці: $A_{дел}$=0001010001100100. ■

Приклад 3.21. Декодувати записане в дельта-коді Еліаса число $A_{дел}$=00111010110.

Рішення. Підраховуємо число нулів у коді $A_{дел}$=00111010110, що стоять ліворуч перед старшою одиницею: s =2, q =s +1=2+1=3.

Виділяємо наступні три розряди, починаючи з першої зліва одиниці: $m_{дв}$=111.

Перетворюємо $m_{дв}$ із двійкового коду до десяткового: m=7, r=m–1=6. Зчитуємо наступні шість розрядів з коду $A_{дел}$=00111<u>010110</u> (підкреслені) і приписуємо до них ліворуч одиницю: $A_{дв}$=1010110.

Перекладаємо з двійкового коду до десяткового: A=86. ■

Приклад 3.22. Декодувати записану в дельта-коді Еліаса послідовність чисел $A_{дел}$=000100101010100011110001011101.

Рішення. Підраховуємо в послідовності зліва нулі до першої значущої одиниці та отримуємо число s =3, q = s +1=3 +1=4.

Виділяємо наступні чотири розряди, починаючи з першої зліва одиниці: $m_{дв}$=1001.

Перетворюємо $m_{дв}$ із двійкового коду до десяткового: m=9, r = 9–1=8. Зчитуємо наступні r=8 розрядів із послідовності $A_{дел}$=000100<u>10101010</u>0011110001011101 (підкреслені) і приписуємо до них ліворуч одиницю: $A_{дв}$ =101010100. Перетворюємо з двійкового коду до десяткового: A=340.

Повторюємо до кінця послідовності, розбиваючи цим послідовність на окремі коди. Після цього декодуємо кожен код.

s_1=3, q_1=s_1+1=3+1=4, $m_{дв1}$=1001, m_1=9, r_1=m_1–1=8, $A_{дел1}$=000100101010100; A_1=101010100$_2$=340;

s_2=1, q_2=s_2+1=1+1=2, $m_{дв2}$=11, m_2=3, r_2=m_2–1=2, $A_{дел2}$=01110; A_2=110$_2$=6;

s_3=2, q_3=s_3+1=2+1=3, $m_{дв3}$=101, m_3=5, r_3=m_3–1=4, $A_{дел3}$=001011101; A_3=11101$_2$=29.

A =340, 6, 29. ■

Код Голомба

Коди Голомба характеризуються цілим параметром m. Для кодування числа A його необхідно представити як A=q·m+r, тобто q = A div m, r= A mod m, де q та r – цілі позитивні числа, $0 \le r < m$, div – операція ділення націло, mod – залишок від цілісного ділення.

Потім r кодується унарним кодом, а q – двійковим. Отримані двійкові послідовності поєднуються у результуючому слові.

Таким чином, якщо k – найменше додатне таке, що $2^k \geq 2m$, $j = 2^{(k-1)} - m$, код Голомба для A із заданим параметром m, утворюється з'єднанням двох кодів: Golomb(A, m) = α(($A-j$) div m) : β(($A-j$) mod m+2j).

Код Голомба забезпечує мінімальне порівняно з іншими відомими кодами число біт на символ у випадку, якщо джерело інформації породжує цілі числа з ймовірностями $P(i)$=

Приклад 3.23. Закодувати кодом Голомба з параметром m=9 число A=68.

Рішення. Обчислюємо цілу частину від ділення

A/m: $q = A$ div m=68 div 9=7.

Обчислюємо залишок від ділення A/m: $r = A$ mod m=68 mod 9=5.

Перекладаємо залишок r двійковий код. При цьому виділяємо для запису k=m біт. В даному випадку k = 9 $r_{дв}$= 0101.

Записуємо цілу частину від ділення A/m в унарному коді: $q_{ун}$=11111110. З'єднуємо коди для цілої частини та залишку: $A_{гол}$=111111100101. ■

Приклад 3.24. Декодувати записане у коді Голомба з параметром m=7 число $A_{гол}$=11110011.

Рішення. Підраховуємо число одиниць у двійковому коді $A_{гол}$=11110011 до першого зліва нульового біта і визначаємо цілу частину від ділення: $q_{ун}$=11110, q=4. Визначимо число біт для запису залишку

$k = m$= 7=3

Зчитуємо наступні після одиниць та прапорового нульового біта три розряди з коду $A_{гол}$=11110<u>011</u> (підкреслені). Визначаємо $r_{дв}$=011. Перекладаємо залишок r у десятковий код: r=3. Визначаємо число $A=q\cdot m+r=4\cdot 7+3=31$. ■

Приклад 3.25. Декодувати записану в коді Голомба з параметром m=6 послідовність чисел $A_{гол}$ =11111100011111111110100110010.

Рішення. Підраховуємо число одиниць у двійковій послідовності $A_{гол}$=11111100011111111110100110010 до першого зліва нульового біта та визначаємо цілу частину від ділення для першого слова $q_{ун}$=1111110, q=6.

Визначимо число біт для запису $k = m$= 6=3.

Зчитуємо наступні після одиниць та прапорного нульового біта три розряди з послідовності $A_{гол}$=11111100111111111100100110010 (підкреслені). Визначаємо $r_{дв}$=001.

Перекладаємо залишок r у десятковий код: r=1.

Визначаємо число $A = q \cdot m + r = 6 \cdot 6 + 1 = 377$. Повторюємо до кінця послідовності, розбиваючи послідовність на окремі коди. Після цього декодуємо кожен код.

$q_{ун1}$= 1111110, q_1=6, $r_{дв1}$=001, r_1=1, $A_1 = q_1 \cdot m + r_1 = 6 \cdot 6 + 1 = 37$;

$q_{ун2}$= 111111110, q_2=8, $r_{дв2}$=100, r_2=4, $A_2 = q_2 \cdot m + r_2 = 8 \cdot 6 + 4 = 52$;

$q_{ун3}$= 110, q_3=2, $r_{дв3}$=010, r_3=2, $A_3 = q_3 \cdot m + r_3 = 2 \cdot 6 + 2 = 14$.

A =37, 52, 14. ■

Код Райса

Код Райса є окремим випадком коду Голомба, коли m є ступенем двійки $m=2^k$. Таким чином, як параметр коду вибирається $k=\log_2 m$. Процедура кодування та декодування у цьому випадку дещо спрощується.

Нехай для запису кодованого числа необхідно l біт. При кодуванні праві k біт результату кодування збігаються з правими k бітами двійкового представлення кодованого числа. $t=l-k$ біт, що залишилися, утворюють двійковий запис цілої частини від поділу, яка перетворюється на унарний код. Тоді код Райсу для числа A із заданим параметром k виглядатиме як Rice(A, k) = α(A div m): β(A mod m).

Приклад 3.26. Закодувати кодом Райса з параметром k=5 число A=211.

Рішення. Перетворимо число A=179 у двійковий код: $A_{дв}$=11010011. Визначаємо загальне число біт у двійковому коді l=8.

Зчитуємо праві k=5 біт (підкреслені) у коді $A_{дв}$= 11010011: $r_{дв}$=10011.

Зчитуємо ліві $t = l - k = 8 - 5 = 3$ біт (підкреслені) в коді $A_{дв}$=11010011.

Визначаємо цілу частину $q_{дв}$=110. Перетворимо на десятковий код q=6. Перетворимо число q=6 на унарний код: $q_{ун}$=1111110. З'єднуємо коди для цілої частини та залишку: $A_{райс}$=111111010011. ■

Приклад 3.27. Декодувати записане в коді Райса з параметром k=4 число $A_{райс}$=111111111100111.

Рішення. Визначаємо загальне число біт у числі, що декодується: A=15. Зчитуємо ліві $t = l-k = 15-4 = 11$ біт (підкреслені) у коді

$A_{\text{райс}}$=111111111100111. Визначаємо код цілої частини в унарному коді: $q_{\text{ун}}$=11111111110. Перетворимо унарний код цілої частини на десятковий код: $q_{\text{дес}}$=10, а потім у двійковий $q_{\text{дв}}$=1010.

Зчитуємо праві k=4 біт (підкреслені) у коді $A_{\text{райс}}$=111111111100111, $r_{\text{дв}}$=0111. З'єднуємо коди для цілої частини та залишку: $A_{\text{дв}}$=10100111.

Перекладаємо з двійкового коду до десяткового: A=167. ■

Приклад 3.28. Декодувати записану в коді Райса з параметром k=6 послідовність чисел $A_{\text{райс}}$ =11001101010110100111000 1010.

Рішення. Підраховуємо число одиниць у двійковій послідовності $A_{\text{райс}}$ =11001101010110100111000 1010 до першого зліва нульового біта і визначаємо цілу частину від ділення для першого слова $q_{\text{ун}}$=110. Перетворимо унарний код цілої частини на десятковий код $q_{\text{дес}}$=2, а потім у двійковий $q_{\text{дв}}$ =10. Зчитуємо наступні k=6 розрядів із послідовності $A_{\text{райс}}$=11001101011010100111000 1010 (підкреслені).

Визначаємо $r_{\text{дв}}$=011010.

З'єднуємо коди для цілої частини та залишку: $A_{\text{дв}}$=10011010. Перекладаємо з двійкового коду до десяткового: A=154.

Повторюємо до кінця послідовності, розбиваючи послідовність на окремі коди. Після цього декодуємо кожен код.

$q_{\text{ун1}}$= 110, q_1=2, $q_{\text{дв1}}$=10, $r_{\text{дв1}}$=011010, $A_{\text{дв1}}$=10011010, A_1 =154;

$q_{\text{ун2}}$= 10, q_2=1, $q_{\text{дв1}}$=1, $r_{\text{дв2}}$=110100, $A_{\text{дв2}}$=1110100, A_2 = 116;

$q_{\text{ун3}}$= 1110, q_3=3, $q_{\text{дв3}}$=11, $r_{\text{дв3}}$=001010, $A_{\text{дв3}}$=11001010, A_3 =202.

A =154, 116, 202. ■

Код Фібоначчі

В основу префіксного коду Фібоначчі покладено представлення чисел у Фібоначчієвій системі числення (приклади 1.3, 1.4). Таким чином, число розглядається як сума чисел Фібоначчі

$$A= a_m F_{A+1} + a_{m-1} F_m +\ldots+ a_i F_{i+1}+\ldots+ a_2 F_3 + a_1 F_2,$$

де F_0=0, F_1=1, F_i=F_{i-2}+F_{i-1}.

{ F_i } = {0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610…}

Оскільки при поданні чисел у Фібоначчієвій системі не може виявитися двох суміжних одиниць, то як ознака кінця кодового слова використовуються дві одиниці, що стоять поруч. Біти при цьому подаються у зворотному порядку (молодший розряд на початку кодового слова).

Приклад 3.29. Закодувати кодом Фібоначчі число A=308.

Рішення. Перекладемо число у Фібоначчієву систему числення згідно з алгоритмом, наведеним у розділі 1.

Знаходимо максимальне число Фібоначчі F_i, яке не перевищує число A= 308 і зафіксуємо його зменшений на одиницю номер k.

$F_i = F_{13} = 233$, $A = 13 - 1 = 12$. Надамо $a_{12}=1$, $a_1 \ldots a_{11}=0$.

Запишемо перше наближення Фібоначчієвого представлення $A_{\text{фіб1}}$=100000000000.

Обчислимо різницю $A = A - F = 308 - 233 = 75$.

Знайдемо наступне максимальне число Фібоначчі F_i,, яке не перевищує число A, і зафіксуємо його зменшений на одиницю номер k.

$F = F_{10} = 55$, $k = 10 - 1 = 9$.

Надамо a_9=1. Друге наближення Фібоначчієвого представлення $A_{\text{фіб2}}$=100100000000.

Обчислимо різницю $A = A - F = 75 - 55 = 20$.

Знайдемо максимальне число Фібоначчі F_i,, яке не перевищує число A, і зафіксуємо його зменшений на одиницю номер k. $F_i = F_7 = 13$, k=7– 1 = 6. Надамо a_6=1. Третє наближення Фібоначчієвого представлення $A_{\text{фіб3}}$=100100100000.

Обчислимо різницю $A = A - F = 20 - 13 = 7$.

Знайдемо максимальне число Фібоначчі F_i, яке не перевищує число A, і зафіксуємо його зменшений на одиницю номер k. $F_i = F_5 = 5$, k=5–1=4. Надамо a_4=1. Четверте наближення Фібоначчієва представлення $A_{\text{фіб4}}$=100100101000.

Обчислимо різницю $A = A - F = 7 - 5 = 2$.

Знайдемо максимальне число Фібоначчі F_i, яке не перевищує число A, і зафіксуємо його зменшений на одиницю номер k. $F_i = F_3 = 2$, k=3–1=2. Надамо a_2=1. П'яте наближення Фібоначчієвого представлення $A_{\text{фіб5}}$=100100101010.

Обчислимо різницю $A = A - F = 2 - 2 = 0$.

Оскільки отримане значення А дорівнює нулю, переходимо до кінця алгоритму і записуємо відповідь: $A_{\text{фіб}}$=100100101010. Записуємо біти числа у зворотному порядку:

$A_{\text{фібзвор}}$=010101001001.

Додаємо наприкінці прапорний одиничний біт: $A_{\text{фібпреф}}$=0101010010011. ■

Приклад 3.30. Декодувати записане у префіксному коді Фібоначчі число $A_{\text{фібпреф}}$=10101000010011.

Рішення. Виключаємо наприкінці прапорний одиничний біт: $A_{\text{фібзвор}}$=1010100001001.

Записуємо біти числа у зворотному порядку: $A_{\text{фіб}}$=1001000010101. Представляємо число A як суму чисел Фібоначчи

A=377+89+8+3+1=478. ■

Приклад 3.31. Декодувати записану у префіксному коді Фібоначчі послідовність чисел $A_{\text{фібпреф}}$=11001011001011100010110010000011.

Рішення. Визначаємо місця у послідовності, де стоять дві одиниці поспіль і розглядаємо їх як місце стиків двох послідовних кодів. Ділимо послідовність на окремі коди.

$A_{\text{фібпреф}}$=11001011 001011 10001011 001000011.

Декодуємо кожен код окремо:

$A_{\text{фібпреф1}}$=1001011, $A_{\text{фібзвор1}}$=100101, $A_{\text{фіб1}}$=101001, A_1=13+3+1=17.

$A_{\text{фібпреф2}}$=001011, $A_{\text{фібзвор2}}$=00101, $A_{\text{фіб2}}$=10100, A_2=8+3=11.

$A_{\text{фібпреф3}}$=10001011, $A_{\text{фібзвор3}}$=1000101, $A_{\text{фіб3}}$=1010001, A_3=21+8+1=30.

$A_{\text{фібпреф4}}$=001000011, $A_{\text{фібзвор4}}$=00100001, $A_{\text{фіб4}}$=10000100, A_4=34+3=37. ■

Код Івен-Роде

Ці коди складаються з послідовності груп довжиною L_1, L_2, L_3, …, L_m біт. Довжина першої групи дорівнює трьом бітам, кожної наступної (i+1)-ої групи визначається значенням числа бітів попередньої i-ої групи. Значення бітів останньої групи є підсумковим значенням всього коду, тобто послідовності груп. Наприкінці послідовності слідує нульовий біт.

При кодуванні формується спочатку остання група, наступна безпосередньо перед нулем, а потім по черзі відновлюються всі попередні групи одна за одною, доки процес буде завершено.

Таким чином, алгоритм представлення числа в коді Івен-Роде зводиться до наступного.

1. Перевести число A, що перетворюється, в двійковий код. Присвоїти величині N значення числа, що перетворюється.

2. Присвоїти коду $A_{\text{ір}}$ значення нуль.

3. Приписати до коду ліворуч двійкове значення N.

4. Якщо N<8, зберегти значення коду та вийти.

5. Присвоїти N число біт його двійкового представлення.

6. Перейти до 3.

Приклад 3.32. Закодувати кодом Івен-Роде число A =611.

Рішення. Перекладаємо A в двійкову систему A=1001100011_2.

Привласнюємо значення N =A = 1001100011_2. Надаємо коду A_{ip} нульове значення A_{ip}=0.

Приписуємо зліва до коду A_{ip} двійкове значення N.

A_{ip} = 1001100011 0.

Надаємо N число біт його двійкового представлення N=10_{10} = 1010_2.

Приписуємо зліва до коду A_{ip} двійкове значення N.

A_{ip}=1010 1001100011 0.

Надаємо N число біт його двійкового представлення N=4_{10} = 100_2.

Приписуємо зліва до коду A_{ip} двійкове значення N.

A_{ip} = 100 1010 1001100011 0.

Оскільки N<8, виходимо з алгоритму.

Остаточне значення коду A_{ip} = 100101010011000110. ■

При декодуванні, навпаки, групи утворюються, починаючи з першої, одна за одною, тобто за значенням бітів вже знайденої групи визначається довжина наступної і так далі, доки не буде знайдена підсумкова група, після якої йде нуль. Алгоритм декодування коду Івен-Роде, таким чином, має вигляд.

1. Отримати три ліві біти закодованого числа A_{ip} і присвоїти це значення двійковій змінній N.

2. Якщо перший прочитаний біт дорівнює нулю, присвоїти шуканому A значення N і вийти з алгоритму.

3. Якщо перший прочитаний біт дорівнює одиниці, отримати наступний біт.

4. Якщо черговий прочитаний біт дорівнює нулю, надати A значення N і вийти з алгоритму.

5. Якщо черговий прочитаний біт дорівнює одиниці, отримати N біт надати N отримані значення і перейти до кроку 4.

Приклад 3.33. Декодувати записане в коді Івен-Роде число A_{ip}=1001000111001100.

Рішення. Зчитуємо три ліві біти закодованого числа A_{ip} 100 і надаємо це значення змінній N=100_2=4_{10}. Оскільки перший з отриманих біт дорівнює одиниці, зчитуємо наступний, четвертий по порядку, біт з числа A_{ip}=100<u>1</u>000111001100.

Оскільки аналізований біт дорівнює одиниці, зчитуємо N=4 біт і надаємо N отримані значення $N=1000_2=8_{10}$. Оскільки перший з отриманих біт дорівнює одиниці, зчитуємо наступний, восьмий по порядку, біт з числа A_{ip}=1001000111001100.

Оскільки аналізований біт дорівнює одиниці, зчитуємо N=8 біт і надаємо N отримані значення $N=11100110_2=230_{10}$.

Оскільки перший з отриманих біт дорівнює одиниці, зчитуємо наступний, шістнадцятий по порядку, біт з числа A_{ip}= 1001000111001100.

Оскільки аналізований біт дорівнює нулю, надаємо A значення $A=N=11100110_2=230_{10}$ і виходимо з алгоритму. ■

3.4 Характеристики систем стиснення даних з втратами та без втрат

При зберіганні інформації та передачі каналами зв'язку неминуче виникає питання скорочення обсягу даних. Тому було розроблено значну кількість алгоритмів та програм для щільного пакування (стиснення) інформації – архіватори, графічні та музичні редактори, програми кодування сигналів та зображень. *Стиснення* (*упаковка, компресія, кодування джерела*) – це перетворення даних, що виробляється з метою зменшення об'єму, який ці дані займають. Зворотна процедура називається *відновленням даних (розпакуванням, декомпресією*).

Основним параметром алгоритмів стиснення даних є *коефіцієнт стиснення* (*ступінь стиснення*) – відношення обсягу вихідних даних до обсягу стиснених даних

$$K_c = S_0/S_c, \tag{3.6}$$

де K_c – коефіцієнт стиснення, S_0 – обсяг вихідних даних, S_c – обсяг стиснутих.

Таким чином, чим вищий коефіцієнт стиснення, тим алгоритм ефективніший. Коефіцієнт стиснення може бути як постійним (деякі алгоритми стиснення звуку, зображення, наприклад, *A*-закон, μ-закон, усічене блочне кодування), так і змінним. У другому випадку він може бути визначений для кожного конкретного повідомлення, або оцінюватись як середній за деяким тестовим набором даних, максимальний або

мінімальний коефіцієнт стиснення. Іншими важливими характеристиками алгоритмів стиснення є час виконання програм упаковки та розпакування даних, необхідний об’єм пам'яті для зберігання програми та даних, якість стиснення - величина, що показує, наскільки сильно упакований вихідний потік, за допомогою застосування до нього повторного стиснення цього ж або іншого алгоритму.

Методи стиснення даних поділяються на два основні класи: *стиснення без втрат* та *стиснення із втратами*. При використанні стиснення без втрат можливе повне відновлення вихідних даних, стиснення з втратами дозволяє відновити дані зі спотвореннями, зазвичай несуттєвими з точки зору подальшого використання відновлених даних. Стиснення без втрат (неспотворююче стиснення) зазвичай використовується для передачі та зберігання текстових даних, комп'ютерних програм, рідше для скорочення об'єму аудіо- та відео даних та інших аналогових сигналів у випадках, коли спотворення небажані, наприклад, дані, одержувані з медичної вимірювальної апаратури, датчиків військової та космічної техніки.

Стиснення із втратами (спотворююче стиснення) має значно більший, ніж стиснення без втрат, ступень стиснення і зазвичай застосовується для скорочення обсягу аудіо- та відеоданих та цифрових фотографій у тих випадках, коли важливий ступінь стиснення, а повна відповідність вихідних та відновлених даних не потрібна. При цьому втрати при стисненні ведуть до несуттєвих спотворень зображення (звуку) які не перешкоджають нормальному сприйняттю, але при звірянні оригіналу та відновленої після стиснення копії можуть бути помічені.

3.5 Алгоритми стиснення без втрат

Існує велика кількість алгоритмів стиснення без втрат, які можна умовно поділити на дві групи:

1. Алгоритми статистичного (ентропійного) стискування. Ця група алгоритмів стискає інформацію, використовуючи нерівномірність частот, із якими різні символи зустрічаються у повідомленні. До алгоритмів цієї групи відносяться алгоритми арифметичного та префіксного кодування, наприклад, розглянуті вище методи Шеннона-Фано та Гаффмана, січних.

2. Поточні та словникові алгоритми. До цієї групи належать алгоритми сімейств RLE (run-length encoding), LZ* та інші. Особливістю всіх алгоритмів цієї групи є те, що при кодуванні використовується не інформація про частоти символів у повідомленні, а інформація про послідовності, що зустрічалися раніше.

В окрему групу можна назвати алгоритми перетворення інформації. Такі алгоритми, наприклад, перетворення Берроуза-Вілера, не виконують безпосереднього стиснення інформації, але їх застосування значно спрощує подальше стиснення з використанням потокових, словникових та ентропійних алгоритмів. Розглянемо найпопулярніші словникові алгоритми стиснення без втрат.

Алгоритм RLE

Кодування довжин серій (англ. *run-length encoding*, *RLE*) або кодування повторів – алгоритм стиснення даних, що замінює послідовності символів, що повторюються (серії) на один символ і число його повторів.

Алгоритм кодування методом RLE має такий вигляд.

1. Прочитати перший символ із вхідного потоку.

2. Додати символ у вихідний потік.

3. Порахувати число повторень символу та додати це число у вихідний потік.

4. Кінець потоку? Якщо так, то перехід до 6.

5. Прочитати наступний символ і перейти до 2.

6. Кінець алгоритму.

Приклад 3.34. Стиснути за допомогою алгоритму RLE послідовність {HHHRRTVVV555GGHHHHBBBBBBFFA33FFF}. Визначити коефіцієнт компресії.

Рішення. Відповідно до алгоритму замінюємо послідовності символів, що повторюються, парами символів, що містять сам символ і число його повторів. Стиснута послідовність матиме вигляд {H 3, R 2, T 1, V 3, 5 3, G 2, H 4, B 6, F 2, A 1, 3 2, F 3}.

Вихідне повідомлення містило 32 символи, закодоване – 24 символи.

Згідно з (3.6) коефіцієнт стиснення дорівнюватиме

$$K_c = S_0/S_c = 32/24 = 1{,}3333.$$

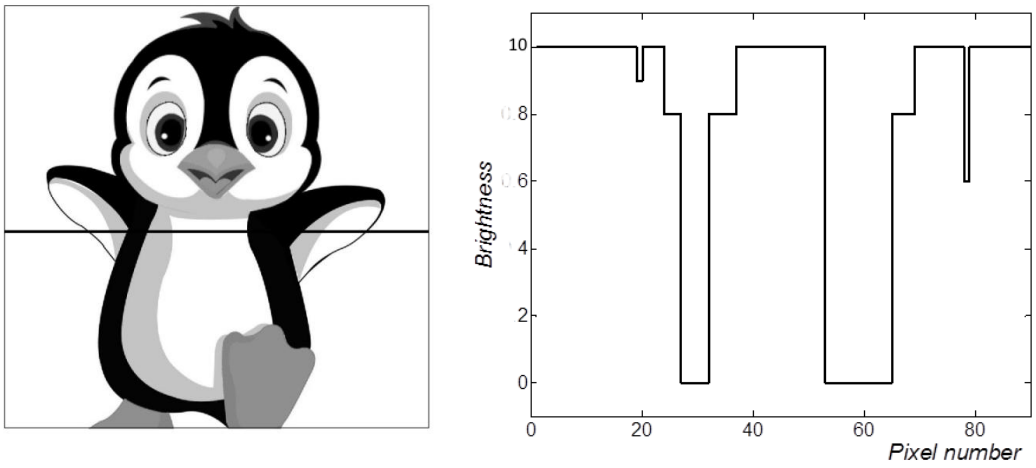
Декодування повідомлень проводиться тривіально – кожна пара символів замінюється ланцюжком першого символу в парі довжиною, що відповідає значенню другого символу. ■

Приклад 3.35. Розпакувати отриману в результаті стиснення методом RLE послідовність {J 2, Q 7, 1 4, B 3, L 6, 8 2}.

Рішення. Відповідно до алгоритму замінюємо пари символів послідовностями перших символів у парі тривалістю, що дорівнює значенню другого символу. Розпакована послідовність матиме вигляд {JJQQQQQQQ1111BBBLLLLLL88}.■

Алгоритм RLE розрахований на зображення з великими областями кольору, що повторюється – ділову графіку, іконки, графічні малюнки. Для зображень із плавним переходом тонів, таких як фотографії, метод RLE застосовується рідко. Звукові дані можуть бути стиснуті за допомогою RLE після того, як до них буде застосовано дельта-кодування.

Приклад 3.36. Стиснути за допомогою RLE алгоритму послідовність, що відповідає виділеному рядку зображення рис.3.3, *а*. Послідовність відносних рівнів яскравостей у рядку має вигляд { 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 9 10 10 10 10 8 8 0 0 0 0 0 8 8 8 8 8 1 0 0 0 10 10 10 0 0 0 0 0 0 0 0 0 0 0 8 8 8 8 10 10 10 10 10 10 10 10 10 6 10 10 10 10 10 10 10 10 1 0} (рис 3.3, *б*). Рівні яскравості закодовані числами від нуля до десяти, чорному кольору відповідає рівень 0, білому рівень 10. Визначити коефіцієнт компресії.



а *б*

Рисунок 3.3 – Зображення, що підлягає стисненню методом RLE,
а – зображення, що кодується, *б* – графік яскравості виділеного рядка

Рішення. Відповідно до алгоритму замінюємо серії символів парами, що містять символ та кількість його повторів. Стиснута послідовність матиме вигляд { 10 18, 9 1, 10 4, 8 3, 0 5, 8 5, 10 16, 0 12, 8 4, 10 10, 6 1, 10 12 }.

Вихідне повідомлення містило 91 символ, закодоване – 24 символи.

Згідно з (3.6) коефіцієнт стиснення дорівнюватиме

$K_c = S_0/S_c = 91/24 = 3{,}7917$. ■

Основним недоліком цього алгоритму є його вкрай низька ефективність на послідовності символів, що не повторюються. Для вирішення проблеми символів, що не повторюються, існують різні методи. Найпростішим методом є така: модифікація: байт, що кодує кількість повторів, повинен зберігати інформацію не тільки про кількість повторів, а й їх наявність.

У деяких випадках алфавіт, в якому записані довжини серій, ділять на дві частини – позитивні та негативні числа. Позитивні числа використовують для запису кількості повторів одного символу, а негативні - для запису кількості неоднакових символів, що йдуть один за одним.

Алгоритм LZW

Група словникових алгоритмів, на відміну від алгоритмів групи RLE, кодує не кількість повторів символів, а послідовності символів, що зустрічалися раніше. Під час роботи аналізованих алгоритмів динамічно створюється таблиця зі списком послідовностей, що вже зустрічалися, і відповідних їм кодів. Цю таблицю часто називають словником, а відповідну групу алгоритмів називають словниковими.

Найбільш популярним словниковим алгоритмом є алгоритм Лемпеля-Зіва-Велча (*Lempel-Ziv-Welch, LZW*) – алгоритм стиснення даних без втрат, створений А. Лемпелем, Я. Зівом та Т. Велчем. Він був опублікований Велчем в 1984 році як поліпшена реалізація алгоритму LZ78, запропонованого Лемпелем і Зівом в 1978 році. Перевагами алгоритму є високий ступінь стиснення текстових та графічних даних, простота апаратної та програмної реалізації. В даний час алгоритм стиснення LZW використовується у файлах формату TIFF, PDF, GIF, PostScript та інших, а також самостійно або у поєднанні з іншими алгоритмами у багатьох популярних програмах стиснення даних (ZIP, ARJ, LHA).

Процес стиснення методом LZW виглядає так: послідовно зчитуються символи вхідного потоку, і відбувається перевірка, чи існує у створеній таблиці рядків поточний рядок. Якщо такий рядок існує, зчитується наступний символ, а якщо рядок не існує, у вихідний потік заноситься код попереднього знайденого рядка, рядок заноситься в таблицю, а пошук починається знову.

Алгоритм кодування методом LZW має такий вигляд.

1. Усі можливі символи заносяться до словника.

2. Величині P надається значення першого символу із вхідного потоку.

3. Кінець потоку? Якщо так, то перехід до 8.

4. Величині C присвоюється значення наступного символу з вхідного потоку.

5. Перевірка, чи є рядок P+C у словнику. Якщо так, то P=P+C.

6. Якщо ні, P передається у вихідний потік, P+C в словник, P=C.

7. Перехід до 4.

8. P у вихідний потік.

9. Кінець алгоритму.

Слід зазначити, що по мірі зростання словника, розрядність коду, необхідного для збереження нового елемента повинна збільшуватись, так, поява у словнику восьмого елемента вимагає чотирьох біт для збереження нових слів, для шістнадцятого слова – п'яти біт, для 32-го – шести і т.д. .

Приклад 3.37. Стиснути за допомогою LZW алгоритму послідовність АБАБАГАБАМАГА. Визначити коефіцієнт компресії.

Рішення.

Крок 1. Заносимо до словника всі символи, що містяться в повідомленні, і присвоюємо їм двійкові коди згідно з табл. 3.11

Таблиця 3.11. – Символи, що містяться в повідомленні АБАБАГАБАМАГА

Символ	Код	Бітовий код
0: А	0	000
1: Б	1	001
2: Г	2	010
3: М	3	011

Крок 2: Читаємо з потоку символи А та Б .

Крок 3. Перевіряємо, чи є рядок АБ у словнику. Такого рядка у словнику поки що немає. Додаємо у словник рядок 4: АБ, у вихідний потік передаємо код 000, що відповідає символу «А».

Крок 4. БА у словнику немає. У словник 5: БА, в потік 001 «Б».

Крок 5. АБ є у словнику. АБА – ні. У словник 6: АБА, в потік 100 «АБ».

Крок 6. АГ у словнику немає. У словник 7: АГ, в потік 010 «Г».

Крок 7. ГА у словнику немає. У словник 8: ГА. З цього кроку починаємо використовувати для запису в словник і передачі в потік чотирибітні коди, в потік 0000 «А».

Крок 8. АБ є у словнику, АБА – є, АБАМ – ні. У словник 9: АБАМ, в потік 0110 «АБА».

Крок 9. МА у словнику немає. У словник 10: МА, в потік 0011 «М».

Крок 10. АГ є у словнику. АДА – ні. У словник 11: АДА, в потік 0111 «АГ».

Крок 11. І, нарешті, останній символ, за ним йде кінець повідомлення, тому ми просто виводимо в потік 0000 «А».

Всі етапи кодування послідовності зведено в табл. 3.12.

Таблиця 3.12. – Кодування повідомлення АБАБАГАБАМАГА

Поточний рядок	Поточний символ	Наступний символ	Потік		Словник
			Код	Біти	
АБ	А	Б	0	000	4: АБ 100
БА	Б	А	1	001	5: БА 101
АБ	А	Б			
АБА	Б	А	4	100	6 АБА 110
АГ	А	Г	0	000	7 АГ 111

Продовження таблиці 3.12

З наступного символу починаємо використовувати чотирибітні коди					
ГА	Г	А	2	0010	8: ГА 1000
АБ	А	Б			
АБА	Б	А			
АБАМ	А	М	6	0110	9: АБАМ 1001
МА	М	А	3	0011	10: МА 1010
АГ	А	Г			
АГА	Г	А	7	0111	11: АГА 1011
А	А		0	0000	

Отже, ми отримуємо закодоване повідомлення {0 1 4 0 2 6 3 7 0} та його бітовий еквівалент {000 001 100 000 0010 0110 0011 0111 0000}.

Кожен символ вихідного повідомлення був закодований групою із трьох біт, повідомлення містило 13 символів. Таким чином, довжина повідомлення становила 3·13=39 біт. Закодоване повідомлення спочатку кодувалося трибітними групами, а з появою у словнику восьмого слова – чотирибітними. Отже, довжина повідомлення становить 4·3+5·4=12+20=32 біта, що на сім біт коротше вихідного. Згідно з (3.1) коефіцієнт стиснення дорівнюватиме

$K_c = S_0/S_c = 39/32 = 1{,}2188$. ■

Приклад 3.38. Стиснути за допомогою LZW алгоритму послідовність PETERPIPERPICKEDPECKPICKLEDPEPPERS (англійська скоромовка «*Peter Piper picked peck of peckled peppers, Пітер Пайпер зібрав пучок маринованих перців*» з виключеними пробілами і службовими словами). Визначити коефіцієнт компресії.

Рішення. Зведемо рішення у табл. 3.13, 3.14, аналогічні 3.11, 3.12.

Таблиця 3.13. – Символи, що містяться в повідомленні PETERPIPERPICKEDPECKPICKLEDPEPPERS

Символ	Код	Бітовий код
0: P	0	0000
1: E	1	0001
2: T	2	0010
3: R	3	0011
4: I	4	0100
5: C	5	0101
6: K	6	0110
7: D	7	0111
8: L	8	1000
9: S	9	1001

Таблиця 3.14. – Кодування повідомлення ETERPIPERPICKEDPECKPICKLEDPEPPERS

Поточний рядок	Поточний символ	Наступний символ	Потік		Словник
			Код	Біти	
PE	P	E	0	0000	10: PE 1010
ET	E	T	1	0001	11: ET 1011
TE	T	E	2	0010	12: TE 1100
ER	E	R	1	0001	13: ER 1101
RP	R	P	3	0011	14: RP 1110
PI	P	I	0	0000	15: PI 1111
З наступного символу починаємо використовувати п’ятибітні коди					
IP	I	P	4	00100	16: IP 10000
PE	P	E			
PER	E	R	10	01010	17: PER 10001
RP	R	P			
RPI	P	I	14	01110	18: RPI 10010
IC	I	C	4	00100	19: IC 10011
CK	C	K	5	00101	20: CK 10100
KE	K	E	6	00110	21: KE 10101
ED	E	D	1	00001	22: ED 10110
DP	D	P	7	00111	23: DP 10111
PE	P	E			
PEC	E	C	10	01010	24: PEC 11000
CK	C	K			
CKP	K	P	20	10100	25: CKP 11001
PI	P	I			
PIC	I	C	15	01111	26: PIC 11010
CK	C	K			
CKL	K	L	20	10100	27: CKL 11001
LE	L	E	8	01000	28: LE 11100
ED	E	D			
EDP	D	P	22	10110	29: EDP 11001
PE	P	E			
PEP	E	P	10	01010	30: PEP 11110
PP	P	P	0	00000	31: PP 11111
З наступного символу починаємо використовувати шостибітні коди					
PE	P	E			
PER	E	R			
PERS	R	S	17	010001	32: PERS 10001
S	S		9	001001	

У канал буде передано повідомлення {0 1 2 1 3 0 4 10 14 4 5 6 1 7 10 20 15 20 8 22 10 0 17 9} та його бітовий еквівалент{0000 0001 0010 0001 0011 0000 00100 01010 01110 00100 00101 00110 00001 00111 01010 10100 01111 10100 01000 10110 01010 00000 010001 001001}.

Кожен символ вихідного повідомлення був закодований групою із чотирьох біт, повідомлення містило 34 символи, отже, довжина повідомлення становила 4·34=136 біт. Стиснуте повідомлення кодувалося чотири-, п'яти- та шестибітними комбінаціями таким чином, що довжина повідомлення склала 6·4+16·5 + 2·6 = 24 + 80 + 12 =116 біт, що на двадцять біт коротше вихідного.

Згідно з (3.1) коефіцієнт стиснення дорівнюватиме

$K_c = S_0/S_c = 136/116 = 1{,}1724$. ■

Алгоритм розпакування (декомпресії) стисненого файлу побудований таким чином, що немає необхідності зберігати та передавати словник, який може бути відновлений з потоку кодів. При розпакуванні спочатку визначається лише вихідний словник, а наступні записи словника ми можемо реконструювати вже на ходу, оскільки вони є просто конкатенацією (злиттям) попередньої послідовності та поточного символу.

Декодер LZW спочатку зчитує індекс (ціле число), шукає цей індекс у словнику та виводить підрядок, пов'язаний з цим індексом. Перший символ цього рядка конкатенується з поточним робочим рядком. Ця нова конкатенація додається до словника (подібно до того, як підрядки були додані під час стиснення). Потім декодований рядок стає поточним робочим рядком (поточний індекс, тобто підрядок, запам'ятовується), і процес повторюється. Таким чином, у процесі кодування та декодування коди до словника додаються під час обробки одного і того ж символу, тобто це відбувається синхронно.

Приклад 3.39. Розпакувати отриману в результаті стиснення у прикладі 10.4 послідовність {0 1 4 0 2 6 3 7 0}. Вихідний словник наведено у табл. 3.11.

Рішення. Процес декодування зведений у табл. 3.15

Таблиця 3.15 – Декодування повідомлення 0 1 4 0 2 6 3 7 0

Дані		Вихід	Словник
Біти	Код		
000	0	А	
001	1	Б	4: АБ
100	4	АБ	5: БА
000	0	А	6: АБА
1000	2	Г	7: АГ
0110	6	АБА	8: ГА
0011	3	М	9: АБАМ
0111	7	АГ	10: МА
0000	0	А	4: А

В результаті декодування отримано послідовність {АБ АБ А Г АБА М АГ А}, яка збігається з вихідною. ■

Перетворення Берроуза-Вілера

Перетворення Берроуза-Вілера (англ. *Burrows-Wheeler transform, BWT*) – алгоритм, що використовується для попередньої обробки даних перед стисненням, розроблений для покращення ефективності подальшого кодування. Алгоритм було запропоновано М. Берроузом та Д. Вілером у 1994 році.

Перетворення Берроуза-Вілера змінює порядок символів у послідовності таким чином, що повторювані підрядки утворюють на виході послідовності однакових символів, що йдуть поспіль. Само по себе BWT не стискає інформацію, але впорядкування даних може підвищити ефективність таких методів кодування, як RLE і алгоритм Гаффмана.

Алгоритм перетворення даних методом BWT має такий вигляд.

1. Складається матриця всіх циклічних зсувів вхідного рядка.

2. Рядки отриманої матриці перестановок сортуються в лексикографічному порядку.

3. Як вихідна послідовність вибирається останній стовпець таблиці перетворення та номер рядка, що збігається з вихідною послідовністю.

Приклад 3.40. Перетворити за допомогою BWT алгоритму послідовність {ПЕРЕТВОРЕННЯ}.

Рішення. Обчислюємо всі можливі циклічні зсуви вихідної послідовності та записуємо їх у вигляді рядків матриці. Першим рядком матриці буде вихідна послідовність, другим рядком – вона ж, зрушена циклічно на один символ вліво і т.д.

Упорядковуємо всі рядки матриці відповідно до лексикографічного (алфавітного) порядку символів, спочатку за першим символом, потім рядки, у яких перші символи рівні – за другим і т. д. Запишемо впорядковану матрицю в правий стовпець табл. 3.6.

Таблиця 3.16 – BWT перетворення повідомлення ПЕРЕТВОРЕННЯ

Номер рядка	Матриця зсувів	Упорядкована матриця
1	ПЕРЕТВОРЕННЯ	ВОРЕННЯПЕРЕ*Т*
2	ЕРЕТВОРЕННЯП	ЕННЯПЕРЕТВО*Р*
3	РЕТВОРЕННЯПЕ	ЕРЕТВОРЕННЯ*П*
4	ЕТВОРЕННЯПЕР	ЕТВОРЕННЯПЕ*Р*
5	ТВОРЕННЯПЕРЕ	ННЯПЕРЕТВОР*Е*
6	ВОРЕННЯПЕРЕТ	НЯПЕРЕТВОРЕ*Н*
7	ОРЕННЯПЕРЕТВ	ОРЕННЯПЕРЕТ*В*
8	РЕННЯПЕРЕТВО	**ПЕРЕТВОРЕННЯ**
9	ЕННЯПЕРЕТВОР	РЕННЯПЕРЕТВ*О*
10	ННЯПЕРЕТВОРЕ	РЕТВОРЕННЯП*Е*
11	НЯПЕРЕТВОРЕН	ТВОРЕННЯПЕР*Е*
12	ЯПЕРЕТВОРЕНН	ЯПЕРЕТВОРЕН*Н*

У вихідний потік передається останній стовпець упорядкованої матриці (виділено курсивом) та номер вихідного рядка (виділено жирним) серед відсортованих. Y = {ТРПРЕНВЯОЕЕН, 8}

Зворотне перетворення Барроуза-Віллера виконується в такий спосіб. Вписуємо в майбутню таблицю впорядкованих зрушень відомий нам останній стовпець. У цьому стовпці знаходяться всі символи, що містяться в послідовності, тому, сортуючи їх, ми отримуємо перший стовпець. Враховуючи, що рядки отримані за допомогою циклічного зсуву вихідного рядка, останній та перший символи у кожному рядку утворюють усі можливі початкові пари. Складемо всі пари символів у рядку та відсортувавши їх, отримуємо перший та другий стовпець.

Продовжуючи таким чином відновлюємо впорядковану матрицю зрушень. Виділивши з матриці рядок із заданим номером, відновлюємо вихідну послідовність. ■

Приклад 3.41. Відновити перетворену за допомогою алгоритму BWT послідовність *Y* = {РББАНАА, 5}.

Рішення. Починаємо формувати матрицю впорядкованих зсувів, внісши до неї останній стовпець РББАНАА (Крок 1 у табл. 3.17). Відсортувати всі символи останнього стовпця. Оскільки в кожному зі стовпців містяться всі символи послідовності, отримуємо перший стовпець (Крок 2).

Символи з останнього та першого стовпця утворюють пари РА, БА, БА, АБ, НБ, АН, АР. Відсортувавши ці пари, отримуємо два перші стовпці матриці впорядкованих сдвгов (Крок 3). Символи останнього стовпця разом із відсортованими парами становлять трійки РАБ, БАН, БАР, АБА, НБА, АНБ, АРА. Відсортувавши трійки, отримуємо три стовпці матриці (Крок 4).

Аналогічно відновлюється вся матриця (Кроки 5-7). Відповідно до заданого за умови номера 5 рядка матриці відновлюємо вихідну послідовність БАРАБАН.

Таблиця 3.17 – Відновлення повідомлення, перетвореного методом BWT

Номер рядка	Крок1	Крок 2		Крок 3		Крок 4	
		До сортування	Після	До	Після	До	Після
1	……Р	Р.…Р	А.…Р	РА.…Р	АБ.…Р	РАБ…Р	АБА…Р
2	……Б	Б.…Б	А.…Б	БА.…Б	АН.…Б	БАН…Б	АНБ…Б
3	……Б	Б.…Б	А.…Б	БА.…Б	АР.…Б	БАР…Б	АРА…Б
4	……А	А.…А	Б.…А	АБ.…А	БА.…А	АБА…А	БАН…А
5	……Н	Н.…Н	Б.…Н	НБ.…Н	БА.…Н	НБА…Н	БАР…Н
6	……А	А.…А	Н.…А	АН.…А	НБ.…А	АНБ...А	НБА...А
7	……А	А.…А	Р.…А	АР.…А	РА.…А	АРА…А	РАБ…А

Продовження табл. 3.17.

Номер строки	Крок 5		Крок 6		Крок 7	Упорядкована матриця
	До	Після	До	Після		
1	РАБА..Р	АБАН..Р	РАБАН.Р	АБАНБ.Р	РАБАНБР	АБАНБАР
2	БАНБ..Б	АНБА..Б	БАНБА.Б	АНБАР.Б	БАНБАРБ	АНБАРАБ
3	БАРА..Б	АРАБ..Б	БАРАБ.Б	АРАБА.Б	БАРАБАБ	АРАБАНБ
4	АБАН..А	БАНБ..А	АБАНБ.А	БАНБА.А	АБАНБАА	БАНБАРА
5	НБАР..Н	БАРА..Н	НБАРА.Н	БАРАБ.Н	НБАРАБН	**БАРАБАН**
6	АНБА..А	НБАР..А	АНБАР.А	НБАРА.А	АНБАРАА	НБАРАБА
7	АРАБ..А	РАБА..А	АРАБА.А	РАБАН.А	АРАБАНА	РАБАНБА

3.6 Алгоритми стиснення із втратами

Стиснення даних із втратами – метод, при використанні якого розпаковані дані відрізняються від вихідних, але ступінь відмінності не суттєва з точки зору їхнього подальшого використання. Цей тип компресії зазвичай застосовується для стиснення статичних зображень, аудіо- та відео та інших оцифрованих сигналів.

Методи стиснення з втратами допускають високий рівень стиснення, при цьому спотворення виявляються у допустимих межах чутливості людських органів. Оскільки багато методів стиснення з втратами ґрунтуються на фізичних особливостях органів чуття людини, то стислий сигнал або зображення практично не відрізняються для людини «на слух» і «на око». Так, урахування психоакустичної моделі сприйняття звуку дозволяє стискати мовні та інші акустичні сигнали в десятки разів практично без втрати якості.

До алгоритмів стиснення з втратами пред'являється ряд вимог: високий ступінь стиснення, незначні спотворення, що вносяться, як кількісні, так і суб'єктивно сприйняті, висока швидкість компресії та декомпресії, стійкість до помилок при передачі даних. При стисненні зображень додатково пред'являються вимоги редагування зображень, масштабованості, можливості показати «загрублені» зображення. Очевидно, що ці вимоги найчастіше взаємовиключні, тому розробка ефективних методів стиснення сигналів і зображень інтенсивно ведеться по теперішній час.

Існують дві основні групи методів стиснення із втратами.

У системах трансформації зображення або аудіо зазвичай перетворюються в новий базисний простір і здійснюється квантування відліків. Результат потім стискається ентропійними методами. Характерними представниками цієї групи є алгоритм JPEG та методи на основі вейвлет-перетворення.

У кодеках, що передбачають, попередні відліки сигналу використовуються для того, щоб передбачити поточний відлік. Помилка між прогнозованими даними та реальними разом з додатковою інформацією, необхідною для передбачення, потім квантується та кодується. До таких методів відноситься більшість алгоритмів кодування мови на основі лінійного передбачення, наприклад, G.728 та MELP.

Алгоритм JPEG

Алгоритм розроблений групою експертів у галузі фотографії (Joint Photographic Expert Group) для стиснення 24-бітних та напівтонових зображень у 1991 році та в різних модифікаціях широко застосовується по теперішній час. Алгоритм поєднує простоту реалізації та високий ступінь стиснення з прийнятною якістю відновлених зображень. На основі алгоритму розроблено один із популярних графічних форматів (розширення .jpg, .jpeg), що застосовується для зберігання фотографій та подібних до них зображень.

Алгоритм JPEG можна умовно поділити кілька етапів.

На першому етапі здійснюється дискретизація кольорового зображення. Дані пікселів перетворюються з колірного простору RGB (англ. Red, Green, Blue – червоний, зелений, синій) у колірний простір YCbCr (Y – компонента яскравості, Cb і Cr – синя і червона кольороворізностні компоненти)

$$\begin{aligned} Y &= 0{,}299 \cdot R + 0{,}578 \cdot G + 0{,}114 \cdot B; \\ Cb &= -0{,}1678 \cdot R - 0{,}3313 \cdot G + 0{,}5 \cdot B + 128; \\ Cr &= 0{,}5 \cdot R - 0{,}4187 \cdot G - 0{,}0813 \cdot B + 128. \end{aligned} \qquad (3.7)$$

Після перетворення для каналів зображення Cb і Cr, відповідальних за колір, може виконуватися субдискретизація (проріджування), яка полягає в тому, що кожному блоку з 4 пікселів (2×2) каналу Y яскравості ставляться у відповідність усереднені значення Cb і Cr (схема

проріджування) «4:2:0»). Таким чином, для кожного блоку 2×2 замість дванадцяти значень (по чотири Y, Cb і Cr) використовується всього шість (чотири Y і по одному усередненому Cb і Cr).

На другому етапі зображення розбивається на блоки 8×8 пікселів, над якими виконується *двовимірне дискретне косинусне перетворення* (*ДКП*). Математично доведено, що ДКП декорелює відліки сигналу чи зображення, тобто зменшує ступінь статистичного зв'язку між близько розташованими відліками. Це дозволяє збільшити обсяг інформації, що міститься у кожному відліку, і тим самим стискати сигнал або зображення.

Дискретне косинусне перетворення послідовності $\{x_i\}$, i=0…N–1 визначається як

$$X_k = \begin{cases} \dfrac{1}{\sqrt{N}} \sum\limits_{i=0}^{N-1} x_i \ \text{ при k} = 0, \\ \sqrt{\dfrac{2}{N}} \sum\limits_{i=0}^{N-1} x_i \cos \dfrac{\pi \cdot k \cdot (2i+1)}{2N} \ \text{ при k} > 0. \end{cases}, \ k = 0 \ldots N-1. \quad (3.8)$$

У матричній формі ДКП має вигляд

$$\overline{X} = C \cdot \overline{x}, \quad (3.9)$$

де $\overline{x}$ – вектор відліків сигналу; $\overline{X}$ – вектор коефіцієнтів ДКП; С-матриця перетворення, в якій елемент, що стоїть на перетині k-го рядка та i-го стовпця, визначається як

$$C_{k,i} = \begin{cases} \dfrac{1}{\sqrt{N}} \ \text{ при k} = 0, \\ \sqrt{\dfrac{2}{N}} \cos \dfrac{\pi \cdot k \cdot (2i+1)}{2N} \ \text{при k} > 0. \end{cases} \quad (3.10)$$

Приклад 3.42. Побудувати матрицю ДКП для N=5. Обчислити ДКП послідовності $\{x_i\} = \{x_0, x_1, x_2, x_3, x_4\} = \{1, -3, 2, -1, 4\}$.

Рішення. Визначаємо матрицю ДКП відповідно до (3.5). Наприклад, при $N = 5$ елемент $C_{2,3}$ (нумерація рядків і стовпців починається з нуля) дорівнює:

$$C_{2,3} = \sqrt{\frac{2}{5}} \cdot \cos\frac{\pi \cdot 2 \cdot 7}{10} = \sqrt{\frac{2}{5}} \cdot \cos\frac{14\pi}{10} = 0{,}6325 \cdot (-0{,}3090) = -0{,}1954.$$

Аналогічно обчислюємо інші елементи матриці та отримуємо

$$C^{(5)} = \begin{bmatrix} 0{,}4472 & 0{,}4472 & 0{,}4472 & 0{,}4472 & 0{,}4472 \\ 0{,}6015 & 0{,}3717 & 0 & -0{,}3717 & -0{,}6015 \\ 0{,}5117 & -0{,}1954 & -0{,}6325 & -0{,}1954 & 0{,}5117 \\ 0{,}3717 & -0{,}6015 & 0 & 0{,}6015 & -0{,}3717 \\ 0{,}1954 & -0{,}5117 & 0{,}6325 & -0{,}5117 & 0{,}1954 \end{bmatrix}.$$

Відповідно до (3.3), (3.4) обчислюємо коефіцієнти перетворення

X_0=0,4472·1+ 0,4472·(–3) + 0,4472·2 +0,4472·(–1) + 0,4472·4 = 1,3416;

X_1 = 0,6015·1+0,3717·(–3) + 0·2 + (–0,3717)·(–1) +(–0,6015)·4 = –2,5479;

X_2 = 0,5117·1 + (–0,1954)·(–3) + (–0,6325)·2 + (–0,1954)·(–1) + 0,5117·4=2,0752;

X_3 = 0,3717·1+(–0,6015)·(–3) +0·2 +0,6015·(–1) +(–0,3717)·4= 0,0878;

X_4 = 0,1954·1+(–0,5117)·(–3) + 0,6325·2 +(–0,5117)·(–1) +0,1954·4 = 4,2888.

Таким чином, послідовність коефіцієнтів ДКП має вигляд

$\{X_k\}$ = {1,3416; –2,5479; 2,0752; 0,0878; 4,2888}.■

Двовимірне дискретне косинусне перетворення від матриці $\{x_{ij}\}$ розмірності $N \times M$ визначається як

$$X_{kl} = A_i \cdot A_k \cdot \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} x_{ij} \cdot \cos\frac{\pi \cdot k \cdot (2i+1)}{2N} \cdot \cos\frac{\pi \cdot l \cdot (2j+1)}{2M}, \quad (3.11)$$

де $A_i = \begin{cases} \frac{1}{\sqrt{N}} & \text{при i} = 0, \\ \sqrt{\frac{2}{N}} & \text{при i} > 0 \end{cases}$.

Більш зручно представляти (3.6) у матричному вигляді

$$X = C \cdot x \cdot C^T, \quad (3.12)$$

де x – матриця відліків зображення; C – матриця ДКП, що визначається згідно (3.5); C^T – транспонована матриця ДКП (транспонування – процес

заміни у матриці рядків на стовпці); X – матриця коефіцієнтів двовимірного ДКП.

Приклад 3.43. Побудувати матрицю ДКП для N=4. Обчислити двовимірне ДКП послідовності $x_{ij} = \begin{bmatrix} 3 & 4 & -1 & 2 \\ 3 & 0 & 1 & 2 \\ 2 & -1 & -2 & 4 \\ 1 & -3 & 1 & 0 \end{bmatrix}$.

Рішення. Визначаємо матрицю ДКП 4×4 відповідно до (3.11) та транспоновану матрицю C^T.

$$C^{(4)} = \begin{bmatrix} 0{,}5 & 0{,}5 & 0{,}5 & 0{,}5 \\ 0{,}6533 & 0{,}2706 & -0{,}2706 & -0{,}6533 \\ 0{,}5 & =0{,}5 & -0{,}5 & 0{,}5 \\ 0{,}3706 & -0{,}6533 & 0{,}6533 & -0{,}2706 \end{bmatrix}; C^T = \begin{bmatrix} 0{,}5 & 0{,}6533 & 0{,}5 & 0{,}3706 \\ 0{,}5 & 0{,}2706 & -0{,}2706 & -0{,}6533 \\ 0{,}5 & -0{,}2706 & -0{,}5 & 0{,}6533 \\ 0{,}5 & -0{,}6533 & 0{,}5 & -0{,}2706 \end{bmatrix}.$$

Визначаємо матрицю коефіцієнтів двовимірного ДКП згідно (3.12). Спочатку обчислюємо допоміжну матрицю $R = x \cdot C^T$. Наприклад,

$R_{3,2} = 0{,}6533\cdot2 + 0{,}2706\cdot(-1) + (-0{,}2706\cdot(-2) + (-0{,}6533)\cdot4 = -1{,}0360.$

Обчисливши інші елементи (рекомендується зробити самостійно), отримуємо допоміжну матрицю R.

$$R = \begin{bmatrix} 4 & 2{,}0063 & 1 & -2{,}9958 \\ 3 & 0{,}3827 & 2 & 0{,}9239 \\ 1{,}5 & -1{,}036 & 4{,}5 & -1{,}1945 \\ -0{,}5 & -0{,}4291 & 1{,}5 & 2{,}8837 \end{bmatrix}.$$

Домножуємо отриману матрицю на матрицю ДКП (рекомендується зробити самостійно) і остаточно отримуємо матрицю коефіцієнтів двовимірного ДКП.

$$X = C \cdot R = \begin{bmatrix} 4 & 0{,}4619 & 4{,}5 & -0{,}1913 \\ 3{,}3457 & 1{,}9749 & -1{,}0031 & -3{,}2678 \\ -0{,}5 & 1{,}1152 & -2 & 0{,}0793 \\ 0{,}2378 & -0{,}2678 & 1{,}4979 & -2{,}9749 \end{bmatrix}. \blacksquare$$

Оскільки блоки, на які розбивається зображення, мають розмір 88 пікселів, для обчислення ДКП використовується матриця відповідного порядку

$$C^{(8)} = \begin{bmatrix} 0{,}354 & 0{,}354 & 0{,}354 & 0{,}354 & 0{,}354 & 0{,}354 & 0{,}354 & 0{,}354 \\ 0{,}490 & 0{,}416 & 0{,}278 & 0{,}098 & -0{,}098 & -0{,}278 & -0{,}416 & -0{,}490 \\ 0{,}462 & 0{,}191 & -0{,}191 & -0{,}462 & -0{,}462 & -0{,}191 & 0{,}191 & 0{,}462 \\ 0{,}416 & -0{,}098 & -0{,}490 & -0{,}278 & 0{,}278 & 0{,}490 & 0{,}098 & -0{,}416 \\ 0{,}354 & -0{,}354 & -0{,}354 & 0{,}354 & 0{,}354 & -0{,}354 & -0{,}354 & 0{,}354 \\ 0{,}278 & -0{,}490 & 0{,}098 & 0{,}416 & -0{,}416 & -0{,}098 & 0{,}490 & -0{,}278 \\ 0{,}191 & -0{,}462 & 0{,}462 & -0{,}191 & -0{,}191 & 0{,}462 & -0{,}462 & 0{,}191 \\ 0{,}098 & -0{,}278 & 0{,}416 & -0{,}490 & 0{,}490 & -0{,}416 & 0{,}278 & -0{,}098 \end{bmatrix} \quad (3.13)$$

Слід зазначити, що для прискорення перетворень (3.9), (3.12) можуть застосовуватися алгоритми з так званого *швидкого дискретного косинусного перетворення*, дозволяють скоротити обсяг обчислень.

В результаті ДКП отримуємо матрицю *X*, в якій коефіцієнти в лівому верхньому куті відповідають низькочастотній складовій зображення, а в нижньому правому – високочастотній.

На третьому етапі відбувається *квантування коефіцієнтів ДКП*, при котрому відкидаються коефіцієнти дискретного косинусного перетворення, які несуттєві для відновлення зображення. Квантування – основний процес, під час виконання якого губляться дані у методі JPEG-стискания.

