П.Ф. ОЛЕКСЕНКО
В.В. КОВАЛЬ
Г.М. РОЗОРІНОВ
Г.О. СУКАЧ

ТЕОРЕТИЧНІ ОСНОВИ ЗАВАДОСТІЙКОГО КОДУВАННЯ

НАУКОВА ДУМКА

ПІДРУЧНИК

НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ФІЗИКИ НАПІВПРОВІДНИКІВ ім. В.Є. ЛАШКАРЬОВА
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ "КПІ"
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

П.Ф. ОЛЕКСЕНКО, В.В. КОВАЛЬ, Г.М. РОЗОРІНОВ, Г.О. СУКАЧ

ТЕОРЕТИЧНІ ОСНОВИ ЗАВАДОСТІЙКОГО КОДУВАННЯ

ЧАСТИНА І

Підручник для вищих навчальних закладів

За редакцією академіка НАН України В.Ф. МАЧУЛІНА

КИЇВ НАУКОВА ДУМКА 2010

ББК 32.811.4я73
Т33

Затверджено Міністерством освіти і науки України
як підручник для студентів
телекомунікаційних спеціальностей
вищих навчальних закладів
(*Лист* № 1/11-2775 *від* 2 *квітня* 2010 *року*)

Розглянуто питання взаємозв'язку між теорією та практикою завадостійкого кодування. Центральне місце займають арифметика полів Галуа, теорія лінійних блокових кодів і теорія циклічних кодів, а також практичні процедури не тільки виявлення помилок, а й визначення їх місцезнаходження. Наведено основні принципи завадостійкого кодування та побудови коригувальних кодів Хеммінга, Боуза—Чоудхурі—Хоквінгема та інших. Значну увагу приділено декодерам й алгоритмам декодування.

Для студентів вищих навчальних закладів, що повинні вміти не тільки побудувати апаратуру кодування-декодування, а й включити цю апаратуру в інформаційно-комунікаційну систему.

Рецензенти:

В.П. БОЮН, доктор технічних наук, професор,
член-кореспондент НАН України
Р.В. КОНАКОВА, доктор технічних наук, професор
Ю.Г. САВЧЕНКО, доктор технічних наук, професор

Видання здійснене за державним контрактом
на випуск наукової друкованої продукції

Науково-видавничий відділ фізико-математичної та технічної літератури

Редактор *О.А. Микитенко*

ISBN 978-966-00-0888-0

ПЕРЕДМОВА

Завдання, які постають перед зв'язком, обумовили розвиток кодів, що контролюють помилки, тому і термінологія цих кодів ґрунтується на теорії зв'язку. Проте такі коди мають багато інших застосувань: інформатика, цифрова аудіо- і відеотехніка тощо. У цифровій аудіо- і відеотехніці основна увага приділяється забезпеченню високої достовірності сигналів, що передаються. Оскільки як самі канали передачі даних, так і устаткування обробки інформації ненадійні, важливого значення набувають механізми детектування помилок. Гарантією високої достовірності передачі інформації є застосування кодів, які контролюють помилки. У разі виявлення помилки проблему, що пов'язана з нею, можна вирішити, здійснивши повторну передачу даних.

Завадостійке кодування застосовується для виявлення і(або) виправлення помилок, які можуть виникнути в дискретному сигналі під час його передачі по каналах зв'язку. Як базовий код, який піддається завадозахисному кодуванню, використовується двійковий код постійної довжини. Такий початковий (базовий) код називається ***первинним, оскільки в подальшому він піддається модифікації***.

Помилки виправляти важче, ніж їх детектувати або їм запобігати. Процедура корекції помилок припускає два суміщені процеси: виявлення помилки і визначення її місця (ідентифікація повідомлення і позиції в повідомленні). Вирішивши ці два завдання (виправлення тривіальне), потрібно інвертувати значення помилкового біта. У наземних каналах зв'язку, де вірогідність помилки невелика, зазвичай використовується метод *виявлення помилок і повторного пересилання* фрагмента, що містить у собі дефектний код. У разі виявлення помилки приймач, який не знає які біти помилкові, просто відкидає цей неправильний блок і запитує його повторну передачу. Така схема ефективна, оскільки потребує мінімум надлишкової інформації.

Для супутникових каналів зв'язку та глибоководних океанських зондів з типовими для них суттєвими затримками сигналів у каналах передачі, обмеженнями за масою та об'ємами обладнання, значним зашумленням каналу привабливими є *системи прямої корекції помилок*. Такі системи — найефективніші. Суть їх у передачі додаткової інформації разом з корисною інформацією, що дозволяє приймачу не тільки виявити помилки, а й виправити їх. Тут

використовується велика кількість відомих завадостійких кодів, які класифікуються за різними ознаками.

Предмет кодування водночас може бути і простий, і складний. Простий у тому сенсі, що його задачі легко пояснити будь-якому інженерові, а складний — оскільки вимагає знань складних розділів сучасної алгебри.

Може здатися, що досить визначити вимоги до "хорошого" коду, а потім здійснити машинний пошук за безліччю всіх можливих кодів. Проте кількість кодів така велика, що неорганізована процедура пошуку не дає бажаний результат.

За сучасних вимог науково-технічного прогресу синтез перспективних інформаційно-комунікаційних систем без завадостійкого кодування неможливий. Проте довгий час цьому перешкоджав розрив між теорією і практикою. Справа в тім, що на думку Е. Берлекемпа — патріарха алгебраїчної теорії завадостійкого кодування — розрахунково-аналітичний апарат теорії, який контролює помилки (теорія полів Галуа), майже недоступний інженеру. Вирішення цієї проблеми і було основною причиною написання даного підручника. Крім того, на сьогодні у цифровій техніці дуже широко використовується різновидність коду, що виправляє кратні помилки, — код Ріда—Соломона, поява якого, без перебільшення, стала видатною подією в теорії та техніці завадостійкого кодування. Тому ми намагалися висвітлити питання, пов'язані з теорією і практикою застосування цього та подібних кодів.

Розділ 1

ОСНОВИ ПЕРЕДАЧІ ЦИФРОВОЇ ІНФОРМАЦІЇ ТА ЇЇ КОДУВАННЯ

З'ясуємо суть термінів "сигнал" і "передача сигналу". Насамперед наведемо визначення деяких термінів.

Зв'язок — це процес переміщення в просторі і часі від відправника до отримувача матеріальних об'єктів або повідомлень щодо них.

Повідомлення — це сукупність отриманих унаслідок спостережень користувачем відомостей про стан будь-якого об'єкта, що переміщуються через середовище поширення. Матеріальний об'єкт та спостерігач є джерелом повідомлення. До складу повідомлення можуть входити адреса, службові відомості, відомості для виявлення та виправлення помилок і т. д.

Для переміщення повідомлення необхідний фізичний процес. Фізична величина (наприклад, струм у провіднику, напруженість електромагнітного поля, тиск звукових хвиль і т.п.), яка відображає повідомлення, називається ***сигналом***.

Форма переміщення повідомлення (або матеріальних об'єктів) від відправника до отримувача залежить від засобів перенесення їх у просторі та часі, від середовища поширення, а також від самих користувачів. Наприклад, поштовий зв'язок, де переміщення матеріальних об'єктів (листів, бандеролей, посилок і т.п.) здійснюється за допомогою механічних носіїв (авто, залізниця, авіа і т.п.), а також носієм може бути людина.

Розвиток сучасної цивілізації характеризується створенням глобального інформаційного простору, який забезпечує ефективну інформаційну взаємодію людей, їх доступ до світових інформаційних ресурсів. Відповідно, виникає актуальна потреба в переміщеннях дедалі зростаючих обсягів інформації.

Інформація — це адресоване повідомлення щодо зміни якого-небудь фізичного параметра. Інформація завжди пов'язана з матеріальним носієм і не може без нього існувати. У техніці такі носії називаються носіями сигналів. Наприклад, носієм сигналів може бути електричний струм з його характеристиками: амплітудою, частотою і фазою.

На великі відстані як звукові, так і відеосигнали ефективно передавати у вигляді електричних сигналів.

Сигнали електрозв'язку — це сигнали, які використовують як змінну величину, що відображає повідомлення, характеристики (параметри) електричного струму, електромагнітних хвиль. Сигнали електрозв'язку порівняно з іншими легко обробляти, вони добре передаються на великі відстані.

Електрозв'язок, телекомунікації (telecommunication) — це вид зв'язку з використанням сигналів електрозв'язку, які переміщуються по провідній, радіо, оптичній або іншій електромагнітній напрямленій системі (середовищі поширення).

Носії інформації змінюються в часі. Математична модель подання сигналу як функції часу є основоположною концепцією теоретичної радіотехніки, техніки зв'язку та інших галузей електронної техніки.

Після передачі задані електричні сигнали на приймальній стороні відновлюються в початковій формі.

Сигнали електрозв'язку залежать від форми повідомлень (первинної інформації), які вони відображають. У зв'язку з цим розрізняють види сигналів електрозв'язку і види електрозв'язку, які отримали свою назву від назви повідомлень: сигнал звукового мовлення і мовні види електрозв'язку; сигнали документального електрозв'язку і види документального електрозв'язку; телевізійний сигнал (рухомі зображення) і телевізійний вид електрозв'язку; факсимільний сигнал (нерухоме зображення) і факсимільний вид електрозв'язку і т. д.

Значення має не вид сигналів, що передаються, а забезпечення надійної та достовірної передачі інформації.

Вихідна інформація може бути джерелом сигналів як в аналоговій, так і в цифровій формі. (Існують також квантовані та дискретні сигнали.) Таким чином, є два основних типи систем зв'язку — ***аналогові*** та ***цифрові***. Кожна з цих систем може бути реалізована, наприклад, з використанням мідного проводу, оптичного волокна, а крім того, працювати взагалі за відсутності фізичного середовища (наприклад, у вакуумі).

Аналоговий сигнал — це сигнал, параметри якого лежать у безперервному просторі, тобто в просторі, що не є дискретним. Аналогові сигнали описуються неперервними функціями часу (наприклад, синуса чи косинуса), тому їх іноді називають *безперервними сигналами*. Аналоговим сигналам протиставляються дискретні (квантовані, цифрові).

Аналогові сигнали використовуються в телефонії, радіомовленні, телебаченні. Ввести такий сигнал у комп'ютер і обробити його неможливо, оскільки на будь-якому інтервалі часу він має безліч значень. Отже, для точного (без похибки) подання його значення потрібні числа нескінченної розрядності. Тому аналоговий сигнал необхідно перетворити так, щоб його можна було записати як послідовність чисел певної розрядності.

Цифровий сигнал — це дискретний сигнал, квантований за амплітудою. Цифрові сигнали є дискретні електричні або світлові імпульси. Такий сигнал має нескінченну ширину спектра, що приводить до неефективного використання всієї смуги пропускання каналу зв'язку. В цьому випадку один канал застосовують для передачі одного цифрового сигналу.

Смуга пропускання — це смуга частот, яку можна використовувати для передачі інформації по каналу зв'язку за даний інтервал часу, тобто це різниця між максимальною і мінімальною частотами. Для аналогових пристроїв вона вимірюється в герцах (Гц).

Середовище поширення інформації — це середовище, в якому сигнал переміщується від відправника до отримувача. На сьогодні можна виділити два основні класи середовищ: *провідне* та *безпровідне*. В провідному середовищі сигнали передаються по фізичній субстанції, зазвичай по міді, склу, пластику тощо. Безпровідне середовище — це таке середовище, в якому сигнал

поширюється без наявності будь-якої фізичної субстанції, навіть без повітря (у вакуумі).

Технічні характеристики каналу визначаються інформаційно-комунікаційною технологією, принципом дії пристроїв, що входять до нього, видом сигналу, властивостями і складом фізичного середовища, в якому поширюються сигнали, властивостями використаного коду.

Ефективність каналу характеризується швидкістю і достовірністю передачі інформації, надійністю роботи пристроїв і затримкою сигналу в часі.

Затримка сигналу в часі — це інтервал часу між відправкою сигналу передавачем і його прийому приймачем.

Математично канал задається безліччю допустимих повідомлень на вході і на виході та набором умовної вірогідності отримання сигналу на виході при вхідному сигналі x. Умовну вірогідність зумовлюють статистичні властивості "шумів" (або перешкод), що спотворюють сигнал у процесі передачі. Надалі будемо розглядати лише дискретні канали.

Можливість каналу передавати інформацію характеризується *пропускною здатністю або ємністю каналу* (C). Для цифрових систем передачі вводиться поняття швидкості передачі інформації, яка вимірюється в бітах на секунду (біт/с).

Для випадку каналу без шуму формула розрахунку пропускної здатності каналу має вигляд

$$C = \lim_{T \to \infty} \frac{\log_2 N(T)}{T},$$

де N — число всіх можливих сигналів за час T.

У разі використання аналогових сигналів інформація перетворюється в електричний сигнал відповідної інтенсивності (амплітуди), неперервної в часі, без переривань та розривів (наприклад, синусоїда), а цифрових електричних сигналів — у послідовність імпульсів, амплітуда яких, незмінна протягом певного часу, стрибком переходить на інший, відмінний від першого, рівень (рис. 1).

Зазначимо, що у разі цифрової передачі сигналів між передавачем та приймачем потрібна наявність фізичного середовища, наприклад, у вигляді металевих проводів або оптичних волокон; цифрові імпульси не можуть ефективно поширюватись у безпровідному середовищі.

Канал передачі даних або канал зв'язку може бути як коротким (від декількох десятків сантиметрів), так і довгим (до декількох тисяч кілометрів). Як аналогові, так і цифрові сигнали під час передачі по каналу зв'язку зазнають електромагнітного впливу, який по-різному діє на аналогові і цифрові сигнали (рис. 2).

Фізичне середовище, яким передаються корисні дані, не може бути абсолютно надійним. Воно діє певним чином на сигнал, що передається, тобто змінює його. Погіршення якості сигналу обумовлюють: *затухання* — втрата потужності сигналу між передавачем та приймачем, *шум* — випадковий, небажаний сигнал, що змінюється в часі, який при попаданні в систему передачі взаємодіє (наприклад, шляхом прямого складання) з корисним сигналом

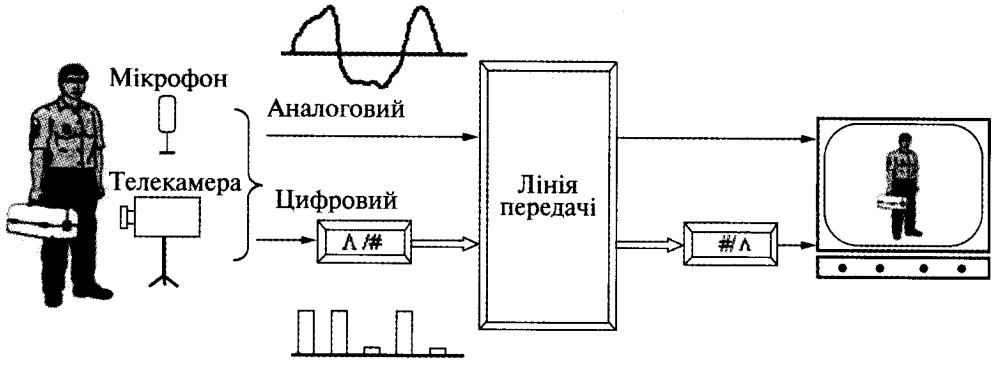


Рис. 1. Схема принципу передачі аналогових та цифрових сигналів

і спотворює його, та *розсіяння* сигналу (зазвичай, для багаточастотного сигналу, коли в міру його поширення більш низькочастотні складові випереджують більш високочастотні, що призводить до "розпливання" сигналу в часі).

Всі ці складові не несуть корисної інформації. Більш того, рівень збурень (зокрема, шуму) буває дуже високим, наприклад, у телефонних та бездротових системах зв'язку. Помилки при передачі — це реальність, яку треба обов'язково враховувати. У різних середовищах *характер перешкод* різний. Помилки можуть бути поодинокі, а можуть виникати групами, відразу декілька. Основним для систем передачі є вилучення (добування) корисної інформації із сигналу з обов'язковим обліком шуму.

Отже, у разі дії будь-якої електричної завади на ***вихідний аналоговий сигнал*** відбувається спотворення форми сигналу, оскільки у каналі передачі майже завжди присутні ті або інші перешкоди. Різниця лише в рівні перешкод та їх

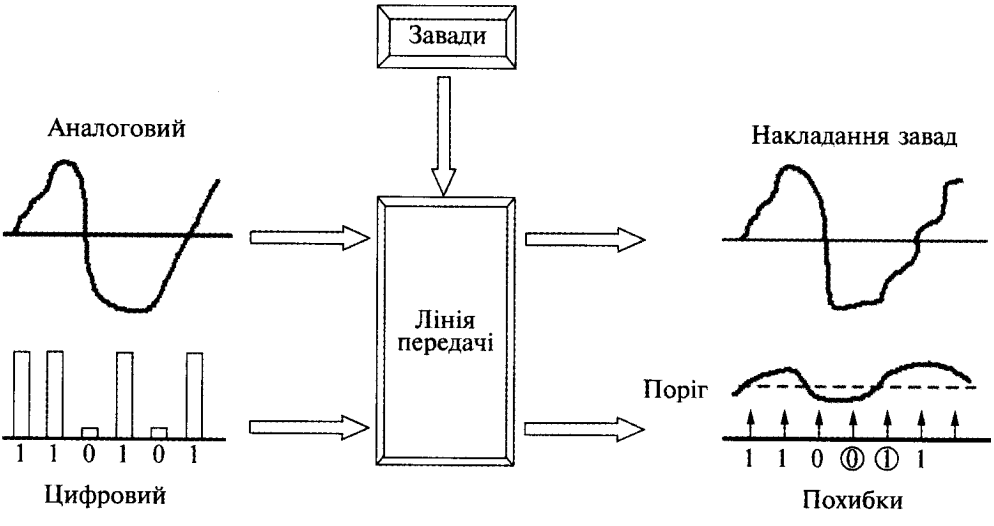


Рис. 2. Приклади впливу завад на аналоговий та цифровий сигнали

спектральному складі. Перешкоди в каналах зв'язку утворюються за різних причин, але результат їх дії на інформацію, що передається, завжди той самий — інформація втрачається (спотворюється). Проте на приймальній стороні сам електричний сигнал за допомогою технічних засобів можна з тією або іншою точністю відтворити і отримати початкову інформацію.

Для передачі інформації в обчислювальних мережах найбільш широко використовується подання інформації за допомогою двійкового алфавіту, що складається із символів "0" та "1". Цей алфавіт за кількістю символів, що входять до нього, мінімальний і технічно найбільш легко реалізується. У двійковому коді можна записувати не тільки будь-яке натуральне число, а й іншу, більш складну інформацію: тексти, картинки, фільми та звук. При дії електричної завади на ***вихідний цифровий сигнал*** інформаційний "0" може перетворитися на "1", а "1" — навпаки, на "0". Унаслідок цього на приймальній стороні можна отримати абсолютно іншу інформацію — викривлену. Тобто при передачі цифрових сигналів по каналу зв'язку неминуче виникають помилки, що вимагає розробки та застосування процедур контролю цих помилок. Виявлення помилок — один із головних факторів проектування систем передачі даних. Іншим аспектом є виправлення помилок. У разі можливості регенерації (відновлення) або регенеративної ретрансляції за допомогою цифрових сигналів можна передавати і запам'ятовувати інформацію без будь-яких спотворень. Можливість такої регенерації є основою високої надійності та достовірності цифрових систем передачі сигналів. Однак вказана висока надійність означає лише те, що можна нехтувати електромагнітним впливом на електричні сигнали в процесі їх передачі по лінії зв'язку (рис. 3).

Тобто при передачі цифрових сигналів можна нехтувати електричним впливом у каналі зв'язку, але після прийому таких електричних сигналів абсолютно необхідний контроль наявності у них помилок.

Основні принципи завадостійкого кодування. ***Кодування*** — перехід від одного способу подання інформації до іншого.

Залежно від способу формування інформації існують два варіанти: наявність можливості повторної передачі інформації і відсутність такої можливості. Наприклад, при телефонній розмові можна перепитати незрозуміле

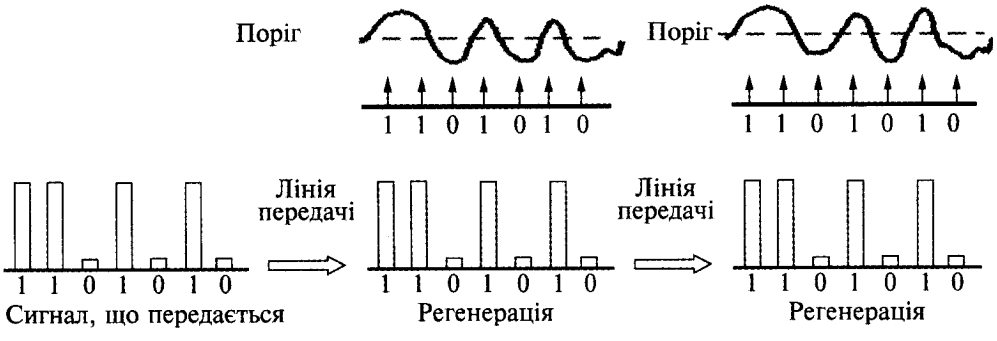


Рис. 3. Схема регенерації цифрового сигналу

слово, тобто перевірити і підтвердити отриману інформацію. Але, якщо з'єднання по телефону здійснюється за допомогою автовідповідача і інформація записується в пам'ять, то, природно, що уточнити незрозумілі місця вже неможливо. Таким чином, розрізняють два випадки:

передана інформація → є можливість повторної передачі,
→ немає можливості повторної передачі;
прийнята інформація → є можливість запиту на повторну передачу,
→ немає можливості запиту на повторну передачу.

Можливість запиту на повторну передачу даних — це звичайний випадок передачі інформації. У разі виявлення помилки в даних, що передаються, з приймальної сторони до сторони, що передає інформацію, надходить вимога на ще одну передачу цих самих даних (запит на повторну передачу кожного елемента даних, так зване дублювання). Повторна передача — це найбільш надійний метод виправлення помилок, хоча і не найбільш ефективний. Крім того, це дуже дорогий засіб виявлення помилок, особливо у разі передачі довгих повідомлень. У цьому випадку надійна передача інформації забезпечується, якщо на приймальній стороні існує можливість виявлення помилок прийнятих даних. Щоб зрозуміти принципи передачі цифрових сигналів, необхідно з'ясувати базові поняття. До базових понять передачі даних належать коди, контроль помилок (їх виявлення та виправлення) та символьна синхронізація.

Код — форма подання повідомлення, що не залежить від його фізичної суті. Це відрізняє код від сигналу, який визначає фізичне подання повідомлення (і коду) у системі зв'язку. На практиці, проте, часто пов'язують абстрактну (символьну) форму коду з фізичними сигналами, називаючи код частотним, часовим, фазовим, амплітудним. Код записують сукупністю (кодових) символів; завадостійкий код дозволяє виявляти та(або) виправляти помилки в сукупності кодових символів.

Кодова комбінація, або, коротше, ***код*** — це сукупність символів кодового алфавіту, які використовують для кодування одного символу (або однієї комбінації символів) початкового алфавіту. При цьому кодова комбінація може містити у собі один символ кодового алфавіту. ***Початковий символ*** — це символ (або комбінація символів) початкового алфавіту, якому відповідає кодова комбінація. Наприклад, оскільки в двійковій системі числення 8 = = 1000 і 8 є початковим символом, то 1000 — це кодова комбінація, або код, для числа 8. Водночас 8 — це початковий символ. Сукупність кодових комбінацій називається *кодом*. Взаємозв'язок символів (або комбінацій символів, якщо кодуються не окремі символи) початкового алфавіту та їх кодових комбінацій складає ***таблицю відповідності*** (або таблицю кодів). Зазначимо, що поняття "код" омонімічне: воно може використовуватися в сенсі кодової комбінації, і в наведеному вище. Аналогічно, поняття "кодова комбінація" синонімічне поняттю "код".

Класифікація кодів. Відомо багато завадостійких кодів, які класифікуються за різними ознаками. Завадостійкі коди можна розділити на два великі класи: ***блокові і безперервні***. При блоковому кодуванні послідовність елемен-

тарних повідомлень джерела розбивається на відрізки і кожному відрізку ставиться у відповідність визначена послідовність (блок) кодових символів, що називається, зазвичай, кодовою комбінацією. Безліч усіх кодових комбінацій, можливих при даному способі блокового кодування, і є блоковим кодом. Довжина блока може бути як постійною, так і змінною. Розрізняють рівномірні і нерівномірні блокові коди. Завадостійкі коди є, як правило, рівномірними.

Блокові коди бувають роздільними і нероздільними. До роздільних належать коди, в яких символи за їх призначенням можуть бути розділеними неінформаційними символами, що несуть інформацію щодо повідомлення, і перевірочними (контрольними) бітами. Такі коди позначаються (n, k), де n — довжина коду (кодована послідовність), k — число інформаційних символів. До нероздільних належать коди, символи яких не можна розділити за їх призначенням на інформаційні та контрольні.

Коди з постійною вагою характеризуються тим, що їх кодові комбінації містять у собі однакове число одиниць. Наприклад, код "3 з 7", в якому кожна кодова комбінація складається з трьох одиниць і чотирьох нулів (стандартний телеграфний код № 3).

Коди з постійною вагою дозволяють виявити всі помилки кратності q = = 1,..., n за винятком випадків, коли число одиниць, що перейшли в нулі, дорівнює числу нулів, що перейшли в одиниці. У повністю асиметричних каналах, у яких існує тільки один вид помилок (перетворення нулів у одиниці або одиниць у нулі), такий код дозволяє виявити всі помилки. У симетричних каналах вірогідність невиявленої помилки можна визначити як вірогідність одночасного спотворення однієї одиниці і одного нуля.

Серед роздільних кодів розрізняють ***лінійні*** та ***нелінійні*** коди. До лінійних належать коди, в яких порозрядна сума за модулем 2 будь-яких двох кодових слів також є кодовим словом. Лінійний код називається систематичним, якщо його перші k символів у будь-якій кодовій комбінації є інформаційними, інші (n–k) символів — контрольними.

Серед лінійних систематичних кодів найбільш простий код — (n, n–k), що містить у собі один контрольний символ, який дорівнює сумі за модулем 2 усіх інформаційних символів. Цей код, що отримав назву коду з перевіркою на парність, дає змогу виявити всі поєднання помилок непарної кратності. Ймовірність не виявленої помилки в першому наближенні можна визначити як ймовірність спотворення двох символів.

Підкласом лінійних кодів є *циклічні коди*. Для них характерно, що всі набори, утворені циклічною перестановкою будь-якої кодованої комбінації, є також кодовими комбінаціями. Ця властивість дає змогу значною мірою спростити кодувальні та декодувальні пристрої, особливо у разі виявлення помилок і виправлення одиночної помилки. Наприклад, коди Хеммінга, Боуза—Чоудхурі—Хоквінгема та ін.

Прикладом нелінійного коду є код Бергера, у якого контрольні символи подають двійковий запис числа одиниць у послідовності інформаційних символів. Наприклад, таким є код: 00000; 00101; 01001; 01110; 10001; 10110; 11010; 11111. Коди Бергера застосовуються в асиметричних каналах. У си-

метричних каналах вони виявляють всі одиничні помилки і деяку частину багатократних.

Безперервні коди характеризуються тим, що операції кодування і декодування проводяться над безперервною послідовністю символів без розбиття її на блоки. Серед безперервних найбільш застосовні ***коди згортки***.

Як відомо, розрізняють канали з незалежними помилками та помилками, що групуються. Відповідно, завадостійкі коди можна розбити на два класи: такі, *що виправляють незалежні помилки* та такі, *що виправляють пакети помилок*. Далі більше уваги головним чином буде приділятися кодам, що виправляють незалежні помилки. Це пояснюється тим, що хоча для виправлення пакетів помилок розроблено багато ефективних кодів, на практиці доцільніше використовувати коди, що виправляють незалежні помилки, разом з пристроєм перемежування (чергування) символів або декореляції помилок. При цьому символи кодованої комбінації не передаються один за одним, а перемішуються з символами інших кодових комбінацій. Якщо інтервал між символами, що належать одній кодовій комбінації, зробити більшим, ніж "пам'ять" каналу, то помилки в межах кодової комбінації можна вважати незалежними, що і дозволяє використовувати коди, що виправляють незалежні помилки.

Якщо повідомлення мають внутрішні кореляційні зв'язки, тобто якщо одне повідомлення деяким чином залежить від іншого, як це зазвичай буває у разі передачі текстів природним способом, то завадостійкість будь-якого коду може бути підвищена за рахунок статистичних зв'язків між повідомленнями. Якщо ж ці зв'язки або слабкі, або невідомі, або їх не можна використовувати для підвищення завадостійкості, то форма подання повідомлення повинна бути надлишковою (надмірною), зокрема, кількість символів у коді повідомлення збільшують, а між кодовими символами вводять штучні кореляційні зв'язки. Тому в деяких випадках завадостійкі коди називають надлишковими.

Надлишковий (надмірний) код — це код, який за рахунок ускладнення структури дозволяє знаходити помилки, що виникли. Розрізняють:

- коди з виявленням помилок: циклічні надмірні коди, коди з контролем парності, код контролю циклічної надмірності тощо;
- коди з виправленням помилок: код Хеммінга та ін.

Для різних перешкод у каналі існують різні за своєю структурою і надмірністю коди. Зазвичай надмірність кодів знаходиться в межах не більше 11 %. Надмірність 25 % застосовується при записі інформації на лазерні диски і в системах цифрового супутникового телебачення.

Введення надлишковості в код дає змогу, крім виявлення і виправлення помилок, підвищити енергетичну ефективність лінії зв'язку, звузити частотний спектр сигналу, що передається, скоротити час входження системи в зв'язок за рахунок підвищення завадозахисної синхронізації, поліпшити кореляційні властивості ансамблю сигналів, простими засобами реалізувати рознесений прийом. Вид завадостійкого коду залежить від структури системи зв'язку.

Кодування з виправленням помилок, по суті, є методом обробки сигналів, призначеним для збільшення надійності передачі по цифрових каналах

зв'язку. Хоча різні схеми кодування значно відрізняються одна від одної і ґрунтуються на різних математичних теоріях, усім їм властиві дві загальні властивості. Перша з них — використання надлишковості. Закодовані цифрові повідомлення завжди містять у собі додаткові, або надлишкові, символи. Ці символи використовують для того, щоб підкреслити індивідуальність кожного повідомлення. Їх завжди вибирають так, щоб зробити маловірогідною втрату повідомленням його індивідуальності через спотворення досить значної кількості символів у разі дії завад. Друга властивість полягає в усередненні шуму. Ефект усереднювання досягається завдяки тому, що надлишкові символи залежать від декількох інформаційних символів. Для розуміння процесу кодування важливо розглянути кожну з цих властивостей окремо.

Розділ 2

ВИНИКНЕННЯ ПОМИЛОК ТА ЇХ ВИЯВЛЕННЯ

Коди — це стандартний набір символів (для комп'ютера — це набір "1" та "0") для кодування тих або інших повідомлень. Розглянемо простий випадок. Нехай необхідно передати код — послідовність символів 10101, і в каналі передачі виникла помилка одного біта. Тоді, як показано на рис. 4, можливі шість варіантів прийнятого сигналу.

Виявлення помилок — це процес поточного контролю передачі цифрових даних та визначення моментів появи помилок. Методи виявлення помилок не виправляють і не ідентифікують помилкові біти, вони визначають тільки сам факт виникнення помилки. Мета процедури виявлення помилок — не допустити в системі невиявлених помилок. Найрозповсюдженішим способом виявлення помилок (контролю помилок) у переданому сигналі є метод контролю надлишковості — додавання до сигналу, що передається, додаткових спеціальних біт інформації для перевірки на наявність помилок передачі. Найпростіший спосіб створення надлишковості досягається багатократним дублюванням символів, що передаються, тобто утворенням символьних блоків: $0 \Rightarrow 00000$, $1 \Rightarrow 11111$. У разі збою рішення при дешифровці ухвалюється за більшістю однакових символів, що залишилися, в блоці.

Існує інший дуже простий, але ефективний захист інформаційного тексту від одиничного збою, який вимагає мінімальної надлишковості в один додатковий символ. Це так звана перевірка на ***парність***. До інформаційної послідовності додається додатковий розряд коду, значення якого визначається стандартним алгоритмом і залежить тільки від парності числа одиниць у коді. Найпростішим та найстарішим принципом виявлення помилок та перевірки цілісності даних, що передаються по будь-якій сигнальній шині (чи зберігаються в пам'яті), є контроль парності (чи непарності). **Парність** — спосіб контролю за передачею блоків цифрових даних за допомогою додавання контрольних бітів, при якому число одиничних бітів завжди має бути парним (або непарним — при контролі непарності).