Квантування здійснюється шляхом поелементного ділення матриці коефіцієнтів ДКП *X* на матрицю квантування Q

$$X_q = \left[\frac{X}{Q} \right], \quad (3.14)$$

де Q – матриця квантування, X_q – матриця квантованих коефіцієнтів,
[ ] – позначення округлення до найближчого цілого.

Для кожної компоненти Y, Cb, Cr можуть задаватися свої таблиці квантування таким чином, що компоненти кольору квантуються більш грубо, ніж компоненти яскравості. Існує два підходи для побудови матриць

квантування. Перший складається у тому, що матриці обчислюються динамічно під час стиснення. Зазвичай застосовується формула

$$Q_{ij} = 1 + (i+j)\cdot r,$$

де r – коефіцієнт, що визначає ступінь стиснення і задається користувачем.

Другий підхід полягає у використанні двох рекомендованих таблиць квантування – для яскравості Q_Y та для кольоровості Q_C,, включених до стандарту JPEG на основі експериментальних досліджень.

$$Q_Y = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}, Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}. \quad (3.15)$$

Після квантування матриця 8×8 перетворюється на 64-елементний вектор за допомогою «зигзаг»-сканування (рис. 3.4).

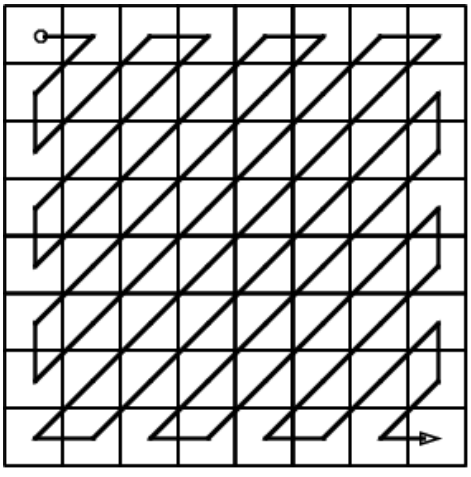


Рисунок 3.4 – «Зигзаг»-сканування

В результаті на початку вектора, як правило, записуватимуться ненульові коефіцієнти, а в кінці утворюватимуться ланцюжки з нулів.

На четвертому етапі відбувається ентропійне квантування коефіцієнтів ДКП. Спочатку ланцюжки однакових символів, зазвичай – розташованих наприкінці вектора нулів, виключаються з допомогою кодування модифікованим методом RLE. Послідовність коефіцієнтів перетворюється на набір кодів, формат яких представлений у табл. 3.18.

Таблиця 3.18 – Формат вихідних кодів JPEG

Символ 1		Символ 2
Лічильник нулів	Довжина	Амплітуда

У поле «лічильник нулів» заноситься кількість нулів, що передують поточному ненульовому символу, у поле «довжина» – число двійкових розрядів, необхідних для кодування чергового символу, у полі «амплітуда» – сам символ. Для полів «лічильник нулів» і «довжина» відводиться фіксоване число біт, як правило, по чотири. Після останнього ненульового коефіцієнта використовується пара (0,0). Якщо вона зустрінеться при декодуванні, значення, що залишилися, дорівнюють нулю.

Символ зазвичай кодується методом Гаффмана з фіксованою таблицею (у стандарті передбачена також можливість арифметичного кодування). Кодування здійснюється відповідно до табл. 3.19.

Таблиця 3.19 – Структура ентропійних кодів

Довжина	Значення		Коди	
	<0	>0	<0	>0
0	0			
1	–1	1	0	1
2	–3, –2	2, 3	00, 01	10, 11
3	–7…–4	4…7	000, 001, 010, 011	100, 101, 110, 111
4	–15…–8	8…15	0000, …, 0001	1000, …, 1111
5	–31…–16	16…31	00000, …, 00001	10000, …, 11111
6	–63…–32	32…63	000000, …00001	100000, …111111
7	–127…–64	64…127	0000000, …	1000000, …
9	–255…–128	128…255	00000000, …	10000000, …

Слід зазначити, що коефіцієнт ДКП $X_{1,1}$ (верхній лівий у матриці), що визначає середню яскравість кодованого фрагмента, зазвичай кодується окремо. При цьому враховуються значення яскравості суміжних фрагментів, обчислюється передбачене значення яскравості та кодується різниця між поточним та передбаченим значеннями.

Приклад 3.44. Стиснути методом JPEG фрагмент зображення (рис. 3.5). Визначити коефіцієнт стиснення.

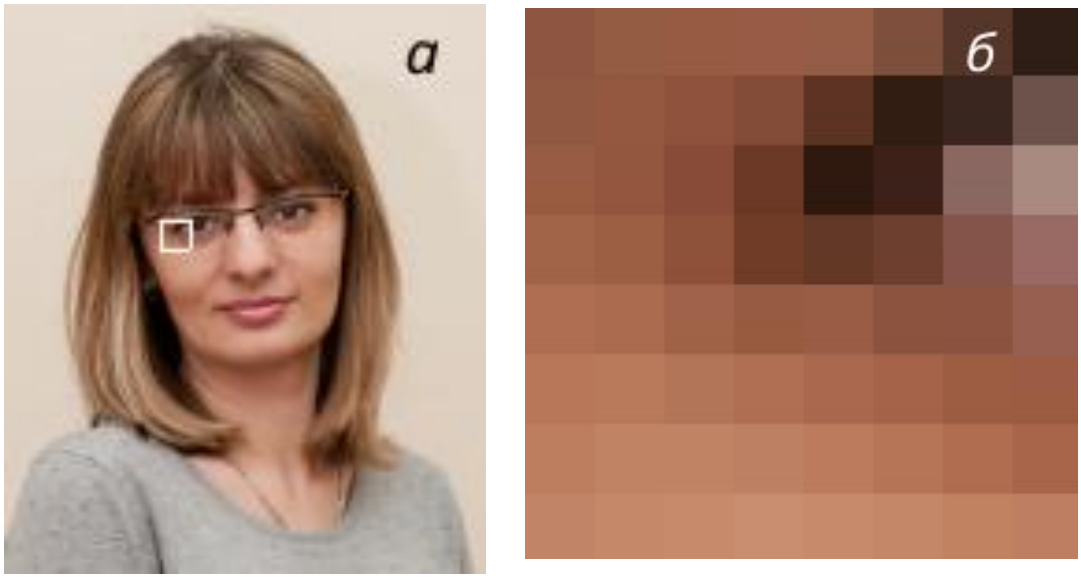


Рисунок 3.5 – Зображення, що стискається методом JPEG зображення (*а* – вихідне зображення, *б* – виділений фрагмент).

Рішення. Виділимо фрагмент зображення та перетворимо його. Для простоти розглядатимемо лише складову яскравості *Y*.

Рівні яскравості фрагмента Y 8×8, колірного простору RGB у простір YCbCr згідно (3.2) закодовані в діапазоні 0…255 утворюють наступну матрицю.

$$x = \begin{bmatrix} 102 & 107 & 107 & 108 & 107 & 94 & 69 & 45 \\ 104 & 105 & 100 & 93 & 69 & 45 & 53 & 93 \\ 108 & 103 & 94 & 76 & 42 & 50 & 113 & 141 \\ 115 & 111 & 99 & 79 & 73 & 80 & 99 & 118 \\ 124 & 122 & 114 & 107 & 109 & 100 & 100 & 110 \\ 132 & 134 & 129 & 125 & 120 & 116 & 110 & 109 \\ 137 & 141 & 141 & 139 & 135 & 131 & 124 & 118 \\ 142 & 146 & 148 & 150 & 148 & 145 & 141 & 138 \end{bmatrix}. \quad (3.16)$$

Обчислимо двовимірне дискретне косинусне перетворення матриці x згідно з формулами (3.7), (3.8). Матриця коефіцієнтів ДКП матиме вигляд

$$X = \begin{bmatrix} 873{,}4 & 58{,}8 & 31{,}2 & -28{,}9 & 5{,}6 & -1{,}9 & -3{,}3 & -0{,}2 \\ -160{,}2 & 21{,}8 & 23{,}3 & -20{,}4 & 7{,}5 & 1{,}7 & -2{,}0 & 0{,}3 \\ 40{,}5 & 19{,}2 & -57{,}9 & 15{,}2 & -1{,}4 & -0{,}3 & 0{,}3 & -2{,}4 \\ 6{,}3 & 32{,}8 & -60{,}7 & 33{,}4 & -4{,}1 & -8{,}8 & 3{,}0 & -0{,}7 \\ 15{,}9 & -1{,}1 & -25{,}9 & 26{,}4 & -7{,}4 & -5{,}2 & 2{,}5 & 2{,}1 \\ 3{,}9 & -7{,}8 & -0{,}2 & 12{,}1 & -14{,}7 & 6{,}3 & -0{,}9 & 0{,}9 \\ 11{,}8 & -15{,}7 & 10{,}5 & 0{,}7 & -9{,}0 & 10{,}5 & -4{,}6 & 0{,}4 \\ 8{,}2 & -10{,}3 & 4{,}7 & -1{,}8 & -2{,}1 & 6{,}5 & -4{,}4 & 2{,}0 \end{bmatrix}. \quad (3.17)$$

Виконаємо квантування коефіцієнтів шляхом поелементного ділення матриці коефіцієнтів X (3.12) на матрицю квантування QY з (3.10) та округлення результатів до цілих значень. Отримуємо матрицю Xq квантованих коефіцієнтів ДКП

$$X_q = \begin{bmatrix} 55 & 5 & 3 & -2 & 0 & 0 & 0 & 0 \\ -13 & 2 & 2 & -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & -4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & -3 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.18)$$

Очевидно, що ненульові компоненти зосереджені у верхній лівій частині матриці, що відповідає низькочастотним просторовим гармонікам.

За допомогою «зигзаг»-сканування (рис 3.4) перетворимо матрицю Xq розміром 8×8 на 64-елементний вектор U.

U = { 55 5 –13 3 2 3 –2 2 1 0 1 2 –4 –1 0 0 0 1 –3 0 0 0 0 –1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0}.

Кодуємо символи методом RLE відповідно до табл. 3.18. На виході кодера утворюється послідовність трійок чисел, що включають поля: «лічильник нулів», «довжина», «амплітуда». Перший елемент вектора для простоти кодуємо за загальними правилами

URLE = { (0, 6, 55), (0, 3, 5), (0, 4, –13), (0, 2, 3), (0, 2, 2), (0, 2, 3), (0, 2, –2), (0, 2, 2), (0, 1, 1), (1, 1, 1), (0, 2, 2), (0, 3, –4), (0, 1, –1), (3, 1, 1), (0, 2, –3), (4, 1, –1), (0, 1, 1), (0, 0) }.

Записуємо двійкові коди символів, при цьому коди Гаффмана отримуємо відповідно до табл. 3.19. Перші два поля містять по чотири біти, число біт в третьому полі визначається параметром «довжина» .

*U*2= { 0000 0110 110111, 0000 0011 101, 0000 0100 0010, 0000 0010 11, 0000 0010 10, 0000 0010 11, 0000 0010 01, 0000 0010 10, 0000 0001 1, 0001 0001 1, 0000 0010 10, 0000 0011 011, 0000 0001 0, 0011 0001 1, 0000 0010 00, 0100 0001 0, 0000 0001 1, 0000 0000 }

Очевидно, що двійково закодований фрагмент зображення містить 180 біт. У той самий час вихідний фрагмент складався з 8×8 = 64 восьмибітних відліків яскравості, тобто вимагав 512 біт.

Таким чином, коефіцієнт стиснення відповідно до (3.1) становить

Kc = S0/Sc = 512/180 = 2,8444. ■

Відновлення зображення, стисненого за алгоритмом JPEG, виконується в порядку, зворотному ущільненню.

1. Декодування амплітуд коефіцієнтів перетворення методом Гаффмана відповідно до табл. 3.19.

2. Декодування послідовності методом RLE .

3. Перетворення вектора з 64 елементів на матрицю квантованих коефіцієнтів $\widehat{X}_q$ 8×8 відповідно до рис 3.4.

4. Відновлення матриці коефіцієнтів ДКП $\widehat{X}_q$ шляхом множення матриці квантованих коефіцієнтів Xq на матрицю квантування Q

$$\widehat{X} = \widehat{X}_q \cdot Q. \tag{3.19}$$

5. Обчислення двовимірного зворотного дискретного косинусного перетворення та округлення результатів.

Відомо [15], що матриця зворотного дискретного косинусного перетворення дорівнює транспонованої матриці прямого ДКП (3.8) $C{-}1{=}CT$.

Обчислення відновленого двовимірного сигналу виконується аналогічно (3.7)

$$\widehat{x} = C^{-1} \cdot \widehat{X} \cdot \left(C^{-1}\right)^T = C^T \cdot \widehat{X} \cdot C \tag{3.20}$$

6. Аналогічний розрахунок компонент Cb , Cr і перетворення з простору YCbCr в простір RGB.

Кількісно ступінь відмінності між вихідним та відновленим після стиснення зображеннями може бути визначений за відносним середньоквадратичним критерієм.

$$\varepsilon_{ско} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{k=1}^{M}\left(x_{ik} - \widehat{x}_{ik}\right)^2}{\sum_{i=1}^{N}\sum_{k=1}^{M} x_{ik}^{\ 2}}}. \tag{3.21}$$

Приклад 3.45. Відновити стислий у прикладі 3.44 фрагмент зображення. Визначити помилку стиснення.

Рішення. Оскільки кодування методом RLE та Хаффмана є повністю оборотним, то матриця $\widehat{X}_q$ квантованих коефіцієнтів (3.18) відновлюється без змін $\widehat{X}_q = X_q$.

Обчислюємо матрицю коефіцієнтів ДКП згідно (3.13).

$$\widehat{X} = \begin{bmatrix} 880 & 55 & 30 & 32 & 0 & 0 & 0 & 0 \\ -156 & 24 & 28 & -19 & 0 & 0 & 0 & 0 \\ 42 & 13 & -64 & 24 & 0 & 0 & 0 & 0 \\ 0 & 34 & -66 & 29 & 0 & 0 & 0 & 0 \\ 18 & 0 & -37 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.22}$$

Відповідно (3.20) обчислюємо зворотне двовимірне дискретне косинусне перетворення від матриці $\widehat{X}$ та округляємо результат до цілих.

$$\widehat{x} = [C^T \cdot \widehat{X} \cdot C].$$

$$\widehat{x} = \begin{bmatrix} 97 & 104 & 114 & 118 & 109 & 88 & 62 & 45 \\ 108 & 107 & 99 & 82 & 66 & 66 & 80 & 95 \\ 113 & 110 & 94 & 65 & 44 & 56 & 98 & 136 \\ 112 & 111 & 108 & 87 & 68 & 73 & 103 & 133 \\ 118 & 122 & 121 & 113 & 103 & 98 & 102 & 109 \\ 133 & 130 & 127 & 124 & 121 & 115 & 109 & 104 \\ 140 & 138 & 137 & 137 & 136 & 133 & 126 & 121 \\ 136 & 143 & 151 & 155 & 154 & 148 & 143 & 139 \end{bmatrix}. \quad (3.23)$$

Очевидно, що матриця (3.23) відновлених рівнів яскравостей фрагмента близька до вихідної (3.16), але дещо відрізняється від неї. Візуальні відмінності між вихідним та відновленим фрагментами сигналу яскравості видно на рис. 3.6.

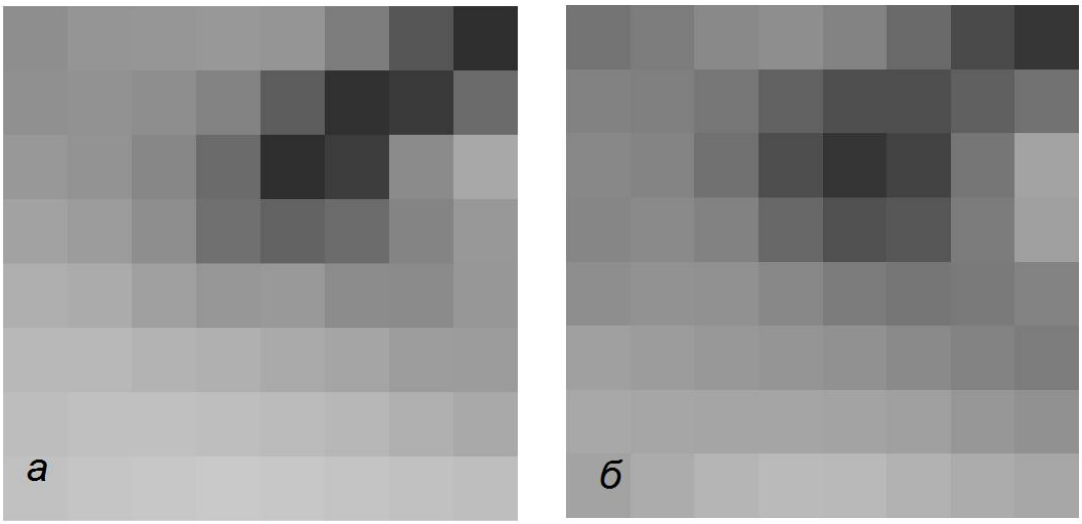


Рисунок 3.6 – Складова яскравості вихідного (*а*) та відновленого (*б*) фрагментів зображення

Обчислена згідно (3.21) відносна середньоквадратична похибка відновлення фрагмента становитиме

$$\varepsilon_{ско} = \sqrt{\frac{\sum_{i=1}^{8}\sum_{k=1}^{8}\left(x_{ik} - \widehat{x}_{ik}\right)^2}{\sum_{i=1}^{8}\sum_{k=1}^{8} x_{ik}^{\ 2}}} = \sqrt{\frac{2968}{810570}} = 0{,}0605. \blacksquare$$

Алгоритм JPEG інтенсивно розвивається та удосконалюється до теперішнього часу. Для стиснення зображень застосовується, зокрема, JPEG 2000 – графічний формат, який замість дискретного косинусного перетворення використовує технологію *вейвлет-перетворення*, що ґрунтується на поданні сигналу у вигляді суперпозиції базових функцій – хвильових пакетів. В результаті такої компресії зображення виходить гладкішим і чіткішим, а розмір файлу в порівнянні з JPEG при однаковій якості виявляється меншим.

На основі алгоритму JPEG також розроблено низку стандартів стиснення відео, наприклад MPEG-4.

Завдання для самостійного вирішення

1. Джерело має алфавіт із десяти символів із ймовірностями $p(x_1)=0{,}21$; $p(x_2)=0{,}05$; $p(x_3)=0{,}11$; $p(x_4)=0{,}12$; $p(x_5)=0{,}13$; $p(x_6)=0{,}09$; $p(x_7)=0{,}14$; $p(x_8)=0{,}02$; $p(x_9)=0{,}10$; $p(x_{10})=0{,}03$. Визначити надмірність повідомлень, складених із цього алфавіту.

2. Джерело має алфавіт із шести символів з ймовірностями $p(x_1)=0{,}31$; $p(x_2)=0{,}19$; $p(x_3)=0{,}16$; $p(x_4)=0{,}14$; $p(x_5)=0{,}11$; $p(x_6)=0{,}09$. Для передачі каналом використовується нерівномірний код $x_1 \to 00$, $x_2 \to 01$, $x_3 \to 10$, $x_4 \to 110$, $x_5 \to 1110$, $x_6 \to 1111$. Визначити середню кількість двійкових розрядів, що припадають на один символ джерела. Порівняти з ентропією джерела.

3. Визначити, які з наведених у таблиці нерівномірних кодів є префіксними.

Символ	x_1	x_2	x_3	x_4	x_5	x_6
Код 1	1	000	001	010	0110	0111
Код 2	01	000	001	110	0110	1111
Код 3	1	010	001	011	0110	0001

4. Для передачі каналом використовується нерівномірний код $x_1 \rightarrow 00$, $x_2 \rightarrow 01$, $x_3 \rightarrow 10$, $x_4 \rightarrow 110$, $x_5 \rightarrow 1110$, $x_6 \rightarrow 1111$. Декодувати повідомлення 110000110111011111110.

5. Побудувати код Шеннона-Фано та код Гаффмана для дискретного джерела, символи якого мають задані у таблиці ймовірності. Визначити максимальну ентропію $H_{\max}$, ентропію джерела $H(X)$, середнє число біт на $n_{сер}$, коефіцієнт стиснення μ. Порівняти ефективність кодів Шеннона-Фано та Гаффмана.

N	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}
7	0.31	0.20	0.17	0.14	0.08	0.06	0.04				
8	0.28	0.16	0.13	0.11	0.10	0.09	0.09	0.04			
9	0.27	0.17	0.12	0.11	0.09	0.08	0.07	0.05	0.04		
10	0.23	0.12	0.11	0.10	0.11	0.10	0.07	0.09	0.04	0.03	
11	0.22	0.13	0.12	0.11	0.11	0.10	0.07	0.06	0.04	0.02	0.02

6. Закодувати унарним двійковим кодом число A =11.

7. Декодувати записане в унарному коді число $A_{ун}$ = 1111111111110.

8. Декодувати записану в унарному коді послідовність $A_{ун}$=11101111111101101001111110.

9. Закодувати гамма-кодом Левенштейна число A = 923.

10. Декодувати записане в гамма-коді Левенштейна число $A_{лев}$=010001010001011.

11. Декодувати записану в гамма коді Левенштейна послідовність чисел $A_{лев}$= 0101011000100010110100010000101011011.

12. Закодувати гамма-кодом Еліаса число A = 713.

13. Декодувати записане в гамма-коді Еліаса число $A_{гамел}$=000000011010010.

14. Декодувати записану в гамма-коді Еліаса послідовність чисел $A_{гамел}$=00001011100100010000000110001010001111.

15. Закодувати дельта-кодом Еліаса число A = 1226.

16. Декодувати записане в дельта-коді Еліаса число $A_{дел}$=0001010100110100.

17. Декодувати записану в дельта-коді Еліаса послідовність чисел $A_{дел}$=00101010100110101100001001101101 01.

18. Закодувати кодом Голомба з параметром m=7 число A=80.

19. Декодувати записане у коді Голомба з параметром m=11 число $A_{гол}$=11111101000.

20. Декодувати записану в коді Голомба з параметром m=5 послідовність чисел $A_{гол}$=11111110011111111111001001110000.

21. Закодувати кодом Райса з параметром k =6 число A = 703.

22. Декодувати записане в коді Райса із заданим параметром k =5 число $A_{райс}$=11111111010111.

23. Декодувати записану у коді Райса з параметром k=3 послідовність чисел $A_{райс}$= 11110010110101111111101101001111110001.

24. Закодувати префіксним кодом Фібоначчі число A = 420.

25. Декодувати записане у префіксному коді Фібоначчі число $A_{фібпреф}$=1010001000010011.

26. Декодувати записану у префіксному коді Фібоначчі послідовність чисел $A_{фібпреф}$=10001011010101110010101101010010 11.

27. Закодувати кодом Івен-Роде число A = 2611.

28. Декодувати записане в коді Івен-Роде число $A_{ір}$=10010111110111010.

29. Стиснути за допомогою алгоритму RLE послідовність {UUUUUTVVVM99GGGGGGGSSSERRRR}. Визначити коефіцієнт компресії.

30. Розпакувати отриману в результаті стиснення методом RLE послідовність {G 4, 3 5, 1 1 , M 6, L 1, D 2}. Визначити коефіцієнт компресії .

31. Стиснути за допомогою LZW алгоритму послідовність SIXSICKHICKSNICKSIXSLICKBRICKS (англ . *Six sick hicks nick six slick bricks, шість хворих провінціалів стягнули шість гладких цеглин*). Визначити коефіцієнт компресії.

32. Розпакувати отриману в результаті стиснення методом LZW послідовність {0 4 1 4 8 2 0 3 13 9 7 16 19 5 6 11 17}. Вихідний словник наведено у таблиці.

Символ	Код	Бітовый код
0: В	0	000
1: Д	1	001
2: З	2	010
3: І	3	011
4: О	4	100
5: П	5	101
6: Р	6	110
7: У	7	111

33. Перетворити за допомогою BWT алгоритму послідовність {КОНТРАБАНДА }.

34. Відновити перетворену за допомогою алгоритму BWT послідовність Y = {ДАТИТРААНК, 2}.

35. Побудувати матрицю ДКП для N=6 . Обчислити ДКП послідовності $\{x_i\} = \{x_0, x_1, x_2, x_3, x_4, x_5\} = \{3, -2, 2, -4, 4, 1\}$.

36. Побудувати матрицю ДКП для N=3 . Обчислити двовимірне ДКП послідовності $x_{ij} = \begin{bmatrix} 4 & -2 & 1 \\ 3 & 2 & -1 \\ -3 & 0 & 5 \end{bmatrix}$.

37. Стиснутий методом JPEG фрагмент зображення закодовано послідовністю U_2 = {0000 0101 11011, 0000 0011 011, 0000 0100 1010, 0000 0010 01, 0000 0011 110, 0010 0010 11, 0000 0001 0, 0011 0001 1, 0100 0010 1 1, 0000 0010 11 , 0101 0001 1, 0000 0000}. Відновити матрицю квант ованих коефіцієнтів.

ГЛАВА 4

ІНФОРМАЦІЙНІ ХАРАКТЕРИСТИКИ КАНАЛІВ ЗВ’ЯЗКУ

4.1 Канали зв’язку. Математичні моделі каналів зв’язку

Канал зв'язку – сукупність пристроїв, призначених для передачі повідомлень від одного місця до іншого або від моменту часу до іншого.

Лінія зв'язку – фізичне середовище, в якому поширюються сигнали, що відповідають повідомленням, що передаються. Вона характеризується діапазоном частот, що несуть сигнал.

Канали зв’язку поділяються на *комутовані* та *виділені*. Комутовані – це канали, створені з окремих ділянок тільки на час передачі по них інформації, після закінчення сеансу зв'язку такий канал розривається. Виділені канали – це канали, які організуються на тривалий час і мають постійні характеристики за довжиною та пропускною здатністю.

За характером сигналів на вході та виході канали характеризуються:

- *дискретні*, тобто. на вході та виході каналу сигнали дискретні;
- *неперервні*, тобто. на вході та виході каналу сигнали безперервні;
- *дискретно-безперервні*, тобто. дискретні з боку входу та безперервні з боку виходу чи навпаки.

Будь-який дискретний, чи дискретно-неперервний, канал містить у собі безперервний канал. Слід пам'ятати, що дискретність і безперервність каналу не пов'язана з характером повідомлень, що передаються. Можна надіслати дискретні повідомлення безперервно та безперервні дискретні.

Дискретний канал вважається заданим, якщо відомі множини символів (алфавіту) на вході та виході, а також ймовірнісні властивості формування (передачі) цих символів.

Для передачі каналом повідомлення із знаків алфавіту джерела x_1, x_2, x_3, …, x_l перетворюється на дискретні послідовності символів з іншого алфавіту y_1, y_2, y_3, …, y_m.

Базовим параметром, що найбільш повно описує канал зв'язку, є матриця каналу

$$M_c = \begin{vmatrix} p(y_1/x_1) & p(y_2/x_1) & \dots & p(y_n/x_1) \\ p(y_1/x_2) & p(y_2/x_2) & \dots & p(y_n/x_2) \\ \dots & \dots & \dots & \dots \\ p(y_1/x_m) & p(y_2/x_m) & \dots & p(y_n/x_m) \end{vmatrix} \quad (4.1)$$

елементами, якою є умовні можливості трансформації символів вхідного алфавіту в символи вихідного алфавіту сигналів.

Якщо кожному символу x_k алфавіту $\{X\}$ однозначно відповідає символ y_k алфавіту $\{Y\}$ канал називається *каналом без перешкод*.

Наявність перешкод призводить до того, що той самий символ x_k може фіксуватися на прийомі у вигляді різних символів y_l ($l \neq k$). Отже, для каналів із перешкодами можна говорити лише про умовну ймовірність $p(y_j/x_i)$ появи на виході приймача символу y_j за умови, що було передано символ x_i.

Дискретні канали поділяють на:

- канали з пам’яттю або без пам’яті;
- канали з постійними або змінними параметрами;
- канали зі стиранням або без стирання;
- симетричні або несиметричні канали.

Якщо ймовірності $p(y_j/x_i)$ залежать від попереднього стану має місце канал з пам'яттю, якщо не залежить, то канал без пам'яті.

Якщо параметри каналу (коефіцієнт передачі, час поширення, число променів у точці прийому) не змінюються в часі, то говорять про канал з постійними параметрами, що має властивість стаціонарності. Якщо параметри змінюються в часі, це канал зі змінними параметрами або нестаціонарний.

Моделі дискретних каналів зв'язку без пам'яті.

1. *Симетричний канал без пам'яті* визначається як дискретний канал, в якому кожен переданий символ може бути прийнятий помилково з фіксованою вірогідністю $p_{\text{пом}}$ і правильно з ймовірністю $1-p_{\text{пом}}$. Тоді ймовірність того, що прийнятий символ y_j, якщо переданий x_i

$$p(y_j / x_i) = \begin{cases} 1 - p_{пом} & при \quad i = j \\ \dfrac{p_{пом}}{m-1} & при \quad i \neq j \end{cases}. \tag{4.2}$$

Якщо m=2 (алфавіт складається з двох символів «0» і «1»), то канал називається *двійково-симетричний канал без пам'яті (ДСК)*, для якого матриця каналу має вигляд

$$M_c = \begin{vmatrix} p(0/0) & p(1/0) \\ p(0/1) & p(1/1) \end{vmatrix} = \begin{vmatrix} 1 - p_{пом} & p_{пом} \\ p_{пом} & 1 - p_{пом} \end{vmatrix}. \tag{4.3}$$

Враховуючи, що ДСК є каналом без пам'яті, для якого ймовірність спотворення кожного наступного символу не залежить від того чи були спотворені попередні, величина $p_{пом}$ повністю визначає якість ДСК і служить вичерпною характеристикою (рис. 4.1).

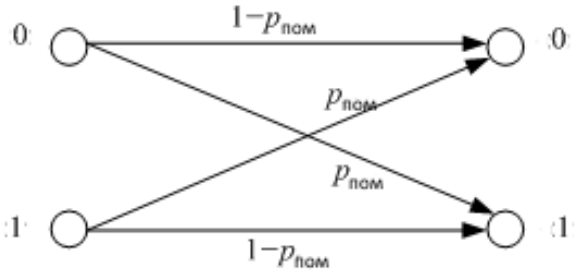


Рисунок 4.1 – Двійково-симетричний канал без пам'яті

Як випливає з моделі, ймовірність спотворення одного символу дорівнює $p_{пом}$. Якщо дані, що передаються в канал, поділяються на блоки з n символів, то згідно з (1.31) ймовірність появи l спотворених символів в блоці становить

$$P_{\ell,n} = C_n^{\ell} p_{пом}^{\ell} (1 - p_{пом})^{n-\ell}. \tag{4.4}$$

2. *Симетричний канал без пам'яті зі стиранням* відрізняється від попереднього тім, що алфавіт на виході канал містить додатковий символ, що позначається знаком s. Цей символ з'являється тоді, коли вирішальна

схема не може розпізнати переданий символ. Якщо m=2, то канал *двійково-симетричний канал із стиранням* (рис. 4.2)

$$M_c = \begin{vmatrix} 1-p_S & p_S \\ p_S & 1-p_S \end{vmatrix}, \tag{4.5}$$

де p_s – ймовірність стирання, $1-p_s$ – ймовірність правильного прийому.

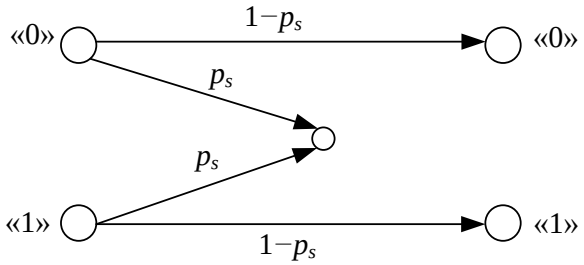


Рисунок 4.2 − Граф моделі канала зі стиранням

3. *Симетричний канал без пам'яті з помилками та стиранням*

Матриця перехідних ймовірностей для даного каналу визначається виразом (4.5)

$$M_c = \begin{vmatrix} 1-p_{пом}-p_s & p_s & p_{пом} \\ p_{пом} & p_s & 1-p_{пом}-p_s \end{vmatrix}, \tag{4.6}$$

де $1-p_{пом}-p_S$ –ймовірність правильного прийому; $p_{пом}$ – ймовірність помилкового прийому; p_S – ймовірність стирання.

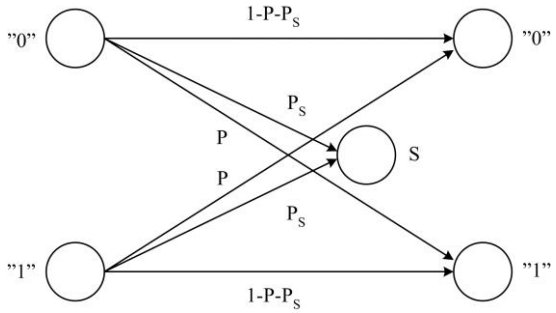


Рисунок 4.3 – Граф симетричного каналу без пам'яті з помилками та стиранням

4. *Несиметричний канал без пам'яті* характеризується, як і попередні моделі, тим, що помилки виникають незалежно одна від одної, проте ймовірність помилок залежить від того, який символ передається

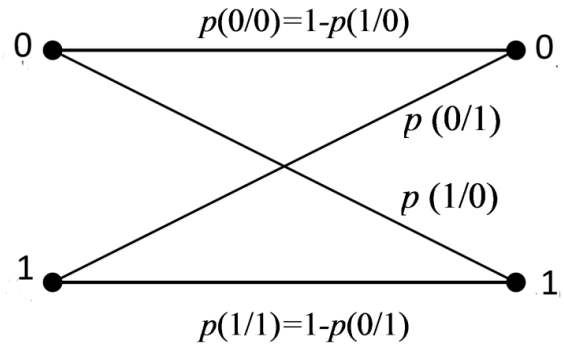


Рисунок 4.4 – Несиметричний канал без пам'яті

В двійковому несиметричному каналі ймовірність $p(1/0)$ ухвалення символу "1" при передачі символу "0" не дорівнює ймовірності $p(1/0)$ ухвалення "0" при передачі "1".

Моделі дискретних каналів зв'язку з пам'яттю.

1. *Марківський канал* − канал, у якому ймовірність помилки утворює простий ланцюг Маркова, тобто. залежить від того правильно чи помилково прийнято попередній символ, але не залежить від того який символ передається. Матриця перехідних ймовірностей такого каналу

$$M_c = \begin{vmatrix} 1-p_1 & p_1 \\ p_2 & 1-p_2 \end{vmatrix}, \qquad (4.7)$$

де p_1 – умовна ймовірність прийняти (i+1)-й символ помилково, якщо i-й принят правильно; $1-p_1$ – умовна ймовірність прийняти (i+1)-й символ правильно, якщо i-й принят правильно; p_2 – умовна ймовірність прийняти (i+1)-й символ помилково, якщо i-й принят помилково; $1-p_2$ – умовна ймовірність прийняти (i+1)-й символ правильно, якщо i-й принят помилково.

2. *Двійковий канал з помилками, що групуються*. У реальних каналах зв'язку помилки, що виникають, мають явно виражену тенденцію до групування, тобто, якщо в послідовності символів з'явився помилковий, то ймовірність спотворення наступного за ним символу зростає. Групування

помилок викликано впливом фізичного середовища та параметрів технічних засобів каналу на якість передачі інформації. Для врахування групування помилок додатково до p_0, що описує канал, вводиться параметр α-коефіцієнт групування помилок. При цьому ймовірність виникнення не менш ніж l помилок на довжині блоку n при $p_0<<1$ і $l<<n$ визначається як

$$P_{\ell,n} \approx \left(\frac{n}{\ell}\right)^{1-\alpha} \cdot p_0 . \qquad (4.8)$$

Коефіцієнт α змінюється в межах від 0 до 1. При α = 0 – маємо канал із незалежними помилками (без пам'яті), при α = 1 – канал, у якому всі помилки поєднуються в один суцільний «пакет». При решті значень α є тенденція до групування помилок, яка тим вища, чим більше α. Таким чином, для даної моделі пара характеристик (параметрів) p_0 и α повністю описують канал.

Неперервним каналом називається канал, призначений для передачі неперервних повідомлень. Канал вважається заданим, якщо відомі статистичні дані про повідомлення на його вході та виході та обмеження, що накладаються на вхідні повідомлення фізичними характеристиками каналу.

Неперервний канал має наступні математичні різновиди:

- ідеальний канал без завад, з постійними параметрами;
- канал зв'язку з адитивним гауссовим шумом;
- гауссовий канал з невизначеною фазою сигналу;
- однопроменевий гауссовий канал із загальними завмираннями;
- гауссовий багатопроменевий канал із завмираннями та адитивним шумом з міжсимвольною інтерференцією.

Ідеальний канал зв'язку без завад – це лінійне коло зі сталою функцією передачі, що зосереджена в обмеженій смузі частот, де відсутні завади. Вихідний сигнал при заданому вхідному буде детермінованим. Вихідний сигнал такого каналу визначається

$$z(t) = \mu \cdot s(t - \tau) \qquad (4.9)$$

де $z(t)$ – суміш сигналу та завади на виході каналу; s(t) – вхідний сигнал; μ – ослаблення або підсилення сигналу; τ – затримка в каналі.

Ця модель каналу зв'язку частіше застосовується для опису кабельних каналів дуже короткої довжини де наявність адитивних завад ще не відчувається.

Канал зв'язку з адитивним гауссовим шумом – це канал з білим або квазібілим шумом (з рівномірною спектральною щільністю в смузі спектра сигналу) в якого $\mu(t) = \mu = \text{const}$, $\tau(t) = \tau = \text{const}$, а завада $\varepsilon(t)$ має гауссовий (нормальний) розподіл ймовірностей.

Із врахуванням цього математична модель неперервного каналу записується у вигляді

$$z(t) = \mu\, s(t - \tau) + \varepsilon\,(t), \tag{4.10}$$

де $\varepsilon(t)$ – гауссовий адитивний шум з нульовим математичним сподіванням.

Гауссовими каналами, наприклад, є канали супутникового зв'язку, проводових ліній при незначних завадах та однопроменевих радіоканалів без завмирань.

4.2 Швидкість передачі та пропускна здатність дискретного каналу зв’язку с завадами та без завад

Для характеристики дискретного каналу зв'язку використовують два поняття швидкості передачі: технічної та інформаційної.

Під *технічною швидкістю передачі* V_τ званої також швидкістю маніпуляції, мають на увазі число елементарних символів (сигналів), що передаються каналом в одиницю часу. Вона залежить від властивостей лінії зв'язку та швидкодії апаратури каналу та визначається з виразу

$$V_\tau = \frac{1}{\tau_{сер}} \text{ Бод}, \tag{4.11}$$

де $\tau_{сер}$ – середнє значення тривалості символу.

Одиницею вимірювання технічної швидкості служить бод – швидкість, коли за одну секунду передається один символ. Інформаційна швидкість визначається середньою кількістю інформації, що передається

каналом зв'язку в одиницю часу. Вона залежить від характеру даного каналу зв'язку (обсяг алфавіту використовуваних символів, технічна швидкість їх передачі, статистичні властивості перешкод у лінії), і від ймовірностей символів, що надходять вхід та їх статистичного взаємозв'язку.

Під *інформаційною швидкістю* розуміють середню кількість інформації, що передається по каналу в одиницю часу

$$V = V_\tau I(Y, X) = \frac{1}{\tau_{сер}} I(Y, X), \tag{4.12}$$

де $I(Y, X)$ – середня кількість інформації, що переноситься одним символом.

Для теорії та практики важливо з'ясувати, до якої межі та яким шляхом можна підвищити швидкість передачі інформації по конкретному каналу зв'язку, тобто. визначити пропускну спроможність каналу.

Пропускна здатність каналу дорівнює тієї максимальної швидкості передачі інформації по даному каналу, якої можна досягти при найдосконаліших способах передачі та прийому

$$C = \max V = V_\tau \max I(Y, X), \tag{4.13}$$

Пропускна здатність каналу та швидкість передачі каналом вимірюються числом двійкових одиниць інформації в секунду (дв.од/с). Розглянемо питання передачі повідомлень для дискретного каналу без завад та дискретного каналу з перешкодами.

Дискретний канал без перешкод.

У будь-якому реальному каналі завжди є перешкоди. Однак, якщо їхній рівень настільки малий, що ймовірність спотворення практично дорівнює нулю, можна вважати, що всі сигнали передаються неспотвореними. У цьому випадку середня кількість інформації, що переноситься одним символом, визначається за формулою

$$I(Y, X) = I(X, Y) = H(X), \tag{4.14}$$

а вираз для інформаційної швидкості (4.12) набуває вигляду

$$V = \frac{1}{\tau_{сер}} \cdot H(X) = \frac{1}{\tau_{сер}} \cdot \left(- \sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \right). \quad (4.15)$$

При цьому максимальне значення кількості інформації на один символ

$$\max I(Y, X) = H_m(X), \quad (4.16)$$

де H_m – максимальна ентропія джерела сигналів, що виходить при рівномірному розподілі ймовірностей символів алфавіту джерела $p(x_1) = p(x_2) = \ldots = p(x_m) = 1/m$. Оскільки максимальна ентропія виражається як $H_m(x) = \log_2 m$, то пропускна здатність дискретного каналу без завад виражена в одиницях інформації за одиницю часу дорівнює

$$C = V_\tau \log_2 m = \frac{1}{\tau_c} \log_2 m. \quad (4.17)$$

Аналіз виразу (4.12) показує, що збільшення швидкості передачі інформації, тобто досягнення максимуму пропускної здатності, може здійснюватися як шляхом збільшення ентропії повідомлення, так і шляхом зменшення тривалості сигналу. В останньому випадку зміни тривалості сигналу призводять до зміни ширини спектра сигналу, при цьому верхня межа цих змін не повинна перевищувати значення смуги пропускання каналу.

Зазвичай канал має обмежену смугу частот, тобто смугою, в межах якої здійснюється передача частот гармонік вхідного сигналу без помітних спотворень. Вважаючи обмеження смуги каналу частот величиною f_B, вкажемо обмеження на тривалість сигналу знизу

$$\tau_c \geq \frac{1}{2 \cdot f_B}. \quad (4.18)$$

Подальше зменшення τ_c призведе до суттєвих спотворень форми сигналу, що знижує якість передачі в каналі. Тому тривалість сигналу τ_c вважатимуться заданою каналом.

У той самий час слід зазначити, що і за нерівноймовірних символах швидкість передачі наближається до пропускної спроможності каналу по мірі подовження повідомлення. Так із властивості E асимптотичної рівноймовірності слідує, що у нескінченно довгих повідомленнях буде реалізовуватися оптимальний (N_B) розподіл символів. Тоді для повідомлень із n символів

$$C = \lim_{n\to\infty} \frac{-n\sum_{i=1}^{m} p_i \cdot \log_2 p_i}{n\sum_{i=1}^{m} \tau \cdot p_i} = \frac{n \cdot \log_2 m}{n \cdot \tau_c} = \frac{1}{\tau_c} \log_2 m. \qquad (4.19)$$

Приклад 4.1. Кількість повідомлень, що передаються з контрольованого пункту про стан чотирьох об'єктів однаково. Спостереженням встановлено, що у середньому об'єкт 1 включено 80%, об'єкт 2 – 40%, об'єкт 3 – 60%, об'єкт 4 – 20%. В решту часу об'єкти вимкнено. Визначити швидкість передачі інформації та пропускну спроможність каналу зв'язку, якщо тривалість одного повідомлення 10^{-3}с.

Рішення. Виходячи з однакової кількості повідомлень про стан об’єктів можна записати, що $p(x_1)=p(x_2)=p(x_3)=p(x_4)=1/4$. З статистики спостережень можна записати вирази для визначення ймовірності того, що об’єкти перебувають у включеному стані $P_в(x_i)$

$P_в(x_1)=0{,}8\ p(x_1)$, $P_в(x_2)=0{,}4\ p(x_2)$, $P_в(x_3)=0{,}6\ p(x_3)$, $P_в(x_4)=0{,}2\ p(x_4)$

та вимкнені $P_{вим}(x_i)$ відповідно

$P_{вим}(x_1)=0{,}2\ \ p(x_1)$, $P_{вим}(x_2)=0{,}6\ \ p(x_2)$, $P_{вим}(x_3)=0{,}4\ \ p(x_3)$, $P_{вим}(x_4)=0{,}8\ p(x_4)$,

Результати розрахунку ймовірностей того, що об'єкти знаходяться у включеному або відключеному стані, зведемо в табл. 4.1.

Таблиця 4.1 – Ймовірності включеного та вимкненого стану об’єкту

$P_в(x_1)$	$P_в(x_2)$	$P_в(x_3)$	$P_в(x_4)$	$P_{вим}(x_1)$	$P_{вим}(x_2)$	$P_{вим}(x_3)$	$P_{вим}(x_4)$
0,2	0,1	0,15	0,05	0,05	0,15	0,1	0,2

Тоді швидкість передачі інформації згідно (4.15) буде рівною

$$V = V_\tau \cdot H(X) = \frac{1}{10^{-3}} \cdot -\begin{pmatrix} 0{,}2 \cdot \log_2 0{,}2 + 0{,}1 \cdot \log_2 0{,}1 + 0{,}15 \cdot \log_2 0{,}15 + \\ + 0{,}05 \cdot \log_2 0{,}05 + 0{,}05 \cdot \log_2 0{,}05 + 0{,}15 \cdot \log_2 0{,}15 + \\ + 0{,}1 \cdot \log_2 0{,}1 + 0.2 \cdot \log_2 0{,}2 \end{pmatrix} =$$

$$= \frac{1}{10^{-3}} \cdot (0{,}464 + 0{,}216 + 0{,}332 + 0{,}410 + 0{,}410 + 0{,}332 + 0{,}216 + 0{,}464) =$$

=284 біт/сек

Пропускна здатність в цьому випадку дорівнює

$$C = V_\tau \cdot \max(H(x) = V_\tau \cdot \log_2 m = 10^3 \cdot 3 = 3000 \textit{біт/сек},$$

де m=8 – загальне число станів системи (чотирьох об'єктів). ■

Приклад 4.2. Джерело виробляє три повідомлення з ймовірностями: $p(x_1)$=0,2; $p(x_2)$=0,5; $p(x_3)$=0,3. Повідомлення незалежні і передаються рівномірним двійковим кодом (m=2) із тривалістю символів, що дорівнює 20 мкс. Визначити швидкість передачі по каналу зв'язку без перешкод.

Рішення.

$$H(X) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i = -(0{,}2 \cdot \log_2 0{,}2 + 0{,}5 \cdot \log g_2 0{,}5 + 0{,}2 \cdot \log_2 0{,}2) =$$

$$= 0{,}4644 + 0{,}5 + 0{,}5211 = 1{,}4855 \textit{біт}.$$

Для передачі трьох повідомлень рівномірним кодом необхідно два розряди, причому тривалість кодової комбінації дорівнює 2τ.

Середня швидкість передачі сигналу $V_\tau = \frac{1}{2\tau} = 25000\,\textit{бод}$.

Швидкість передачі біт/сек $V=V_\tau \cdot H(X)$ = 25000·1,4855 = 37137,5 бит/сек. ■

Для двійкового алфавіту (m=2) $max\big(H(Z)\big) = 1$ біт/симв. Тому

$$C_{дв} \approx 2 \cdot f_в . \qquad (4.20)$$

Наприклад, можна вважати, що двійковий канал без шумів зі смугою пропускання 3,1 кГц має пропускну здатність 6200 біт/с.

Дискретний канал із перешкодами.

Дія перешкод у дискретних каналах порушує взаємну однозначність відображення вихідних і вхідних символів каналу, тому за прийнятим сигналом y_j можна лише з деякою ймовірністю стверджувати, який із сигналів $\{x_i\}$ був переданий насправді.

Джерело інформації характеризується апріорним розподілом ймовірностей $p(x_i)$ (i = 1, …, m), який визначає ентропію джерела $H(X)$. Після отримання символу y_j (j = 1, …, m) на приймальній стороні каналу джерело може бути охарактеризовано апостеріорним розподілом $p(x_i/y_j)$ (i, j = 1, …, m) символів алфавіту X, де $p(x_i/y_j)$ – ймовірність того, що передано символ x_i,, якщо прийнято сигнал y_j. Апостеріорний розподіл визначає ентропію $H(X/Y)$ вхідного сигналу за умови, що вихідний сигнал каналу відомий. $H(X/Y)$ можна назвати втратою інформації, обумовленої впливом перешкод повідомлення, що передається каналом зв'язку.

Якщо від загальної кількості інформації (на один елемент повідомлення) $H(X)$ відняти втрату інформації, обумовлену перешкодами $H(X/Y)$, то отримаємо кількість інформації, яка міститься в прийнятій сукупності повідомлень Y щодо сукупності повідомлень X, що передається

$$I(Y, X) = H(X) - H(X / Y) \tag{4.21}$$

або

$$I(Y, X) = H(Y) - H(Y / X). \tag{4.22}$$

Рівень перешкод лінії зв'язку може бути настільки великий, що повідомлення X і Y стають статистично незалежними. У цьому випадку $H(X/Y) = H(X)$ и $H(Y/X) = H(Y)$, кількість інформації, що міститься в Y щодо X дорівнює нулю

$$I(Y, X) = H(Y) - H(Y / X) = 0, \tag{4.23}$$

тобто. повідомлення, що характеризується станами Y, не містять жодної інформації про повідомлення, що характеризуються станами X.

У разі незалежності окремих станів елементів повідомлень ентропія на виході лінії

$$H(Y) = -\sum_{j=1}^{m} p(y_i) \log_2 p(y_i), \tag{4.24}$$

при цьому число букв алфавіту $Y = \{y_1, y_2, \ldots y_m\}$ дорівнює числу букв алфавіту $X = \{x_1, x_2, \ldots x_m\}$. Тоді умовна ентропія дорівнює

$$H(Y/X) = -\sum_{i=1}^{m} p(x_i) \sum_{j=1}^{m} p(y_j / x_i) \log_2 p(y_j / x_i). \qquad (4.25)$$

Таким чином, швидкість передачі інформації по дискретному каналу з перешкодами визначається

$$V = V_\tau I(Y, X) = V_\tau (H(X) - H(X/Y)), \qquad (4.26)$$

де

$$I(Y, X) = H(Y) - H(Y/X) =$$
$$= -\sum_{j=1}^{m} p(y_i) \log_2 p(y_i) + \sum_{i=1}^{m} p(x_i) \sum_{j=1}^{m} p(y_j / x_i) \log_2 p(y_j / x_i). \qquad (4.27)$$

Помножуючи у формулі (4.27) $H(Y)$ на величину $\sum_{i=1}^{m} p(x_i / y_j) = 1$ отримаємо.

$$I(Y, X) = -\sum_{i=1}^{m} \sum_{j=1}^{m} p(y_j) p(x_i / y_j) \log_2 p(y_j) +$$
$$+ \sum_{i=1}^{m} \sum_{j=1}^{m} p(x_i) p(y_j / x_i) \log_2 p(y_j / x_i) \qquad (4.28)$$

Сумісний розподіл ймовірностей вхідного та вихідного сигналів:
$p(x_i y_j) = p(x_i) p(y_j / x_i) = p(y_j) p(x_i / y_j)$.

Тоді перетворимо до виду

$$I(Y, X) = -\sum_{i=1}^{m} \sum_{j=1}^{m} p(x_i y_j) \log_2 p(y_j) + \sum_{i=1}^{m} \sum_{j=1}^{m} p(x_i y_j) \log_2 p(y_j / x_i) =$$
$$= \sum_{i=1}^{m} \sum_{j=1}^{m} p(x_i y_j) \log_2 \frac{p(y_j / x_i)}{p(y_j)} = \sum_{i=1}^{m} \sum_{j=1}^{m} p(x_i y_j) \log_2 \frac{p(x_i y_j)}{p(x_i) p(y_j)}. \qquad (4.29)$$

Таким чином, швидкість передачі інформації по дискретному каналу з перешкодами

$$V = V_\tau \sum_{i=1}^{m} \sum_{j=1}^{m} p(x_i y_j) \log_2 \frac{p(x_i y_j)}{p(x_i) p(y_j)}. \qquad (4.30)$$

Пропускна здатність дискретного каналу із перешкодами, тобто. максимально можлива швидкість передачі інформації, обчислюється за формулою

$$C = V_\tau \max I(Y, X) = V_\tau \max(H(X) - H(X/Y)), \qquad (4.31)$$

де максимум визначається за всіма можливими розподілами ймовірностей, що характеризує джерело сигналів.

Якщо технічна швидкість V_τ передачі елементів повідомлень фіксована, пропускна здатність може досягатися за рахунок зміни статистичних властивостей символів за допомогою їх перетворення (кодування).

Пропускна здатність деяких типів дискретних каналів.

Симетричний недвійковий канал. Нехай дискретним симетричним каналом передаються m-позиційні символи з ймовірністю помилкового прийому p_0. Оскільки величина p_0 залежить від того, який символ передавався, то можливість отримати символ y_j, якщо передавався символ x_i буде

$$p(y_j / x_i) = \begin{cases} 1 - p_0 & при \quad i = j \\ \dfrac{p_0}{m-1} & при \quad i \neq j \end{cases}. \qquad (4.32)$$

Пропускна здатність каналу визначається формулою (4.31). Виконавши деякі перетворення, визначимо пропускну здатність дискретного симетричного каналу

$$C = V_\tau (\log_2 m + (1 - p_0)\log_2 (1 - p_0) + p_0 \log_2 \frac{p_0}{m-1}). \qquad (4.33)$$

Симетричний двійковий канал

Двійкові джерела характеризуються передачею лише двох можливих повідомлень, ймовірність помилки складає p_0 , ймовірність вірного прийому $1-p_0$. Тоді пропускна здатність з (4.33) визначається при незалежних та рівно ймовірних вхідних повідомленнях як

$$C = V_\tau (1 + (1 - p_0)\log_2 (1 - p_0) + p_0 \log_2 p_0)\ldots \qquad (4.34)$$

Приклад 4.3. По дискретному каналу зв’язку с завадами передаються двійкові кодові повідомлення. Ймовірність спотворення одиночних сигналів $p_0=10^{-2}$, а тривалість елемента коду 10^{-3} с. Визначити пропускну здатність каналу зв'язку.

Рішення. Згідно з (4.34) пропускна здатність каналу зв’язку буде складати.

$$C = V_\tau (1 + (1 - p_0)\log_2 (1 - p_0) + p_0 \log_2 p_0) = 10^3 (1 + 0{,}99 \cdot \log_2 0{,}99 + \\ + 0{,}01 \cdot \log_2 0{,}01) = 1000 \cdot (1 - 0{,}066 - 0{,}014) = 1000 \cdot 0{,}092 = 920 \textit{біт/сек}.$$.■

Двійковий канал з помилками та стиранням.

Кількість інформації, що передається по двійковому каналу з помилками та стиранням

$$I(X,Y) = H(Y) + \sum_{k=1}^{3} p_k \log_2 p_k , \qquad (4.35)$$

де $p_1 = 1-p-p_s$; $p_2 = p_0$; $p_3 = p_s$.

Можна показати (застосовуючи метод Лагранжа), що максимум ентропії $H(Y)$, відповідний максимуму кількості інформації $I(X,Y)$, буде при симетричному вході каналу. При цьому ймовірності вихідних символів будуть $p(y_1)=p(y_2)=0.5(1-p_S)$ и $p(y_S)=p_S$.

Тоді пропускна спроможність каналу

$$C = V_\tau [(1 - p_S) + p_0 \cdot \log_2 p_0 - (1 - p_S) \cdot \log_2 (1 - p_S) + \\ + (1 - p_0 - p_S) \cdot \log_2 (1 - p_0 - p_S)] . \qquad (4.36)$$

За відсутності перешкод $p_0=0$ і символи y_1 и y_2 можуть з'явитися тільки під час передачі символів x_1 и x_2 відповідно. Однак при цьому з ймовірністю p_S може відбутися стирання символів та пропускна спроможність каналу

$$C = V_\tau [(1 - p_S)]. \qquad (4.37)$$

тобто визначатиметься лише ймовірністю стирання символів.

Приклад 4.4. По дискретному каналу зв’язку с завадами та стиранням передаються двійкові кодові повідомлення. Ймовірність спотворення одиночних сигналів $p=p_{10}=p_{01}=10^{-2}$, ймовірність стирання

$p=p_{10}=p_{01}=2\cdot10^{-2}$, а тривалість елемента коду 10^{-3} с. Визначити пропускну здатність каналу зв'язку.

Рішення. Згідно з (4.36) пропускна здатність каналу зв’язку буде складати.

$$C = V_\tau[(1-p_S) + p_0 \cdot \log_2 p_0 - (1-p_S)\cdot\log_2(1-p_S) +$$
$$+ (1-p_0-p_S)\cdot\log_2(1-p_0-p_S)] = 10^3 \cdot (0{,}98 + 0{,}01\cdot\log_2 0{,}01 - 0{,}98\cdot\log_2 0{,}98 + \cdot \blacksquare$$
$$+ 0{,}985\cdot\log_2 0{,}985) = 1000\cdot(0{,}98 - 0{,}0664 + 0{,}0290 - 0{,}0426) = 900 \textit{біт/сек}.$$

4.3 Швидкість передачі та пропускна здатність неперервного каналу зв’язку с завадами та без завад

Неперервним каналом називається канал, на вхід якого подається і з виходу знімається неперервний сигнал. Канал вважається заданим, якщо відомі статистичні дані про повідомлення на його вході та виході та обмеження, що накладаються на вхідні повідомлення фізичними характеристиками каналу. При розгляді інформаційних характеристик неперервного каналу (швидкості передачі, пропускної здатності, коефіцієнта використання) застосовують модель реального каналу, що називається *гаусовим каналом* , для якого виконуються такі умови:

1. Фізичні параметри відомі та детерміновані.

2. Смуга пропускання каналу обмежена частотою f_c герц.

3. У каналі діє адитивний білий гауссів шум (АБГШ) (з рівномірним частотним спектром і нормальним розподілом амплітуд).

4. Статистичний зв'язок між сигналом і шумом відсутній, а ширина спектра сигналу та перешкоди обмежена смугою пропускання каналу.

Гауссовими каналами, наприклад, є канали супутникового зв'язку, проводові лінії при незначних повадах та однопроменеві радіоканали без завмирань.

Припустимо по вказаному гауссовому каналу (рис. 4.1) передається неперервний сигнал $x(t)$ із середньою потужністю $P_x = \sigma_x^2$. На виході каналу фіксується сигнал $y(t)$, який спотворений адитивним гаусовим шумом $\xi(t)$ із середньою потужністю $P_\xi = \sigma_\xi^2$.

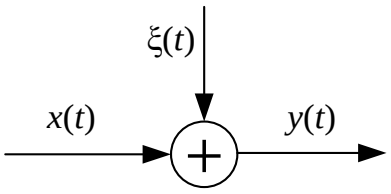


Рисунок 4.5 – Схема неперервного каналу зв'язку

Будемо вважати, що тривалість T сигналу, що передається, досить велика, так що можна замінити неперервні реалізації $x(t)$ і $y(t)$ послідовностями з $N=2f_k \cdot T$ відліків, взятих через інтервали $T_д=(2f_c)^{-1}$ де f_c– верхня межа смуги пропускання. При цьому для простоти вважаємо, що нижня межа дорівнює нулю, тобто $\Delta f_c = f_c$ і враховуємо, що згідно теоремі відліків частота дискретизації $f_д \geq 2f_c$. Тоді середня кількість інформації, що припадає на відлік сигналу, що передається по каналу

$$I(Y,X) = H_\Delta(Y) - H_\Delta(Y/X), \tag{4.38}$$

де $H_\Delta(Y)$ та $H_\Delta(Y/X)$ – диференціальна ентропія відліку сигналу $x(t)$ та умовна диференціальна ентропія відліку сигналу $y(t)$ відповідно.