При цьому в найпростішому випадку до послідовності інформаційних бітів додають ще один біт (контрольний) — надлишковий біт (у загальному випадку це декілька бітів — контрольна сума), що не є частиною сигналу, який передається.

Контрольна сума — це, в загальному сенсі, функція від змістовної частини кадру (інформаційного слова довжиною k), область значень якої — слова фіксованої довжини m. Ці m надлишкових бітів додаються зазвичай в кінець кадру інформації, що передається. При прийомі кодової послідовності контрольна сума обчислюється заново і порівнюється з тією, що зберігається в кадрі. Якщо вони розрізняються, то це ознака помилки в передачі корисної інформації.

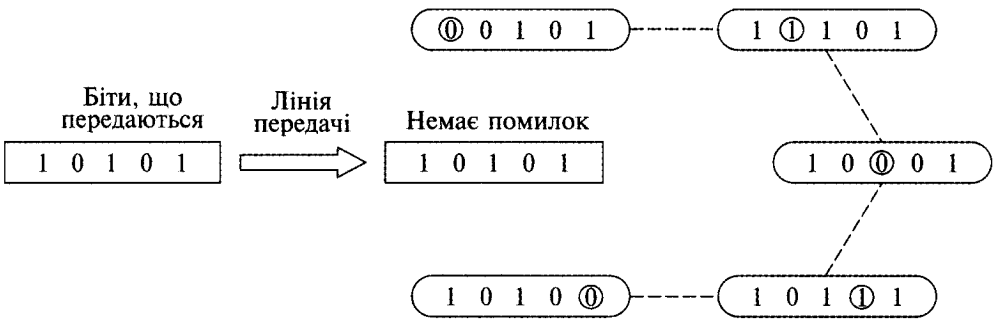


Рис. 4. Приклади варіантів сигналу при помилці в одному біті

Простим прикладом коду з виявленням однієї помилки є код з бітом парності. Конструкція його така: до початкового слова додається біт парності. Контрольний біт (контрольну суму) формують з урахуванням значень інформаційних бітів у сигналі, що передається. Наприклад, при контролі парності при парній кількості "1" у початковій послідовності інформаційних бітів значення контрольного (надлишкового) біта дорівнює "0", а при непарній кількості "1" — "1". Тобто при контролі парності загальна кількість одиниць повинна бути парною. Якщо приймачем отримано слово з непарною кількістю одиниць, то при передачі відбувся збій інформації — виникла помилка. При контролі непарності — навпаки. Загальна кількість одиниць інформаційного сигналу разом з контрольним бітом має бути непарною. Отже, потрібно формувати контрольний біт таким чином, щоб загальна кількість одиниць у сигналі, що передається, з урахуванням контрольного біта була непарною. Таким чином, допустимі слова цього коду мають непарну кількість одиниць.

Позначимо послідовність інформаційних бітів: a_4, a_3, a_2, a_1, a_0. Тоді значення контрольного біта c_0 (при контролі парності) формується так:

$$c_0 = 0 \quad \text{при} \quad a_4 + a_3 + a_2 + a_1 + a_0 \quad — \textit{при парному},$$

$$c_0 = 1 \quad \text{при} \quad a_4 + a_3 + a_2 + a_1 + a_0 \quad — \textit{при непарному}.$$

При вказаному формуванні контрольного біта загальна кількість "1" у переданій послідовності бітів a_4, a_3, a_2, a_1, a_0, c_0 буде парною.

Якщо при передачі такої послідовності бітів під впливом електричних завад з'явиться не більше однієї помилки, то вона буде виявлена при контролі загальної кількості "1" на приймальній стороні:

$$a_4 + a_3 + a_2 + a_1 + a_0 + c_0 = \begin{cases} \textit{парне} — \textit{немає помилки}, \\ \textit{непарне} — \textit{є помилка}. \end{cases}$$

Розглянемо це докладніше. Нехай послідовність інформаційних бітів має вигляд a_4, a_3, a_2, a_1, a_0, а кодова послідовність, що передається з контрольним

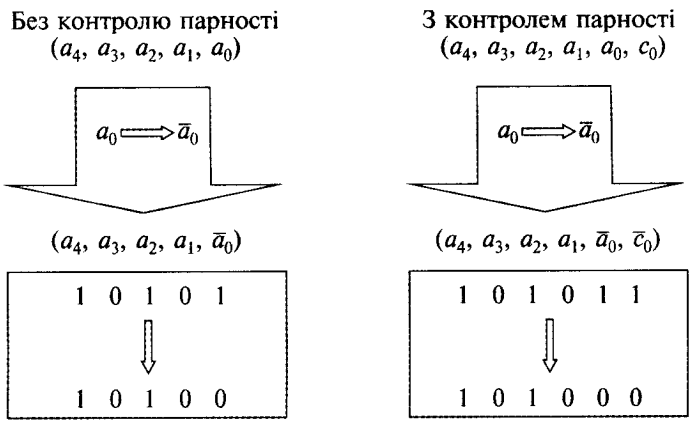


Рис. 5. Схема, за якою відбувається зміна одного інформаційного біта

(надлишковим) бітом c_0, має вигляд a_4, a_3, a_2, a_1, a_0, c_0. У разі появи помилки в одному із бітів кодової послідовності, що передається, має місце абсолютно нова інформація (рис. 5).

При передачі інформаційної послідовності з додаванням контрольного біта, у разі зміни одного інформаційного біта в усій послідовності обов'язково змінюються два біта, варіант, при якому змінюється тільки один біт кодової послідовності, що передається, неможливий. Перевага такого методу — це насамперед простота. Явні недоліки методу — необхідність затрат на запам'ятовування та зберігання зайвих бітів парності, незахищеність від подвійних помилок (а також помилкове спрацьовування при помилці в біті парності), зупинка системи навіть при непринциповій помилці (скажімо, у відеокадрі).

Отже, у разі появи помилки одного біта в процесі передачі кодова послідовність з контролем парності перетворюється на таку послідовність, яку неможливо отримати з початкової інформаційної послідовності. Завдяки цьому можна відрізнити її від послідовності, що передається, а також з'являється можливість виявлення помилки. Інакше у разі, коли виникають помилки у двох бітах. Тоді неможливо визначити, є помилка в прийнятій послідовності бітів чи ні, оскільки при зміні логічного стану даних, що передаються, ознака парності (чи непарності) не змінилася (рис. 6).

Розглянемо загальну ідею того, як за допомогою спеціального кодування можна добитися скільки завгодно високої надійності передачі. У цьому випадку вводять міжкодові інтервали.

Розвиток принципу контролю парності приводить до корегуючого коду Хеммінга, який дозволяє не тільки виявляти, а й виправляти одиничні помилки. Можливість виправлення помилки ґрунтується на повторені k разів контролю парності, але не всього слова відразу, а k певних груп його розрядів. Слово розбивається на групи так, щоб номер кожного розряду однозначно визначався за його належністю або неналежністю до цих груп.

Загальна ідея методу. На безлічі слів довжиною n визначають відстань Хеммінга (метрика Хеммінга) між двома кодовими словами, яка дорівнює кількості різних (які не збігаються між собою) бітів на відповідних позиціях

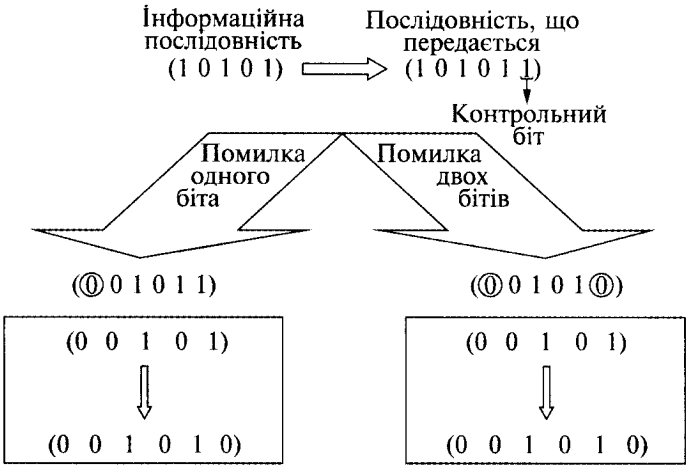


Рис. 6. Схема появи помилки в одному та у двох бітах

(див. нижче). Сенс цієї відстані той самий, що ми вкладаємо в поняття звичайної відстані. Передавач і приймач "домовляються щодо правил роботи". Якщо сигнал, що передається, не збігається з сигналом на приймальній стороні, то виникла помилка в одному або декількох бітах.

При передачі інформації по каналу зв'язку вірогідність помилки залежить від відношення сигнал/шум на вході приймача. Таким чином, при постійному рівні шуму вирішальне значення має потужність передавача. У системах супутникового, мобільного та інших типів зв'язку гостро стоїть питання економії енергії. Крім того, в певних системах зв'язку (наприклад, телефонних) необмежене підвищення потужності сигналу неможливе за технічних причин.

Оскільки завадостійке кодування дає можливість виправляти помилки, то при його застосуванні потужність передавача можна знизити, залишаючи швидкість передачі інформації незмінною. Енергетичний виграш визначається як різниця відношення сигнал/шум за наявності та відсутності процесу кодування. Неможливість запиту на повторну передачу інформації — це випадок, наприклад, супутникового зв'язку. Тут переданий сигнал необхідно відновлювати на підставі прийнятого сигналу. У разі виникнення помилки необхідно визначити, який біт є помилковим і виправити його. У цьому випадку застосовується метод прямого виправлення помилок із використанням кодів для виправлення помилок.

Вибір між цими двома методами виправлення помилок потрібно робити, виходячи з конкретних вимог до систем і особливостей їх застосування.

Розділ 3

МІЖКОДОВА ВІДСТАНЬ І МОЖЛИВІСТЬ ВИПРАВЛЕННЯ ПОМИЛОК

Розподіл кодів, що отримується у випадку, наприклад, однієї помилки в кодованому слові, називається ***областю розподілу*** (див. рис. 4). Використовуючи таке поняття, як відстань між кодованими словами, можна легко проаналізувати можливість виявлення і виправлення помилок.

Міжкодова відстань — це найкоротший шлях від одного кодованого слова до іншого. Наприклад, на рис. 7, *а* видно, що кодоване слово, яке складається з одних інформаційних бітів, при помилковій передачі одного біта, як і раніше, залишається кодованим словом. Тут міжкодова відстань дорівнює 1. На рис. 7, *б* наведено кодоване слово з контрольним бітом c_0. У цьому випадку інше кодоване слово буде мати місце лише при зміні двох бітів, тобто міжкодова відстань дорівнює 2.

Згідно з визначенням, міжкодова відстань — число найкоротших шляхів для отримання іншого коду. Можна вважати, що n-розрядний код (a_{n-1}, a_{n-2},..., a_1, a_0) відповідає вершинам одиничного гексаедра n-вимірного евклідового простору, а відстань між двома вершинами дорівнює мінімальній кількості ребер, що з'єднують ці дві вершини.

Хеммінгом було запропоновано означення міжкодової відстані, або ***відстані Хеммінга***, — це міра (точніше ***метрика***) відмінності об'єктів однакової розмірності. Сенс цієї відстані той самий, що ми вкладаємо в поняття звичайної відстані. Іншими словами, відстанню Хеммінга між двома двійковими послідовностями (векторами) називається число позицій, в яких вони різні. Так, відстань Хеммінга між числовими послідовностями (векторами) 00001 і 10011 дорівнює 2 (за кількістю бітів, що розрізняються в цих послідовностях).

Оскільки n-вимірний простір важко зобразити графічно, то зазвичай розглядають тривимірний простір, який зображується за допомогою правильного шестигранника (гексаедра або куба) з одиничними координатами по трьох взаємно перпендикулярних осях (рис. 8).

З рис. 8 видно, що кількість різних координат між двома ближніми сусідніми вершинами (наприклад, вздовж осі x між вершинами (000) і (100)) дорівнює 1. Отже, відстань Хеммінга між цими ж вершинами (наприклад, вздовж тієї ж осі x) дорівнює 1, тобто перехід із точки (000) до точки (100) здійснюється вздовж 1 ребра куба. Водночас кількість різних координат між двома вершинами від (000) до (011) становить 2. Це означає, що відстань між цими вершинами (між кодами (000) та (011)) дорівнює 2, тобто перехід із точки (000) до точки (011) здійснюється, як мінімум, уздовж 2-х ребер куба. Якщо порівняти вершини (100) і (011), то з'ясується, що кількість різних координат при цьому дорівнює 3, тобто відстань між двома цими кодами — 3.

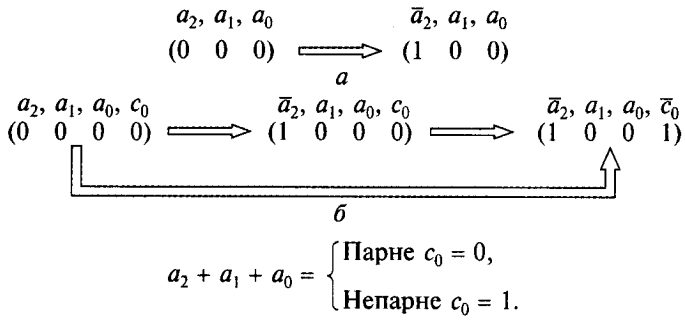


Рис. 7. Приклад міжкодової відстані

Щоб пройти від (100) до (011), потрібно, щонайменше, пройти вздовж трьох ребер куба. Узагальнюючи, можна стверджувати, що, якщо порівнювати координати двох будь-яких кодованих слів, то найменшу відстань Хеммінга для них можна визначити кількістю різних координат.

Мінімальна відстань Хеммінга $d_{\min}$ є важливою характеристикою лінійного блокового коду. Це відстань, на якій розташовані два різні коди, вона визначає іншу, не менш важливу характеристику — **корегувальну здатність**:

$$t = \left[\frac{d_{\min} - 1}{2}\right]$$ округляємо "вниз", так щоб $2t < d_{\min}$.

Корегувальна здатність визначає, скільки помилок передачі коду (типу зміни "1" на "0" і навпаки) можна *гарантовано* виправити. Тобто навколо кожного коду A маємо t-окіл A_t, який складається зі всіх можливих варіантів передачі коду A з числом помилок (зміни "1" на "0" і навпаки), що не перевищує t (рис. 9). Ніякі два околи двох будь-яких кодів не перетинаються один з одним, оскільки відстань між кодами (тобто центрами цих околів) завжди перевищує їх два радіуси $d_H(A, B) \geq d_{\min} > 2t$. Таким чином, отримавши спотворений код з A, декодер приймає рішення, яким був початковий код A, виправляючи тим самим не більше ніж t помилок.

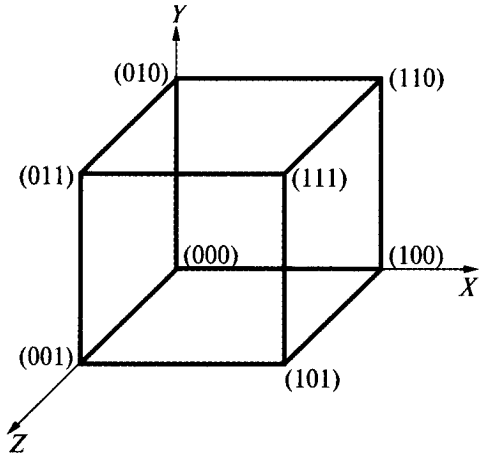


Рис. 8. Тривимірний одиничний гексаедр і його вершини

Виявлювальна і корегувальна здатності кодів залежать від кодової відстані між словами, що чисельно дорівнює мінімальній кількості помилок, яка може перетворити одне слово на інше. Наприклад, у семирозрядних двійкових кодів: 0111100; 0100101; 0010110, перша група (слово) відрізняється від другої в трьох розрядах, друга від третьої — в чотирьох розрядах, перша від третьої — в трьох розрядах. Мінімальна відстань d

між цими словами дорівнює 3. Якщо в першому слові відбудеться 3 помилки, то воно може перетворитися або на друге, або на третє слово. У разі декодування така помилка не буде виявлена. Максимальне число помилок, яке в даному випадку може бути виявлене, дорівнює 2. Якщо в першому слові відбулася помилка в другому розряді, то отримане слово відрізняється від другого в чотирьох розрядах, від третього — в двох розрядах, від першого — в одному розряді. За максимальної правдоподібності методу при декодуванні дійшли висновку, що, найімовірніше, передавалося перше слово. Для правильного декодування необхідно, щоб максимальна кількість помилок у слові, що передалося, перетворювала його на слово, яке б відрізнялося від остаточного в найменшому числі розрядів. Щоб виправляти всі комбінації з t помилок, необхідно і достатньо, щоб $d_{\min} > 2t + 1$.

У загальному випадку, якщо розглядати n-розрядну послідовність, то у двох кодованих слів $a = (a_{n-1}, a_{n-2},..., a_1, a_0)$ і $b = (b_{n-1}, b_{n-2},..., b_1, b_0)$ відстань Хеммінга d визначається за формулою

$$d(a,b) = c_{n-1} + c_{n-2} + ... + c_1 + c_0 ,$$

де $c_i = \begin{cases} 1, & \text{якщо } a_i \neq b_i, \\ 0, & \text{якщо } a_i = b_i . \end{cases}$

У разі одиничної помилки n–розрядного кодованого слова має місце область розподілу для кодових слів з відстанню Хеммінга, що дорівнює 1, а при подвійній помилці — 2.

Розглянемо, наприклад, які двійкові коди відповідають інформації A = = 1111 і B = 0000 при помилці в одному біті. Області розподілу при одиничній помилці зображені на рис. 9, де показано, що при одиничній помилці області розподілу кодованих слів A і B не мають загальних ділянок, тобто повністю відокремлені одна від одної. Отже, у разі таких кодів одиничні помилки можна повністю виправити.

Розглянемо випадок подвійної помилки у тих самих словах A і B (рис. 10). У цьому разі відбувається накладення області розподілу кодованого слова A на область розподілу кодованого слова B. При цьому неможливо визначити, яке з кодованих слів має дві помилки: чи слово A, чи слово B. Отже, у разі подвійної помилки можна лише встановити появу помилок. Регулярні методи побудови кодів, що коректують помилки, які запропонував Хемінг, мали фундаментальне значення. Вони продемонстрували інженерам практичну можливість досягнення тих меж, на які вказували закони теорії інформації. Ці коди знайшли практичне застосування при створенні комп'ютерних систем. Метод Хеммінга дає змогу "на льоту" виправляти одиничні і виявляти подвійні помилки.

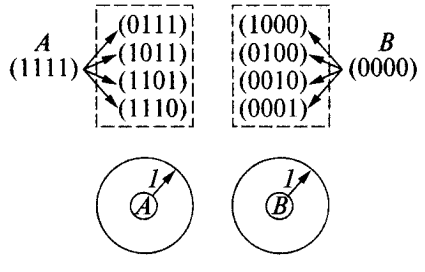


Рис. 9. Схематичне зображення областей розподілу для кодованих слів A і B при одиничних помилках

На рис. 10 видно, що відстань Хеммінга для слів A і B дорівнює 4. Таким чином, за довжиною міжкодової відстані можна зробити висновок щодо можливості виявлення помилок і їх виправлення. Але говорячи про міжкодові відстані, слід враховувати, що вони різні для кожного конкретного кодованого слова. Якщо визначити відстані між всіма кодованими словами, то серед них будуть і найкоротші відстані. Такі відстані називають мінімальними міжкодовими відстанями (ММКВ). При ММКВ, що дорівнює 3, області радіусом 1 і кодованими словами в центрі будуть віддалені одна від одної на одиницю (рис. 11). Це означає, що при таких ММКВ можна виправляти всі одиничні помилки. Операцію виправлення одиничних помилок скорочено позначають SEC (single error correction).

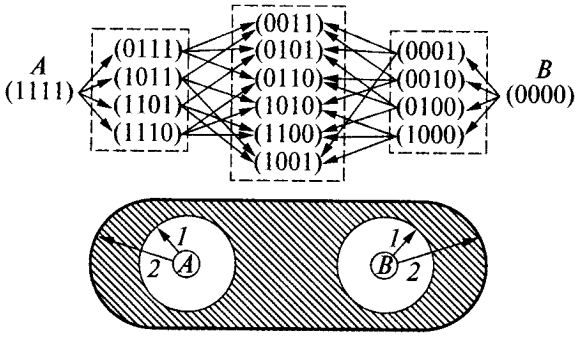


Рис. 10. Схематичне зображення області розподілу при подвійній помилці в кодованих словах A і B

Розглянемо ММКВ і питання виправлення декількох помилок. Якщо ММКВ дорівнює 4, то можливе виявлення подвійної помилки і виправлення одиничної помилки (рис. 12).

Операцію виявлення подвійної помилки скорочено позначають DED (double error detection). Таким чином, код з ММКВ, що дорівнює 4, є кодом SEC.DED. У разі коли ММКВ дорівнює 5, існують дві можливості виправлення помилок — SEC і DEC (double error correction) (рис.13). Далі коли ММКВ дорівнює 5 і більше, можливі DEC або SEC.TED (triple error detection — виявлення потрійної помилки).

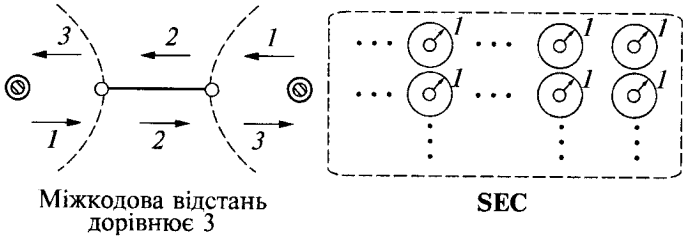


Рис. 11. Схема визначення міжкодової відстані та код виправлення одиничної помилки

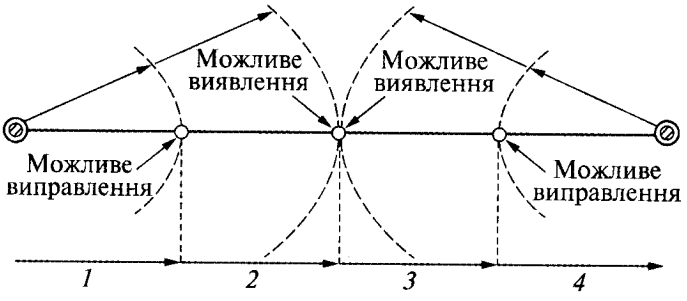


Рис. 12. Схема визначення міжкодової відстані для коду SEC.DED

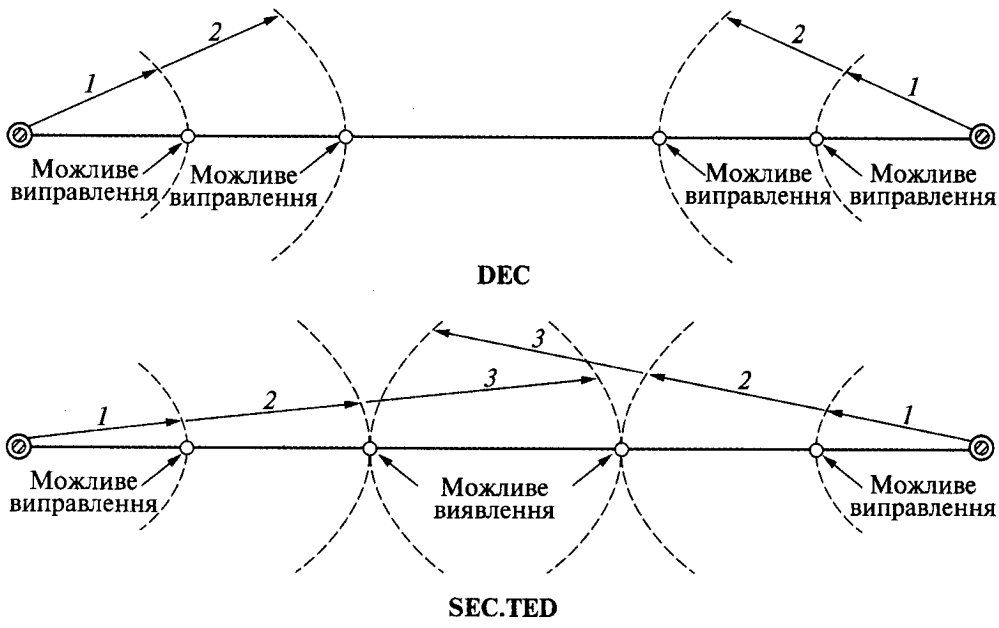


Рис. 13. Схема, за якою відбувається виправлення помилок коду з визначеною міжкодовою відстанню

Це може відбуватися таким чином.

1. Якщо ММКВ перевищує $2t + 1$, то можна виправити всі помилки цього коду від однієї до t (див. рис. 13). Такий код називають кодом з виправленням t-кратної помилки.

2. Якщо ММКВ дорівнює або перевищує $2t_1 + t_2 + 1$, то можна виправити всі помилки від однієї до t_1 і виявити до $t_1 + t_2$ помилок. Такий код називають кодом з виправленням t_1 помилки і виявленням $t_1 + t_2$ помилки.

Розділ 4

КОД ХЕММІНГА

Коди Хеммінга — найвідоміші і, ймовірно, найперші з тих кодів, що самоконтролюються. Вони були запропоновані Р. Хеммінгом (1915—1998 рр.) у 1946 р., який використав для їх побудови двійкову систему числення.

До винаходу кодів Хеммінга підштовхнули незручності в роботі з перфокартами на релейній обчислювальній машині Bell Model V. На ній він працював у дні, коли не було операторів, і йому самому доводилося вводити інформацію. Хоч як там було, але він запропонував коди, здатні коректувати помилки в каналах зв'язку, у тому числі й у магістралях передачі даних у комп'ютерах, передусім між процесором і пам'яттю.

Коди Хеммінга є підтвердженням практичної реалізації можливостей, на які вказує теорема Шеннона. Для побудови оптимального коду, що коректується, досить приписати до кожного слова один додатковий (контрольний) двійковий розряд і вибрати цифру з цього розряду так, щоб загальна кількість одиниць у зображенні будь-якого числа була, наприклад, парною. Одинична помилка в будь-якому з розрядів переданого слова (зокрема, можливо, і в контрольному розряді) змінить парність загальної кількості одиниць. Лічильник за модулем 2, що підраховує кількість одиниць, які містяться серед двійкових цифр числа, може давати сигнал щодо наявності помилок. У цьому разі немає ніяких вказівок, в якому саме розряді відбулася помилка і, отже, немає можливості її виправити. Залишаються непоміченими також помилки, що виникають одночасно в двох, у чотирьох або взагалі в парній кількості розрядів. Неможливо виправити подвійні й багатократні бітові помилки та пакети помилок. Відсутня також ідентифікація помилок у своїх (власних) бітах коду.

Проте подвійні, а тим більше чотирикратні помилки вважаються маловірогідними. Для побудови коду, що самокорегується, розрахованого на виправлення одиничних помилок, одного контрольного розряду недостатньо. Їх мінімальна кількість визначається за допомогою граничної формули Хеммінга.

Отже, код Хеммінга — це код з виправленням помилок. Він використовується для виправлення помилок у синхронних потоках даних. Розглянемо п'ятисимвольну двійкову послідовність: a_4, a_3, a_2, a_1, a_0. Контроль парності для неї полягає в додаванні ще одного контрольного біта c_0. З'ясуємо, яким чином розраховується значення контрольного біта. Для того щоб дізнатися, чи є деяке число парним або непарним, його ділять на 2 і визначають, чому дорівнює залишок. Позначимо деяке число N. Тоді $N : 2 = Q$, а залишок від ділення — R ($0 \le R < 2$).

Якщо залишок $R = 0$, то число N — парне, якщо $R = 1$, то число N — непарне. Іншими словами, якщо кратні 2 числа замінювати на 0, то можна

легко визначити, парним чи непарним є число N. Ця умова перевірки числа за модулем 2 (mod 2) записується так:

$$N = 2Q + R = R \pmod 2.$$

Запам'ятаємо, що перевірка за mod 2 полягає в заміні 2 на 0. Значення контрольного біта c_0 за mod 2 визначається за формулою

$$c_0 = a_4 + a_3 + a_2 + a_1 + a_0 \pmod 2.$$

Таким чином, якщо результат суми кратний 2 (парне число), то $c_0 = 0$, а якщо ні (непарне число), то $c_0 = 1$.

Контроль парності на приймальній стороні полягає в перевірці за mod 2 усіх прийнятих бітів:

$$S = a_4 + a_3 + a_2 + a_1 + a_0 + c_0 \pmod 2.$$

Якщо $S = 0$, то помилки немає, а якщо $S = 1$, то це вказує на наявність помилки.

Контроль парності при використанні коду SED дозволяє отримати код з виправленням помилки шляхом збільшення числа контрольних бітів.

Щоб виявити, а тим більше виправити багатократні помилки, необхідно відмовитися від оптимальності коду. Для цього потрібно збільшити на декілька додаткових двійкових символів кількість контрольних бітів, що додаються до інформаційної послідовності, що складається з k розрядів, тобто навмисно ввести деяку надмірну надлишковість, яка змогла б допомогти виявити або виправити всі помилки. Необхідне число двійкових символів, що додатково вводяться, позначимо m, тоді довжина кодової послідовності — $n = k + m$ розрядів.

Для виправлення одиничної помилки в n-розрядній кодовій послідовності потрібно виявити помилковий біт, тобто визначити положення того біта, в якому відбулася помилка.

Кількість станів, що описуються за допомогою m контрольних бітів, становить 2^m. Якщо

$$2^m \geq n + 1, \tag{1}$$

то забезпечується можливість виправлення одиничних помилок такого коду (рис. 14).

Оскільки $n = k + m$, то із співвідношення (1) отримаємо

$$2^m - m \geq k + 1. \tag{2}$$

Приймемо, що унаслідок завад (випадкових або навмисних) один (або жодний) з $k + m$ двійкових символів може перетворюватися з "1" на "0" або, навпаки, з "0" на "1". Нехай $n = k + m$ події, унаслідок яких помилка взагалі не відбудеться, відбудеться на рівні першого, другого, ..., $(k + m)$-го символу кодового набору, рівноймовірні. Таким чином, якщо кількість m контрольних бітів щодо k інформаційних бітів вибрати такою, щоб вона задовольняла співвідношення (2), то можливе виправлення одиничних помилок.

Рис. 14. Умова, за якою можна виправляти одиничні помилки

Якщо події рівноймовірні, кодове слово складається з n бітів, число контрольних бітів дорівнює m, то даний код буде кодом SEC у тому випадку, коли

$$n \le 2^m - 1. \tag{3}$$

Скориставшись рівністю $n = k + m$, перепишемо формулу (3) у вигляді

$$n \le 2^m - 1 = 2^{n-k} - 1 = \frac{2^n}{2^k} - 1. \tag{4}$$

Тоді

$$n + 1 \le \frac{2^n}{2^k}, \tag{5}$$

$$2^k \le \frac{2^n}{n+1}. \tag{6}$$

Нерівність (6) визначає граничне число інформаційних бітів k, при якому ще можливе виправлення одиничних помилок двійкового коду, що складається з n бітів. Ця формула — ***гранична формула Хеммінга.***

Таким чином, для виявлення самого факту наявності одиничної помилки і встановлення її позиції необхідно сформувати інформацію в кількості не менше $n = k + m$ бітів. Джерелом цієї інформації є лише додатково введені m двійкових символів, оскільки інші k символів унаслідок оптимальності кодування до межі зайняті описом самого тексту. Зазначимо, що m двійкових символів у кращому разі можуть містити у собі інформацію в m бітів.

При формуванні коду, що відповідає граничній формулі Хеммінга, з урахуванням рівності

$$\begin{cases} n = 2^m - 1, \\ n = m + k, \end{cases} \tag{7}$$

Таблиця 1. Обмеження Хеммінга

Число контрольних бітів, m	Загальне число бітів, n	Число інформаційних бітів, k
1	1	0
2	3	1
3	7	4
4	15	11
5	31	26
6	63	57
7	127	120
8	255	247
9	511	502
10	1023	1013
11	2047	2036

можна визначити необхідне число контрольних бітів з урахуванням обмежень на співвідношення між m, n та k (табл. 1). Біти Хеммінга можуть розміщатись після інформаційних біт, перед ними або в проміжках між ними.

На підставі співвідношень $n' \le n$, $k' \le k$, $n' = m + k'$ і відомого числа контрольних бітів m можна сформувати код SEC. Такий код, що отримано з урахуванням результатів табл.1, називають ***кодом Хеммінга (n, k)***. Зазвичай код Хеммінга характеризується двома цілими числами, наприклад, (11,7), що означає: при передачі 7-бітового інформаційного коду використовують 4 контрольних біта (7 + 4 = 11). У цьому разі передбачається, що в одному із бітів є помилка, а в двох або більше бітах істотно менш вірогідна. З урахуванням цього виправлення помилки здійснюється з певною вірогідністю.