Таким чином, за час T каналом передається кількість інформації

$$I(Y,X) = N \cdot [H_\Delta(Y) - H_\Delta(Y/X)], \tag{4.39}$$

тоді швидкість передачі інформації

$$V(Y,X) = \frac{N}{T}\left[H_\Delta(Y) - H_\Delta(Y/X)\right], \tag{4.40}$$

де $\frac{N}{T} = \frac{1}{T_Д} = f_Д$ – частота дискретизації сигналу .

Остаточно отримаємо швидкість передачі в неперервному каналі

$$V(Y,X) = 2f_c\left[H_\Delta(Y) - H_\Delta(Y/X)\right], \tag{4.41}$$

Оскільки у гауссовому каналі діє перешкода – білий гаусівський шум, який має нормальний розподіл, то диференціальна ентропія сигналу/шуму

$$H_{\Delta}(Y / X) = H_{\Delta}(\xi), \tag{4.42}$$

де $H(\xi)$ – ентропія N-вимірного випадкового вектора перешкоди, компонентами якого є випадкові величини у відповідних перерізах неперервного адитивного гауссового білого шуму $\xi(t)$.

Оскільки значення білого шуму в моменти відліків некорельовані

$$H_{\Delta}(\xi) = N \cdot h(\xi) = 2 f_c \cdot T \cdot h(\xi), \tag{4.43}$$

де $h(\xi)$ – диференціальна ентропія в середньому на один відлік. В даному випадку, оскільки шум розподілений за нормальним законом, то відповідно до прикладів (2.20), (2.23) диференціальна ентропія визначається як

$$h(\xi) = \log_2 \sqrt{2\pi e \sigma_{\xi}^2}\,. \tag{4.44}$$

Таким чином, отримуємо наступний вираз для середньої кількості інформації, що передається по неперервному каналу зі смугою пропускання Δf_k.

$$V(Y, X) = 2\Delta f_c (H_{\Delta}(Y) - \log_2 \sqrt{2\pi e \sigma_{\xi}^2})\,. \tag{4.45}$$

Пропускною здатністю неперервного каналу називається максимально досяжна швидкість передачі інформації $\max\{V(Y, X)\}$

$$C = \max\{V(Y, X)\} = 2\Delta f_c (H_{\Delta}(Y) - \log_2 \sqrt{2\pi e \sigma_{\xi}^2})\,. \tag{4.46}$$

Оскільки σ_{ξ} задано, то максимальне значення виразу (4.46) буде забезпечено за умови максимізації диференціальної ентропії вихідного сигналу $H_{\Delta}(Y)$. Середні потужності вхідного сигналу $x(t)$ і перешкоди $\xi(t)$ обмежені, тому середня потужність вихідного сигналу $y(t)$ також обмежена. За такої умови величина диференціальної ентропії $H_{\Delta}(Y)$ буде

максимальною у разі, якщо $y(t)$ характеризується нормальним законом розподілу. Якщо сумарний сигнал $y(t)$ та одна з його складових ξ(t) розподілені за нормальним законом, те й друга складова, тобто. вхідний сигнал $x(t)$ також повинен мати нормальний закон розподілу. Таким чином, диференціальна ентропія вихідного сигналу

$$H_{\Delta}(Y) = \log_2 \sqrt{2\pi e \sigma_x^2} = \log_2 \sqrt{(\sigma_x^2 + \sigma_\xi^2) 2\pi e}\,. \qquad (4.47)$$

Підставляючи (4.46) у (4.47) остаточно отримаємо

$$C = 2\Delta f_c [\log_2 \sqrt{(\sigma_x^2 + \sigma_\xi^2) 2\pi e} - \log_2 \sqrt{\sigma_\xi^2 2\pi e}] = \Delta f_c \log_2 \frac{\sigma_x^2 + \sigma_\xi^2}{\sigma_\xi^2}. \qquad (4.48)$$

Таким чином, для неперервного каналу формула (4.48) переходить у відому формулу Шеннона для пропускної здатності гауссового неперервного каналу з флуктуаційною перешкодою

$$C = \Delta f_c \log_2 \left(1 + \frac{P_x}{P_\xi} \right), \qquad (4.49)$$

де $P_x = \sigma_x^2$ – середня потужність корисного сигналу, $P_\xi = \sigma_\xi^2$ – середня потужність перешкоди.

Як очевидно з (4.49), пропускну здатність каналу можна регулювати шляхом зміни Δf_k або P_x . При цьому залежність пропускної здатності каналу Δf_k при постійній потужності сигналу нелінійна . Це пов'язано з тим, що потужність перешкоди P_ξ також залежить від ширини частотного спектра. Енергетичний спектр білого шуму рівномірний, тому потужність такої перешкоди можна у вигляді

$$P_\xi = P_0 \Delta f_c, \qquad (4.50)$$

де P_0 – потужність, що припадає на смугу в 1 Гц (спектральна щільність потужності перешкоди).

Підставивши (4.50) в (4.49), отримаємо вираз, що визначає характер залежності пропускної здатності каналу від ширини його смуги

пропускання

$$C = \Delta f_c \log_2\left(1 + \frac{P_x}{P_0 \Delta f_c}\right). \qquad (4.51)$$

При розширенні смуги пропускання каналу Δf пропускна здатність збільшується, але прагне до кінцевої межі

$$C_m = \lim_{\Delta f_к \to \infty} C = \lim_{\Delta f_к \to \infty} \Delta f_c \log_2\left(1 + \frac{P_x}{P_0 \Delta f_c}\right) = \frac{P_x}{\ln 2 \cdot P_0} = 1{,}4427 \frac{P_x}{P_0}. \qquad (4.52)$$

Графік залежності пропускної здатності неперервного гауссового каналу зв'язку від ширини смуги пропускання відповідно до (4.52) має вигляд, показаний на рис. 4.6.

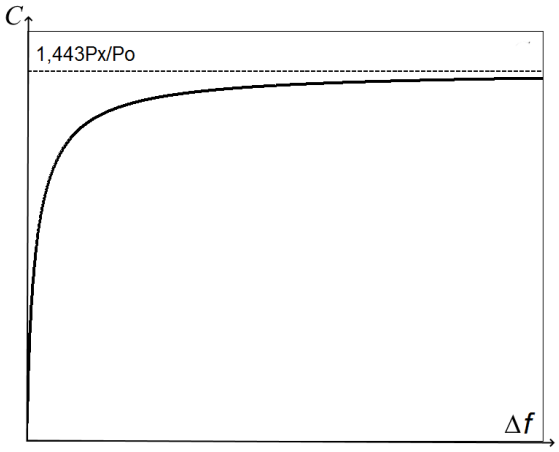


Рисунок 4.6 – Залежність пропускної здатності від смуги пропускання каналу

З рис. 4.5 видно, що зі збільшенням смуги Δf_c величина C росте швидко доти, доки настане рівність $P_x = P_\xi$. При подальшому розширенні смуги каналу величина зростає повільно, асимптотично наближаючись до значення $1{,}443 P_x/P_0$.

Якщо сигнал змішаний із шумом, то амплітуда сигналу може бути виміряна лише з точністю до ефективного значення шуму. Іншими словами, невизначеність оцінки точного значення амплітуди сигналу дорівнює квадратному кореню із середнього квадрата шумової напруги. Як випливає із зазначених вище припущень, зміна вхідного сигналу менше, ніж $\sigma = \sqrt{P_E}$ приймач не розрізняє. Отже, кількість рівнів, яка може бути

помітна без помилок, визначиться з виразу

$$M = \frac{\sqrt{P_x + P_E}}{P_E} = \sqrt{1 + \frac{P_x}{P_E}}\,. \qquad (4.53)$$

Отже, найбільша кількість інформації, що переноситься кожним імпульсом M різних рівнів, дорівнює

$$I = \log_2 \sqrt{1 + \frac{P_x}{P_E}} = \frac{1}{2}\log_2\left(1 + \frac{P_x}{P_E}\right). \qquad (4.54)$$

Приклад 4.5. Визначити пропускну здатність неперервного каналу з перешкодами типу білого шуму Гаусса . Смуга пропускання каналу Δf_k=1000Гц. Дисперсія корисного сигналу $\sigma_x^2 = P_x = 4000B^2$. Спектральна щільність потужності перешкоди $P_0 = 0{,}002^2$ /Гц.

Рішення. Пропускна здатність неперервного сигналу з перешкодами згідно (4.51) $C = \Delta f_c \log_2\left(1 + \frac{P_x}{P_\xi}\right)$. Якщо в каналі діє білий гауссів шум, то відповідно до (4.50) потужність перешкоди $P_\xi = P_0 \Delta f_k$.

Тоді отримаємо

$$C = \Delta f_k \log_2\left(1 + \frac{P_x}{P_0 \Delta f_k}\right) = 1000\log_2\left(1 + \frac{4\cdot 10^3}{2\cdot 10^{-3}\cdot 10^3}\right) = 1000\log_2 2001 =$$

= 1096,65 біт/С. ■

Приклад 4.6. Відповідно до рекомендацій Міжнародного консультативного комітету з телефонії та телеграфії (МККТТ) як стандартний телефонний канал приймається канал тональної частоти (ТЧ) з ефективною смугою частот 300–3400 Гц. Відношення сигнал/шум (SNR) за потужністю становить 31 дБ. Визначити пропускну здатність каналу .

Рішення. Перекладаємо децибели у відносні одиниці. Відношення двох значень енергетичної величини, таких як потужність та енергія , виражене в децибелах, визначається за такою формулою: $D_P = 10\lg\frac{P_x}{P_\xi}$.

Тоді відношення сигнал/шум дорівнює $\frac{P_x}{P_\xi} = 10^{\frac{D_P}{10}} = 10^{3{,}1} = 1258{,}93.$

Смуга частот каналу $\Delta f_c = f_{\max} - f_{\min} = 3400 - 300 = 3100$ Гц.

Пропускна здатність неперервного каналу з перешкодами згідно (7.12)

$$C = \Delta f_c \log_2\left(1 + \frac{P_x}{P_\xi}\right) = 3100 \cdot \log_2(1 + 1258{,}93) = 3100 \cdot 10{,}2991 = 31927{,}21 \text{ біт/С}. \blacksquare$$

Приклад 4.7. Визначити необхідну пропускну здатність та смугу пропускання телевізійного каналу. Розмір кадру 1280×720 пікселів, частота кадрів $f_{\text{кадр}} = 50$ Гц. Середня ентропія одного пікселя $H_{\text{п}} = 1{,}2$ біта, необхідне відношення сигнал/шум $S = 22$ дБ.

Рішення. Відношення сигнал/шум у відносних одиницях $\frac{P_x}{P_\xi} = 10^{\frac{D_P}{10}} = 10^{2{,}2} = 158{,}489$.

Число пікселів у зображенні:

$N_{\text{пікс}} = N_1 \cdot N_2 = 1280 \cdot 720 = 921600$ пікселів.

Оскільки за одну секунду має бути передано 50 кадрів, то необхідна пропускна здатність каналу:

$C = N_{\text{пікс}} \cdot f_{\text{кадр}} \cdot H_{\text{п}} = 921600 \cdot 50 \cdot 1{,}2 = 55296000$ біт/ С = 52,734 Мбіт/С.

Відповідно до (7.12) пропускна здатність каналу становить. $C = \Delta f_c \log_2\left(1 + \frac{P_x}{P_\xi}\right)$.

Тоді потрібна смуга частот каналу

$$\Delta f_c = \frac{C}{\log_2\left(1 + \frac{P_x}{P_\xi}\right)} = \frac{55296000}{\log_2 159{,}489} = \frac{5529600}{7{,}3173} = 7556886 \text{ Гц} = 7{,}557 \text{ МГц}. \blacksquare$$

Приклад 4.8 Визначити час, необхідний для передачі по каналу, записаного фрагмента оцифрованого сигналу. Тривалість фрагмента п'ять секунд, частота дискретизації $f_{\text{д}} = 8$ КГц, число бітів на відлік $n = 12$. Ширина лінії пропускання каналу $\Delta f_c = 10000$ Гц. Дисперсія корисного сигналу $\sigma_x^2 = P_x = 25B^2$. Спектральна густина потужності перешкоди $P_0 = 0{,}0004^2$ /Гц .

Рішення. Кількість інформації у фрагменті сигналу

$V = n \cdot f_{\text{д}} \cdot T$ біт $= 12 \cdot 8000 \cdot 5 = 480000$ біт.

Пропускна здатність каналу з перешкодами

$$C = \Delta f_c \log_2\left(1+\frac{P_x}{P_0 \Delta f_k}\right) = 10000 \log_2\left(1+\frac{25}{4\cdot 10^{-4}\cdot 10^4\cdot}\right) = 10000 \log_2 7{,}25 =$$

= 28 580 біт/С.

Час передачі сигналу $T = V / C$ = 480000/28580 = 16,795 С. ■

Приклад 4.9. Ширина лінії пропускання АБГШ каналу Δf_c = 1,5 КГц. Виражене децибелах відношення сигнал/шум D_p змінюється з 5 до 40 дБ з кроком 5 дБ.

Розрахувати зміну пропускної здатності каналу та побудувати графік залежності пропускної здатності каналу від відношення сигнал/шум $C=f(P_x/P_\xi)$.

Рішення. Пропускна здатність каналу становить $C = \Delta f_c \log_2\left(1+\frac{P_x}{P_\xi}\right)$.

Відношення сигнал/шум у відносних одиницях $\frac{P_x}{P_\xi} = 10^{\frac{D_P}{10}}$.

Таким чином, пропускна здатність $C = \Delta f_c \log_2\left(1+10^{\frac{D_P}{10}}\right)$.

Підставимо в $D_P = 10\lg\frac{P_x}{P_\xi}$ значення D_p = 5…25 дБ та заповнимо таблицю.

Таблиця 4.1 – Залежність пропускної здатності каналу від відношення сигнал/шум

D_p	5	10	15	20	25	30	35	40
C	1993	4485	6976	9467	11959	14450	16942	19433

Побудуємо графік за даними табл. 4.1

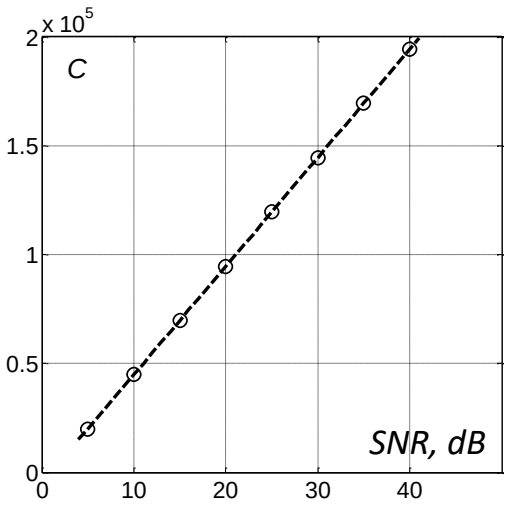


Рисунок 4.7 – Графік залежності пропускної здатності каналу від відношення сигнал/шум

Завдання для самостійного вирішення

1. В інформаційному каналі використовується алфавіт із чотирма різними символами. Тривалість всіх символів однакова і дорівнює $\tau = 1$ мс. Визначити пропускну здатність каналу за відсутності шумів.

2. В інформаційному каналі використовується код, за якого забороняється передача поспіль двох однакових символів. Алфавітний код складається з чотирьох різних символів. Можливості передачі всіх дозволених пар символів однакові. Тривалості всіх символів також однакові і дорівнюють $\tau = 1$ мс. Визначити швидкість передачі.

3. У дискретному каналі передачі використовуються три різних символи з тривалостями $\tau_1 = \tau_2 = 10$ мс і $\tau_3 = 20$ мс. Визначити пропускну спроможність каналу.

4. У канал зв'язку передаються повідомлення довжиною $n = 10$ елементів, кожен з яких може приймати $m = 4$ стани з ймовірностями $P_1 = 0{,}2$; $P_2 = 0{,}3$; $P_3 = 0{,}1$; $P_4 = 0{,}4$. Час передачі одного повідомлення $\tau = 0{,}1$ с. Визначити швидкість передачі інформації та пропускну здатність каналу зв'язку.

5. По бінарному каналу передаються повідомлення: 1110011101, 1110000001. Тривалість кожного елемента повідомлення $\tau = 10$ мс. Визначити швидкість передачі кожного повідомлення та пропускну здатність двійкового каналу.

6. У канал зв'язку передаються повідомлення від джерела ергодичного, що виробляє $m = 3$ елемента. У кодових комбінаціях заборонено передачу двох однакових елементів. Імовірність передачі всіх дозволених кодових комбінацій завдовжки $n = 3$ однакова. Тривалість кожного елемента $\tau = 10$ мс. Визначити швидкість передачі інформації та пропускну здатність каналу зв'язку.

7. У бінарному каналі ймовірності придушення та відтворення хибного сигналу однакові та рівні $P_{10} = P_{01} = P = 10^{-3}$. Тривалість символів однакова і дорівнює $\tau = 1$ мс. Визначити пропускну здатність бінарного симетричного каналу.

8. По лінії зв'язку з перешкодами передається чотири повідомлення. Ансамбль об'єднання описується таблицею.

	X_1	X_2	X_3	X_4
Y_1	0,1	0,05	0,05	0,15
Y_2	0,03	0,05	0,1	0,04
Y_3	0,07	0,03	0,05	0,06
Y_4	0	0,07	0,05	0,1

Тривалість повідомлення $\tau = 2$ мс. Визначити швидкість передачі повідомлень та пропускну здатність каналу зв'язку.

9. По дискретному каналу зв'язку з перешкодами передається кодове повідомлення 1100110011. Імовірність спотворення одиночних символів $P_{10} = P_{01} = P = 10^{-1}$, а тривалість елемента коду $\tau = 1 \cdot 10^{-2}$ с. Визначити швидкість передачі та пропускну здатність каналу зв'язку.

10. З контрольованого пункту передаються повідомлення про зміну становища об'єктів. Кожен об'єкт може перебувати в одному з двох положень «увімкнений» або «вимкнений». Спостереженням встановлено, що з 50 переданих повідомлень 40 відноситься до першого об'єкта, 2 – до другого і 8 –до третього. Об'єкти працюють незалежно один від одного, а положення об'єктів є рівноймовірними. Визначити швидкість передачі інформації та пропускну здатність дискретного каналу, якщо тривалість кожного повідомлення 1 мс.

11. Кількість повідомлень, що передаються з контрольованого пункту, про стан трьох об'єктів, однакова. Спостереженням встановлено, що у середньому об'єкт 1 «включено» 60%, об'єкт 2 – 30%, а об'єкт 3 –

40%. Решту часу об'єкти «відключені». Визначити швидкість передачі інформації та пропускну здатність каналу зв'язку, якщо тривалість одного повідомлення 5 мс.

12. Визначити обсяг інформації, що міститься у зображенні з 500 рядків по 500 елементів у кожному рядку. Яскравість кожного елемента передається вісьмома квантованими рівнями. Різні градації яскравості рівноймовірні, а яскравості різних елементів не корельовані.

13. Зображення задачі 12 має бути передане радіолінією, на вході якої діє білий гаусівський шум з питомою потужністю $P_{ОШ} = 10^{-4}$ Вт/Гц. Ширина смуги пропускання приймального пристрою $\Delta F_C = 1000$ Гц. Час передачі 1 час. Визначити мінімально можливе значення потужності корисного сигналу на вході приймача.

14. Визначити довжину магнітної стрічки для запису одного зображення, якщо ентропія зображення $H(X) = 600000$ дв.од./зобр., а шириною стрічки записується 30 дв.ед. інформації при густині запису 10 дв.од./мм.

15. Сигнал з амплітудою 1 В передається по каналу зв'язку, в якому від носіння сигнал/шум дорівнює 10 дБ. Визначити абсолютну похибку в тілі вимірювань.

16. По безперервному каналу передається сигнал, спектр якого обмежений смугою частот 30 Гц. Визначити пропускну здатність каналу зв'язку таким чином, щоб похибка сигналу, що передається, не перевищувала 1%.

17. Безперервний канал зв'язку з пропускною здатністю 40 дв.од./с призначений передачі квантованого сигналу зі смугою частот 5 Гц. Визначити кількість різних рівнів вимірюваного сигналу та похибку вимірювань.

18. Радіолінією, на вході якої діє гаусівський шум з питомою потужністю 10-8 Вт/Гц, передається 1024 повідомлення протягом $1 \cdot 10^{-1}$с. Визначити мінімальну потужність корисного сигналу на вході приймача, якщо смуга пропускання приймача дорівнює 100 Гц.

19. Відношення сигнал/шум в лінії зв'язку дорівнює 10^{-1}, а смуга пропускання каналу зв'язку 1 кГц. Визначити пропускну спроможність каналу зв'язку.

20. Визначити пропускну здатність каналу зв'язку за умови, що сигнал $\sin 500\pi t$ має бути відновлений з похибкою не більшою ніж 0,57 В.

ГЛАВА 5

ПРИНЦИПИ ЗАВАДОСТІЙКОГО КОДУВАННЯ

5.1 Види та основні параметри коригуючих кодів

При передачі інформації каналами зв'язку та її зберіганні на носіях повідомлення, що передаються або зберігаються, неминуче піддаються впливу перешкод. При цьому частина символів може спотворюватися, що неминуче призводить до зниження достовірності прийому та загалом до зниження надійності та ефективності інформаційних систем.

Вирішення цієї проблеми, тобто підвищення достовірності передачі в каналах зв'язку з перешкодами, здійснюється за допомогою кодів, що дозволяють виявляти або (та) виправляти помилки. Такі коди називаються *завадостійкими* або *коригуючими.*

Теоретичні основи завадостійкого кодування повідомлень базуються на основній теоремі Шеннона про кодування каналу з перешкодами. Ця теорема говорить.

За будь-якої продуктивності джерела повідомлень, меншою ніж пропускна, здатність каналу, існує такий спосіб кодування, який дозволяє забезпечити передачу всієї інформації, створюваної джерелом повідомлень, з якою завгодно малою ймовірністю помилки. Якщо ж продуктивність джерела повідомлень більше пропускної спроможності каналу, способу кодування, що дозволяє вести передачу інформації з якою завгодно малою ймовірністю помилки, не існує.

Загальним принципом побудови кодів, що виправляють помилки, є *надмірність*. При запису (передачі) до корисні даних додаються спеціальним чином обчислені додаткові (надлишкові) *перевірочні символи*. Під час читання (прийому) ці надлишкові символи використовуються для виявлення та виправлення помилок. Надлишкові символи залежать від декількох інформаційних символів, тобто інформація, що міститься в кодовій послідовності X, перерозподіляється також і на перевірочні символи.

За способом роботи з даними коди, що виправляють помилки, поділяються на *згорткові*, що працюють з послідовностями символів як з

безперервним потоком і *блокові* (блочні), що поділяють інформацію на фрагменти постійної довжини та обробляють кожен з них окремо.

При використанні блокових кодів *кодер каналу* за допомогою заданого алгоритму кодування перетворює блоки з *k інформаційних символів* шляхом додавання перевірочних *r* символів на більш довгі двійкові послідовності, що складаються з *n* символів і звуться *кодовими словами*. $r = n–k$ надлишкових символів, що додаються до кожного інформаційного блоку кодером, не несуть жодної додаткової інформації, і використовуються лише для виявлення або виправлення помилок, що виникають у процесі передачі.

Блоковий код називається *рівномірним*, якщо значення *n* залишається постійним для всіх повідомлень. В іншому випадку код називається *нерівномірним*.

Залежно від методів внесення надмірності блокові коди поділяються на *роздільні* та *нероздільні*. При кодуванні роздільними кодами у вихідних послідовностях символів можна розмежувати інформаційні символи і перевірочні, тобто код містить постійну інформаційну частину довжиною *k* символів та надмірну (перевіркову) довжиною $r=n–k$ символів. У нероздільних кодах таке розмежування неможливе.

Роздільні коди, у свою чергу, поділяються на *систематичні* та *несистематичні*. Систематичними називаються коди, в яких контрольні (перевірочні) елементи є лінійними комбінаціями інформаційних елементів. Несистематичні коди такої властивісті не мають.

Параметри коригуючих кодів

1. Основа коду *m* – кількість різних символів, що використовуються для надсилання повідомлень. Для двійкових кодів $m=2$.

2. Потужність коду N_p – число кодових комбінацій, що використовуються при передачі повідомлень

$$N_p = 2^k. \qquad (5.1)$$

3. Повне число кодових комбінацій *N* – число всіх можливих комбінацій

$$N = 2^n. \qquad (5.2)$$

4. Відносна надмірність коду R – відношення числа перевірочних символів до довжини коду

$$R = \frac{r}{n} = \frac{n-k}{n} = 1 - \frac{k}{n} \quad . \tag{5.3}$$

У загальному випадку відносну надмірність розраховують за формуло

$$R = 1 - \frac{\log_2 N_p}{\log_2 N} . \tag{5.4}$$

5. Швидкість коду V – відношення числа інформаційних символів до довжини коду

$$V = k/n = 1 - R. \tag{5.5}$$

6. Вага кодової комбінації w – кількість одиниць у кодовій комбінації.

7. Кодова відстань (відстань Гемінґа) d – число однойменних розрядів двох кодових комбінацій, у яких значення символів не збігаються. Для визначення кодової відстані необхідно скласти ці комбінації за модулем два і знайти вагу w отриманого результату.

8. Мінімальна кодова відстань $d_{\min}$ – мінімальна кількість символів, якими будь-яка кодова комбінація відрізняється від іншої (по всіх парах кодових слів). Щоб отримати $d_{\min}$, необхідно знайти кодову відстань для всіх можливих поєднань кодових векторів і вибрати мінімальне значення.

Приклад 5.1. Для кодових комбінацій x_1={10010111} та x_2={11011011} визначити вагу кожної з комбінацій та кодову відстань між ними.

Рішення. Порахуємо число одиниць у кожній із комбінацій

$w_1 = 5$, $w_2 = 6$.

Для визначення кодової відстані складемо дві кодові комбінації за модулем два і порахуємо число одиниць у результаті.

x_{12}=10010111 ⊕ 11011011 = 01001100

(⊕ – символ додавання за модулем два, 0⊕0=0, 0⊕1=1, 1⊕0=1, 1⊕1=0).

$d = w_{12} = 3$. ■

Приклад 5.2. Визначити мінімальну кодову відстань для коду, що складається із комбінацій {1100, 1000, 1011, 1101}.

Рішення. Зведемо в таблицю кодові відстані між кожною парою з усіх можливих комбінацій

Кодове слово	1100	1000	1011	1101
1100	0	1	3	1
1000	1	0	2	2
1011	3	2	0	2
1101	1	2	2	0

Очевидно, що мінімальна кодова відстань кодових комбінацій $d_{\min}=1$. ■

Приклад 5.3. Визначити мінімальну кодову відстань для коду, що складається з комбінацій {000, 011, 101, 110}.

Рішення. Зведемо в таблицю кодові відстані між кожною парою з усіх можливих комбінацій

Кодове слово	000	011	101	110
000	0	2	2	2
011	2	0	2	2
101	2	2	0	2
110	2	2	2	0

Для цього коду мінімальна кодова відстань становить $d_{\min}=2$. ■

Мінімальна відстань Гемінга є найважливішою характеристикою коду, оскільки визначає його корегуючу здатність. Справді, нехай має місце код, що містить усі можливі трибітові комбінації: {000, 001, 010, 011, 100, 101, 110, 111}. В цьому випадку $d_{\min}=1$, тобто одиночна помилка переводить дозволену комбінацію в дозволену. Це випадок ненадлишкового коду, що не має коригуючої здатності.

Розглянемо тепер код, що включає дозволені комбінації {000, 011, 101, 110}. Згідно з результатами прикладу 5.3, мінімальна кодова відстань для цього коду $d_{\min}=2$. Тоді комбінації {001, 010, 100, 111} складуть

підмножину заборонених кодових комбінацій. Аналізуючи виділені підмножини, можна бачити, що в даному випадку одноразова помилка завжди переводить комбінацію з підмножини дозволених у підмножину заборонених. Таким чином, при $d_{\min}=2$ код здатний виявляти всі поодинокі помилки.

Надмірність коду згідно (5.4) складе

$$R = 1 - \frac{\log_2 N_p}{\log_2 N} = 1 - \frac{\log_2 4}{\log_2 8} = 1 - \frac{2}{3} = \frac{1}{3} \approx 0{,}333.$$

У загальному випадку при необхідності виявляти помилки кратності t_d кодове відстань коду має бути більше t_d принаймні на 1

$$d_{\min} \geq t_d + 1. \qquad (5.6)$$

Виправляти помилки код з $d_{\min}=2$ не може, оскільки та ж сама заборонена комбінація в даному випадку може бути отримана при одноразовій помилці з різних дозволених комбінацій. Наприклад, комбінація 001 може бути отримана при спотворенні як третього біта комбінації 000, так і другого біта комбінації 011 або першого комбінації 101.

Для виправлення одиночної помилки кожній дозволеній кодовій комбінації необхідно зіставити підмножини заборонених комбінацій. Щоб ці підмножини не перетиналися, Гемінґова відстань коду, що застосовується, повинна бути не менше трьох. Наприклад, якщо в якості дозволених обрані комбінації 000 та 111, то за наявності одиночної помилки в комбінації 000 вона може перейти {001, 010, 100}. При наявності ж помилки в 111 комбінація може перейти в {011, 101, 110}.

Приймач здійснює декодування таким чином, що прийнята кодова комбінація ототожнюється з дозволеною, яка відрізняється від отриманої в найменшій кількості символів. Таке декодування називається декодуванням *методом максимальної правдоподібності*. Так, у разі прийому однієї з комбінацій з підмножини {001, 010, 100} приймається рішення, що переданий код 000, а якщо прийнято одну з комбінацій {011, 101, 110}, то код 111. Таким чином, код з $d_{\min}=3$ здатний виправляти одноразову помилку.

У загальному випадку для забезпечення можливості виправлення всіх помилок кратності до t_c включно кожна з помилок повинна

призводити до забороненої комбінації, що належить до підмножини вихідної дозволеної кодової комбінації.

Підмножина кожної з дозволених n-розрядних комбінацій складається із заборонених комбінацій, що є наслідком впливу:

1. Одиничних помилок (розташовуються на сфері радіусом d=1 та їх число дорівнює C_n^1.

2. Подвійних помилок (розташовуються у сфері радіусом d=2, їх число дорівнює C_n^2 тощо.

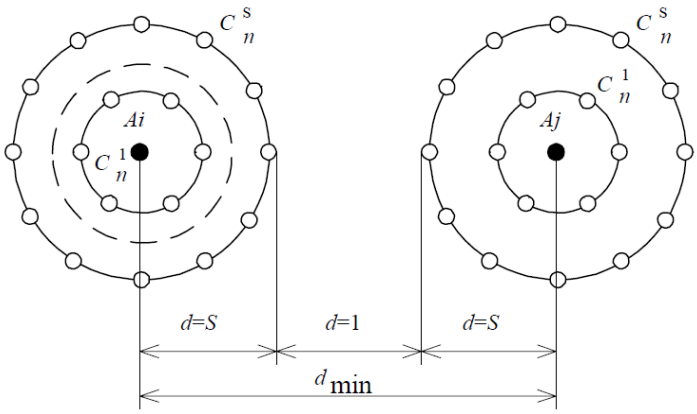


Рисунок 5.1 – Співвідношення між коригуючою здатністю та кодовою відстанню

Зовнішня сфера підмножини має радіус d=S і містить C_n^S заборонених кодових комбінацій. Кратність помилок, що виправляються, t_c буде співпадати з радіусом зовнішньої сфери S. Оскільки зазначені підмножини не повинні перетинатися, мінімальна кодова відстань між дозволеними комбінаціями повинна задовольняти співвідношенню

$$d_{\min} \geq 2 \cdot S+1 = 2 \cdot t_c + 1. \qquad (5.7)$$

Для виправлення всіх помилок кратності до t_c та одночасного виявлення всіх помилок кратності не більше t_d кодова відстань повинна задовольняти умові

$$d_{\min} \geq t_d +\cdot\ t_c + 1. \qquad (5.8)$$

Коригувальні можливості коду в залежності від кодової відстані зведені в табл. 5.1

Таблиця 5.1 – Коригувальні можливості коду

$d_{\min}$	t_d	t_c	Можливість, що забезпечується кодом
1	0	0	Відрізняти одну комбінацію від іншої
2	1	0	Виявляти одноразові помилки
3	1		Виправляти одноразові помилки
	2	0	Виявляти дворазові помилки
4	2	1	Виправляти одноразові та виявляти дворазові помилки
	3	0	Виявляти триразові помилки

Питання мінімально необхідної надмірності, за якої код забезпечує необхідні коригуючі властивості, одне із найважливіших у теорії кодування. Було отримано цілу низку меж, що пов'язують мінімальну кількість надлишкових символів r і кодову відстань $d_{\min}$.

Межа Гемінґа

$$r \geq \log_2\left(1+\sum_{i=1}^{t} C_n^i\right), \tag{5.9}$$

де n – загальна кількість символів у кодовому слові, $t=(d-1)/2$.

Якщо в (5.9) має місце точна рівність, код називається *досконалим*.

Приклад 5.4. Визначити мінімально можливу кількість надлишкових символів r згідно з межею Гемінґа для випадків:

$d=3$, $n=7$; $d=5$, $n=14$;

Рішення. Нехай $d=3$, $n=7$. Тоді $t=(3-1)/2=1$.

$$r_{\min} = \log_2\left(1+\sum_{i=1}^{t} C_n^i\right) = \log_2(1+C_7^1) = \log_2(1+7) = \log_2 8 = 3.$$

Оскільки $r_{\min}$ – ціле, має місце досконалий код з параметрами $r=3$, $n=7$, $k=7-3=4$.

Нехай $d=5$, $n=14$. Тоді $t=(5-1)/2=2$.

$$r_{\min} = \log_2\left(1+\sum_{i=1}^{2} C_{15}^i\right) = \log_2(1+C_{14}^1+C_{14}^2) = \log_2(1+14+91) =$$

$$= \log_2 106 = 6{,}7279.$$

Округлюємо $r_{\min}$ до цілого значення $r_{\min} = \lceil 6{,}7279 \rceil = 7$.

Таким чином, для побудови коду потрібне виконання умови $r \geq 7$, тоді кількість інформаційних символів не перевищує $k_{\max} \leq 14 - 7 = 7$. ■

Межа Плоткіна

$$r \geq 2d - 2 - \log_2 d. \qquad (5.10)$$

Приклад 5.5. Визначити мінімально можливу кількість надлишкових символів r відповідно до межі Плоткіна для кода з d=5.

Рішення. $r_{\min} = 2 \cdot 5 - 2 - \log_2 5 = 10 - 2 - 2{,}3219 = 5{,}56781$.

Округлюємо $r_{\min}$ до цілого значення $r_{\min} = \lceil 5{,}56781 \rceil = 6$. ■

Межа Варшамова-Гілберта

$$r \geq \log_2 \left(1 + \sum_{i=1}^{d-2} C_n^i \right), \qquad (5.11)$$

Приклад 5.6. Визначити мінімально можливу кількість надлишкових символів r згідно з межею Варшамова-Гілберта для кода з d=5, n =14.

Рішення.

$$r_{\min} = \log_2 \left(1 + \sum_{i=1}^{3} C_{14}^i \right) = \log_2 (1 + C_{14}^1 + C_{14}^2 + C_{14}^3) =$$

$$= \log_2 (1 + 14 + 91 + 364) = \log_2 470 = 8{,}8765.$$

Округлюємо $r_{\min}$ до цілого значення $r_{\min} = \lceil 8{,}8765 \rceil = 9$. ■

В більшості випадків межа Варшамова-Гильберта дає найточніші значення. Так, у розглянутих прикладах для випадку d = 5, n = 14 максимальне, тобто найближче до дійсного значення $r_{\min}$ = 9, дає саме межа Варшамова-Гільберта. Справді, дослідженнями встановлено, що мінімально можливе число перевірочних символів випадку d=5, n =14 становить $r_{\min}$ = 9.

Для кодів з $d_{\min}$=3 отримано точне співвідношення між числом перевірочних символів r і довжиною коду n

$$r \geq \log_2(n+1). \qquad (5.12)$$

У додатку Е та літературі [18] наведено таблиці зі значеннями мінімальної кодової відстані для деяких n і k.

5.2 Матричне представлення лінійних кодів

Лінійними називають блокові коди, в яких перевірочні символи є лінійними комбінаціями інформаційних символів, тобто лінійні коди є систематичними. Найбільш зручним математичним апаратом, що описує процес побудови та декодування лінійних кодів, є апарат лінійної алгебри.

Як було показано вище, основними характеристиками лінійних кодів є кількість інформаційних розрядів k, число перевірочних r та загальна кількість розрядів $n = k + r$.

Кодову комбінація систематичного (n, k) коду можна записати як

$$v = \{a_1, a_2, \ldots, a_k, c_1, c_2, \ldots, c_r\},$$

де a_i – інформаційні символи, c_j – перевірочні символи.

Зазначимо, що тут і далі всі операції кодування та декодування провадяться над кінцевим полем GF(2), тобто за модулем два.

Вихідний безнадлишковий k-розрядний код утворює k-вимірний кодовий простір. Будь-які інформаційні комбінації k-розрядного коду можуть бути представлені як вектори цього простору. Тоді для завдання будь-якого вектора потрібно мати лише базис, що визначає цей простір, а кодові комбінації будуть визначатися лінійними операціями над базисом. Нехай вектори $\vec{g}_1 = [g_{11}, g_{12} \ldots g_{1n}], \vec{g}_2 = [g_{21}, g_{22} \ldots g_{2n}], \ldots \vec{g}_k = [g_{k1}, g_{k2} \ldots g_{kn}]$ утворюють базис лінійного простору C. За визначенням базису, будь-який вектор $\vec{v} \in C$ можна подати у вигляді лінійної комбінації базисних векторів $\vec{v} = c_1\vec{g}_1 + c_2\vec{g}_2 + \ldots + c_k\vec{g}_k$.

Найзручніше проводити формування кодових комбінацій шляхом домноження вектора, складеного з інформаційних розрядів на *породжуючу матрицю* коду

$$\vec{v} = \vec{a}G = [a_1, a_2 \ldots a_k] \cdot \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}, \qquad (5.13)$$

де $\vec{a}$ – вектор-рядок інформаційних символів,

$\vec{v}$ – кодова комбінація, що передається в канал зв'язку,

G – породжуюча матриця.

Як рядки утворюючої матриці можуть бути взяті будь-які n-значні вектори, що відповідають наступним умовам

1. Вектори повинні бути лінійно незалежними.

2. Вектори повинні бути ненульовими.

3. Кожен вектор повинен мати вагу не менше заданої мінімальної кодової відстані коду $w \geq d_{\min}$.

4. Кодова відстань між будь-якими парами рядків має бути не меншою за $d_{\min}$.

Найзручніше записувати породжуючу матрицю G розмірністю $k \times n$ у вигляді інформаційної одиничної матриці A розмірністю $k \times k$ і приписаної до неї праворуч перевірочної підматриці B розмірністю $k \times r$. Перевірочна підматриця формується, таким чином, щоб вага w (число ненульових елементів) кожного рядка було не менше ніж $d_{\min} - 1$. Сума за модулем два будь-якої пари рядків не повинна мати вагу меншу за $d_{\min} - 2$

$$G = [A|B] = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & b_{11} & b_{12} & \ldots & b_{1r} \\ 0 & 1 & 0 & 0 & b_{21} & b_{22} & \ldots & b_{2r} \\ 0 & 0 & 1 & 0 & b_{31} & b_{32} & \ldots & b_{3r} \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 1 & b_{k1} & b_{k2} & \ldots & b_{kr} \end{array}\right]. \qquad (5.14)$$

Розгляд (5.13) і (5.14) показує, що інформаційні розряди коду v, що формується, збігаються з розрядами вихідної послідовності a

$$v_1 = a_1,\ v_2 = a_2, \ldots\ v_k = a_k. \qquad (5.15)$$

Дійсно, згідно (5.13)

$v_1 = a_1 \cdot g_{11} \oplus a_2 \cdot g_{21} \oplus \ldots \oplus a_k \cdot g_{k1} = a_1 \cdot 1 \oplus a_2 \cdot 0 \oplus \ldots \oplus a_k \cdot 0 = a_1$ і т.д.

Відповідно до (5.13) та (5.14) алгоритм формування перевірочних розрядів $c_1, c_2, \ldots c_r$ може бути записаний у вигляді:

$$
\begin{aligned}
c_1 = v_{k+1} &= a_1 \cdot b_{11} \oplus a_2 \cdot b_{21} \oplus \ldots \oplus a_k \cdot b_{k1}; \\
c_2 = v_{k+2} &= a_1 \cdot b_{12} \oplus a_2 \cdot b_{22} \oplus \ldots \oplus a_k \cdot b_{k2}; \\
&\ldots\ldots\ldots\ldots\ldots \\
c_r = v_n &= a_1 \cdot b_{1r} \oplus a_2 \cdot b_{2r} \oplus \ldots \oplus a_k \cdot b_{kr};
\end{aligned}
\qquad (5.16)
$$

Приклад 5.7. Побудувати породжуючу матрицю лінійного коду, здатного виправляти одиночну помилку при передачі 16 повідомлень.

Рішення. Визначимо кількість інформаційних символів у коді. Оскільки потужність коду N_p =16, то згідно з (5.1) $k=\log_2 N_p = \log_2 16=4$.

Щоб забезпечити можливість виправлення одиночних помилок, має дотримуватися умова $d_{\min}=2t_c+1=2\cdot 1+1=3$.

Відповідно до формули (5.12) підбираємо k =4, n =7, r = $n - k$ =3,

$r = 3 = \log_2(7+1)$.

Побудуємо породжуючу матрицю коду. Інформаційна підматриця A буде являти собою одиничну матрицю розмірності 4×4.

$$
A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Перевірочна підматриця B матиме розмірність 4×3. Рядки підматриці повинні містити не менше $d_{\min}-1=3-1=2$ одиниць. Вибираємо із усіх трибітових двійкових слів чотири слова з вагою не менше двох. Виберемо, наприклад, комбінації {011, 101, 110, 111}. Тоді підматриця B матиме вигляд:

$$
B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.
$$

Можна бачити, що кодова відстань між рядками під матриці $d_{\min}-2=3-2=1$ забезпечується. Об'єднуючи підматриці A і B, отримуємо породжуючу матрицю G.

$$
G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.
$$

Для цієї матриці виконується умова $d_{\min} \geq 3$. ■

Приклад 5.8. Побудувати породжуючи матрицю лінійного коду з параметрами $d = 4$, $k = 3$, $r = 5$.

Рішення. Визначаємо $n = r + k = 5+3=8$.

Інформаційна підматриця A буде являти собою одиничну матрицю розмірності 3×3

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Перевірочна підматриця B матиме розмірність 3×5. Рядки підматриці повинні містити не менше $d_{\min}–1=4–1=3$ одиниць. Вибираємо зі всіх п'ятибітових комбінацій слова з вагою не менше трьох. Таких виявляється шістнадцять {00111, 01011, 01101, 01110, 01111, 10011, 10101, 10110, 10111, 11001, 11010, 11011, 111}. Підберемо серед них три комбінації, кодова відстань між якими не менше $d_{\min}–2 = 4–2 = 2$.

Це, наприклад, комбінації {00111, 01011, 01101}. Тоді підматриця B матиме вигляд

$$B = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Об'єднуючи підматриці A і B, отримуємо породжуючу матрицю G.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Для рядків цієї матриці виконується умова $d_{\min} \geq 4$. ■

Приклад 5.9. Закодувати за допомогою коду, розглянутого у прикладі 5.7, вектор $\vec{a}$ = [1 1 0 1].

Рішення. Відповідно до (5.13)

$$\vec{v} = \vec{a}G = [1101] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Зручно представити результат як суму за модулем два першого, другого та четвертого рядків (за місцями розташування одиниць у вихідному коді)

$\vec{v}$ =1·[1000011] ⊕ 1·[0100101] ⊕ 0·[0010110] ⊕ 1·[0001111] =

= [1101001]. ■

Приклад 5.10. Закодувати за допомогою коду, розглянутого у прикладі 5.8 всі можливі двійкові вектори. Перевірити корегуючи здатність коду.

Рішення. Відповідно до (5.13)

$$\vec{v} = \vec{a}G = \left[a_1 a_2 a_3\right] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Кодуємо всі можливі трибітові комбінації:

$\vec{v}_{000}$=0·[10000111] ⊕ 0·[01001011] ⊕ 0·[00101101] = [00000000];

$\vec{v}_{001}$=0·[10000111] ⊕ 0·[01001011] ⊕ 1·[00101101] = [00101101];

$\vec{v}_{010}$=0·[10000111] ⊕ 1·[01001011] ⊕ 0·[00101101] = [01001011];

$\vec{v}_{011}$=0·[10000111] ⊕ 1·[01001011] ⊕ 1·[00101101] = [01100110];

$\vec{v}_{100}$=1·[10000111] ⊕ 0·[01001011] ⊕ 0·[00101101] = [10000111];

$\vec{v}_{101}$=1·[10000111] ⊕ 0·[01001011] ⊕ 1·[00101101] = [10101010];

$\vec{v}_{110}$=1·[10000111] ⊕ 1·[01001011] ⊕ 0·[00101101] = [11001100];

$\vec{v}_{111}$=1·[10000111] ⊕ 1·[01001011] ⊕ 1·[00101101] = [11100001].

Аналіз отриманих комбінацій показує, що умова $d_{\min}\geq 4$ виконується.■

При передачі послідовності каналом зв'язку окремі розряди можуть спотворюватися і на приймач надходить вектор , який в деяких розрядах відрізняється від переданого

$\vec{y} = \vec{v} \oplus \vec{\varepsilon}$, де $\vec{\varepsilon}$ – вектор помилок.

Алгоритм пошуку помилок у прийнятій послідовності ґрунтується на рівності (5.16). Обчислюється низка контрольних символів s_1, s_2,… s_r

$$
\begin{aligned}
s_1 &= c_1 \oplus a_1 \cdot b_{11} \oplus a_2 \cdot b_{21} \oplus \ldots \oplus a_k \cdot b_{k1}; \\
s_2 &= c_2 \oplus a_1 \cdot b_{12} \oplus a_2 \cdot b_{22} \oplus \ldots + a_k \cdot b_{k2}; \\
&\ldots\ldots\ldots\ldots\ldots \\
s_r &= c_r \oplus a_1 \cdot b_{1r} \oplus a_2 \cdot b_{2r} \oplus \ldots \oplus a_k \cdot b_{kr}.
\end{aligned}
\qquad (5.17)
$$

Сукупність контрольних символів $s_1, s_2, \ldots s_r$. називається *синдром* (за аналогією з медичним синдромом – сукупністю симптомів, що дозволяють діагностувати те чи інше захворювання).

Якщо помилок у каналі не сталося, всі розряди синдрому дорівнюють нулю. Справді,

$s_1 = c_1 \oplus a_1 \cdot b_{11} \oplus a_2 \cdot b_{21} \oplus \ldots \oplus a_k \cdot b_{k1} = c_1 \oplus c_1 = 0$ тощо.

Якщо у каналі сталася помилка, один або декілька розрядів синдрому набувають одиничних значень, при цьому вид синдрому дозволяє у деяких випадках визначити, в якому саме розряді сталася помилка.

У матричному вигляді процес декодування (5.17) зводиться до обчислення вектора синдрому помилки $\vec{S}$

$$
\vec{S} = \vec{y} \cdot H^T, \qquad (5.18)
$$

де H^T – транспонована перевірочна матриця (транспонування – процес заміни у матриці рядків на стовпці).

Перевірочна матриця H розмірністю $r \times n$ складається з двох підматриць – одиничної підматриці E розмірністю $r \times r$ та перевірочної підматриці D розмірністю $r \times k$. Рядками перевірочної підматриці D є стовпці перевірочної підматриці B породжуючої матриці розмірністю $k \times r$. До отриманої підматриці D праворуч дописується одинична підматриця E. Таким чином, перевірочна матриця H має вигляд

$$
H = \left[D \middle| E \right] = \left[\begin{array}{ccccc|ccccc}
b_{11} & b_{21} & b_{31} & \ldots & b_{k1} & 1 & 0 & 0 & \ldots & 0 \\
b_{12} & b_{22} & b_{32} & \ldots & b_{k2} & 0 & 1 & 0 & \ldots & 0 \\
\cdot & \cdot & \cdot & \ldots & \cdot & \cdot & \cdot & \cdot & \ldots & \cdot \\
b_{1r} & b_{2r} & b_{3r} & \ldots & b_{kr} & 0 & 0 & 0 & \ldots & 1
\end{array}\right]. \qquad (5.19)
$$

Приклад 5.11. Для коду, розглянутого в прикладі 5.7, побудувати перевірочну матрицю. Обчислити синдроми для послідовностей

y_1 = [1101001], y_2 = [1101011].

Рішення. Для цього коду $k = 4$, $n = 7$, $r = 3$.

Побудуємо перевірочну матрицю коду. Підматриця E являтиме собою одиничну матрицю розмірності 3×3

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Подматриця D буде мати розмірність 3×4 и являти собой транспоновану підматрицю B породжуючої матриці

$$D = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

Об'єднуючи підматриці D та E, отримуємо перевірочну матрицю H

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Обчислимо синдроми для заданих в умовах завдання векторів. Для першої послідовності y_1 = [1101001]

$$\vec{S} = \vec{y} \cdot H^T = [1101001] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

s_1=1·0⊕1·1⊕0·1⊕1·1⊕0·1⊕0·0⊕1·0=0⊕1⊕0⊕1⊕0⊕0⊕0=0;

s_2=1·1⊕1·0⊕0·1⊕1·1⊕0·0⊕0·1⊕1·0=1⊕0⊕0⊕1⊕0⊕0⊕0=0;

s_3=1·1⊕1·1⊕0·0⊕1·1⊕0·0⊕0·0⊕1·1=1⊕1⊕0⊕1⊕0⊕0⊕1=0.

Усі розряди синдрому дорівнюють нулю, тому можна дійти висновку, що помилок під час передачі не сталося. Для другої послідовності

y_1 = [1101011].

s_1=1·0⊕1·1⊕0·1⊕1·1⊕0·1⊕1·0⊕1·0=0⊕1⊕0⊕1⊕0⊕0⊕0=0;

s_2=1·1⊕1·0⊕0·1⊕1·1⊕0·0⊕1·1⊕1·0=1⊕0⊕0⊕1⊕0⊕0⊕0=1;

s_3=1·1⊕1·1⊕0·0⊕1·1⊕0·0⊕1·0⊕1·1=1⊕1⊕0⊕1⊕0⊕0⊕1=0.

Один із розрядів синдрому дорівнює одиниці, тому робимо висновок, що один або кілька розрядів коду при передачі спотворилися.

Спробуємо з'ясувати, в якому саме розряді помилка. Синдром

S=[s_1, s_2, s_3]=[010].

Можна помітити, що виконалося перше та третє перевірочне співвідношення і не виконалося друге. Це означає, що номер помилкового розряду відповідає номеру стовпця перевірочної матриці, що містить символи 101. Це сьомий стовпець, отже помилка відбулася в сьомому символі. ■

Приклад 5.12. Для коду, розглянутого в прикладі 5.8, побудувати перевірочну матрицю. Обчислити синдром для послідовності

y = [01101101].

Рішення. Для цього коду k =3, n =8, r =5.

Підматриця E являтиме собою одиничну матрицю розмірності 5×5

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Подматриця D буде мати розмірність 5×3 и являти собой транспоновану підматрицю B породжуючої матриці

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Об'єднуючи підматриці D та E, отримуємо перевірочну матрицю H.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Обчислимо синдроми для заданого в умовах вектора

y=[01101101]

$$\vec{S} = \vec{y} \cdot H^T = [01101101] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

s_1=0·0⊕1·0⊕1·1⊕0·1⊕1·1⊕1·0⊕0·0⊕1·0=0⊕0⊕1⊕0⊕1⊕0⊕0⊕0=0;

s_2=0·0⊕1·1⊕1·1⊕0·0⊕1·1⊕1·0⊕0·0⊕1·0=0⊕1⊕1⊕0⊕1⊕0⊕0⊕0=1;

s_3=0·1⊕1·0⊕1·1⊕0·0⊕1·0⊕1·1⊕0·0⊕1·0=0⊕0⊕1⊕0⊕0⊕1⊕0⊕0=0;

s_4=0·1⊕1·1⊕1·0⊕0·0⊕1·0⊕1·0⊕0·1⊕1·0=0⊕1⊕0⊕0⊕0⊕0⊕0⊕0=1;

s_5=0·1⊕1·1⊕1·1⊕0·0⊕1·0⊕1·0⊕0·1⊕1·1=0⊕1⊕1⊕0⊕0⊕0⊕0⊕1=1.

Деякі з розрядів синдрому дорівнюють одиниці, тому робимо висновок, що в прийнятому повідомленні містяться помилки.

5.3 Коди з перевіркою на парність

Найпростішим лінійним блоковим кодом є (n, n–1) *код з перевіркою на парність* . Код утворюється шляхом додавання до комбінації що передається одного перевірочного символу (0 або 1), так щоб загальна кількість одиниць у комбінації, було парним. Для цього контрольний біт парності (біт паритету, анг. *Parity bit*) формується як сума за модулем два всіх інформаційних розрядів.

Відносна надмірність коду з перевіркою на парність

$R = r/n = 1/n$, швидкість коду $V = k/n = (n - 1)/n$.

Кодову комбінацію систематичного (n, k) коду можна записати як $v = \{a_1, a_2, \ldots, a_k, c_1\}$, де a_i – інформаційні символи, c_1 – перевірочний символ:

$$c_1 = a_1 \oplus a_2 \oplus \ldots \oplus a_k . \qquad (5.20)$$

Приклад 5.13. Закодувати кодом із перевіркою на парність інформаційну послідовність A = {0 1 1 0 1 0 1}. Визначити відносну надмірність та швидкість коду.

Рішення. Формуємо перевірочний розряд коду згідно (5.20).

$c_1 = a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0.$

Тоді кодова послідовність виглядає як

v = {0 1 1 0 1 0 1 0}.

Для цього коду $k = 7$, $r = 1$, $n = k + r = 8$.

Відносна надмірність коду $R = 1/n = 1/8$,

швидкість коду $V = k/n = = 7/8$. ■

Код з перевіркою на парність має мінімальну кодову відстань $d_{\min}=2$, тобто може виявляти поодинокі помилки та помилки непарної кратності. Якщо в процесі передачі коду відбулася одна помилка, причому неважливо, в якій його позиції, то загальна кількість одиниць прийнятої послідовності вже не буде парною. Таким чином, ознакою відсутності помилки у прийнятій послідовності Y може бути парність числа одиниць. Тому такі коди отримали назву кодів з перевіркою на парність.

Породжуюча матриця даного коду має розмірність $(n-1)\times n$ і може буте представлена у вигляді

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Перевірочна матриця містить лише один рядок і виглядає як

$$H = \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \end{bmatrix}.$$

Тобто синдром коду складається з одного розряду та обчислюється як

$$s_1 = p = a_1 \oplus a_2 \oplus \dots \oplus a_k \oplus c_1 = y_1 \oplus y_2 \oplus \dots \oplus y_n, \qquad (5.21)$$

де $y_1 \dots y_n$ – розряди прийнятої кодової комбінації.

Приклад 5.14. Декодувати закодовану кодом з перевіркою на парність кодову послідовність Y = {0 1 1 1 0 1 0 0 1 1 1 0}.

Рішення. Обчислюємо перевірочне співвідношення (розряд синдрому) згідно (5.21).

$s = y_1 \oplus y_2 \oplus \ldots \oplus y_{12} =$

$= 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1.$

Оскільки s=1, робимо висновок, що під час передачі сталася поодинока помилка або помилка непарної кратності, наприклад потрійна.

Даний код дозволяє тільки виявляти одноразові помилки і всі помилки непарної кратності, тому що тільки в цих випадках кількість одиниць комбінації стане непарним. Якщо ж у прийнятій послідовності відбулося дві помилки, то загальна кількість одиниць у ній знову стане парною і помилка не буде виявлена. ■

Оцінимо ймовірність помилки, що не виявляється . Ця ймовірність визначається як сума ймовірностей помилок парної кратності

$$P_{\text{нп}} = P_2 + P_4 + \ldots + P_m, \tag{5.22}$$

де P_i – ймовірність помилки кратності i, m – максимальне парне число, що не перевищує n .

Підставляючи (1.31) до (5.22), отримуємо

$$P_{нп} = \sum_{i=2, i-парне}^{n} C_n^i \cdot p^i \cdot (1-p)^{(n-i)}, \tag{5.23}$$

де p – ймовірність помилки в окремому розряді. ■

Приклад 5.15. Визначити ймовірність невиявленої помилки для коду з перевіркою на парність з n = 6. Ймовірність бітової помилки p = 0,1.

Рішення. Згідно (5.22), (5.23)

$P_2 = C_6^2 \cdot 0{,}1^2 \cdot (0{,}9)^4 = 15 \cdot 0{,}01 \cdot 0{,}6561 = 0{,}098415,$

$P_4 = C_6^4 \cdot 0{,}1^4 \cdot (0{,}9)^2 = 15 \cdot 0{,}0001 \cdot 0{,}81 = 0{,}001215,$

$P_6 = C_6^6 \cdot 0{,}1^6 \cdot (0{,}9)^0 = 1 \cdot 0{,}000001 \cdot 1 = 0{,}000001.$

Ймовірність помилки , що не виявляється

$P_{\text{нп}} = P_2 + P_4 + P_6 = 0{,}099631.$

У той же час ймовірність спотворення хоча б одного символу в n-бітному повідомленняі складе $P_{\text{п}} = 1-(1-p)^n$. Для цього прикладу $P_{\text{п}}$=1–(1–p) = =1–(1–0,1)^6 = 0,4686. Таким чином, розглянутий код дозволяє виявляти 78,7% помилок. ■

Перевагами коду з перевіркою на парність є мала надмірність і простота реалізації, недоліком – невисока коригуюча здатність. Код може визначити, чи є в повідомленні помилка, але не може виявити її місцезнаходження. У цьому випадку можна повторити передачу слова, в якому було допущено помилку. Крім того, контроль на парність іноді застосовується спільно з більш потужними кодами.

5.4 Код із простим повторенням

В основу *коду з простим повторенням* покладено повторення кодової вихідної комбінації. Код, таким чином, ділиться на інформаційну та перевірочну складові, причому перевірочна складова повторює інформаційну. Для цього коду $r = k = n/2$.

Відносна надмірність коду з простим повторенням :

$R = r/n = 1/2$, швидкість коду $V = k/n = 1/2$.