Як приклад розглянемо код Хеммінга (7,4). Для нього $n = 7$, $k = 4$ і $m = 3$. Він є кодом SEC вигляду a_3, a_2, a_1, a_0, c_2, c_1, c_0.

Оскільки використовується 3 контрольних біта, кількість станів, що перевіряються, становить $2^3 = 8$ (табл. 2).

При такому розподілі перевірних станів вирази для контролю парності на приймальній стороні мають вигляд

$$S_1 = 1 \cdot a_3 + 0 \cdot a_2 + 1 \cdot a_1 + 0 \cdot a_0 + 1 \cdot c_2 + 0 \cdot c_1 + 1 \cdot c_0,$$
$$S_2 = 0 \cdot a_3 + 1 \cdot a_2 + 1 \cdot a_1 + 0 \cdot a_0 + 0 \cdot c_2 + 1 \cdot c_1 + 1 \cdot c_0,$$
$$S_4 = 0 \cdot a_3 + 0 \cdot a_2 + 0 \cdot a_1 + 1 \cdot a_0 + 1 \cdot c_2 + 1 \cdot c_1 + 1 \cdot c_0,$$

тобто

$$S_1 = a_3 + a_1 + c_2 + c_0 \pmod 2, \quad (8)$$
$$S_2 = a_2 + a_1 + c_1 + c_0 \pmod 2, \quad (9)$$
$$S_4 = a_0 + c_2 + c_1 + c_0 \pmod 2. \quad (10)$$

Таким чином, шляхом розрахунку факторів, що контролюють парність величин S_1, S_2 і S_4, можна визначити у разі наявності одиничної помилки її місцезнаходження, а потім і виправити її операцією інверсії.

Розглянемо тепер як краще сформувати контрольні біти c_2, c_1, c_0 на передавальній стороні. Оскільки тут помилки відсутні, то при контролі парності величини S_1, S_2, S_4 дорівнюють 0. Тобто при вказаному контролі парності вирази (8), (9) і (10) можна записати так:

$$0 = a_3 + a_1 + c_2 + c_0 \pmod 2, \quad (11)$$
$$0 = a_2 + a_1 + c_1 + c_0 \pmod 2, \quad (12)$$
$$0 = a_0 + c_2 + c_1 + c_0 \pmod 2. \quad (13)$$

Таблиця 2. Відповідність помилкових станів, що перевіряються

Стани, що перевіряються			Помилкові стани
S_4	S_2	S_1	
0	0	0	Помилки немає
0	0	1	Помилка першого біта (a_3)
0	1	0	Помилка другого біта (a_2)
0	1	1	Помилка третього біта (a_1)
1	0	0	Помилка четвертого біта (a_0)
1	0	1	Помилка п'ятого біта (c_2)
1	1	0	Помилка шостого біта (c_1)
1	1	1	Помилка сьомого біта (c_0)

Тоді можна визначити параметри контрольних бітів c_2, c_1, c_0. Для цього спочатку знайдемо суму виразів (11)—(13):

$$0 = a_3 + a_1 + c_2 + c_0 + a_2 + a_1 + c_1 + c_0 + a_0 + c_2 + c_1 + c_0 = \\ = a_3 + a_2 + 2a_1 + a_0 + 2c_2 + 2c_1 + 3c_0 = a_3 + a_2 + a_0 + c_0 \pmod 2. \quad (14)$$

Додаючи до обох сторін рівності (14) величину c_0, отримуємо співвідношення для визначення контрольного біта c_0:

$$c_0 = a_3 + a_2 + a_1 + 2c_0 = a_3 + a_2 + a_0 \pmod 2. \quad (15)$$

Унаслідок додавання рівностей (11) та (14) одержимо

$$0 = a_3 + a_1 + c_2 + c_0 + a_3 + a_2 + a_0 + c_0 = a_2 + a_1 + a_0 + c_2 \pmod 2. \quad (16)$$

Додавши до обох сторін рівності (16) величину фактора c_2, отримаємо формулу для визначення значення контрольного біта c_2:

$$c_2 = a_2 + a_1 + a_0 \pmod 2. \quad (17)$$

При додаванні рівностей (12) та (14) маємо

$$0 = a_2 + a_1 + c_1 + c_0 + a_3 + a_2 + a_0 + c_0 = a_3 + a_1 + a_0 + c_1 \pmod 2. \quad (18)$$

Додавши до обох сторін рівності (18) величину c_1, отримаємо формулу для визначення значення контрольного біта c_1:

$$c_1 = a_3 + a_1 + a_0 \pmod 2. \quad (19)$$

Таким чином, на стороні, що передає, контрольні біти c_2, c_1, c_0 слід формувати на підставі формул (15), (17) і (19), тобто використовувати систему рівнянь:

$$\begin{cases} c_2 = a_2 + a_1 + a_0 \pmod 2, \\ c_1 = a_3 + a_1 + a_0 \pmod 2, \\ c_0 = a_3 + a_2 + a_0 \pmod 2. \end{cases} \quad (20)$$

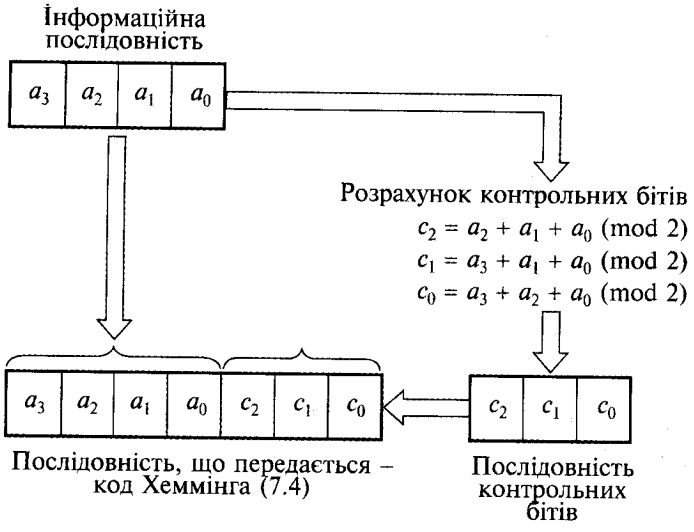


Рис. 15. Схема формування коду Хеммінга (7,4)

Контрольні біти додають до інформаційних, унаслідок чого формується кодова послідовність бітів, що передається по каналу передачі даних (рис.15).

На приймальній стороні на підставі прийнятої кодової послідовності a_3, a_2, a_1, a_0, c_2, c_1, c_0 за допомогою рівності контролю парності визначаються фактори контролю парності:

$$\begin{cases} S_4 = a_0 + c_2 + c_1 + c_0 \pmod 2, \\ S_2 = a_2 + a_1 + c_1 + c_0 \pmod 2, \\ S_1 = a_3 + a_1 + c_2 + c_0 \pmod 2. \end{cases} \qquad (21)$$

Якщо в прийнятій послідовності помилок немає, то величини S_4, S_2, S_1 дорівнюють "0". Наприклад,

$$S_1 = a_3 + a_1 + c_2 + c_0 = a_3 + a_1 + c_2 + c_0 = a_3 + a_1 + (a_2 + a_1 + a_0) + (a_3 + a_2 + a_0) =$$
$$= 2a_3 + 2a_2 + 2a_1 + 2a_0 = 0 \pmod 2.$$

Аналогічно обчислюються величини $S_2 = 0 \pmod 2$, та $S_4 = 0 \pmod 2$.

У разі появи помилки в процесі передачі даних відбувається інверсія біта з "0" на "1" або навпаки, — з "1" на "0". Таку інверсію можна виявити, додавши "1" при розрахунку за mod 2. Тобто у разі помилки біта a, його перетворюють на біт $a + 1 \pmod 2$:

a	$a + 1 \pmod 2$
0	0 + 1 = 1
1	1 + 1 = 0

Отже, помилковий біт визначається так: $\overline{a} = a + 1 \pmod 2$.

На приймальній стороні при появі одиничної помилки з семи бітів один біт інвертується. Наприклад, у разі помилки першого біта для визначення a_3 одержуємо: $\overline{a}_3 = a_3 + 1 \pmod 2$. Розрахунки з використанням системи рівнянь (21) дозволяють отримати для визначення параметрів контролю парності такі співвідношення:

$$S_4 = a_0 + c_2 + c_1 + c_0 = 0 \pmod 2,$$
$$S_2 = a_2 + a_1 + c_1 + c_0 = 0 \pmod 2, \qquad (21a)$$
$$S_1 = \overline{a}_3 + a_1 + c_2 + c_0 = (a_3 + 1) + a_1 + c_2 + c_0 = 1 + a_3 + a_1 + c_2 + c_0 = 1 \pmod 2.$$

Із системи рівнянь (21а) випливає, що оскільки a_3 входить тільки в формулу для S_1, то у разі помилки a_3 маємо $S_4 = 0$, $S_2 = 0$, $S_1 = 1$. За станом контролю (S_4, S_2, S_1) = (0 0 1) стає зрозуміло, що відбулася помилка в першому біті.

Аналогічно, у разі помилки, що виникла у другому біті (a_2), отримуємо, (S_4, S_2, S_1) = (0 1 0), у третьому біті (a_1) — (S_4, S_2, S_1) = (0 1 1), у четвертому біті (a_0) — (S_4, S_2, S_1) = (1 0 0), у п'ятому біті (c_2) — (S_4, S_2, S_1) = (1 0 1), у шостому біті (c_1) — (S_4, S_2, S_1) = (1 1 0), у сьомому біті (c_0) — (S_4, S_2, S_1) = (1 1 1).

Після визначення положення помилкового біта для виправлення помилки досить інвертувати цей біт, додавши "1" за mod 2. Наприклад, для сьомого біта отримаємо: $\overline{c}_0 + 1 = c_0 \pmod 2$.

Процедура виправлення помилок на приймальній стороні з використанням коду Хеммінга (7,4) наведена на рис. 16.

Кодоване слово в коді Хеммінга (7,4) має такий вигляд:

$$\underbrace{a_3\ a_2\ a_1\ a_0}_{\text{Інформаційні біти}} \quad \underbrace{c_2\ c_1\ c_0}_{\text{Контрольні біти}}$$

$$c_2 = a_2 + a_1 + a_0 \pmod 2,$$
$$c_1 = a_3 + a_1 + a_0 \pmod 2,$$
$$c_0 = a_3 + a_2 + a_0 \pmod 2.$$

Наприклад, у разі помилки у першому біті інформації отримуємо інверсію a_3, унаслідок якої він перетворюється на $\overline{a}_3$. Оскільки a_3 входить у контрольні біти, то вони також інвертуються і перетворюються на $\overline{c}_1$ і $\overline{c}_0$. Всього одержуємо три інвертованих біти — a_3, c_1, c_0, тому міжкодова відстань дорівнює 3.

У разі помилки у четвертому біті інформації маємо інверсію a_0; він перетворюється на $\overline{a}_0$. Отже, маємо чотири інвертованих біти — a_0, c_2, c_1, c_0, міжкодова відстань становить 4. При інверсії всіх інформаційних бітів a_3, a_2, a_1, a_0 міжкодова відстань досягає максимального значення і дорівнює 7.

У разі використання простого коду Хеммінга можна виправляти одиничну помилку. Однак за рахунок збільшення кількості контрольних бітів можна формувати складніші коди, які виправляють декілька помилок. Наприклад, для

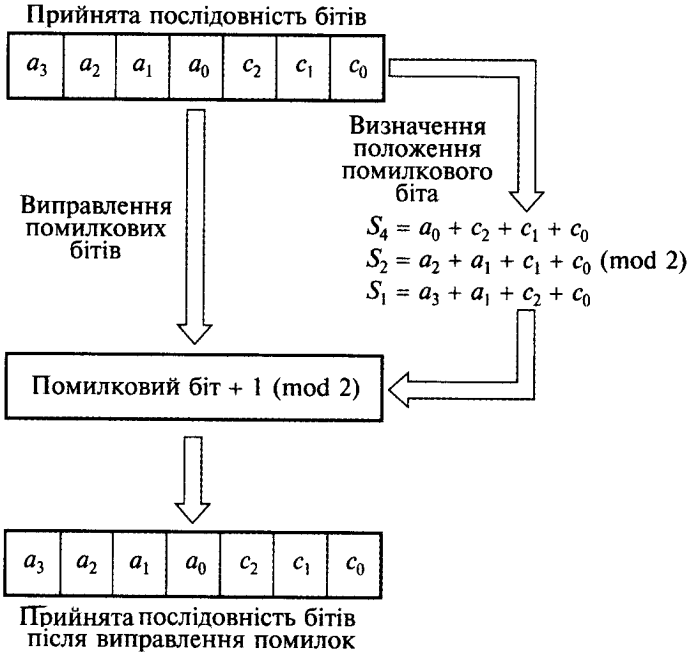


Рис. 16. Схема відновлення коду Хеммінга (7, 4)

виправлення подвійної помилки число контрольних бітів визначається таким чином: *число безпомилкових станів + число станів з одиничною помилкою + число станів з подвійною помилкою = число станів, які можна описати контрольними бітами.*

Розділ 5

СКІНЧЕННІ (КІНЦЕВІ) ПОЛЯ І РОЗРАХУНКИ ЗА МОДУЛЕМ

Особливість кодів з виправленням помилок — це можливість опису місцеположення конкретного біта інформації за допомогою многочлена.

Многочлен — сума одночленів. Якщо всі одночлени в многочлені зведені до стандартного вигляду, то говорять, що це ***многочлен стандартного вигляду***. Вираз алгебри, що не містить у собі операцій ділення і добування кореня (цілий вираз), завжди може бути зведений до многочлена стандартного вигляду. Степенем многочлена є найвищий із степенів його доданків.

Іноді виникає проблема перетворити многочлен так, щоб він був записаний у вигляді добутку декількох співмножників. Таке тотожне перетворення називається ***розкладанням многочлена на множники***. У цьому випадку говорять, що многочлен ділиться на кожен із цих співмножників. Розкладаючи многочлен на множники, застосовують три основні прийоми: винесення множника за дужку, використання формул скороченого множення та спосіб групування.

Для опису фізичних явищ часто використовують періодичні функції типу $\sin \omega t$. Тригонометрична функція $\sin(\omega t + 2n\pi) = \sin \omega t$ (n — ціле число) є гармонічною функцією з періодом 2π. Періодичність притаманна не тільки різноманітним фізичним явищам, а й повсякденню. Наприклад, кожен тиждень має сім однакових днів, а рік — 12 однакових місяців, які періодично змінюють один одного (рис. 17).

Основні принципи кодування у цьому разі такі. Кодоване в двійковому коді n-розрядне число записується у вигляді полінома $(n-1)$-го степеня деякої змінної x, причому коефіцієнтами полінома є двійкові знаки відповідних розрядів. Запис, читання і передача кодових комбінацій у циклічному коді здійснюються в певній послідовності (за спадним чи зростаючим степенем змінної x).

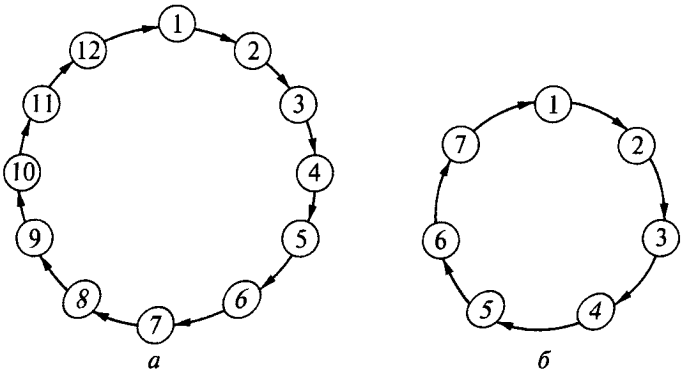


Рис. 17. Приклади періодичності в повсякденні: *а* — періодична зміна місяців; *б* — періодична зміна днів тижня

Розглянемо як математично відобразити цю періодичність. Нехай a і b — кількість днів з початку року (з 1 січня). Те, що a і b позначають однакові дні тижня, можна описати таким чином: різниця $a–b$ є кратною 7. Тобто $a–b = 7m$ (m — ціле число). При такому обмеженні a і b будуть відповідати однаковим дням тижня. Математично це можна записати таким чином: $a \equiv b$ (mod 7) (mod — скорочена форма написання modulo). Запис mod$n$ означає конгруентність при n, що є дільником. Отже, запис $a \equiv b$ (mod7) означає, що a ***конгруентне*** (подібне) b, коли дільником є 7.

Для будь-якого простого числа p кільце залишків за модулем p (mod$p$) — це скінченне поле з p елементів, які позначають елементи цього поля і можуть бути репрезентовані цілими числами, які додаються і множаться за модулем p. Ця конструкція узагальнює поле, яке відповідає $p = 2$. Будь-яке скінченне поле містить у собі p^n-елементів. Простим прикладом скінченного поля є кільце простого числа.

Серед арифметичних операторів комп'ютерної мови BASIC існує оператор MOD. Якщо, використавши цей оператор, ввести в комп'ютер команду PRINT 125MOD7, то отримаємо відповідь 6. Тобто операція MOD у даному випадку полягає в тому, щоб виділити залишок після ділення числа 125 на 7: 125:7 = 17 і залишок 6. Отже, запис $a \equiv b$ (mod7) означає, що залишок від ділення a : 7 дорівнює залишку від ділення b : 7.

Ділення цілого числа a на додатне ціле число n можна записати з використанням такого виразу:

$$a = q \cdot n + r \quad (0 \le r < n). \tag{22}$$

З рівняння (22) можна одержати багато важливих співвідношень, одним з яких є математичне обґрунтування теорії кодів.

При діленні цілого числа a на n залишок r може приймати одне із значень: n–1, n–2, ..., 1, 0. Усі ці цілі числа можна класифікувати за mod$n$. Наприклад, дні тижня відповідають залишку від ділення на 7, тобто їх можна класифікувати за mod7. Указаний залишок пов'язаний з днем тижня таким чином: 0 — неділя, 1 — понеділок, 2 — вівторок, 3 — середа, 4 — четвер, 5 — п'ятниця, 6 — субота.

Розглянемо множину F, елементами якої є залишки від ділення на 7:

$$F \equiv \{0, 1, 2, 3, 4, 5, 6\}. \tag{23}$$

Проаналізуємо операції додавання і множення серед елементів множини F:

Операція додавання	***Операція множення***
$4 + 5 \equiv 2$ (mod 7)	$4 \cdot 5 \equiv 6$ (mod 7)
$6 + 4 \equiv 3$ (mod 7)	$6 \cdot 4 \equiv 3$ (mod 7)
.	.
.	.
.	.

Операції додавання та множення елементів множини F відповідають вмісту табл. 3.

Далі проведемо аналіз реалізації можливості виконання операцій віднімання та ділення між елементами множини F:

Операція віднімання	*Операція множення*
$6 - 2 = 4 \pmod 7$	$4 \cdot 5 \equiv 6 \pmod 7$
$5 - 3 = 2 \pmod 7$	$6 \cdot 4 \equiv 3 \pmod 7$

Процес віднімання від більшого числа меншого зрозумілий. Розглянемо, як відняти більше число від меншого, наприклад 1 – 6. Звернемося знову до таблиці додавання (див. табл. 3). Підберемо комбінації по два елементи так, щоб у сумі вони давали 0:

$$0 + 0 \equiv 0 \pmod 7,$$

$$1 + 6 \equiv 0 \pmod 7,$$

$$2 + 5 \equiv 0 \pmod 7,$$

$$3 + 4 \equiv 0 \pmod 7.$$

Тепер виберемо, наприклад, вираз $1 + 6 \equiv 0 \pmod 7$ і віднімемо від обох частин цієї тотожності число 6, отримаємо: $1 \equiv -6 \pmod 7$. Поміняємо ліву і праву частини тотожності місцями: $-6 \equiv 1 \pmod 7$. Отже, – 6 при додаванні за mod 7 конгруентне (тотожно подібне) 1, тобто

$$1 - 6 \equiv 1 + (-6) \equiv 1 + 1 \equiv 2 \pmod 7.$$

Як бачимо, віднімання 6 аналогічне додаванню числа – 6. Якщо взяти два елементи a і x, сума яких дорівнює 0, а саме, $a + x \equiv 0 \pmod 7$, то операція віднімання a здійснюється шляхом додавання x.

Запишемо операцію ділення (за винятком ділення на 0):

$$6 : 3 \equiv 2 \pmod 7,$$

$$4 : 2 \equiv 2 \pmod 7.$$

Таблиця 3. Результати додавання та множення елементів множини F за mod 7

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Дільниками числа 6 є числа 2 і 3, а дільником числа 4 — 2. Якщо ділення не має дільників або менше число ділиться на більше, наприклад 1 : 6, то замість ділення можна використати операцію множення на 6^{-1} : $1 : 6 = 1 \cdot 6^{-1}$.

Звернемося до табл. 3 та підберемо по два елементи, результат множення яких становить 1:

$$1 \cdot 1 \equiv 1 \pmod 7,$$
$$2 \cdot 4 \equiv 1 \pmod 7,$$
$$3 \cdot 5 \equiv 1 \pmod 7,$$
$$6 \cdot 6 \equiv 1 \pmod 7.$$

З наведених співвідношень можна отримати значення, обернені кожному елементу:

$$1^{-1} \equiv 1 \pmod 7,$$
$$2^{-1} \equiv 4 \pmod 7,$$
$$3^{-1} \equiv 5 \pmod 7,$$
$$4^{-1} \equiv 2 \pmod 7,$$
$$5^{-1} \equiv 3 \pmod 7,$$
$$6^{-1} \equiv 6 \pmod 7.$$

Отже, операцію ділення можна виконати так:

$$1 : 6 = 1 \cdot 6^{-1} \equiv 1 \cdot 6 \equiv 6 \pmod 7.$$

Згідно з наведеним вище, для безлічі елементів множини з основою за mod n можна вільно виконувати операції додавання, віднімання і множення. Операцію ділення (за винятком ділення на 0) можна здійснювати лише в межах табличних значень n.

Поле — це множина F з двома бінарними операціями (*адитивна операція*, або додавання $(a+b)$) і (*мультиплікативна операція*, або множення $(a \cdot b)$). Для того щоб безліч елементів множини, на якій задані операції додавання і множення, була полем, необхідно, щоб для кожної з цих операцій виконувалися всі групові аксіоми (див. нижче), а також дистрибутивний закон, тобто для будь-яких трьох елементів поля a, b, c була справедлива рівність $a \cdot (b+c) = a \cdot b + a \cdot c$ і $(b+c) \cdot a = b \cdot a + c \cdot a$.

Крім того, для кожної операції група має бути комутативною, тобто повинні виконуватися операції $a + b = b + a$ і $a \cdot b = b \cdot a$. Відмітимо, що групові властивості для операції множення справедливі для всіх ненульових елементів поля, які, в свою чергу, повинні бути оборотні.

У більш широкому сенсі, множина, в якій можна вільно виконувати операції додавання, віднімання, множення та ділення, і є ***полем***. На практиці мають справу з дійсними числами, проводячи над ними чотири арифметичні дії. Ця безліч дійсних чисел називається ***полем дійсних чисел***. А в такій множині, яка складається з елементів, наприклад за mod 7, поле містить у собі скінченне число елементів і його називають ***кінцевим (або скінченним) полем***.

Скінченні поля були вперше описані французьким вченим Еварістом Галуа (Galois) (1811—1832 рр.), тому їх назвали ***поля Галуа*** (скорочено — *GF*). Галуа створив теорію, яка до цього часу стоїть у фокусі математичної думки. Грунтуючись на працях Лагранжа та Гаусса, він (не досягнувши 19-річчя) висловив і реалізував ідею про те, що до поля коефіцієнтів многочлена (полінома) можна приєднати його корені, і отримати нове поле — розширення колишнього (вихідного) поля. Проаналізувавши числові рівняння, він ввів поняття групи, тобто сукупності таких підстановок між їх коренями, які не порушують раціональних співвідношень між ними. Ця група визначає для кожного рівняння алгебраїчну структуру його коренів. Зокрема, рівняння можуть бути розв'язані в радикалах тоді і тільки тоді, коли його група належить до так званих дозволених груп. Таким чином, питання щодо розв'язку кожного конкретного рівняння в радикалах може бути вирішено за допомогою скінченної кількості дій. Виявляється, що групу можна визначити, не знаючи коренів рівняння $f(x) = 0$, а користуючись лише так званими міркуваннями симетрії.

Ця процедура подібна процедурі вирощування монокристала, що росте шар за шаром. Кристалографічні осі і грані такого кристала мають особливу симетрію (елементи симетрії), яка характерна для даної групи кристалів. І можливо, що від цієї симетрії залежить розв'язок початкового рівняння! Здогадка була вірною, і Галуа зумів довести свою гіпотезу до строгої теорії. Для цього йому довелося створити першу математичну теорію довільних симетрій — так звану теорію груп. У наші дні поняття групи входить у першу десятку математичних термінів, що найбільш вживаються.

Група (в сучасному визначенні) — це будь-яка множина G, на якій задана двомісна операція алгебри, тобто правило, що зіставляє кожним двом елементам з множини G певний третій елемент з цієї самої множини G. У цьому разі виконуються наступні аксіоми, які важливі в подальших застосуваннях:

а) операція асоціативна, тобто $(ab)c = a(bc)$ для будь-яких a, b, c із множини G;

б) множина G містить у собі одиничний елемент, тобто такий елемент e, що $ae = ea = a$ для будь-якого із елементів G;

в) для будь-якого елемента a із множини G існує обернений елемент, тобто такий елемент a^{-1} з множини G, для якого $aa^{-1} = a^{-1}a = e$.

У цьому визначенні прийнято запис операції у вигляді множення, але можна було б використати будь-який інший значок — плюс, мінус, композицію, зірочку тощо. Зараз важливо те, що безліч S_n усіх перестановок n символів щодо множення, визначеного в пункті "а", — теж група. Відмітимо також, операція в групі не завжди комутативна, тобто не завжди підкоряється аксіомі: $ab = ba$ для всіх елементів a та b. Якщо ж операція комутативна, то і група називається ***комутативною***.

Кількість прикладів груп можна значно збільшити, якщо скористатися наступним поняттям: частина H множини G називається її ***підмножиною*** (або ***підгрупою***), якщо множина H замкнута щодо множення і отримання

обернених елементів, тобто разом з будь-якими двома своїми елементами a, b містить у собі також їх добуток ab та обернені величини a^{-1} та b^{-1}. Зрозуміло, що така множина H сама є групою щодо операцій, наявних у множині G. Якщо H — підгрупа групи G, то записують співвідношення $H \leq G$. Наприклад, парні числа щодо додавання є підгрупою групи цілих чисел щодо додавання, але група раціональних чисел без нуля щодо множення не є підгрупою групи дійсних чисел щодо додавання. Хоча перша множина і є частиною другої, але операції в них різні.

Скінченне поле або ***поле Галуа*** $GF(q)$ — це поле, яке складається зі скінченної множини елементів. Число елементів поля q називають порядком поля. Скінченні поля використовуються для побудови більшості відомих кодів і їх декодування.

Залежно від значення q розрізняють прості або розширені поля. Поле називають простим, якщо q — просте число. Для позначення простих чисел використовуватимемо символ p. Просте поле утворюють числа за модулем p: 0, 1, 2, ..., p–1, а операції додавання і множення виконуються за модулем p.

Найменше поле Галуа містить у собі лише два елементи: "0" та "1", арифметичні операції над якими поводяться майже як звичайно, за винятком правила 1 + 1 = 0. Легко переконатися, що 0 і 1 як одиничні елементи для операцій додавання і множення не змінюють значення інших елементів поля для відповідної операції. Крім того, видно, що для кожного елемента для операції додавання і для ненульових елементів для операції множення є обернені елементи. Це поле $GF(2)$, або двійкове. Воно широко застосовується в комп'ютерних науках і теорії кодування. Ідея застосування поля полягає в тому, що доцільно розглядати послідовності з нулів й одиниць як елементи деякої пов'язаної з ним алгебраїчної структури. Алгебраїчні операції в цій структурі приводять до ряду важливих конструкцій в означених галузях. Засновані на теорії скінченних полів алгоритми перевірки на простоту і факторизацію цілих чисел відіграють важливу роль у сучасній прикладній теорії чисел.

Теорія Галуа дає єдиний "елегантний" підхід для вирішення низки класичних завдань: 1) які фігури можна побудувати з використанням циркуля та лінійки; 2) які алгебраїчні рівняння можна розв'язати за допомогою стандартних операцій алгебри (додавання, віднімання, множення, ділення та добування кореня).

З кінцевого поля з основою за mod 7, за винятком 0, формується множина $\hat{F} = \{1, 2, 3, 4, 5, 6\}$. У цій множині $\hat{F}$ можливі тільки операції множення і ділення. Усі елементи множини $\hat{F}$ можна відобразити одним елементом. Наприклад, якщо за основу візьмемо елемент 3, то отримаємо

$$3^0 \equiv 1 \pmod 7,$$

$$3^1 \equiv 3 \pmod 7,$$

$$3^2 \equiv 2 \pmod 7,$$

$$3^3 \equiv 6 \pmod 7,$$

$$3^4 \equiv 4 \pmod 7,$$

$$3^5 \equiv 5 \pmod 7),$$
$$3^6 \equiv 1 \equiv 3^0 \pmod 7).$$

Отже, дану множину $\hat{F}$ можна записати так:

$$\hat{F} = \{3^0, 3^1, 3^2, 3^3, 3^4, 3^5\}. \quad (24)$$

У разі, коли всі елементи множини $\hat{F}$ можна утворити від одного елемента, цей елемент називають ***вихідним*** (***або початковим***).

Серед скінченних полів мінімальну кількість елементів має множина з дільником за mod 2. Дільниками в такій множині є "0" і "1". Необхідною умовою формування поля з елементами "0" і "1" є можливість виконання в ньому чотирьох арифметичних операцій. Розглянемо ці операції в даному полі.

Операція додавання:

$$\begin{aligned} 0 + 0 &\equiv 0 \pmod 2), \\ 0 + 1 &\equiv 1 \pmod 2), \\ 1 + 0 &\equiv 1 \pmod 2), \\ 1 + 1 &\equiv 0 \pmod 2). \end{aligned} \quad (25)$$

Операція множення:

$$\begin{aligned} 0 \cdot 0 &\equiv 0 \pmod 2), \\ 0 \cdot 1 &\equiv 0 \pmod 2), \\ 1 \cdot 0 &\equiv 0 \pmod 2), \\ 1 \cdot 1 &\equiv 1 \pmod 2). \end{aligned}$$

У табл. 4 наведено результати операцій додавання і множення за mod 2. На підставі результатів табл. 4 отримуємо також інші арифметичні операції.

Операція віднімання:

$$\begin{aligned} 0 - 0 &\equiv 0 \pmod 2) \\ 0 - 1 &\equiv 1 \pmod 2) \\ 1 - 0 &\equiv 1 \pmod 2) \\ 1 - 1 &\equiv 1 \pmod 2). \end{aligned} \quad (26)$$

Операція ділення (виключаючи ділення на 0):

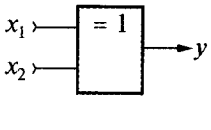
$$1 : 1 = 1 \pmod 2).$$

Тобто в множині {0,1} можливе виконання чотирьох арифметичних операцій, отже цю множину можна назвати полем.

Поле з двох елементів "0" і "1" може відображати два стани і тому є найприйнятнішим для формування цифрових сигналів. Таке поле з двох елементів є полем $GF(2)$. З порівняння операцій дода-

Таблиця 4. Результати додавання і множення за mod 2

+	0	1	×	0	1
0	0	1	0	0	0
1	1	0	1	0	1



x_1	x_2	$y = x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

$y = x_1 \cdot \bar{x}_2 + \bar{x}_1 \cdot x_2 = x_1 \oplus x_2$

а *б*

Рис. 18. Приклад елемента **ВИКЛЮЧНИЙ АБО** (або функція **НЕРІВНОЗНАЧНОСТІ**) (*а*) і таблиця істинності (*б*)

вання (25) і віднімання (26) видно, що їх кінцеві результати збігаються. Це означає, що в полі Галуа *GF*(2) результати додавання і віднімання однакові.