Породжуюча матриця коду з простим повторенням має розмірність $(n/2)\times n$ і виглядає як

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (5.24)$$

Приклад 5.16. Закодувати кодом із простим повторенням інформаційну послідовність $A = \{0\ 1\ 1\ 0\ 1\}$.

Рішення. Приписуємо до інформаційної послідовності аналогічну до неї перевірочну. Тоді кодова послідовність має вигляд

$v = \{0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\}$. ■

При декодуванні порівнюються інформаційна та перевірочна частини коду. Якщо вони збігаються, помилки відсутні. Якщо хоча в одному розряді символи відрізняються, робиться висновок про наявність помилки.

Перевірочна матриця збігається з породжуючою (5.23). Розрядами синдрому є суми за модулем два аналогічних розрядів першої та другої частини коду

$$s_1 = y_1 \oplus y_{k+1},\ s_2 = y_2 \oplus y_{k+2},\ \ldots\ s_i = y_i \oplus y_{k+i} \ldots s_r = y_k \oplus y_n . \quad (5.25)$$

Приклад 5.17. Декодувати закодовану кодом із простим повторенням кодову послідовність $Y = \{0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\}$.

Рішення. Обчислюємо синдром згідно (5.25). В даному випадку

$n = 8,\ k = r = n/2 = 4.$

$s_1 = y_1 \oplus y_5 = 0 \oplus 0 = 0$, $s_2 = y_2 \oplus y_6 = 1 \oplus 1 = 0$,

$s_3 = y_3 \oplus y_7 = 1 \oplus 0 = 1$, $s_4 = y_4 \oplus y_8 = 1 \oplus 1 = 0$,

Оскільки один із розрядів синдрому (s_3) дорівнює одиниці, робимо висновок, що під час передачі каналом зв'язку сталася помилка. Аналізуючи синдром, можна припустити, що помилка відбулася в третьому або сьомому розрядах, але встановити точне розташування помилки і виправити відповідний біт не є можливим. ■

Код із простим повторенням дозволяє виявляти всі помилки непарної кратності, а також більшість помилок парної кратності, за винятком помилок в елементах, що стоять на тих самих позиціях в інформаційній та перевірочній частинах. Ймовірність невиявленої помилки для цього коду

$$P_{нп} = \sum_{i=1}^{n/2} C_{n/2}^{i} \cdot p^{2i} \cdot (1-p)^{(n-2i)}, \quad (5.26)$$

де p – ймовірність помилки в окремому розряді.

Приклад 5.18. Визначити ймовірність невиявленої помилки для коду з простим повторенням з $n = 6$. Ймовірність бітової помилки $p = 0{,}1$.

Рішення. Згідно (5.26).

$P_{нп} = P_2 + P_4 + P_6$

$P_2 = C_3^1 \cdot 0{,}1^2 \cdot (0{,}9)^4 = 3 \cdot 0{,}01 \cdot 0{,}6561 = 0{,}019683,$

$P_4 = C_3^2 \cdot 0{,}1^4 \cdot (0{,}9)^2 = 3 \cdot 0{,}0001 \cdot 0{,}81 = 0{,}000243,$

$P_6 = C_3^3 \cdot 0{,}1^6 \cdot (0{,}9)^0 = 1 \cdot 0{,}000001 \cdot 1 = 0{,}000001.$

Ймовірність помилки, що не виявляється .

$P_{нп} = P_2 + P_4 + P_6 = 0{,}019927.$ ■

Порівняння результатів прикладів 5.15 і 5.18 показує, що код з повторенням має більш високу коригуючу здатність, проте і істотно більшу надмірність.

Можливе підвищення коригуючої здатності коду шляхом повторення інформаційної частини коду з кратністю більшою за два. Так, код із триразовим повторенням дозволяє не тільки виявляти, а й виправляти помилки.

Застосовувати коди з повторенням може бути доцільним, коли простота засобів кодування та декодування виявляється важливішою за швидкість передачі.

5.5 Кореляційний код

Різновидом коду із повторенням є *кореляційний код.* При кодуванні інформації кореляційним кодом нульовий інформаційний символ записується комбінацією біт 01 одиничний – бітами 10.

Характеристики кореляційного коду збігаються з характеристиками коду із простим повторенням. Відносна надмірність кореляційного коду

$R = 1/2$, швидкість коду $V = 1/2$.

Приклад 5.19. Закодувати кореляційним кодом інформаційну послідовність $A = \{0\ 1\ 1\ 0\ 1\}$.

Рішення. Замінюємо в послідовності одиниці на комбінацію 10, нулі на 01. Тоді результат кодування $v = \{0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\}$.

Ознака помилки при декодуванні – поява у парних елементах однакових символів 00 або 11 (замість 01 та 10). Розрядами синдрому є інверсії суми за модулем парних розрядів

$$s_1 = \overline{y_1 \oplus y_2}, s_2 = \overline{y_3 \oplus y_4}, \ldots s_i = \overline{y_{2i-1} \oplus y_{2i}}, \ldots s_r = \overline{y_{n-1} \oplus y_n}. \qquad (5.27)$$

При непарному k (5.27) можна перетворити. Наприклад,

$$s_1 = \overline{y_1 \oplus y_2} = y_1 \oplus y_2 \oplus 1 = (y_1 \oplus y_2) \oplus$$
$$\oplus (y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_8 \oplus y_9 \oplus y_{10}) =$$
$$= y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_8 \oplus y_9 \oplus y_{10}.$$

Аналогічно $s_2 = y_1 \oplus y_2 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_8 \oplus y_9 \oplus y_{10}$ і т.д. ■

Приклад 5.20. Декодувати закодовану кореляційним кодом послідовність $Y = \{0\ 1\ 1\ 1\ 1\ 0\}$.

Рішення. Обчислюємо синдром згідно (5.27). В даному випадку

$n = 6$, $k = r = n/2 = 3$.

$$s_1 = y_3 \oplus y_4 \oplus y_5 \oplus y_6 = 1 \oplus 1 \oplus 1 \oplus 0 = 1,$$

$$s_2 = y_1 \oplus y_2 \oplus y_5 \oplus y_6 = 0 \oplus 1 \oplus 1 \oplus 0 = 0,$$

$$s_3 = y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1.$$

Оскільки декотрі з розрядів синдрому (s_1, s_3) дорівнюють одиниці, робимо висновок, що під час передачі по каналу зв'язку сталася помилка. ■

Ймовірність невиявленої помилки для кореляційного коду збігається з такою ймовірністю для коду з повторенням і визначається співвідношенням (5.26).

У протоколах передачі даних фізичного рівня такий метод кодування називається *манчестерським кодом* (*код Манчестер- II , абсолютний біімпульсний код*). Під час передачі інформації манчестерським кодом кожен біт передається протягом фіксованого часу, причому інформаційним є перехід від низького рівня сигналу до високого або навпаки в середині кожного періоду. Назва коду походить від університету Манчестера , де таке кодування було вперше використано для зберігання даних на магнітному барабані комп'ютера *Манчестерський Марк I*.

Існують різні угоди кодування символів манчестерським кодом. Перше було вперше запропоновано Г. Е. Томасом у 1949. Нульовому символу відповідає перепад від низького рівня до високого, тобто комбінації біт 01, для одиничного символу перепад рівнів буде від високого до низького, тобто 10. Іншої угоди дотримується стандарт IEEE 802.4 (шина з маркерним доступом) та стандарт IEEE 802.3 (Ethernet). За цією угодою логічний нуль представлений перепадом сигналу від високого до низького, а логічна одиниця – перепадом сигналу від низького до високого.

Крім функції виявлення помилок, манчестерський код має ряд корисних властивостей. Сигнал, закодований манчестерським кодом, є самосинхронізуючим, тобто для передачі даних не потрібна додаткова лінія передачі тактових імпульсів , за рахунок того, що за час передачі одного біта даних, незалежно від того, це одиниця це або нуль, забезпечується один перехід з одного рівня на інший. Це дозволяє

приймачеві синхронізувати свій внутрішній тактовий генератор із тактовим генератором передавача.

Крім того, закодована манчестерським кодом інформація, не утворює постійної складової навіть у разі передачі довгих послідовностей з нулів і одиниць, тому електричні з'єднання, що передають сигнал, можуть бути гальванічно розв'язані , наприклад, за допомогою трансформатора.

5.6 Інверсний код

Різновидом коду з повторенням є *інверсний код* (*код Бауера*). Відмінність його від коду з простим повторенням полягає в тому, що при парному числі одиниць у вихідній комбінації друга перевірочна комбінація передається без зміни, а при непарному повторення відбувається з інверсією символів.

Характеристики інверсного коду збігаються з характеристиками коду із простим повторенням. Відносна надмірність кореляційного коду R=1/2, швидкість коду V = 1/2.

Оскільки вирази для перевірочних розрядів коду можна записати як

$c_k = v_{i+k} = a_i \oplus (a_1 \oplus a_2 \oplus \ldots \oplus a_i \oplus \ldots \oplus a_k)$,

то породжуючи матриця інверсного коду розмірністю $(n/2)\times n$ має вигляд

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \qquad (5.28)$$

Приклад 5.21. Закодувати інверсним кодом інформаційні послідовності A_1 = {0 1 0 0 1}, A_2 = {0 1 1 0 1}.

Рішення. Оскільки у послідовності A_1 міститься парне число одиниць, то перевірочні розряди (виділені курсивом) точно повторюють інформаційні.

v_1 = {0 1 0 0 1 *0 1 0 0 1*}.

Послідовність A_2 містить непарне число одиниць, тому перевірочні розряди є інверсією інформаційних інформаційних.

v_2 = {0 1 1 0 1 *1 0 0 1 0*}. ■

Виявлення помилок в інверсному коді проводиться шляхом підсумовування за модулем першої і другої половини прийнятої кодової комбінації. Причому, якщо у перших $n/2$ символах число одиниць буде парним, то інші $n/2$ символи приймаються у прямому вигляді, а при непарному – в инвертированном.

Застосовуючи вираз для побудови перевірочної матриці до породжуючої матриці інверсного коду (5.28), визначаємо його перевірочну матрицю

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (5.29)$$

Приклад 5.22. Декодувати закодовану інверсним кодом кодову послідовність $Y = \{0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\}$.

Рішення. Для цього коду $n = 8$, $k = r = n/2 = 4$.

Перевірочна матриця відповідно до (5.29)

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Обчислюємо синдром згідно (5.18)

$$\vec{S} = \vec{y} \cdot H^T = [0111010010] \cdot \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

s_1=0·0 ⊕1·1 ⊕1·1 ⊕1·1 ⊕0·1 ⊕1·1 ⊕0·0 ⊕0·0 ⊕1·0 ⊕0·0 = 0;

s_2=0·1 ⊕1·0 ⊕1·1 ⊕1·1 ⊕0·1 ⊕1·0 ⊕0·1 ⊕0·0 ⊕1·0 ⊕0·0 = 0;

s_3=0·1 ⊕1·1 ⊕1·0 ⊕1·1 ⊕0·1 ⊕1·0 ⊕0·0 ⊕0·1 ⊕1·0 ⊕0·0 = 0;

s_4=0·1 ⊕1·1 ⊕1·1 ⊕1·0 ⊕0·1 ⊕1·0 ⊕0·0 ⊕0·0 ⊕1·1 ⊕0·0 = 1;

s_5=0·1 ⊕1·1 ⊕1·1 ⊕1·1 ⊕0·0 ⊕1·0 ⊕0·0 ⊕0·0 ⊕1·0 ⊕0·1 = 1.

Розряди синдрому s_4, s_5 приймають одиничне значення, отже в одному з символів у парах 4-9 і 5-10 відбулася помилка. Справді, оскільки сума за модулем двох перших п'яти символів дорівнює одиниці, то друга половина повинна була б бути прийнята з інверсією, тобто мати вигляд C={1 0 0 0 1}. Оскільки значення дев'ятого та десятого розрядів відрізняються від передбачуваних, робимо висновок про наявність помилок у четвертому/дев'ятому та п'ятому/десятому розрядах. ■

При $k\geq 4$ код має мінімальну кодову відстань $d_{\min}=4$, тобто дозволяє виявити одноразові, двократні та триразові помилки, а також усі помилки непарної кратності та більшість помилок парної кратності за винятком одночасного спотворення двох, чотирьох і т. д. інформаційних розрядів та відповідних їм перевірочних розрядів.

Ймовірність помилки, що не виявляється, для інверсного коду

$$P_{нп} = \sum_{i=2,i-парне}^{n/2} C_{n/2}^{i} \cdot [p^{i} \cdot (1-p)^{(n/2-i)}]^2. \qquad (5.30)$$

Приклад 5.23. Визначити ймовірність невиявленої помилки для коду з простим повторенням з n = 10. Ймовірність бітової помилки p = 0,1.

Рішення. Згідно (5.29)

$P_{нп} = P_4 + P_8$, де

$P_4 = C_5^2 \cdot [0{,}1^2 \cdot (0{,}9)^3]^2 = 10 \cdot [0{,}01 \cdot 0{,}729]^2 = 10 \cdot 0{,}000053144 = 0{,}00053144;$

$P_8 = C_5^4 \cdot [0{,}1^4 \cdot (0{,}9)^1]^2 = 5 \cdot [0{,}0001 \cdot 0{,}9]^2 = 4{,}05 \cdot 10^{-8}.$

$P_{нп} = P_4 + P_8 = 0{,}00053148.$

Очевидно, що в більшості випадків у (5.29) можна враховувати тільки перший доданок.

5.7 Коди з постійною вагою

Коди з постійною вагою – рівномірні двійкові коди, в яких усі дозволені комбінації містять однакову кількість одиниць. Ці коди є нероздільними, тобто не можна відокремити інформаційні розряди від перевірочних. Часто коди з постійною вагою позначають як m/l, де n – число розрядів коду, m – число одиниць (вага) кодової комбінації, l – число нулів, $l = n - m$.

Прикладом коду із постійною вагою є *Міжнародний телеграфний код МТК-3* . У цьому коді всі дозволені кодові комбінації мають вагу рівну трьом, розрядність комбінацій $n = 7$, тобто код може бути позначений як 3/4 . До кодів із постійною вагою відноситься також 1/1 кореляційний код.

Ще одним прикладом кодів із постійною вагою є *унітарний код* (в англомовній літературі *unitary code, one-hot*) – двійковий код фіксованої довжини, що містить лише одну одиницю – прямий унітарний код або лише один нуль – зворотний (інверсний, *one-cold*) унітарний код. Довжина коду n визначається кількістю об'єктів, що кодуються, тобто кожному об'єкту відповідає окремий розряд коду, а значення коду положенням 1 або 0 в кодовому слові.

Приклад 5.24. Закодувати п'ятирозрядним унітарним кодом числа від одного до п'яти.

Рішення. Результати кодування зведемо до таблиці 5.2.

Таблиця 5.2 – Кодування чисел унітарним кодом

Число	Прямий унітарний код	Зворотній код
1	00001	11110
2	00010	11101
3	00100	11011
4	01000	10111
5	10000	01111

Унітарне кодування використовується в цифровій схемотехніці, теорії автоматів, нейронних мережах, системах комп'ютерного перекладу.

Визначимо число дозволених комбінацій для $m/(n-m)$ коду з постійною вагою.

Відповідно (1.31) загальна кількість n-розрядних комбінацій, що містять m одиниць, становить $N_p = C_n^m = \frac{n!}{(n-m)!m!}$. Відносна надмірність коду з постійною вагою

$$R = 1 - \frac{\log_2 N_p}{\log_2 N} = 1 - \frac{\log_2 C_n^m}{n}. \qquad (5.31)$$

■

Приклад 5.25. Визначити відносну надмірність та швидкість 3/4 коду МТК-3.

Рішення. Для цього коду $m = 3$, $l = 4$, $n = m + l = 3 + 4 = 7$.

Згідно (5.30)

$$R = 1 - \frac{\log_2 C_7^4}{7} = 1 - \frac{\log_2 35}{7} = 1 - \frac{5{,}1293}{7} = 1 - 0{,}7328 = 0{,}2672.$$

Швидкість 3/4 коду $V = 1 - R = 1 - 0{,}2672 = 0{,}7328$. ■

Нероздільність кодів з постійною вагою може викликати певні проблеми під час кодування. Зазвичай спочатку вибирається абетка системи, тобто визначається набір символів, що підлягають кодуванню. Далі кожній літері алфавіту зіставляється кодове слово з постійною вагою. Так, у коді «три з семи» МТК-3 латинській літері А відповідає семирозрядний код 0011010, що містить три одиниці, літері В код 0011001, літері С – 1001100 і т.д.

Декодування прийнятих комбінацій зводиться до визначення їхньої ваги. Якщо вага відрізняється від заданої, робиться висновок, що комбінація прийнята з помилкою. Код з постійною вагою виявляє всі помилки непарної та частину помилок парної кратності. Не виявляються лише звані *помилки зміщення*, у разі виникнення яких число переходів з одиниці на нуль дорівнює числу переходів із нуля на одиницю і, таким чином, зберігається вага комбінації.

Приклад 5.26. Декодувати закодовану 3/4 кодом з постійною вагою послідовність $Y = \{0\ 0\ 1\ 1\ 1\ 0\ 1\}$.

Рішення. Підраховуємо число одиниць у прийнятій послідовності. Отримуємо $m_{од} = 4$. Це значення відрізняється від заданого $m = 3$, тому робимо висновок про наявність помилки у прийнятій послідовності. ■

Визначимо ймовірність одиночної помилки зміщення для *m*/*l* коду. При цьому будемо припускати симетричність каналу зв'язку, тобто рівність ймовірностей *переходу* з нуля в одиницю і з одиниці в нуль.

Для виникнення помилки зміщення необхідно, щоб одна з *m* одиниць перейшла в нуль і одночасно один із *l* нулів – в одиницю. Ймовірність першої події $P_{10} = C_m^1 \cdot p^1 \cdot (1-p)^{(m-1)}$, ймовірність другої $P_{01} = C_l^1 \cdot p^1 \cdot (1-p)^{(l-1)}$.

Користуючись теоремою множення ймовірностей сумісних та незалежних подій, згідно (2.15) отримуємо

$$P_{зм1} = P_{10} \cdot P_{01} = [C_m^1 \cdot p^1 \cdot (1-p)^{(m-1)}] \cdot [C_l^1 \cdot p^1 \cdot (1-p)^{(l-1)}] =$$
$$= C_m^1 \cdot C_l^1 \cdot p^2 \cdot (1-p)^{(n-2)}.$$

У загальному випадку для *n*-бітової кодової послідовності, закодованої *m*/(*n*–*m*) кодом з постійною вагою, можна записати вираз для ймовірності помилки, що не виявляється

$$P_{нп} = \sum_{i=1}^{\min m,l} P_{зм} = \sum_{i=1}^{\min m,l} C_m^i \cdot C_{n-m}^i \cdot p^{2i} \cdot (1-p)^{(n-2i)}. \qquad (5.32)$$

■

Приклад 5.27. Визначити ймовірність помилки для 3/4 коду з постійною вагою. Ймовірність бітової помилки $p = 0{,}1$.

Рішення. Для цього коду $m = 3$, $l = 4$, $n = m + l = 3 + 4 = 7$.

Відповідно (5.32) $P_{\text{але}} = P_{\text{см1}} + P_{\text{см1}} + P_{\text{см1}}$.

$P_{зм1} = C_3^1 \cdot C_4^1 \cdot p^2 \cdot (1-p)^{(7-2)} = 3 \cdot 4 \cdot 0{,}1^2 \cdot 0{,}9^5 = 12 \cdot 0{,}01 \cdot 0{,}5905 = 0{,}0708588;$

$P_{зм2} = C_3^2 \cdot C_4^2 \cdot p^4 \cdot (1-p)^{(7-4)} = 3 \cdot 6 \cdot 0{,}1^4 \cdot 0{,}9^3 18 \cdot 0{,}0001 \cdot 0{,}729 = 0{,}0013122;$

$P_{зм3} = C_3^3 \cdot C_4^3 \cdot p^6 \cdot (1-p)^{(7-6)} = 1 \cdot 4 \cdot 0{,}1^6 \cdot 0{,}9^1 = 4 \cdot 0{,}000001 \cdot 0{,}9 = 0{,}0000036.$

$P_{\text{нп}} = 0{,}0708588 + 0{,}0013122 + 0{,}0000036 = 0{,}0721746$. ■

Порівняння отриманих результатів з результатами прикладів (5.16) і (5.22) показує меншу ефективність кодів з постійною вагою в порівнянні з кодами із повторенням для симетричних каналів. Значно ефективніше їх застосування в асиметричних каналах, де ймовірність помилок зміщення невелика, наприклад, при частотній маніпуляції в каналах з селективними завмираннями.

5.8 Код Бергера

Коди Бергера (коди з підсумовуванням) відносяться до множини роздільних несистематичних кодів. При представленні чисел кодом Бергера підраховується кількість одиниць в інформаційній частині коду, після чого формуються перевірочні розряди, що являють собою двійкове представлення числа одиниць. Таким чином, число перевірочних розрядів r дорівнює найменшому цілому числу, що перевищує $\log_2 k$, тобто

$$r = \lceil \log_2 (k+1) \rceil. \tag{5.33}$$

Таким чином, відносна надмірність коду Бергера

$$R = \frac{r}{n} = \frac{\lceil \log_2 (k+1) \rceil}{k + \lceil \log_2 (k+1) \rceil}. \tag{5.34}$$

Швидкість коду Бергера складає

$$V = \frac{k}{n} = \frac{k}{k + \lceil \log_2 (k+1) \rceil}. \tag{5.35}$$

Приклад 5.28. Закодувати кодом Бергера інформаційну послідовність A={0110001}. Визначити відносну надмірність та швидкість даного коду.

Рішення. Число інформаційних розрядів складає k = 7. Число перевірочних розрядів згідно (12.14) $r = \lceil \log_2 (7+1) \rceil = \lceil \log_2 8 \rceil = \lceil 3 \rceil = 3$. Дійсно, трьох двійкових розрядів достатньо для подання будь-якої можливої кількості одиниць від нуля до семи.

Загальна кількість розрядів $n = k + r = 7 + 3 = 10$.

Підраховуємо кількість одиниць в інформаційній частині $m_{од} = 3$. Записуємо отримане число в двійковому коді, виділяючи для перевірочної частини три розряди $m_{од2}$ = {0 1 1}. Інвертуємо отриманий двійковий код та отримуємо перевірочну частину коду Бергера C = {1 0 0}. Об'єднуємо інформаційну та перевірочні частини коду

v = {0 1 1 0 0 0 1 1 0 0}. ■

Відносна надмірність та швидкість даного коду, обчислені згідно (5.34), (5.35) складуть відповідно

$R = r/n = 3/10 = 0{,}3$, швидкість коду $V = k/n = 7/10 = 0{,}7$.

При декодуванні коду Бергера підраховується кількість одиниць в інформаційній частині та порівнюється з контрольною кодовою комбінацією, утвореною як інверсія перевірочних розрядів. При збігу двох чисел робиться висновок про відсутність помилок у прийнятій кодовій комбінації. Якщо є розбіжність, комбінація бракується.

Приклад 5.29. Декодувати закодовану (14, 10) кодом Бергера послідовність Y = {0 1 1 1 1 0 1 1 1 0 1 0 0 1}.

Рішення. Число інформаційних розрядів коду k = 10, число перевірочних r= 14 – 10 = 4.

Виділяємо інформаційну частину та підраховуємо в ній число одиниць A = {0 1 1 1 1 0 1 1 1 0}, $m_{од}$ = 7. Записуємо отримане число у двійковому коді, виділяючи для перевірочної частини r = 4 розряду $m_{од2}$={0 1 1 1 }.

Зчитуємо перевірочні розряди коду та інвертуємо їх C = {1001}, $C_{інв}$= {0110}. Порівнюючи підраховане число одиниць $m_{од2}$ і контрольну комбінацію $C_{інв}$, виявляємо розбіжність і робимо висновок про наявність помилок.■

У симетричних каналах зв'язку коди Бергера виявляють усі поодинокі помилки та деяку частину багаторазових, однак, загалом є малоефективними. Найчастіше застосовуються такі коди в асиметричних каналах зв'язку, де ймовірніше перетворення нулів в одиниці, або навпаки.

Справді, асиметрична помилка по-різному впливає на інформаційні та перевірочні символи кодової комбінації завдяки інвертуванню останніх . Розглянемо випадок, коли одиниці частіше перетворюються на нулі. Тоді при виникненні помилки в інформаційній частині кількість одиниць, що містяться в ній, зменшується і стає менше контрольної суми. У той самий час, якщо під час передачі одиниці перетворилися на нулі, то у перевірочній частини міститься більше нулів, а за рахунок інверсії контрольна сума може лише збільшуватися.

Таким чином, двійкове число, обчислене за інформаційними символами, що приймаються, завжди менше числа, утвореного інвертованими перевірочними символами. При цьому помилки, що

складаються у перетворенні одиниць у нулі, будуть виявлені. Це справедливо і для випадку, коли нулі перетворюються на одиниці.

5.9 Код Ріда-Маллера

Коди Ріда-Маллера утворюють клас двійкових систематичних кодів, що характеризуються широким спектром значень надмірності та мінімальної кодової відстані. Параметри кодів визначаються двома величинами m та $D \le m$ і пов'язані з ними в такий спосіб. Довжина коду $n=2^m$, кількість інформаційних розрядів $k = \sum_{i=0}^{D} C_m^i$, мінімальна кодова відстань $d_{\min} = 2^{m-D.}$

Приклад 5.30. Визначити характеристики n, k, r, $d_{\min}$, R кодів Ріда-Маллера для параметрів m = 3, 4, 5; D = 1… m – 1.

Рішення. Нехай m =4, D =2. У цьому випадку довжина коду $n=2^4=16$; кількість інформаційних розрядів $k = \sum_{i=0}^{2} C_4^i = C_4^0 + C_4^1 + C_4^2 = 1+4+6 = 11$;

кількість перевірочних розрядів $r=n-k=16-11=5$; відносна надмірність коду $R=r/n=11/16=0{,}6875$; мінімальна кодова відстань $d_{\min}=2^{4-2}=4$. Аналогічно розраховані характеристики для інших параметрів зведені в табл. 5.3.

Таблиця 5.3 – Характеристики кодів Ріда-Маллера

m	D	n	k	r	$d_{\min}$	R
3	1	8	4	4	4	0,5
	2		7	1	2	0,125
4	1	16	5	11	8	0,6875
	2		11	5	4	0,3125
	3		15	1	2	0,0625
5	1	32	6	26	16	0,8125
	2		16	16	8	0,5
	3		26	6	4	0,1875
	4		31	1	2	0,03125

Породжуюча матриця кодів Ріда-Маллера розмірністю $k \times n$ будується порядкóво в декілька етапів. Рядок нульового порядку) складається з $n=2^m$ одиниць. Наступні $C_m^1 = m$ рядків першого порядку, утворюють матрицю $m \times n$, як стовпці якої обрані двійкові числа від нуля до $n-1$.

Далі йдуть C_m^2 рядків векторів другого порядку, отриманих шляхом перемноження всіх можливих пар рядків першого порядку, потім C_m^3 рядків третього порядку, які є всіма можливими добутками трьох рядків першого порядку і т. д. Загальна кількість рядків, таким чином, становить

$$C_m^0 + C_m^1 + C_m^2 + C_m^3 + ... + C_m^D = \sum_{i=0}^{D} C_m^i = k .$$

Приклад 5.31. Побудувати матрицю коду Ріда-Маллера для $m = 3$, $D=2$.

Рішення. Рядок нульового порядку містить $n = 2^3 = 8$ одиниць.

$G_0 = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$.

Записуємо $m = 3$ рядків першого порядку таким чином, щоб вони склали матрицю, стовпці якої утворюють двійкові числа від нуля до семи.

$G_{11} = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$;

$G_{12} = [0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ ]$;

$G_{13} = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ ]$.

Для запису рядків другого порядку перебираємо всі попарні поелементні добутки рядків першого порядку.

$G_{21} = G_{11} \times G_{12} = [0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]$;

$G_{22} = G_{11} \times G_{13} = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 1]$;

$G_{22} = G_{12} \times G_{13} = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ ]$.

Об'єднуємо рядки нульового, першого і другого порядку в породжуючи матрицю G розмірності 7×8.

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \blacksquare$$

Приклад 5.32. Побудувати матрицю коду Ріда-Маллера для m=4, D=3.

Рішення. З табл. 5.2 визначаємо характеристики коду: n =16, k =15.

Послідовно записуємо рядки нульового, першого, другого та третього порядків G_0 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1].

Рядки першого порядку:

G_{11} = [0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1];

G_{12} = [0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1];

G_{13} = [0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1];

G_{14} = [0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1].

Рядки другого порядку (попарні добутки)

$G_{21} = G_{11} \times G_{12}$ = [0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1];

$G_{22} = G_{11} \times G_{13}$ = [0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1];

$G_{23} = G_{11} \times G_{14}$ = [0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1];

$G_{24} = G_{12} \times G_{13}$ = [0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1];

$G_{25} = G_{12} \times G_{14}$ = [0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1];

$G_{26} = G_{13} \times G_{14}$ = [0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1].

Потрійні добутки

$G_{31} = G_{11} \times G_{12} \times G_{13}$ = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1];

$G_{32} = G_{11} \times G_{12} \times G_{14}$ = [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1];

$G_{33} = G_{11} \times G_{13} \times G_{14}$ = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1];

$G_{34} = G_{12} \times G_{13} \times G_{14}$ = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1];

Об'єднуємо рядки нульового, першого, другого і третього порядку в породжуючи матрицю G розмірності 15×16

$$G = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}.$$

Оскільки код Ріда-Маллера є систематичним, побудова кодової комбінації може бути виконана відповідно до загального співвідношення (5.13). ■

Приклад 5.33. Закодувати (8, 4) кодом Ріда-Маллера інформаційну послідовність A = {1 1 0 1}.

Рішення. Використовуючи результати прикладу 12.19, будуємо матрицю коду 4×8 .

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Відповідно до (5.13) обчислюємо кодову послідовність.

$$v = A \cdot G = [1101] \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

де A – вектор рядок інформаційних символів,

v – кодова комбінація, що передається в канал зв'язку,

G – породжуюча матриця .

$v_1 = a_1 \cdot g_{1,1} \oplus a_2 \cdot g_{2,1} \oplus a_3 \cdot g_{3,1} \oplus a_4 \cdot g_{4,1} = 1 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 = 1;$

$v_2 = a_1 \cdot g_{1,2} \oplus a_2 \cdot g_{2,2} \oplus a_3 \cdot g_{3,2} \oplus a_4 \cdot g_{4,2} = 1 \cdot 1 \oplus 1 \cdot 1 \oplus 0 \cdot 0 \oplus 1 \cdot 0 = 0;$

$v_3 = a_1 \cdot g_{1,3} \oplus a_2 \cdot g_{2,3} \oplus a_3 \cdot g_{3,3} \oplus a_4 \cdot g_{4,3} = 1 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 1 \oplus 1 \cdot 0 = 1;$

$v_4 = a_1 \cdot g_{1,4} \oplus a_2 \cdot g_{2,4} \oplus a_3 \cdot g_{3,4} \oplus a_4 \cdot g_{4,4} = 1 \cdot 1 \oplus 1 \cdot 1 \oplus 0 \cdot 1 \oplus 1 \cdot 0 = 0;$

$v_5 = a_1 \cdot g_{1,5} \oplus a_2 \cdot g_{2,5} \oplus a_3 \cdot g_{3,5} \oplus a_4 \cdot g_{4,5} = 1 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 1 = 0;$

$v_6 = a_1 \cdot g_{1,6} \oplus a_2 \cdot g_{2,6} \oplus a_3 \cdot g_{3,6} \oplus a_4 \cdot g_{4,6} = 1 \cdot 1 \oplus 1 \cdot 1 \oplus 0 \cdot 0 \oplus 1 \cdot 1 = 1;$

$v_7 = a_1 \cdot g_{1,7} \oplus a_2 \cdot g_{2,7} \oplus a_3 \cdot g_{3,7} \oplus a_4 \cdot g_{4,7} = 1 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 1 \oplus 1 \cdot 1 = 0;$

$v_8 = a_1 \cdot g_{1,8} \oplus a_2 \cdot g_{2,8} \oplus a_3 \cdot g_{3,8} \oplus a_4 \cdot g_{4,8} = 1 \cdot 1 \oplus 1 \cdot 1 \oplus 0 \cdot 1 \oplus 1 \cdot 1 = 0.$

Таким чином, v = {1 0 1 0 0 1 0 1}. ■

Приклад 5.34. Закодувати (16, 11) кодом Ріда-Маллера інформаційну послідовність A = {1 1 0 0 1 0 1 0 1 0 0}.

Рішення. Використовуючи результати прикладу 5.32, будуємо породжуючу матрицю коду 11×16.

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \qquad (5.36)$$

Відповідно до (5.13) обчислюємо кодову послідовність

$v = A \cdot G$, наприклад

$v_1 = a_1 \cdot g_{1,1} \oplus a_2 \cdot g_{2,1} \oplus a_3 \cdot g_{3,1} \oplus a_4 \cdot g_{4,1} \oplus a_5 \cdot g_{5,1} \oplus a_6 \cdot g_{6,1} \oplus a_7 \cdot g_{7,1} \oplus a_8 \cdot g_{8,1}$ $\oplus \oplus a_9 \cdot g_{9,1} \oplus a_{10} \cdot g_{10,1} \oplus a_{11} \cdot g_{11,1} = 1 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \oplus$ $0 \cdot 0 \oplus \ \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 0 \cdot 0 = 1$;

$v_2 = a_1 \cdot g_{1,2} \oplus a_2 \cdot g_{2,2} \oplus a_3 \cdot g_{3,2} \oplus a_4 \cdot g_{4,2} \oplus a_5 \cdot g_{5,2} \oplus a_6 \cdot g_{6,2} \oplus a_7 \cdot g_{7,2} \oplus a_8 \cdot g_{8,2}$ $\oplus \oplus a_9 \cdot g_{9,2} \oplus a_{10} \cdot g_{10,2} \oplus a_{11} \cdot g_{11,2} = 1 \cdot 1 \oplus 1 \cdot 1 \oplus 0 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \oplus$ $0 \cdot 0 \oplus \ \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 0 \cdot 0 = 0$;

$v_3 = a_1 \cdot g_{1,3} \oplus a_2 \cdot g_{2,3} \oplus a_3 \cdot g_{3,3} \oplus a_4 \cdot g_{4,3} \oplus a_5 \cdot g_{5,3} \oplus a_6 \cdot g_{6,3} \oplus a_7 \cdot g_{7,3} \oplus a_8 \cdot g_{8,3}$ $\oplus \oplus a_9 \cdot g_{9,3} \oplus a_{10} \cdot g_{10,3} \oplus a_{11} \cdot g_{11,3} = 1 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 1 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 1 \cdot 0 \oplus$ $0 \cdot 0 \oplus \ \oplus 1 \cdot 0 \oplus 0 \cdot 0 \oplus 0 \cdot 0 = 1$.

Виконавши аналогічні обчислення для v_4 … v_{16} (пропонується перевірити самостійно), отримуємо закодовану (16, 11) кодом Ріда-Маллера послідовність

$v = \{1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\}$. ■

Шляхом перестановки стовпців породжуючу матрицю G коду Ріда-Маллера можна привести до вигляду (5.14) $G = [A|B]$, де A – одинична матриця, B – перевірочна підматриця. Після нескладних перетворень може бути побудована перевірочна матриця H та проведено процес декодування шляхом обчислення та аналізу синдрому.

Існує, проте, ефективніший метод *мажоритарного декодування* кодів Ріда-Маллера (мажоритарний – що базується на більшості, від франц . *majorité* «більшість») . Розглянемо цей метод на прикладі (16 , 11) коду Ріда-Маллера , для котрого $m = 4$, $D = 2$.

Підставимо у вирази для відліків кодової послідовності кодової послідовності значення елементів матриці, що виробляє, і отримаємо:

$v_1 = a_1;\cdot$ $\quad v_2 = a_1 \cdot \oplus\, a_2;$

$v_3 = a_1 \cdot \oplus\, a_3;$ $\quad v_4 = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_3 \cdot \oplus\, a_6;$

$v_5 = a_1 \cdot \oplus\, a_4 \cdot;$ $\quad v_6 = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_4 \cdot \oplus\, a_7;$

$v_7 = a_1 \cdot \oplus\, a_3 \cdot \oplus\, a_4 \cdot \oplus\, a_9;$ $\quad v_8 = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_3 \cdot \oplus\, a_4 \oplus a_6 \cdot \oplus\, a_7 \cdot \oplus\, a_9;$

$v_9 = a_1 \cdot \oplus\, a_5 \cdot;$ $\quad v_{10} = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_5 \cdot \oplus\, a_8;$

$v_{11} = a_1 \cdot \oplus\, a_3 \cdot \oplus\, a_5 \cdot \oplus\, a_{10};$ $\quad v_{12} = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_3 \cdot \oplus\, a_5 \oplus a_6 \cdot \oplus\, a_8 \cdot \oplus\, a_{10};$

$v_{13} = a_1 \cdot \oplus\, a_4 \cdot \oplus\, a_5 \cdot \oplus\, a_{11};$ $\quad v_{14} = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_4 \cdot \oplus\, a_5 \oplus a_7 \cdot \oplus\, a_8 \cdot \oplus\, a_{11};$

$v_{15} = a_1 \cdot \oplus\, a_3 \cdot \oplus\, a_4 \cdot \oplus\, a_5 \oplus a_9 \cdot \oplus\, a_{10} \cdot \oplus\, a_{11};$

$v_{16} = a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_3 \cdot \oplus\, a_4 \cdot \oplus\, a_5 \oplus a_6 \cdot \oplus\, a_7 \cdot \oplus\, a_8 \cdot \oplus\, a_9 \cdot \oplus\, a_{10} \cdot \oplus\, a_{11}.$

Сукупність рівнянь можна розрішити щодо v_i;

$a_1 = f(v_1, v_2, \ldots, v_i, \ldots v_{16}).$

Принцип пошуку перевірочних співвідношень полягає у виявленні таких поєднань стовпців, сума яких за модулем два міститиме лише одну одиницю в розряді. Наприклад, у матриці, що розглядається сума перших чотирьох стовпців має вигляд

$$\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} \oplus \begin{bmatrix}1\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} \oplus \begin{bmatrix}1\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} \oplus \begin{bmatrix}1\\1\\1\\0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix} \begin{matrix}v_1\\v_2\\v_3\\v_4\\v_5\\v_6\\v_7\\v_8\\v_9\\v_{10}\\v_{11}\end{matrix}. \qquad (5.37)$$

На підставі рівності (5.37) можна записати перевірочне співвідношення $a_6 = v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_3 \cdot \oplus\, v_4$.

Дійсно,

$v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_3 \cdot \oplus\, v_4 = (a_1) \oplus (a_1 \cdot \oplus\, a_2) \oplus (a_1 \cdot \oplus\, a_3) \cdot \oplus\, (a_1 \cdot \oplus\, a_2 \cdot \oplus\, a_3 \cdot \oplus\, a_6) = a_6.$

Можна показати, що кожен із інформаційних символів може бути отриманий $d = 2^{m-D}$ способами (в даному випадку $m = 4$, $D=2$, $d=2^{m-D}=4$).

Таким чином, a_6 можна отримати ще трьома способами, тобто отримати чотири незалежні рівняння (пропонується перевірити самостійно):

$a_6 = v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_3 \cdot \oplus\, v_4$; $a_6 = v_5 \cdot \oplus\, v_6 \cdot \oplus\, v_7 \cdot \oplus\, v_8$;

$a_6 = v_9 \cdot \oplus\, v_{10} \cdot \oplus\, v_{11} \cdot \oplus\, v_{12}$; $a_6 = v_{13} \cdot \oplus\, v_{14} \cdot \oplus\, v_{15} \cdot \oplus\, v_6$.

Зазначимо, що відлік a_6 відповідає рядку породжуючої матриці $G_{21}=G_{11}\times G_{12}$. Аналогічно можна побудувати співвідношення і для інших відліків, що визначаються попарними добутками рядків першого порядку в базовій матриці

$$
\begin{aligned}
&G_{21} = G_{11} \times G_{12} \sim a_6 = v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_3 \cdot \oplus\, v_4 = v_5 \cdot \oplus\, v_6 \cdot \oplus\, v_7 \cdot \oplus\, v_8 = \\
&= v_9 \cdot \oplus\, v_{10} \cdot \oplus\, v_{11} \cdot \oplus\, v_{12} = v_{13} \cdot \oplus\, v_{14} \cdot \oplus\, v_{15} \cdot \oplus\, v_6; \\
&G_{22} = G_{11} \times G_{13} \sim a_7 = v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_5 \cdot \oplus\, v_6 = v_3 \cdot \oplus\, v_4 \cdot \oplus\, v_7 \cdot \oplus\, v_8 = \\
&= v_9 \cdot \oplus\, v_{10} \cdot \oplus\, v_{13} \cdot \oplus\, v_{14} = v_{11} \cdot \oplus\, v_{12} \cdot \oplus\, v_{15} \cdot \oplus\, v_{16}; \\
&G_{23} = G_{11} \times G_{14} \sim a_8 = v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_9 \cdot \oplus\, v_{10} = v_3 \cdot \oplus\, v_4 \cdot \oplus\, v_{11} \cdot \oplus\, v_{12} = \\
&= v_5 \cdot \oplus\, v_6 \cdot \oplus\, v_{13} \cdot \oplus\, v_{14} = v_7 \cdot \oplus\, v_8 \cdot \oplus\, v_{15} \cdot \oplus\, v_{16}; \qquad (5.38) \\
&G_{24} = G_{12} \times G_{13} \sim a_9 = v_1 \cdot \oplus\, v_3 \cdot \oplus\, v_5 \cdot \oplus\, v_7 = v_2 \cdot \oplus\, v_4 \cdot \oplus\, v_6 \cdot \oplus\, v_8 = \\
&= v_9 \cdot \oplus\, v_{11} \cdot \oplus\, v_{13} \cdot \oplus\, v_{15} = v_{10} \cdot \oplus\, v_{12} \cdot \oplus\, v_{14} \cdot \oplus\, v_{16}; \\
&G_{25} = G_{12} \times G_{14} \sim a_{10} = v_1 \cdot \oplus\, v_3 \cdot \oplus\, v_9 \cdot \oplus\, v_{11} = v_2 \cdot \oplus\, v_4 \cdot \oplus\, v_{10} \cdot \oplus\, v_{12} = \\
&= v_5 \cdot \oplus\, v_7 \cdot \oplus\, v_{13} \cdot \oplus\, v_{15} = v_6 \cdot \oplus\, v_8 \cdot \oplus\, v_{14} \cdot \oplus\, v_{16}; \\
&G_{26} = G_{13} \times G_{14} \sim a_{11} = v_1 \cdot \oplus\, v_5 \cdot \oplus\, v_9 \cdot \oplus\, v_{13} = v_2 \cdot \oplus\, v_6 \cdot \oplus\, v_{10} \cdot \oplus\, v_{14} = \\
&= v_3 \cdot \oplus\, v_7 \cdot \oplus\, v_{11} \cdot \oplus\, v_{15} = v_4 \cdot \oplus\, v_8 \cdot \oplus\, v_{12} \cdot \oplus\, v_{16}.
\end{aligned}
$$

За цими співвідношеннями на підставі розрядів кодової комбінації v_k можуть бути відновлені символи вихідної послідовності a_i . Помилки, що виникли під час передачі, можуть зробити деякі зі співвідношень (5.38) несправедливими, але кожна помилка впливає тільки на одне із співвідношень. Отже, якщо вибирати a_i рівним величині, обчисленої в більшості зі співвідношень, то при будь-якій комбінації з 2^{m-D-1} –1 (в даному випадку 2^{4-2-1}–1=1) помилок значення a_i буде декодовано правильно.

Визначати інформаційні символи a_1–a_5, що відповідають рядкам першого порядку породжуючої матриці G за формулами , аналогічними (5.38), не можна, оскільки при складанні стовпців одиниці неминуче з’являються у кількох розрядах. Тому закодоване повідомлення необхідно

перетворити так, щоб для нього були справедливі співвідношення векторів першого порядку. Для цього до закодованого повідомлення v додають за модулем два домножені на раніше визначені значення a_i рядки другого порядку G_{21}-G_{26}, утворені попарним перемноженням рядків першого порядку

$$v' = v \oplus a_6 \cdot G_{21} \oplus a_7 \cdot G_{22} \oplus a_8 \cdot G_{23} \oplus a_9 \cdot G_{24} \oplus a_{10} \cdot G_{25} \oplus a_{10} \cdot G_{26}. \qquad (5.39)$$

Складаємо рівняння, аналогічні (5.38), але відповідні рядкам першого порядку породжуючої матриці G.

$$\begin{aligned}
&G_{11} \sim a_2 = v'_1 \cdot \oplus\, v'_2 \cdot = v'_3 \cdot \oplus\, v'_4 = v'_5 \cdot \oplus\, v'_6 \cdot = v'_7 \cdot \oplus\, v'_8 = v'_9 \cdot \oplus\, v'_{10} \cdot = \\
&= v'_{11} \cdot \oplus\, v'_{12} = v'_{13} \cdot \oplus\, v'_{14} \cdot = v'_{15} \cdot \oplus\, v'_{16}; \\
&G_{12} \sim a_3 = v'_1 \cdot \oplus\, v'_3 \cdot = v'_2 \cdot \oplus\, v'_4 = v'_5 \cdot \oplus\, v'_7 \cdot = v'_6 \cdot \oplus\, v'_8 = v'_9 \cdot \oplus\, v'_{11} \cdot = \\
&= v'_{10} \cdot \oplus\, v'_{12} = v'_{13} \cdot \oplus\, v'_{15} \cdot = v'_{14} \cdot \oplus\, v'_{16}; \\
&G_{13} \sim a_4 = v'_1 \cdot \oplus\, v'_5 \cdot = v'_2 \cdot \oplus\, v'_6 = v'_3 \cdot \oplus\, v'_7 \cdot = v'_4 \cdot \oplus\, v'_8 = v'_9 \cdot \oplus\, v'_{13} \cdot = \\
&= v'_{10} \cdot \oplus\, v'_{14} = v'_{11} \cdot \oplus\, v'_{15} \cdot = v'_{12} \cdot \oplus\, v'_{16}; \\
&G_{14} \sim a_5 = v'_1 \cdot \oplus\, v'_9 \cdot = v'_2 \cdot \oplus\, v'_{10} = v'_3 \cdot \oplus\, v'_{11} \cdot = v'_4 \cdot \oplus\, v'_{13} = v'_5 \cdot \oplus\, v'_{13} \cdot = \\
&= v'_6 \cdot \oplus\, v'_{14} = v'_7 \cdot \oplus\, v'_{15} \cdot = v'_8 \cdot \oplus\, v'_{16}.
\end{aligned} \qquad (5.40)$$

Знову застосовуємо мажоритарний алгоритм, тобто надаємо $a_2 \ldots a_5$ ті значення , які виникають у перевірочних співвідношеннях частіше.

Для визначення a_1 необхідно обчислити добутки $a_2 \ldots a_5$ на відповідні їм рядки першого порядку породжуючої матриці і додати ці вектори за модулем два до раніше перетвореної послідовності v'

$$v'' = v' \oplus a_2 \cdot G_{11} \oplus a_3 \cdot G_{12} \oplus a_4 \cdot G_{13} \oplus a_5 \cdot G_{14}. \qquad (5.41)$$

Цей вектор повинен бути нульовим , якщо a_1=0, і одиничним, якщо a_1=1, тому потрібно вибирати a_1 рівним тому із значень 0, 1, яке зустрічається у послідовності частіше.

Аналогічний алгоритм може бути побудований для довільних значень m та D. Спочатку записуються рівняння для a_i, відповідних векторам найвищого порядку D, потім – порядку $(D-1)$ і т.д.

Приклад 5.35. Декодувати закодовану (16, 11) кодом Ріда-Маллера інформаційну послідовність

Y = {1 0 1 0 1 1 0 0 0 1 1 1 0 0 1 1}.

Рішення. Відповідно до табл. (5.3) код Ріда-Маллера забезпечує мінімальну кодову відстань $d_{\min}$=4, тобто дозволяє виявляти подвійні та виправляти поодинокі помилки. Визначимо інформаційні символи $a_6 - a_{11}$, що відповідають рядкам другого порядку породжуючої матриці G за формулами (5.38).

Обчислимо, наприклад, символ a_6, відповідний рядку $G_{21} = G_{11} \times G_{12}$ породжуючої матриці, чотирма незалежними способами на підставі рівнянь (5.38):

$a_{61} = v_1 \cdot \oplus\, v_2 \cdot \oplus\, v_3 \cdot \oplus\, v_4 = 1 \cdot \oplus\, 0 \cdot \oplus\, 1 \cdot \oplus\, 0 = 0;$

$a_{62} = v_5 \cdot \oplus\, v_6 \cdot \oplus\, v_7 \cdot \oplus\, v_8 = 1 \cdot \oplus\, 1 \cdot \oplus\, 0 \cdot \oplus\, 0 = 0;$

$a_{63} = v_9 \cdot \oplus\, v_{10} \cdot \oplus\, v_{11} \cdot \oplus\, v_{12} = 0 \cdot \oplus\, 1 \cdot \oplus\, 1 \cdot \oplus\, 1 = 1;$

$a_{64} = v_{13} \cdot \oplus\, v_{14} \cdot \oplus\, v_{15} \cdot \oplus\, v_6 = 0 \cdot \oplus\, 0 \cdot \oplus\, 1 \cdot \oplus\, 1 = 0.$

Значення a_6, обчислене трьома способами, виявилося рівним нулю, четвертим способом – рівним одиниці . Застосовуючи критерій більшості , остаточно встановлюємо $a_6 = 0$.

Аналогічно розраховуємо значення $a_7 - a_{11}$ і зводимо результати обчислень у табл. 5.4.

Таблиця 5.4 – Обчислення символів $a_6 - a_{11}$

Номер символу i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	Результат голосування
6	0	0	1	0	$a_6 = 0$
7	1	1	1	0	$a_7 = 1$
8	0	1	0	0	$a_8 = 0$
9	1	1	0	1	$a_9 = 1$
10	1	0	0	0	$a_{10} = 0$
11	0	0	1	0	$a_{11} = 0$

Для визначення інформаційних з символів $a_2 - a_5$, відповідних рядкам першого порядку породжуючої матриці G відповідно до (5.39) до закодованого повідомленню v додаємо за модулем два домножені на

раніше обчислені значення a_i рядки другого порядку $G_{21} – G_{26}$, визначені у прикладі 5.31.

v	1 0 1 0 1 1 0 0 0 1 1 1 0 0 1 1
$a_6 \cdot G_{21}$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$a_7 \cdot G_{22}$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$a_8 \cdot G_{23}$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$a_9 \cdot G_{24}$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$a_{10} \cdot G_{25}$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$a_{11} \cdot G_{25}$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
v'	1 0 1 0 1 0 1 0 0 1 1 1 0 1 0 1

Визначимо інформаційні символи $a_2 – a_5$, що відповідають рядкам першого порядку породжуючої матриці G за формулами (5.40).

Обчислимо, наприклад, символ a_2 відповідний рядку G_{11} породжуючої матриці вісьмома незалежними способами на підставі рівнянь (5.40)

$a_{21} = v'_1 \cdot \oplus\, v'_2 \cdot = 1 \oplus 0 = 1;$ $a_{22} = v'_3 \cdot \oplus\, v'_4 = 1 \oplus 0 = 1;$

$a_{23} = v'_5 \cdot \oplus\, v'_6 \cdot = 1 \oplus 0 = 1;$ $a_{24} = v'_7 \cdot \oplus\, v'_8 = 1 \oplus 0 = 1;$

$a_{25} = v'_9 \cdot \oplus\, v'_{10} \cdot = 0 \oplus 1 = 1;$ $a_{26} = v'_{11} \cdot \oplus\, v'_{12} \cdot = 1 \oplus 1 = 0;$

$a_{27} = v'_{13} \cdot \oplus\, v'_{14} \cdot\cdot = 0 \oplus 1 = 1;$ $a_{28} = v'_{15} \cdot \oplus\, v'_{16} \cdot = 0 \oplus 1 = 1.$

Значення a_2, обчислене сімома способами, виявилося рівним одиниці, восьмим способом – рівним нулю. Застосовуючи критерій більшості, остаточно встановлюємо $a_2 = 1$.

Аналогічно розраховуємо значення $a_3 – a_5$ і зводимо результати обчислень у табл. 5.5.

Таблиця 5.5 – Обчислення символів $a_2 – a_5$

Номер символу i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}	a_{i8}	Результат голосування
2	1	1	1	1	1	0	1	1	$a_2 = 1$
3	0	0	0	0	1	0	0	0	$a_3 = 0$
4	0	0	0	0	0	0	1	0	$a_4 = 0$
5	1	1	0	1	1	1	1	1	$a_5 = 1$

Для визначення інформаційного символу a_1 , відповідного рядку нульового порядку породжуючої матриці G відповідно до (5.41) до раніше перетвореної послідовності v' додаємо за модулем два домножені раніше обчислені значення a_i на рядки першого порядку $G_{11} - G_{14}$, визначені у прикладі 5.31.

v' 1 0 1 0 1 0 1 0 0 1 1 1 0 1 0 1

$a_2 \cdot G_{11}$ 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

$a_3 \cdot G_{12}$ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

$a_4 \cdot G_{13}$ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

$a_5 \cdot G_{14}$ 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

v'' 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1

У векторі v'' переважають одиниці (15 проти одного нуля), тому a_1 присвоюється одиничне значення, $a_1 = 1$.

Об'єднуючи дані табл. 5.4, 5.5 та обчислене значення a_1, отримуємо відновлене значення вихідний послідовності $A = \{1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\}$. ■

Порівнюючи отриманий результат з даними прикладу 5.34 виявляємо, що декодована послідовність дійсно була результатом кодування послідовності $A=\{1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\}$ із внесеною помилкою в одинадцятому розряді. ■

Приклад 5.36. Декодувати закодовану (16, 11) кодом Ріда-Маллера інформаційну послідовність

$Y = \{1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\}$.

Рішення. Визначимо інформаційні символи $a_6 - a_{11}$, що відповідають рядкам другого порядку породжуючої матриці G за формулами (5.37).

Результати обчислень зведемо у табл. 5.6.

Таблиця 5.6 – Обчислення символів $a_6 - a_{11}$

Номер символу i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	Результат голосування
6	1	0	1	0	a_6 = ?
7	1	0	0	1	a_7 = ?
8	1	1	0	0	a_8 = ?
9	0	1	1	0	a_9 = ?
10	1	1	0	0	a_{10} = ?
11	0	1	1	0	a_{11} = ?

Аналіз табл. 5.6 показує, що при обчисленні всіх символів голоси розділилися навпіл, і прийняти певне рішення неможливо. Звідси робимо висновок, що сталася подвійна помилка чи помилка вищої кратності. Код з $d_{\min}=4$ здатний виявити таку помилку поряд з виправленням одноразових, але не здатний її виправити.

Дійсно, як випливає з прикладу 5.33, декодована послідовність була результатом кодування послідовності $A=\{1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\}$ із внесеними помилками у третьому та десятому розрядах.

5.8 Код Гемінґа

Коди Гемінґа відносяться до числу систематичних роздільних кодів. Існує два різновиди кодів Гемінґа – коди з $d_{\min}=3$, виправляючі всі поодинокі або виявляючі подвійні помилки, і так звані *розширені коди Гемінґа* з $d_{\min}=4$, що виправляють поодинокі та виявляють подвійні помилки.

Цікавою є історія появи цих кодів. З 1946 року Річард Гемінґ працював у компанії *Bell Labs*, де виконував розрахунки в рамках проекту «Манхеттен» на електромеханічному комп'ютері *Bell Model V*. Дані вводилися в машину за допомогою ненадійних пристроїв читання перфокарт і в процесі введення часто відбувалися помилки. Ці збої суттєво уповільнювали обчислення, тому протягом кількох років Гемінґ шукав ефективний алгоритм виправлення помилок. У 1950 році він запропонував спосіб кодування, який нині відомий як код Гемінґа.

Для звичайного коду Гемінґа з $d_{\min}=3$ повинно виконуватись співвідношення

$$n \le 2^r - 1, \tag{5.42}$$

тобто

$$r \ge \log_2 (n + 1) \tag{5.43}$$

$$k \le 2^r - 1 - r, \tag{5.44}$$

де k – число інформаційних розрядів коду, r – число перевірочних та n – загальна кількість розрядів.

Порівняємо (5.42), (5.43) з оцінками межі Гемінґа (5.9). Відповідно до (5.9) при $d_{\min}=3$, $t=1$ і

$$r \geq \log_2\left(1+\sum_{i=1}^{1} C_n^i = 1+n\right).$$

Таким чином, якщо у (5.42) має місце точна рівність, то код Гемінґа є досконалим.

Як буде показано нижче, при застосуванні розширеного коду Гемінґа з $d_{\min}=4$ до нього додається один перевірочний розряд.

У табл. 5.7 зведені параметри найчастіше вживаних кодів Гемінґа.

Таблиця 5.7 – Параметри деяких кодів Гемінґа

	k	4	11	26	57	120	247
Звичайний код Гемінґа $d_{\min}=3$	r	3	4	5	6	7	8
	n	7	15	31	63	127	255
	$R=r/n$	0,429	0,267	0,161	0,095	0,055	0,031
Розширений код Гемінґа $d_{\min}=4$	R	4	5	6	7	8	9
	N	8	16	32	64	128	256
	$R=r/n$	0,5	0,313	0,188	0,109	0,063	0,035

Кодування та декодування кодами Гемінґа може проводитися за допомогою породжуючої та перевірочної матриці виду (5.14) та (5.19), однак з метою спрощення апаратної та програмної реалізації кодеків Гемінґ запропонував використовувати таке розташування стовпців перевірочної матриці, щоб номер i-го стовпця матриці відповідав двійковому представленню числа i. В цьому випадку розряди синдрому, отримані з перевірочних рівнянь, утворюють двійкове представлення номера розряду комбінації, у якому сталася помилка.

При кодуванні інформації перевірочні розряди розташовуються не в кінці кодової комбінації, а на позиціях, номери яких виражаються ступенями двійки (2^0, 2^1, 2^2 ... 2^{r-1}), тобто на першій, другій, четвертій, восьмій і т.д. позиціях. Інформаційні розряди розміщуються на позиціях , що залишилися , для яких в стовпці перевірочної матриці міститься більше однієї одиниці.

Для обчислення k-го перевірочного розряду за модулем два сумуються ті розряди послідовності, що кодується, для яких в k-му рядку перевірочної матриці містяться одиниці.

Приклад 5.37. Закодувати (7, 4) кодом Гемінґа інформаційну послідовність $A = \{1\ 1\ 0\ 1\}$.

Рішення. Побудуємо перевірочну матрицю кодування (7, 4) коду Гемінґа. Для (7, 4) коду кількість інформаційних розрядів 4 загальна кількість розрядів 7. Число перевірочних розрядів $r=n-k=7-4=3$. Розмірність матриці $r \times n$, тобто 3×7. У стовпці матриці записуємо двійкові представлення чисел від 1 до 7

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (5.45)$$

Формуємо розряди коду Гемінґа. Перевірочні розряди розміщуються на позиціях, номери яких виражаються ступенями двійки, тобто на першій, другій та четвертій позиціях. Інформаційні розряди розміщуються на позиціях, для яких у стовпці перевірочної матриці міститься більше однієї одиниці, тобто позиціях 3, 5, 6, 7 . Таким чином, закодована послідовність матиме вигляд

$v = \{c_1\ c_2\ a_1\ c_2\ a_2\ a_3\ a_4\}$.

Формуємо інформаційні розряди коду Гемінґа.

$v_3 = a_1 = 1$, $v_5 = a_2 = 1$, $v_6 = a_3 = 0$, $v_7 = a_4 = 1$.

Для отримання перевірочних розрядів підсумовуємо за модулем два розряди коду, для яких у відповідному рядку перевірочної матриці містяться одиниці.

$v_1 = c_1 = v_3 \oplus v_5 \oplus v_7 = a_1 \oplus a_2 \oplus a_4 = 1 \oplus 1 \oplus 1 = 1$;

$v_2 = c_2 = v_3 \oplus v_6 \oplus v_7 = a_1 \oplus a_3 \oplus a_4 = 1 \oplus 0 \oplus 1 = 0$;

$v_4 = c_3 = v_5 \oplus v_6 \oplus v_7 = a_2 \oplus a_3 \oplus a_4 = 1 \oplus 0 \oplus 1 = 0$.

Таким чином, (7, 4) код Гемінґа має вигляд

$v = \{v_1\ v_2 \ldots v_7\} = \{1\ 0\ 1\ 0\ 1\ 0\ 1\}$. ■

Приклад 5.38. Закодувати (15, 11) кодом Гемінґа інформаційну послідовність $A = \{0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\}$.

Рішення. Для цього коду $n = 15$, $k = 11$, $r = n - k = 4$. Побудуємо перевірочну матрицю 4×15. У стовпці матриці записуємо двійкові представлення чисел від 1 до 15

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (5.46)$$

Перевірочні розряди розміщуються на позиціях, номери яких виражаються ступенями двійки, тобто позиціях 1, 2, 4, 8. Інформаційні розряди – на позиціях 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15.

Формуємо інформаційні розряди коду Гемінґа.

$v_3 = a_1 = 0$, $v_5 = a_2 = 0$, $v_6 = a_3 = 1$, $v_7 = a_4 = 0$, $v_9 = a_5 = 1$, $v_{10} = a_6 = 0$, $v_{11} = a_7 = 1$, $v_{12} = a_8 = 1$, $v_{13} = a_9 = 1$, $v_{14} = a_{10} = 0$, $v_{15} = a_{11} = 1$.

Формуємо перевірочні розряди коду Гемінґа $c_1 \ldots c_4$.. Для отримання перевірочних розрядів підсумовуємо за модулем два розряди коду, для яких у відповідному рядку перевірочної матриці містяться одиниці.

$c_1 = v_1 = v_3 \oplus v_5 \oplus v_7 \oplus v_9 \oplus v_{11} \oplus v_{13} \oplus v_{15} = a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7 \oplus a_9 \oplus a_{11} = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$;

$c_2 = v_2 = v_3 \oplus v_6 \oplus v_7 \oplus v_{10} \oplus v_{11} \oplus v_{14} \oplus v_{15} = a_1 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_7 \oplus a_{10} \oplus a_{11} = = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$;

$c_3 = v_4 = v_5 \oplus v_6 \oplus v_7 \oplus v_{12} \oplus v_{13} \oplus v_{14} \oplus v_{15} = a_2 \oplus a_3 \oplus a_4 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{11} = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$;

$c_4 = v_8 = v_9 \oplus v_{10} \oplus v_{11} \oplus v_{12} \oplus v_{13} \oplus v_{14} \oplus v_{15} = a_5 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{11} = = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$.

Поєднуючи інформаційні та перевірочні розряди, остаточно отримуємо: $v = \{v_1\, v_2 \ldots v_{15}\} = \{0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\}$. ■

Декодування кодів Гемінґа відбувається, як у всіх систематичних кодів, з урахуванням співвідношення (5.18). Відповідно до правил матричного множення при обчисленні розряди синдрому підсумовуються за модулем два ті розряди декодованої послідовності y_k, для яких у відповідному рядку перевірочної матриці містяться одиниці.

Спеціальний вид перевірочних матриць (5.45), (5.46) та аналогічних їм дозволяє відразу ж будувати з розрядів синдрому двійковий код номера

спотвореного розряду. Після цього значення виявленого спотвореного розряду може бути виправлено протилежне.

Приклад 5.39. Декодувати закодовану (7, 4) кодом Гемінґа інформаційну послідовність Y = {1 0 0 1 0 1 1}.

Рішення. Обчислюємо розряди синдрому, підставляючи в (5.18) перевірочну матрицю (7, 4) коду:

$$S = Y \cdot H^T = [1\,0\,0\,1\,0\,1\,1] \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

$s_1 = y_4 \oplus y_5 \oplus y_6 \oplus y_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1;$

$s_2 = y_2 \oplus y_3 \oplus y_6 \oplus y_7 = 0 \oplus 0 \oplus 1 \oplus 1 = 0;$

$s_3 = y_1 \oplus y_3 \oplus y_5 \oplus y_7 = 1 \oplus 0 \oplus 0 \oplus 1 = 0.$

Записуємо синдром та визначаємо його чисельне значення.

$S = \{s_1\ s_2\ s_3\} = \{1\ 0\ 0\} = 4.$

Оскільки не всі розряди синдрому дорівнюють нулю, сталася помилка, номер якої визначається значенням синдрому. S = 4, тому спотворено четвертий біт. Виправляємо спотворене значення четвертого біта з одиниці на нуль, $y_{4\text{випр}}$=0.

Записуємо виправлене повідомлення $Y_{\text{випр}}$ = {1 0 0 0 0 1 1}.

Виділяємо інформаційні біти, які за прикладом 12.24 розміщуються на позиціях 3, 5, 6, 7. $A = \{y_3 y_5 y_6 y_7\} = \{0\ 0\ 1\ 1\}$. ■

Приклад 5.40. Декодувати закодовану (15, 11) кодом Гемінґа інформаційну послідовність Y = {1 1 1 0 1 0 0 1 1 0 1 1 0 0 1}.

Рішення. Обчислюємо розряди синдрому, підставляючи в (5.18) перевірочну матрицю (15, 11) коду приклада (5.40). Для цього підсумовуємо за модулем два ті розряди коду y_k, для яких у відповідному рядку перевірочної матриці містяться одиниці

$s_1 = y_8 \oplus y_9 \oplus y_{10} \oplus y_{11} \oplus y_{12} \oplus y_{13} \oplus y_{14} \oplus y_{15} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1;$

$s_2 = y_4 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_{12} \oplus y_{13} \oplus y_{14} \oplus y_{15} = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1;$

$s_3 = y_2 \oplus y_3 \oplus y_6 \oplus y_7 \oplus y_{10} \oplus y_{11} \oplus y_{14} \oplus y_{15} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0;$

$s_4 = y_1 \oplus y_3 \oplus y_5 \oplus y_7 \oplus y_9 \oplus y_{11} \oplus y_{13} \oplus y_{15} = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0.$

Записуємо синдром та визначаємо його чисельне значення

$S = \{s_1\ s_2\ s_3\ s_4\} = \{1\ 1\ 0\ 0\} = 12.$

Оскільки не всі розряди синдрому дорівнюють нулю, сталася помилка, номер якої визначається значенням синдрому. S=12, тому спотворений дванадцятий біт. Виправляємо спотворене значення біта y_4 з одиниці на нуль, $y_{4\text{випр}} = 0$.

Записуємо виправлене повідомлення

$Y_{\text{випр}} = \{1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\}.$

Виділяємо інформаційні біти, які за прикладом 5.37 розміщуються на позиціях 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15.

$A = \{y_3\ y_5\ y_6\ y_7\ y_9\ y_{10}\ y_{11}\ y_{12}\ y_{13}\ y_{14}\ y_{15}\} = \{1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\}.$ ■

Розширений код Гемінґа з $d_{\min} = 4$ будується на базі коду Гемінґа з $d_{\min} = 3$ шляхом додавання додаткового контрольного символу перевірки на парність закодованої послідовності. Такий код забезпечує виправлення всіх одноразових помилок та виявлення всіх дворазових помилок та помилок непарної кратності.

Приклад 5.41. Закодувати розширеним (8, 4) кодом Гемінґа інформаційну послідовність $A = \{0\ 1\ 0\ 1\}$.

Рішення. На першому етапі закодуємо задану послідовність звичайним (7, 4) кодом Гемінґа. Формуємо інформаційні розряди коду. Як було показано у прикладі 12.24, інформаційні розряди розміщуються на позиціях позиціях 3, 5, 6,7

$v_3 = a_1 = 0,\ v_5 = a_2 = 1,\ v_6 = a_3 = 0,\ v_7 = a_4 = 1.$

Перевірочні розряди розташовуються на першій, другій та четвертій позиціях. Обчислюємо перевірочні розряди на основі отриманих у прикладі 5.36 співвідношення

$v_1 = c_1 = v_3 \oplus v_5 \oplus v_7 = a_1 \oplus a_2 \oplus a_4 = 0 \oplus 1 \oplus 1 = 0;$

$v_2 = c_2 = v_3 \oplus v_6 \oplus v_7 = a_1 \oplus a_3 \oplus a_4 = 0 \oplus 0 \oplus 1 = 1;$

$v_4 = c_3 = v_5 \oplus v_6 \oplus v_7 = a_2 \oplus a_3 \oplus a_4 = 1 \oplus 0 \oplus 1 = 0.$

Таким чином, (7, 4) код Гемінґа має вигляд

$v = \{v_1\ v_2 \ldots v_7\} = \{0\ 1\ 0\ 0\ 1\ 0\ 1\}.$

Для формування розширеного (8, 4) коду Гемінґа, з $d_{\min} = 4$ код Гемінґа (7, 4) необхідно доповнити бітом контролю на парність

$p = y_8 = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1.$

Тоді розширений (8, 4) код Гемінґа $Y_{\text{рх}} = \{0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\}.$ ■

Декодування розширеного коду Гемінґа відбувається у зворотній послідовності в два етапи. Спочатку відкидається біт контролю на парність та обчислюються розряди синдрому S коду з $d_{\min}$ =3. Далі проводиться контроль на парність всієї прийнятої комбінації та обчислюється біт контролю на парність (біт паритету) p.

При цьому можуть виникнути такі варіанти.

1. Обидві перевірки дають нульові синдроми, $S = 0$, $p = 0$. У цьому випадку повідомлення не спотворене, виправлення не потребує.

2. Обидві перевірки дають ненульові синдроми, $S \neq 0$, $p \neq 0$. Сталася поодинока помилка, двійковий код синдрому вказує номер спотвореного розряду.

3. Синдром коду з $d_{\min}$ =3 містить ненульові розряди, а біт контролю на парність нульовий, $S \neq 0$, $p = 0$. У цьому випадку відбулася непоправна помилка парної кратності, наприклад, подвійна.

4. Усі розряди синдрому коду з $d_{\min}$ =3 дорівнюють нулю, розряд контролю на парність одиничний, $S = 0$, $p \neq 0$. Або сталася помилка в біті паритету, або потрійна помилка чи помилка вищої непарної кратності.

Приклад 5.42 Декодувати закодовану розширеним (8, 4) кодом Гемінґа інформаційну послідовність Y = {0 1 0 0 0 0 1 1}.

Рішення. Відділимо крайній правий біт контролю на парність та проведемо декодування семибітного повідомлення Y = {0 1 0 0 0 0 1} кодом Гемінґа (7,4).

Обчислюємо розряди синдрому аналогічно до прикладу 5.40

$$s_1 = y_4 \oplus y_5 \oplus y_6 \oplus y_7 = 0 \oplus 0 \oplus 0 \oplus 1 = 1;$$

$$s_2 = y_2 \oplus y_3 \oplus y_6 \oplus y_7 = 1 \oplus 0 \oplus 0 \oplus 1 = 0;$$

$$s_3 = y_1 \oplus y_3 \oplus y_5 \oplus y_7 = 0 \oplus 0 \oplus 0 \oplus 1 = 1.$$

Записуємо синдром та визначаємо його чисельне значення

$S = \{s_1\, s_2\, s_3\} = \{1\ 0\ 1\} = 5$.

Проводимо контроль на парність прийнятого вихідного восьмибітного повідомлення, навіщо сумуємо за модулем два всі розряди

$$s_4 = p = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_8 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1.$$

Таким чином, синдром та біт контролю на парність мають ненульове значення. В цьому випадку сталася поодинока помилка, номер якої визначається значенням синдрому. S=5, тому спотворений п'ятий біт. Виправляємо спотворене значення п'ятого біта з нуля на одиницю, $y_{5\text{випр}}$=1.

Записуємо виправлене повідомлення $Y_{\text{випр}}$ = {0 1 0 0 1 0 1 1}. ■

Приклад 5.43. Декодувати закодовану розширеним (8, 4) кодом Гемінґа інформаційну послідовність Y = {0 0 1 0 0 0 1 0}.

Рішення. Відділимо крайній правий біт контролю на парність та проведемо декодування семибітного повідомлення Y = {0 0 1 0 0 0 1} кодом Гемінґа (7,4).

Обчислюємо розряди синдрому аналогічно до прикладу 5.39

$$s_1 = y_4 \oplus y_5 \oplus y_6 \oplus y_7 = 0 \oplus 0 \oplus 0 \oplus 1 = 1;$$

$$s_2 = y_2 \oplus y_3 \oplus y_6 \oplus y_7 = 0 \oplus 1 \oplus 0 \oplus 1 = 0;$$

$$s_3 = y_1 \oplus y_3 \oplus y_5 \oplus y_7 = 0 \oplus 1 \oplus 0 \oplus 1 = 0.$$

Записуємо синдром та визначаємо його чисельне значення

$S = \{s_1\, s_2\, s_3\} = \{1\ 0\ 0\} = 4$.

Проводимо контроль на парність прийнятого вихідного восьмибітного повідомлення, навіщо для чого сумуємо за модулем два всі розряди

$$s_4 = p = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_8 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0.$$

Синдром містить ненульовий розряд, біт контролю на парність дорівнює нулю. $S \neq 0$, p=0. Робимо висновок, що сталася непоправна помилка парної кратності, найімовірніше, подвійна. ■

У деяких випадках немає необхідності у виконанні суворих рівностей (5.42)-(5.44) і тоді застосовуються так звані *усічені* (або *укорочені*) коди Гемінґа. Укорочений код формується з вихідного систематичного шляхом вибору лише таких кодових слів, наприкінці яких стоять l нульових символів. Оскільки нульові символи не несуть жодної інформації, довжину коду можна зменшити на l символів з одночасним скороченням кількості інформаційних символів та кодових слів. Результуючий ($n–l$, $k–l$) код буде мати ту ж кодову відстань, що і вихідний. Легко показати, що породжуючи матриця G' укороченого коду виходить з канонічної матриці G вихідного коду видаленням з неї l рядків і стовпців, а перевірочна H' – шляхом видалення з матриці H вихідного коду l правих стовпців.

Так, наприклад, якщо необхідно виправляти поодинокі помилки в міжнародному телеграфному коді МТК-2, що має п'ять інформаційних розрядів, згідно з (5.44) потрібно чотири перевірочні символи і, таким чином, доцільно застосовувати (9, 5) код, що є усіченим від класичного коду Гемінґа (15, 11). Число інформаційних символів та загальна кількість символів у цьому коді зменшується на шість.

Приклад 5.44. Закодувати усіченим (9, 5) кодом Гемінґа інформаційну послідовність A = {1 0 0 1 1}.

Рішення. Виключаємо із наведених у прикладі 5.38 співвідношень для (15,11) кодів Гемінґа символи $a_6 - a_{11}$.

Тоді інформаційні розряди (9, 5) коду Гемінґа:

$v_3 = a_1 = 1$, $v_5 = a_2 = 0$, $v_6 = a_3 = 0$, $v_7 = a_4 = 1$, $v_9 = a_5 = 1$.

Перевірочні розряди коду *Гемінґа* c_1 … c_4. Для отримання перевірочних розрядів підсумовуємо за модулем два розряди коду, для яких у відповідному рядку перевірочної матриці містяться одиниці

$c_1 = v_1 = a_1 \oplus a_2 \oplus a_4 \oplus a_5 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$;

$c_2 = v_2 = a_1 \oplus a_3 \oplus a_4 = 1 \oplus 0 \oplus 1 = 0$;

$c_3 = v_4 = a_2 \oplus a_3 \oplus a_4 = 0 \oplus 0 \oplus 1 = 1$;

$c_4 = v_8 = a_5 \oplus = 1$.

Поєднуючи інформаційні та перевірочні розряди, остаточно отримуємо: v = {v_1 v_2 ... v_9} = {1 0 1 1 0 0 1 1 1}. ■

Приклад 5.45. Декодувати закодовану усіченим (9, 5) кодом Гемінґа інформаційну послідовність Y = {1 0 1 1 0 1 1 1 1}.

Рішення. Після видалення шести правих стовпців перевірочна матриця H' усіченого (9, 5) коду матиме вигляд

$$H' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (5.47)$$

Обчислюємо розряди синдрому, підставляючи в (5.18) отриману перевірочну матрицю (5.46)

$s_1 = y_8 \oplus y_9 = 1 \oplus 1 = 0$;

$s_2 = y_4 \oplus y_5 \oplus y_6 \oplus y_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$;

$s_3 = y_2 \oplus y_3 \oplus y_6 \oplus y_7 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$;

$s_4 = y_1 \oplus y_3 \oplus y_5 \oplus y_7 \oplus y_9 \oplus y_{11} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$.

Записуємо синдром та визначаємо його чисельне значення

S={s_1 s_2 s_3 s_4}= {0 1 1 0} = 6.

Виправляємо спотворене значення біта y_6 з одиниці на нуль, $y_{6\text{випр}}$ =0.

Записуємо виправлене повідомлення $Y_{\text{випр}}$ = {1 0 1 1 0 0 1 1 1}. ■

Порівнюючи результати з попереднім прикладом, відзначаємо, що дійсно було прийнято закодоване в попередньому прикладі повідомлення, в якому було спотворено шостий біт.

Код Гемінґа використовується в деяких технологіях зберігання даних, наприклад RAID 2 (*RAID,* англ. *Redundant Array of Independent Disks*) , надлишковий масив незалежних дисків - метод віртуалізації даних для об'єднання кількох фізичних дискових пристроїв у логічний модуль для підвищення стійкості до відмови і продуктивності). Крім того, метод Гемінґа застосовується у пам'яті типу ECC (англ. *Error-Correcting Code memory*, пам'ять з корекцією помилок - комп'ютерної пам'яті , яка автоматично розпізнає та виправляє бітові помилки.

Широке поширення коди Гемінґа отримали в системах супутникової радіонавігації для захисту від помилок інформації, що передається з борту супутника Так , в системі GPS NavStar використовується (32,26) код Гемінґа, в системі ГЛОНАСС – (85, 77) код.

Завдання для самостійного вирішення

1. Закодувати кодом із перевіркою на парність інформаційну послідовність A = {1 1 0 1 0 1 1 0 1 0 1}. Визначити відносну надмірність та швидкість коду.

2. Декодувати (вказати наявність помилки) закодовані кодом з перевіркою на парність кодові послідовності Y_1 = 1 0 1 0 1 1 0 1 1 0 0 1 1 1 0}, Y_2 = {0 0 1 1 0 1 0 0 1 1 1 0 1 0 1}.

3. Визначити ймовірність невиявленої помилки для коду з перевіркою на парність з n = 8. Ймовірність бітової помилки p = 0,2.

4. Закодувати кодом із простим повторенням інформаційну послідовність A = {1 0 1 1 0 1 1 0 1}.

5. Декодувати (вказати наявність помилки) закодовану кодом із простим повторенням кодову послідовність Y = {1 0 1 1 1 1 0 0 1 1}.

6. Визначити ймовірність невиявленої помилки для коду з простим повторенням з n = 8. Ймовірність бітової помилки p = 0,2.

7. Закодувати кореляційним кодом інформаційну послідовність

A = {1 0 1 1 0 1 1 0 1}.

8. Декодувати (вказати наявність помилки) закодовану кореляційним кодом кодову послідовність Y = {1 0 0 1 1 0 0 1 0 0 0 1 1 0}.

9. Закодувати інверсним кодом інформаційні послідовності

A_1 = {1 0 1 1 1 0 0 1}, A_2 = {1 1 1 0 1 1 0 1}.

10. Декодувати (вказати наявність помилки) закодовану інверсним кодом кодові послідовності

Y_1 = {1 0 0 1 1 1 0 1 1 0 1 0}, Y_2 = {1 0 1 0 1 1 1 0 1 0 1 0 0 0}.

11. Визначити ймовірність невиявленої помилки для коду з простим повторенням з n = 12. Ймовірність бітової помилки p = 0,2.

12. Закодувати десятирозрядним унітарним кодом число 8.

13. Визначити відносну надмірність та швидкість коду 3 з 9 (3/6) коду Code 39, що широко використовується для штрих кодування товарів.

14. Декодувати (вказати наявність помилки) закодовану 3/6 кодом із постійною вагою послідовності

Y_1 = {1 0 0 1 0 1 0 0 0}, Y_2 = {0 0 1 0 0 1 1 0 1}.

15. Визначити ймовірність помилки для 3/6 коду з постійною вагою. Ймовірність бітової помилки p = 0,2.

16. Закодувати кодом Бергера інформаційну послідовність

A = {1 1 0 0 1 1 0 0 0 1}.

Визначити відносну надмірність та швидкість даного коду.

17. Декодувати (вказати наявність помилки) закодовані (16, 12) кодом Бергера послідовності Y_1 = {0 1 1 1 1 0 1 1 1 0 1 1 0 1 1 0},

Y_1 = {0 1 1 1 1 0 1 0 1 0 1 1 0 1 1 0}.

18. Побудувати матрицю коду Ріда-Маллера для m = 5, D =3.

19. Закодувати (8, 4) кодом Ріда-Маллера інформаційну послідовність A = {0 1 1 1}.

20. Закодувати (8, 7) кодом Ріда-Маллера інформаційну послідовність A = {0 1 0 1 1 1 1}.

21. Закодувати (16, 11) кодом Ріда-Маллера інформаційну послідовність A = {1 0 1 0 1 0 1 0 1 1 0}.

22. Закодувати (16, 15) кодом Ріда-Маллера інформаційну послідовність A = {1 1 1 0 1 0 1 0 1 0 1 1 0 1 0}.

23. Закодувати (8, 4) кодом Ріда-Маллера інформаційну послідовність A = {1 1 0 1}.

24. Декодувати закодовані (8, 4) кодом Ріда-Маллера послідовності

Y_1 = {1 1 0 0 0 0 1 1}, Y_2 = {0 1 1 1 0 1 1 0},

Y_3 = {0 0 1 1 1 1 1 0}.

25. Декодувати закодовані (16, 11) кодом Ріда-Маллера послідовності

Y_1 = {0 1 1 0 1 1 1 1 0 1 0 1 0 0 1 1},

Y_2={1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1}, Y_3= {0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 1}.

26. Закодувати (7, 4) кодом Гемінґа інформаційну послідовність

A = {0 1 1 1}.

27. Закодувати (15, 11) кодом Гемінґа інформаційну послідовність

A = {1 0 1 0 0 0 1 1 1 0 0}.

28. Закодувати (31, 26) кодом Гемінґа інформаційну послідовність

A = {1 1 1 1 0 1 0 1 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 1 0 0}.

29. Декодувати закодовану (7, 4) кодом Гемінґа інформаційну послідовність Y = {1 1 1 1 0 0 1}.

30. Декодувати закодовану (15, 11) кодом Гемінґа інформаційну послідовність Y = {1 1 1 1 0 1 0 1 1 0 1 1 1 0 0}.

31. Закодувати розширеним (8, 4) кодом Гемінґа інформаційну послідовність A = {1 1 1 1}.

32. Закодувати розширеним (16, 11) кодом Гемінґа інформаційну послідовність A = {1 0 1 1 0 0 1 1 0 0 0}.

33. Закодувати розширеним (32, 26) кодом Гемінґа інформаційну послідовність

A={0 1 1 1 0 1 0 1 1 1 1 1 1 0 0 1 0 1 0 0 0 1 1 0 0 0}.

34. Декодувати закодовані розширеним (8, 4) кодом Гемінґа інформаційна послідовність Y_1 = {1 1 1 0 1 0 0 1},

Y_2 = {1 0 1 0 0 1 0 1}, Y_3 = {1 1 1 0 0 0 0 0}.

35. Декодувати закодовані розширеним (16, 11) кодом Гемінґа інформаційні послідовності

Y_1 ={1 1 0 0 0 1 1 1 1 0 0 1 0 0 1 1}, Y_2 ={1 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0},

Y_3 ={1 1 0 0 0 1 1 1 1 0 0 1 1 0 1 0}, Y_4 ={1 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0}.

36. Закодувати усіченим (10, 6) кодом Гемінґа інформаційну послідовність A = {0 1 0 1 1 1}.

37. Декодувати закодовану усіченим (10, 6) кодом Гемінґа інформаційну послідовність Y = {1 0 1 1 1 0 1 1 1 0}.

ГЛАВА 6

ЦИКЛІЧНІ КОДИ. КОДИ БЧХ ТА РІДА-СОЛОМОНА

6.1 Визначення циклічних кодів

Різновидом систематичних кодів, що виявляють та виправляють помилки та пакети помилок різної кратності, є *циклічні коди* (англ. *CRC, Cyclic Redundance Code*). Назва «*циклічні*» пов'язана з тим, що у цих кодів будь-яка кодова комбінація може бути отримана шляхом циклічної перестановки (циклічного зсуву, циклічного зрушення) символів комбінації, що належить до цього ж коду. При циклічному зрушенні коду вправо крайній правий символ стає на крайню ліву позицію, при циклічному зрушенні вліво – навпаки, крайній лівий стає на крайню праву позицію.

Приклад 6.1. Циклічно зрушити кодову комбінацію A={1010001} на два розряди праворуч, а кодову комбінацію B={110101} на три розряди вліво.

Рішення. Записуємо кодову комбінацію A={1010001} як A={a_6, a_5, a_4, a_3, a_2, a_1, a_0}. При зрушенні вправо відбувається переміщення a_5=a_6, a_4=a_5, … , a_i= a_{i+1}, … , a_0=a_1, крайній правий символ стає на початок a_6=a_0.

Тоді результатом зсуву комбінації на один розряд праворуч буде комбінація $\vec{A}^{(1)} = \{1101000\}$. Зсуваємо отриману комбінацію ще раз циклічно вправо: $\vec{A}^{(2)} = \{0110100\}$.

Аналогічно робимо зрушення вліво комбінації B={110101}. b_5=b_4, b_4=b_3, … , b_i= b_{i-1}, … , b_1= b_0, крайній лівий символ стає наприкінці b_0= b_5.

Результатом зсуву комбінації B на один розряд ліворуч буде комбінація $\vec{B}^{(1)} = \{101011\}$. Зсуваємо отриману комбінацію циклічно вліво ще двічі: $\vec{B}^{(2)} = \{010111\}; \vec{B}^{(3)} = \{101110\}$. ■

Таким чином, якщо кодова комбінація {a_{n-1}, a_{n-2}, …, a_3, a_2, a_1, a_0} є дозволеною комбінацією циклічного коду, то комбінація {a_0, a_{n-1}, a_{n-2}, …, a_3, a_2, a_1} також належить до цього коду. Так, наприклад, циклічні зрушення кодової комбінації 10011 – коди 11001, 11100, 01110, 00111 також будуть дозволеними кодовими комбінаціями.

Як і інші блокові коди, циклічний код визначається такими параметрами як довжина кодової послідовності n, кількість інформаційних символів k, кількість перевірочних символів $r = n - k$.

Циклічні коди зручніше розглядати, представляючи комбінацію двійкового коду не як послідовність нулів і одиниць, а вигляді полінома (многочлена, багаточлена) певного ступеня

$$A(x) = a_n x^{n-1} + a_{n-1} x^{n-2} + \ldots + a_1 x^1 + a_0 x^0 \qquad (6.1)$$

де x – формальна змінна, що служить для вказівки своїм ступенем позиції кодового символу як коефіцієнта полінома; a_i − коефіцієнти полінома, які є елементами кодового слова, для двійкових кодів коефіцієнти можуть приймати значення 0 і 1.

Приклад 6.2. Представити двійкову послідовність A={011100101} у вигляді полінома від змінної x:

Рішення. Запишемо вихідну кодову послідовність у вигляді A={a_8, a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0} і надамо значення a_i коефіцієнтам полінома $A(x)$.

$A(x) = a_8x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 = 0\cdot x^8 + 1\cdot x^7 + 1\cdot x^6 + 1\cdot x^5 + 0\cdot x^4 + 0\cdot x^3 + 1\cdot x^2 + 0\cdot x^1 + 1\cdot x^0 = x^7 + x^6 + x^5 + x^2 + 1.$

Подання кодових комбінацій у формі (6.1) дозволяє звести дії над кодовими комбінаціями до операцій над багаточленами. При цьому складання двійкових багаточленів зводиться до підсумовування по модулю два коефіцієнти при рівних ступенях змінної x. ■

Приклад 6.3. Скласти двійкові послідовності A={1011001010} та B={1011001} у двійковій та поліноміальній формі.

Рішення. Складемо послідовності у двійковій формі. Послідовність із меншою кількістю символів доповнимо спереду трьома нулями для вирівнювання числа символів.

$$\begin{array}{r} \oplus\ 1011001010 \\ \underline{\quad 0001011001} \\ \quad 1010010011 \end{array}$$

Переведемо двійкові коди на поліноміальну форму.

$A(x) = a_9x^9 + a_8x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 =$
$= 1\cdot x^9 + 0\cdot x^8 + 1\cdot x^7 + 1\cdot x^6 + 0\cdot x^5 + 0\cdot x^4 + 1\cdot x^3 + 0\cdot x^2 + 1\cdot x^1 + 0\cdot x^0 =$
$= x^9 + x^7 + x^6 + x^3 + x.$

$B(x) = a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 =$

$=1\cdot x^6 + 0\cdot x^5 + 1\cdot x^4 + 1\cdot x^3 + 0\cdot x^2 + 0\cdot x^1 + 1\cdot x^0 = x^6 + x^4 + x^3 + 1.$

Складаємо поліноми, наводячи коефіцієнти за модулем два.

$C(x) = A(x) + B(x) = x^9 + x^7 + 2x^6 + x^4 + 2x^3 + x + 1=$

$= x^9 + x^7 + x^4 + x^1 + x^0 = x^9 + x^7 + x^4 + x + 1.$

Результати, отримані двома способами, збігаються. ■

Множення двійкових поліномів проводиться за звичайним правилом перемноження ступінних функцій, проте отримані в цьому випадку коефіцієнти при кожному ступені складаються за модулем два.

Приклад 6.4. Перемножити двійкові послідовності A={1011001010} и B={1011001} у двійковій та поліноміальній формі.

Рішення. Перемножимо поліноми у двійковій формі.

$$
\begin{array}{r}
\times\ 1\,0\,1\,1\,0\,0\,1\,0\,1\,0 \\
\underline{\qquad\quad 1\,0\,1\,1\,0\,0\,1} \\
1\,0\,1\,1\,0\,0\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\
1\,0\,1\,1\,0\,0\,1\,0\,1\,0 \\
1\,0\,1\,1\,0\,0\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\
\underline{1\,0\,1\,1\,0\,0\,1\,0\,1\,0} \\
1\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1\,1\,1\,0\,1\,0\,.
\end{array}
$$

Перемножимо послідовності, представлені в поліноміальній формі, після чого наведемо коефіцієнти по два модулі.

$C(x) = A(x) \times B(x) = (x^9 + x^7 + x^6 + x^3 + x) \times (x^6 + x^4 + x^3 + 1) = (x^{15} + x^{13} + x^{12} + x^9 + x^7) + (x^{13} + x^{11} + x^{10} + x^7 + x^5) + (x^{12} + x^{10} + x^9 + x^6 + x^4) + (x^9 + x^7 + + x^6 + x^3 + x) = x^{15} + 2x^{13} + 2x^{12} + x^{11} + 2x^{10} + 3x^9 + 3x^7 + 2x^6 + x^5 + x^4 + x^3 + x = x^{15} + x^{11} + x^9 + x^7 + x^5 + x^4 + x^3 + x.$

Результати, отримані двома способами, збігаються. ■

Ділення здійснюється за правилами ділення ступінних функцій, при цьому як операції складання, так і операції віднімання замінюються операціями підсумовування за модулем два.

Приклад 6.5. Поділити двійкові послідовності A={10111011001010} та B={10011001} у двійковій та поліноміальній формі.

Рішення. Ділимо послідовності в стовпчик у двійковій формі. Оскільки за модулем два віднімання еквівалентне додаванню, процедура ділення зводиться до ланцюжка зрушень і додавань. Жирним виділені одержувані біти частки, курсивом остача.

$$
\begin{array}{l}
\mathbf{1}\,0\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0 \;\big|\; \underline{1\,0\,0\,1\,1\,0\,0\,1} \\
1\,0\,0\,1\,1\,0\,0\,1 \qquad\qquad\quad\; 1\,0\,1\,0\,0\,1\,0 \\
\;\;\mathbf{0\,1}\,0\,0\,0\,1\,0\,0\,0 \\
\;\;\;\;1\,0\,0\,1\,1\,0\,0\,1 \\
\;\;\;\;0\,\mathbf{0\,0\,1}\,0\,0\,0\,1\,1\,0\,1\,0 \\
\qquad\quad 1\,0\,0\,1\,1\,0\,0\,1 \\
\qquad\quad 0\,\mathbf{0}\,0\,1\,0\,1\,0\,0\,0 \\
\qquad\qquad\;\; \mathbf{0\,1\,0\,1\,0\,0\,0}
\end{array}
$$

Перекладаємо двійкові коди у поліноміальну форму:

A={10111011001010} $\rightarrow A(x) = x^{13} + x^{11} + x^{10} + x^9 + x^7 + x^6 + x^3 + x$.

B= {10011001} $\rightarrow B(x) = x^7 + x^4 + x^3 + 1$.

При діленні поліномів куточком у частку заносяться ступені x, рівні різниці старшого ступеня остачі та старшого ступеня дільника

$$
\begin{array}{l}
x^{13} + x^{11} + x^{10} + x^9 + x^7 + x^6 + x^3 + x \;\big|\; \underline{x^7 + x^4 + x^3 + 1} \\
x^{13} + x^{10} + x^9 + x^6 \qquad\qquad\qquad\qquad\quad x^6 + x^4 + x \\
\qquad x^{11} + x^7 + x^3 + x \\
\qquad x^{11} + x^8 + x^7 + x^4 \\
\qquad\quad\; x^8 + x^4 + x^3 + x \\
\qquad\quad\; x^8 + x^5 + x^4 + x \\
\qquad\qquad\;\; x^5 + x^3 \\
\qquad\qquad\;\; \boldsymbol{x^5 + x^3}\,.
\end{array}
$$

Частка від ділення поліномів $Q(x) = x^6 + x^4 + x$,

Остача (виділена жирним) $R(x) = x^5 + x^3$.

Результати двома способами збігаються. ■

Циклічний зсув ліворуч у поліноміальній формі еквівалентний множенню полінома на x. Справді, якщо вихідна кодова комбінація виражається поліномом $A(x) = a_n x^{n-1} + a_{n-1}x^{n-2} + \ldots + a_1 x^1 + a_0 x^0$, то зсув ліворуч на один розряд відповідає множенню полінома на x.

$$\overleftarrow{A}(x) = x \cdot A(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x^2 + a_0 x,$$

Оскільки має місце циклічне зрушення, у старшому члені необхідно замінити x^n на 1

$$\overleftarrow{A}_ц(x) = a_{n-1}x^{n-1} + \ldots + a_1 x^2 + a_0 x + a_n. \qquad (6.2)$$

З математичної точки зору ця операція еквівалентна приведенню полінома за модулем $x^n + 1$

$$\overleftarrow{A}_ц(x) = x \cdot A(x) \bmod (x^n + 1). \qquad (6.3)$$

Приклад 6.6. Циклічно зрушити кодову комбінацію A={101100101} ліворуч у поліноміальній формі.

Рішення. Перекладаємо двійковий код у поліноміальну форму.

$A(x) = a_8x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 =$

$= x^8 + x^6 + x^5 + x^2 + 1.$

Зрушимо послідовність на один розряд вліво шляхом домноження на x. $\overleftarrow{A}(x) = x \cdot A(x) = x^9 + x^7 + x^6 + x^3 + x.$

Наведемо поліном за модулем шляхом обчислення остачі від ділення на $x^9 + 1$.

$$\overleftarrow{A}_ц(x) = x \cdot A(x) \bmod (x^9 + 1).$$

$$\begin{array}{l} x^9 + x^7 + x^6 + x^3 + x \,\big|\, \underline{x^9 + 1} \\ x^9 + 1 \qquad\qquad\quad 1 \\ \quad x^7 + x^6 + x^3 + x + 1 \\ \quad \boldsymbol{x^7 + x^6 + x^3 + x + 1.} \end{array}$$

Частка від ділення поліномів $Q(x) = 1$,

Остача (виділена жирним) $R(x) = x^7 + x^6 + x^3 + x + 1.$

Циклічний зсув послідовності на один розряд ліворуч у двійковій формі $\overleftarrow{A}_ц(x)$ = {011001011}.

Результати, отримані двома способами, збігаються. ■

Ідея виправлення помилок циклічними кодами полягає в тому, що дозволені комбінації циклічного повинні без остачі ділитися на деякий *породжуючий (утворюючий) поліном* $g(x)$, що, вибирається з числа незвідних поліномів, тобто, прийнята послідовність повинна задовольняти умові

$$y(x) = a(x) \cdot g(x). \qquad (6.4)$$

Декодування циклічного коду зводиться до ділення прийнятої кодової послідовності на породжуючий поліном. Якщо при діленні утворюється нульова остача, робиться висновок про відсутність помилок. Якщо остача ненульова, то її вигляд визначає місце знаходження помилок.

У літературі з теорії кодування, наприклад формулюються вимоги, що пред'являються до породжуючого полінома циклічних кодів.

По-перше, *породжуючий поліном* має бути *незвідним*. *Незвідним називається многочлен*, який не може бути представлений як добуток

многочленів нижчих ступенів, тобто, такий многочлен ділиться без залишку лише на себе чи одиницю, і ділиться ні який інший многочлен.

Приклад 6.7. Знайти всі незвідні многочлени першого, другого та третього ступеня.

Рішення. Існує всього два многочлени першого ступеня, x та $x+1$. Оскільки єдиний поліном меншого, ніж одиниця, ступеня – одиниця, обидва поліноми першого ступеня, x та $x+1$ незвідні.

Існує чотири поліноми другого ступеня: x^2, x^2+1, x^2+x, x^2+x+1. Перевіримо, які з них діляться без остачі на поліноми нижчого ступеня. Поліномами нижчого, ніж другий ступінь є x та $x + 1$. Очевидно, що поліноми x^2 та x^2+x діляться на поліном x і, таким чином, не є незвідними, а поліноми x^2+1 и x^2+x+1 дають при діленні на x одиницю у остачі.

Спробуємо розділити поліноми x^2+1 та x^2+x+1 на $x + 1$.

$$\begin{array}{l} x^2 + 1 \,|\, \underline{x + 1} \\ x^2 + x \,|\, x + 1 \\ \quad x + 1 \\ \quad x + 1 \\ \quad\quad \mathbf{0} \end{array}$$

Частка від ділення $Q(x)=x+1$, остача $R(x)=0$, отже, поліном x^2+1 не є незвідним.

Поділимо тепер на $x+1$ поліном x^2+x+1.

$$\begin{array}{l} x^2 + x + 1 \,|\, \underline{x + 1} \\ x^2 + x \quad\; |\, x \\ \quad\quad \mathbf{1} \end{array}$$

Частка від ділення $Q(x)=x$, остача $R(x)=1$. Таким чином, поліном x^2+x+1 не ділиться на жодний з поліномів меншого ступеня і, отже, є незвідним.

Двійкові поліноми третього ступеня зведені у перші два стовпці табл. 6.1.

Таблиця 6.1 – Двійкові поліноми третього ступеня

Двійкове подання	Поліноміальне подання	Розкладання на множники
1000	x^3	$x \cdot x \cdot x$
1001	$x^3 + 1$	$(x+1) \cdot (x^2 + x + 1)$
1010	$x^3 + x$	$x \cdot (x+1) \cdot (x+1)$
1011	$x^3 + x + 1$	незвідний
1100	$x^3 + x^2$	$x \cdot x \cdot (x+1)$
1101	$x^3 + x^2 + 1$	незвідний
1110	$x^3 + x^2 + x$	$x \cdot (x^2 + x + 1)$
1111	$x^3 + x^2 + x + 1$	$(x+1) \cdot (x+1) \cdot (x+1)$

Якщо кубічний багаточлен можна розкласти на множники, то одним із співмножників має бути поліном першого ступеня x або $x+1$.

Поліноми x^3, x^3+x, x^3+x^2 и x^3+x^2+x діляться без остачі на поліном x. Знайдемо остачі від ділення поліномів третього ступеня на поліном $x+1$.

$$
\begin{array}{l}
x^3 + 1 \;|\underline{x + 1} \\
x^3 + x^2 \;|\, x^2 + x + 1 \\
\quad x^2 + 1 \\
\quad x^2 + x \\
\qquad x + 1 \\
\qquad x + 1 \\
\qquad\quad \mathbf{0}
\end{array}
$$

$$
\begin{array}{l}
x^3 + x + 1 \;|\underline{x + 1} \\
x^3 + x^2 \;\;\;|\, x^2 + x \\
\quad x^2 + x + 1 \\
\quad x^2 + x \\
\qquad\quad 1 \\
\qquad\quad \mathbf{1}
\end{array}
$$

$$
\begin{array}{l}
x^3 + x^2 + 1 \;|\underline{x + 1} \\
x^3 + x^2 \;\;\;|\, x^2 \\
\qquad 1 \\
\qquad \mathbf{1}
\end{array}
$$

$$
\begin{array}{l}
x^3 + x^2 + x + 1 \;|\underline{x + 1} \\
x^3 + x^2 \;\;\;\;\;|\, x^2 + 1 \\
\qquad x + 1 \\
\qquad x + 1 \\
\qquad\quad \mathbf{0}
\end{array}
$$

Поліноми x^3+1 та x^3+x^2+x+1 діляться без остачі на поліном $x+1$ і, отже, незвідними не є, в той час, як поліноми x^3+x+1 та x^3+x^2+1 не діляться на жоден з поліномів меншого ступеню і є незвідними. ■

У додатку Б в табл. Б.1 представлені незвідні поліноми різних ступенів.

Незвідні поліноми є дільниками бінома виду x^n+1, і, таким чином, будь-який біном x^n+1, може бути представлений у вигляді добутку поліномів

$$(x^n+1) = \varphi_1(x)\cdot\varphi_2(x)\cdot\ldots\cdot\varphi_l(x)\,. \qquad (6.5)$$

Приклад 6.8. Знайти усі незвідні многочлени – дільники двійкового двочлена x^7+1.

Рішення. Розглянемо як можливі дільники поліноми першого ступеня. Це поліноми x та $x+1$. На поліном x поліном x^7+1 завідомо не ділиться. Спробуємо розділити його на поліном $x+1$.

$$\begin{array}{l}
x^7 + 1 \;\;|\underline{x+1} \\
x^7 + x^6 \;\;|\,x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \\
\quad x^6 + x^5 \\
\quad\quad x^5 + x^4 \\
\quad\quad\quad x^4 + x^3 \\
\quad\quad\quad\quad x^3 + x^2 \\
\quad\quad\quad\quad\quad x^2 + x \\
\quad\quad\quad\quad\quad\quad x + 1 \\
\quad\quad\quad\quad\quad\quad x + 1 \\
\quad\quad\quad\quad\quad\quad\quad \mathbf{0}
\end{array}$$

Поліном розділився без остачі. Частка від ділення

$Q(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$

Спробуємо розділити отриману частку на $x+1$.

$$\begin{array}{l}
x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \;\;|\underline{x+1} \\
x^6 + x^5 \qquad\qquad\qquad\qquad\qquad\quad |\,x^5 + x^3 + x \\
\quad x^4 + x^3 + x^2 + x + 1 \\
\quad x^4 + x^3 \\
\quad\quad\quad x^2 + x + 1 \\
\quad\quad\quad x^2 + x \\
\quad\quad\quad\quad\quad 1 \\
\quad\quad\quad\quad\quad \mathbf{1}
\end{array}$$

Частка від ділення $Q(x) = x^5 + x^3 + x$, остача $R(x) = 1$.

Таким чином, $x + 1$ не є дільником полінома $x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ і, отже, і полінома $x^7 + 1$. Єдиним незвідним поліномом другого ступеня є поліном $x^2 + 1$.

Перевіримо, чи ділиться на нього поліном $x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$.

$$\begin{array}{l} x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \;\big|\; \underline{x^2 + 1} \\ x^6 + x^4 \qquad\qquad\qquad\qquad\quad \big|\; x^4 + x^3 + 1 \\ \quad x^5 + x^3 + x^2 + x + 1 \\ \quad x^5 + x^3 \\ \qquad\quad x^2 + x + 1 \\ \qquad\quad x^2 + 1 \\ \qquad\qquad x \\ \qquad\qquad \boldsymbol{x} \end{array}$$

При діленні утворилася ненульова остача x, тому x^2+1 не є дільником полінома $x^7 + 1$. Розглянемо як можливі дільники поліноми третього ступеня. Це поліноми $x^3 + x + 1$ та $x^3 + x^2 + 1$.

Спробуємо розділити $x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ на $x^3 + x + 1$.

$$\begin{array}{l} x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \;\big|\; \underline{x^3 + x + 1} \\ x^6 + x^4 + x^3 \qquad\qquad\quad \big|\; x^3 + x^2 + 1 \\ \quad x^5 + x^2 + x + 1 \\ \quad x^5 + x^3 + x^2 \\ \qquad\quad x^3 + x + 1 \\ \qquad\quad x^3 + x + 1 \\ \qquad\qquad \mathbf{0} \end{array}$$

Частка від ділення – поліном $Q(x) = x^3 + x^2 + 1$, який, своєю чергою, є незвідним поліномом, остача $R(x)$ дорівнює нулю.

Таким чином, біном x^7-1 має наступне розкладання .

$$x^7 + 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1)\,. \blacksquare$$

Розглянемо деякі закономірності розкладання бінома x^n+1 в добуток незвідних поліномів.

Для парних n поліном $x^n + 1$ розкладається у добуток

$x^n + 1 = (x^m + 1)\cdot(x^m + 1) = x^{2m} + x^m + x^m + 1 = x^{2m} + 1$, где $m = n/2$, і, таким чином, може бути зведений до добутку поліномів непарних ступенів.

Для $n = 3, 5, 11, 13, 19, 29$ розкладання має вигляд:

$x^n + 1 = (x + 1)\cdot(x^{n-1} + x^{n-2} + \ldots + x + 1)$, наприклад,

$x^5 + 1 = (x + 1)\cdot(x^4 + x^3 + x^2 + x + 1)$,

що визначає лише два типи кодів цих довжин.

1. Код з породжуючим поліномом $g(x)=x+1$ – це код з перевіркою на парність з кодовою відстанню $d_{\min}=2$, що виявляє помилки непарної кратності.

2. Код з породжуючим поліномом $g(x) = x^{n-1} + x^{n-2} + \ldots + x + 1$ – це код із n-кратним повторенням.

Розкладання біномів x^n+1 для деяких інших значень наведені в табл. 6.2.

Таблиця 6.2. – Розкладання полінома x^n+1 на незвідні поліноми

n	Дільники
7	$(x+1)\cdot(\underline{\boldsymbol{x^3+x+1}})\cdot(\underline{\boldsymbol{x^3+x^2+1}})$
9	$(x+1)\cdot(\underline{\boldsymbol{x^2+x+1}})\cdot(x^6+x^3+1)$
15	$(x+1)\cdot(\underline{\boldsymbol{x^2+x+1}})\cdot(\underline{\boldsymbol{x^4+x+1}})\cdot(\underline{\boldsymbol{x^4+x^3+1}})\cdot(x^4+x^3+x^2+x+1)$
17	$(x+1\cdot(x^8+x^5+x^4+x^3+x+1)\cdot(x^8+x^7+x^6+x^4+x^2+x+1)$
21	$(x+1)\cdot(\underline{\boldsymbol{x^2+x+1}})\cdot(\underline{\boldsymbol{x^3+x+1}})\cdot(\underline{\boldsymbol{x^3+x^2+1}})\cdot(x^6+x^4+x^2+x+1)\cdot$ $\cdot(x^6+x^5+x^4+x^2+1)$
23	$(x+1)\cdot(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1)\cdot(x^{11}+x^9+x^7+x^6+x^5+x+1)$
25	$(x+1)\cdot(x^4+x^3+x^2+x+1)\cdot(x^{20}+x^{15}+x^{10}+x^5+1)$
27	$(x+1)\cdot(\underline{\boldsymbol{x^2+x+1}})\cdot(x^6+x^3+1)\cdot(x^{18}+x^9+1)$
31	$(x+1)\cdot(\underline{\boldsymbol{x^5+x^3+1}})\cdot(\underline{\boldsymbol{x^5+x^2+1}})\cdot(\underline{\boldsymbol{x^5+x^4+x^3+x^2+1}})\cdot$ $\cdot(\underline{\boldsymbol{x^5+x^4+x^3+x+1}})\cdot(\underline{\boldsymbol{x^5+x^4+x^2+x+1}})\cdot(\underline{\boldsymbol{x^5+x^3+x^2+x+1}})$

Слід зазначити, що не всякий незвідний поліном придатний для використання в якості утворюючого полінома циклічного коду. Оскільки процедура декодування циклічного коду зводиться до аналізу залишку від поділу прийнятої комбінації на породжуючий поліном, то по вигляду залишку можна зробити висновок про характер помилки. Наприклад, для виявлення одиночної помилки та визначення її розташування слід визначити, який із залишків від поділу вектора помилки {0 0 0 …1… 0 0 }, тобто полінома x^l відповідає отриманому. Природно, що для правильного декодування всі залишки мають бути різними.

Перевіримо, наприклад, які з незвідних поліномів придатні для формування (15, 11) кодів. Оскільки кількість перевірочних розрядів $r=n-k$, розглянемо в якості можливих поліноми четвертого ступеня.

Виберемо із табл. 6.2 три можливі поліноми x^4+x+1, x^4+x^3+1 та $x^4+x^3+x^2+x+1$ і проаналізуємо залишки від ділення x^l на ці поліноми. Результати зведемо у табл. 6.3.

Таблиця 6.3 – Остачі від ділення x^l на незвідні поліноми

l	x^l	Остачі від ділення на поліном		
		x^4+x+1 10011	x^4+x^3+1 11001	$x^4+x^3+x^2+x+1$ 11111
0	1	1	1	1
1	x	x	x	x
2	x^2	x^2	x^2	x^2
3	x^3	x^3	x^3	x^3
4	x^4	$x + 1$	$x^3 + 1$	$x^3 + x^2 + x + 1$
5	x^5	$x^2 + x$	$x^3 + x + 1$	1
6	x^6	$x^3 + x^2$	$x^3 + x^2 + x + 1$	x
7	x^7	$x^3 + x + 1$	$x^2 + x + 1$	x^2
8	x^8	$x^2 + 1$	$x^3 + x^2 + x$	x^3
9	x^9	$x^3 + x$	$x^2 + 1$	$x^3 + x^2 + x + 1$
10	x^{10}	$x^2 + x + 1$	$x^3 + x$	1
11	x^{11}	$x^3 + x^2 + x$	$x^3 + x^2 + 1$	x
12	x^{12}	$x^3 + x^2 + x + 1$	$x + 1$	x^2
13	x^{13}	$x^3 + x^2 + 1$	$x^2 + x$	x^3
14	x^{14}	$x^3 + 1$	$x^3 + x^2$	$x^3 + x^2 + x + 1$
15	x^{15}	1	1	1

З табл. 6.3 очевидно, що для поліномів $x^4 + x + 1$ та $x^4 + x^3 + 1$ період послідовностей, складених з остач, T=15, тобто різним розташуванням одиночної помилки відповідають різні остачі, і ці поліноми можуть бути обрані в якості утворюючих.

У той же час для полінома $x^4 + x^3 + x^2 + x + 1$ остачі повторюються з періодом $T = 5$, різним поодиноким помилкам відповідають однакові остачі, і цей поліном не годиться як породжуючий.

Незвідний поліном $p(x)$ ступеня n, що генерує послідовність залишків максимального періоду $T = 2^n - 1$ називається *примітивним* . Як

ми з'ясували, не всі незвідні поліноми є примітивними, але всі примітивні поліноми незвідні.

Примітивні поліноми виділено у табл. 6.3. У додатку Б містяться таблиці незвідних і примітивних поліномів деяких ступенів.

Аналогічний пошук поліномів можна зробити і для кодів з більшою кодовою відстанню. Так, (15,7) код з $d_{\min}=5$, що виправляє подвійні помилки, породжується поліномом $g(x) = (x^4 + x + 1)\cdot(x^4 + x^3 + x^2 + x + 1)$.

6.2 Методи кодування та декодування двійкових послідовностей циклічними кодами

Комбінації циклічного коду можна отримати, помножуючи многочлен $a(x)$ на породжуючий поліном коду $g(x)$. Такий спосіб легко реалізується, однак він має той недолік, що комбінації коду, що отримуються в результаті множення, не містять інформаційних символів у явному вигляді. Після виправлення помилок такі комбінації для виділення інформаційних символів доводиться ділити на породжуючий поліном. Ситуацію можна спростити, якщо контрольні символи переписати на кінець коду, тобто. після інформаційних розрядів. Таким чином, циклічний код можна утворити двома способами.

1. *Несистематичне кодування* – множення інформаційної комбінації двійкового циклічного коду $a(x)$ на породжуючий поліном $g(x)$

$$v(x) = a(x) \cdot g(x). \tag{6.6}$$

Недолік такого способу кодування, полягає в тому, що отримуваний код не має чіткого поділу розрядів на інформаційні та перевірочні, що ускладнює процес декодування.

2. *Систематичне кодування* – множення інформаційної комбінації $a(x)$ на одночлен x^r, що має той самий ступінь, що і породжуючий поліном $g(x)$, з додаванням до цього добутку остачі $R(x)$, отриманої після ділення добутку $a(x)\cdot x^r$ на породжуючий поліном $g(x)$.

Дійсно, помножимо інформаційну комбінацію $a(x)$ на одночлен x^r і розділимо результат на породжуючий поліном $g(x)$

$$\frac{a(x)\cdot x^r}{g(x)} = q(x) + \frac{R(x)}{g(x)}, \tag{6.7}$$

де $q(x)$ – частка, $R(x)$ – остача від ділення.

Помножуючи обидві частини (6.7) на $g(x)$., після перетворення отримаємо

$$v(x) = q(x) \cdot g(x) = a(x) \cdot x^r + R(x). \qquad (6.8)$$

Таким чином, кодове слово складається з незмінної інформаційної частини a, після якої розташовані $(n-k)$ перевірочних символів. Перевірочні символи є коефіцієнтами полінома $R(x)$, тобто остачі від ділення $a(x) \cdot x^{n-k}$ на породжуючий поліном $g(x)$.

Оскільки при декодуванні зазвичай зручно, коли інформаційні та контрольні символи у комбінаціях циклічного коду відокремлені один від одного, частіше застосовують другий спосіб побудови циклічного коду.

Приклад 6.9. Закодувати послідовність $a(x)=x^{10}+x^8+x^7+x^3+1$ несистематичним методом циклічним кодом з породжуючим поліномом $g(x)=x^4+x+1$.

Рішення. Представимо вихідну послідовність і породжуючий поліном у двійковій формі та визначимо кодову послідовність відповідно до (6.6).

a = {1 0 1 1 0 0 0 1 0 0 1}, g = {1 0 0 1 1}.

$v = a \times g$.

```
          1 0 1 1 0 0 0 1 0 0 1
                  × 1 0 0 1 1
          1 0 1 1 0 0 0 1 0 0 1
        1 0 1 1 0 0 0 1 0 0 1
      0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0
  1 0 1 1 0 0 0 1 0 0 1
1 0 1 0 1 1 0 0 0 0 0 1 0 1 1.
```

У двійковій формі v ={1 0 1 0 1 1 0 0 0 0 0 1 0 1 1}, у поліноміальній $v(x) = x^{14} + x^{12} + x^{10} + x^9 + x^3 + x + 1$. ■

Приклад 6.10. Закодувати несистематичним методом циклічним кодом з породжуючим поліномом $g(x)=x^3+x+1$ всі можливі чотирирозрядні послідовності.

Рішення. Подамо породжуючий поліном у двійковій формі: g={1011}. Загальна кількість чотирирозрядних послідовностей складе $N=2^4=16$. Запишемо послідовно всі можливі чотирирозрядні послідовності

у двійковій формі та обчислимо циклічні коди у несистематичній формі згідно (6.6)

a_0 = { 0 0 0 0 }, $v_0 = a_0 \times g$ = { 0 0 0 0 0 0 0 },

a_1 = { 0 0 0 1 }, $v_1 = a_1 \times g$ = { 0 0 0 1 0 1 1 },

a_2 = { 0 0 1 0 }, $v_1 = a_2 \times g$ = { 0 0 1 0 1 1 0 },

a_3 = { 0 0 1 1 },

$$
\begin{array}{r}
0\,0\,1\,1 \\
\times\ 1\,0\,1\,1 \\
\hline
0\,0\,1\,1 \\
0\,0\,1\,1 \\
0\,0\,0\,0 \\
0\,0\,1\,1 \\
\hline
0\,0\,1\,1\,1\,0\,1.
\end{array}
$$

$v_3 = a_3 \times g$ = { 0 0 1 1 1 0 1 }.

Аналогічні операції здійснюємо для решти чотирирозрядних двійкових кодів (рекомендується виконати самостійно). Результати зведемо у табл. 6.4.

Таблиця 6.4 – Циклічне кодування чотирирозрядних послідовностей

k	Вихідна послідовність $a_k(x)$	Породжуючий поліном $g(x) = x^3 + x + 1$	Циклічний (7,4) код $v_k(x) = a_k(x) \cdot g(x)$
0	0000	1011	0 0 0 0 0 0 0
1	0001		0 0 0 1 0 1 1
2	0010		0 0 1 0 1 1 0
3	0011		0 0 1 1 1 0 1
4	0100		0 1 0 1 1 0 0
5	0101		0 1 0 0 1 1 1
6	0110		0 1 1 1 0 1 0
7	0111		0 1 1 0 0 0 1
8	1000		1 0 1 1 0 0 0
9	1001		1 0 1 0 0 1 1
10	1010		1 0 0 1 1 1 0
11	1011		1 0 0 0 1 0 1
12	1100		1 1 1 0 1 0 0
13	1101		1 1 1 1 1 1 1
14	1110		1 1 0 0 0 1 0
15	1111		1 1 0 1 0 0 1

Аналіз таблиці 6.4 показує, що кодові послідовності дійсно є взаємними циклічними зрушеннями. Так ланцюжок циклічних зрушень праворуч утворюють послідовності

$a_1 \to a_{11} \to a_{14} \to a_7 \to a_8 \to a_4 \to a_2$,

а також $a_3 \to a_{10} \to a_5 \to a_9 \to a_{15} \to a_{12} \to a_6$.