Поле *GF*(2) має ще одну перевагу. При розробці практичних електронних схем елемент ***ВИКЛЮЧНИЙ АБО (сувора диз'юнкція)*** можна застосовувати як для виконання операцій додавання, так і операцій віднімання. Таблиця істинності для операцій, що виконуються за допомогою елемента ***ВИКЛЮЧНИЙ АБО*** (чи функції **НЕРІВНОЗНАЧНОСТІ**), наведена на рис.18. Така таблиця повністю збігається з результатами операції додавання в полі Галуа *GF*(2), що описуються рівняннями (25). Із таблиці істинності функції **НЕРІВНОЗНАЧНОСТІ** випливає, що ця функція збігається з функцією **АБО** в усіх випадках, крім одного, коли всі вхідні змінні дорівнюють одиниці, тому вона і отримала назву функції ***ВИКЛЮЧНИЙ АБО***.

МНОГОЧЛЕН З КОЕФІЦІЄНТІВ *GF*(2) І ОБЧИСЛЮВАЛЬНІ СХЕМИ

Найзручнішим способом подання в цифровому вигляді інформаційної або кодової послідовності є многочлен. Наприклад, многочлен інформаційної послідовності a_3, a_2, a_1, a_0 записується так:

$$a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 = a_3x^3 + a_2x^2 + a_1x + a_0 \ ,$$

де враховано, що за часом найпершим на вхід системи передачі поступає біт a_3. У цьому многочлені x^i (i = 3...0) позначають лише місцезнаходження біта. Тоді можна вважати, що x^{-1} позначає затримку на один період одного синхроімпульсу.

Між місцезнаходженням біта в інформаційній послідовності і показником степеня x в многочлені спостерігається взаємно однозначна відповідність:

положення 1-го біта $a_3 \leftrightarrow x^3$,

положення 2-го біта $a_2 \leftrightarrow x^2$,

положення 3-го біта $a_1 \leftrightarrow x^1 = x$,

положення 4-го біта $a_0 \leftrightarrow x^0 = 1$.

Коефіцієнти многочлена a_3, a_2, a_1, a_0 приймають тільки значення "0" або "1". Тобто можна вважати, що коефіцієнт a_1 являє собою елемент у полі Галуа *GF*(2). Те, що коефіцієнти многочлена належать до *GF*(2), свідчить про можливість розрахунків коефіцієнтів для x^i з таким самим показником степеня.

Якщо многочлен з коефіцієнтами *GF*(2) має вигляд

$$A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1x + a_0 \ , \qquad a_i = 0, 1,$$

то розрахунок аналогічний випадку з використанням цілих чисел, а саме, за mod 2. Проаналізуємо операції додавання, віднімання, множення та ділення з двома парами многочленів: $(x^3 + 1)$ та $(x^2 + x + 1)$ і $(x^2 + x)$ та $(x^3 + x^2 + 1)$.

Операція додавання:

$$(x^3 + 1) + (x^2 + x + 1) = x^3 + x^2 + x,$$
$$(x^2 + x) + (x^3 + x^2 + 1) = x^3 + x + 1.$$

Операція віднімання:

$$(x^3 + 1) - (x^2 + x + 1) = x^3 + x^2 + x \ ;$$
$$(x^2 + x) - (x^3 + x^2 + 1) = x^3 + x + 1.$$

Звідси видно, що для многочлена з коефіцієнтами $GF(2)$ результати додавання і віднімання є однаковими.

Операція множення:

$$(x^3 + 1) \cdot (x^2 + x + 1) = x^5 + x^4 + x^3 + x^2 + x + 1,$$

$$\begin{array}{rr} & x^3 + \quad 1 \\ \times & x^2 + x + 1 \\ \hline + & x^3 + \quad 1 \\ & x^4 + \quad x \\ x^5 + & \quad x^2 \\ \hline x^5 + x^4 + x^3 + x^2 + x + 1 & \end{array}$$

$$(x^2 + x) \cdot (x^3 + x^2 + 1) = x^5 + x^3 + x^2 + x;$$

$$\begin{array}{rr} & x^2 + x \\ \times & x^3 + x^2 + 1 \\ \hline + & x^4 + x^3 \\ x^5 + x^4 & \\ \hline x^5 + x^3 + x^2 + x & \end{array}$$

Операція ділення:

Зазначимо, що під час операції ділення так само, як і у разі використання цілих чисел, залишається залишок від ділення. Тобто, якщо $A(x)$ ділиться на $B(x)$ із залишком, то

$$A(x) = Q(x) \cdot B(x) + R(x). \qquad (27)$$

При цьому найбільший показник степеня залишку $R(x)$ завжди менший, ніж показник степеня $B(x)$. Наприклад $A(x) = x^4 + x^3 + 1$, $B(x) = x^2 + 1$,

$$\begin{array}{rl} - x^4 + x^3 + 1 & \mid x^2 + 1 \\ \hline x^4 \quad + x^2 & \mid x^2 + x + 1 \\ \hline - x^3 + x^2 + 1 & \\ x^3 \quad + x & \\ \hline - x^2 + x + 1 & \\ x^2 \quad + 1 & \\ \hline x & \end{array}$$

Тобто $x^4 + x^3 + 1 = (x^2 + x + 1) \cdot (x^2 + 1) + x$.

Далі, якщо, наприклад, $A(x) = x^4 + 1$ *а* $B(x) = x^2 + 1$, то

$$\begin{array}{rl} - x^4 + 1 & \mid x^2 + 1 \\ \hline x^4 + x^2 & \mid x^2 + 1 \\ \hline - x^2 + 1 & \\ x^2 + 1 & \\ \hline 0 & \end{array}$$

Таким чином, маємо $x^4 + 1 = (x^2 + 1)(x^2 + 1)$.

У цьому разі говорять, що $A(x) = x^4 + 1$ ділиться на $B(x) = x^2 + 1$ без залишку. Тоді $B(x)$ називається множником $A(x)$.

Вираз $x^4 + 1$ можна додатково розкласти на множники:

$$x^4 + 1 = (x^2 + 1)\,(x^2 + 1) = (x^2 + 1)^2 = \{(x + 1)\,(x + 1)\}^2 = (x + 1)^4.$$

Серед цілих чисел є прості числа, множниками яких є "1" і самі ці числа. Серед многочленів з коефіцієнтів поля $GF(2)$ теж є многочлени, множниками яких є "1" і самі ці многочлени. Наприклад, у многочлена $x^2 + x + 1$ множниками є 1 і $x^2 + x + 1$, тобто два множники. Многочлени, які не можна розкласти на множники, називаються *нескоротними*, а многочлени, які можна розкласти на множники, називаються ***звідними*** многочленами, або ***поліномами***.

Незвідний многочлен (або ***нескоротний***) — це многочлен, що не можна розкласти на нетривіальні (неконстантні) поліноми (многочлени). У кільці багатьох багаточленів ***незвідні многочлени*** відіграють таку саму важливу роль, як і прості числа в кільці цілих чисел.

Між цілими числами і многочленами існує така відповідність:

цілі числа		***многочлени***
прості числа	—	нескоротні,
складені числа	—	звідні.

У випадку цілих чисел множина, елементами якої є залишки n–1, n–2, ..., 1, 0 за mod n (n — просте число) утворює поле. Аналогічно і многочлени множини, елементами якої є залишки за mod (нескоротний многочлен), утворюють поле.

Незвідний многочлен над полем k — це многочлен $p(x_1,x_2,..,x_n)$ від n змінних над полем k, що є простим елементом кільця $k[x_1,x_2,..,x_n]$, тобто незображуваний у вигляді добутку $p = qr$ (q і r — многочлени з коефіцієнтами з k відмінні від константи). Многочлен називається ***таким, що абсолютно не зводиться***, якщо він не зводиться над замиканням алгебри поля коефіцієнтів. Многочлени однієї змінної, що абсолютно не зводяться, — це многочлени 1-го степеня і лише такі. У разі декількох змінних існують многочлени, що абсолютно не зводяться, скільки завгодно високого степеня, наприклад будь-який многочлен вигляду

$$p(x_1,x_2,..,x_{n-1}) + x_n$$

абсолютно не зводиться.

Як приклад розглянемо нескоротний многочлен $x^2 + x + 1$. Використовуючи формулу (27), отримуємо

$$A(x) = Q(x)\left(x^2 + x + 1\right) + R(x)\ , \qquad (28)$$

де показник степеня $R(x) < 2$. Отже, як $R(x)$ можна розглядати чотири залишки: 0, 1, x, $x + 1$. Як наслідок утворюється множина $F_2 = 0, 1, x, x + 1$.

Проаналізуємо властивості цієї множини F_2. Сформуємо таблицю результатів операцій додавання і множення (табл. 5). Як бачимо, за результатами множення для отримання "1" потрібні всі елементи 1, x, $x + 1$, окрім 0, тобто:

$$1 \times 1 \equiv 1 \pmod{x^2 + x + 1},$$

$$x \times (x + 1) \equiv 1 \pmod{x^2 + x + 1},$$

$$(x + 1) \times (x + 1) \equiv 1 \pmod{x^2 + x + 1}.$$

Таблиця 5. Результати додавання та множення за mod $x^2 + x + 1$ для нескоротного многочлена

+	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

×	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

Отже, множина F_2 залишків mod $x^2 + x + 1$ утворює скінченне поле.

Розглянемо тепер ділення многочлена залишку на многочлен, який не є нескоротним. Якщо як звідний многочлен взяти $x^2 + 1$, то отримаємо чотири залишки: 0, 1, x, $x + 1$. Результати операцій додавання і множення за mod $x^2 + 1$ наведені в табл. 6.

З таблиці множення (див. табл. 6) видно, що "0" можна отримати, перемноживши елементи, які не обов'язково дорівнюють 0, наприклад

$$(x+1)(x+1) \equiv 0 \pmod{x^2+1}.$$

У цьому випадку виконувати операцію ділення на $(x + 1)$ не можна. Тобто при дільнику за mod $x^2 + 1$ можна додавати, віднімати, множити, але не можна ділити. Це означає, що дана множина не утворює поля.

У разі використання нескоротного многочлена $x^2 + x + 1$ як елемента множини F_2 поле не утворюється. Якщо ж сформувати з наведених вище елементів множину F_2^*, в якій виключено 0, то в ній можливі тільки операції множення і ділення:

$$F_2^* = \{1,\ x,\ x+1\}. \tag{29}$$

У множині, створеній на базі mod $x^2 + x + 1$, можливе формування й інших елементів з тих, що є в наявності, зокрема:

$$x^0 \equiv 1 \pmod{x^2+x+1},$$
$$x^1 \equiv x \pmod{x^2+x+1},$$
$$x^2 \equiv x+1 \pmod{x^2+x+1},$$
$$x^3 \equiv 1 \equiv x^0 \pmod{x^2+x+1}.$$

Таким чином, множину (29) можна записати у вигляді

$$F_2^* = \{1,\ x,\ x^2\}. \tag{30}$$

Таблиця 6. Результати додавання та множення для нескоротного многочлена за mod $x^2 + 1$

+	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

×	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	1	$x+1$
$x+1$	0	$x+1$	$x+1$	0

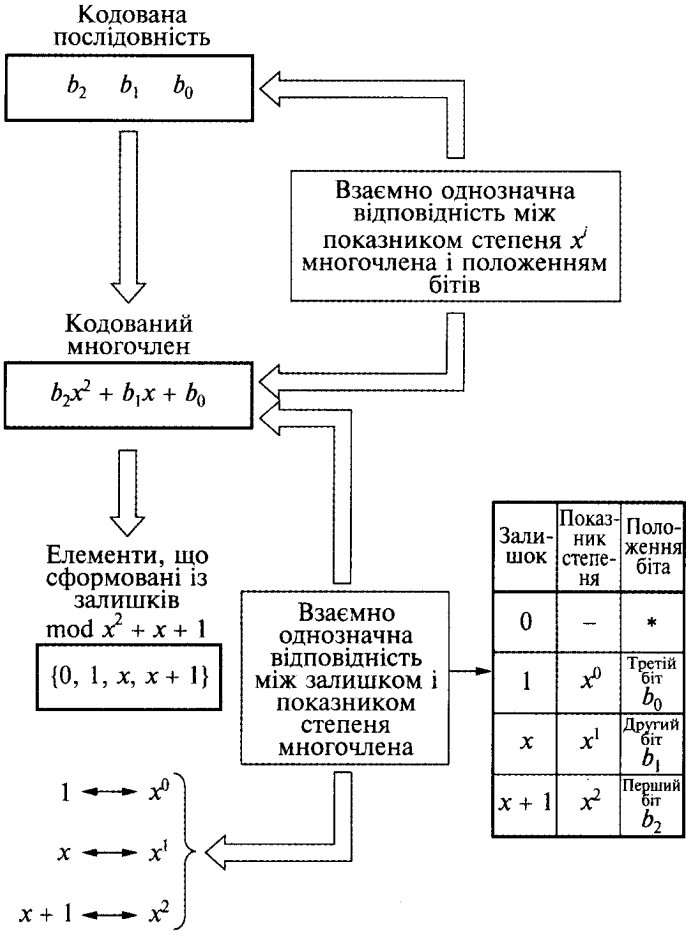
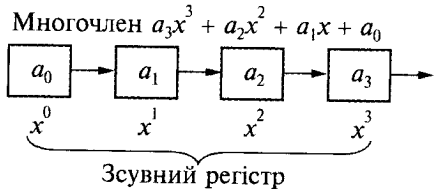


Рис. 19. Відповідність між залишками і кодованою послідовністю

Те, що елементи скінченного поля, сформованого із залишків mod $x^2 + x + 1$, відображаються показником степеня x, дозволяє говорити про взаємно однозначну відповідність між положенням біта в кодовій послідовності і залишком незвідного многочлена. Ця відповідність подана на рис. 19.

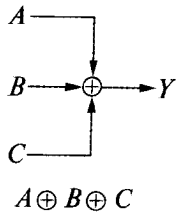
Із залишків незвідного многочлена коефіцієнтів $GF(2)$ утворено поле $GF(2^2)$. Це означає, що з $GF(2)$ шляхом збільшення порядку незвідного многочлена можна сформувати многочлен $GF(2^n)$. Якщо степінь такого многочлена дорівнює n, то кількість елементів утвореної із залишків множини становить 2^n. Якщо виключити елемент "0" і позначити всі $(2^n - 1)$ елементи x, тобто врахувати, що елемент x періодичний з періодом $(2^n - 1)$, то такий незвідний многочлен називають ***початковим*** многочленом.

Враховуючи те, що многочлен відображає часове положення бітів, його коефіцієнти можна накопичувати в регістрі зсуву. При розрахунку многочлена достатньо лише знати коефіцієнти поля $GF(2)$. Такий розрахунок можна виконувати за допомогою операції ***ВИКЛЮЧНИЙ АБО*** (рис. 20), або так званого підсумовування за модулем 2 (mod 2).



Синхро-сигнали	Вихідна послідовність
0	$a_3 \leftarrow x^3$
1	$a_2 \leftarrow x^2$
2	$a_1 \leftarrow x^1$
3	$a_0 \leftarrow x^0$

Додавання $A + B + C = Y \pmod 2$



A	B	C	$Y = A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Рис. 20. Схема додавання за модулем 2 та розрахунок коефіцієнтів многочлена $a_3x^3 + a_2x^2 + a_1x + a_0$

Суматори пристрою виконують операцію формування перевірних символів, які в подальшому послідовно надходять на вихід кодера. Ще одним важливим елементом, якого з розрахункових виразів не видно, є перемикальна схема. Крім вказаних елементів, необхідні також інші додаткові схеми, наприклад генератори сигналів з різними частотами, схеми синхронізації, або генератори синхроімпульсів для синхронізації робочих генераторів, схеми синтезу частот тощо.

Пристрої для додавання і віднімання можна реалізувати на базі суматорів за mod 2 (чи операції ***ВИКЛЮЧНИЙ АБО***). Обов'язковою умовою у цьому разі є правильне розташування коефіцієнтів у часі.

Пристрої множення і ділення відіграють не менш важливу роль у цифрових пристроях обробки інформації. Наприклад, ми маємо в цифрових кодах значення напруги U і струму I у колі. Щоб отримати значення потужності, потрібно вказані коди перемножити, а значення опору кола U поділити на I. Частіше зустрічаються завдання, в яких необхідно значення якоїнебудь величини множити і ділити на різні коефіцієнти, що також можуть бути змінними.

Помножувальна і ділильна схеми виконують роль як декодера, призначеного для виправлення помилок, так і кодера, що використовується для формування коду виправлення помилок на базі многочлена.

Помножувальна схема. Припустимо, необхідно перемножити два многочлени $A(x) = x^4 + x^3 + 1$ і $B(x) = x^2 + x + 1$:

$$
\begin{array}{l}
\times \quad x^4 + x^3 + 0\cdot x^2 + 0\cdot x + 1 \\
\qquad\qquad\qquad\qquad x^2 + x + 1 \\
\hline
x^4 + x^3 + 0\cdot x^2 + 0\cdot x + 1 \\
+ \quad x^5 + x^4 + 0\cdot x^3 + 0\cdot x^2 + x \\
x^6 + x^5 + 0\cdot x^4 + 0\cdot x^3 + x^2 \\
\hline
x^6 + \qquad\qquad x^3 + x^2 + x + 1
\end{array}
\qquad
\begin{array}{l}
\leftarrow x^0 \\
\leftarrow x^1 \\
\leftarrow x^2
\end{array}
\Big\}\ \text{добуток на коефіцієнт...}
$$

Дії з многочленами виконуються так само, як і в звичайній алгебрі, тільки додавання здійснюється за модулем 2. Схематично цей процес розрахунку добутку двох множників відповідає роботі помножувальної схеми.

Якщо переставити місцями множники $A(x)$ і $B(x)$, то отримаємо той самий результат:

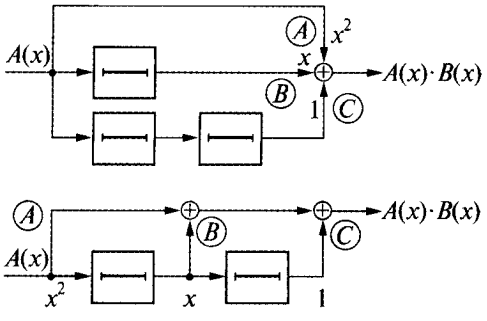
$$
\begin{array}{rl}
\times & x^2 + x + 1 \\
 & x^4 + x^3 + 0\cdot x^2 + 0\cdot x + 1 \\
\hline
 & x^2 + x + 1 \qquad \leftarrow 1\cdot B(x) \\
 & 0\cdot x^3 + 0\cdot x^2 + 0\cdot x \qquad \leftarrow 0\cdot x\cdot B(x) \\
+ & 0\cdot x^4 + 0\cdot x^3 + 0\cdot x^2 \qquad \leftarrow 0\cdot x^2\cdot B(x) \\
 & x^5 + x^4 + x^3 \qquad \leftarrow x^3\cdot B(x) \\
 & x^6 + x^5 + x^4 \qquad \leftarrow x^4\cdot B(x) \\
\hline
 & x^6 + x^3 + x^2 + x + 1
\end{array}
$$

добуток $B(x)$ на член многочлена...

Отже, множення $A(x)$ на $B(x)$ зводиться до додавання добутків множників на коефіцієнти x^2, x^1, x^0 з певним зміщенням добутків на ту або іншу кількість розрядів. При цьому необхідно враховувати послідовність операцій в часі, а саме: добуток на коефіцієнт x^2 заноситься в суматор безпосередньо, добуток на коефіцієнт x^1 затримується на час одного такту (синхроімпульсу) і надходить на вхід суматора із зміщенням на один розряд зсувного регістра, а добуток на x^0 затримується на час двох тактів (синхроімпульсів) і надходить на вхід суматора із зміщенням на два розряди зсувного регістра. На виході суматора отримуємо добуток на $x^2 + x + 1$ (рис. 21).

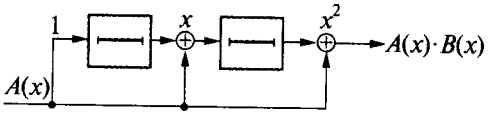
Якщо розглядати обернену залежність $A(x)$ і $B(x)$, то в помножувальній схемі такого типу суматори потрібно розмістити між розрядами зсувного регістра (рис. 22).

З наведених процесів видно, що операція множення в двійкових кодах досить проста: множене або передається на суматор, якщо в даному розряді множника знаходиться "1", або не передається ("0"). За імпульсом "Скидання" перший множник (множене) записується в 1-й регістр $RG1$, а 2-й регістр $RG2$ обнуляється. На тактовий вхід регістра $RG2$ передаються послідовно роз-



Помножувальна схема $B(x)$					
Синхроімпульс	Вхід $A(x)$	Добуток на			Вихід $A(x)\cdot B(x)$ $A + B + C$
		x^2	x	1	
0	$x^4 \to 1$	1	0	0	$1 \leftarrow x^6$
1	$x^3 \to 1$	1	1	0	$0 \leftarrow x^5$
2	$x^2 \to 0$	0	1	1	$0 \leftarrow x^4$
3	$x^1 \to 0$	0	0	1	$1 \leftarrow x^3$
4	$x^0 \to 1$	1	0	0	$1 \leftarrow x^2$
5	0	0	1	0	$1 \leftarrow x^1$
6	0	0	0	1	$1 \leftarrow x^0$

Рис. 21. Помножувальна схема многочленів $A(x)$ та $B(x)$



Синхроімпульси	Вхід $A(x)$	1 розряд регістра	2 розряд регістра	Входи $A(x) \cdot B(x)$
0	$x^4 \to 1$	0	0	$1 \leftarrow x^6$
1	$x^3 \to 1$	1	1	$0 \leftarrow x^5$
2	$x^2 \to 0$	1	0	$0 \leftarrow x^4$
3	$x^1 \to 0$	0	1	$1 \leftarrow x^3$
4	$x^0 \to 1$	0	0	$1 \leftarrow x^2$
5	0	1	1	$1 \leftarrow x^1$
6	0	0	1	$1 \leftarrow x^0$

Рис. 22. Помножувальна схема у разі оберненої залежності $A(x)$ і $B(x)$

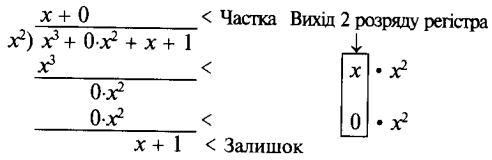
ряди другого множника. Якщо розряд другого множника дорівнює "1", то за даним тактовим імпульсом у регістр $RG2$ записується значення розряду множеного. У наступному такті синхроімпульс надходить на вхід зсувного регістра $RG1$ і зсуває множене на один розряд. Якщо він, наприклад, дорівнює "0", то в регістр $RG2$ з виходу суматора нічого не записується. Пристрій працює доти, поки не будуть перебрані всі розряди множеного і множника. Кількість тактових імпульсів повинна мати $(n + m)$ розрядів (n — кількість розрядів множеного, m — кількість розрядів множника). Отже, усі елементи схеми повинні мати також $(n + m)$ розрядів.

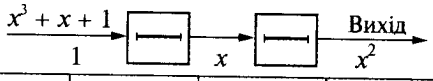
У разі одночасного виконання операцій множення і ділення в схемі такого типу зсувний регістр використовується для виконання обох операцій.

Схема ділення. Побудова пристроїв ділення досить складна. Операція ділення може бути здійснена у вигляді або многочленів, або двійкових кодів. Схема ділення реалізується на зсувних регістрах із вбудованими суматорами за модулем 2. Вид схеми визначається типом многочлена, на який ділять. У процесі ділення за допомогою такого пристрою знаходиться залишок. Розглянемо процедуру та схеми ділення на деякі многочлени, що найбільше використовуються.

Приклад та схема ділення многочлена $x^3 + x + 1$ на дільник x^2 наведені на рис. 23.

На вхід пристрою ділення подається вхідна комбінація $x^3 + x + 1$. У процесі ділення на дільник x^2 отримуємо залишок $x + 1$. Тобто спочатку формується коефіцієнт для x. Це пояснюється тим, що найпершій коефіцієнт з'являється при затримці x^3 на два синхроімпульси після його введення. Для цього використовується два розряди зсувного регістра (кількість розрядів зсувного регістра, що використовуються в схемі ділення, дорівнює максимальному степеню дільника).



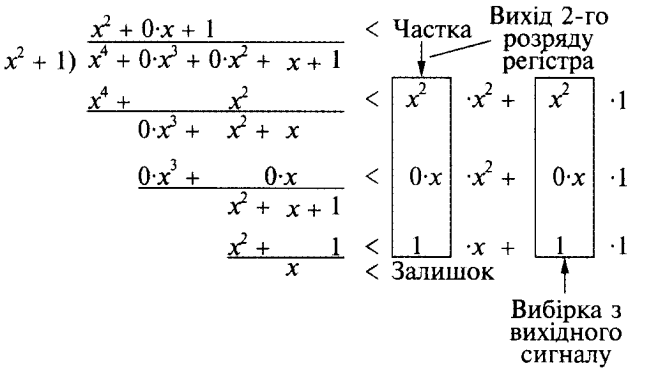


Синхроімпульси	Вхід	Вихід 1-го розряду регістра	Вихід 2-го розряду регістра
0	$x^3 \to 1$	0	$0 \leftarrow x^3$
1	$x^2 \to 0$	1	$0 \leftarrow x^2$
2	$x^1 \to 1$	0	$1 \leftarrow x^1$
3	$x^0 \to 1$	1	$0 \leftarrow x^0$
4	0	1 (Залишок)	$1 \leftarrow x^1$
5	0	0	$1 \leftarrow x^0$

Рис. 23. Схема ділення на x^2

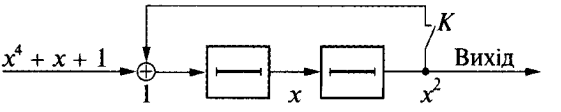
Після того, як повністю введено вираз $x^3 + x + 1$, на виході формується послідовність, що є часткою. У цьому разі в зсувний регістр повинен надійти коефіцієнт многочлена, що не перевищує x. Тобто після введення $x^3 + x + 1$ залишковий вміст регістра є залишком від ділення.

Тепер нехай потрібно поділити вираз $x^4 + x + 1$ на дільник $x^2 + 1$. Відобразимо процес так:



На виході схеми, що формує різницю після кожного етапу ділення, одержуємо різницю початкового многочлена та добутку дільника на результат ділення двох многочленів. Якщо після ділення виразу $x^4 + x + 1$ на дільник $x^2 + 1$ різниця більша ніж нуль, то на виході схеми з'являється "1". Якщо є переповнення (різниця менша ніж нуль), то на виході схеми формується "0". З кожним тактом роботи відбувається послідовне зсування тактів у дільнику та у вихідному регістрі в протилежних напрямах. Кількість тактів роботи схеми визначається розрядністю кінцевого результату.

Тобто при діленні на $x^2 + 1$ необхідно від стану "1" (вхідного сигналу) віднімати коефіцієнти частки (реально виконується додавання). Схема ділення на $x^2 + 1$ наведена на рис. 24.



Синхроімпульси	Вхідний сигнал	Вихід 1-го розряду регістра	Вихід 2-го розряду регістра	Перемикач K
0	$x^4 \longrightarrow 1$	0	$0 \longleftarrow x^4$	Частка
1	$x^3 \longrightarrow 0$	1	$0 \longleftarrow x^3$	
2	$x^2 \longrightarrow 0$	0	$1 \longleftarrow x^2$	ВКЛ
3	$x^1 \longrightarrow 1$	1	$0 \longleftarrow x^1$	
4	$x^0 \longrightarrow 1$	1	$1 \longleftarrow x^0$	
5	0	0 (Залишок)	$1 \longleftarrow x^1$	ВИКЛ
6	0	0	$0 \longleftarrow x^0$	

Рис. 24. Схема ділення на x^2+1

У момент закінчення введення многочлена $x^4 + x + 1$ у зсувному регістрі з'являється залишок. Після цього досить його зафіксувати і відключити схему (встановити перемикач у положення **ВИКЛ**), яка віднімає вихідний сигнал (коефіцієнти частки) від вхідного сигналу.

У цілому, в схемі віднімання виконуються такі операції:

- початкове установлення в "0" вмісту зсувного регістра;
- автоматичне введення "0" після закінчення введення всіх коефіцієнтів;
- автоматична генерація "0" в положенні перемикача **ВИКЛ**.

Таким чином, кодовану послідовність, що передається, можна побудувати множенням заданого вхідного многочлена, що подається на вхід схеми, на одночлен x^{n-k} (n та k — загальна кількість бітів, що передається, та кількість інформаційних бітів), з наступним додаванням до цього добутку залишку після ділення вхідної комбінації на дільник. При декодуванні прийняту кодову комбінацію необхідно поділити на дільник. Наявність залишку від ділення вказує на помилку при передачі сигналу.

Розглянемо ще один типовий приклад ділення $x^4 + x^3$ на дільник $x^2 + x + 1$:

$$
\begin{array}{r}
x^2 + 0\cdot x + 1 \\
x^2 + x + 1 \overline{)\, x^4 + x^3 + 0\cdot x^2 + 0\cdot x + 0} \\
\underline{x^4 + x^3 + x^2} \\
0\cdot x^3 + x^2 + 0\cdot x \\
\underline{0\cdot x^3 + 0\cdot x^2 + 0\cdot x} \\
x^2 + 0\cdot x + 0 \\
\underline{x^2 + x + 1} \\
x + 1
\end{array}
\quad
\begin{array}{l}
< \text{Частка} \\
\\
< x^2 \cdot x^2 + x^2 \cdot (x+1) \\
\\
< 0\cdot x \cdot x^2 + 0\cdot x \cdot (x+1) \\
\\
< 1 \cdot x^2 + 1 \cdot (x+1) \\
< \text{Залишок}
\end{array}
$$

Вихід 2-го розряду регістра

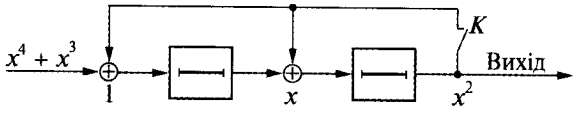
Віднімання вхідного сигналу від сигналу 1-го розряду регістра

Приклад побудови схеми ділення многочлена $x^4 + x^3$ на дільник $x^2 + x + 1$ наведено на рис. 25. На вхід схеми надходить многочлен $x^4 + x^3$. У процесі ділення на дільник $x^2 + x + 1$ отримуємо залишок $x + 1$.

Помножувально-ділильна схема. Використовуючи зсувні регістри, можна одночасно виконувати операції множення і ділення (рис. 26).

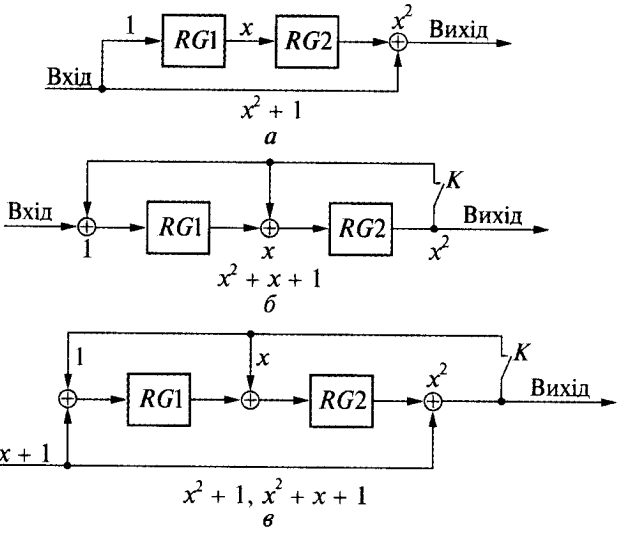
Розглянемо схему, в яку вводиться многочлен $x^3 + x + 1$ і виконується множення на $x^2 + 1$ і ділення на $x^2 + x + 1$. Спочатку здійснюємо множення:

$$
\begin{array}{r}
\times\ x^3 + x + 1 \\
x^2 + 1 \\
\hline
x^3 + x + 1 \\
x^5 + x^3 + x^2 \\
\hline
x^5 + x^2 + x + 1
\end{array}
$$



Синхроімпульси	Вхідний сигнал	Вихід 1-го розряду регістра	Вихід 2-го розряду регістра	Перемикач K
0	$x^4 \longrightarrow 1$	0	$0 \longleftarrow x^4$	ВКЛ
1	$x^3 \longrightarrow 1$	1	$0 \longleftarrow x^3$	
2	$x^2 \longrightarrow 0$	1	$1 \longleftarrow x^2$	
3	$x^1 \longrightarrow 0$	1	$0 \longleftarrow x^1$	
4	$x^0 \longrightarrow 0$	0	$1 \longleftarrow x^0$	
5	0	1 (Залишок)	$1 \longleftarrow x^1$	ВИКЛ
6	0	0	$1 \longleftarrow x^0$	

Рис. 25. Схема ділення на $x^2 + x + 1$



Синхроімпульси	Вхідний сигнал	Вихід 1-го розряду регістра	Вихід 2-го розряду регістра	Вихідний сигнал	Перемикач K
0	$x^3 \longrightarrow 1$	0	0	$1 \longleftarrow x^3$ (Частка)	ВКЛ
1	$x^2 \longrightarrow 0$	0	1	$1 \longleftarrow x^2$ (Частка)	
2	$x^1 \longrightarrow 1$	1	1	$0 \longleftarrow x^1$ (Частка)	
3	$x^0 \longrightarrow 1$	1	1	$0 \longleftarrow x^0$ (Частка)	
4	0	1 (Залишок)	1	$1 \longleftarrow x^1$ (Залишок)	ВИКЛ
5	0	0	1	$1 \longleftarrow x^0$ (Залишок)	

Рис. 26. Схема множення на $x^2 + 1$ і ділення на $x^2 + x + 1$

Схема множення наведена на рис. 26, *а*. На рис. 26, *б* подана схема ділення на $x^2 + x + 1$. Результат операції ділення записується так:

$$
\begin{array}{rl}
 & x^3 + x^2 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \leftarrow \text{Частка} \\
x^2 + x + 1) & x^5 + \qquad\qquad\quad x^2 + x + 1 \\
 & \underline{x^5 + x^4 + x^3} \\
 & \quad x^4 + x^3 + x^2 \\
 & \quad \underline{x^4 + x^3 + x^2} \\
 & \qquad\qquad\qquad\qquad x + 1 \qquad\qquad \leftarrow \text{Залишок}
\end{array}
$$

Для об'єднання схем множення і ділення зазвичай перетворюють структуру щодо зсувних регістрів. Отримана помножувально-ділильна схема показана на рис. 26, *в*.