Послідовності a_0 та a_{13} інваріантні до зсуву.

Крім того, очевидно, що мінімальна відстань Гемінґа для побудованого коду d_{min}=3, тобто код може виправляти поодинокі помилки. ■

Приклад 6.11. Закодувати послідовність $a(x) = x^2 + x$ систематичним методом циклічним кодом з породжуючим поліномом $g(x) = x^3 + x + 1$.

Рішення. Для формування систематичного циклічного коду множимо поліном $a(x)$ на x^r, тобто приписуємо до коду праворуч r нулів, де r – порядок (старший ступінь) полінома $g(x)$, що визначає число перевірочних розрядів у коді.

$g(x) = x^3 + x + 1$. Таким чином, r=3.

$h(x) = a(x) \times x^3 = (x^2 + x) \times x^3 = x^5 + x^4$.

Представимо отриману послідовність h та породжуючий поліном g у двійковій формі h = {0 1 1 0 0 0 0}, g = {1 0 1 1} та поділимо їх.

0 1 1 0 0 0 0 | 1 0 1 1
1 0 1 1 | 0 1 1 1
1 1 1 **0**
1 0 1 1
1 0 1 0
1 0 1 1
0 0 1

Частка від ділення q ={0 1 1 1}, остача R = {0 0 1}.

У поліноміальній формі $q(x) = x^2 + x + 1$, $R(x) = 1$.

Підсумовуємо поліном $h(x) = a(x) \times x^4$ із остачею $R(x)$.

$v(x) = a(x) \cdot x^r + R(x) = x^5 + x^4 + 1$.

У двійковій формі v = {0 1 1 0 0 0 1}. ■

Приклад 6.12. Закодувати послідовність $a(x) = x^{10} + x^7 + x^5 + x^4 + 1$ систематичним методом циклічним кодом з породжуючим поліномом $g(x)=x^4+x+1$.

Рішення. Для формування систематичного циклічного коду множимо поліном $a(x)$ на x^r. Число перевірочних розрядів дорівнює старшому ступеню породжуючого полінома $g(x) = x^4 + x + 1$, тобто r = 4.

$h(x) = a(x) \times x^3 = (x^{10} + x^7 + x^5 + x^4 + 1) \times x^4 = x^{14} + x^{11} + x^9 + x^8 + x^4.$

$h = \{1\,0\,0\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\}$, $g = \{1\,0\,0\,1\,1\}$.

Ділимо отриманий добуток h на породжуючий поліном

$$\begin{array}{l}
\mathbf{1}\,0\,0\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0 \,|\, \underline{1\,0\,0\,1\,1} \\
1\,0\,0\,1\,1 \qquad\qquad\qquad\qquad\quad |\, 1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,1 \\
\quad \mathbf{0\,0\,0\,1}\,1\,1\,0\,0 \\
\qquad\quad 1\,0\,0\,1\,1 \\
\qquad\quad \mathbf{1}\,1\,1\,1\,0 \\
\qquad\quad 1\,0\,0\,1\,1 \\
\qquad\quad\ \mathbf{1}\,1\,0\,1\,1 \\
\qquad\quad\ 1\,0\,0\,1\,1 \\
\qquad\quad\ \ \mathbf{1}\,0\,0\,0\,0 \\
\qquad\quad\ \ 1\,0\,0\,1\,1 \\
\qquad\quad\ \ \ \mathbf{0\,0\,1}\,1\,0\,0\,0 \\
\qquad\quad\ \ \ \ 1\,0\,0\,1\,1 \\
\qquad\quad\ \ \ \ \ \mathbf{1\,0\,1\,1}
\end{array}$$

Частка від ділення $q = [1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,1]$, остача $R = [1\,0\,1\,1]$.

У поліноміальній формі $q(x) = x^{10} + x^6 + x^5 + x^4 + x^3 + 1$,

$R(x) = x^3 + x + 1$.

Підсумовуємо поліном $h(x) = a(x) \times x^4$ із остачею $R(x)$.

$v(x) = a(x) \cdot x^r + R(x) = x^{14} + x^{11} + x^9 + x^8 + x^4 + x^3 + x + 1$.

У двійковій формі $v = \{1\,0\,0\,1\,0\,1\,1\,0\,0\,0\,1\,1\,0\,1\,1\}$. ■

Методи декодування повідомлень, закодованих несистематичними циклічними кодами, зводяться до зберігання в пам'яті таблиці остач від ділення прийнятої комбінації на породжуючий поліном, відповідних всім можливим розташуванням помилок і пошуку в таблиці даного залишку. Цей метод прийнятний при невеликій довжині коду та кратності помилок, інакше таке декодування вимагає значних апаратних та часових витрат.

Оскільки найчастіше застосовується систематичний метод кодування, розглянемо принцип виправлення помилок у повідомленнях, закодованих систематичними циклічними кодами.

Алгоритм виявлення та виправлення помилок включає наступні кроки:

1. Прийнята комбінація $y(x)$ ділиться на породжуючий поліном $g(x)$: $y(x) = q(x) \cdot g(x) \oplus R(x)$.

2. Підраховується вага w (кількість одиниць) у остачі $R(x)$. Якщо $w \le t_c$, де t_c – число помилок, що виправляються цим кодом, прийняту комбінацію складають за модулем два із остачею. Сума $y'(x) = y(x) \oplus R(x)$

дає виправлену комбінацію. Зокрема, якщо $R(x) = 0$, робиться висновок, що помилок не сталося.

3. Якщо $w > t_c$, то виконують циклічний зсув прийнятої комбінації ліворуч на один розряд. Комбінацію $\vec{y}^{(1)}$, отриману в результаті циклічного зсуву, ділять на породжуючий поліном $g(x)$.

$$\vec{y}^{(1)}(x) = \vec{q}^{(1)}(x) \cdot g(x) + \vec{R}^{(1)}(x)$$

4. Якщо в результаті повторного ділення вага отриманої остачі $w \le t_c$, то $\vec{y}^{(1)}(x)$ підсумовують із остачею, потім проводять циклічний зсув праворуч на один розряд комбінації, отриманої в результаті підсумовування останнього діленого $\vec{y}^{(1)}(x)$ з останньою остачею $\vec{R}^{(1)}(x)$. Отримана комбінація вже не містить помилок.

5. Якщо після першого циклічного зсуву та наступного ділення остача виходить такою, що її вага $w > t_и$, то переходять до п.3 і повторюють операцію до тих пір, поки не буде досягнуто $w \le t_c$.

6. Якщо $w \le t_c$, комбінацію, отриману в результаті останнього циклічного зсуву, підсумовують із остачею від поділу цієї комбінації на породжуючий багаточлен, а потім проводять циклічний зсув праворуч на стільки розрядів, на скільки була зсунута сумована з останньою остачею комбінація щодо прийнятої комбінації. В результаті отримаємо виправлену комбінацію

Приклад 6.13. Декодувати повідомлення, y={0010001}, закодоване циклічним кодом, що виправляє одиночну помилку t_c=1 з утворюючим поліномом g = {1 0 1 1}.

Рішення. Загальна кількість розрядів коду n=7, число перевірочних розрядів r – порядок (старший ступінь) полінома $g(x) = x^3 + x + 1$, r = 3. Число інформаційних розрядів $k = n - r = 7 - 3 = 4$.

Ділимо вихідний поліном у на породжуючий поліном g.

0 0 1 0 0 0 1 | 1 0 1 1
1 0 1 1 | 0 0 1 0
0 1 1 1
1 1 1

Остача від ділення R = {1 1 1}. Вага (число одиниць) w остачі дорівнює трьом. Оскільки $w > t_c$, робимо циклічний зсув повідомлення вліво на один розряд: $\vec{y}^{(1)}$ = {0 1 0 0 0 1 0}.

Ділимо зрушений поліном $\vec{y}^{(1)}$ на породжуючий поліном g .

0 1 0 0 0 1 0 | 1 0 1 1
1 0 1 1 | 0 1 0 1
0 1 1 1 0
1 0 1 1
1 0 1

Остача від ділення R = {1 0 1}. Вага остачі w =2. Оскільки $w > c$, робимо циклічний зсув повідомлення вліво на один розряд:

$\vec{y}^{(2)}$ = {1 0 0 0 1 0 0}.

Ділимо зрушений поліном $\vec{y}^{(2)}$ на породжуючий поліном g .

1 0 0 0 1 0 0 | 1 0 1 1
1 0 1 1 | 1 1 0 1
0 1 1 1 0
1 0 1 1
1 0 1 0
0 0 1

Остача від ділення R = {0 0 1}. Вага остачі w=1. Оскільки $w=t_c$, припиняємо зрушення та приступаємо до виправлення помилки. Останню зрушену кодову комбінацію поелементно підсумовуємо з останньою остачею.

$y_{зс} = \vec{y}^{(2)} \oplus R$ = 1 0 0 0 1 0 0 ⊕ 0 0 1= 1 0 0 0 1 0 1.

Оскількі в ході декодування вихідна кодова комбінація була двічі зсунута вліво, усуваємо цю дію шляхом триразового циклічного зсуву $y_{зс}$ праворуч

1 0 0 0 1 0 1 → 1 1 0 0 0 1 0 → 0 1 1 0 0 0 1.

Виправлена кодова комбинація $y_{испр}(x)$ = {0 1 1 0 0 0 1}.

Очевидно, що виявлено та виправлено помилку у другому розряді. Інформаційні розряди займають k=4 ліві розряди повідомлення, тобто вихідне повідомлення, закодоване кодером каналу, A = {0 1 1 0}. ■

Приклад 6.14. Декодувати повідомлення y={101101100011011}, закодоване систематичним циклічним кодом, що виправляє одиночну помилку t_c=1 з утворюючим поліномом g = {1 0 0 1 1}.

Рішення. Загальна кількість розрядів коду n=15, число перевірочних розрядів r – порядок (старший ступінь) полінома $g(x) = x^4 + x + 1$, r = 4.

Число інформаційних розрядів $k = n - r = 15 - 4 = 11$.

Ділимо вихідний поліном y на породжуючий поліном g

```
1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 | 1 0 0 1 1
1 0 0 1 1                     | 1 0 1 0 1 0 0 1
  0 1 0 1 1 1
    1 0 0 1 1
      0 1 0 0 0 0
        1 0 0 1 1
          0 0 1 1 0 1 1
              1 0 0 1 1
                1 0 0 0 0
                1 0 0 1 1
                  0 0 1 1 1 1
                      1 1 1 1
```

Остача від ділення $R = \{1\ 1\ 1\ 1\}$. Вага w остачі $w = 4$.

Оскільки $w > t_и$, робимо циклічний зсув повідомлення вліво на один розряд: $\overleftarrow{y}^{(1)} = \{0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\}$.

Ділимо зрушений поліном $\overleftarrow{y}^{(1)}$ на породжуючий поліном g

```
0 1 1 0 1 1 0 0 0 1 1 0 1 1 1 | 1 0 0 1 1
  1 0 0 1 1                   | 1 0 1 0 1 0 0 1
    1 0 0 0 0
    1 0 0 1 1
      0 0 1 1 0 0 1
          1 0 0 1 1
            1 0 1 0 1
            1 0 0 1 1
              0 1 1 0 0 1
                1 0 0 1 1
                  1 0 1 0 1
                  1 0 0 1 1
                    0 1 1 0 1
                      1 1 0 1
```

Остача від ділення $R = \{1\ 1\ 0\ 1\}$. Вага w остачі $w = 3$.

Оскільки $w > t_и$, робимо циклічний зсув повідомлення вліво на один розряд: $\overleftarrow{y}^{(2)} = \{1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\}$.

Ділимо зрушений поліном $\overleftarrow{y}^{(2)}$ на породжуючий поліном g

1 1 0 1 1 0 0 0 1 1 0 1 1 1 0 | 1 0 0 1 1
1 0 0 1 1 | 1 1 0 0 1 1 0 1 1 0 1
1 0 0 0 0
1 0 0 1 1
0 0 1 1 0 0 1
1 0 0 1 1
1 0 1 0 1
1 0 0 1 1
0 1 1 0 0 1
1 0 0 1 1
1 0 1 0 1
1 0 0 1 1
0 1 1 0 1 0
1 0 0 1 1
1 0 0 1

Остача від ділення $R = \{1\ 0\ 0\ 1\}$. Вага w остачі $w = 2$.

Оскільки $w > t_и$, робимо циклічний зсув повідомлення вліво на один розряд: $\vec{y}^{(3)} = \{1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\}$.

Ділимо зрушений поліном $\vec{y}^{(2)}$ на породжуючий поліном g

1 0 1 1 0 0 0 1 1 0 1 1 1 0 1 | 1 0 0 1 1
1 0 0 1 1 | 1 0 1 0 1 1 1 0 1 0 0
0 1 0 1 0 0
1 0 0 1 1
0 1 1 1 1 1
1 0 0 1 1
1 1 0 0 0
1 0 0 1 1
1 0 1 1 1
1 0 0 1 1
0 1 0 0 1 1
1 0 0 1 1
0 0 0 0 0 1
0 0 0 1

Остача від ділення $R = \{0\ 0\ 0\ 1\}$. Вага остачі $w = 1$. Оскільки $w = t_и$, припиняємо зрушення та приступаємо до виправлення помилки. Останню зрушену кодову комбінацію поелементно підсумовуємо з останньою остачею.

$y_{зс} = \vec{y}^{(3)} \oplus R = 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1 \oplus 0\ 0\ 0\ 1 =$

$= 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0.$

Оскільки в ході декодування вихідна кодова комбінація була тричі зсунута вліво, усуваємо цю дію шляхом чотириразового циклічного зсуву $y_{зс}$ праворуч

1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 → 0 1 0 1 1 0 0 0 1 1 0 1 1 1 0 →

0 0 1 0 1 1 0 0 0 1 1 0 1 1 1 → 1 0 0 1 0 1 1 0 0 0 1 1 0 1 1.

Виправлена кодова комбінація y_c = { 1 0 0 1 0 1 1 0 0 0 1 1 0 1 1 }.

Очевидно, що виявлено та виправлено помилку у третьому розряді. Інформаційні розряди займають k = 11 лівих розрядів повідомлення, тобто вихідне повідомлення, закодоване кодером каналу,

A = { 1 0 0 1 0 1 1 0 0 0 1 }. ■

6.3 Розширені кінцеві поля Галуа. Мінімальні многочлени

Поля з кінцевим числом елементів q називають *полями Галуа* на ім'я першого дослідника Евариста Галуа і позначають $GF(q)$. Якщо задати незвідний многочлен $p(x)$ ступеня m, коефіцієнти якого належать полю $GF(q)$, то можна побудувати розширене поле Галуа, що містить q^m елементів, включаючи нульовий елемент. Елементами розширеного поля $GF(q^m)$ можуть бути всі многочлени ступеня $m - 1$ або менше, коефіцієнти яких лежать у простому полі $GF(q)$. Число q^m називається *порядком розширеного поля* та визначає кількість різних многочленів. При цьому для побудови розширеного поля використовуємий поліном $p(x)$ повинен бути незвідним у полі $GF(q)$, тобто його не можна розкласти на множники, використовуючи тільки многочлени з коефіцієнтами $GF(q)$. Також поліном $p(x)$ немає коренів у полі $GF(q)$.

Таким чином, поле, утворене за допомогою поліномів над полем $GF(q)$ по модулю незвідного багаточлена $p(x)$ ступеня m називається *розширенням поля* ступеня m над $GF(q^m)$ або *розширеним кінцевим полем*.

Далі ми розглядатимемо поля порядку $GF(2)$, елементами якого є значення {0, 1}. Тоді принцип побудови поля $GF(2^m)$ як розширення поля $GF(2)$ ґрунтується на використанні незвідного многочлена $p(x)$, коефіцієнти {0, 1} якого належать полю $GF(2)$, як модуль поля $GF(2^m)$. Цей многочлен $p(x)$ має ту властивість, що його не можна розкласти на множники, використовуючи тільки многочлени з коефіцієнтами з поля $GF(2)$. Таблиця незвідних многочленів представлена таблиці Б.1 додатка Б.

Приклад 6.15. Побудувати розширене кінцеве поле $GF(2^3)$ на базі незвідного многочлена $p(x) = x^3 + x + 1$.

Рішення. Побудуємо розширене кінцеве поле $GF(2^3)=GF(8)$ для простого поля $GF(2)=\{0, 1\}$. Елементами β поля $GF(2^3)$ будуть многочлени, побудовані за допомогою незвідного многочлена $p(x) = x^3 + x + 1$. Позначимо примітивний елемент поля $\alpha=x$ тоді маємо $\alpha^3 = \alpha + 1 \rightarrow x^3 = x + 1$. Обчислимо ступеня елемента

$\boldsymbol{\alpha^0} \rightarrow \underline{1}$;

$\boldsymbol{\alpha^1} \rightarrow \underline{x^1}$;

$\boldsymbol{\alpha^2} \rightarrow \underline{x^2}$;

$\boldsymbol{\alpha^3} \rightarrow x^3 = \underline{x + 1}$;

$\boldsymbol{\alpha^4} \rightarrow x^4 = x^3 \cdot x = (x + 1) \cdot x = \underline{x^2 + x}$;

$\boldsymbol{\alpha^5} \rightarrow x^5 = x^4 \cdot x = (x^2 + x) \cdot x = x^3 + x^2 = x + 1 + x^2 = \underline{x2 + x + 1}$;

$\boldsymbol{\alpha^6} \rightarrow x^6 = x^5 \cdot x = (x^2 + x + 1) \cdot x = x^3 + x^2 + x = x + 1 + x^2 + x = \underline{x^2 + 1}$;

$\boldsymbol{\alpha^7} \rightarrow x^7 = x^6 \cdot x = (x^2 + 1) \cdot x = x^3 + x = x + 1 + x = \underline{1}$.

Представлення елементів розширеного кінцевого поля $GF(2^3)$, заданого незвідним многочленом $p(x) = x^3 + x + 1$ вказані у таблиці 6.5.

Таблиця 6.5 – Різні представлення елементів поля $GF(2^3)$

Елементи поля	Ступінь α	Поліном	Вектор
0	0	0	000
β_0	α^0	1	001
β_1	α^1	x	010
β_2	α^2	x^2	100
β_3	α^3	$x + 1$	011
β_4	α^4	$x^2 + x$	110
β_5	α^5	$x^2 + x + 1$	111
β_6	α^6	$x^2 + 1$	101

Будь-яке розширене поле містить одиничний елемент щодо операції додавання, який прийнято позначати «0» і називати нулем. Для елемента поля α існує *адитивний зворотний елемент* «− α». Крім того, розширене поле містить одиничний елемент щодо операції множення, який позначається як "1" і називається одиницею. Для елемента поля α *існує мультиплікативний зворотний елемент «*α^{-1}». У цьому випадку операції віднімання та поділу видаються як операції складання та множення.

Наприклад,

$a-b=a+(-b)$,

$a/b=a\cdot b^{-1}$.

Для розширених полів $GF(2^m)$, (де $n=2^m-1$ − це число ненульових елементів поля) виконується рівність $\alpha^n=1$. Тоді для розширеного поля $GF(2^3)$ число ненульових елементів становить $n=2^3-1=7$ та елемент $\beta_7 \rightarrow \alpha^7=\alpha^0=1$.

Таким чином, для всіх елементів поля виконується рівність $\alpha^{n+1}=\alpha$; $\alpha^{n+2}=\alpha^2$ і т.д. І таким же способом можна знайти обернений елемент $\alpha^{-1}=\alpha^{-1+n}=\alpha^{n-1}$. Наприклад, для розширеного поля $GF(2^3)$ знайдемо обернений елемент для $\alpha^{-1} \rightarrow \alpha^{-1}=\alpha^{7-1}=\alpha^6$.

Поле $GF(2^4)$ містить $n-1=2^4-1=15$ ненульових елементів, що утворюють мультиплікативну групу, а з нульовим елементом – адитивну. Усі елементи поля видаються через ступінь примітивного елемента α, звідки випливає принцип формування правила множення, як складання ступенів $\alpha^i(i=\overline{0,n-1})$ за модулем n = 15.

Наприклад,

$\alpha^7\cdot\alpha^5=\alpha^{12}$,

$\alpha^{12}\cdot\alpha^7=\alpha^{19}=\alpha^{15+4}=\alpha^4$.

Операція ділення є операцією множення на мультиплікативно-зворотний елемент α^{n-i} або як різниця ступенів.

Наприклад,

$\alpha^{14}/\alpha^{11}=\alpha^{14}\cdot\alpha^{15-11}=\alpha^{18}=\alpha^{15+3}=\alpha^3$,

$\alpha^{14}/\alpha^{11}=\alpha^{14-11}=\alpha^3$.

З представлення α^i через многочлени від x над полем $GF(2^4)$ (додаток В) випливає реальне представлення цих елементів як наборів з m (зокрема m=4) двійкових елементів.

Приклад 6.16. Виконати додавання елементів поля $GF(2^4)$ $\alpha^7+\alpha^{14}$.

Рішення.

$\alpha^7=x^3+x+1=1011$,

$\alpha^{14}=x^3+1=1001$,

де молодші розряди розташовуються праворуч.

Звідси випливає і правило складання елементів α^i як додавання багаточленів або їх двійкових уявлень (еквівалентів).

Наприклад,

$$\alpha^7 + \alpha^{14} = (x^3 + x + 1) + (x^3 + 1) = x = \alpha$$

або

$$\begin{array}{r} \oplus 1011 \\ 1001 \\ \hline 0010 \end{array} \Rightarrow x = \alpha.$$

У таблицях 6.6 та 6.7 представлені арифметичні операції для елементів поля $GF(2^3)$.

Таблиця 6.6 – Додавання елементів поля $GF\ (2^3)$

+	**0**	**1**	$\boldsymbol{\alpha^1}$	$\boldsymbol{\alpha^2}$	$\boldsymbol{\alpha^3}$	$\boldsymbol{\alpha^4}$	$\boldsymbol{\alpha^5}$	$\boldsymbol{\alpha^6}$
0	0	1	α^1	α^2	α^3	α^4	α^5	α^6
1	1	0	α^3	α^6	α^1	α^5	α^4	α^2
$\boldsymbol{\alpha^1}$	α^1	α^3	0	α^4	1	α^2	α^6	α^5
$\boldsymbol{\alpha^2}$	α^2	α^6	α^4	0	α^5	α^1	α^3	1
$\boldsymbol{\alpha^3}$	α^3	α^1	1	α^5	0	α^6	α^2	α^4
$\boldsymbol{\alpha^4}$	α^4	α^5	α^2	α^1	α^6	0	1	α^3
$\boldsymbol{\alpha^5}$	α^5	α^4	α^6	α^3	α^2	1	0	α^1
$\boldsymbol{\alpha^6}$	α^6	α^2	α^5	1	α^4	α^3	α^1	0

Таблиця 6.7 – Множення елементів поля $GF(2^3)$

*****	**0**	**1**	$\boldsymbol{\alpha^1}$	$\boldsymbol{\alpha^2}$	$\boldsymbol{\alpha^3}$	$\boldsymbol{\alpha^4}$	$\boldsymbol{\alpha^5}$	$\boldsymbol{\alpha^6}$
0	0	0	0	0	0	0	0	0
1	0	1	α^1	α^2	α^3	α^4	α^5	α^6
$\boldsymbol{\alpha^1}$	0	α^1	α^2	α^3	α^4	α^5	α^6	1
$\boldsymbol{\alpha^2}$	0	α^2	α^3	α^4	α^5	α^6	1	α^1
$\boldsymbol{\alpha^3}$	0	α^3	α^4	α^5	α^6	1	α^1	α^2
$\boldsymbol{\alpha^4}$	0	α^4	α^5	α^6	1	α^1	α^2	α^3
$\boldsymbol{\alpha^5}$	0	α^5	α^6	1	α^1	α^2	α^3	α^4
$\boldsymbol{\alpha^6}$	0	α^6	1	α^1	α^2	α^3	α^4	α^5

Щоб не повторювати ці операції постійно, найчастіше при малих довжинах n використовують таблиці додавання та множення в полі $GF(2^m)$, побудовані спеціально для цього поля.

При побудові двійкових БЧХ кодів та їх декодуванні використовуються корені двійкових поліномів, що відповідають кодовим комбінаціям. З теорії поліномів над полем дійсних чисел (не кінцевих) відомо, що поліном ступеня m завжди має m коренів, тільки не всі вони обов'язково лежать у полі дійсних чисел. Частина коренів може бути у полі комплексних чисел як деякому розширенні поля дійсних чисел. А аналогія цьому є і в кінцевих полях. Будь-який многочлен ступеня m, у тому числі і незведний над полем $GF(2)$ (що не має коренів серед елементів цього поля), завжди має m коренів у розширенні $GF(2^m)$, і цими коренями є частина елементів поля $GF(2^m)$.

Наприклад, незвідний поліном $p(x) = x^3 + x + 1$ не має коренів в полі $GF(2) = \{0, 1\}$, тому що при підстановці 0 і 1 до многочлену $p(x)$ отримаємо

$p(0) = 0^3 + 0 + 1 = 1$; $p(1) = 1^3 + 1 + 1 = 1$.

Однак якщо використовувати як корінь елемент α^2 розширеного поля $GF(2^3)$, то отримаємо

$p(\alpha^2) = (\alpha^2)^3 + \alpha^2 + 1 = \alpha^6 + \alpha^2 + 1 = 1 + 1 = 0$.

Елементи α^i, α^{2i}, α^{4i}, …, α^{τ} з розширеного поля $GF(2^m)$, які є коренямим одного і того ж многочлена з коефіцієнтами з поля $GF(2)$, називаються *сполученими.* Ступені сполучених елементів поля утворюють *циклотомічні суміжні класи*

$$C_i = \left\{ i, 2i, 4i, 8i, ..., 2^{k-1} i \right\},$$

де i – ступінь α елемента розширеного поля $GF(2^m)$.

Повертаючись до попереднього прикладу, можна переконатися, що поліном $p(x) = x^3 + x + 1$ поряд з α^2 ще має в $GF(2^3)$ наступні корені:

- $\alpha \rightarrow p(\alpha) = (\alpha^1)^3 + \alpha + 1 = \alpha^3 + \alpha + 1 = 1 + 1 = 0$.
- $\alpha^4 \rightarrow p(\alpha^4) = (\alpha^4)^3 + \alpha^4 + 1 = \alpha^{12} + \alpha^4 + 1 = \alpha^5 + \alpha^4 + 1 = 1 + 1 = 0$.

Таким чином, елементи α, α^2, α^4 є сполученими.

Двійковий поліном найменшого ступеня, для якого елемент α^i з розширеного кінцевого поля $GF(2^m)$ є коренем, називається *мінімальним многочленом* $\varphi_i(x)$. При цьому ступінь мінімального многочлена дорівнює числу елементів (потужності) відповідного циклотомічного класу. Тоді многочлен $\varphi_s(x)$ за допомогою його коренів може бути представлений

$$\varphi_s(x) = \prod_{i_s \in C_s} (x + \alpha^{i_s}), \qquad (6.9)$$

(«–» замінений на «+», тому що обчислення здійснюються за модулем 2 та операції віднімання та складання не розрізняються).

У таблиці 6.8 представлені мінімальні многочлени кожного циклотомічного класу елементів розширеного поля $GF(2^3)$.

Таблиця 6.8 – Розподіл елементів за циклотомічними класами поля $GF(2^3)$ та мінімальні многочлени для кожного класу

Циклотомічні класи C_s	Сполучені елементи. Корені полінома $\varphi_i(x)$	Мінімальний многочлен $\varphi_i(x)$
$C_0 = \{0\}$	1	$\varphi_0(x) = x + 1$
$C_1 = \{1, 2, 4\}$	$\alpha, \alpha^2, \alpha^4$	$\varphi_1(x) = x^3 + x + 1$
$C_3 = \{3, 6, 5\}$	$\alpha^3, \alpha^6, \alpha^5$	$\varphi_3(x) = x^3 + x^2 + 1$

Розподіл елементів розширеного кінцевого поля $GF(2^4)$ за циклотомічними класами із зазначенням відповідних мінімальних многочленів представлено в додатку Д.

Відомо, що біном $x^{n-1} - 1 = x^{n-1} + 1$ ступеня $n = 2^m - 1$ має розкладання на помножувачі першого ступеня у полі $GF(2^m)$

$$x^{n-1} + 1 = (x - \beta_1)(x - \beta_2)... = \prod_{i=0}^{2^m - 2} (x + \alpha^i). \qquad (6.10)$$

З урахуванням формули (6.9) біном $x^{n-1} + 1$ має наступне розкладання на незвідні двійкові помножувачі у двійковому полі

$$(x^{n-1} + 1) = \prod_{i=0}^{M} \varphi_{l_i}(x), \qquad (6.11)$$

де M – число циклотомічних класів.

Наприклад, використовуючи знайдені мінімальні многочлени в полі $GF(2)$, корені яких належать розширеному полю $GF(2^3)$, розкладемо біном $x^{p^m - 1} - 1 = x^7 - 1 = x^7 + 1$ з коефіцієнтами з поля $GF(2)$ на прості помножувачі (незвідні поліноми) $\varphi_i(x)$. Оскільки додавання та віднімання за модулем 2

невиразні, то записи x^7-1 і x^7+1 еквівалентні $x^7+1=(x+1)(x^3+x+1)(x^3+x^2+1)$. Корені кожного з дільників бінома x^7+1 показані у табл. 6.8.

6.4 Кодування БЧХ кодів

Коди Боуза-Чоудхурі-Хоквінгема (БЧХ) представляють великий клас кодів, здатних виправляти кілька помилок. Коди БЧХ поєднують незначну надмірність з прийнятною коригуючою здатністю, забезпечуючи при цьому простоту алгоритмами кодування та декодування.

Коди БЧХ зазвичай задаються через корені породжуючого многочлена ступеня $n-k$. Дані коди визначаються поданим нижче чином.

БЧХ код довжини n, що виправляє t-кратні помилки − це циклічний блоковий код над полем $GF(p)$ кореням породжуючого многочлена якого є α^b, α^{b+1}, … α^{b+2t-1}, де α – елемент розширеного кінцевого поля $GF(p^m)$; b–ціле число.

Таким чином, породжуючий многочлен $g(x)$ БЧХ коду (n, k, d_{min}) може бути представлений найменшим загальним кратним

$$g(x) = НЗК\left[M_b(x), M_{b+1}(x), ..., M_{b+2t-1}(x)\right], \qquad (6.12)$$

де $M_j(x)$ – мінімальні многочлени елементів α^j.

БЧХ коди, які мають довжину $n = p^m-1$ і задаються породжуючим поліномом $g(x)$ (6.12), називають *примітивними кодами БЧХ* . Найбільшого поширення набули двійкові примітивні БЧХ коди довжини $n = 2^{-1}$, де m – будь-яке ціле число: n = 3, 7, 15, 31, 63, 127, …. здатні виправляти t помилок. У додатку Б (табл. Б. 3) дані параметри деяких двійкових примітивних БЧХ кодів, що породжують многочлени, які представлені у вісімковій системі числення. Однак іноді побудовані таким чином коди БЧХ можуть виправляти більше *помилок*. Тому величина $d = 2t+1$ називається *конструктивною кодовою відстанню* коду. Справжня мінімальна відстань коду d можливо більше конструктивного, тобто. ряд кодів БЧХ може виправляти помилки кратності більшої, ніж та, яку задають під час побудови цього коду.

Якщо довжина коду n є дільником 2^m-1, але не збігається з цим числом, то код називається *неприміти́вним.* У табл. Б.4 додатку Б наведено

параметри деяких непримітивних двійкових циклічних кодів і дано поліноми у восьмеричній формі.

Одним із представників непримітивних циклічних кодів є код Голея, відкритий у 1949 році ще до появи теорії кодування. Це код із довжиною n = 23, k = 12 здатний виправляти потрійні помилки. Породжуючий поліном $g(x) = x^{11}+x^9+x^7+x^6+x^5+x+1$ є дільником бінома $x^{23}-1$, тому його корінь β належить показнику 23, тобто $\beta^{23}= 1$. Циклотомічний класс C_β складається з 11 елементів $\{\alpha^1, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^9, \alpha^{18}, \alpha^{13}, \alpha^3, \alpha^6, \alpha^{12}\}$. У ряді коренів є лише 4 послідовні ступені $\alpha^1, \alpha^2, \alpha^3, \alpha^4$, тому конструктивна відстань цього коду дорівнює 5, а дійсне значення мінімальної відстані дорівнює 7. Код Голея (23, 12, 7) займає унікальне місце в теорії кодування в силу гарної упаковки сфер, проте на практиці через малу довжину код знайшов обмежене вживання.

Як правило, для двійкових БЧХ кодів значення ступеня b (першого елемента в (6.12)) вибирають b=1, що призводить до породжуючого полінома найменшого ступеню, тобто до найменшої кількості надлишкових символів у кодовому слові. Тоді коренями породжуючого многочлена є елементи розширеного поля $GF(p^m)$: $\alpha, \alpha^2, \alpha^3, \ldots \alpha^{2t}$ і відповідно породжуючий поліном (n, k, d_{min}) двійкового БЧХ кодів дорівнює

$$g(x) = НЗК\left[M_1(x), M_2(x), ..., M_{2t}(x)\right]. \qquad (6.13)$$

Набір елементів $\alpha, \alpha^2, \alpha^3, \ldots \alpha^{2t}$ називається *нулями коду БЧХ* .

Таким чином, для побудови двійкового примітивного БЧХ коду заданої довжини $n = p^m-1$, ступеня m та кількості помилок, яку необхідно виправити, t слід виконати таке:

− вибрати примітивний многочлен ступеня m та побудувати розширене поле $GF(p^m)$;

− знайти мінімальні многочлени $\varphi_j(x)$ для α^j, де j=1, …, 2t;

− знайти породжуючий поліном $g(x) = НЗК\left[\varphi_1(x), \varphi_2(x),..., \varphi_{2t}(x)\right]$.

Приклад 6.17. Побудувати примітивний БЧХ код довжини n=2^3-1=7 (m =3), який здатен виправляти одиночні помилки t= 1, якщо розширене поле $GF(2^3)$ синтезується примітивним поліномом $p(x)$=x^3+x+1.

Рішення. Для знаходження породжуючого многочлена $g(x)$ скористаємося мінімальними многочленами $\varphi_i(x)$, знайденими для

елементів розширеного поля $GF(2^3)$ $\{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$ у таблиці 6.8. Оскільки кількість помилок, що виправляються заданим кодом t=1, то необхідно використовувати $2t$=2 коренів: α, α^2. Ці елементи входять до одного суміжного класу (α, α^2, α^4). Тому мінімальним многочленом для коренів α, α^2 з табл. 6.8 є поліном третього ступеня

$\varphi_1(x) = \varphi_2(x) = x^3+x+1$.

Породжуючий поліном дорівнює

$$g(x) = НЗК\left[M_1(x), M_2(x)\right] = НЗК\left[\varphi_1(x), \varphi_2(x)\right] = x^3 + x + 1.$$

Таким чином породжуючий поліном БЧХ кода визначається виразом $g(x)=x^3+x+1$.

Оскільки ступінь многочлена $g(x)$ дорівнює 3, кількість перевірочних розрядів у кодовому слові r = 3, а кількість інформаційних символів $k = n–r$ і k = 7–3 = 4. Ми отримали код із параметрами (7, 4, 3), здатний виправляти 1-кратні помилки. ■

Приклад 6.18. Побудувати примітивний БЧХ код довжини $n=2^4-1=15$ (m=4), який здатен виправляти двократні помилки t=2, якщо розширене поле $GF(2^4)$ синтезується примітивним поліномом $p(x)=x^4+x+1$.

Рішення. Побудуємо двійковий БЧХ код довжини $n=2^4-1=15$ (m = 4), здатний виправляти помилки t=2. Елементи розширеного поля $GF(2^4)$ та мінімальні многочлени для цих елементів представлені у додатоку В. Оскільки кількість помилок, що виправляються заданим кодом t=2, то необхідно використовувати $2t$ = 4 коренів: α, α^2, α^3, α^4.

$$g(x) = НЗК\left[x^4 + x^3 + x^2 + x + 1, x^4 + x + 1\right] = x^8 + x^7 + x^6 + x^4 + 1.$$

Оскільки ступінь многочлена $g(x)$ дорівнює 8, кількість перевірочних розрядів у кодовому слові r = 8, а кількість інформаційних символів $k = n – r$ і k = 15–8 = 7. Ми отримали код із параметрами (15, 7, 5), здатний виправляти двократні помилки. ■

Лінійність циклічних БЧХ кодів означає застосовність до них всіх методів кодування та декодування, розглянутих у попередньому підрозділі 6.2.

Оскільки будь-який кодовий поліном $v(x)$ *несистематичного циклічного БЧХ коду* є добутком інформаційного $u(x)$ і породжуючого $g(x)$ поліномів компоненти відповідного вектора $v(x)$ є згорткою інформаційної послідовності $u_0, u_1, \ldots, u_{k-1}$ з послідовністю коефіцієнтів породжуючого

полінома g_0, g_1, …, g_{r-1}. Тоді коефіцієнти многочленів $u(x)$ і $g(x)$ згортаються регістром зсуву, оскільки

$$v_j = \sum_{i=0}^{r} g_i u_{j-i} . \qquad (6.14)$$

Подібна операція реалізується фільтром з кінцевою імпульсною характеристикою (КІХ-фільтром), який у свою чергу тотожний лінійному регістру зсуву представленому на рис. 6.1.

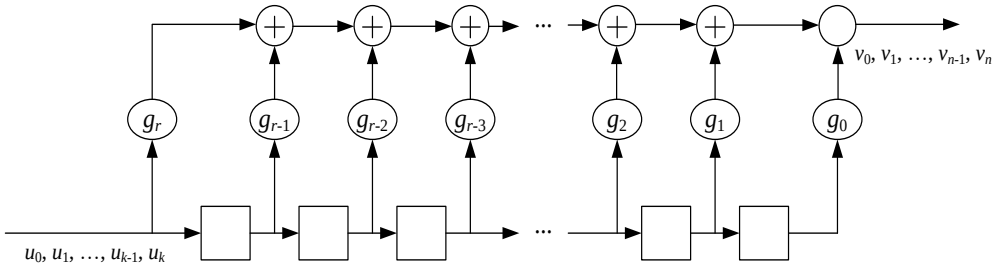


Рисунок 6.1 – Лінійний регістр зсуву без зворотнього зв'язку

На рис. 6.2 наведено приклад множення інформаційної послідовності на поліном $g(x) = x^8 + x^7 + x^4 + x^2 + x + 1$.

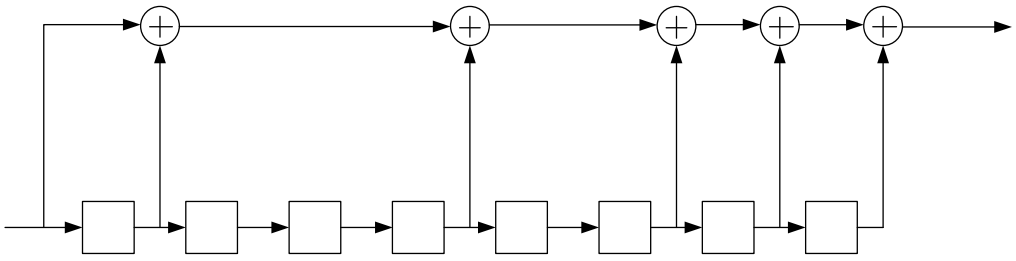


Рисунок 6.2 – Пристрій множення на багаточлен $x^8 + x^7 + x^4 + x^2 + x + 1$

Для побудови циклічних *систематичних БЧХ кодів* доцільно використовувати регістри зсуву з лінійним зворотним зв'язком, що відкриває додаткові можливості спрощення кодера в апаратній реалізації. Можна інтерпретувати n символів*, що містяться в регістрі довжини n* як коефіцієнти многочлена ступеня $n-1$. Для циклічного зсуву многочлена використовується замкнутий в кільце регістр зсуву.

Ключовою операцією в побудові систематичного кодового полінома циклічного БЧХ коду є знаходження залишку $R(x)$ від поділу добутку $u(x)$ x^r на породжуючий поліном $g(x)$. Регістр зсуву можна використовувати для поділу довільного многочлена на фіксований многочлен. Ланцюг, що виконує цю операцію, майже повторює звичайну процедуру поділу многочленів. Щоб зрозуміти, яким чином структура, подана далі, обчислює $R(x)$, розділимо $u(x)\cdot x^r = u_{k-1}\,x^{n-1} + u_{k-2}\,x^{n-2} + \ldots + u_0\,x^{n-k}$ на $g(x)=x^r + g_{r-1}x^{r-1} + \ldots + g_0$, використовуючи правило довгого поділу.

Перша ітерація

$$\begin{array}{l|l} u_{k-1}x^{n-1} & x^r+g_{r-1}+\ldots+g_0 \\ \hline u_{k-1}x^{n-1}+u_{k-1}g_{r-1}x^{n-2}+\ldots+u_{k-1}g_0x^{n-r-1} & u_{k-1}x^{n-r-1} \\ \hline r_{n-2}x^{n-2}+r_{n-3}x^{n-3}+\ldots+r_{n-r-1}x^{n-r-1}=r_1(x) & \end{array}$$

Друга ітерація

$$\begin{array}{l|l} (r_{n-2}+u_{k-2})x^{n-2}+r_{n-3}x^{n-3}+\ldots+r_{n-r-1}x^{n-r-1} & x^r+g_{r-1}+\ \ldots\ +g_0 \\ \hline (r_{n-2}+u_{k-2})x^{n-2}+\ldots+(r_{n-2}+u_{k-2})g_0x^{n-r-2} & (r_{n-2}+u_{k-2})x^{n-r-2} \\ \hline r_{n-3}x^{n-3}+r_{n-4}x^{n-4}+\ldots+r_{n-r-2}x^{n-r-2}=r_2(x) & \end{array}$$

На першій ітерації $g(x)$ множиться на $u_{k-1}x^{n-r-1}$ і результат віднімається від діленого. У результаті перший залишок $r_1(x) = r_{n-2}\,x^{n-2} + r_{n-3}\,x^{n-3} + \ldots + r_{n-r-1}\,x^{n-r-1}$ має ступінь не більше $n-1$ і після складання з $u_{k-2}x^{n-2}$ використовується на другій ітерації як нове подільне. При виконанні другої ітерації з нього віднімається $(r_{1,n-2} + u_{k-2})$ $x^{n-r-2}g(x)$, даючи другий залишок $r_2(x) = r_{n-3}x^{n-3}+r_{n-4}x^{n-4}+\ldots+r_{n-r-2}x^{n-r-2}$, ступінь якого не перевищує $n-3$. Залишок $r_2(x)$ знову підсумовується з $u_{k-3}x^{n-3}$ і використовується як подільне на третій ітерації і т.д.. Легко бачити, що на i-й ітерації попередній залишок, складений з $u_{k-i}x^{n-i}$ служить ділимим, від якого віднімається добуток полінома $g(x)$ на x у ступені, що зрівнює ступеня віднімання і ділимого, а також на коефіцієнт, що дорівнює старшому коефіцієнту ділимого. Результатом цього виявляється черговий залишок, з яким виконуються самі дії на наступній ітерації. Після k ітерації k-й залишок має ступінь, менший r, будучи необхідним залишком $r(x)$.

Таким чином, на k-му кроці обчислюються рекурентні рівності для частки $q(x)$ та залишку від поділу $R(x)$

$$\begin{aligned} q^{(r)}(x) &= q^{(r-1)}(x) + R_{n-r}^{(r-1)} x^{k-r}, \\ R^{(r)}(x) &= R^{(r-1)} + R_{n-r}^{(r-1)} x^{k-r} g(x). \end{aligned} \tag{6.15}$$

Подібний поділ реалізується систематичним циклічним кодером, показаним на рис. 6.3. і заснованим на регістрі зсуву з лінійним зворотним зв'язком.

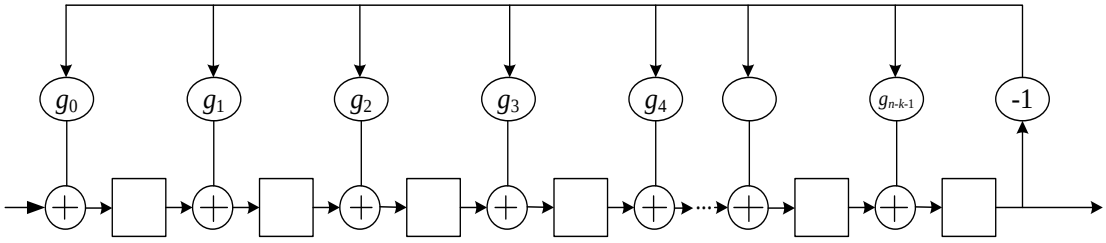


Рисунок 6.3 – Пристрої поділу на многочлен $g(x)$

Зображена на рис. 6.3 ланцюг є ланцюгом поділу довільного багаточлена на фіксований багаточлен $g(x)$. Єдиною не відображеною на ланцюзі операцією є віднімання члена $R_{n-r}x^{n-r}$ із самого себе, виконувати яку не треба, тому що результат завжди дорівнює нулю. Після n зрушень на виході регістру буде обчислена частка, а регістрі виявиться записаним залишком від поділу. На рис. 6.4 наводиться приклад реалізації ділення довільного многочлена на многочлен $g(x) = x^8 + x^7 + x^4 + x^2 + x + 1$.

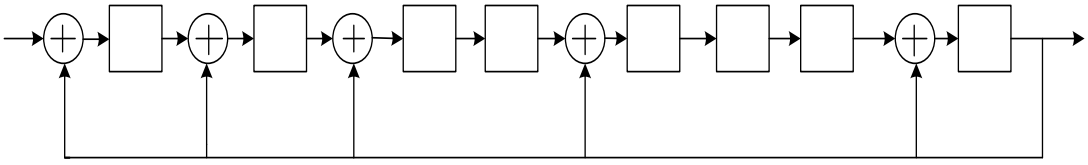


Рисунок 6.4 – Пристрій ділення на багаточлен $g(x) = x^8 + x^7 + x^4 + x^2 + x + 1$

Алгоритм кодування двійковим БЧХ кодом полягає в наступному.

1. Завдання параметрів кодера:

t – кількість помилок, що виправляються кодом;

n – довжина коду, що дорівнює 2^m-1;

b – значення ступеня елемента α^b розширеного поля $GF(2^m)$, з якого починається перерахування $2t$ коренів α^b, α^{b+1}, … α^{b+2t-1}.

2. Знаходження мінімальної кодової відстані $d_{min} = 2t\text{-}1$.

3. Побудова розширеного поля $GF(2^m)$ із заданим примітивним многочленом $p(x)$.

4. Знаходження всіх мінімальних многочленів для кожного елемента розширеного поля $GF(2^m)$: $\varphi_j(x)$ для α^j.

5. Знаходження мінімальних багаточленів $[\varphi_1(x), \varphi_2(x),..., \varphi_{2t}(x)]$ для коренів α^b, α^{b+1}, … α^{b+2t-1}.

6. Обчислення породжуючого полінома $g(x)$ як найменшого загального кратного мінімальних многочленів $g(x) = НЗК[\varphi_1(x), \varphi_2(x),..., \varphi_{2t}(x)]$.

7. Знаходження кількості перевірочних символів $r = \deg[g(x)]$.

8. Знаходження кількості інформаційних символів $k = n–r$.

9. Завдання інформаційної послідовності $u(x)$ довжини k.

10. Кодування інформаційної послідовності $u(x)$ побудованим БЧХ кодом:

− несистематичний БЧХ код $v(x) = u(x)g(x)$;

− систематичний БЧХ код $v(x) = u(x)x^r + R(x)$, $R(x)=u(x)\cdot x^r \bmod g(x)$.

Приклад 6.19. Побудувати двійковий БЧХ код довжини $n = 2^m-1 = 15$ ($m = 4$), який здатний виправляти 3-кратні помилки ($t = 3$). Розширене поле $GF(2^4)$, що синтезується примітивним многочленом $p(x)=x^4+x+1$, представлено в таблиці В.1 (додаток В). Закодувати несистематичнім та систематичнім побудованим БЧХ кодом інформаційну послідовність $u(x) = x^4+x^3+x+1$.

Рішення. Використовуючи алгоритм кодування БЧХ кодом виконаємо обчислення.

1. Параметри кода: $t = 3$, $n = 15$, $b = 1$.

2. Мінімальна кодова відстань $d_{\min}= 2t+1 = 2\cdot3+1 = 7$.

3. Побудоване розширене поле $GF(2^4)$ представлено у табл. В.1 (додаток В).

4. Мінімальні многочлени для елемента поля

$\alpha^0 - x+1$;
$\alpha^1 - x^4+x+1$;
$\alpha^2 - x^4+x+1$;
$\alpha^3 - x^4+x^3+x^2+x+1$;
$\alpha^4 - x^4+x+1$;
$\alpha^5 - x^2+x+1$;
$\alpha^6 - x^4+x^3+x^2+x+1$;
$\alpha^7 - x^4+x^3+1$;
$\alpha^8 - x^4+x+1$;
$\alpha^9 - x^4+x^3+x^2+x+1$;
$\alpha^{10} - x^2+x+1$;
$\alpha^{11} - x^4+x^3+1$;
$\alpha^{12} - x^4+x^3+x^2+x+1$;
$\alpha^{13} - x^4+x^3+1$;
$\alpha^{14} - x^4+x^3+1$.

5. Оскільки кількість помилок що може виправити заданий код $t = 3$ та $b = 1$, то необхідно використовувати елементи поля α^b, α^{b+1}, …, $\alpha^{b+2t-1} = \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$. Мінімальні багаточлени для відповідних кореней:

– $\alpha, \alpha^2, \alpha^4 \rightarrow \varphi_1(x) = x^4+x+1$;

– $\alpha^3, \alpha^6 \rightarrow \varphi_3(x) = x^4+x^3+x^2+x+1$;

– $\alpha^5 \rightarrow \varphi_5(x) = x^2+x+1$.

6. Породжуючий поліном двійкового БЧХ коду довжини n=15 визначається виразом

$$g(x) = НЗК\left[M_1(x), M_2(x), M_3(x)\right] = НЗК\left[\varphi_1(x), \varphi_3(x), \varphi_5(x)\right],$$

$$g(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1.$$

7. Кількість перевірочних символів $r = \deg[g(x)] \rightarrow r = 10$.

8. Кількість інформаційних символів $k = n–r = 15–10 = 5$.

9. Кодування інформаційної послідовності $u(x) = x^4+x^3+x+1$ несистематичним БЧХ кодом

$$v(x) = u(x)g(x).$$

$$v(x) = (x^4 + x^3 + x + 1)(x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1) = x^{14} + x^{13} + x^{11} + x^{10} +$$
$$+ x^{12} + x^{11} + +x^9 + x^8 + x^9 + x^8 + x^6 + x^5 + x^8 + x^7 + x^5 + x^4 + x^6 + x^5 +$$
$$+ x^3 + x^2 + x^5 + x^4 + x^2 + +x + x^4 + x^3 + x + 1 =$$
$$= x^{14} + x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^4 + 1.$$

10. Кодування інформаційної послідовності $u(x) = x^4+x^3+x+1$ систематичним БЧХ кодом

$v(x) = u(x)x^r + R(x)$, $R(x) = u(x)\cdot x^r \bmod g(x)$.

$u(x)\cdot x^r = (x^4+x^3+x+1)\cdot x^{10} = x^{14}+x^{13}+x^{11}+x^{10}$;

$R(x) = u(x)\cdot x^r \bmod g(x) = (x^{14}+x^{13}+x^{11}+x^{10}) \bmod (x^{10}+x^8+x^5+x^4+x^2+x+1)$

$$
\begin{array}{l|l}
x^{14}+x^{13}+x^{11}+x^{10} & x^{10}+x^8+x^5+x^4+x^2+x+1 \\
\underline{x^{14}+x^{12}+x^9+x^8+x^6+x^5+x^4} & \overline{x^4+x^3+1} \\
x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^6+x^5+x^4 & \\
\underline{x^{13}+x^{11}+x^8+x^7+x^5+x^4+x^3} & \\
x^{10}+x^9+x^7+x^6+x^3 & \\
\underline{x^{10}+x^8+x^5+x^4+x^2+x+1} & \\
x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1 &
\end{array}
$$

$R(x) = x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1,$

$v(x) = u(x)x^r + R(x) = x^{14}+x^{13}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1.$ ■

6.5 Декодування БЧХ кодів

Коди БЧХ є циклічними, і, отже, до них застосовні будь-які методи декодування циклічних кодів. Однак є значно кращі алгоритми, які спеціально розроблені для декодування БЧХ кодів. Історично перший метод декодування був знайдений Пітерсоном у 1960 р. для двійкових кодів, а також Горенстейном та Цирлером у 1961 р. для недвійкових кодів. Істотне просування у бік спрощення алгоритму було знайдено Берлекемпом в 1968 р. і надалі вдосконалено Мессі в 1969 р.

Для опису принципів декодування БЧХ кодів введемо основні позначення, що використовуються під час роботи алгоритмів виправлення помилок у прийнятій кодовій послідовності $f(x)$, якщо передано кодове слово $v(x)$.

Нехай під час передачі кодової послідовності $v(x)$ сталося v помилок і був прийнятий поліном

$$f(x)=v(x)+e(x), \tag{6.16}$$

де $e(x)$ – многочлен помилок

$$e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \ldots + e_{j_v} x^{j_v}, \tag{6.17}$$

де e_{j_l} – величина l-ої помилки (якщо двійковий БЧХ код $e_{j_l} = 1$), v – число помилок у прийнятому слові ($0 \le v \le t$).

Декодеру невідомі ані позиції помилок – j_1, j_2, … j_v, ані значення помилок – $e_{j_1}, e_{j_2}, ..., e_{j_v}$, ні кількість помилок, що відбулися під час передачі кодової послідовності. Для виправлення помилок необхідно знайти значення всіх цих чисел.

Головною ідеєю декодування БЧХ кодів є використання елементів кінцевого поля для нумерації позицій кодового слова. Наприклад, для прийнятого кодового вектора $f(x) = (f_0, f_1, \ldots, f_{n-1})$ нумерація позицій кодового слова відповідає елементам поля $GF(p^m)$. Нумерація цих позицій називаються *локаторами позицій*.

Значення кодового вектора	f_0	f_1	$f_2 \ldots f_{n-1}$	
Локатори позицій	1	α^1	$\alpha^2 \ldots \alpha^{n-1}$	

Оскільки α^i – локатор i-й позиції, α^i різні для всіх i, то локатори і позиції однозначно визначають одне одного. Таким чином, у многочлені помилок $e(x)$ позиції помилок j_1, j_2, … j_v можуть бути позначені через *локатори помилок*: α^{j_k} – локатор k-ї помилки, $k = 1, 2, \ldots t$.

Тоді необхідно знайти такі множини:

$\{e_{j_1}, e_{j_2}, ..., e_{j_v}\}$ – значення помилок, для двійкових БЧХ кодів значення помилок e_j належать $\{0, 1\}$.

$\{\alpha^{j_1}, \alpha^{j_2}, ..., \alpha^{j_v}\}$ – локатори помилок, де елементи α^i належать розширеному полю $GF(p^m)$.

Як було зазначено раніше, елементи поля $GF(p^m)$ є нулями коду в точках $f(\alpha^i) = 0$, якщо вони належать кореням породжуючого полінома $g(x)$, тобто для будь-яких α^i, що є коренями $g(x)$, виконується рівність

$$f(\alpha^j) = v(\alpha^j) + e(\alpha^j) = e(\alpha^j). \qquad (6.18)$$

Таким чином, $f(\alpha^j) = \sum_{j=0}^{n-1} e_j \alpha_i^j$, де $i = 1, 2, \ldots r$ для всіх α^j є коренямим $g(x)$.

Значення прийнятого кодового многочлена $f(x)$ в нулях коду називаються синдромами та позначаються S_j. *Синдром* прийнятого кодового слова $f(x)$ складається з $2t$ компонент: $S_j = f(\alpha^i)$ для $j = 1, 2, \ldots 2t$,

i=b, b+1, …, b+2t −1, при цьому не залежить від переданого слова та визначається лише поліномом помилок

$$S_j = f(\alpha^i) = v(\alpha^i) + e(\alpha^i) = e(\alpha^i),$$

де $e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + ... + e_{j_v} x^{j_v}$.

Тоді, з урахуванням коренів породжуючого полінома – нулів коду $v(\alpha^i) = 0$, отримаємо наступну систему з $2t$ рівнянь щодо v невідомих локаторів помилок α^j і v невідомих величин помилок e_j

$$\begin{aligned}
S_1 &= f(\alpha^b) = e_{j_1}\alpha^{bj_1} + e_{j_2}\alpha^{bj_2} + ... + e_{j_v}\alpha^{bj_v} \\
S_2 &= f(\alpha^{b+1}) = e_{j_1}\alpha^{(b+1)j_1} + e_{j_2}\alpha^{(b+1)j_2} + ... + e_{j_v}\alpha^{(b+1)j_v} \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
S_{2t} &= f(\alpha^{b+2t-1}) = e_{j_1}\alpha^{(b+2t-1)j_1} + e_{j_2}\alpha^{(b+2t-1)j_2} + ... + e_{j_v}\alpha^{(b+2t-1)j_v}
\end{aligned} \tag{6.19}$$

Для двійкового БЧХ коду зі значенням b=1 отримаємо компоненти синдрому

$$\begin{aligned}
S_1 &= e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + ... + e_{j_v}\alpha^{j_v} \\
S_2 &= e_{j_1}\alpha^{2j_1} + e_{j_2}\alpha^{2j_2} + ... + e_{j_v}\alpha^{2j_v} \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
S_{2t} &= e_{j_v}\alpha^{2tj_1} + e_{j_2}\alpha^{2tj_2} + ... + e_{j_v}\alpha^{2tj_v}.
\end{aligned} \tag{6.20}$$

За допомогою отриманої системи рівнянь, знаючи значення компонентів синдрому S={S_1, S_2, …, S_{2t}}, можна визначити число помилок v, їх локатори α^j і їх величини e_j. Після цього легко знайти компоненти синдромного вектора за прийнятою послідовності. Обчислення синдромів БЧХ коду S_j за прийнятою послідовністю $f(x)$ здійснюється шляхом ділення многочлена $f(x)$ на мінімальні многочлени $\varphi_i(x)$. В результаті ділення знаходяться залишки

$$R_j(x) = f(x) \bmod \varphi_j(x), j = 1, 2, \ldots 2t. \tag{6.21}$$

Далі компоненти синдрому перебувають як значення залишків у нулях коду {α^j}

$$S_j = R_j(\alpha^j). \tag{6.22}$$

При цьому компоненти S_j з непарними номерами (j =1, 3, … $2t$−1) обчислюються за формулою (6.22), а компоненти S_j з парними номерами (j = 2, 4, … $2t$) обчислюються

$$S_{2j} = S_j^2. \tag{6.23}$$

Однак систему рівнянь (6.20) важко вирішувати безпосередньо, тому скористаємося штучним прийомом, визначивши деякі проміжні змінні, які можуть бути обчислені за компонентами синдрому і за якими можна обчислити локатори помилок. Для цього введемо *поліном локаторів помилок* від змінної x

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + ... + \sigma_v x^v. \tag{6.24}$$

Поліном локаторів помилок визначається як многочлен, коренями якого є величини, *обернені до локаторів помилок* α^j

$$\sigma(x) = (1 - x\alpha^{j_1})(1 - x\alpha^{j_2}) ... (1 - x\alpha^{j_v}) = \prod_{l=1}^{v} (1 + \alpha^{j_l}). \tag{6.25}$$

Якщо відомі коефіцієнти многочлена $\sigma(x)$, то обчислення локаторів помилок потрібно знайти його корені. Тому за заданими компонентами синдрому необхідно обчислити коефіцієнти $\sigma_1, \sigma_2, \ldots \sigma_v$.