Припустимо тепер, що необхідно здійснити множення x^2 і ділення на $x^2 + x + 1$. Помножувально-ділильні схеми такого типу застосовуються як кодуючі схеми при формуванні кодів з виправленням помилок (рис. 27).

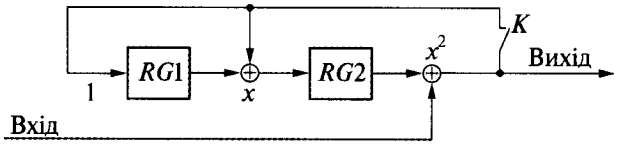


Рис. 27. Помножувально-ділильна схема для множення на x^2 і ділення на $x^2 + x + 1$

Розділ 7

ЦИКЛІЧНИЙ КОД

Широкого поширення на практиці набув клас лінійних кодів, які називаються ***циклічними***. Такі коди є різновидом систематичних кодів і тому мають усі їх властивості. Характерна особливість циклічного коду, яка і визначає його назву, — якщо кодована послідовність із n символів: $a_0\ a_1\ a_2 \ldots\ a_{n-1}\ a_n$ належить даному коду, то і комбінація $a_n\ a_0\ a_1\ a_2\ \ldots\ a_{n-1}$, яка отримана циклічною перестановкою елементів, також належить йому.

Отже, ***циклічним кодом*** називається лінійний блоковий (n, k)-код, який характеризується властивістю циклічності, тобто унаслідок зсування вліво на один крок будь-якого дозволеного кодового слова одержують також дозволене кодове слово, яке належить цьому самому коду, і в якого безліч кодових слів записується сукупністю многочленів степеня $(n-1)$ і нижче, що діляться на деякий многочлен $G(x)$ степеня $m = n-k$, який є співмножником двочлена x^{n+1}. Враховуючи все це можна сказати, що циклічний код — це код, усі робочі комбінації якого діляться на ***многочлен*** $G(x)$, ***що породжує*** (***або породжувальний многочлен*** $G(x)$), без залишку. Тобто для його побудови в принципі досить знати породжувальний многочлен $G(x)$. Ці властивості застовуються при побудові кодів, кодувальних і декодувальних пристроїв, а також для виявлення та виправлення помилок.

Таким чином, ***циклічні коди*** — це спеціальна група кодів, для побудови яких можуть бути використані циклічні властивості квадратних матриць, а також коди, які описуються незвідними многочленами, що створюють ***породжувальні поліноми***. Наприклад, для кодової комбінації 111001 поліноміальний запис має вигляд

$$A(x) = 1 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 = x^5 + x^4 + x^3 + 1.$$

Циклічні коди — це ціла сім'я завадостійких кодів, що містять у собі як один з різновидів коди Хеммінга. У цілому така ситуація забезпечує велику гнучкість з погляду можливості реалізації кодів з необхідною здатністю виявлення і виправлення помилок, що виникають під час передачі кодових комбінацій по каналу зв'язку. Циклічний код належить до систематичних блокових (n, k)-кодів, у яких контрольні m та інформаційні k розряди розташовані на строго певних місцях — перші k розрядів є комбінацією первинного коду, а наступні $(n-k)$ розрядів є перевірними (контрольними) : $n = k + m$.

Побудова циклічних кодів ґрунтується на операції ділення кодової комбінації, що передається, на породжувальний незвідний поліном (многочлен) степеня m. Залишок від ділення використовується для формування перевірних розрядів. При цьому операції ділення передує операція множення, що здійснює зсув вліво k-розрядної інформаційної кодової комбінації на m розрядів.

При декодуванні прийнятої n-розрядної кодової комбінації знову здійснюється ділення на породжувальний поліном. Синдромом помилки в таких кодах є наявність залишку від ділення прийнятої кодової комбінації на породжувальний поліном. Якщо синдром дорівнює нулю, то вважається, що помилок немає. Інакше, за допомогою отриманого синдрому можна визначити номер розряду прийнятої кодової комбінації, в якому відбулася помилка, і виправити її.

Проте не виключається можливість виникнення в кодових комбінаціях багатократних помилок, що може призвести до помилкових виправлень і(або) не виявлення помилок при трансформації однієї дозволеної комбінації в іншу.

Породжувальним поліномом (або многочленом) циклічного (n, k) коду із c базових комбінацій називається такий ненульовий поліном:

$$g(x) = \sum_{i=0}^{r} g_i x^i ,$$

степінь якого найменший і коефіцієнт при старшому степені $g_r = 1$. Можна показати, що всі кодові слова конкретного циклічного коду кратні певному породжувальному поліному $G(x)$. Такий поліном у цьому разі є дільником двочлена x^n-1.

Одне з основних завдань розробників пристроїв захисту від помилок при передачі дискретних повідомлень по каналах зв'язку, — це вибір породжувального многочлена $G(x)$ для побудови циклічного коду, що забезпечує необхідну мінімальну кодову відстань для гарантійного виявлення і виправлення t-кратних помилок. У кожного циклічного коду є свої особливості формування $G(x)$. Тому при вивченні конкретних циклічних кодів розглядатимуться відповідні способи побудови многочлена $G(x)$.

Ідея побудови циклічних кодів базується на використанні незвідних многочленів, тобто таких многочленів, які не можуть бути записані у вигляді добутку многочленів нижчих степенів. Отже, це такий многочлен, який ділиться тільки на самого себе або на одиницю і не ділиться ні на який інший многочлен. На такий многочлен ділиться без залишку двочлен $x^n - 1$. Незвідні многочлени в теорії циклічних кодів відіграють роль породжувальних (генераторних або, що створюють) многочленів (поліномів), від виду яких, власне, і залежать основні характеристики отриманого коду: надлишковість і корегувальна здатність. Враховуючи властивість циклічності зсування кодових комбінацій, можна записати породжувальну матрицю циклічного коду у такому вигляді:

$$V = \begin{bmatrix} p(x) \\ p(x)\cdot x - c_2(x^n - 1) \\ p(x)\cdot x^2 - c_3(x^n - 1) \\ \dots \\ p(x)\cdot x^{m-1} - c_m(x^n - 1) \end{bmatrix},$$

де $p(x)$ — початкова (вихідна) кодова комбінація, на основі якої отримано всі інші $(m - 1)$ базові комбінації, $c_i = 0$ або $c_i = 1$ ("0", якщо результуючий

степінь поліномів (многочленів) $p(x) \cdot x^j$ не перевершує $(n-1)$, "1", якщо — перевершує).

Початкова комбінація $p(x)$ називається породжувальною (генераторною) комбінацією. Для побудови циклічного коду досить правильно вибрати початкову комбінацію $p(x)$. Усі інші кодові комбінації подібні до таких самих у груповому коді.

У процесі виправлення помилки з використанням циклічного коду при визначенні положення помилкового біта застосовується залишок многочлена. Як випливає з визначення, в циклічному коді кодові слова представляються у вигляді многочленів

$$v(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + \ldots + v_1x^1 + v_0x^0,$$

де n — довжина коду; v_i — коефіцієнти з поля $GF(q)$.

Якщо код побудований над полем $GF(2)$, то коефіцієнти приймають значення "0" або "1" і код називається двійковим.

Довжина циклічного коду називається ***примітивною*** і сам код називається ***примітивним***, якщо його довжина $n = q^m - 1$ над полем $GF(q)$. Якщо довжина коду менша, ніж довжина примітивного коду, то код називається укороченим або неприміти вним.

Наведемо найважливішу властивість кінцевих полів. Безліч усіх ненульових елементів кінцевого поля утворює групу по операції множення, тобто мультиплікативну групу порядку $q-1$. Розглянемо сукупність елементів мультиплікативної групи, утворену деяким елементом α і всіма його вищими степенями: α^2, α^3 тощо. Оскільки група скінченна, то повинні з'явитися повторення, тобто елементи $\alpha^i = \alpha^j$. Помножимо цю рівність на $(\alpha^i)^{-1} = (\alpha^{-1})^i$, отримаємо $1 = \alpha^{j-i}$. Отже, деяка міра α дорівнює "1".

Найменше додатне число e, таке, що $\alpha^e = 1$, називається порядком елемента α. Сукупність елементів $1, \alpha, \alpha^2, \ldots, \alpha^{e-1}$ утворює підгрупу, оскільки добуток будь-яких двох елементів належить цій сукупності, а елемент, обернений до α^j дорівнює α^{e-j} і теж входить у цю сукупність. Група, яка складається зі всіх степенів одного з її елементів, називається ***циклічною групою***.

З розглянутої властивості скінченних полів випливають два важливі наслідки. Перший з них стверджує, що коренями многочлена $x^{q-1}-1$ є всі $q-1$ ненульові елементи поля $GF(q)$, тобто

$$x^{q-1} + 1 = \prod_{\substack{\alpha = GF(q) \\ \alpha \neq 0}} (x - \alpha).$$

У полі $GF(q)$ елемент α, що має порядок $e = q-1$, називається ***примітивним***. Тому будь-який ненульовий елемент $GF(q)$ є степенем примітивного елемента. Ще одним наслідком з розглянутої властивості є те, що будь-яке скінченне поле $GF(q)$ містить у собі примітивний елемент, тобто мультиплікативна група $GF(q)$ — циклічна.

У SEC використовується ***примітивний*** многочлен, який є незвідним многочленом. Якщо степінь примітивного многочлена дорівнює m, то формується поле $GF(2^m)$, елементом якого є залишок примітивного многочлена.

Таким чином, якщо степінь такого многочлена дорівнює 2^m, то за допомогою його залишків можна описати 2^m станів.

Дозволена кодова комбінація циклічного коду формується так (рис. 28).

На підставі інформаційної послідовності визначають інформаційний многочлен $A(x) = a_{k-1}\,x^m + a_{k-2}\,x^{m-1} + \ldots + 1$, що задає корегувальну здатність, і число контрольних розрядів m, а також початкову кодову комбінацію циклічного коду (n, k) у вигляді многочлена $A_{k-1}(x)$.

1. Якщо степінь породжувального многочлена $G(x)$ дорівнює m, то многочлен початкової кодової (інформаційної) комбінації $A_{k-1}(x)$ множимо на x^m:

$$A_{k-1}(x) \cdot x^m.$$

2. Ділимо отриманий результат на породжувальний поліном $G(x)$. Доповнюємо початкову інформаційну комбінацію, що передається, до дозволеної шляхом додавання до неї залишку $R(x)$ від ділення отриманого в попередньому пункті добутку на породжувальний многочлен:

$$A_{k-1}(x) \cdot x^m / \; G(x) = R(x),$$

де степінь залишку $R(x)$ менший, ніж параметр m.

3. Остаточно дозволена кодова комбінація циклічного коду записується так:

$$A_{n-1}(x) = A_{k-1}(x) \cdot x^m + R(x). \qquad (31)$$

Для виявлення помилок у прийнятій кодовій комбінації досить поділити її на породжувальний многочлен (поліном). Якщо прийнята комбінація дозволена, то залишок від ділення буде нульовим. Ненульовий залишок свідчить про те, що прийнята комбінація має помилки. За виглядом залишку (синдрому) можна в деяких випадках також зробити висновок щодо характеру помилки, її місцезнаходження, а також виправити її.

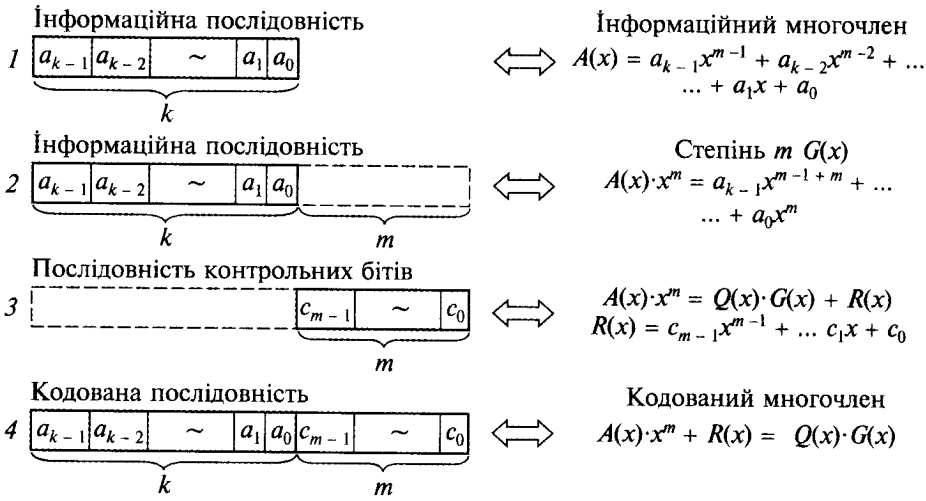


Рис. 28. Схема формування циклічного коду

4. При додаванні залишку $R(x)$ і добутку $A(x) \cdot x^m$ їх сума повинна ділитися на породжувальний многочлен $G(x)$:

$$A(x) \cdot x^m + R(x) = Q(x) \cdot G(x). \qquad (32)$$

Сума $A(x) \cdot x^m + R(x)$ є ***кодованим многочленом***. Звідси зрозумілий сенс породжувального многочлена $G(x)$.

Приклад. Нехай потрібно закодувати комбінацію вигляду 10110111, що відповідає многочлену $A(x) = x^7 + x^5 + x^4 + x^2 + x^1 + 1$.

Визначаємо число контрольних символів $m = 5$. Породжувальним многочленом для комбінації, що передається, є многочлен типу $G(x) = x^5+x^4+x^1+1$, тобто 110011.

Помножимо $A(x)$ на x^m:

$$A(x) \cdot x^m = (x^7+x^5+x^4+x^2+x^1+1) \cdot x^5 = x^{12}+x^{10}+x^9+x^7+x^6+x^5 = 1011011110000.$$

Поділимо отриманий добуток на породжувальний поліном $x^5+x^4+x^1+1$:

$$A(x) \cdot x^m \,/\, P(x) = (x^{12}+x^{10}+x^9+x^7+x^6+x^5) \,/\, (x^5+x^4+x^1+1) = 11010111 + 01001.$$

При діленні необхідно враховувати, що замість звичного арифметичного ділення виконується операція ділення за модулем 2. Залишок $R(x) = 01001$ підсумовуємо з інформаційною послідовністю $x^7 + x^5 + x^4 + x^2 + x^1 + 1$. Як наслідок отримаємо закодоване повідомлення:

$$F(x) = A(x) + R(x) = 1011011101001.$$

У отриманій кодовій комбінації циклічного коду інформаційні символи $A(x) = 10110111$ записуються спочатку, а контрольні $R(x) = 01001$ — за ними. На приймальному боці закодований потік даних, включаючи і залишок $R(x)$, ділиться на породжувальний поліном. Якщо повідомлення прийнято безпомилково, то залишок дорівнює нулю, якщо з помилкою — то ні.

Алгоритм визначення помилки. Нехай маємо n-елементну комбінацію ($n = k + m$), тоді:

1. Отримаємо залишок від ділення $F(x)$, що відповідає помилці в старшому розряді, на породжувальний поліном $G(x)$:

$$F(x) \,/\, G(x) = R_0(x).$$

2. Поділимо цей поліном $H(x)$ знову на $G(x)$, і одержуємо новий поточний залишок $R(x)$.

3. Порівняємо залишки $R_0(x)$ і $R(x)$:

- якщо вони рівні, то помилка відбулася в старшому розряді;
- якщо вони не рівні, то збільшуємо степінь прийнятого полінома на x і знову виконаємо ділення:

$$H(x) \cdot x \,/\, G(x) = R(x).$$

4. Знову порівняємо отриманий залишок з $R_0(x)$:

- якщо вони рівні, то помилки будуть у наступному розряді;
- якщо не рівні, то помножимо $H(x)$ на x^2 і повторюватимемо ці операції доти, доки $R(x)$ не буде дорівнювати $R_0(x)$.

Помилка буде в тому розряді, що відповідає числу, на яке підвищено степінь $H(x)$, плюс одиниця.

Розглянемо властивості ***циклічного коду***. Візьмемо як породжувальний многочлен $G(x)$ рівність

$$G(x) = x^3 + x + 1. \tag{33}$$

Вираз $x^3 + x + 1$ — це незвідний многочлен, що відповідає восьми елементам: 0, 1, x, $x + 1$, x^2, $x^2 + 1$, $x^2 + x$, $x^2 + x + 1$. Степінь многочлена $x^3 + x + 1$ дорівнює 3; множина, елементами якої є залишки цього многочлена, — скінченне поле з 8 елементів. Тобто $GF(8) = GF(2^3)$, а $x^3 + x + 1$ є примітивним многочленом.

Множину, елементами якої є залишки незвідного многочлена, позначимо F_3:

$$F_3 = 0,\ 1,\ x,\ x+1,\ x^2, x^2+1,\ x^2+x,\ x^2+x+1.$$

З множини F_3 виключимо елемент "0", тоді отримаємо множину F_3^*:

$$F_3^* = 1,\ x,\ x+1,\ x^2, x^2+1,\ x^2+x,\ x^2+x+1.$$

Одним, будь-яким елементом F_3^* можна відобразити всі елементи цієї множини. Візьмемо, наприклад, елемент x, у цьому разі:

$$\begin{aligned}
x^0 &= 0\cdot(x^3 + x + 1) && + 1\\
x^1 &= 0\cdot(x^3 + x + 1) && + x\\
x^2 &= 0\cdot(x^3 + x + 1) && + x^2\\
x^3 &= 0\cdot(x^3 + x + 1) && + x + 1\\
x^4 &= x\cdot(x^3 + x + 1) && + x^2 + x\\
x^5 &= (x^2 + 1)\cdot(x^3 + x + 1) && + x^2 + x + 1\\
x^6 &= (x^3 + x + 1)\cdot(x^3 + x + 1) && + x^2 + 1\\
x^7 &= (x^4 + x^2 + x + 1)\cdot(x^3 + x + 1) && + 1
\end{aligned}$$

Оскільки існує взаємно однозначна відповідність:

$$1 \leftrightarrow x^0, x \leftrightarrow x^1, x^2 \leftrightarrow x^2, x+1 \leftrightarrow x^3, x^2+x \leftrightarrow x^4, x^2+x+1 \leftrightarrow x^5, x^2+1 \leftrightarrow x^6,$$

то множину F_3^* можна подати так:

$$F_3^* = \{1,\ x,\ x^2, x^3, x^4, x^5, x^6\}.$$

Наведемо ще одну особливість циклічного коду, а саме:

$$x^7 = x^{2^3-1} = 1. \tag{34}$$

Оскільки у міру збільшення степеня аргумента x відбувається повернення до початкового значення, то $7 = 2^3 - 1$ є періодом множини F_3^* (рис. 29).

Розглянемо кратність многочлена $x^3 + x + 1$. Якщо як многочлен коефіцієнтів $GF(2)$ для $A(x)$ взяти

$$R(x) = A(x) \cdot (x^3 + x + 1), \qquad (35)$$

то одержимо mod $(x^7 + 1)$ для $R(x)$. Це означає, що $R(x)$ не може мати члена більше ніж x^7, а показник степеня $A(x)$ не перевищує 3. Таким чином,

$$A(x) = a_3x^3 + a_2x^2 + a_1x + a_0. \qquad (36)$$

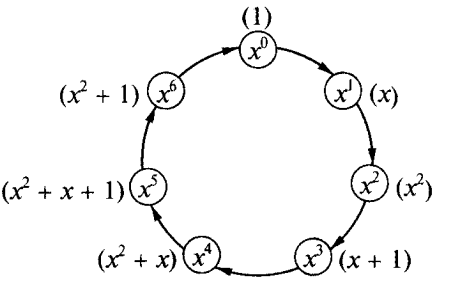


Рис. 29. Приклад періодичності елементів F_3^*

А це, у свою чергу, означає, що довжина інформаційної послідовності кодованого слова, в якому породжувальним є незвідний многочлен типу $x^3 + x + 1$, не може перевищувати 4.

Якщо показник породжувального многочлена дорівнює m, довжина кодованих бітів — n, а довжина інформаційних бітів — k, то існують такі залежності:

$$\begin{aligned} k &= n - m, \\ n &= 2^m - 1. \end{aligned} \qquad (37)$$

Це означає також, що степінь m породжувального многочлена відповідає довжині контрольних бітів m.

Оскільки циклічний код — це код, що отримуємо, ділячи кодований многочлен на породжувальний, то у разі породжувального многочлена $x^3 + x + 1$ і кодованого многочлена $W(x)$ для кодованого слова $(b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ маємо співвідношення

$$W(x) = b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0,$$

яке обов'язково буде ділитися на $x^3 + x + 1$. Тоді і $x \cdot W(x)$ теж, природно, ділитиметься на $x^3 + x + 1$. Тобто $x \cdot W(x)$ теж є кодованим многочленом. Це означає, що має місце відповідність

$$W(x) \leftrightarrow (b_6, b_5, b_4, b_3, b_2, b_1, b_0),$$

$$x \cdot W(x) \leftrightarrow (b_5, b_4, b_3, b_2, b_1, b_0, b_6).$$

Якщо послідовність $(b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ є кодованим словом, то і наступна послідовність $(b_5, b_4, b_3, b_2, b_1, b_0, b_6)$ після циклічної перестановки теж є кодованим словом. Отже, код, отриманий на основі породжувального многочлена $G(x)$, є циклічним кодом.

Розглянемо детально процес виправлення помилки на основі циклічного коду. Нехай породжувальний многочлен має вигляд:

$$G(x) = x^3 + x + 1.$$

Оскільки степінь такого многочлена дорівнює 3, то із залежностей (37) отримуємо циклічний код, у якого:

- довжина кодованих бітів $n = 2^3 - 1 = 7$;
- довжина інформаційних бітів $k = n - m = 4$;
- довжина контрольних бітів $m = 3$.

Фактично отримано код Хеммінга (7, 4), а залежність (37) повністю повторює систему рівнянь (7). Співвідношення (37) демонструють формування коду Хеммінга як циклічного.

Якщо узяти інформаційну послідовність, то одержимо такий інформаційний многочлен:

$$A(x) = a_3x^3 + a_2x^2 + a_1x + a_0.$$

Оскільки степінь породжувального многочлена $x^3 + x + 1$ дорівнює 3, то, помноживши $A(x)$ на x^3, отримаємо

$$A(x) \cdot x^3 = a_3x^6 + a_2x^5 + a_1x^4 + a_0x^3.$$

Поділивши $A(x) \cdot x^3$ на $G(x) = x^3 + x + 1$, знайдемо залишок $R(x)$:

$$\begin{array}{r|l}
 & a_3x^3 + a_2x^2 + (a_3 + a_1)x + (a_3 + a_2 + a_0) \\
\hline
x^3 + x + 1 & a_3x^6 + a_2x^5 + a_1x^4 + a_0x^3 \\
 & \underline{a_3x^6 + a_3x^4 + a_3x^3} \\
 & a_2x^5 + (a_3 + a_1)x^4 + (a_3 + a_0)x^3 \\
 & \underline{a_2x^5 + a_2x^3 + a_2x^2} \\
 & (a_3 + a_1)x^4 + (a_3 + a_2 + a_0)x^3 + a_2x^2 \\
 & \underline{(a_3 + a_1)x^4 + (a_3 + a_1)x^2 + (a_3 + a_1)x} \\
 & (a_3 + a_2 + a_0)x^3 + (a_3 + a_2 + a_1)x^2 + (a_3 + a_1)x \\
 & \underline{(a_3 + a_2 + a_0)x^3 + (a_3 + a_2 + a_0)x + (a_3 + a_2 + a_0)} \\
\text{Залишок } R(x) \rightarrow & (a_3 + a_2 + a_1)x^2 + (a_2 + a_1 + a_0)x + (a_3 + a_2 + a_0)
\end{array}$$

Таким чином, залишок подамо так:

$$R(x) = (a_3 + a_2 + a_1)x^2 + (a_2 + a_1 + a_0)x + (a_3 + a_2 + a_0) = c_2x^2 + c_1x + c_0.$$

Тут

$$c_2 = a_3 + a_2 + a_1 \pmod 2,\ c_1 = a_2 + a_1 + a_0 \pmod 2,\ c_0 = a_3 + a_2 + a_0 \pmod 2.$$

Якщо до $A(x) \cdot x^3$ додати отриманий залишок, то отримаємо кодований за кодом Хеммінга (7, 4) многочлен:

$$V(x) = A(x) \cdot x^3 + R(x) = a_3x^6 + a_2x^5 + a_1x^4 + a_0x^3 + c_2x^2 + c_1x + c_0. \quad (38)$$

Після визначення кодованого многочлена одержимо кодовану послідовність, що передається

$$V = (a_3, a_2, a_1, a_0, c_2, c_1, c_0).$$

Увесь описаний вище процес кодування подано на рис. 30.

Під час передачі по каналу зв'язку кодованої послідовності під впливом завад можуть виникнути помилки коду вигляду: $0 \rightarrow 1$ або $1 \rightarrow 0$. При цьому

1. Інформаційна послідовність (a_3, a_2, a_1, a_0)	→	Інформаційний многочлен $A(x) = a_3x^3 + a_2x^2 + a_1x + a_0$
2. Породжувальний многочлен $G(x) = x^3 + x + 1$	→	Добуток $A(x) \cdot x^3 = a_3x^6 + a_2x^5 + a_1x^4 + a_0x^3$
3. $[A(x) \cdot x^3]/\ G(x)$ Залишок $R(x)$	→	$R(x) = c_2x^2 + c_1x + c_0$ $c_2 = a_3 + a_2 + a_1 \pmod 2$ $c_1 = a_2 + a_1 + a_0 \pmod 2$ $c_0 = a_3 + a_2 + a_0 \pmod 2$
4. Додавання до $A(x) \cdot x^3$ залишка $R(x)$	→	Кодований многочлен $V(x) = A(x) \cdot x^3 + R(x) =$ $= a_3x^6 + a_2x^5 + a_1x^4 + a_0x^3 + c_2x^2 + c_1x + c_0$ ↓ Ділення $V(x)$ на $G(x)$
5. Кодований многочлен $V(x)$	→	Кодована послідовність, що передається $(a_3, a_2, a_1, a_0, c_2, c_1, c_0)$

Рис. 30. Процес кодування для виправлення помилок

кодована послідовність, що приймається, відрізняється від кодованої послідовності, що передавалась. Зміна 0 → 1 або 1 → 0 аналогічна додаванню 1 до біта, що передаються: 1 + 1 = 0 (mod 2), 0 + 1 = 1 (mod 2).

Розглянемо випадок помилки першого біта кодованої послідовності, що передається. Його можна інтерпретувати як додавання "1" до першого біта і додавання "0" до решти всіх інших бітів. Оскільки кодована послідовність, що передається, складається з 7 бітів, то додавання "1" до першого біта і додавання "0" до решти бітів відповідає помилковій послідовності

$$E = (1\ 0\ 0\ 0\ 0\ 0\ 0),$$

яка додається до послідовності V.

Помилкову послідовність можна подати у вигляді помилкового многочлена, аналогічно тому, як кодована послідовність, що передається, відображається кодованим многочленом.

При помилці першого біта отримуємо

$$E_1(x) = x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x + 0 = x^6.$$

Аналогічно, у разі помилок в інших бітах одержуємо

$$E_2(x) = x^5,\ E_3(x) = x^4,\ E_4(x) = x^3,\ E_5(x) = x^2,\ E_6(x) = x,\ E_7(x) = 1.$$

Якщо помилкову послідовність позначити як $(e_6, e_5, e_4, e_3, e_2, e_1, e_0)$, то у разі появи одиничної помилки в лінії передачі з набору елементів e_6—e_0 тільки одна величина буде дорівнювати 1.

Помилковий многочлен подається так:

$$E(x) = e_6 \cdot x^6 + e_5 \cdot x^5 + e_4 \cdot x^4 + e_3 \cdot x^3 + e_2 \cdot x^2 + e_1 \cdot x + e_0.$$

Отже, у разі виникнення помилки коду в лінії передачі прийнята кодована послідовність U має вигляд

$$U = V + E = (a_3, a_2, a_1, a_0, c_2, c_1, c_0) + (e_6, e_5, e_4, e_3, e_2, e_1, e_0) =$$
$$= (a_3 + e_6,\ a_2 + e_5,\ a_1 + e_4,\ a_0 + e_3,\ c_2 + e_2,\ c_1 + e_1,\ c_0 + e_0) \pmod 2.$$

Якщо прийняту послідовність відобразити у вигляді многочлена, що приймається, то отримаємо

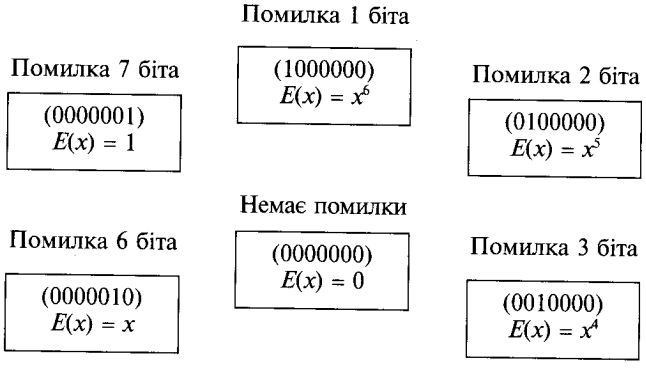
$$U(x) = V(x) + E(x) = (a_3 + e_6)\cdot x^6 + (a_2 + e_5)\cdot x^5 + (a_1 + e_4)\cdot x^4 + (a_0 + e_3)\cdot x^3 +$$
$$+ (c_2 + e_2)\cdot x^2 + (c_1 + e_1)\cdot x + (c_0 + e_0) \pmod 2.$$

Цей процес наведено на рис. 31.

На приймальній стороні прийнята кодована послідовність перевіряється; в ній знаходяться і виправляються помилки.

1. Кодована послідовність, що передається $(a_3, a_2, a_1, a_0, c_2, c_1, c_0)$ $\Leftrightarrow$ Кодований многочлен $V(x) = a_3x^6 + a_2x^5 + a_1x^4 + a_0x^3 + c_2x^2 + c_1x + c_0$

2. Помилкова послідовність $(e_6, e_5, e_4, e_3, e_2, e_1, e_0)$ $\Leftrightarrow$ Помилковий многочлен $E(x) = e_6x^6 + e_5x^5 + e_4x^4 + e_3x^3 + e_2x^2 + e_1x + e_0$



3. Прийнята кодована послідовність $U = V + E = (a_3 + e_6, a_2 + e_5, a_1 + e_4, a_0 + e_3, c_2 + e_2, c_1 + e_1, c_0 + e_0)$ $\Leftrightarrow$ Прийнятий многочлен $U(x) = V(x) + E(x) = (a_3 + e_6,)x^6 + (a_2 + e_5,)x^5 + \ldots + (c_0 + e_0)$

Рис. 31. Схема помилкової, прийнятої і тієї, що передається, кодованих послідовностей

Нехай у лінії передачі з'явилася одинична помилка (наприклад, помилка першого біта в одному кодованому слові). Тоді прийнятий многочлен має вигляд

$$U(x) = V(x) + E(x).$$

Якщо поділити цей многочлен на породжувальний многочлен, то у разі відсутності помилки

$$U(x) = V(x) = Q(x) \cdot G(x)$$

і многочлен $U(x)$ можна поділити на $G(x)$ без залишку ($R(x) = 0$).