Рівняння, що зв'язують синдромні компоненти та коефіцієнти полінома локаторів помилок, мають вигляд

$$S_{j+v} + \sigma_1 S_{j+v-1} + \sigma_2 S_{j+v-2} + ... + \sigma_v S_j = 0, \; j = 1, ... v. \tag{6.26}$$

Таким чином, отримуємо систему рівнянь

$$\sigma_1 S_{j+v-1} + \sigma_2 S_{j+v-2} + ... + \sigma_v S_j = -S_{j+v}, j = 1, ... v,$$

тобто систему лінійних рівнянь, що зв'язують компоненти синдрому з коефіцієнтами поліному $\sigma(x)$. Цю систему можна записати у матричному вигляді

$$\begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{v-1} & S_v \\ S_2 & S_3 & S_4 & \cdots & S_v & S_{v+1} \\ S_3 & S_4 & S_5 & \cdots & S_{v+1} & S_{v+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ S_v & S_{v+1} & S_{v+2} & \cdots & S_{2v-2} & S_{2v-1} \end{bmatrix} \begin{bmatrix} \sigma_v \\ \sigma_{v-1} \\ \sigma_{v-2} \\ \vdots \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_{v+1} \\ -S_{v+2} \\ -S_{v+3} \\ \vdots \\ -S_{2v} \end{bmatrix}. \qquad (6.27)$$

Ця система рівнянь отримала назву – *ключове рівняння декодування БЧХ кодів*. Вирішуючи ключове рівняння (6.27), ми отримаємо коефіцієнти полінома локаторів помилок $\sigma(x)$. Позиції помилок $j_1, j_2, \ldots j_v$ знаходяться як елементи поля $GF(p^m)$, обернені до коренів полінома локаторів помилок $\sigma(x)$. За знайденими локаторами визначаються величини помилок із системи рівнянь (6.20).

Рішення ключового рівняння потребує інтенсивних обчислень у процедурі декодування БЧХ кодів. В даний час відомі такі методи вирішення ключового рівняння:

1. *Алгоритм Берлекемпа-Мессі* (англ. *Berlekamp–Massey algorithm, BMA*).
2. *Алгоритм Евкліда* (англ. *Euclidean algorithm, EA*).
3. Пряме рішення – *алгоритм Пітерсона-Горенстейна-Цірлера* (англ. *Peterson–Gorenstein–Zierler, PGZ*).

Таким чином, загальний алгоритм декодування циклічних БЧХ кодів включає реалізацію наступних завдань.

1. Обчислити значення компонентів синдрому S_j прийнятого полінома $f(x)$ в нулях коду $S_j = f(\alpha^i)$ ($j = 1, 2, \ldots, 2t$, $i = b, b+1, \ldots, b+2t-1$).

2. Знайти коефіцієнти $\sigma_1, \sigma_2, \ldots \sigma_v$ і побудувати поліном локаторів помилок $\sigma(x)$ використовуючи один з алгоритмів розв'язання ключового рівняння (BMA , EA , PGZ).

3. Знайти корені полінома локаторів помилок, якими є елементи α^i розширеного поля $GF(p^m)$. Для пошуку коренів $\sigma(x)$ використовується *метод Ченя* – послідовна підстановка всіх ненульових елементів α^i ($i = 0, 1, \ldots 2^m-2$) та перевірка умови $\sigma(\alpha^i) = 0$.

4. Знайти значення позицій помилок $j_1\, j_2 \ldots j_v$ як обернені величини знайденим кореням $\alpha^i=\alpha^{-j}$. Зворотний елемент $\beta^{-1}=\alpha^b$ елемента $\beta=\alpha^k$ визначається як $b = 2^m - 1 - k$. Також обернені елементи можуть бути знайдені за допомогою таблиці множення елементів поля $GF(p^m)$.

5. Знайти значення помилок (етап непотрібний для бінарних кодів).
6. Виправити прийняте кодове слово.

Рішення ключового рівняння – Алгоритм Берлекемпа-Мессі

Основними обчисленнями при декодуванні БЧХ кодів є вирішення системи рівнянь (6.27). Оскільки рядки матриці S лінійно незалежні, то визначник матриці S не дорівнює нулю і матриця оборотна, що дає єдине рішення $(\sigma_1, \sigma_2, \ldots, \sigma_v)$ системи (6.27). При невеликих значеннях v цю систему можна вирішити за допомогою звернення матриці і для цього необхідно зробити близько v^3операцій. Однак на практиці часто потрібно використовувати коди, що виправляють велику кількість помилок і тоді потрібен ефективніший метод вирішення ключового рівняння. Такий метод був відкритий Е.Берлекемпом в 1968 році і застосований до лінійних кодів Джеймсом Мессі в 1969.

Припустимо, що нам відомий вектор, тоді перший рядок системи рівнянь визначає значення S_{v+1} через значення $S_1, \ldots, S_v$, другий рядок визначає S_{v+2} через $S_2, \ldots, S_{v+1}$ і т.д.

$$
\begin{aligned}
&S_{v+1} = \sigma_1 S_v + \sigma_2 S_{v-1} + \sigma_3 S_{v-2} + \ \ldots \ + \sigma_{v-1} S_2 + \sigma_v S_1;\\
&S_{v+2} = \sigma_1 S_{v+1} + \sigma_2 S_v + \sigma_3 S_{v-1} + \ \ldots \ + \sigma_{v-1} S_3 + \sigma_v S_2;\\
&S_{v+3} = \sigma_1 S_{v+2} + \sigma_2 S_{v+1} + \sigma_3 S_v + \ \ldots \ + \sigma_{v-1} S_4 + \sigma_v S_3;\\
&\quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots\\
&S_{2v-1} = \sigma_1 S_{2v} + \sigma_2 S_{2v-3} + \sigma_3 S_{2v-4} + \ldots + \sigma_{v-1} S_v + \sigma_v S_{v-1}\\
&S_{2v} = \sigma_1 S_{2v-1} + \sigma_2 S_{2v} + \sigma_3 S_{2v-3} + \ \ldots \ + \sigma_{v-1} S_{v+1} + \sigma_v S_v.
\end{aligned}
\qquad (6.28)
$$

Для фіксованого σ це рівняння визначає регістр зсуву з лінійним зворотним зв'язком, множники у відводах якого задаються вектором σ. На рисунку 6.5 представлений регістр зсуву зі зворотним зв'язком, побудований для кожного рядка системи рівнянь (6.28). При цьому в початковому стані лінійний регістр зсуву містить послідовність $S_v S_{v-1} \ldots S_1$. Для кожного такту на виході регістра зсуву утворюється елемент синдрому помилки $S_1, S_2, \ldots, S_{2v}$.

Загалом цей послідовний процес описується рівнянням

$$S_j = -\sum_{i=1}^{v} \sigma_i S_{j-i}, \ j = v+1, \ldots, 2v. \qquad (6.29)$$

Переформульоване таким чином завдання зводиться до побудови регістру зсуву з лінійним зворотним зв'язком, що генерує відому послідовність компонент синдрому. Завдання полягає в тому, щоб серед великої кількості таких регістрів знайти регістр зсуву з найменшою довжиною. Це дозволить визначити вектор помилки мінімальної ваги у прийнятому слові, тобто. визначити багаточлен локаторів помилок $\sigma(x)$ найменшого ступеня.

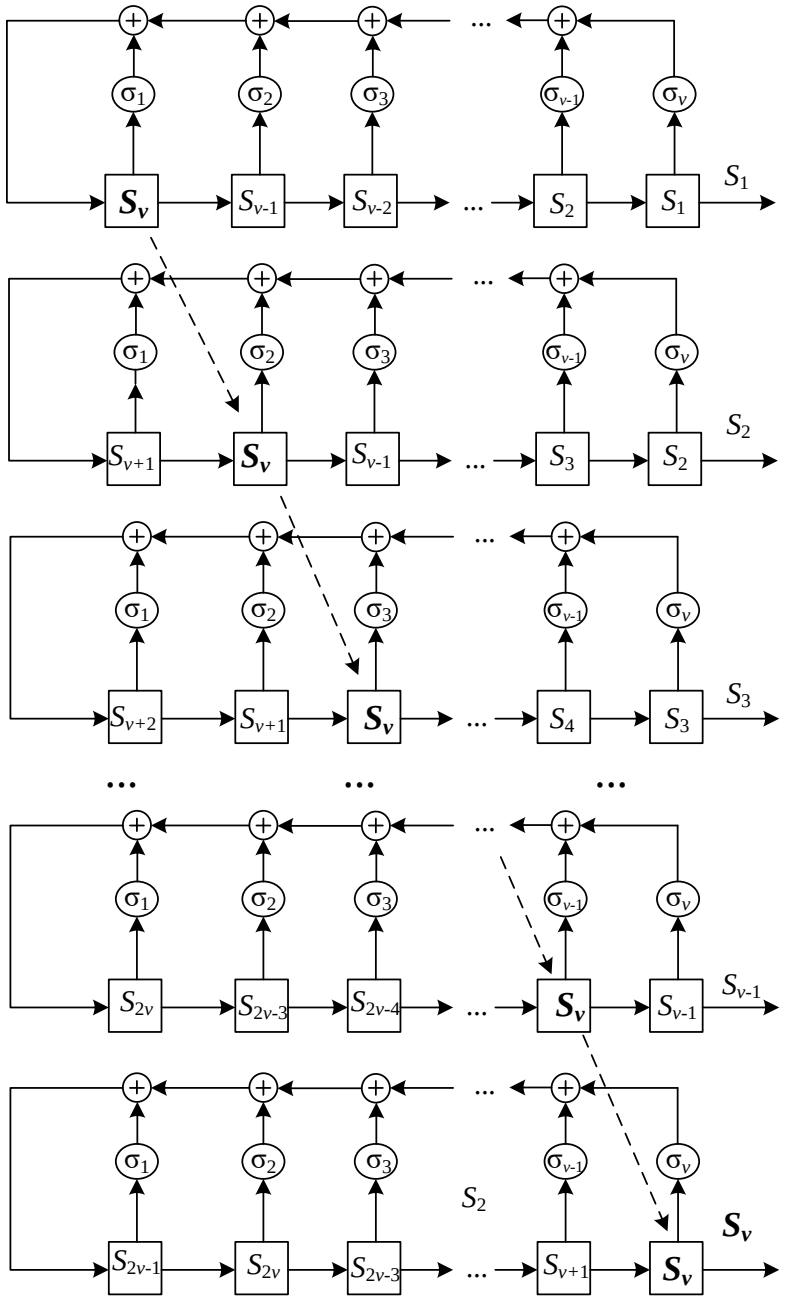


Рисунок 6.5 – Багаточлен локаторів помилок у ланцюзі регістру зсуву

Таким чином, алгоритм Берлекемпа-Мессі найкраще розглядати як ітеративний процес побудови мінімального лінійного регістру (зсуву) із зворотним зв'язком (ЛРЗЗ), у загальному вигляді показаного на рис. 6.6. який генерує відому послідовність компонент синдрому $S_1, S_2, \ldots, S_{2t}$.

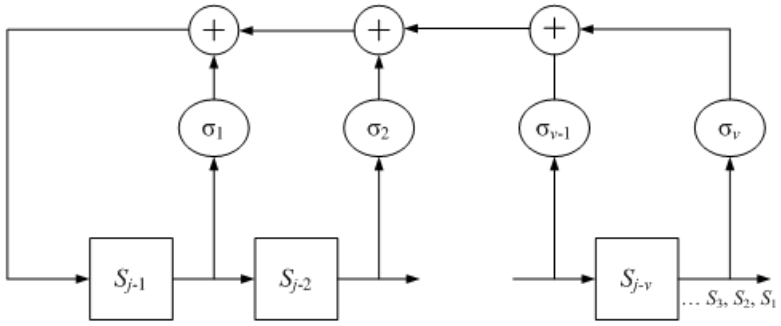


Рисунок 6.6 – ЛРЗЗ із відведеннями $\sigma_1, \sigma_2, \ldots \sigma_v$ та виходами $S_1, S_2, \ldots, S_{2t}$

Для побудови необхідного регістру зсуву потрібно знайти дві величини:

− довжину регістра зсуву L;

− багаточлен зворотного зв'язку $\sigma(x) = \sigma_v x^v + \sigma_{v-1} x^{v-1} + \ \ldots \ + \sigma x + 1$, де $\deg[\sigma(x)] \le L$.

Таким чином, необхідно знайти регістр зсуву із зворотним зв'язком найменшої довжини, який генерує послідовність $S_1, S_2, \ldots, S_{2t}$ за відповідного початкового стану. Для кожної ітерації i, починаючи з $k = 1$, будується регістр зсуву, що генерує перші k компонент синдрому, і метою алгоритму Берлекемпа-Мессі є побудова многочлена (зворотного зв'язку) $\sigma^{(k)}(x)$ найменшого ступеня, що задовольняє наступному рівнянню

$$\sum_{j=0}^{L_k} S_{k-j} \sigma_j^{(k)} = 0. \qquad (6.30)$$

Вирішення цього завдання еквівалентне умові, що поліном

$$\sigma^{(k)}(x) = 1 + \sigma_1^{(k)} x + \ldots + \sigma_{L_k}^{(k)} x^{L_k} \qquad (6.31)$$

є многочленом зворотного зв'язку ЛРЗЗ, який генерує обмежену послідовність синдромів.

Регістр зсуву з довжиною L_k та коефіцієнтами $\sigma^{(k)}(x)$ є регістром зсуву мінімальної довжини, що генерує S_1, S_2, …, S_k і на початок k – ітерації вже буде заданий список регістрів зсуву: L_1, $\sigma^{(1)}(x)$; L_2, $\sigma^{(2)}(x)$; ..., L_{k-1}, $\sigma^{(k-1)}(x)$. Необхідно знайти спосіб побудови нового регістра мінімальної довжини L_k, $\sigma^{(k)}(x)$, що генерує послідовність S_1, S_2, …, S_{k-1}, S_k. Це можна виконати, використовуючи попередній регістр зсуву, в якому за потреби будуть належним чином змінені довжина та вагові множники. Тоді на k – ітерації обчислимо наступний вихід (k–1)-го регістру зсуву

$$S_k^{'} = -\sum_{j=1}^{n-1} \sigma_j^{(r-1)} S_{k-j} \,. \tag{6.32}$$

Віднімаючи $S_k^{'}$ з необхідного виходу S_k отримуємо величину d_k, що називається *несумісністю* (нев'язкою)

$$d_k = S_k - S_k^{'} = S_k + \sum_{j=1}^{n-1} \sigma_j^{(k-1)} S_{k-j} \,, \tag{6.33}$$

яка є мірою відповідності синдромної послідовності і генерується лінійним регістром зсуву зі зворотним зв'язком, а також містить коригуючий множник для обчислення $\sigma^k(x)$ на наступній ітерації. При цьому можливі два випадки:

− якщо $d_k = 0$, то рівняння (6.32) задовольняє рівності

$$\sigma^{(k)}(x) = \sigma^{(k-1)}(x),\ L_k = L_{k-1} \tag{6.34}$$

і k-та ітерація буде закінчена;

− якщо $d_k \neq 0$, то необхідно змінити вагові множники в ланцюзі зворотного зв'язку регістру зсуву

$$\sigma^{(k)}(x) = \sigma^{(k-1)}(x) + d_k d_m^{-1} x^{k-m} \sigma^{(m-1)}(x) \,, \tag{6.35}$$

де $\sigma^{(m-1)}(x)$ – один із багаточленів регістру зсуву, що зустрічалися в одній із попередніх ітерацій, при цьому вибирається $m < k$ і таке, що $d_m \neq 0$, тобто m дорівнює номеру останнього кроку, що передує кроку k, на якому $d_m \neq 0$.

Ітеративне обчислення $\sigma^{(k)}(x)$ триває доки задовольняються одне чи обидві умови:

$$k-1 \geq L_k + t - 1 \quad або \quad k = 2t.$$

Початковими умовами алгоритму є

$\sigma(x) = 1$, $L = 0$, $k = 0$, $B(x) = 1$.

Приклад 6.20. Знайти регістр зсуву із зворотним зв'язком найменшої довжини, який генерує послідовність $s_1, s_2, \ldots = 010111100$.

Рішення. Знайдемо довжину регістра зсуву L. Якби послідовність складалася з одних нулів, то ми б зробили висновок, що $L = 0$. У разі послідовності з усіх одиниць $L=1$ $\sigma_1=1$, а рівняння (6.27) має вигляд $s_i+\sigma_1 s_{i-1}=0$. Оскільки це відповідає умові завдання, то перевіряємо гіпотезу у тому, що $L = 2$. Система (6.27) тоді має вигляд

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Єдине рішення рівняння $\sigma_1=0$, $\sigma_2=1$. Перевіримо, чи породжує відповідний регістр зсуву всю послідовність

$\sigma_1 s_2+\sigma_2 s_1=\sigma_1 1+\sigma_2 0 = 0 = s_3$;

$\sigma_1 s_3+\sigma_2 s_2=\sigma_1 0+\sigma_2 1 = 1 = s_4$;

$\sigma_1 s_4+\sigma_2 s_3=\sigma_1 1+\sigma_2 0 = 0 \neq s_5 = 1$.

На п'ятому символі отримали розбіжність, отже, гіпотеза неправильна. Перевіримо гіпотезу $L = 3$.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_3 \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Вирішення даної системи $\sigma_1=0$, $\sigma_2=1$, $\sigma_3=1$. Тоді отримаємо

$\sigma_1 s_3+\sigma_2 s_2+\sigma_3 s_1= \sigma_1 1+\sigma_2 0+\sigma_3 0 = 1 = s_4$;

$\sigma_1 s_4+\sigma_2 s_3+\sigma_3 s_2= \sigma_1 1+\sigma_2 0+\sigma_3 1 = 1 = s_5$;

$\sigma_1 s_5+\sigma_2 s_4+\sigma_3 s_3= \sigma_1 1+\sigma_2 1+\sigma_3 0 = 1 = s_6$;

$\sigma_1 s_6+\sigma_2 s_5+\sigma_3 s_4= \sigma_1 1+\sigma_2 1+\sigma_3 1 = 0 \neq s_7 = 1$.

Знову отримали розбіжність, але вже на сьомому символі.

Перевіримо гіпотезу $L = 4$.

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_4 \\ \sigma_3 \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Вирішення даної системи $\sigma_1=0$, $\sigma_2=0$, $\sigma_3=1$, $\sigma_4=1$. Тоді отримаємо

$\sigma_1 s_4+\sigma_2 s_3+\sigma_3 s_2+\sigma_4 s_1= \sigma_1 1+\sigma_2 0+\sigma_3 1+\sigma_4 0 = 1 = s_5$;

$\sigma_1 s_5+\sigma_2 s_4+\sigma_3 s_3+\sigma_4 s_2= \sigma_1 1+\sigma_2 1+\sigma_3 0+\sigma_4 1 = 1 = s_6$;

$\sigma_1 s_6+\sigma_2 s_5+\sigma_3 s_4+\sigma_4 s_3= \sigma_1 1+\sigma_2 1+\sigma_3 1+\sigma_4 0 = 1 = s_7$;

$\sigma_1 s_7+\sigma_2 s_6+\sigma_3 s_5+\sigma_4 s_4= \sigma_1 1+\sigma_2 1+\sigma_3 1+\sigma_4 1 = 0 = s_8$;

$\sigma_1 s_8+\sigma_2 s_7+\sigma_3 s_6+\sigma_4 s_5= \sigma_1 0+\sigma_2 1+\sigma_3 1+\sigma_4 1 = 0 = s_9$.

Таким чином, завдання вирішено. ■

В алгоритмі використовуються такі позначення:

S_j – компоненти синдрому помилки;

$\sigma(x)$ – розраховується поліном локаторів помилок;

k – номер ітерації алгоритму;

d_k – значення нев'язки, тобто сума лівої та правої частин k+1-го рівняння, що обертається в нуль, якщо поліном локаторів задовольняє рівняння;

m – номер кроку, на якому поліном локаторів помилок останній раз модифікувався;

L – кількість членів у лівій частині рівнянь, що відображає кількість спотворених символів, що передбачається під час ітерацій алгоритму;

$B(x)$ – поліном модифікації, який використовується як допоміжний для модифікації полінома локаторів для «підстроювання» його під рівняння системи.

Починаючи з полінома локаторів помилок $\sigma(x)=1$, обчислюється нев'язка цього полінома першому рівнянню. Далі враховуючи цю нев'язку, робиться корекція полінома локаторів, при необхідності нарощується його ступінь, а далі все повторюється так, щоб за умови задоволення всіх попередніх рівнянь він також став задовольняти поточному рівнянню, і так доти, доки не будуть задоволені всі рівняння. Оскільки в алгоритмі ступінь полінома нарощується лише за необхідності, то рішенням і буде поліном мінімального ступеня.

В результаті роботи алгоритму, якщо справжня кількість спотворених символів більше, ніж кратність помилок, що виправляються,

можливо (але необов'язково), що ступінь полінома локаторів не буде збігатися з передбачуваною кількістю спотворених символів, тобто. $L \neq \deg[\sigma(x)]$. І тут вважається, що рішення системи рівнянь не знайдено. Цей випадок слід трактувати як неможливість виправлення помилок. Однак, якщо навіть $L = \deg[\sigma(x)]$, це все одно не є стовідсотковою гарантією того, що знайдено коректний поліном локаторів. Якщо поліном локаторів $\sigma(x)$ успішно знайдено, то тоді знаходять усі корені полінома локаторів помилок $\sigma(\alpha^j) = 0$ шляхом перебору всіх ненульових елементів поля Галуа.

Приклад 6.21. Виконати декодування кодового повідомлення $f(x) = x^{14}+x^{11}+x^8+x^6+x^4+x^3+x^2+x+1$ для БЧХ коду (15, 5,7). Породжуючий поліном коду дорівнює $g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$, b=1, t=3 Розширене поле $GF(2^4)$ має базовий примітивній поліном $p(x) = x^4+x+1$ (Додаток В).

Рішення.

1. Знайдемо компоненти синдромного вектора по формулі

$S_i = f(\alpha^j)$, i=1, …, 2t; j=1, …, 2t.

$S_1=f(\alpha)=(\alpha)^{14}+(\alpha)^{11}+(\alpha)^8+(\alpha)^6+(\alpha)^4+(\alpha)^3+(\alpha)^2+(\alpha)+1=\underline{\alpha.}$

$S_2=f(\alpha^2)=(\alpha^2)^{14}+(\alpha^2)^{11}+(\alpha^2)^8+(\alpha^2)^6+(\alpha^2)^4+(\alpha^2)^3+(\alpha^2)^2+(\alpha^2)+1=\alpha^{28}+\alpha^{22}+\alpha^{16}+\alpha^{12}+$
$+\alpha^8+\alpha^6+\alpha^4+\alpha^2+1= \alpha^{13}+\alpha^7+\alpha^1+\alpha^{12}+\alpha^8+\alpha^6+\alpha^4+\alpha^2+1= \alpha^5+\alpha^{13}+\alpha^{14}+\alpha^{10}+1= \underline{\alpha^2.}$

$S_3 = f(\alpha^3)=(\alpha^3)^{14}+(\alpha^3)^{11}+(\alpha^3)^8+(\alpha^3)^6+(\alpha^3)^4+(\alpha^3)^3+(\alpha^3)^2+(\alpha^3)^1+1=\alpha^{42}+\alpha^{33}+\alpha^{24}+\alpha^1$
$^8+\alpha^{12}+\alpha^9+\alpha^6+\alpha^3+1 = \alpha^{42}+\alpha^{33}+\alpha^{24}+\alpha^{18}+\alpha^{12}+\alpha^9+\alpha^6+\alpha^3+1=$
$= \alpha^{12}+\alpha^3+\alpha^9+\alpha^3+\alpha^{12}+\alpha^9+\alpha^6+\alpha^3+1=\underline{\alpha^8.}$

$S_4 = f(\alpha^4)=(\alpha^4)^{14}+(\alpha^4)^{11}+(\alpha^4)^8+(\alpha^4)^6+(\alpha^4)^4+(\alpha^4)^3+(\alpha^4)^2+(\alpha^4)^1+1=\alpha^{56}+\alpha^{44}+\alpha^{32}+\alpha^2$
$^4+\alpha^{16}+\alpha^{12}+\alpha^8+\alpha^4+1 = \alpha^{11}+\alpha^{14}+\alpha^2+\alpha^9+\alpha^1+\alpha^{12}+\alpha^8+\alpha^4+1=$
$= \alpha^{10}+\alpha^{11}+\alpha^{13}+\alpha^5+1=\underline{\alpha^4.}$

$S_5 = f(\alpha^5)=(\alpha^5)^{14}+(\alpha^5)^{11}+(\alpha^5)^8+(\alpha^5)^6+(\alpha^5)^4+(\alpha^5)^3+(\alpha^5)^2+(\alpha^5)^1+1=\alpha^{70}+\alpha^{55}+\alpha^{40}+\alpha^3$
$^0+\alpha^{20}+\alpha^{15}+\alpha^{10}+\alpha^5+1 = \alpha^{10}+\alpha^{10}+\alpha^{10}+\alpha^0+\alpha^5+\alpha^0+\alpha^{10}+\alpha^5+1=\underline{1.}$

$S_6 = f(\alpha^6)=(\alpha^6)^{14}+(\alpha^6)^{11}+(\alpha^6)^8+(\alpha^6)^6+(\alpha^6)^4+(\alpha^6)^3+(\alpha^6)^2+(\alpha^6)^1+1=\alpha^{84}+\alpha^{66}+\alpha^{48}+\alpha^3$
$^6+\alpha^{24}+\alpha^{18}+\alpha^{12}+\alpha^6+1 = \alpha^9+\alpha^6+\alpha^3+\alpha^6+\alpha^9+\alpha^3+\alpha^{12}+\alpha^6+1=\underline{\alpha^1.}$

$S_1 = \alpha^1$, $S_2 = \alpha^2$, $S_3 = \alpha^8$, $S_4 = \alpha^4$, $S_5 = 1$, $S_6 = \alpha^1$.

2. Виконаємо алгоритм Берлекемпа-Мессі та знайдемо поліном локаторів помилок

Ініціалізація.

$\sigma(x) = 1$, $L = 0$, $k = 0$, $B(x) = 1$.

Крок 1. $k = 1$.

$$d_1 = S_1 + \sum_{j=1}^{L=0} \sigma_{j-1} S_{1-j} = S_1 = \alpha;$$

d_1=0, α = 0 ? – ні → модифікація многочлена зворотного зв'язку;

$\sigma^*(x) = \sigma(x) + d_1 \cdot x \cdot B(x) = 1 + \alpha \cdot x$;

$2L \le k-1$, $0 \le 0$? – так → довжина регістру збільшується;

$B(x) = d_1^{-1} \cdot \sigma(x) = \alpha^{14} \cdot 1 = \alpha^{14}$; {для $d_1 = \alpha$ зворотній елемент $d_1^{-1} = \alpha^{14}$}

$L = k - L = 1 - 0 = 1$;

$\sigma(x) = \sigma^*(x) = 1 + \alpha \cdot x$;

$k = 2t$, 1 = 6 ? – ні → збільшуємо шаг ітерації;

$k = k+1 = 1+1 = 2$.

Крок 2. $k = 2$.

$$d_2 = S_2 + \sum_{j=1}^{L=1} \sigma_{j-1} S_{2-j} = S_2 + \sigma_1 S_1 = \alpha^2 + \alpha \cdot \alpha = 0;$$

$d_2 = 0$, 0 = 0 ? – так → без модифікації многочлена зворотного зв'язку;

$B(x) = x \cdot B(x) = x \cdot \alpha^{14} = \alpha^{14} x$;

$k = 2t$, 2 = 6 ? – ні → збільшуємо шаг ітерації;

$k = k+1 = 2+1 = 3$.

Крок 3. $k = 3$.

$$d_3 = S_3 + \sum_{j=1}^{L=1} \sigma_{j-1} S_{3-j} = S_3 + \sigma_1 S_2 = \alpha^8 + \alpha \cdot \alpha^2 = \alpha^8 + \alpha^3 = \alpha^{13};$$

$d_3 = 0$, $\alpha^{13} = 0$? – ні → модифікація многочлена зворотного зв'язку;

$\sigma^*(x) = \sigma(x) + d_3 \cdot x \cdot B(x) = 1 + \alpha \cdot x + \alpha^{13} \cdot x \cdot \alpha^{14} x = 1 + \alpha \cdot x + \alpha^{27} x^2 = 1 + \alpha \cdot x + \alpha^{12} x^2$;

$2L \le k-1$, $2 \le 2$? – так → довжина регістру збільшується;

$B(x) = d_3^{-1} \cdot \sigma(x) = \alpha^2 \cdot (1 + \alpha x) = \alpha^2 + \alpha^3 x$; {для $d_3 = \alpha^{13}$ зворотній елемент $d_3^{-1} = \alpha^2$}

$L = k - L = 3 - 1 = 2$;

$\sigma(x) = \sigma^*(x) = 1 + \alpha \cdot x + \alpha^{12} x^2$;

$k = 2t$, 3=6 ? – ні → збільшуємо шаг ітерації;

$k = k+1 = 3+1 = 4$.

Крок 4. $k = 4$.

$$d_4 = S_4 + \sum_{j=1}^{L=2} \sigma_{j-1} S_{4-j} = S_4 + \sigma_1 S_3 + \sigma_2 S_2 = \alpha^4 + \alpha \cdot \alpha^8 + \alpha^{12} \cdot \alpha^2 =;$$

$$= \alpha^4 + \alpha^9 + \alpha^{14} = \alpha^{14} + \alpha^{14} = 0$$

$d_4 = 0$, 0 = 0 ? – да → без модифікації многочлена зворотного зв'язку;

$B(x) = x \cdot B(x) = x\,(\alpha^2+\alpha^3 x)=\alpha^2 x+\alpha^3 x^2$;

$k = 2t$, 4 = 6 ? – ні → збільшуємо шаг ітерації;

$k = k+1 = 4+1 = 5$.

Крок 5. $k = 5$.

$$d_5 = S_5 + \sum_{j=1}^{L=2} \sigma_{j-1} S_{5-j} = S_5 + \sigma_1 S_4 + \sigma_2 S_3 = 1 + \alpha \cdot \alpha^4 + \alpha^{12} \cdot \alpha^8 =$$

$$= 1 + \alpha^5 + \alpha^{20} = 1 + \alpha^5 + \alpha^5 = 1 + 0 = 1;$$

$d_5 = 0$, 1 = 0 ? – нет → модифікація многочлена зворотного зв'язку;

$\sigma^*(x) = \sigma(x)+d_5 \cdot x \cdot B(x) = (1+\alpha x+\alpha^{12}x^2)+1 \cdot x \cdot (\alpha^2 x+\alpha^3 x^2) = 1+\alpha x+\alpha^{12}x^2 +$

$+\alpha^2 x^2+\alpha^3 x^3 = 1+\alpha x+\alpha^7 x^2+\alpha^3 x^3$;

$2L \le k-1$, $4 \le 4$? – да → довжина регістру збільшується;

$B(x) = d_5^{-1} \cdot \sigma(x) = 1 \cdot (1+\alpha \cdot x+\alpha^{12}x^2) = 1+\alpha \cdot x+\alpha^{12}x^2$ {для $d_5 = \alpha^0$ зворотній елемент $d_5^{-1}=1$};

$L = k-L = 5-2 = 3$;

$\sigma(x) = \sigma^*(x) = 1+\alpha x+\alpha^7 x^2+\alpha^3 x^3$;

$k = 2t$, 5 = 6 ? – ні → збільшуємо шаг ітерації;

$k = k+1 = 5+1 = 6$.

Крок 6. $k = 6$.

$$d_6 = S_5 + \sum_{j=1}^{L=3} \sigma_{j-1} S_{5-j} = S_6 + \sigma_1 S_5 + \sigma_2 S_4 + \sigma_3 S_3 = \alpha + \alpha \cdot 1 + \alpha^7 \cdot \alpha^4 + \alpha^3 \cdot \alpha^8 =$$

$$= \alpha + \alpha + \alpha^{11} + \alpha^{11} = 0 + 0 = 0;$$

$d_6 = 0$, 0 = 0 ? – так → без модифікації многочлена зворотного зв'язку;

$B(x) = x \cdot B(x) = x \cdot (1+\alpha \cdot x+\alpha^{12}x^2) = x+\alpha x^2+\alpha^{12}x^3$;

$k = 2t$, 6 = 6 ? – так → остання ітерація, кінець.

У табл. 6.10 наведені необхідні розрахунки для знаходження многочлена локаторів помилок за допомогою алгоритм Берлекемпа-Мессі

Таблиця 6.10 – Обчислення полінома локаторів помилок за допомогою алгоритму Берлекемпа-Мессі

k	d_k	$\sigma^*(x)$	$B(x)$	$\sigma(x)$	L
0	–	–	1	1	0
1	α	$1+\alpha\cdot x$	α^{14}	$1+\alpha\cdot x$	1
2	0	$1+\alpha\cdot x$	$\alpha^{14}x$	$1+\alpha\cdot x$	1
3	α^{13}	$1+\alpha\cdot x+\alpha^{12}x^2$	$\alpha^2+\alpha^3 x$	$1+\alpha\cdot x+\alpha^{12}x^2$	2
4	0	$1+\alpha\cdot x+\alpha^{12}x^2$	$\alpha^2 x+\alpha^3 x^2$	$1+\alpha\cdot x+\alpha^{12}x^2$	2
5	1	$1+\alpha x+\alpha^7 x^2+\alpha^3 x^3$	$1+\alpha\cdot x+\alpha^{12}x^2$	$1+\alpha x+\alpha^7 x^2+\alpha^3 x^3$	3
6	0	$1+\alpha x+\alpha^7 x^2+\alpha^3 x^3$	$x+\alpha x^2+\alpha^{12}x^3$	$1+\alpha x+\alpha^7 x^2+\alpha^3 x^3$	3

Таким чином, поліном локаторів помилок дорівнює

$$\sigma(x) = 1+\alpha x+\alpha^7 x^2+\alpha^3 x^3.$$

Ступінь полінома локаторів помилок $\deg[\sigma(x)] = 3$ – відповідно у прийнятої послідовності $f(x)$ три помилки. Для пошуку позицій цих помилок необхідно знайти корені поліному локаторів помилок методом Ченя .

3. Пошук коренів поліному локаторів помилок методом Ченя – перевірка умови $\sigma(\alpha^j) = 0$ для всіх ненульових елементів розширеного поля $GF(2^4)$

$\sigma(\alpha^0) = 1+\alpha+\alpha^7+\alpha^3=\alpha^3+\alpha^3=0;$

$\sigma(\alpha^1) = 1+\alpha\cdot\alpha+\alpha^7\alpha^2+\alpha^3\alpha^3=1+\alpha^2+\alpha^9+\alpha^6=\alpha^4;$

$\sigma(\alpha^2) = 1+\alpha^1\alpha^2+\alpha^7\alpha^4+\alpha^3\alpha^6=1+\alpha^3+\alpha^{11}+\alpha^9 = \alpha^{14}+\alpha^2=\alpha^{13};$

$\sigma(\alpha^3) = 1+\alpha^1\alpha^3+\alpha^7\alpha^6+\alpha^3\alpha^9=1+\alpha^4+\alpha^{13}+\alpha^{12}=\alpha^1+\alpha^1=0;$

$\sigma(\alpha^4) = 1+\alpha^1\alpha^4+\alpha^7\alpha^8+\alpha^3\alpha^{12}=1+\alpha^5+\alpha^{15}+\alpha^{15}=\alpha^{10}+0=\alpha^{10};$

$\sigma(\alpha^5) = 1+\alpha^1\alpha^5+\alpha^7\alpha^{10}+\alpha^3\alpha^{15}=1+\alpha^6+\alpha^2+\alpha^3=\alpha^{13}+\alpha^6=1;$

$\sigma(\alpha^6) = 1+\alpha^1\alpha^6+\alpha^7\alpha^{12}+\alpha^3\alpha^{18}=1+\alpha^7+\alpha^4+\alpha^6=\alpha^9+\alpha^{12}=\alpha^8;$

$\sigma(\alpha^7) = 1+\alpha^1\alpha^7+\alpha^7\alpha^{14}+\alpha^3\alpha^{21}=1+\alpha^8+\alpha^{11}+\alpha^9=\alpha^{14}+\alpha^2=\alpha^{13};$

$\sigma(\alpha^8) = 1+\alpha^1\alpha^8+\alpha^7\alpha^{16}+\alpha^3\alpha^{24}=1+\alpha^9+\alpha^8+\alpha^{12}=\alpha^7+\alpha^9=1;$

$\sigma(\alpha^9) = 1+\alpha^1\alpha^9+\alpha^7\alpha^{18}+\alpha^3\alpha^{27}=1+\alpha^{10}+\alpha^{10}+1=\alpha^5+\alpha^5=0;$

$\sigma(\alpha^{10}) = 1+\alpha^1\alpha^{10}+\alpha^7\alpha^{20}+\alpha^3\alpha^{30}=1+\alpha^{11}+\alpha^{12}+\alpha^3=\alpha^{12}+\alpha^{10}=\alpha^3;$

$\sigma(\alpha^{11}) = 1+\alpha^1\alpha^{11}+\alpha^7\alpha^{22}+\alpha^3\alpha^{33}=1+\alpha^{12}+\alpha^{14}+\alpha^6=\alpha^{11}+\alpha^8=\alpha^7;$

$\sigma(\alpha^{12}) = 1+\alpha^1\alpha^{12}+\alpha^7\alpha^{24}+\alpha^3\alpha^{36}=1+\alpha^{13}+\alpha^1+\alpha^9=\alpha^6+\alpha^3=\alpha^2;$

$\sigma(\alpha^{13}) = 1+\alpha^1\alpha^{13}+\alpha^7\alpha^{26}+\alpha^3\alpha^{39}=1+\alpha^{14}+\alpha^3+\alpha^{12}=\alpha^3+\alpha^{10}=\alpha^{12};$

$\sigma(\alpha^{14}) = 1+\alpha^{1}\alpha^{14}+\alpha^{7}\alpha^{28}+\alpha^{3}\alpha^{42}=1+\alpha^{15}+\alpha^{5}+\alpha^{0}=0+\alpha^{10}=\alpha^{10}$

Корені поліному $\sigma(x)$ – $\boldsymbol{\alpha^{0}, \alpha^{3}, \alpha^{9}}$.

4. Знайдемо обернені елементи для корнів поліному $\sigma(x)$.

− для елемента α^{3} оберненим є елемент $\boldsymbol{\alpha^{12}}$, оскільки з табл. Г.2 (додаток Г) знаходимо $\alpha^{3} \cdot \alpha^{12} = 1$;

− для елемента α^{9} оберненим є елемент $\boldsymbol{\alpha^{6}}$, оскільки з табл. Г.2 (додаток Г) знаходимо $\alpha^{9} \cdot \alpha^{6} = 1$;

− для елемента α^{0} оберненим є елемент $\boldsymbol{\alpha^{0}}$, оскільки з табл. Г.2 (додаток Г) знаходимо $\alpha^{0} \cdot \alpha^{0} = 1$.

Обернені елементи – $\boldsymbol{\alpha^{0}}$, $\boldsymbol{\alpha^{12}}$, $\boldsymbol{\alpha^{6}}$.

5. Розрахуємо вектор помилок та виправимо $f(x)$.

Знайдемо позиції помилок як ступені обернених елементів – $j_1 = 12$, $j_2 = 6$, $j_3 = 0$. Таким чином вектор помилок дорівнює – $e(x) = x^{12}+x^{6}+1$.

Для виправлення помилок в кодовій послідовності $f(x)$ виконаємо додавання

$v(x) = f(x)+e(x)=x^{14}+x^{11}+x^{8}+x^{6}+x^{4}+x^{3}+x^{2}+x+1+x^{12}+x^{6}+1=x^{14}+x^{12}+$
$+x^{11}+x^{8}+x^{4}+x^{3}+x^{2}+x.$

6.6 Кодування кодів Ріда-Соломона

Особливе місце серед БЧХ кодів займають коди Ріда-Соломона РС коди), відкриті ще до винаходу БЧХ кодів. Ці коди мають унікальні властивості, які будуть розглядатися нижче. Вони лежать в основі багатьох кодових конструкцій (каскадних кодів) і використовуються у багатьох прикладних задачах (контролерах магнітних дисків, оптичних дисків, накопичувачів на магнітних стрічках). Також коди РС використовують у багатьох системах передачі. Наприклад, знаменитий РС код (255, 223, 33) використовується в системах космічного зв'язку НАСА. На аудіо компакт-дисках стандарту CD цифрова інформація записується за допомогою комбінації укорочених (32, 28) та (28, 24) кодів РС. Стандарт цифрового телевізійного мовлення DVB-T (Digital Video Broadcasting – Terrestrial) включає (204, 188) код РС, а його подальша модифікація DVB-H використовує (255, 191) код РС.

Коди РС визначаються над полем $GF(p^{m})$ як безліч кодових слів, компоненти яких дорівнюють значенням деяких певних многочленів. Далі розглядаються коди над кінцевими полями характеристики $p^{m}=2^{m}$ як

найпоширеніші. Таким чином, символи кодових слів є елементами поля $GF(2^m)$, а довжина коду дорівнює $n = 2^m-1$. Будь-яке інформаційне повідомлення може бути розглянуте як послідовність блоків (символів) по m біт. Тоді позначимо таке повідомлення у вигляді інформаційного полінома ступеня $k-1$

$$u(x) = u_0 + u_1 x + u_2 x^2 + \quad ... \quad + u_{k-1} x^{k-1} \tag{6.36}$$

з коефіцієнтами u_i, що належать полю $GF(2^m)$, $i = 0, 1, \ldots, k-1$. Всього існує 2^{mk} таких многочленів.

Нехай, наприклад, інформаційна послідовність складається з 12 біт та дорівнює u = 101011100110. Для подання в поліноміальній формі $u(x)$ у розширеному полі $GF(2^3)$ необхідно розбити інформаційну послідовність на блоки по m=3 біта і кожному блоку поставити відповідно до табл. 1.5 елемент поля

$101 = \alpha^6$, $011 = \alpha^3$, $100 = \alpha^2$, $110 = \alpha^4$.

Таким чином, інформаційна послідовність є поліномом виду

$u(x) = \alpha^6 x^3 + \alpha^3 x^2 + \alpha^2 x^1 + \alpha^4$.

Для кодів РС справедливо, що символи кодових комбінацій (коефіцієнти кодових многочленів) і корені цих многочленів (нулі коду) – це елементи одного поля $GF(2^m)$. Обчислюючи значення многочлена (6.36) для ненульових елементів поля $GF(2^m)$, отримуємо кодове слово $v(x)$ коду РС з параметрами $(2^m-1, k, d)$

$$v = (u(\alpha^0), u(\alpha^1), u(\alpha^2), ..., u(\alpha^{2^m-2})). \tag{6.37}$$

Нулями РС коду є $2t$ послідовних ступенів примітивного елемента поля Галуа α^b, α^{b+1}, ..., α^{b+2t-1}. Якщо при побудові породжуючого полінома $g(x)$ ігнорувати пов'язані за ступенем два цикли елементів $\alpha, \alpha^2, \ldots, \alpha^{2t}$ і складати його як добуток біномів виду $x + \alpha^i$, то поліном виявиться недвійковим. Внаслідок цього код, який він породжує, також виявиться недвійковим, із символами, що належать розширеному полю $GF(2^m)$. Оскільки над полем $GF(2^m)$ мінімальний багаточлен дорівнює $\varphi_i(x) = (x + \alpha^i)$ і поле символів збігається з полем локаторів помилок, то всі мінімальні многочлени і всі дільники породжуючого многочлена є

лінійними (тобто мають ступінь 1). Таким чином, породжуючий поліном коду РС визначається

$$g(x) = \prod_{i=b}^{b+2t-1} (x + \alpha^i), \tag{6.38}$$

де b – будь-яке ціле число, яке набуває значення ступеня першого кореня.

Як правило, значення b приймають рівним 0 або 1. Наприклад, при b= 1 породжуючий поліном записується у вигляді

$$g(x) = (x + \alpha^1)(x + \alpha^2)\ldots(x + \alpha^{2t}). \tag{6.39}$$

Приклад 6.22. Побудувати РС код із мінімальною кодовою відстанню d =5 над розширеним полем $GF(2^3)$, що породжується незвідним многочленом $p(x) = x^3+x+1$.

Рішення. Для цього коду, здатного виправляти дворазові помилки, довжина дорівнює $n = 2^3-1 = 7$. Оскільки код здатний виправляти $t = 2$ помилки, то для побудови породжуючого полінома необхідно використовувати $2t$ =4 послідовних елементів розширеного поля $GF(2^3)$. Для ступеня першого кореня b = 1 і для елементів α^1, α^2, α^3, α^4 поля $GF(2^3)$ знайдемо породжуючий поліном використовуючи (6.39)

$$\begin{aligned}
&g(x) = (x+\alpha^1)(x+\alpha^2)(x+\alpha^3)(x+\alpha^4) = (x^3 + x\alpha^2 + x\alpha + \alpha^3)(x+\alpha^3)(x+\alpha^4) = \\
&= (x^3 + x^2\alpha^2 + x^2\alpha + x\alpha^3 + x^2\alpha^3 + x\alpha^5 + x\alpha^4 + \alpha^6)(x+\alpha^4) = x^4 + x^3\alpha^2 + x^3\alpha + x^2\alpha^3 + \\
&+ x^3\alpha^3 + x^2\alpha^5 + x^2\alpha^4 + x\alpha^6 + x^3\alpha^4 + x^2\alpha^6 + x^2\alpha^5 + x\alpha^7 + x^2\alpha^7 + x\alpha^9 + x\alpha^8 + \alpha^{10} = \\
&= x^4 + (\alpha^2 + \alpha + \alpha^3 + \alpha^4)x^3 + (\alpha^3 + \alpha^5 + \alpha^4 + \alpha^6 + \alpha^5 + \alpha^7)x^2 + (\alpha^6 + \alpha^7 + \alpha^9 + \alpha^8)x + \alpha^{10}.
\end{aligned}$$

Використовуючи таблицю складання елементів розширеного поля $GF(2^3)$ (6.6), а також з урахуванням, що $\alpha^7 = 1$, $\alpha^9 = \alpha^2$, $\alpha^8 = \alpha^1$, $\alpha^{10} = \alpha^3$ отримаємо

$$g(x) = x^4 + (\alpha^4 + \alpha^6)x^3 + (\alpha^5 + \alpha^4)x^2 + (\alpha^2 + \alpha^4)x + \alpha^3 = x^4 + \alpha^3x^3 + x^2 + \alpha x + \alpha^3.$$

Таким чином, знайдено код РС з параметрами (7, 3, 5), для якого інформаційний многочлен є послідовністю трьох вісімкових символів (що еквівалентно 9 бітам). При відображенні символів $GF(2^3)$ вектора довжини 3 отримуємо двійковий (21, 9, 5) код, здатний виправляти до двох випадкових помилок і будь-який пакет до 4-х помилок.

Якщо вибрати значення ступеня першого кореня b=0, тоді для формування породжуючого полінома використовуємо корені α^0, α^1, α^2, α^3 і отримаємо

$$g(x) = (x + \alpha^0)(x + \alpha^1)(x + \alpha^2)(x + \alpha^3) = x^4 + \alpha^2 x^3 + \alpha^5 x^2 + \alpha^5 x + \alpha^6 . \blacksquare$$

Кодування РС кодів

Для кодування інформаційної послідовності $u(x)$ РС кодом з параметрами (n , k , d) використовуються алгоритми кодування циклічних кодів.

Тоді з урахуванням вищесказаного можна представити алгоритм кодування інформаційних повідомлень з використанням кодів РС при заданій довжині повідомлення k і кратності помилки, що виправляється t.

1. Задати основні параметри коду.

- довжина коду $n=2^m-1$ (m – ціле число, яке використовується при побудові розширеного поля $GF(2^m)$);
- кількість помилок t , що виправляються кодом Ріда-Соломона;
- ступінь b першого елемента α^b розширеного поля $GF(2^m)$, який використовується при побудові породжує полінома;
- неприводимый багаточлен $p(x)$ для побудови розширеного поля $GF(2^m)$.

2. Побудувати розширене поле $GF(2^m)$, що породжується багаточленом $p(x)$, що не наводиться.

3. Задати нулі коду α^i – елементів розширеного поля $GF(2^m)$ α^b, α^{b+1}, ..., α^{b+2t-1} для побудови полінома, що породжує.

4. Формування полінома, що породжує $g(x) = \prod_{i=b}^{b+2t-1} (x + \alpha^i)$.

5. Знаходження кількості перевірочних символів r = deg[$g(x)$].

6. Знаходження кількості інформаційних символів $k = n - r$.

7. Завдання інформаційної послідовності u (x) довжини k.

8. Кодування інформаційної послідовності u (x) побудованим РС кодом:

− несистематичний РС код $v(x) = u(x)g(x)$;

− систематичний РС код $v(x) = u(x)x^r + R(x)$, R (x) = u (x)· x^r mod $g(x)$.

При передачі кодового слова по каналу зв'язку або при зберіганні даних на носії інформація може бути спотворена в силу дії шумів у каналі передачі даних або пошкодження носія даних. У такому разі можна говорити, що на кодову послідовність буде накладено деяке спотворення $e(x)$ і поліном $v(x)$ складатиметься разом з деяким поліномом спотворень $e(x)$, і в результаті матимемо поліном спотвореного кадру $f(x) = v(x) + e(x)$. У цьому можуть спотворюватися будь-які коефіцієнти кодового полінома, як інформаційні, і надлишкові.

Приклад 6.23. Побудувати код Ріда − Соломона над розширеним полем $GF(2^3)$, що породжується многочленом $p(x) = x^3+x+1$. Код РС має виправляти двократні помилки $t = 2$. Довжина коду $n = 2^3 - 1 = 7$. Вибрати значення ступеня першого кореня $b=4$. Закодувати несистематичним РС кодом інформаційну послідовність, що складається з трьох вісімкових символів $u(x)=\alpha^3x^2 + \alpha x + \alpha^5$.

Рішення.

1. *Побудувати породжуючий поліном коду РС.*

Для побудови *породжуючого* полінома заданого коду необхідно використовувати $2t = 4$ послідовних елементів розширеного поля $GF(2^3)$: α^4, α^5, α^6, α^7. З урахуванням елементів поля $GF(2^3)$ та таблиці складання елементів поля (табл. 6.6) побудуємо поліном, що породжує

$$g(x) = (x+\alpha^4)(x+\alpha^5)(x+\alpha^6)(x+1) = x^4 + (\alpha^2+1)x^3 + (\alpha^2+1)x^2 +$$
$$+ (\alpha+1)x + \alpha = x^4 + \alpha^6x^3 + \alpha^6x^2 + \alpha^3x + \alpha.$$

Таким чином, побудований (7, 3, 5) код РС з багаточленом, що породжує $g(x) = x^4 + \alpha^6 x^3 + \alpha^6 x^2 + \alpha^3 x + \alpha$.

2. *Кодування інформаційної послідовності $u(x)$ кодом РС.*

Закодуємо інформаційну послідовність $u(x)$ отриманим кодом

$$v(x) = u(x)\cdot g(x) = (\alpha^3x^2 + \alpha\tilde{o} + \alpha^5)(x^4+\alpha^6x^3 + \alpha^6x^2 + \alpha^3x + \alpha) =$$
$$= \alpha^3x^6 + \alpha^9x^5 + \alpha^9x^4 + \alpha^6x^3 + \alpha^4x^2 + \alpha x^5 + \alpha^7x^4 + \alpha^7x^3 + \alpha^4x^2 + \alpha^2x + \alpha^5x^4 + \alpha^{11}x^3 +$$
$$+ \alpha^{11}x^2 + \alpha^8x + \alpha^6 = \alpha^3x^6 + (\alpha^9+\alpha)x^5 + (\alpha^9+\alpha^7+\alpha^5)x^4 + (\alpha^6+\alpha^7+\alpha^{11})x^3 +$$
$$+ (\alpha^4+\alpha^4+\alpha^{11})x^2 + (\alpha^2+\alpha^8)x + \alpha^6 = \alpha^3x^6 + (\alpha^2+\alpha)x^5 + (\alpha^2+1+\alpha^5)x^4 +$$
$$(\alpha^6+1+\alpha^4)x^3 + (0+\alpha^4)x^2 + (\alpha^2+\alpha)x + \alpha^6 =$$
$$= \alpha^3x^6 + \alpha^4x^5 + \alpha x^4 + \alpha x^3 + \alpha^4x^2 + \alpha^4x + \alpha^6.$$

Таким чином, кодова послідовність

$v(x) = \alpha^3x^6 + \alpha^4x^5 + \alpha x^4 + \alpha x^3 + \alpha^4x^2 + \alpha^4x + \alpha^6$. ■

6.7 Декодування кодів Ріда-Соломона

Нехай прийнятий кодовий поліном $f(x)$, що складається з *інформаційних* і надлишкових символів, прийнятий по мережі зв'язку або зчитаний з носія: $f(x) = v(x) + e(x)$.

Тоді *синдром помилки* являє собою сукупність компонентів S_j, $j = 1, \ldots, 2t$, обчислюваних шляхом підстановки в поліном $f(x)$ коренів α^b, α^{b+1}, ..., α^{b+2t-1} породжуючого полінома $g(x)$. Зважаючи на те, що, як ми раніше встановили, поліном $f(x)$ неспотвореної послідовності обертається в нуль при підстановці в нього коренів α^b, α^{b+1}, ... , α^{b+2t-1}, тоді справедлива рівність

$$S_j = f(\alpha^i) = e(\alpha^i), \tag{6.40}$$

де j – номер компоненти синдрому ($j = 1, 2, \ldots, 2t$) ; i – ступінь кореня елемента поля ($i = b, b+1, \ldots, b + 2t - 1$).

Таким чином, синдром не залежить від самої вихідної неспотвореної послідовності $v(x)$ і повністю характеризується лише помилкою $e(x)$. Саме на цьому й базується вся технологія розшифровки синдрому з метою знаходження локаторів помилок та їх величин.

Для подальшого розгляду процедури декодування кодів РС необхідно запровадити поняття *полінома синдрому помилки* $S(x)$, який може бути представлений як

$$S(x) = S_1 + S_2 x + \quad \ldots \quad + S_{2t} x^{2t-1}. \tag{6.41}$$

Очевидно, що якщо всі компоненти синдрому дорівнюють нулю, то можна говорити, що спотворень не було, в іншому випадку, якщо синдром ненульовий, то можна декодувати і виправити помилки, що відбулися в кодовій послідовності.

Алгоритми декодування кодів РС близькі до алгоритмів для декодування двійкових кодів БЧХ. Як було зазначено раніше, у процесі декодування двійкових БЧХ кодів обчислюється багаточлен локаторів помилок

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \ldots + \sigma_v x^v, \tag{6.42}$$

де v – число помилок у прийнятому слові ($0 \le v \le t$).

За допомогою коренів α^k полінома локаторів помилок $\sigma(x)$ знаходяться локатори позицій помилок $\alpha^{-j}=\alpha^k$ і відповідно позиції помилок $j_1, j_2, \ldots j_v$ (ступеня локаторів помилок) для многочлена помилок

$$e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \ldots + e_{j_v} x^{j_v} . \qquad (6.43)$$

Оскільки для двійкових БЧХ кодів коефіцієнтами (значеннями помилок $e_{j_1}, e_{j_2}, \ldots, e_{j_v}$) полінома $e(x)$ є біти 0 або 1, то для виправлення помилок необхідно інвертувати спотворені біти на знайдених позиціях.

Однак для декодування кодів РС, які мають справу не з бітами, а символами, необхідно знайти не тільки позиції помилок $j_1, j_2, \ldots j_v$ але і значення помилок $e_{j_1}, e_{j_2}, \ldots, e_{j_v}$. При цьому декодер невідома кількість помилок v, які відбулися при передачі кодової послідовності.

Для знаходження значень помилок e_{j_l} введемо поняття полінома значень помилок, зв'язавши його з синдромним поліномом $S(x)$ і поліномом локаторів помилок $\sigma(x)$ наступним чином

$$\Omega(x) = (S(x) \cdot \sigma(x)) \bmod x^{2t}. \qquad (6.44)$$

У цьому випадку поліном локаторів помилок має ступінь $\deg[\sigma(x)] = v$, а поліном значень помилок має ступінь $\deg[\Omega(x)] = v - 1$.

Знаходження значень помилок здійснюється за допомогою *алгоритму Форні*. Нижче наведено вираз методу Форні, що використовується для кодів РС, який дозволяє знайти значення помилки e_{j_l} на позиціях j_l $(1 \le l \le v,\ b \le j \le b+2t-1)$

$$e_{j_l} = (\alpha^{j_l})^{1-b} \frac{\Omega(\alpha^{-j_l})}{\sigma'(\alpha^{-j_l})}, \qquad (6.45)$$

де $\sigma'(x)$ – формальна похідна по x багаточлена локаторів помилок $\sigma(x)$, $1 \le l \le v$; $b \le j \le b+2t-1$, b – ступінь першого кореня; α^{-j} – корінь $\sigma(x)$; α^j – локатор помилки.

На етапі обчислення помилок поліном локаторів помилок $\sigma(x)$ вже знайдено, тому формальну похідну можна обчислити

$$\sigma'(x) = \sum_{j=1}^{v} (j\sigma_j) x^{j-1}. \qquad (6.46)$$

В алгоритмі декодування кодів РС для вирішення ключового рівняння (6.27)
та знаходження коефіцієнтів $\sigma_1, \sigma_2, \ldots, \sigma_v$ полінома локаторів помилок $\sigma(x)$ можуть використовуватися вже вказані раніше методи:

− алгоритм Пітерсона − Горенстейна − Цирлера (PGZ);

− алгоритм Берлекемпа − Мессі (ВМА);

− алгоритм Евкліда (ЕА).

Далі для декодування кодів Ріда-Соломона буде використано *алгоритм Евкліда.* В алгоритмі декодування кодів Ріда-Соломона для вирішення ключового рівняння та знаходження полінома локаторів помилок з використанням алгоритму Евкліда обчислюється не тільки поліном локаторів помилок $\sigma(x)$, а й одночасно поліном значень помилок $\Omega(x)$.

В основі алгоритму Евкліда лежить добре відома процедура знаходження найбільшого спільного дільника (НСД) двох поліномів (або цілих чисел).

Алгоритм Евкліда є рекурентним методом знаходження НСД двох чисел або двох багаточленів. В основі цього алгоритму лежить таке твердження. Нехай a та b − два цілих числа або багаточлена, причому $a \geq b$ або $\deg[(a)] \geq \deg[(b)]$. Розділимо a на b. Якщо залишок від поділу r дорівнює 0, то $d = b$ є найбільшим спільним дільником. Якщо залишок не дорівнює 0, замінимо a на b також замінимо b на r і повторимо поділ.

Наприклад, необхідно знайти НСД цілих чисел a=186 і b=66. Дотримуючись алгоритму Евкліда , отримуємо :

Крок 1. $r = 186 \bmod 66 = 54$, $r \neq 0 \rightarrow a = 66$, $b = 54$.

Крок 2. $r = 66 \bmod 54 = 12$, $r \neq 0 \rightarrow a = 54$, $b = 12$.

Крок 3. $r = 54 \bmod 12 = 6$, $r \neq 0 \rightarrow a = 12$, $b = 6$.

Крок 4. $r = 12 \bmod 6 = 0$, $r = 0$.

Таким чином, НСД (186, 54) = 6.

У процесі знаходження НСД, згідно з алгоритмом Евкліда , обчислюються два числа f і g (або багаточлена $f(x)$ і $g(x)$), для яких $fa+gb = d$. На кожному кроці алгоритму обчислюються проміжні числа (багаточлени) f_i і g_i такі , що $f_i\, a + g_i\, b = r_i$. При цьому f_i та g_i виходять з урахуванням двох попередніх значень за формулою

$$f_i = g_i = E_i = E_{i-2} - q_i \cdot E_{i-1}, \tag{6.47}$$

де q_i – відношення двох попередніх залишків

$$q_i = \left[\frac{r_{i-2}}{r_{i-1}} \right]. \tag{6.48}$$

Знак [ ] означає цілу частину дробу.

Використовуючи рекурентні співвідношення між f_i, g_i і r_i , виконаємо наступні перетворення в алгоритмі знаходження НСД двох чисел a = 186 та b=66.

Крок 1. Задаємо початкові установки: $\boldsymbol{i = -1}$, $f_{-1} = 1$, $g_{-1} = 0$.

Наводимо рівняння до необхідної форми

$f_{-1}a + g_{-1}b = r_{-1} \rightarrow r_{-1} = 1\cdot 186 + 0\cdot 66 = \underline{186}$.

Крок 2. Задаємо початкові установки: $\boldsymbol{i = 0,}$ $f_0 = 0$, $g_0 = 1$.

Наводимо рівняння до необхідної форми

$f_0 a + g_0 b = r_0 \rightarrow r_0 = 0\cdot 186 + 1\cdot 66 = \underline{66.}$

Крок 3 $\boldsymbol{i = 1}$.

$f_1 = E_{-1} - q_1 \cdot E_0 = f_{-1} - q_1 \cdot f_0$, $g_1 = E_{-1} - q_1 \cdot E_0 = g_{-1} - q_1 \cdot g_0$,

$q_1 = [r_{-1}/r_0] = [186/66] = 2$,

$f_1 = 1 - 2\cdot 0 = 1$, $g_1 = 0 - 2\cdot 1 = -2$,

$f_1 a + g_1 b = r_1 \rightarrow r_1 = 1\cdot 186 - 2\cdot 66 = \underline{54}$.

Крок 4. $\boldsymbol{i = 2}$.

$f_2 = f_0 - q_2 \cdot f_1$, $g_2 = g_0 - q_2 \cdot g_1$,

$q_2 = [r_0/r_1] = [66/54] = 1$,

$f_2 = 0 - 1\cdot 1 = -1$, $g_2 = 1 - 1\cdot(-2) = 3$,

$f_2 a + g_2 b = r_2 \rightarrow r_2 = -1\cdot 186 + 3\cdot 66 = \underline{12}$.

Крок 5. $\boldsymbol{i = 3}$.

$f_3 = f_1 - q_3 \cdot f_2$, $g_3 = g_1 - q_3 \cdot g_2$,

$q_3 = [r_1/r_2] = [54/12] = 4,$

$f_3 = 1-4\cdot(-1) = 5, \quad g_2 = -2-4\cdot 3 = -14,$

$f_3 a + g_3 b = r_3 \rightarrow r_3 = 5\cdot 186 + (-14)\cdot 66 = \underline{6}.$

Крок 6. $\boldsymbol{i = 4}$.

$f_4 = f_2 - q_4 \cdot f_3, \quad g_4 = g_2 - q_4 \cdot g_3,$

$q_4 = [r_2/r_3] = [12/6] = 2,$

$f_4 = -1-2\cdot 5 = -11, \quad g_2 = 3-2\cdot(-14) = 31,$

$f_4 a + g_4 b = r_4 \rightarrow r_4 = (-11)\cdot 186 + 31\cdot 66 = \underline{0}.$

Процес знаходження НСД (186, 66) можна також подати у табличному вигляді (табл. 6.11).

Таблиця 6.11 – Процес знаходження НСД (186, 66)

Крок i	f_i	g_i	$d_i\,(r_i)$	q_i	$r_i = f_i \cdot a + g_i \cdot b$
−1	1	0	186	-	1·186+0·66=186
0	0	1	66	-	0·186+1·66=66
1	1	−2	54	[186/66] = 2	1·186−2·66=54
2	−1	3	12	[66/54] = 1	− 1·186+3·66=12
3	5	−14	6	[54/12] = 4	5·186−14·66=6
4	−11	31	0	[12/6] = 2	− 11·186+31·66=0

Виконані перетворення використовуються при декодуванні кодів РС для вирішення ключового рівняння алгоритмом Евкліда. Оскільки декодування РС кодів здійснюється для многочленів, повторимо вищевикладені міркування стосовно многочленів.

Якщо існує найбільший спільний дільник $d(x)$ двох багаточленів $a(x)$ і $b(x)$, то існують багаточлени $f(x)$ і $g(x)$, такі, що справедливо $a(x)\cdot f(x) + b(x)\cdot g(x) = d(x)$.

Наприклад, необхідно знайти найбільший спільний дільник $d(x)$ двох багаточленів $a(x) = x^3 + 1$ та $b(x) = x^2 + 1$, а також многочлени $f(x)$ та $g(x)$, для яких виконується $a(x)\cdot f(x) + b(x)\cdot g(x) = d(x)$ у двійковому полі.

Крок 1. Задаємо початкові установки: $\boldsymbol{i = -1}$, $f_{-1}(x) = 1$, $g_{-1}(x) = 0$.

Наводимо рівняння до необхідної форми

$f_{-1}(x)a(x) + g_{-1}(x)b(x) = r_{-1}(x) \rightarrow r_{-1}(x) = 1\cdot(x^3+1) + 0\cdot(x^2+1) = \underline{x^{3+1}}.$

Крок 2. Задаємо початкові установки: $\boldsymbol{i = 0}$, $f_0(x) = 0$, $g_0(x)=1$.

Наводимо рівняння до необхідної форми

$f_0(x)a(x) + g_0(x)b(x) = r_0(x) \rightarrow r_0(x) = 0\cdot(x^3+1) + 1\cdot(x^2+1) = \underline{x^2+1.}$

Крок 3. $\boldsymbol{i = 1}$.

$f_1(x) = E_{-1}(x) - q_1(x)\cdot E_0(x) = f_{-1}(x) - q_1(x)\cdot f_0(x)$,

$g_1(x) = E_{-1}(x) - q_1(x)\cdot E_0(x) = g_{-1}(x) - q_1(x)\cdot g_0(x)$,

$q_1(x) = [r_{-1}(x)/r_0(x)] = [x^3+1/x^2+1] = \underline{x}$,

$f_1(x) = 1 + x\ 0 = 1$, $g_1(x) = 0 + x\cdot 1 = x$,

$f_1(x)a(x) + g_1(x)b(x) = r_1(x) \rightarrow r_1(x) = 1\cdot(x^3+1) + x\cdot(x^2+1) = x+1$.

Крок 4. $\boldsymbol{i = 2}$.

$f_2(x) = E_0(x) - q_2(x)\cdot E_1(x) = f_0(x) - q_2(x)\cdot f_1(x)$,

$g_2(x) = E_0(x) - q_2(x)\cdot E_1(x) = g_0(x) - q_2(x)\cdot g_1(x)$,

$q_2(x) = [r_0(x)/r_1(x)] = [x^2+1/x+1] = \underline{x+1}$,

$f_2(x) = 0 + (x+1)\ 1 = x+1$, $g_2(x) = 1 + (x+1)\ x = x^2+x+1$,

$f_1(x)a(x) + g_1(x)b(x) = r_1(x) \rightarrow r_1 = (x+1)\cdot(x^3+1) + (x^2+x+1)\cdot(x^2+1) = 0$.

Шукане значення НСД $d(x)$ визначається останнім ненульовим значенням $r_i(x)$, тобто $d(x)=r_1(x)=x+1$. Процес знаходження НСД (x^{3+1}, x^{2+1}) можна також подати у табличному вигляді (табл. 6.12).

Таким чином, НСД (x^{3+1}, x^{2+1}) = $x+1$.

Таблиця 6.12 – Процес знаходження НСД (x^{3+1}, x^{2+1})

Крок i	f_i	g_i	$d_i(r_i)$	q_i	$r_i = f_i\cdot a + g_i\cdot b$
−1	1	0	x^3+1	-	$1\cdot(x^3+1)+0\cdot(x^2+1)= x^3+1$
0	0	1	x^2+1	-	$0\cdot(x^3+1)+1(x^2+1)= x^2+1$
1	1	x	$x+1$	$[x^3+1/x^2+1]=x$	$1\cdot(x^3+1)+x\cdot(x^2+1)=x+1$
2	$x+1$	x^2+x+1	0	$[x^2+1/x+1]=x+1$	$(x+1)(x^3+1)+(x^2+x+1)(x^2+1)=0$

В основі алгоритму Евкліда для кодів РС лежить процедура знаходження НСД двох поліномів $d(x) = x^{2t}$ і $c(x) = S(x)$ і декодування цікавляться не кінцевим результатом алгоритму Евкліда, а проміжними результатами. Якщо на i-му кроці алгоритму для деяких значень $\Omega_i(x) = r_i(x)$, $\sigma_i(x) = b_i(x)$ і $\deg[r_i(x)] < t$ отримано вираз

$$r_i(x) = a_i(x)x^{2t} + b_i(x)S(x), \qquad (6.49)$$

тоді поліном значень помилок

$$\Omega(x) = S(x)\sigma(x) + x^{2t}a(x). \qquad (6.50)$$

При цьому використовується властивість алгоритму Евкліда:

$\deg[b_i(x)] + \deg[r_{i-1}(x)] = \deg[d(x)]$.

Якщо $d(x) = x^{2t}$, то $\deg[\sigma_i(x)]+\deg[\Omega_{i-1}(x)] = 2t$,

$\deg[\sigma_i(x)] + \deg[\Omega_i(x)] < 2t$.

З появою $v \le t$ помилок маємо $\deg[\Omega_i(x)] < \deg[\sigma_i(x)] \le t$.

Існує єдиний з точністю до постійного множника (елемента поля) многочлен $\sigma_i(x)$ ступеня меншого або рівного t, що задовольняє рівняння (6.47). Крім того, якщо $\deg[\Omega_{i-1}(x)] \ge t$ при $\deg[\sigma_i(x)] \le t$ і $\deg[\Omega_i(x)] < t$, то $\deg[\sigma_{i+1}(x)] > t$. Тому проміжні результати на i-му кроці дають єдине рішення, що цікавить нас, ключового рівняння.

Таким чином, алгоритм Евкліда для вирішення ключового рівняння слід застосовувати доти, доки не буде виконана умова $\deg[\Omega_i(x)] < t$.

При обчисленні значень $\sigma_i(x)$ та $\Omega_i(x)$ слід використовувати властивість алгоритму Евкліда − кожне із значень $\sigma_i(x)$ і $\Omega_i(x)$ виходить із двох попередніх значень за такою формулою :

$$E_i(x) = E_{i-2}(x) + q_i(x)E_{i-1}(x), \text{ где } q_i(x) = \left[\frac{\Omega_{i-2}(x)}{\Omega_{i-1}(x)}\right].$$

Після того, як знайдено багаточленів локаторів і значень помилок, багаточлен помилок обчислюється за алгоритмом Форні.

Алгоритм Евкліда для декодування кодового полінома $f(x)$, закодованого кодом РС, здатного виправляти помилки кратності t, довжини $n=2^m-1$ з породжуючим многочленом $g(x)$, коренямим якого є елементи α^b, α^{b+1}, …, α^{b+2t-1} розширеного поля $GF(2^m)$, полягає в наступному:

1. Обчислити компоненти синдрому

$S_j = f(\alpha^j)$, $j=1, \ldots, 2t$, $i = b, \ldots, b+2t-1$.

2. Побудувати синдромний многочлен $S(x)$

$$S(x) = S_1 + S_2x + \ \ldots \ + S_{2t}x^{2t-1}.$$

3. Встановити початкові значення

$\sigma_{-1}(x) = 0, \sigma_0(x) = 1, \Omega_{-1}(x) = x^{2t}, \Omega_0(x) = S(x)$.

4. Перевірити ступінь полінома $\deg[\Omega_i(x)] < t$ якщо ступінь менше значення t тоді виконати пункт 5, інакше виконати пункт 7.

5. Обчислити значення $q_i(x) = \left[\frac{\Omega_{i-2}(x)}{\Omega_{i-1}(x)}\right]$.

6. Встановити нові значення та виконати пункт 4

$\Omega_i(x) = \Omega_{i-2}(x) + q_i(x) \cdot \Omega_{i-1}(x)$, $\sigma_i(x) = \sigma_{i-2}(x) + q_i(x) \cdot \sigma_{i-1}(x)$.

7. Покласти $\sigma(x) = \sigma_i(x)$, $\Omega(x) = \Omega_i(x)$.

8. Знайти значення коренів α^i полінома локаторів помилок $\sigma(x)$ за допомогою методу Ченя $\sigma(\alpha^i) = 0$, потім обчислити локатори позицій помилок $\alpha^j = \alpha^{-i}$ як величини, обернені до коренів α^i.

9. Знайти позицію помилок. Ступені локаторів помилок α^j є значеннями позицій помилок $j_1, j_2 \ldots j_v$.

10. Обчислити формальну похідну многочлена локаторів помилок $\sigma'(x)$

$$\sigma'(x) = \sum_{j=1}^{v} (j\sigma_j) x^{j-1}.$$

11. За допомогою алгоритму Форні знайти значення помилок

$$e_l = (\alpha^j)^{1-b} \frac{\Omega(\alpha^{-j})}{\sigma'(\alpha^{-j})}.$$

12. Формується многочлен помилок $e(x)$, коефіцієнтами якого є значення помилок e_l, а ступенями доданків є локатори помилок.

13. Виправити помилки в прийнятій кодовій послідовності, за допомогою додавання кодової комбінації і многочлена помилок $v(x) = f(x) + e(x)$.

Приклад 6.24. Код Ріда-Соломона для розширеного поля $GF(2^3)$, ($p(x) = x^3+x+1$) виправляє двократні помилки $t = 2$. Довжина коду $n = 2^3 - 1 = 7$, значення ступеня першого кореня $b=4$. Несистематичним РС кодом закодовано інформаційну послідовність $u(x)=\alpha^3x^2+\alpha + \alpha^5$, що складається з трьох вісімкових символів. Кодовий поліном – $v(x)=\alpha^3x^6+\alpha^4x^5+\alpha x^4+\alpha x^3+\alpha^4x^2+\alpha^4x+\alpha^6$. При передачі кодової послідовності $v(x)$ була прийнята послідовність з помилками $f(x)=v(x)+e(x)$, де $e(x) = \alpha^3x^6 + \alpha^2x^2$ – вектор помилки. Продемонструвати процес виправлення помилок у прийнятій послідовності $f(x)$.

Рішення.

Під час передачі кодової послідовності $v(x)$ сталося дві помилки, і прийнята кодова послідовність дорівнює

$f(x) = v(x) + e(x) = (\alpha^3x^6+\alpha^4x^5+\alpha x^4+\alpha x^3+\alpha^4x^2+\alpha^4x+\alpha^6) + (\alpha^3x^6+\alpha^2x^2) =$
$= \alpha^4x^5+\alpha x^4+\alpha x^3+\alpha x^2+\alpha^4x+\alpha^6.$

1. Знаходимо компоненти синдромного вектора використовуючи $2t = 4$ послідовних коренів породжуючого полінома α^4, α^5, α^6, α^7. Усі ступені елемента α повинні бути приведені по модулю 7.

Наприклад: $\alpha^{22} = \alpha^{22 \bmod 7} = \alpha^1$.

$S_1 = f(x=\alpha^4) = \alpha^4(\alpha^4)^5+\alpha(\alpha^4)^4+\alpha(\alpha^4)^3+\alpha(\alpha^4)^2+\alpha^4(\alpha^4)+\alpha^6 = \alpha^4\cdot\alpha^{20}+\alpha\cdot\alpha^{16}+$
$+\alpha\cdot\alpha^{12}+\alpha\cdot\alpha^8+\alpha^4\cdot\alpha^4+\alpha^6 = \alpha^{24}+\alpha^{17}+\alpha^{13}+\alpha^9+\alpha^8+\alpha^6 = \alpha^3+\alpha^3+\alpha^6+\alpha^2+\alpha+\alpha^6 =$
$= \alpha^2+\alpha = \underline{\alpha^4;}$

$S_2 = f(x=\alpha^5) = \alpha^4(\alpha^5)^5+\alpha(\alpha^5)^4+\alpha(\alpha^5)^3+\alpha(\alpha^5)^2+\alpha^4(\alpha^5)+\alpha^6 = \alpha^4\cdot\alpha^{25}+\alpha\cdot\alpha^{20}+$
$+\alpha\cdot\alpha^{15}+\alpha\cdot\alpha^{10}+\alpha^4\cdot\alpha^5+\alpha^6 = \alpha^{29}+\alpha^{21}+\alpha^{16}+\alpha^{11}+\alpha^9+\alpha^6 = \alpha+1+\alpha^2+\alpha^4+\alpha^2+\alpha^6 =$
$= \alpha^3+\alpha^4+\alpha^6 = \alpha^6+\alpha^6 = \underline{0;}$

$S_3 = f(x=\alpha^6) = \alpha^4(\alpha^6)^5+\alpha(\alpha^6)^4+\alpha(\alpha^6)^3+\alpha(\alpha^6)^2+\alpha^4(\alpha^6)+\alpha^6 = \alpha^4\cdot\alpha^{30}+$
$+\alpha\cdot\alpha^{24}+\alpha\cdot\alpha^{18}+\alpha\cdot\alpha^{12}+\alpha^4\cdot\alpha^6+\alpha^6 = \alpha^{34}+\alpha^{25}+\alpha^{19}+\alpha^{13}+\alpha^{10}+\alpha^6 = \alpha^6+\alpha^4+$
$+\alpha^5+\alpha^6+\alpha^3+\alpha^6 = 1+\alpha^3+\alpha^6 = \alpha+\alpha^6 = \underline{\alpha^5;}$

$S_4 = f(x=\alpha^7=1) = \alpha^4(1)^5+\alpha(1)^4+\alpha(1)^3+\alpha(1)^2+\alpha^4(1)+\alpha^6 = \alpha^4+\alpha+\alpha+\alpha+\alpha^4+\alpha^6 =$
$= \alpha + \alpha^6 = \underline{\alpha^5.}$

Компоненти синдрома дорівнюють $\boldsymbol{S_1 = \alpha^4; S_2 = 0; S_3 = \alpha^5; S_4 = \alpha^5.}$

2. Знаходимо синдромний многочлен (6.41).

$S(x) = S_1x^0+S_2x^1+S_3x^2+S_4x^3 = \alpha^4 + \alpha^5x^2 + \alpha^5x^3.$

3. Встановлення початкових значень та виконання алгоритму Евкліда.

$\sigma_{-1}(x) = 0, \sigma_0(x) = 1,$

$\Omega_{-1}(x) = x^{2t} = x^4, \Omega_0(x) = S(x) = \alpha^4 + \alpha^5x^2 + \alpha^5x^3,$

$i = 0.$

Перевірка умови $\deg[\Omega_0(x)] < t \rightarrow (3<2)$? – ні

– крок 1. $i = 1.$

$$q_1(x) = \left[\frac{\Omega_{-1}(x)}{\Omega_0(x)}\right] = \left[\frac{x^4}{\alpha^4 + \alpha^5x^2 + \alpha^5x^3}\right] = \alpha^2x + \alpha^2.$$

$$
\begin{array}{r|l}
x^4 & \alpha^5x^3+\alpha^5x^2+\alpha^4 \\
\cline{2-2}
x^4+\alpha^7x^3+\alpha^6x & \alpha^2x+\alpha^2 \\
\cline{1-1}
\alpha^7x^3+\alpha^6x & \\
\alpha^7x^3+\alpha^7x^2+\alpha^6 & \\
\cline{1-1}
\alpha^7x^2+\alpha^6x+\alpha^6 &
\end{array}
$$

$\Omega_1(x) = \Omega_{-1}(x)+q_1(x)\cdot\Omega_0(x) = x^4 + (\alpha^2x+\alpha^2)\cdot(\alpha^5x^3+\alpha^5x^2+\alpha^4) = x^4+\alpha^7x^4+\alpha^7x^3+$
$+\alpha^6x+\alpha^7x^3+\alpha^7x^2+\alpha^6 = (1+\alpha^7)x^4+(\alpha^7+\alpha^7)x^3+\alpha^7x^2+\alpha^6x+ \alpha^6 = 0\cdot x^4+0\cdot x^3+ \alpha^7x^2+$
$+\alpha^6x+ \alpha^6 = \alpha^7x^2+\alpha^6x+ \alpha^6 = x^2+\alpha^6x+ \alpha^6.$

$\sigma_1(x) = \sigma_{-1}(x)+q_1(x)\cdot\sigma_0(x) = 0+ (\alpha^2x+\alpha^2)\cdot 1 = \alpha^2x+\alpha^2.$

Перевірка умови $\deg[\Omega_1(x)] < t \rightarrow (2<2)$? – ні

– крок 2. $i = 2$.

$$q_2(x) = \left[\frac{\Omega_0(x)}{\Omega_1(x)}\right] = \left[\frac{\alpha^5x^3+\alpha^5x^2+\alpha^4}{x^2+\alpha^6x+\alpha^6}\right] = \alpha^5x+1.$$

$$
\begin{array}{r|l}
\alpha^5x^3+\alpha^5x^2+\alpha^4 & \alpha^4x+\alpha^5 \\
\cline{2-2}
\alpha^5x^3+\alpha^4x^2+\alpha^4x & \alpha^5x+1 \\
\cline{1-1}
x^2+\alpha^4x+\alpha^4 & \\
x^2+\alpha^6x+\alpha^6 & \\
\cline{1-1}
\alpha^6x^2+\alpha^2x+\alpha^3 &
\end{array}
$$

$\Omega_2(x) = \Omega_0(x)+q_2(x)\cdot\Omega_1(x) = \alpha^5x^3+\alpha^5x^2+\alpha^4+(\alpha^5x+1)\cdot(x^2+\alpha^6x+\alpha^6) = \alpha^5x^3+$
$+\alpha^5x^2+\alpha^4+\alpha^5x^3+\alpha^{11}x^2+\alpha^{11}x+x^2+\alpha^6x+\alpha^6 = \alpha^5x^3+\alpha^5x^2+\alpha^4+\alpha^5x^3+\alpha^4x^2+\alpha^4x+x^2+$
$+\alpha^6x+\alpha^6 = (\alpha^5+\alpha^5)\,x^3+(\alpha^5+\alpha^4+1)\,x^2+(\alpha^4+\alpha^6)\,x+\alpha^4+\alpha^6=$
$= 0\cdot x^3+(1+1)\,x^2+\alpha^3x+\alpha^3= = 0\cdot x^3+0\cdot x^2+\alpha^3x+\alpha^3 = \alpha^3x+\alpha^3.$

$\sigma_2(x) = \sigma_0(x)+q_2(x)\cdot\sigma_1(x) = 1+(\alpha^5x+1)(\alpha^2x+\alpha^2) = 1+\alpha^7x^2+\alpha^7x+\alpha^2x+\alpha^2 =$
$= x^2+(1+\alpha^2)\,x+\alpha^2+1 = x^2+\alpha^6x+\alpha^6.$

Перевірка умови $\deg[\Omega_2(x)] < t \rightarrow (1<2)$? – так. Стоп!!!

4. Формування багаточлена локаторів та значень помилок.

$\sigma(x) = \sigma_2(x) = x^2+\alpha^6x+\alpha^6.$

$\Omega(x) = \Omega_2(x) = \alpha^3x+\alpha^3.$

5. Знаходимо корені багаточлена локаторів помилок $\sigma(x)$.

Для знаходження коренів багаточлена використовується метод Ченя – перевірка для всіх ненульових елементів поля $GF(2^3)$.

$\sigma(x) = x^2+\alpha^6x+\alpha^6.$

$\sigma(x=\alpha) = \alpha^2+\alpha^6\alpha+\alpha^6 = \alpha^2 +\alpha^7+\alpha^6 = \alpha^2+1+\alpha^6 = \alpha^6 +\alpha^6 = 0.$

$\sigma(x=\alpha^2) = \alpha^4+\alpha^6\alpha^2+\alpha^6 = \alpha^4+\alpha^8+\alpha^6 = \alpha^4+\alpha^1+\alpha^6 = \alpha^2+\alpha^6 = 1.$

$\sigma(x=\alpha^3) = \alpha^6+\alpha^6\alpha^3+\alpha^6 = \alpha^6+\alpha^9+\alpha^6 = \alpha^6+\alpha^2+\alpha^6 = \alpha^2+0 = \alpha^2.$

$\sigma(x=\alpha^4) = \alpha^8+\alpha^6\alpha^4+\alpha^6 = \alpha^1+\alpha^{10}+\alpha^6 = \alpha^1+\alpha^3+\alpha^6 = 1+\alpha^6 = \alpha^2.$

$\sigma(x=\alpha^5) = \alpha^{10}+\alpha^6\alpha^5+\alpha^6 = \alpha^3+\alpha^{11}+\alpha^6 = \alpha^3+\alpha^4+\alpha^6 = \alpha^6+\alpha^6 = 0.$

$\sigma(x=\alpha^6) = \alpha^{12}+\alpha^6\alpha^6+\alpha^6 = \alpha^5+\alpha^{12}+\alpha^6 = \alpha^5+\alpha^5+\alpha^6 = 0+\alpha^6 = \alpha^6.$

$\sigma(x=\alpha^7=1) = 1+\alpha^6+\alpha^6 = 1+0 = 1.$

Таким чином, корені полінома помилок $\sigma(x)$ − $\underline{\alpha \text{ и } \alpha^5.}$

6. Знаходимо локатори та позиції помилок.

Локатори помилок дорівнюють елементам, які є зворотним коренямм многочлена помилок:

− для елемента α^1 зворотнім є елемент α^6, оскільки з табл. Г.2 (Додаток Г) знаходимо $\alpha^1 \cdot \alpha^6 = 1$;

− для елемента α^5 зворотнім є елемент α^2, оскільки з табл. Г.2 (Додаток Г) знаходимо $\alpha^5 \cdot \alpha^2 = 1$.

Таким чином, локаторами помилок є елементи α^6, α^2. Позиції помилок знаходяться як значення ступенів локаторів помилок α^6, α^2.

Помилки прийнятої послідовності $f(x)$ знаходяться на позиціях $j_1 = 6$, $j_2 = 2$.

7. Обчислимо значення помилок та знайдемо багаточлен помилок $e(x)$.

Величини помилок (коефіцієнтів) e_1, e_2 для формування полінома помилок $e(x)$ знаходять за допомогою методу Форні

$$e_l = (\alpha^j)^{1-b} \frac{\Omega(\alpha^{-j})}{\sigma'(\alpha^{-j})}.$$

де $\sigma`(x) = 2x + \alpha^6 = \alpha^6$ − формальна похідна багаточлена локаторів помилок; $b = 4$ − ступінь першого кореня; $\alpha^{-j} = \alpha$, α^5 − корені $\sigma(x)$; $\alpha^j = \alpha^6$, α^2 – локатор помилок, $\Omega(x) = \alpha^3 x + \alpha^3$.

Обчислимо значення коефіцієнта e_1 для локатора помилки $\alpha^j = \alpha^6$ та для кореня $\alpha^{-j} = \alpha^1$:

$$e_1 = (\alpha^6)^{1-4} \frac{\Omega(\alpha^1)}{\sigma'(\alpha^1)} = (\alpha^6)^{-3} \frac{\alpha^3\alpha + \alpha^3}{\alpha^6} = \alpha^{-18}(\alpha^4 + \alpha^3)\alpha^{-6} = \alpha^{-4}\alpha^6\alpha^1 = \alpha^3\alpha^7 = \alpha^3$$

Обчислимо значення коефіцієнта e_1 для локатора помилок $\alpha^j = \alpha^2$ та для кореня $\alpha^{-j} = \alpha^5$:

$$e_1 = (\alpha^2)^{1-4} \frac{\Omega(\alpha^5)}{\sigma'(\alpha^5)} = (\alpha^2)^{-3} \frac{\alpha^3\alpha^5 + \alpha^3}{\alpha^6} = \alpha^{-6}(\alpha^5 + \alpha^3)\alpha^{-6} = \alpha \cdot 1 \cdot \alpha = \alpha^2.$$

Таким чином, многочлен помилок $e(x) = \alpha^3 x^6 + \alpha^2 x^2$.

8. Виправимо помилки у прийнятій послідовності *f*(*x*).

Виправлення помилок у прийнятій послідовності *f*(*x*) здійснюється шляхом складання комбінації *f*(*x*) и полінома помилок *e*(*x*)

$v(x) = f(x) + e(x) = (\alpha^4 x^5 + \alpha x^4 + \alpha x^3 + \alpha x^2 + \alpha^4 x + \alpha^6) + (\alpha^3 x^6 + \alpha^2 x^2) =$

$= \alpha^3 x^6 + \alpha^4 x^5 + \alpha x^4 + \alpha x^3 + \alpha^4 x^2 + \alpha^4 x + \alpha^6.$

Таким чином, виправлена кодова послідовність без помилок має вигляд $v(x) = \alpha^3 x^6 + \alpha^4 x^5 + \alpha x^4 + \alpha x^3 + \alpha^4 x^2 + \alpha^4 x + \alpha^6$.

Для відновлення інформаційної послідовності *u*(*x*), яка передавалася, необхідно виконати ділення $u(x) = v(x) / g(x)$.

$$
\begin{array}{l|l}
\alpha^3 x^6+\alpha^4 x^5+\alpha x^4+\alpha x^3+\alpha^4 x^2+\alpha^4 x+\alpha^6 & x^4+\alpha^6 x^3+\alpha^6 x^2+\alpha^3 x+\alpha \\
\alpha^3 x^6+\alpha^2 x^5+\alpha^2 x^4+\alpha^6 x^3+\alpha^4 x^2 & \alpha^3 x^2+\alpha x+\alpha^5 \\
\hline
\quad \alpha x^5+\alpha^4 x^4+\alpha^5 x^3+\alpha^4 x+\alpha^6 & \\
\quad \alpha x^5+\alpha^7 x^4+\alpha^7 x^3+\alpha^4 x^2+\alpha^2 x & \\
\hline
\quad\quad \alpha^5 x^4+\alpha^4 x^3+\alpha^4 x^2+\alpha x+\alpha^6 & \\
\quad\quad \alpha^5 x^4+\alpha^4 x^3+\alpha^4 x^2+\alpha x+\alpha^6 & \\
\hline
\quad\quad\quad 0 &
\end{array}
$$

Декодована інформаційна послідовність дорівнює *u*(*x*) = $\alpha^3 x^2+\alpha x+\alpha^5$.

Завдання для самостійного вирішення

1. Циклічно зрушити кодову комбінацію A={1010100111011} на чотири розряди праворуч, а кодову комбінацію B={100001110101} на п'ять розрядів ліворуч.

2. Представити двійкову послідовність A={001101011110101} у вигляді полінома від змінної x.

3. Представити поліном $x^{17}+x^{14}+x^{12}+x^{11}+x^{10}+x^6+x^4+x^3+x^2$ у вигляді двійкової послідовності

4. Скласти двійкові послідовності A={1010100111010} та B={11101100011} у двійковій та поліноміальній формі.

5. Перемножити двійкові послідовності A={1101100101011} та B={1011100010} у двійковій та поліноміальній формі.

6. Поділити двійкові послідовності A={1011011100111001010} та B={1100100010} у двійковій та поліноміальній формі.

7. Циклічно зрушити кодову комбінацію A={1010110010100} ліворуч на три розряди у поліноміальній формі.

8. Знайти всі незвідні багаточлени четвертого ступеня.

9. Знайти усі незвідні поліноми – дільники двійкового полінома x^9+1.

10. Закодувати послідовність $a(x)=x^3+1$ несистематичним та систематичним методом циклічним кодом з породжуючим поліномом $g(x) = x^3+x^2 + 1$.

11. Закодувати послідовність A={01110101010} несистематичним і систематичним методом циклічним кодом з породжуючим поліномом g = {11001}.

12. Декодувати повідомлення A={0100100011011110},

y_2={0100101011011010}, закодовані систематичним циклічним кодом, що виправляє поодиноку помилку t_c=1 з утворюючим поліномом g = {11001}.

13. Почергово внести у повідомлення y={1100101}, закодоване систематичним циклічним кодом, з t_c=1 та утворюючим поліномом g = {1101} помилку в кожному із семи розрядів та декодувати отримані повідомлення.

14. Побудувати двійковий БЧХ код довжини n = 15 який здатний виправляти трикратні помилки.

15. Закодувати двійковим БЧХ кодом (15, 5, 7) інформаційну послідовність $u(x) = x^6 + x^5 + x^3 + x + 1$.

16. Декодувати прийняту кодову послідовність $f(x) = x^{14}+x^{11}+x^8+x^6+x^4+x^3+x^2+x+1$ для БЧХ коду (15, 5, 7) з породжуючим багаточленом $g(x) = x^{10}+x^8+x^5+x^4+x^2+x+1$.

17. Знайдіть породжуючий багаточлен коду Ріда-Соломона, довжиною 15, що виправляє подвійні помилки.

18. Закодувати інформаційну послідовність $u(x)=\alpha^5x^2+\alpha x +\alpha^2$ кодом Ріда-Соломона (7, 3, 5).

19. Знайдіть поліном локаторів помилок та поліном значень помилок для коду РС довжиною 15, що виправляє подвійні помилки, якщо прийняте кодове слово $f(x) = \alpha x^{10} + \alpha^{10}x^2$.

20. На вхід декодера надійшла комбінація $f(x) = \alpha + x^5 + x^{10}$. Обчислити синдромний поліном та значення помилок, використовуючи алгоритм Евкліда та Форні.

СПИСОК ЛІТЕРАТУРИ

1. Безруков В. В. Теорія інформації: [Навч. посібник] / В. В. Безруков, В. Я. Кізяков, В. І. Профатілов. – Дніпропетровськ : ДИИТ (ДДТУЗТ, 2001. – 110 с.

2. Жураковський Ю. П. Теорія інформації та кодування: [Підручник] / Ю. П. Жураковський, В. П. Полторак. – К. : Вища школа, 2001.– 255 с.

3. Жураковський Ю. П. Теорія інформації та кодування в задачах: [Навчальний посібник]/ Ю. П. Жураковський, В. В. Гніліцький. – Житомир: ЖІТІ, 2002. – 230 с.

4. Кожевников В. Л. Теорія інформації та кодування [Текст]: [Навч. посібник] / В. Л. Кожевников, А. В. Кожевников. – Д.: Національний гірничий університет, 2011. – 108 с.

5. Курко А. М. Введення в теорію інформації [Електронний ресурс]: Посібник до вивчення дисципліни «Теорія інформації» / А. М. Курко, В. Я. Решетняк. – Тернопіль: Тернопільський національний технічний університет ім. Івана Пулюя, 2017 – 108 с.– Режим доступу: http://elartu.tntu.edu.ua/handle/lib/21919

6. Кузьмін І. В. Основи теорії інформації та кодування : [Підручник] / І. В. Кузьмін, І. В. Троцишин, А. І. Кузьмін, В. О. Кедрус, В. Р. Любчик; За ред. Іван Васильович Кузьмін.– 3–тє вид.– Хмельницький : ХНУ, 2009.– 373 с.

7. Сорока Л. С. Основи теорії інформації: [Навчальний посібник] / Л. С. Сорока.– Харків: ХНУ ім. В.Н.Каразіна, 2007. – 264 с.

8. Тулякова Н. О. Теорія інформації: [Навчальний посібник] [Електронний ресурс] / Н. О. Тулякова. – Суми: СумДУ, 2008. – 212 с. – Режим доступу: http://elkniga.info/book_156.html.

9. Теорія інформації і кодування: курс лекцій [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за спеціальністю 124 «Системний аналіз» /; уклад.: А.Є.Коваленко. Київ : КПІ ім. Ігоря Сікорського, 2020. 248 с...

10. Іващенко П.В. Основи теорії інформації: навч. посіб. / П.В. Іващенко –Одеса: ОНАЗ ім. О.С. Попова, 2015. – 53 с

11. Беркман Л.Н., Бондарчук А.П., Гайдур Г.І., Чумак Н.С. Кодування джерел інформації та каналів зв’язку. Навч. посібник підготовлено для самостійної роботи студентів вищих навчальних закладів

за кредитно-модульною організацією навчального процесу. – Київ: ННІТІ ДУТ, 2018. – 91с.

12. Майданюк В. П. Методи і засоби комп'ютерних інформаційних технологій. Кодування зображень. Навчальний посібник. – Вінниця: ВДТУ, 2001. – 63 с.

13. Рябенький В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: Новий світ- 2000, 2009, - 736 с.

14. Коваленко А.Є. Теорія інформації та кодування: Практикум для студентів напряму підготовки 6.040303 «Системний аналіз». Київ:НТУУ «КПІ», 2014.198 с.

15. Бабак В.П. Обробка сигналів: Підручник / В.П. Бабак, В.С. Хандецький, Е. Шрюфер. – К.:Либідь. – 1996. – 392 с.

16. Тулякова Н.О. Теорія інформації: Навчальний посібник. Суми: Вид-во СумДУ, 2008. 212 с. 2010. 248с.

17. Mauro Barni, Benedetta Tondi Lecture notes on Information Theory and Coding. Siena: Universit` a degli Studi di Siena Facolt`a di Ingegneria, 2012. 156.

18. http://www.codetables.de/BKLC/Tables.php?q=2&n0=1&n1=256&k0=1&k1=256

19. https://web.posibnyky.vntu.edu.ua/firen/6bilynskyj_elektronni_systemy/

20. https://tik-diit.dp.ua/m356/

21. https://dut.edu.ua/ua/lib/1/category/2125

22. https://studfile.net/preview/5465934/

ДОДАТОК А
ЗНАЧЕННЯ ЛОГАРИФМІВ

Таблиця А.1 – Значення величин $-p\log_2 p$

P	$-p\log_2 p$	p	$-p\log_2 p$	p	$-p\log_2 p$
0,00	0,0000	0,36	0,5306	0,71	0,3508
0,01	0,0664	0,37	0,5307	0,72	0,3412
0,02	0,1129	0,38	0,5305	0,73	0,3314
0,03	0,1518	0,39	0,5298	0,74	0,3215
0,04	0,1858	0,4	0,5288	0,75	0,3113
0,05	0,2161	0,41	0,5274	0,76	0,3009
0,06	0,2435	0,42	0,5256	0,77	0,2903
0,07	0,2686	0,43	0,5236	0,78	0,2796
0,08	0,2915	0,44	0,5211	0,79	0,2687
0,09	0,3127	0,45	0,5184	0,8	0,2575
0,1	0,3322	0,46	0,5153	0,81	0,2462
0,11	0,3503	0,47	0,5120	0,82	0,2348
0,12	0,3671	0,48	0,5083	0,83	0,2231
0,13	0,3826	0,49	0,5043	0,84	0,2113
0,14	0,3971	0,5	0,5000	0,85	0,1993
0,15	0,4105	0,51	0,4954	0,86	0,1871
0,16	0,4230	0,52	0,4906	0,87	0,1748
0,17	0,4346	0,53	0,4854	0,88	0,1623
0,18	0,4453	0,54	0,4800	0,89	0,1496
0,19	0,4552	0,55	0,4744	0,9	0,1368
0,2	0,4644	0,56	0,4684	0,91	0,1238
0,21	0,4728	0,57	0,4623	0,92	0,1107
0,22	0,4806	0,58	0,4558	0,93	0,0974
0,23	0,4877	0,59	0,4491	0,94	0,0839
0,24	0,4941	0,6	0,4422	0,95	0,0703
0,25	0,5000	0,61	0,4350	0,96	0,0565
0,26	0,5053	0,62	0,4276	0,97	0,0426
0,27	0,5100	0,63	0,4199	0,98	0,0286
0,28	0,5142	0,64	0,4121	0,99	0,0144
0,29	0,5179	0,65	0,4040	1,00	0,0000
0,3	0,5211	0,66	0,3956		
0,31	0,5238	0,67	0,3871		
0,32	0,5260	0,68	0,3783		
0,33	0,5278	0,69	0,3694		
0,34	0,5292	0,7	0,3602		
0,35	0,5301				

Таблиця А.2 – Значення величин $\log_2 n$

n	$\log_2 n$	n	$\log_2 n$	n	$\log_2 n$
1	0,0000	7	2,8074	16	4,0000
2	1,0000	8	3,000	20	4,3219
3	1,5850	9	3,1699	25	4,6439
4	2,0000	10	3,3219	50	5,6439
5	2,3219	11	3,4594	100	6,6439
6	2,5850	12	3,5849	200	7,6439

ДОДАТОК Б

НЕЗВІДНІ ТА ПОРОДЖУЮЧІ ПОЛІНОМИ

Таблиця Б.1 – Незвідні поліноми

Ступінь	Многочлен	Двійковий вектор
1	$x + 1$	11
2	$x^2 + x + 1$	111
3	$x^3 + x + 1$ $x^3 + x^2 + 1$	1011 1101
4	x^4+x+1 x^4+x^3+1 $x^4+x^3+x^2+x+1$	10011 11001 11111
5	x^5+x^2+1 x^5+x^3+1 $x^5+x^3+x^2+x+1$ $x^5+x^4+x^2+x+1$ $x^5+x^4+x^3+x+1$ $x^5+x^4+x^3+x^2+x+1$	100101 101001 101111 110111 111011 111101
6	x^6+x+1 x^6+x^3+1 $x^6+x^4+x^2+x+1$ $x^6+x^4+x^3+x+1$ x^6+x^5+1 $x^6+x^5+x^2+x+1$ $x^6+x^5+x^3+x^2+1$ $x^6+x^5+x^4+x+1$ $x^6+x^5+x^4+x^2+1$	1000011 1001001 1010111 1011011 1100001 1100111 1101101 1110011 1110101
7	x^7+x+1 x^7+x^3+1 $x^7+x^3+x^2+x+1$ $x^7+x^4+x^3+x^2+1$ $x^7+x^5+x^2+x+1$ $x^7+x^5+x^3+x+1$ $x^7+x^6+x^3+x+1$ $x^7+x^6+x^4+x+1$	10000011 10001001 10001111 10011101 10100111 10101011 11001011 11010011
8	$x^8+x^4+x^3+x+1$ $x^8+x^4+x^3+x^2+1$ $x^8+x^5+x^3+x+1$ $x^8+x^5+x^3+x^2+1$ $x^8+x^6+x^5+x^2+1$ $x^8+x^7+x^3+x+1$ $x^8+x^7+x^5+x^3+1$	100011011 100011101 100101011 100101101 101100101 110001011 110101001
9	x^9+x+1 x^9+x^4+1 $x^9+x^4+x^2+x+1$ $x^9+x^4+x^3+x+1$ $x^9+x^5+x^4+x+1$ $x^9+x^6+x^5+x^2+1$	1000000011 1000010001 1000010111 1000011011 1000110011 1001100101
10	$x^{10}+x^3+1$	10000001001

Таблиця Б.2 – Примітивні многочлени до ступеня $m = 20$

$x+1$	x^6+x+1	$x^{11}+x^2+1$	$x^{16}+x^{12}+x^3+x+1$
x^2+x+1	x^7+x^3+1	$x^{12}+x^6+x^4+x+1$	$x^{17}+x^3+1$
x^3+x+1	$x^8+x^4+x^3+x^2+1$	$x^{13}+x^4+x^3+x+1$	$x^{18}+x^7+1$
x^4+x+1	x^9+x^4+1	$x^{14}+x^9+x^5+x+1$	$x^{19}+x^5+x^2+x+1$
x^5+x^2+1	$x^{10}+x^3+1$	$x^{15}+x+1$	$x^{20}+x^3+1$

Таблиця Б.3 – Породжуючі поліноми для примітивних БЧХ кодів

n	***k***	***t***	***r***	Породжуючий поліном у вісімковій системі числення
7	4	1	3	13
15	11	1	4	23
	7	2	8	721
	5	3	10	2467
31	26	1	5	45
	21	2	10	3551
	16	3	15	107657
	11	5	20	5423325
	6	7	25	313365047
63	57	1	6	103
	51	2	12	12471
	45	3	18	1701317
	39	4	24	166623567
	36	5	27	1033500423
	30	6	33	1574641656547
	24	7	39	17323260404441
	18	10	45	1363026512351725
127	120	1	7	211
	113	2	14	41567
	106	3	21	11554743
	99	4	28	3447023271
	92	5	35	624730022327
	85	6	42	130704476332273
	78	7	49	26230002166130115
	71	9	56	6255010713253127753
	64	10	63	1206534025570773100045
255	247	1	8	435
	239	2	16	267543
	231	3	24	156720665
	223	4	32	75626641375
	215	5	40	2315754726421
	207	6	48	16176560567636227
	199	7	56	7633031270420722341
	191	8	64	2663470176115333714567
	187	9	68	52755313540001322236351
	179	10	76	22624710717340432416300455

Таблиця Б.4 – Породжуючі поліноми для непримітивних БЧХ кодів

m	***n***	***k***	***t***	Що породжує поліном у вісімковій формі
8	17	9	2	727
6	21	12	2	1663
11	23	12	3	5343
10	33	22	2	5145
10	33	12	4	3777
20	41	21	4	6647133
23	47	24	5	42073357
12	65	63	2	10761
12	65	40	4	354303067
9	73	46	4	1717773537

ДОДАТОК В

РОЗШИРЕНЕ КІНЦЕВЕ ПОЛЕ $GF(2^4)$

Таблиця В.1 − Різні види елементів поля $GF(2^4)$

Елементи поля $GF(2^4)$	$p(x)=x^4+x+1$			$p(x)=x^4+x^3+1$		
	Представлення елементів поля			Представлення елементів поля		
	поліном	вектор	ступінь α	поліном	вектор	ступінь γ
0	0	0	0	0	0	0
β_0	1	0001	1	1	0001	1
β_1	x	0010	α^1	x	0010	γ^1
β_2	x^2	0100	α^2	x^2	0100	γ^2
β_3	x^3	1000	α^3	x^3	1000	γ^3
β_4	$x+1$	0110	α^4	x^3+1	1001	γ^4
β_5	x^2+x	0110	α^5	x^3+x+1	1011	γ^5
β_6	x^3+x^2	1100	α^6	x^3+x^2+x+1	1111	γ^6
β_7	x^3+x+1	1011	α^7	x^2+x+1	0111	γ^7
β_8	x^2+1	0101	α^8	x^3+x^2+x	1110	γ^8
β_9	x^3+x	1010	α^9	x^2+1	0101	γ^9
β_{10}	x^2+x+1	0111	α^{10}	x^3+x	1010	γ^{10}
β_{11}	x^3+x^2+x	1110	α^{11}	x^3+x^2+1	1101	γ^{11}
β_{12}	x^3+x^2+x+1	1111	α^{12}	$x+1$	0011	γ^{12}
β_{13}	x^3+x^2+1	1101	α^{13}	x^2+x	0110	γ^{13}
β_{14}	x^3+1	1001	α^{14}	x^3+x^2	1100	γ^{14}

$\alpha^{15} = \alpha^0 = 1$; $\gamma^{15} = \gamma^0 = 1$

ДОДАТОК Г

АРИФМЕТИЧНІ ТАБЛИЦІ ПОЛІВ ГАЛУА $GF(2^4)$

Таблиця Г.1 – Додавання в розширеному полі $GF(2^4)$ ($p(x) = x^4+x+1$)

+	**0**	**1**	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	1	0	α^4	α^8	α^{14}	α^1	α^{10}	α^{13}	α^9	α^2	α^7	α^5	α^{12}	α^{11}	α^6	α^3
α^1	α^1	α^4	0	α^5	α^9	1	α^2	α^{11}	α^{14}	α^{10}	α^3	α^8	α^6	α^{13}	α^{12}	α^7
α^2	α^2	α^8	α^5	0	α^6	α^{10}	α	α^3	α^{12}	1	α^{11}	α^4	α^9	α^7	α^{14}	α^{13}
α^3	α^3	α^{14}	α^9	α^6	0	α^7	α^{11}	α^2	α^4	α^{13}	α	α^{12}	α^5	α^{10}	α^8	1
α^4	α^4	α	1	α^{10}	α^7	0	α^8	α^{12}	α^3	α^5	α^{14}	α^2	α^{13}	α^6	α^{11}	α^9
α^5	α^5	α^{10}	α^2	α	α^{11}	α^8	0	α^9	α^{13}	α^4	α^6	1	α^3	α^{14}	α^7	α^{12}
α^6	α^6	α^{13}	α^{11}	α^3	α^2	α^{12}	α^9	0	α^{10}	α^{14}	α^5	α^7	α	α^4	1	α^8
α^7	α^7	α^9	α^{14}	α^{12}	α^4	α^3	α^{13}	α^{10}	0	α^{11}	1	α^6	α^8	α^2	α^5	α^1
α^8	α^8	α^2	α^{10}	1	α^{13}	α^5	α^4	α^{14}	α^{11}	0	α^{12}	α	α^7	α^9	α^3	α^6
α^9	α^9	α^7	α^3	α^{11}	α	α^{14}	α^6	α^5	1	α^{12}	0	α^{13}	α^2	α^8	α^{10}	α^4
α^{10}	α^{10}	α^5	α^8	α^4	α^{12}	α^2	1	α^7	α^6	α	α^{13}	0	α^{14}	α^3	α^9	α^{11}
α^{11}	α^{11}	α^{12}	α^6	α^9	α^5	α^{13}	α^3	α	α^8	α^7	α^2	α^{14}	0	1	α^4	α^{10}
α^{12}	α^{12}	α^{11}	α^{13}	α^7	α^{10}	α^6	α^{14}	α^4	α^2	α^9	α^8	α^3	1	0	α	α^5
α^{13}	α^{13}	α^6	α^{12}	α^{14}	α^8	α^{11}	α^7	1	α^5	α^3	α^{10}	α^9	α^4	A	0	α^2
α^{14}	α^{14}	α^3	α^7	α^{13}	1	α^9	α^{12}	α^8	α	α^6	α^4	α^{11}	α^{10}	α^5	α^2	0

Таблиця Г.2 – Множення в розширеному полі $GF(2^4)$ ($p(x) = x^4+x+1$)

*	0	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
α^1	0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1
α^2	0	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1
α^3	0	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2
α^4	0	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3
α^5	0	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4
α^6	0	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5
α^7	0	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6
α^8	0	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7
α^9	0	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8
α^{10}	0	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9
α^{11}	0	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}
α^{12}	0	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}
α^{13}	0	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}
α^{14}	0	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}

Таблиця Г.3 – Додавання в розширеному полі $GF(2^4)$ ($p(x) = x^4+x^3+1$)

+	**0**	**1**	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	1	0	α^{12}	α^9	α^4	α^3	α^{10}	α^8	α^{13}	α^6	α^2	α^5	α^{14}	α^1	α^7	α^{11}
α^1	α^1	α^{12}	0	α^{13}	α^{10}	α^5	α^4	α^{11}	α^9	α^{14}	α^7	α^3	α^6	1	α^2	α^8
α^2	α^2	α^9	α^{13}	0	α^{14}	α^{11}	α^6	α^5	α^{12}	α^{10}	1	α^8	α^4	α^7	α^1	α^3
α^3	α^3	α^4	α^{10}	α^{14}	0	1	α^{12}	α^7	α^6	α^{13}	α^{11}	α^1	α^9	α^5	α^8	α^2
α^4	α^4	α^3	α^5	α^{11}	1	0	α^1	α^{13}	α^8	α^7	α^{14}	α^{12}	α^2	α^{10}	α^6	α^9
α^5	α^5	α^{10}	α^4	α^6	α^{12}	α^1	0	α^2	α^{14}	α^9	α^8	1	α^{13}	α^3	α^{11}	α^7
α^6	α^6	α^8	α^{11}	α^5	α^7	α^{13}	α^2	0	α^3	1	α^{10}	α^9	α^1	α^{14}	α^4	α^{13}
α^7	α^7	α^{13}	α^9	α^{12}	α^6	α^8	α^{14}	α^3	0	α^4	α^1	α^{11}	α^{10}	α^2	1	α^5
α^8	α^8	α^6	α^{14}	α^{10}	α^{13}	α^7	α^9	1	α^4	0	α^5	α^2	α^{12}	α^{11}	α^3	α^1
α^9	α^9	α^2	α^7	1	α^{11}	α^{14}	α^8	α^{10}	α^1	α^5	0	α^6	α^3	α^{13}	α^{12}	α^4
α^{10}	α^{10}	α^5	α^3	α^8	α^1	α^{12}	1	α^9	α^{11}	α^2	α^6	0	α^7	α^4	α^{14}	α^{13}
α^{11}	α^{11}	α^{14}	α^6	α^4	α^9	α^2	α^{13}	α^1	α^{10}	α^{12}	α^3	α^7	0	α^8	α^5	1
α^{12}	α^{12}	α^1	1	α^7	α^5	α^{10}	α^3	α^{14}	α^2	α^{11}	α^{13}	α^4	α^8	0	α^9	α^6
α^{13}	α^{13}	α^7	α^2	α^1	α^8	α^6	α^{11}	α^4	1	α^3	α^{12}	α^{14}	α^5	α^9	0	α^{10}
α^{14}	α^{14}	α^{11}	α^8	α^3	α^2	α^9	α^7	α^{13}	α^5	α^1	α^4	α^{13}	1	α^6	α^{10}	0

Таблиця Г.4 – Множення в розширеному полі $GF(2^4)$ ($p(x) = x^4+x^3+1$)

*	**0**	**1**	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
α^1	0	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1
α^2	0	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1
α^3	0	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2
α^4	0	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3
α^5	0	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4
α^6	0	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5
α^7	0	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6
α^8	0	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7
α^9	0	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8
α^{10}	0	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9
α^{11}	0	α^{11}	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}
α^{12}	0	α^{12}	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}
α^{13}	0	α^{13}	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}
α^{14}	0	α^{14}	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}

ДОДАТОК Д

МІНІМАЛЬНІ МНОГОЧЛЕНИ ДЛЯ ЕЛЕМЕНТІВ ПОЛЯ ГАЛУА $GF(2^4)$

Таблиця Д.1 – Розподіл елементів за циклотомічними класами поля $GF(2^4)$ ($p(x)=x^4+x+1$), мінімальні многочлени

Циклотомічні класи	Сполучені елементи. Корені многочлена $\varphi_i(x)$	Мінімальні многочлени $\varphi_i(x)$
$C_0=\{0\}$	α^0	$\varphi_0(x)=x+1$
$C_1=\{1, 2, 4, 8\}$	$\alpha^1, \alpha^2, \alpha^4, \alpha^8$	$\varphi_1(x)=x^4+x+1$
$C_3=\{3, 6, 9, 12\}$	$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$\varphi_3(x)=x^4+x^3+x^2+x+1$
$C_5=\{5, 10\}$	α^5, α^{10}	$\varphi_5(x)=x^2+x+1$
$C_7=\{7, 11, 13, 14\}$	$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$\varphi_7(x)=x^4+x^3+1$

Таблиця Д.2 – Розподіл елементів за циклотомічними класами поля $GF(2^4)$ ($p(x)=x^4+x^3+1$), мінімальні многочлени

Циклотомічні класи	Сполучені елементи. Корені многочлена $\varphi_i(x)$	Мінімальні многочлени $\varphi_i(x)$
$C_0=\{0\}$	α^0	$\varphi_0(x) = x + 1$
$C_1=\{1, 2, 4, 8\}$	$\alpha^1, \alpha^2, \alpha^4, \alpha^8$	$\varphi_1(x) = x^4 + x^3 + 1$
$C_3=\{3, 6, 9, 12\}$	$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$\varphi_3(x) = x^4 + x^3 + x^2 + x + 1$
$C_5=\{5, 10\}$	α^5, α^{10}	$\varphi_5(x) = x^2 + x + 1$
$C_7=\{7, 11, 13, 14\}$	$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$\varphi_7(x) = x^4 + x + 1$

ДОДАТОК Е

КОРЕГУВАЛЬНА ЗДІБНІСТЬ ДЕЯКИХ ДВІЙКОВИХ КОДІВ

Таблиця Е.1 – Залежність мінімальної кодової відстані $d_{\min}$ від загальної кількость біт n та кількості інформаційних біт k

n\k	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	1														
2	2	1													
3	3	2	1												
4	4	2	2	1											
5	5	3	2	2	1										
6	6	4	3	2	2	1									
7	7	4	4	3	2	2	1								
8	8	5	4	4	2	2	2	1							
9	9	6	4	4	3	2	2	2	1						
10	10	6	5	4	4	3	2	2	2	1					
11	11	7	6	5	4	4	3	2	2	2	1				
12	12	8	6	6	4	4	4	3	2	2	2	1			
13	13	8	7	6	5	4	4	4	3	2	2	2	1		
14	14	9	8	7	6	5	4	4	4	3	2	2	2	1	
15	15	10	8	8	7	6	5	4	4	4	3	2	2	2	1
16	16	10	8	8	8	6	6	5	4	4	4	2	2	2	2

Навчальне видання

ІВАШКО Андрій Володимирович
КРИЛОВА Вікторія Анатоліївна

ТЕОРІЯ ІНФОРМАЦІЇ ТА КОДУВАННЯ В ПРИКЛАДАХ І ЗАДАЧАХ

Навчальний посібник

для студентів спеціальностей «Автоматизація та комп'ютерно-інтегровані технології», «Телекомунікація та радіотехніка» усіх форм навчання вищих навчальних закладів

Відповідний за випуск Зуєв А. В.
Роботу до друку рекомендував Дудник О.В.
Редактор Алексєєва І. А.

План 2022 р., Поз.116
Підписано до друку 05.11.2022. формат 60×84 1/16. Папір друк. № 2.
Друк – різографія. гарнітураTimes New Roman. Розум. друк. арк.3,2.
Обл. – вид. арк. 2,7.Наклад 100 прим. Зам. № .Ціна договірна.

Видавничий центр НТУ «ХПІ». 61002, Харків, вул .. Фрунзе, 21.
Свідоцтво про реєстрацію ДК № 3657 від 24.12.2009 р.

Друкарня НТУ «ХПІ», 61002, Харків, вул. Фрунзе, 21.