У разі появи помилки помилковий многочлен $E(x) \neq 0$ і при діленні многочлена $U(x)$ на $G(x)$ залишок $R(x) \neq 0$. Тоді

$$U(x) = V(x) + E(x) = Q(x) \cdot G(x) + E(x)$$

і при діленні $U(x)$ на $G(x)$ залишок буде дорівнювати залишку, який отримуємо при діленні $E(x)$ на $G(x)$. Отже,

$$U(x) = E(x) \pmod{G(x)}.$$

Визначення залишку $R(x)$ при діленні $E(x)$ на $G(x)$ здійснюється так:

$$\begin{array}{l|l}
 & e_6x^3+e_5x^2+(e_6+e_4)x+(e_6+e_5+e_3) \\
x^3+x+1) & e_6x^6+\ e_5x^5+\ \ e_4x^4+\ \ e_3x^3+\ \ e_2x^2+\ \ e_1x+\ \ e_0 \\
 & e_6x^6+\ \ e_6x^4+\ \ e_6x^3 \\
\hline
 & e_5x^5+(e_6+e_4)x^4+\ (e_6+e_3)x^3+\ \ e_2x^2 \\
 & e_5x^5+\ \ e_5x^3+\ \ e_5x^2 \\
\hline
 & (e_6+e_4)x^4+(e_6+e_5+e_3)x^3+\ (e_5+e_2)x^2+\ \ e_1x \\
 & (e_6+e_4)x^4+\ \ (e_6+e_4)x^2+\ (e_6+e_4)x \\
\hline
 & (e_6+e_5+e_3)x^3+(e_6+e_5+e_4+e_2)x^2+\ (e_6+e_4+e_1)x+\ \ e_0 \\
 & (e_6+e_5+e_3)x^3+\ \ (e_6+e_5+e_3)x+(e_6+e_5+e_3) \\
\hline
R(x) \rightarrow & (e_6+e_5+e_4+e_2)x^2+(e_5+e_4+e_3+e_1)x+(e_6+e_5+e_3+e_0)
\end{array}$$

Тобто залишок $R(x) = (e_6+e_5+e_4+e_2)x^2+(e_5+e_4+e_3+e_1)x+(e_6+e_5+e_3+e_0)$. Серед бітів e_6—e_0 помилкової послідовності тільки один біт дорівнює 1, а всі інші дорівнюють 0. У табл. 7 відображена залежність між помилковим бітом і залишком від ділення.

Таблиця 7. Залежність між помилковим бітом, залишком та степенем елемента x поля $GF(2^3)$

Помилковий біт	Залишок $R(x)$ від $G(x)$	Степінь елемента x $GF(2^3)$
Відсутній (e_6—$e_0 = 0$)	0	—
Перший $e_6 = 1$ (інші дорівнюють 0)	$x^2 + 1$	x^6
Другий $e_5 = 1$ (інші дорівнюють 0)	$x^2 + x + 1$	x^5
Третій $e_4 = 1$ (інші дорівнюють 0)	$x + 1$	x^4
Четвертий $e_3 = 1$ (інші дорівнюють 0)	$x^2 + x$	x^3
П'ятий $e_2 = 1$ (інші дорівнюють 0)	x^2	x^2
Шостий $e_1 = 1$ (інші дорівнюють 0)	x	x^1
Сьомий $e_6 = 1$ (інші дорівнюють 0)	1	x^0

Із табл. 7 видно, що між залишком і положенням помилкового біта існує взаємно однозначна відповідність:

$R(x) = x^2 + 1 \rightarrow x^6$ помилка 1-го біта,
$R(x) = x^2 + x + 1 \rightarrow x^5$ помилка 2-го біта,
$R(x) = x + 1 \rightarrow x^4$ помилка 3-го біта,
$R(x) = x^2 + x \rightarrow x^3$ помилка 4-го біта,
$R(x) = x^2 \rightarrow x^2$ помилка 5-го біта,
$R(x) = x \rightarrow x^1$ помилка 6-го біта,
$R(x) = 1 \rightarrow x^0$ помилка 7-го біта,
$R(x) = 0 \rightarrow$ помилка відсутня.

Таким чином, за залишком $R(x)$ можна визначити положення помилкового біта і, додавши до нього одиницю, виправити помилку.

Помилку можна виправити автоматично, якщо використати періодичність залишку $R(x)$ (див. рис. 29).

Розділ 8

КОДЕКИ ЦИКЛІЧНОГО КОДУ

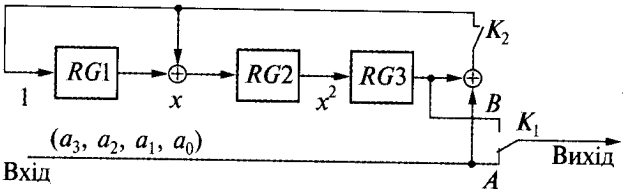
У разі передачі, наприклад, аналогового аудіо- чи відеосигналу, що є широкосмуговим сигналом, останній має бути конвертований у вузькосмугову форму так, щоб його можна було передати за допомогою цифрової системи. Цей процес називається аналого-цифровим перетворенням і здійснюється в пристрої, що називається кодек (кодер і декодер).

Кодек — пристрій або спеціальна комп'ютерна програма — програмний модуль, що виконує перетворення потоку даних або сигналу для подальшого їх використання. Кодеки можуть як кодувати потік або сигнал (для передачі, зберігання або шифрування), так і розкодовувати (для перегляду або зміни у форматі, що більше відповідає цим операціям). Тобто вони перетворюють аналогові сигнали у цифровий код і, навпаки, цифровий код в аналогові сигнали.

Кожен кодек використовує свої — або оригінальні, або специфічні — алгоритми стиснення даних і, як правило, не сумісний з даними, які були стиснені іншим кодеком. Кодеки часто застосовуються при цифровій обробці відеосигналу і звуку. Останні розробки кодеків називають ***кофідеками***, тому що вони об'єднують функції кодека, що полягають в аналого-цифровому та цифро-аналоговому перетворенні, й фільтрації на прийомі та передачі в одній великій інтегральній мікросхемі. Фільтри на вході і виході виконують функції обмеження смуги частот, подавлення шумів, запобігання накладенню спектрів.

Аудіо- і відеосигнали вимагають спеціалізованих методів стиснення. Інженери і математики випробували безліч теоретичних та практичних способів для вирішення цієї проблеми.

Більшість кодеків для звукових і візуальних даних використовують, щоб одержувати прийнятний розмір готового (стисненого) файла. Існують також кодеки, що стискають без втрат, але у багатьох випадках малопомітне поліпшення якості сигналу не виправдовує істотного збільшення обсягу даних. Майже єдиним винятком є ситуація, коли дані будуть піддаватися подальшій обробці: тоді повторювані втрати при кодуванні (декодуванні) вплинуть на якість аудіо- та відеосигналів. Така обробка складається із чотирьох послідовних процесів: фільтрації, дискретизації, квантування та кодування. Зокрема, для звукового сигналу, смуга частот якого обмежена рівнем 4000 Гц, здійснюється дискретизація з частотою в 2 рази більшою (8000 відліків у секунду). Це частота дискретизації Найквіста — вона потрібна для забезпечення якісного відновлення широкосмугового сигналу із його цифрового подання у приймачі. Кожний відлік сигналу виражається, як мінімум, 8-ми розрядним дискретним двійковим кодом. Отримані квантовані сигнали кодуються з використанням спеціального кодового набору. Як приклад, зручно знову використовувати код Хеммінга (див. рис. 30).



Синхро-імпульси	Вхідний сигнал	Вихід *RG*1	Вихід *RG*2	Вихід *RG*3	Вихід кодера	Перемикач *K*
0	$x^3 \to a_3$	0	0	0	$a_3 \leftarrow x^6$	
1	$x^2 \to a_2$	a_3	a_3	0	$a_2 \leftarrow x^5$	$K_1 \to A$
2	$x^1 \to a_1$	a_2	$a_3 + a_2$	a_3	$a_1 \leftarrow x^4$	$K_2 \to$ ВКЛ
3	$x^0 \to a_0$	$a_3 + a_1$	$a_3 + a_2 + a_1$	$a_3 + a_2$	$a_0 \leftarrow x^3$	
4	0	$a_3 + a_2 + a_0$	$a_2 + a_1 + a_0$	$a_3 + a_2 + a_1$	$a_3 + a_2 + a_1 \leftarrow x^2$	
5	0	0	$a_3 + a_2 + a_0$	$a_2 + a_1 + a_0$	$a_2 + a_1 + a_0 \leftarrow x^1$	$K_1 \to B$
6	0	0	0	$a_3 + a_2 + a_0$	$a_3 + a_2 + a_0 \leftarrow x^0$	$K_2 \to$ ВИКЛ

Рис. 32. Схема формування коду Хеммінга (7, 4)

Для апаратної реалізації процедури кодування на стороні, що передає, необхідно мати схеми множення x^3 і ділення на $x^3 + x + 1$. На рис. 32 проілюстрована функціональна схема формування коду Хеммінга (7, 4), в якій помножувальний і ділильний пристрої об'єднані в одне ціле.

Під час надходження на вхід кодувального пристрою інформаційної послідовності (див. рис. 32), на його виході виробляється передавана кодована послідовність $V = (a_3, a_2, a_1, a_0, c_2, c_1, c_0)$. До неї в лінії передачі додається помилкова послідовність, в якій тільки один член дорівнює 1, і, як наслідок, на виході реалізується послідовність, що приймається:

$$U = V + E = (a_3 + e_6,\ a_2 + e_5,\ a_1 + e_4,\ a_0 + e_3,\ c_2 + e_2,\ c_1 + e_1,\ c_0 + e_0) \pmod 2.$$

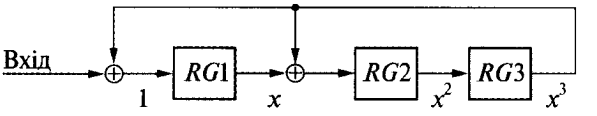
Прийнятий многочлен

$$U(x) = V(x) + E(x) = (a_3 + e_6)\cdot x^6 + (a_2 + e_5)\cdot x^5 + (a_1 + e_4)\cdot x^4 + (a_0 + e_3)\cdot x^3 + \\ + (c_2 + e_2)\cdot x^2 + (c_1 + e_1)\cdot x + (c_0 + e_0) \pmod 2,$$

який відображає прийняту кодовану послідовність, ділять на породжувальний многочлен $G(x) = x^3 + x + 1$ і досліджують залишок для визначення наявності помилки. Якщо помилка є, то визначають її положення, якщо помилки немає, то залишок дорівнює 0 (рис. 33).

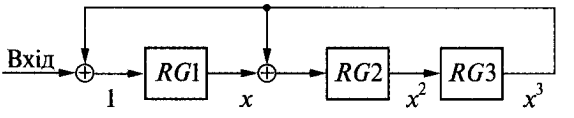
Оскільки залишок від ділення переданої кодованої послідовності на $G(x)$ дорівнює 0, то залишок від ділення прийнятої кодованої послідовності на $G(x)$ буде таким самим, що і залишок від ділення помилкової послідовності на $G(x)$.

Помилкову послідовність подають на вхід схеми ділення і досліджують її (рис. 34).



Синхро-імпульс	Вхідний сигнал	Вихід *RG*1	Вихід *RG*2	Вихід *RG*3
0	$x^6 \to a_3$	0	0	0
1	$x^5 \to a_2$	a_3	0	0
2	$x^4 \to a_1$	a_2	a_3	0
3	$x^3 \to a_0$	a_1	a_2	a_3
4	$x^2 \to a_3 + a_2 + a_1$	$a_3 + a_0$	$a_3 + a_1$	a_2
5	$x^1 \to a_2 + a_1 + a_0$	$a_3 + a_1$	$a_3 + a_2 + a_0$	$a_3 + a_1$
6	$x^0 \to a_3 + a_2 + a_0$	$a_3 + a_2 + a_0$	0	$a_3 + a_2 + a_0$
7	0	0	0	0
		(1)	(x)	(x^2)
		Залишок		

Рис. 33. Схема ділення на $G(x) = x^3 + x + 1$



Синхро-імпульс	Вхідний сигнал	Вихід *RG*1	Вихід *RG*2	Вихід *RG*3
0	$x^6 \to e_6$	0	0	0
1	$x^5 \to e_5$	e_6	0	0
2	$x^4 \to e_4$	e_5	e_6	0
3	$x^3 \to e_3$	e_4	e_5	e_6
4	$x^2 \to e_2$	$e_6 + e_3$	$e_6 + e_4$	e_5
5	$x^1 \to e_1$	$e_5 + e_2$	$e_6 + e_4 + e_3$	$e_6 + e_4$
6	$x^0 \to e_0$	$e_6 + e_4 + e_1$	$e_6 + e_5 + e_4 + e_2$	$e_6 + e_5 + e_3$
7	0	(1)	(x)	(x^2)
		$e_6 + e_5 + e_3 + e_0$	$e_5 + e_4 + e_3 + e_1$	$e_6 + e_5 + e_4 + e_2$
		Залишок		

Рис. 34. Схема отримання залишку від ділення помилкової послідовності на $G(x)$

Із рис. 34 видно, що залишок можна записати так:

$$R(x) = (e_6 + e_5 + e_4 + e_2) \cdot x^2 + (e_5 + e_4 + e_3 + e_1) \cdot x + (e_6 + e_5 + e_3 + e_0).$$

Цей результат аналогічний отриманому раніше в розд. 7, і відповідає положенню помилкового біта.

Помилковий біт є інверсією $\overline{a}$ правильного біта, тому для відновлення останнього досить до помилкового біта додати 1:

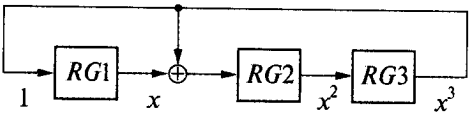
$$\overline{a} + 1 = a \pmod 2.$$

Існують різні схеми для виправлення помилкового біта при використанні коду з 7 бітів. Найзручніше, проте, скористатися періодичністю елементів $GF(2^3)$. Періодичність залишку породжувального многочлена $G(x)= x^3 + x + 1$ відповідає mod $x^2 + x + 1$, унаслідок чого при будь-якому переміщенні елементів у схемі ділення на $x^3 + x + 1$ повинна спостерігатися періодичність елементів.

На рис. 35 наведено схему ділення без входу.

Якщо в зсувні регістри заносяться нулі, то вони зберігатимуться в регістрах скільки завгодно довго. Нехай в зсувні регістри заносяться наступні значення: $RG1 = 1$, $RG\,2 = 0$, $RG\,3 = 0$. Такий стан можливий тільки в тому випадку, якщо залишок $R(x) = 1$. На рис. 35 наведено математичні вирази для залишків, що отримують при зміні стану схеми. У цьому разі залишок є періодичним. Від моменту появи залишку $R(x) = 1$ до моменту, коли він знову з'явиться, проходить 7 тактів (синхроімпульсів). Іншими словами, за час проходження 7 синхроімпульсів з'являються всі елементи множини $GF(2^3)$, за винятком елемента "0". Ці елементи з'являються в такій послідовності: x, x^2, x^3, x^4, x^5, x^6, $x^7 = 1$. При такій роботі схеми ділення на $x^3 + x + 1$, коли в неї послідовно заносяться всі залишки, крім 0, після деякого синхроімпульсу обов'язково з'явиться 1. Оскільки між залишком і помилковим бітом існує залежність:

$e_0 \to 1 \quad \to x^0 \quad x^7 = x^7$ (після 7-го синхроімпульса)
$e_1 \to x \quad \to x^1 \quad x^6 = x^7$ (після 6-го синхроімпульса)
$e_2 \to x^2 \quad \to x^2 \quad x^5 = x^7$ (після 5-го синхроімпульса)
$e_3 \to x+1 \quad \to x^3 \quad x^4 = x^7$ (після 4-го синхроімпульса)
$e_4 \to x^2+x \quad \to x^4 \quad x^3 = x^7$ (після 3-го синхроімпульса)
$e_5 \to x^2+x+1 \quad \to x^5 \quad x^2 = x^7$ (після 2-го синхроімпульса)
$e_6 \to x^2+1 \quad \to x^6 \quad x^1 = x^7$ (після 1-го синхроімпульса)
$1 \quad \to x^7$



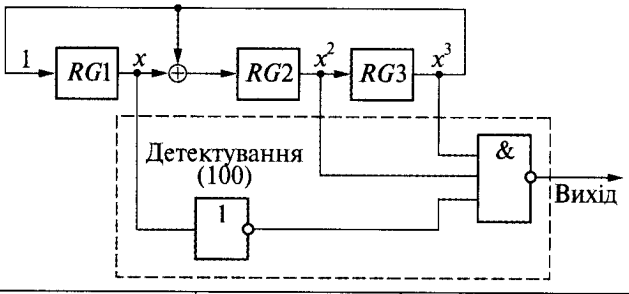
Синхроімпульс	Вихід $RG1$	Вихід $RG2$	Вихід $RG3$	Залишок	Степінь x $GF(2^3)$
0	1	0	0	1	$1 = (x^0)$
1	0	1	0	x	x
2	0	0	1	x^2	x^2
3	1	1	0	$x + 1$	x^3
4	0	1	1	$x^2 + x$	x^4
5	1	1	1	$x^2 + x + 1$	x^5
6	1	0	1	$x^2 + 1$	x^6
7	1	0	0	1	$1 = (x^0)$

Рис. 35. Схема ділення на $x^3 + x + 1$ без входу

то при занесенні залишку в схему ділення на $x^3 + x + 1$($RG1$, $RG2$, $RG3$) = (1 0 0), стає можливим виправлення помилкового біта за умови, що біт виправлення помилки збігається за часом з бітом, в якому відбулася помилка. Якщо перевірити вихід схеми детектування початкового стану (1 0 0) при занесенні в схему ділення на $x^3 + x + 1$ залишку, який отримали від ділення помилкової послідовності на $G(x)$, то буде реалізована ситуація, зображена на рис. 36. При послідовному розташуванні бітів (e_6, e_5, e_4, e_3, e_2, e_1, e_0) і поєднанні в часі з прийнятою кодованою послідовністю за допомогою зсувних регістрів, на виході схеми детектування початкового стану є і сигнал виправлення помилки. Подібна схема виправлення помилки для коду Хеммінга (7, 4) наведена на рис. 37.

При безперервному надходженні кодів використовують декілька однакових схем декодування, перемикаючи їх відповідним чином. Інакше, можна використовувати дві схеми ділення (рис. 38). У цьому випадку після детектування залишку його заносять в нижню ділильну схему, а у верхню ділильну схему заносять "0". При детектуванні залишку наступного коду у верхній схемі декодера Хеммінга, в його нижній схемі виконується перевірка помилкової послідовності і виправляються помилкові біти.

Розглянуті принципи і пристрої можуть застосовуватися для будь-яких кодів Хеммінга (n, k). Істотне спрощення схем кодування і декодування досягається при використанні постійного запам'ятовувального пристрою (ПЗП). Якщо інформаційну послідовність використовувати як адресу, а кодовану послідовність розмістити в елементах пам'яті ПЗП, який виготовляється для виконання певних, стандартних для цифрової техніки функцій та в якому інформація записується або при його виготовленні, або після виготовлення

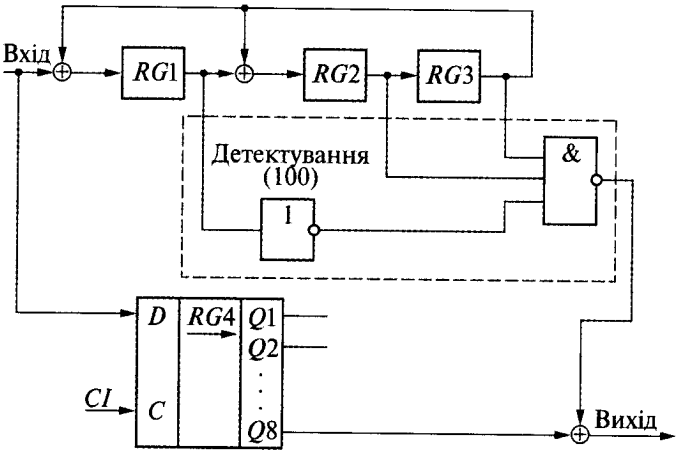


Синхроімпульс	Вихід $RG1$	Вихід $RG2$	Вихід $RG3$	Вихід детектора (100)
0	$e_6 + e_5 + e_3 + e_0$	$e_5 + e_4 + e_3 + e_1$	$e_6 + e_5 + e_4 + e_2$	—
1	$\underline{e_6} + e_5 + e_4 + e_2$	$e_4 + e_3 + e_2 + e_0$	$e_5 + e_4 + e_3 + e_1$	e_6
2	$\underline{e_5} + e_4 + e_3 + e_1$	$e_6 + e_3 + e_2 + e_1$	$e_4 + e_3 + e_2 + e_0$	e_5
3	$\underline{e_4} + e_3 + e_2 + e_0$	$e_5 + e_2 + e_1 + e_0$	$e_6 + e_3 + e_2 + e_1$	e_4
4	$e_6 + \underline{e_3} + e_2 + e_1$	$e_6 + e_4 + e_1 + e_0$	$e_5 + e_2 + e_1 + e_0$	e_3
5	$e_5 + \underline{e_2} + e_1 + e_0$	$e_6 + e_5 + e_3 + e_0$	$e_6 + e_4 + e_1 + e_0$	e_2
6	$e_6 + e_4 + e_1 + \underline{e_0}$	$e_6 + e_5 + e_4 + e_2$	$e_6 + e_5 + e_3 + e_0$	e_1
7	$e_6 + e_5 + e_3 + \underline{e_0}$	$e_5 + e_4 + e_3 + e_1$	$e_6 + e_5 + e_4 + e_2$	e_0

У послідовності e_6—e_0 тільки один біт дорівнює "1"

Рис. 36. Схема детектування початкового стану ділення на $x^3 + x + 1$

5*



Синхро-імпульс	Вхід	RG4 Q1	RG4 Q2	RG4 Q3	RG4 Q4	RG4 Q5	RG4 Q6	RG4 Q7	RG4 Q8	RG1	RG2	RG3	Вихід детектора (100)	Вихід
0	$x^6 \rightarrow a_3 + e_6$	0	0	0	0	0	0	0	0	0	0	0	—	—
1	$x^5 \rightarrow a_2 + e_5$	$a_3 + e_6$	0	0	0	0	0	0	0	e_6	0	0	—	—
2	$x^4 \rightarrow a_1 + e_4$	$a_2 + e_5$	$a_3 + e_6$	0	0	0	0	0	0	e_5	e_6	0	—	—
3	$x^3 \rightarrow a_0 + e_1$	$a_1 + e_4$	$a_2 + e_5$	$a_1 + e_6$	0	0	0	0	0	e_4	e_5	e_6	—	—
4	$x^2 \rightarrow c_2 + e_2$	$a_0 + e_3$	$a_1 + e_4$	$a_2 + e_5$	$a_3 + e_6$	0	0	0	0	$e_6 + e_3$	$e_6 + e_4$	e_5	—	—
5	$x^1 \rightarrow c_1 + e_1$	$c_2 + e_2$	$a_0 + e_3$	$a_1 + e_4$	$a_2 + e_5$	$a_3 + e_6$	0	0	0	$e_6 + e_2$	$e_6 + e_5 + e_5$	$e_6 + e_4$	—	—
6	$x^0 \rightarrow c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$a_0 + e_1$	$a_1 + e_4$	$a_2 + e_5$	$a_3 + e_6$	0	0	$e_6 + e_4 + e_1$	$e_6 + e_5 + e_4 + e_2$	$e_6 + e_5 + e_1$	—	—
7	0	$c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$a_0 + e_3$	$a_1 + e_4$	$a_2 + e_5$	$a_3 + e_6$	0	$e_6 + e_5 + e_3 + e_0$	$e_5 + e_4 + e_5 + e_1$	$e_6 + e_5 + e_4 + e_2$	—	—
8	0	0	$c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$a_0 + e_3$	$a_1 + e_4$	$a_2 + e_5$	$a_3 + e_6$	$e_6 + e_5 + e_2 + e_1$	$e_4 + e_5 + e_2 + e_0$	$e_5 + e_4 + e_3 + e_1$	e_6	$a_3 \leftarrow x^6$
9	0	0	0	$c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$a_0 + e_3$	$a_1 + e_4$	$a_2 + e_5$	$e_6 + e_4 + e_3 + e_1$	$e_6 + e_5 + e_2 + e_1$	$e_4 + e_3 + e_2 + e_0$	e_5	$a_2 \leftarrow x^5$
10	0	0	0	0	$c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$a_0 + e_5$	$a_1 + e_4$	$e_4 + e_3 + e_2 + e_0$	$e_6 + e_2 + e_1 + e_0$	$e_6 + e_3 + e_2 + e_1$	e_4	$a_1 \leftarrow x^4$
11	0	0	0	0	0	$c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$a_0 + e_3$	$e_6 + e_3 + e_2 + e_1$	$e_6 + e_4 + e_1 + e_0$	$e_6 + e_2 + e_1 + e_0$	e_3	$a_0 \leftarrow x^3$
12	0	0	0	0	0	0	$c_0 + e_0$	$c_1 + e_1$	$c_2 + e_2$	$e_6 + e_2 + e_1 + e_0$	$e_6 + e_5 + e_4 + e_2$	$e_6 + e_4 + e_1 + e_0$	e_2	$c_2 \leftarrow x^2$
13	0	0	0	0	0	0	0	$c_0 + e_0$	$c_1 + e_1$	$e_6 + e_4 + e_1 + e_0$	$e_6 + e_5 + e_4 + e_2$	$e_6 + e_5 + e_4 + e_2$	e_1	$c_1 \leftarrow x^1$
14	0	0	0	0	0	0	0	0	$c_0 + e_0$	$e_6 + e_5 + e_2 + e_0$	$e_5 + e_4 + e_2 + e_1$	$e_6 + e_5 + e_4 + e_2$	e_0	$c_0 \leftarrow x^0$

У послідовності e_6—e_0 тільки один біт дорівнює "1"

Тільки для розрахунку помилкового біта

Рис. 37. Схема декодера коду Хеммінга (7, 4)

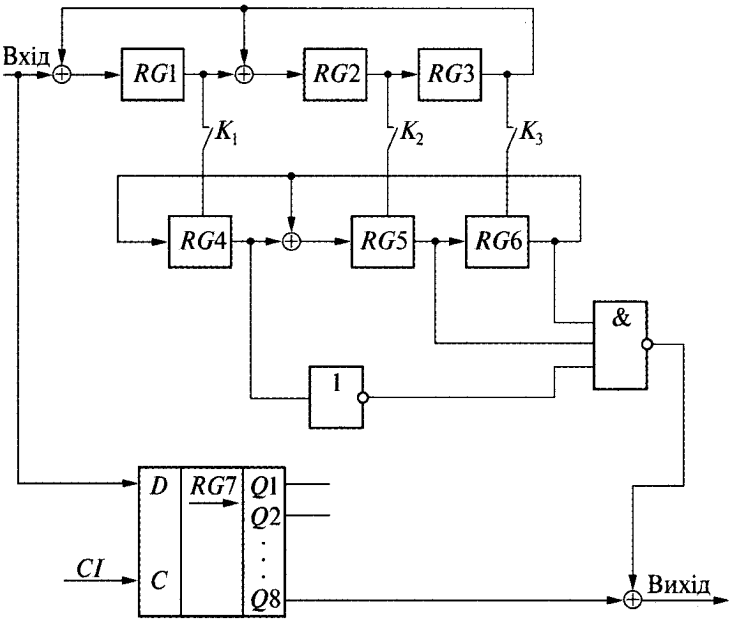


Рис. 38. Схема декодера Хеммінга (7, 4) з двома дільними схемами

всієї інтегральної схеми (так звані програмовані ПЗП), то на виходах можна відразу отримати необхідну (дійсну) кодовану послідовність (рис. 39). Вміст ПЗП, в якому, на відміну від оперативного запам'ятовувального пристрою, записується не один біт, а ціле слово, довжиною 4 або 8 біт (тому воно має декілька інформаційних виходів), формується у вигляді таблиці програмування. Основним параметром ПЗП є інфомаційна ємність інтегральної схеми, наприклад позначення 1024×8 біт означає, що вона містить у собі 1024 слова довжиною 8 біт кожне.

Аналогічно, в процесі декодування при подачі на адресні входи кодованої послідовності, що приймається, на виходах ПЗП отримують виправлену прийняту послідовність.

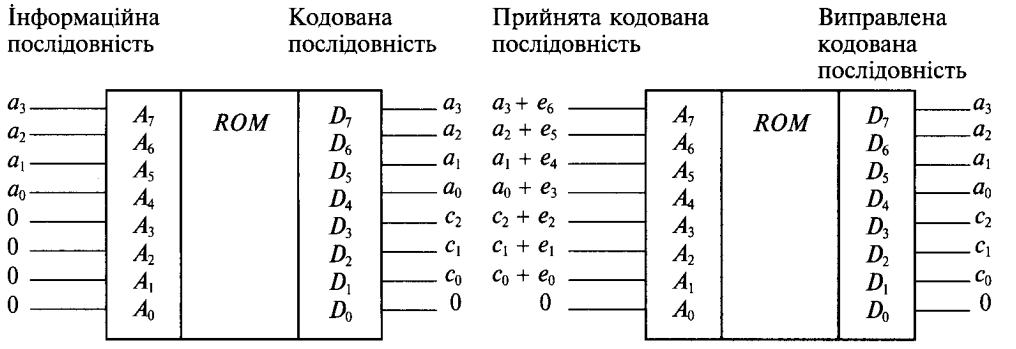


Рис. 39. Схема кодування та декодування з використанням постійного запам'ятовувального пристрою

КОРЕГУВАЛЬНИЙ КОД ХЕММІНГА

Ефективність кодів визначається кількістю помилок, які той може виправити, кількістю надлишкової інформації, додавання якої потрібно для цього, а також складністю реалізації кодування і декодування (як апаратно, так і у вигляді програми). Корегувальна здатність коду кількісно може бути визначена вірогідністю виявлення або виправлення помилок різних типів. Розробка кодів, що мають максимальну корегувальну здатність при заданій надлишковості, а також кодів, що забезпечують задану корегувальну здатність при мінімальній надлишковості — одне з найважливіших завдань теорії кодування. ***Корегувальні коди*** застосовуються при передачі і обробці інформації в обчислювальній техніці, телеграфії, телефонії, телемеханіці та техніці зв'язку, де можливі спотворення сигналу унаслідок дії різного роду завад. Під ***здатністю корегувального коду*** розуміється його властивість виявляти і (або) виправляти помилку максимальної кратності q. Ця здатність пов'язана з його кодовою відстанню. ***Відстанню d_{ij} між кодами*** (кодовими комбінаціями) i та j називається число різних розрядів в кодових комбінаціях i та j. Наприклад, якщо є коди 01 і 10, відстань між ними дорівнює 2: вони розрізняються в двох розрядах.

Код Хеммінга будується так: до інформаційних розрядів кодової комбінації додається обчислена за формулою (37) кількість контрольних розрядів, які формуються шляхом підрахунку парності суми одиниць для визначених груп інформаційних розрядів. При прийомі такої кодової комбінації з отриманих інформаційних і контрольних розрядів шляхом аналогічних підрахунків парності складають корегувальне число, яке дорівнює нулю за відсутності помилки, або вказує номер помилкового розряду.

Розглянемо породжувальний многочлен для контролю парності. Для інформаційної послідовності $A = (a_3, a_2, a_1, a_0)$ многочлен $A(x)$ матиме вигляд

$$A(x) = a_3x^3 + a_2x^2 + a_1x + a_0. \qquad (39)$$

Оскільки у разі контролю парності залишок від ділення $A(x)$ дорівнює 0 або 1, то незвідний многочлен має вигляд $x + 1$. Такий многочлен використовується як породжувальний многочлен для контролю парності. Тобто код для контролю парності є одним з видів циклічного коду, для якого можна використовувати описані вище схеми.

На передавальній стороні кодування здійснюється в такому порядку:

1) многочлен $A(x)$ множать на x і визначають залишок R від породжувального многочлена $G(x) = x + 1$. Тобто формують співвідношення $A(x) \cdot x = Q(x) \cdot (x + 1) + R$, де R дорівнює 1 або 0. Звідси знаходять залишок R:

$$\begin{array}{rl} & a_3x^2+(a_3+a_2)x+(a_3+a_2+a_1) \quad \leftarrow \text{Частка } Q(x) \\ x+1) & a_3x^3+\quad a_2x^2+\quad a_1x+\quad a_0 \\ & a_3x^3+\quad a_3x^2 \\ & \hline (a_3+a_2)x^2+\quad a_1x \\ & (a_3+a_2)x^2+\quad (a_3+a_2)x \\ & \hline (a_3+a_2+a_1)x+\quad a_0 \\ & (a_3+a_2+a_1)x+(a_3+a_2+a_1) \\ & \hline (a_3+a_2+a_1+a_0) \quad \leftarrow \text{Залишок } R. \end{array}$$

Отже, $R = a_3 + a_2 + a_1 + a_0 \pmod 2$;

2) із суми $A(x)\cdot x + R$ формують кодований многочлен:

$$A(x)\cdot x + R = a_3x^4 + a_2x^3 + a_1x^2 + a_0x + (a_3 + a_2 + a_1 + a_0) = \\ = a_3x^4 + a_2x^3 + a_1x^2 + a_0x + c_0, \qquad (40)$$

де $c_0 = a_3 + a_2 + a_1 + a_0 \pmod 2$.

Схема кодування для контролю парності реалізується за допомогою елементів множення на x і ділення на $x + 1$ (рис. 40).

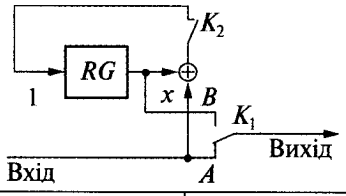
Такий контроль парності всіх бітів називається ***загальним контролем парності***.

Після проходження сигналу по каналу передачі до кодованої послідовності додається помилкова послідовність (e_4, e_3, e_2, e_1, e_0), в якій тільки 1 біт може дорівнювати "1" (помилковий розряд). Тобто передбачається можливість одиничної помилки. Тоді кодована послідовність, що приймається, може бути записана так:

$$U = V + E = (a_3 + e_4,\ a_2 + e_3,\ a_1 + e_2,\ a_0 + e_1,\ c_0 + e_0) \pmod 2.$$

Отже, многочлен $U(x)$, що приймається, має такий вигляд:

$$U(x) = V(x) + E(x) = (a_3 + e_4)\cdot x^4 + (a_2 + e_3)\cdot x^3 + (a_1 + e_2)\cdot x^2 + (a_0 + e_1)\cdot x + \\ + (c_0 + e_0) \pmod 2.$$



Синхроімпульс	Вхід	Вихід *RG*	Вихід	Стан ключів
0	$x^3 \to a_3$	0	$a_3 \leftarrow x^4$	
1	$x^2 \to a_2$	a_3	$a_2 \leftarrow x^3$	$K_1 \to A$
2	$x^1 \to a_1$	$a_3 + a_2$	$a_1 \leftarrow x^2$	$K_2 \to$ ВКЛ
3	$x^0 \to a_0$	$a_3 + a_2 + a_1$	$a_0 \leftarrow x^1$	
4	0	$a_3 + a_2 + a_1 + a_0$	$a_3 + a_2 + a_1 + a_0 \leftarrow x^0$	$K_1 \to B$ $K_2 \to$ ВИКЛ

Рис. 40. Схема кодування для контролю парності

Залишок від ділення $U(x)$ на $G(x) = x + 1$ дорівнює залишку від ділення $E(x)$ на $G(x)$:

$$
\begin{array}{l}
\qquad\qquad e_4x^3+(e_4+e_3)x^2+(e_4+e_3+e_2)x+(e_4+e_3+e_2+e_1) \qquad\qquad \leftarrow \text{Частка} \\
x+1)\ e_4x^4+ \quad e_3x^3+ \quad e_2x^2+ \quad e_1x+ \quad e_0 \\
\quad\ e_4x^4+ \quad e_4x^3 \\
\hline
\qquad (e_4+e_3)x^3+ \quad e_2x^2 \\
\qquad (e_4+e_3)x^3+ \quad (e_4+e_3)x^2 \\
\hline
\qquad\qquad (e_4+e_3+e_2)x^2+ \quad e_1x \\
\qquad\qquad (e_4+e_3+e_2)x^2+ \quad (e_4+e_3+e_2)x \\
\hline
\qquad\qquad\qquad (e_4+e_3+e_2+e_1)x+ \quad e_0 \\
\qquad\qquad\qquad (e_4+e_3+e_2+e_1)x+(e_4+e_3+e_2+e_1) \\
\hline
\qquad\qquad\qquad\qquad\qquad (e_4+e_3+e_2+e_1+e_0) \quad \leftarrow \text{Залишок.}
\end{array}
$$

Таким чином, залишок $R = e_4+e_3+e_2+e_1+e_0 \pmod 2$.

Оскільки серед e_4—e_0 тільки один член може дорівнювати одиниці, то, якщо $R = 1$ — це свідчить про наявність помилки, якщо ж $R = 0$ — помилки немає.

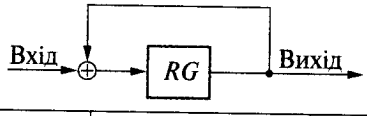
Декодер при загальному контролі парності реалізується за допомогою схеми ділення на $x + 1$ (рис. 41).

Код Хеммінга (7, 4) є кодом SEC. Проте, якщо збільшити його міжкодову відстань, то можна отримати код SEC.DED. Інакше кажучи, з породжувального многочлена коду Хеммінга (7, 4) і породжувального многочлена загального контролю парності можна сформувати коректувальний код Хеммінга (7, 3) SEC.DED.

Доведемо це. Породжувальний многочлен $G_1(x)$ коду Хеммінга (7, 4) має вигляд $G_1(x) = x^3 + x + 1$, а породжувальний многочлен $G_2(x)$ загального контролю парності — $G_2(x) = x + 1$. Знайдемо код, породжувальним многочленом якого є многочлен, що дорівнює добутку многочленів $G_1(x)$ і $G_2(x)$:

$$G(x) = G_1(x) \cdot G_2(x) = (x^3 + x + 1)(x + 1) = x^4 + x^3 + x^2 + 1. \quad (41)$$

Період многочлена $G(x)$ визначається мінімальним значенням n у виразі $x^n + 1$, який є дільником даного многочлена. Оскільки період $G_1(x) = x^3 + x + 1$



Синхроімпульс	Вхід	Вихід *RG*	Вихід
0	$x^4 \to e_4$	0	—
1	$x^3 \to e_3$	e_4	—
2	$x^2 \to e_2$	$e_4 + e_3$	—
3	$x^1 \to e_1$	$e_4 + e_3 + e_2$	—
4	$x^0 \to e_0$	$e_4 + e_3 + e_2 + e_1$	—
5	0	$e_4 + e_3 + e_2 + e_1 + e_0$	$e_4 + e_3 + e_2 + e_1 + e_0$

У послідовності e_4—e_1 тільки один член може дорівнювати "1".

Рис. 41. Схема декодера при контролі парності

дорівнює $2^3 - 1 = 7$, то його можна поділити на $x^7 + 1$. Отже, многочлен $G_1(x)$ є множником $x^7 + 1$. Як бачимо, $x^7 + 1$ ділиться також на $G_2(x) = x + 1$.

Поділимо тепер $x^7 + 1$ на добуток многочленів $G_1(x)$ і $G_2(x)$, тобто на вираз (41):

$$
\begin{array}{r|l}
 & x^3+x^2+\quad\quad 1 \\
\hline
x^4+x^3+x^2+1) & x^7+\quad\quad\quad 1 \\
 & x^7+x^6+x^5+\quad x^3 \\
\hline
 & \quad x^6+x^5+\quad x^3 \\
 & \quad x^6+x^5+x^4+\quad x^2 \\
\hline
 & \quad\quad\quad x^4+\ x^3+x^2+\ 1 \\
 & \quad\quad\quad x^4+x^3+x^2+\ 1 \\
\hline
 & \quad\quad\quad\quad\quad\quad 0
\end{array}
$$

Таким чином, вираз $x^7 + 1$ можна розкласти на такі множники:

$$x^7 + 1 = (x^3 + x^2 + 1)(x^4 + x^3 + x^2 + 1) =$$

$$= (x^3 + x^2 + 1)(x + 1)(x^3 + x + 1) = (x^3 + x^2 + 1)G(x).$$

Це означає, що період породжувального многочлена $G(x)$, так само, як і період породжувального многочлена Хеммінга $G_1(x)$, дорівнює 7.

Помножимо тепер інформаційний многочлен $A(x)$ на многочлен $G(x)$:

$$R(x) = A(x) \cdot G(x) = A(x)(x^4 + x^3 + x^2 + 1), \tag{42}$$

де $R(x)$ має сенс mod $(x^7 + 1)$. Отже, степінь многочлена $A(x)$ не перевищує 2, тобто

$$A(x) = a_2 x^2 + a_1 x + a_0. \tag{43}$$

Оскільки у многочлена $G(x)$ степінь дорівнює 4, а період — 7, то код, породжувальним многочленом якого є $G(x) = G_1(x) \cdot G_2(x)$, — циклічний код (7, 3) з наступними параметрами: число кодованих біт $n = 7$, число контрольних біт $m = 4$, число інформаційних біт $k = 3$.

Уточнимо порядок формування циклічного коду (7, 3).

1. Якщо інформаційна послідовність задана параметрами a_2, a_1, a_0, то інформаційний многочлен $A(x)$ має вигляд

$$A(x) = a_2 x^2 + a_1 x + a_0.$$

2. Оскільки степінь породжувального многочлена $G(x) = x^4 + x^3 + x^2 + 1$ дорівнює 4, то, помноживши многочлен $A(x)$ на x^4, отримаємо:

$$A(x) \cdot x^4 = a_2 x^6 + a_1 x^5 + a_0 x^4.$$

Поділивши отриманий добуток $A(x) \cdot x^4$ на многочлен $G(x)$, одержимо залишок $R(x)$:

$$
\begin{array}{rl}
 & a_2x^2+(a_2+a_1)x+(a_1+a_0) \qquad \leftarrow \text{Частка} \\
x^4+x^3+x^2+1) & \overline{a_2x^6+\quad a_1x^5+\quad a_0x^4} \\
 & \underline{a_2x^6+\quad a_2x^5+\quad a_2x^4+\qquad\qquad a_2x^2} \\
 & \quad (a_2+a_1)x^5+(a_2+a_0)x^4+\qquad\qquad a_2x^2 \\
 & \quad \underline{(a_2+a_1)x^5+(a_2+a_1)x^4+(a_2+a_1)x^3+\qquad\quad (a_2+a_1)x} \\
 & \qquad\qquad (a_1+a_0)x^4+(a_2+a_1)x^3+\qquad a_2x^2+(a_2+a_1)x \\
 & \qquad\qquad \underline{(a_1+a_0)x^4+(a_1+a_0)x^3+\quad (a_1+a_0)x^2+\qquad\quad (a_1+a_0)} \\
\text{Залишок} \rightarrow & \qquad\qquad\qquad (a_2+a_0)x^3+(a_2+a_1+a_0)x^2+(a_2+a_1)x+(a_1+a_0)
\end{array}
$$

Отже, залишок можна подати так:

$$R(x) = (a_2 + a_0)x^3 + (a_2 + a_1 + a_0)x^2 + (a_2 + a_1)x + (a_1 + a_0) = c_3x^3 + c_2x^2 + c_1x + c_0,$$

де

$$
\begin{aligned}
c_3 &= a_2 + a_0 \pmod 2, \\
c_2 &= a_2 + a_1 + a_0 \pmod 2, \\
c_1 &= a_2 + a_1 \pmod 2, \\
c_0 &= a_1 + a_0 \pmod 2.
\end{aligned}
$$

3. До многочлена $A(x) \cdot x^4$ додамо залишок $R(x)$, отримаємо кодований многочлен:

$$V(x) = A(x) \cdot x^4 + R(x) = a_2x^6 + a_1x^5 + a_0x^4 + c_3x^3 + c_2x^2 + c_1x + c_0.$$

Таким чином, кодований многочлен $V(x)$, що передається, може бути записаний у вигляді

$$V = (a_2, a_1, a_0, c_3, c_2, c_1, c_0).$$

Дослідимо зв'язок між циклічним кодом (7, 3) і кодом Хеммінга (7, 4).

Приймемо один із інформаційних бітів a_0 в послідовності коду Хеммінга (7, 4) (a_3, a_2, a_1, a_0, c_2, c_1, c_0) за контрольний біт при загальному контролі парності: наприклад, $a_0 \rightarrow c_3$. Тоді має місце кодована послідовність виду (a_3, a_2, a_1, c_3, c_2, c_1, c_0).

Якщо формально виконати перепозначення інформаційних бітів наступним чином: $a_3 \rightarrow a_2$, $a_2 \rightarrow a_1$, $a_1 \rightarrow a_0$, то кодована послідовність матиме вигляд (a_2, a_1, a_0, c_3, c_2, c_1, c_0), де

$$
\begin{aligned}
c_2 &= a_2 + a_1 + a_0 \pmod 2, \\
c_1 &= a_1 + a_0 + c_3 \pmod 2, \\
c_0 &= a_2 + a_1 + c_3 \pmod 2.
\end{aligned}
$$

Оскільки c_3 є контрольним бітом при загальному контролі парності, то

$$c_3 = a_2 + a_1 + a_0 + c_2 + c_1 + c_0 =$$

$$= a_2 + a_1 + a_0 + (a_2 + a_1 + a_0) + (a_1 + a_0 + c_3) + (a_2 + a_1 + c_3) = a_2 + a_0 \pmod 2.$$

Отже, решту контрольних бітів кодованої послідовності можна подати так:

$$c_2 = a_2 + a_1 + a_0 \pmod 2,$$

$$c_1 = a_1 + a_0 + c_3 = a_1 + a_0 + (a_2 + a_0) = a_2 + a_1 \pmod 2,$$

$$c_0 = a_2 + a_1 + c_3 = a_2 + a_1 + (a_2 + a_0) = a_1 + a_0 \pmod 2.$$

Отримана послідовність коду Хеммінга (7, 4) (a_2, a_1, a_0, c_3, c_2, c_1, c_0), де $c_3 = a_2 + a_0$ (mod 2), $c_2 = a_2 + a_1 + a_0$ (mod 2), $c_1 = a_2 + a_1$ (mod 2), $c_0 = a_1 + a_0$ (mod 2), аналогічна циклічному коду (7, 3) з породжувальним многочленом $G(x) = x^4 + x^3 + x^2 + 1$.

Цей код називають ***корегувальним кодом Хеммінга*** (SEC.DED); у нього міжкодова відстань дорівнює 4.

Формування корегувального коду Хеммінга (7, 3) можна виконати за допомогою схеми множення на x^4 і схеми ділення на $x^4 + x^3 + x^2 + 1$ (рис. 42).

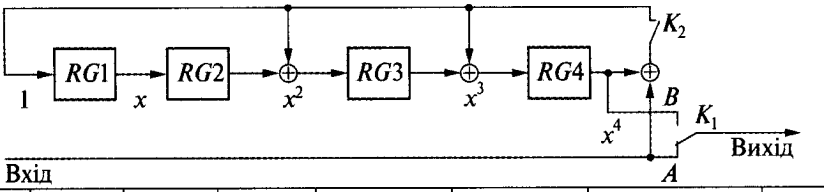
До передаваної послідовності $V = (a_2, a_1, a_0, c_3, c_2, c_1, c_0)$ в лінії передачі додається помилкова послідовність $E = (e_6, e_5, e_4, e_3, e_2, e_1, e_0)$, в якій тільки два члени можуть дорівнювати 1. Тобто кодована послідовність, що приймається, має вигляд

$$U = V + E = (a_2 + e_6,\ a_1 + e_5,\ a_0 + e_4,\ c_3 + e_3,\ c_2 + e_2,\ c_1 + e_1,\ c_0 + e_0) \pmod 2.$$

У процесі декодування достатньо проаналізувати тільки помилкову послідовність E. Породжувальний многочлен $G(x)$ коректувального коду Хеммінга (7, 3) формується унаслідок перемножування двох многочленів:

$$G(x) = (x + 1)(x^3 + x + 1).$$

Отже, код, в якому $G(x)$ використовується як породжувальний многочлен, ділиться і на $x + 1$, і на $x^3 + x + 1$.



Синхро-імпульс	Вхід	Вихід *RG*1	Вихід *RG*2	Вихід *RG*3	Вихід *RG*4	Вихід кодера	Стан перемикачів
0	$x^2 \to a_2$	0	0	0	0	$a_2 \leftarrow x^6$	
1	$x^1 \to a_1$	a_2	0	a_2	a_2	$a_1 \leftarrow x^5$	$K_1 \to A$
2	$x^0 \to a_0$	$a_2 + a_1$	a_2	$a_2 + a_1$	a_1	$a_0 \leftarrow x^4$	$K_1 \to$ Вкл
3	0	$a_1 + a_0$	$a_2 + a_1$	$a_2 + a_1 + a_0$	$a_2 + a_1$	$a_2 + a_1 \leftarrow x^3$	
4	0	0	$a_1 + a_0$	$a_2 + a_1$	$a_2 + a_1 + a_0$	$a_2 + a_1 + a_0 \leftarrow x^2$	
5	0	0	0	$a_1 + a_0$	$a_2 + a_1$	$a_2 + a_1 \leftarrow x^1$	$K_1 \to B$
6	0	0	0		$a_1 + a_0$	$a_1 + a_0 \leftarrow x^0$	$K_1 \to$ Викл

Рис. 42. Схема формування корегувального коду Хеммінга (7, 3)

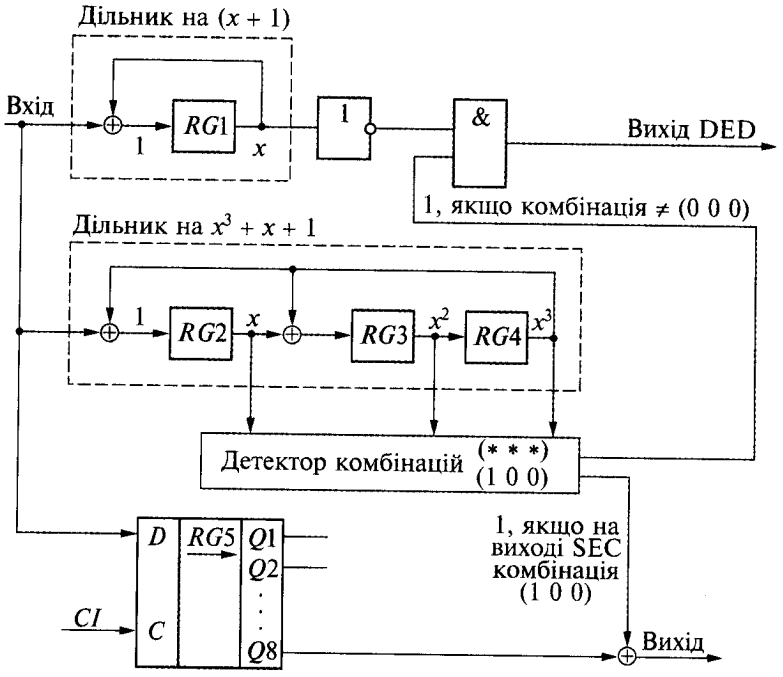


Рис. 43. Схема декодера корегувального коду Хеммінга (7, 3)

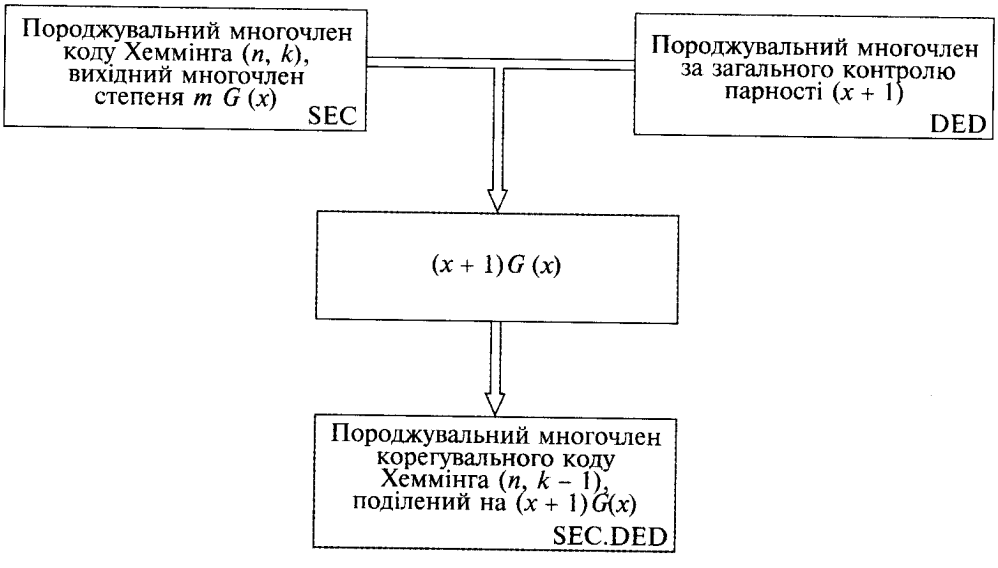


Рис. 44. Схема, за якою одержують корегувальний код Хеммінга SEC.DED із коду Хеммінга SEC

Тому як декодер можна застосувати схему декодування корегувального коду Хеммінга (7, 3), що включає декодер при загальному контролі парності (дільник на $x + 1$) і декодер коду Хеммінга (7, 4) (дільник на $x^3 + x + 1$) (рис. 43).

Помилковий многочлен $E(x) = e_6 \cdot x^6 + e_5 \cdot x^5 + e_4 \cdot x^4 + e_3 \cdot x^3 + e_2 \cdot x^2 + e_1 \cdot x + e_0$ ділять на породжувальний многочлен коду Хеммінга (7, 4) $G_1(x) = x^3 + x + 1$ і отримують залишок

$$R_1(x) = (e_6 + e_5 + e_4 + e_2)x^2 + (e_5 + e_4 + e_3 + e_1)x + (e_6 + e_5 + e_3 + e_0) \pmod 2.$$

Далі, поділивши помилковий многочлен $E(x)$ на породжувальний многочлен при загальному контролі парності $G_2(x) = x + 1$, одержали залишок

$$R_2 = e_6 + e_5 + e_4 + e_3 + e_2 + e_1 + e_0 \pmod 2$$

Якщо в помилковій послідовності $E(x)$ з'явилося два помилкових біта, кожний із яких дорівнює 1, то виконують сумісний аналіз залишків $R_1(x)$ і $R_2(x)$. При цьому можливі такі варіанти:

1. $R_1(x) = 0$, $R_2 = 0$ — помилка відсутня;
2. $R_1(x) \neq 0$, $R_2 = 1$ — виправлення одиничної помилки (SEC);
3. $R_1(x) \neq 0$, $R_2 = 0$ — виявлення подвійної помилки (DED);
4. $R_1(x) = 0$, $R_2 = 1$ — потрійна помилка (виходить за рамки можливостей корегувального коду Хеммінга (7, 3)).

Функціональна схема утворення корегувального коду Хеммінга SEC.DED із коду Хеммінга SEC наведена на рис. 44.

КОД ХЕММІНГА І КОД БОУЗА–ЧОУДХУРІ–ХОКВІНГЕМА

Французький вчений А. Хоквінгем (1959 р.) та американці Р.К. Боуз і Д.К. Рой-Чоудхурі (1960 р.) відкрили великий клас кодів, що забезпечують довільну мінімальну кодову відстань: $d_{\min} \geq 5$. Їх назвали коди Боуза—Чоудхурі—Хоквінгема (БЧХ). Ці коди належать до численного класу циклічних кодів і здатні виправляти багатократні помилки. Примітивним кодом БЧХ, що виправляє t помилок, називається код довжиною $n = q^m - 1$ над $GF(q)$, для якого елементи є коренями породжувального багаточлена $\alpha^1, \alpha^2, ..., \alpha^{2t}$ (α — примітивний елемент поля $GF(q^m)$).

Коди БЧХ — це широкий клас циклічних кодів у теорії кодування, що використовуються для захисту інформації від помилок. Вони розрізняються можливістю побудови коду із заздалегідь певними корегувальними властивостями, а саме, з мінімальною кодовою відстанню. Вартими уваги щодо кодів БЧХ є, щонайменше, чотири обставини.

1. Серед кодів БЧХ за невеликих довжин існують хороші (але не кращі з відомих) коди.

2. Відомі відносно прості методи і технічні засоби їх кодування і декодування.

3. Коди Ріда—Соломона, що є окремим випадком кодів БЧХ — широко відомим підкласом недвійкових кодів — мають певні оптимальні властивості та прогнозовану вагову структуру.

4. Коди БЧХ є базою для вивчення багатьох інших класів кодів.

Формальний опис коду БЧХ. Код БЧХ — код, який можна задати породжувальним поліномом. Для його знаходження необхідно заздалегідь визначити довжину коду n (вона не може бути довільною) і необхідну мінімальну відстань $d \leq n$. Знайти породжувальний поліном можна з використанням наступних операцій.

Нехай α — примітивний елемент поля $GF(q^m)$ (тобто $\alpha^{q^m-1} = 1$, $\alpha^i \neq 1$, $i < q^m - 1$), $\beta = \alpha^s$, $s = (q^m - 1)$, n — елемент поля $GF(q^m)$ порядку n. Тоді нормований поліном $G(x)$ мінімального степеня над полем $GF(q)$, коренями якого є $d - 1$ степенів $\beta^{l_0}, \beta^{l_0+1}, ..., \beta^{l_0+d-2}$, що передують підряд один за одним елементу β, для деякого цілого l_0 (зокрема, числа 0 і 1), є породжувальним поліномом коду БЧХ над полем $GF(q)$ з довжиною n і мінімальною відстанню $d_0 \geq d$.

Кількість перевірних символів m дорівнює степеню многочлена $G(x)$, число інформаційних символів — $k = n - m$, величина d називається *конструктивною відстанню коду* БЧХ. Якщо $n = q^m - 1$, то код називається *при-*

мітивним, інакше — непримітивним. Так само, як і для циклічного коду, кодовий поліном $c(x)$ може бути отриманий з інформаційного полінома $A(x)$, степеня не більше ніж k–1, шляхом перемножування поліномів $A(x)$ та $G(x)$:

$$c(x) = A(x) \cdot G(x).$$

Для кодування кодами БЧХ застосовуються такі самі методи, як і для кодування циклічними кодами.

Розглянемо код БЧХ SEC.DED, який використовують для цифрової передачі звукового супроводу в супутниковому зв'язку. Існують два типи цього коду: SEC.DED БЧХ (7, 3) та SEC.DED БЧХ (63, 56).

Многочлен, що породжує код SEC.DED БЧХ (7, 3), має вигляд $G(x) = x^4 + x^3 + x^2 + 1$. Цей многочлен не відрізняється від корегувального многочлена (41) коду Хеммінга (7, 3). Оскільки породжувальні многочлени однакові, то і властивості кодів теж однакові, як і схеми їх кодування та декодування.

Природно припустити, що і код SEC.DED БЧХ (63, 56) аналогічний коректувальному коду Хеммінга (63, 56). Перевіримо це припущення.

Многочлен, що породжує код БЧХ (63, 56), має вигляд

$$G(x) = x^7 + x^6 + x^2 + 1. \qquad (44)$$

Його можна розкласти на множники:

$$G(x) = (x+1)(x^6 + x + 1) = G_1(x) \cdot G_2(x). \qquad (45)$$

Співмножник $G_2(x) = x^6 + x + 1$ є незвідним многочленом, для якого математична безліч залишків становить $2^6 = 64$. Таким чином, код (45), породжувальним многочленом якого є співмножник $G_2(x)$, відповідає кінцевому полю, що містить у собі 64 елементи. Виключивши елемент "0", отримаємо 63 елементи, які можна відобразити елементом x із степенем, що збільшується. Період такої послідовності $2^6 - 1 = 63$. Отже, $G_2(x)$ є початковим многочленом, а код, породжувальним многочленом якого є $G_2(x)$, — циклічний код SEC. Оскільки степінь породжувального многочлена становить 6, то такий код відповідає коду Хеммінга (63, 57) SEC з параметрами:

- число контрольних бітів $m = 6$;
- число кодованих бітів $n = 2^m - 1 = 63$;
- число інформаційних бітів $k = n - m = 57$.

Кодувальна і декодувальна схеми цього коду наведені на рис. 45 та 46.

Породжувальний многочлен $G_1(x) = x + 1$ використовується для загального контролю парності. Тому код, породжувальним многочленом якого є $G(x) = G_1(x) \cdot G_2(x)$, — код Хеммінга (63, 56) SEC.DED. Для нього схеми кодування і декодування продемонстровані на рис. 47 та 48.

У принципі код БЧХ можна називати кодом Хеммінга. Теоретично такий код дозволяє виправляти одночасно t помилок.

Нагадаємо, якщо прийняти, що степінь незвідного многочлена дорівнює n, то елемент x має період $2^n - 1$. Це записується так:

$$x^{2^n-1} = x^0 = 1. \qquad (46)$$

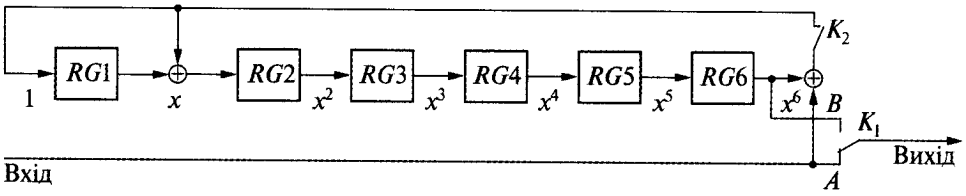


Рис. 45. Схема формування коду Хеммінга (63, 57) — коду БЧХ (63, 57) (множення на x^6 і ділення на $x^6 + x + 1$)

Тоді

$$x^{2^n-1} - 1 = 0. \tag{47}$$

Зазначимо, що формула (47) аналогічна формулі $x^{2^n-1} + 1 = 0$.

Помноживши обидві частини рівняння (47) на x, отримаємо

$$x^{2^n} - x = 0. \tag{48}$$

Якщо співвідношення (48) розглядати як рівняння коефіцієнтів $GF(2)$, то його корені розраховують так:

$$x^{2^n} - x = x\left(x^{2^n-1} - 1\right) = 0.$$

Тоді

$$x_1 = 0, \; x_{2^n-1} \text{ корені рівняння } x^{2^n-1} - 1 = 0.$$

Загальна кількість коренів рівняння (48) становить 2^n. Тобто скінченне поле $GF(2^n)$ з числом елементів 2^n є множиною, що складається з усіх коренів рівняння коефіцієнтів $GF(2)$.

Нехай, наприклад, $n = 3$. Тоді формулу (48) подамо у вигляді

$$x^8 - x = 0, \tag{49}$$

і множина усіх коренів рівняння (49) — скінченне поле $GF(2^3) = GF(8)$. Воно містить у собі корінь $x_1 = 0$ та ще сім коренів рівняння $x^7 - 1 = 0$: α_1, α_2, α_3, α_4, α_5, α_6, α_7. Запишемо множину з усіх восьми коренів рівності (49):

$$F_3 = \{0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}. \tag{50}$$

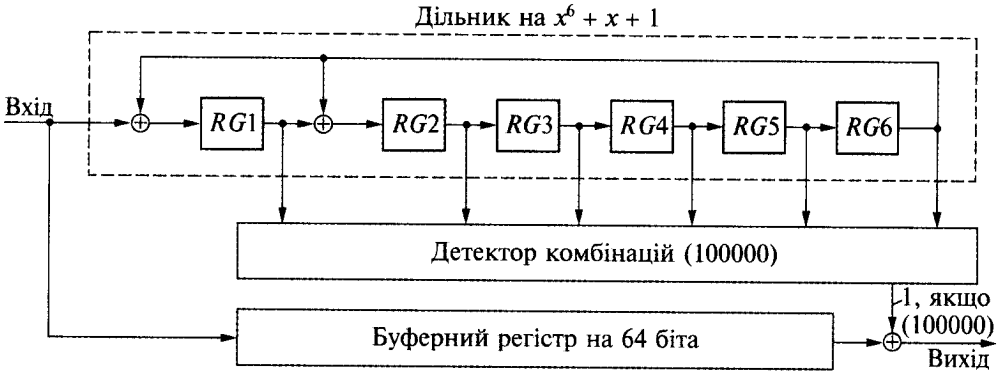


Рис. 46. Схема декодера коду Хеммінга (63, 57) — коду БЧХ (63, 57)

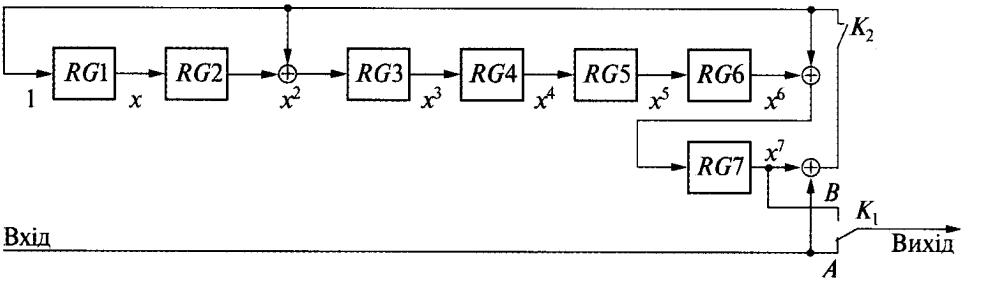


Рис. 47. Схема формування корегувального коду Хеммінга (63, 56) — коду БЧХ (63, 56) (множення на x^7 та ділення на $x^7 + x^6 + x^2 + 1$)

Оскільки множина F_3 є скінченним полем, то, видаливши з нього елемент "0", отримаємо множину

$$F_3^* = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}. \quad (51)$$

Серед елементів множини F_3^* обов'язково є елемент, що породжує решту інших елементів. Такий елемент називається ***початковим (вихідним) елементом***. Якщо початковий елемент множини F_3^* позначити α, то всі елементи α_1, α_2, α_3, α_4, α_5, α_6, α_7 можна подати як елемент α з певним степенем. Далі, оскільки початковий елемент α теж є коренем рівняння $x^7 - 1$, то $\alpha^7 = 1$. Як наслідок, множину F_3^* можна переписати у вигляді

$$F_3^* = \{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7 = 1\}. \quad (52)$$

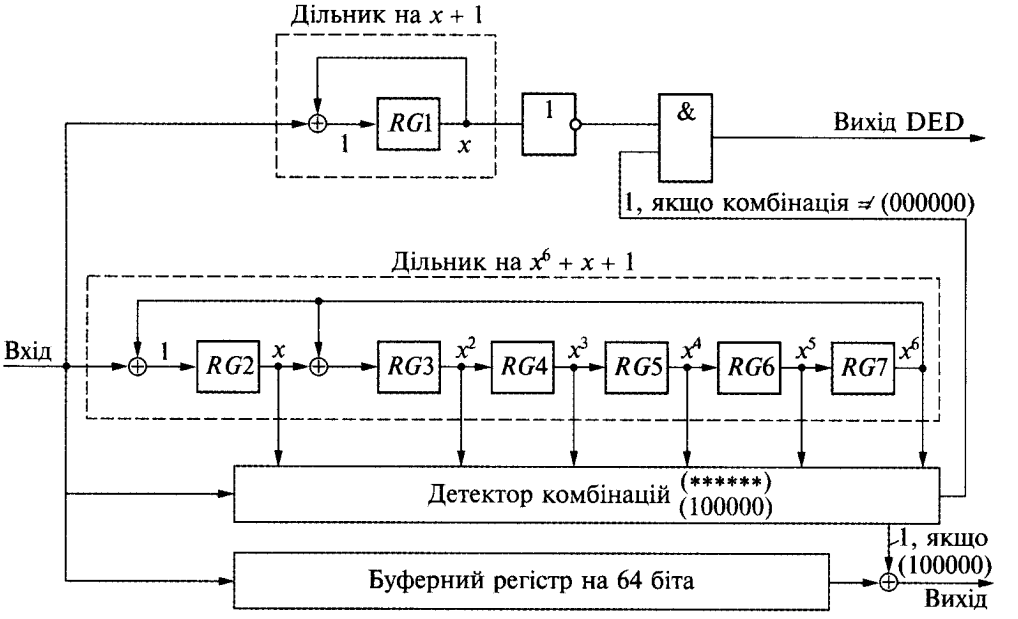


Рис. 48. Схема декодера корегувального коду Хеммінга (63, 56) — коду БЧХ (63, 56)

Серед незвідних множників рівняння $x^7 - 1$, є незвідний многочлен, коренем якого є початковий елемент α. Цей многочлен називається ***початковим многочленом***. Тобто, ***початковий многочлен — це незвідний многочлен, коренем якого є початковий елемент*** α.

Многочлен $x^7 - 1$ можна розкласти на такі незвідні множники:

$$x^7 - 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1). \tag{53}$$

Приймемо за початковий многочлен співмножник $x^3 + x + 1$ і будемо вважати, що α є його коренем. Тоді можна записати

$$\alpha^3 + \alpha + 1 = 0. \tag{54}$$

У формулі (54) аргумент x формально замінено на корінь α. Тобто елемент x множини із залишків незвідного многочлена $x^3 + x + 1$ є насправді не що інше, як корінь α (початковий елемент).

Тепер візьмемо елементи $\alpha, \alpha^2, \ldots, \alpha^7$ множини $GF(8)$ і опишемо їх за допомогою многочлена, початковим елементом якого є α. Зведемо рівняння (54) до вигляду

$$\alpha^3 = \alpha + 1. \tag{55}$$

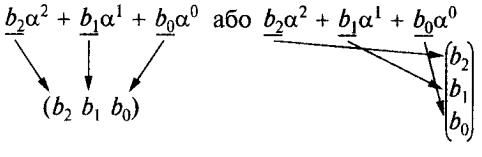
Ґрунтуючись на співвідношенні (55), збільшення степенів α можна записати так:

$$\begin{cases} \alpha^0 = 1, \\ \alpha^1 = \alpha, \\ \alpha^2 = \alpha^2, \\ \alpha^3 = \alpha + 1, \\ \alpha^4 = \alpha^2 + \alpha, \\ \alpha^5 = \alpha^2 + \alpha + 1, \\ \alpha^6 = \alpha^2 + 1, \\ \alpha^7 = \alpha^0 = 1. \end{cases} \tag{56}$$

Графічно залежність, що описується системою рівнянь (56), можна подати у вигляді періодичності елементів α множини F_3^* (рис. 49).

Зіставляючи рис. 49 та рис. 29 можна побачити їх формальну подібність. Система рівнянь (56) описує взаємно однозначну залежність між коренями α многочлена F_3^* і елементами множини $GF(2^3)$ (за виключенням елемента 0, табл. 8).

Векторне подання елементів α (див. табл. 8) — це формальний запис коефіцієнтів многочлена з коренем α, які розглядаються як елементи поля $GF(2^3)$ і зібрані разом. Тобто маємо наступні відповідності:



Наведені три типи подання розрізняються формою, але по суті аналогічні.

Оскільки елементи α, α^2, α^3, α^4, α^5, α^6, α^7 множини F_3^* є також і коренями многочлена $x^7 - 1$, то можна записати таке рівняння:

$$x^7 - 1 = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5)(x-\alpha^6)(x-\alpha^7). \qquad (57)$$

Нагадаємо, що α є коренем початкового многочлена $x^3 + x + 1$. Це многочлен 3-го степеня, що має ще два корені. Оскільки $x^3 + x + 1$ є множником многочлена $x^7 - 1$, то серед коренів α, α^2, α^3, α^4, α^5, α^6, α^7 є і три його корені. Тому

$$G(\alpha) = \alpha^3 + \alpha + 1. \qquad (58)$$

Розглянемо тепер елемент α^2. Як бачимо,

$$G(\alpha^2) = \alpha^6 + \alpha^2 + 1 = (\alpha^3 + \alpha + 1)^2 = 0.$$

Таким чином, α^2 також є коренем многочлена. Аналогічно можна досліджувати інші п'ять членів і знайти корені. Вчинимо простіше. Як було з'ясовано, якщо α — корінь початкового многочлена, то α^2 також корінь (початковий елемент) многочлена $G(x)$. Тоді, якщо α^2 — корінь, то α^4 також корінь $G(x)$. Перевіримо це припущення:

$$G(\alpha^4) = \alpha^{12} + \alpha^4 + 1 = (\alpha^6 + \alpha^2 + 1)^2 = 0.$$

З'ясовується, що коренями початкового многочлена $x^3 + x + 1$ є елементи α, α^2, α^4. Остаточно переконаємося в цьому:

$$\begin{aligned}(x-\alpha)(x-\alpha^2)(x-\alpha^4) &= [x^2 - (\alpha+\alpha^2)x + \alpha^3](x-\alpha^4) = \\ &= x^3 - (\alpha^4 + \alpha^2 + \alpha)x^2 + (\alpha^6 + \alpha^5 + \alpha^3)x - \alpha^7 = \\ &= x^3 + (\alpha^4 + \alpha^2 + \alpha)x^2 + (\alpha^6 + \alpha^5 + \alpha^3)x + \alpha^7.\end{aligned}$$

Упорядкуємо коефіцієнти α, використавши систему (56):

$$\alpha^4 + \alpha^2 + \alpha = (\alpha^2 + \alpha) + \alpha^2 + \alpha = 2\alpha^2 + 2\alpha = 0,$$

$$\begin{aligned}&\alpha^6 + \alpha^5 + \alpha^3 = \\ &= (\alpha^2 + 1) + (\alpha^2 + \alpha + 1) + (\alpha + 1) = \\ &= 2\alpha^2 + 2\alpha + 3 = 1,\end{aligned}$$

$$\alpha^7 = 1.$$

Як наслідок отримаємо многочлен

$$(x-\alpha)(x-\alpha^2)(x-\alpha^4) = x^3 + x + 1.$$

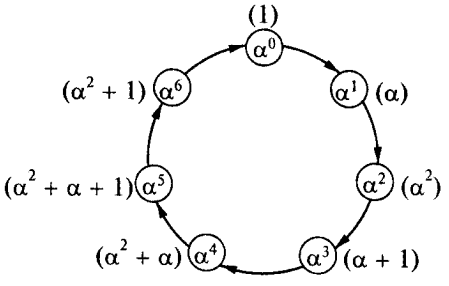


Рис. 49. Приклад періодичності елемента α множини F_3^*

6*

Таблиця 8. Різні типи подання періодичності елемента α

Елемент $GF(2^3)$	Подання у вигляді многочлена	Векторне подання	Елемент $GF(2^3)$	Подання у вигляді многочлена	Векторне подання
0	—	(0 0 0) або $\begin{pmatrix}0\\0\\0\end{pmatrix}$	α^4	$\alpha^2+\alpha$	(1 1 0) або $\begin{pmatrix}1\\1\\0\end{pmatrix}$
$\alpha^0=1$	1	(0 0 1) або $\begin{pmatrix}0\\0\\1\end{pmatrix}$	α^5	$\alpha^2+\alpha+1$	(1 1 1) або $\begin{pmatrix}1\\1\\1\end{pmatrix}$
α	α	(0 1 0) або $\begin{pmatrix}0\\1\\0\end{pmatrix}$	α^6	α^2+1	(1 0 1) або $\begin{pmatrix}1\\0\\1\end{pmatrix}$
α^2	α^2	(1 0 0) або $\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\alpha^7=\alpha^0$	1	(0 0 1) або $\begin{pmatrix}0\\0\\1\end{pmatrix}$
α^3	$\alpha+1$	(0 1 1) або $\begin{pmatrix}0\\1\\1\end{pmatrix}$			

Отже, якщо розглядати поле $GF(2^3)$, то елементи α, α^2, α^4 утворюють многочлен.

Відомо, що многочлен з мінімальним показником степеня для заданих елементів, що є його коренями, називають ***мінімальним многочленом***. Такий мінімальний многочлен містить у собі всі елементи α. Покажемо це, скориставшись наведеним вище порядком дій. А саме, якщо α — корінь мінімального многочлена, то α^2 — також його корінь; якщо α^2 — корінь мінімального многочлена, то α^4 — також його корінь; якщо α^4 — корінь мінімального многочлена, то α^8 — також його корінь і т. д. Проте корені α, α^2, α^4, α^8,... є елементами скінченного поля і послідовність цих коренів — теж скінченна. У цьому разі оскільки α, α^2, α^4, $\alpha^8=\alpha\cdot\alpha^7=\alpha$, $\alpha^{16}=(\alpha^8)^2=\alpha^2$,..., то відбувається періодичне повторення коренів α, α^2, α^4. Тому маємо, що многочлен x^3+x+1 — мінімальний.

Тепер знайдемо мінімальний многочлен для елемента α^3. Спочатку отримаємо послідовність коренів α^3, α^6, $\alpha^{12}=\alpha^7\cdot\alpha^5=\alpha^5$, $\alpha^{24}=\alpha^{21}\cdot\alpha^3=\alpha^3$,...

Послідовність α^3, α^6, α^5 теж є коренями мінімального многочлена

$$(x-\alpha^3)(x-\alpha^6)(x-\alpha^5)=[x^2-(\alpha^3+\alpha^6)x+\alpha^9](x-\alpha^5)=$$

$$=x^3-(\alpha^3+\alpha^6+\alpha^5)x^2+(\alpha^9+\alpha^8+\alpha^{11})x-\alpha^{14}=$$

$$=x^3+(\alpha^3+\alpha^6+\alpha^5)x^2+(\alpha^9+\alpha^8+\alpha^{11})x+\alpha^{14}.$$

Упорядкуємо коефіцієнти α біля змінної x, використавши систему (56):

$$\alpha^3 + \alpha^6 + \alpha^5 = (\alpha + 1) + (\alpha^2 + 1) + (\alpha^2 + \alpha + 1) = 2\alpha^2 + 2\alpha + 3 = 1,$$

$$\alpha^9 + \alpha^8 + \alpha^{11} = \alpha^2 + \alpha + \alpha^4 = 2\alpha^2 + 2\alpha = 0,$$

$$\alpha^{14} = \alpha^7 \cdot \alpha^7 = 1.$$

Отже,

$$(x - \alpha^3)(x - \alpha^6)(x - \alpha^5) = x^3 + x^2 + 1.$$

Тобто мінімальний многочлен з елементами α^3, α^5, α^6 має вигляд $x^3 + x^2 + 1$. Залишається ще один елемент α^7. Послідовність для α^7 може бути подана так:

$$\alpha^7 = 1, \quad \alpha^{14} = (\alpha^7)^2 = 1, \quad \alpha^{28} = (\alpha^7)^4 = 1, \ldots$$

Тоді запишемо мінімальний многочлен з елементом $\alpha^7 = x - \alpha^7 = x + 1$.

У табл. 9 наведено мінімальні многочлени для елементів α поля $GF(2^3)$, за виключенням нульового елемента.

Аналіз отриманих результатів показує, що код БЧХ — це код, який утворюється за допомогою породжувального многочлена, коренями якого є елементи зі зростаючими степенями: α, α^2, ..., α^{2t-1}, α^{2t} (α — корінь початкового многочлена).

При формуванні корегувального коду Хеммінга SEC.DED було використано многочлен коду Хеммінга і породжувальний многочлен загального контролю парності.

З'ясуємо, що відбуватиметься, якщо до коду БЧХ додати многочлен $x + 1$ загального контролю парності. Коренем незвідного многочлена $x + 1$ є 1. Запишемо його так: $1 = \alpha^0$. Це свідчить, що навіть у разі збільшення α^0 при формуванні породжувального многочлена, безперервна послідовність коренів α зі зростаючими степенями породжувального многочлена не порушується. У зв'язку з тим, що код БЧХ визначено як код з безперервною послідовністю коренів α зі зростаючими степенями породжувального многочлена, для збереження вказаної безперервності, як і раніше, використовується код БЧХ.

У коді БЧХ, що отримано за допомогою породжувального многочлена, коренями якого є α зі зростаючими степенями, з доданим коренем α^0 (α^0, α^1, α^2, ..., α^{2t}), міжкодова відстань збільшується на 1. Такий код БЧХ дає можливість виправляти t помилок і виявляти $t + 1$ помилку.

Розглянемо процес формування конкретного коду БЧХ. Нехай початковий многочлен має вигляд $x^3 + x + 1$, коренем якого є початковий елемент α.

Таблиця 9. Вигляд мінімальних многочленів для елементів поля $GF(2^3)$

Елемент	Мінімальний многочлен	Елемент	Мінімальний многочлен
$\alpha^0 = 1$	$x + 1$	α^4	$x^3 + x + 1$
α	$x^3 + x + 1$	α^5	$x^3 + x^2 + 1$
α^2	$x^3 + x + 1$	α^6	$x^3 + x^2 + 1$
α^3	$x^3 + x^2 + 1$	($\alpha^7 = 1$)	($x + 1$)

Кількість бітів коду становить $2^3 - 1 = 7$. Розташуємо корені α у порядку зростання їх степенів, і дослідимо породжувальний многочлен, коренем якого є тільки початковий елемент α. Мінімальний за α многочлен відповідатиме початковому многочлену $x^3 + x + 1$. Як було показано, у мінімальному многочлені $x^3 + x + 1$ є ще два корені, а саме: α^2 і α^4. Неважко з'ясувати, що породжувальний многочлен, коренем якого є початковий елемент α, аналогічний породжувальному многочлену, коренями якого є елементи α і α^2. Це свідчить, що як многочлен породжувального коду БЧХ краще використовувати ***мінімальний загальний многочлен*** (LCM) мінімального многочлена із безперервно зростаючим степенем α.

Таким чином, породжувальний многочлен дорівнює LCM (мінімальний многочлен за α, мінімальний многочлен за α^2,..., мінімальний многочлен за α^{2t-1}, мінімальний многочлен за α^{2t}). З вигляду породжувального многочлена $G(x) = x^3 + x + 1$ випливає, що кількість контрольних бітів дорівнює 3.

Якщо за початковий прийняти многочлен, коренем якого є α, і на підставі породжувального многочлена, коренями якого є елементи α, α^2 (α з безперервно зростаючим степенем), сформувати код БЧХ, то це буде код БЧХ (7, 4) SEC з такими параметрами:

- число кодованих бітів $n = 7$;
- число контрольних бітів $m = 3$;
- число інформаційних бітів $k = 4$;
- породжувальний многочлен — $x^3 + x + 1$;
- здатність виправлення помилок SEC.

Цей код аналогічний коду Хеммінга (7, 4) SEC. На рис. 50 наведено процес утворення коду БЧХ і взаємозв'язок параметрів, що його створюють.

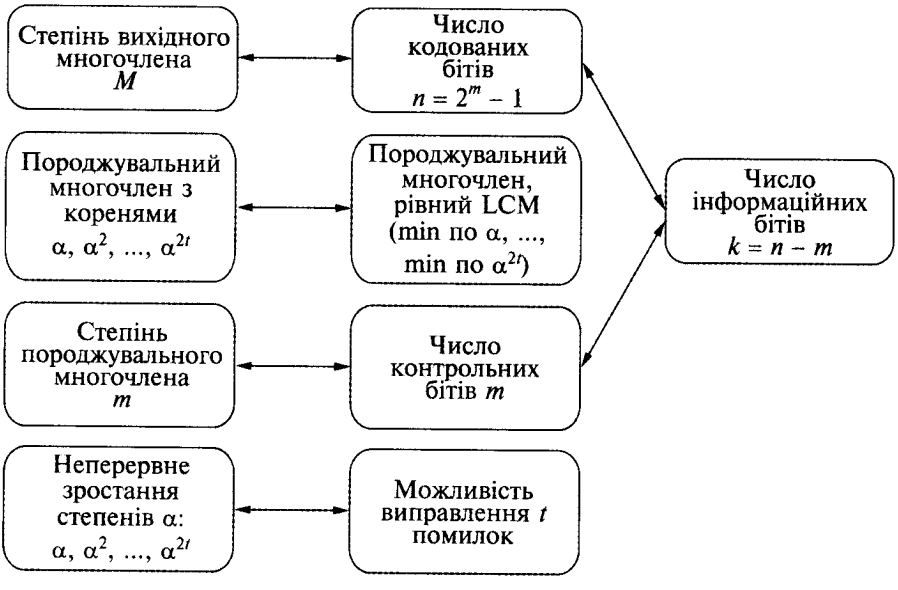


Рис. 50. Взаємозв'язок між кодами різних типів

Сформуємо тепер код БЧХ на основі многочлена, коренями якого є елементи α, α^2, α^3 при безперервному зростанні степеня α. З використанням результатів, наведених у табл. 9, отримаємо:

- мінімальний многочлен за $\alpha \Rightarrow x^3 + x + 1$;
- мінімальний многочлен за $\alpha^2 \Rightarrow x^3 + x + 1$;
- мінімальний многочлен за $\alpha^3 \Rightarrow x^3 + x^2 + 1$.

Тоді породжувальний многочлен можна подати так:

$$G(x) = LCM\left(x^3 + x + 1,\ x^3 + x + 1,\ x^3 + x^2 + 1\right) = \left(x^3 + x + 1\right)\left(x^3 + x^2 + 1\right) =$$

$$= x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$$

Корені мінімального многочлена $x^3 + x + 1$ — α, α^2, α^4, корені мінімального многочлена $x^3 + x^2 + 1$ — α^3, α^5, α^6. Тоді корені можна записати в порядку зростання їх степенів: α, α^2, α^3, α^4, α^5, α^6.

Оскільки початковим многочленом є $x^3 + x + 1$, то даний код відповідає коду БЧХ (7, 1) ТЕС з такими параметрами:

- число кодованих бітів $n = 7$;
- число контрольних бітів $m = 6$;
- число інформаційних бітів $k = 1$;
- породжувальний многочлен — $x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$;
- здатність виправлення помилок ТЕС.

Збільшення кількості контрольних бітів, що передаються разом з інформаційними бітами, для виправлення помилок, що виникають, є необхідною, але не недостатньою умовою у разі передачі великих масивів інформації. На практиці ефективна передача інформації — це велика проблема.

Розглянемо код БЧХ, при формуванні якого за початковий многочлен взято многочлен $x^3 + x + 1$, а коренями є α^0, α^1, α^2 при безперервному зростанні степенів. Із використанням результатів, наведених у табл. 9, отримаємо:

- мінімальний многочлен за $\alpha^0 \Rightarrow x + 1$;
- мінімальний многочлен за $\alpha^1 \Rightarrow x^3 + x + 1$;
- мінімальний многочлен за $\alpha^2 \Rightarrow x^3 + x + 1$.

Звідси, породжувальний многочлен коду БЧХ можна записати у вигляді мінімального загального многочлена:

$$G(x) = LCM\left(x + 1,\ x^3 + x + 1,\ x^3 + x + 1\right) = \left(x + 1\right)\left(x^3 + x + 1\right) = x^4 + x^3 + x^2 + 1.$$

Це відповідає коду БЧХ (7, 3) SEC.DED з такими параметрами:

- число кодованих бітів $n = 7$;
- число контрольних бітів $m = 4$;
- число інформаційних бітів $k = 3$;
- породжувальний многочлен — $x^4 + x^3 + x^2 + 1$;
- здатність виправлення помилок SEC.DED.

Тепер неважко визначити, що код, при утворенні якого як початковий многочлен використовували вираз $x^6 + x + 1$, а корені — α^0, α^1, α^2 (при безперервному зростанні степенів α), — код БЧХ (63, 56) SEC.DED, аналогічний корегувальному коду Хеммінга (63, 56).

Таким чином, зв'язок між кодом БЧХ і кодом Хеммінга цілком зрозумілий, тому декодування коду БЧХ здійснюється аналогічно декодуванню коду Хеммінга, хоч і складніше унаслідок необхідності розв'язку рівнянь порядку t і визначення їх коренів.

На завершення коротко опишемо процедуру формування породжувального многочлена БЧХ. Нехай параметр q дорівнює 2, необхідна довжина коду — $n = 2^4 - 1 = 15$ і мінімальна міжкодова відстань — $d_0 \geq d = 5$. Приймемо, що α — примітивний елемент поля $GF(16)$, і α, α^2, α^3, α^4 — чотири степені примітивного елемента α, що розташовані у порядку зростання. Вони належать двом класам функцій над полем $GF(2)$, яким відповідають незвідні поліноми $f_1(x) = x^4 + x + 1$ та $f_2(x) = x^4 + x^3 + x^2 + x + 1$. Тоді коренями поліному

$$G(x) = f_1(x)f_2(x) = x^8 + x^7 + x^6 + x^4 + 1$$

є елементи α, α^2, α^3, α^4, і він є породжувальним поліномом БЧХ-коду.

ЛІНІЙНІ КОДИ. КОНТРОЛЬ ТА ВИПРАВЛЕННЯ ПОМИЛОК

Практично всі коди, що використовуються, — лінійні. Це пов'язано з тим, що нелінійні коди значно складніше досліджувати, і для них важко забезпечити прийнятну легкість кодування і декодування.

Лінійний блоковий код — це такий код, безліч кодових слів якого утворює k-вимірний лінійний підпростір (C-простір) в n-вимірному лінійному просторі, ізоморфному простору k-бітових векторів. Ізоморфізм характеризує наявність подібності у різних об'єктів.

Ефективність кодів визначається кількістю помилок, які той може виправити, кількістю надмірної інформації, додавання якої потрібне, а також складністю реалізації кодування і декодування (як апаратно, так і у вигляді програми для комп'ютера).

На відміну від будь-яких кодів (зокрема, нелінійних) методи декодування лінійних кодів можна істотно спростити. У цьому разі для кожного прийнятого вектора обчислюється синдром (див. нижче). Потім за ним визначається вектор помилки, за допомогою якого визначається передане кодове слово.

Важливою особливістю коду Хеммінга є можливість створення на його основі лінійного коду. Зв'язок лінійного коду з кодами інших типів наведено на рис. 51.

Повернемося до коду Хеммінга (7, 4) a_3, a_2, a_1, a_0, c_2, c_1, c_0. Для побудови лінійного коду необхідно встановити відповідність між станами помилок і станами контролю. Для циклічного коду Хеммінга (7, 4) така відповідність задається параметрами табл. 7, де показані помилкові стани при зростанні степенів елемента x. Якщо замість x підставити α, то на підставі даних табл. 7 отримаємо табл. 10, в якій відображена відповідність між помилковими бітами і елементами поля множини $GF(2^3)$.

Якщо на підставі табл. 10 вивести рівняння контролю парності, то стане можливим створення лінійного коду. Це рівняння має такий вигляд

$$S = \alpha^6 \cdot a_3 + \alpha^5 \cdot a_2 + \alpha^4 \cdot a_1 + \alpha^3 \cdot a_0 + \alpha^2 \cdot c_2 + \alpha^1 \cdot c_1 + \alpha^0 \cdot c_0. \qquad (59)$$

Таблиця 10. Відповідність між помилковими бітами і елементами поля $GF(2^3)$

Помилковий біт	Стан помилки (елемент $GF(2^3)$)	Помилковий біт	Стан помилки (елемент $GF(2^3)$)
Немає помилки	0	a_0	α^3
a_3	α^6	c_2	α^2
a_2	α^5	c_1	α^1
a_1	α^4	c_0	$\alpha^0 = 1$

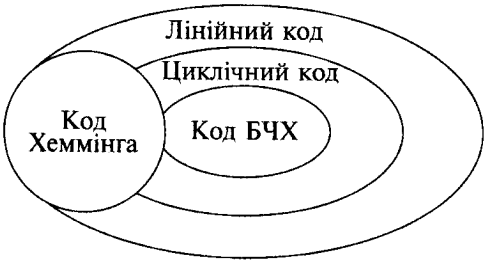


Рис. 51. Схема взаємозв'язків кодів, що контролюють помилки

Нагадаємо, що можливі різні форми подання елементів поля $GF(2^3)$ (див. табл. 8). Скористаємося векторним поданням. Для цього за допомогою параметрів табл. 8 складемо табл. 11.

Згідно з даними табл. 11 вектори рядків і стовпців пов'язані між собою такими матрицями:

$$(0\ 0\ 0)^T = \begin{pmatrix}0\\0\\0\end{pmatrix},$$

$$\begin{pmatrix}0\\0\\0\end{pmatrix}^T = (0\ 0\ 0),\ (1\ 0\ 1)^T = \begin{pmatrix}1\\0\\1\end{pmatrix},\ \begin{pmatrix}1\\0\\1\end{pmatrix}^T = (1\ 0\ 1),\ \ldots$$

Отже, запис за допомогою векторів рядків і стовпців аналогічний поданню за зростаючими степенями початкового елемента α многочлена. Взагалі при виборі способу подання зазвичай керуються практичними міркуваннями, які спрощують застосування принципів завадостійкості з використанням контролю парності.

Записуючи лінійний код (59) з використанням векторів, отримуємо

$$S = \begin{pmatrix}1\\0\\1\end{pmatrix}\cdot a_3 + \begin{pmatrix}1\\1\\1\end{pmatrix}\cdot a_2 + \begin{pmatrix}1\\1\\0\end{pmatrix}\cdot a_1 + \begin{pmatrix}0\\1\\1\end{pmatrix}\cdot a_0 + \begin{pmatrix}1\\0\\0\end{pmatrix}\cdot c_2 + \begin{pmatrix}0\\1\\0\end{pmatrix}\cdot c_1 + \begin{pmatrix}0\\0\\1\end{pmatrix}\cdot c_0\ . \qquad (60)$$

Т а б л и ц я 11. Векторне подання елементів поля $GF(2^3)$

Елемент $GF(2^3)$	Векторне подання		Елемент $GF(2^3)$	Векторне подання	
	Рядок	Стовпець		Рядок	Стовпець
0	(0 0 0)	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	α^3	(0 1 1)	$\begin{pmatrix}0\\1\\1\end{pmatrix}$
α^6	(1 0 1)	$\begin{pmatrix}1\\0\\1\end{pmatrix}$	α^2	(1 0 0)	$\begin{pmatrix}1\\0\\0\end{pmatrix}$
α^5	(1 1 1)	$\begin{pmatrix}1\\1\\1\end{pmatrix}$	α	(0 1 0)	$\begin{pmatrix}0\\1\\0\end{pmatrix}$
α^4	(1 1 0)	$\begin{pmatrix}1\\1\\0\end{pmatrix}$	$\alpha^0 = 1$	(0 0 1)	$\begin{pmatrix}0\\0\\1\end{pmatrix}$

Якщо виконати підстановку

$$S = \begin{pmatrix} S_2 \\ S_1 \\ S_0 \end{pmatrix}, \tag{61}$$

то лінійний код (60) можна подати у вигляді системи рівнянь:

$$\begin{cases} S_2 = 1\cdot a_3 + 1\cdot a_2 + 1\cdot a_1 + 0\cdot a_0 + 1\cdot c_2 + 0\cdot c_1 + 0\cdot c_0 \pmod 2, \\ S_1 = 0\cdot a_3 + 1\cdot a_2 + 1\cdot a_1 + 1\cdot a_0 + 0\cdot c_2 + 1\cdot c_1 + 0\cdot c_0 \pmod 2, \\ S_0 = 1\cdot a_3 + 1\cdot a_2 + 0\cdot a_1 + 1\cdot a_0 + 0\cdot c_2 + 0\cdot c_1 + 1\cdot c_0 \pmod 2 . \end{cases} \tag{62}$$

На передавальній стороні помилки не виникають. Отже, $S = 0$ ($S_2 = 0$, $S_1 = 0$, $S_0 = 0$). Тоді для приймальної сторони система рівнянь (62) запишеться так:

$$\begin{cases} 0 = a_3 + a_2 + a_1 + c_2 \pmod 2, \\ 0 = a_2 + a_1 + a_0 + c_1 \pmod 2, \\ 0 = a_3 + a_2 + a_0 + c_0 \pmod 2, \end{cases} \tag{63}$$

звідки

$$\begin{cases} c_2 = a_3 + a_2 + a_1 \pmod 2, \\ c_1 = a_2 + a_1 + a_0 \pmod 2, \\ c_0 = a_3 + a_2 + a_0 \pmod 2. \end{cases} \tag{64}$$

Ця система рівнянь аналогічна співвідношенням для контрольних бітів (38), що встановлені для циклічного коду Хеммінга (7, 4).

На приймальній стороні для формування контрольних бітів доцільно використовувати систему рівнянь (64). Передавану кодовану послідовність (слово)

$$v = (a_3, a_2, a_1, a_0, c_2, c_1, c_0) \tag{65}$$

можна подати системою рівнянь:

$$\begin{cases} a_3 = a_3 \\ a_2 = \quad a_2 \\ a_1 = \qquad a_1 \\ a_0 = \qquad\quad a_0 \pmod 2 . \\ c_2 = a_3 + a_2 + a_1 \\ c_1 = \quad a_2 + a_1 + a_0 \\ c_0 = a_3 + a_2 + \quad a_0. \end{cases} \tag{66}$$

Якщо співвідношення (66) записати у вигляді матриці, то отримаємо породжувальну матрицю кодованої послідовності:

$$\underset{\text{Кодоване слово}}{(a_3, a_2, a_1, a_0, c_2, c_1, c_0)} = \underset{\text{Інформаційне слово}}{(a_3, a_2, a_1, a_0)} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \qquad (67)$$

Оскільки $a = (a_3, a_2, a_1, a_0)$, то, враховуючи співвідношення (64) та (67), отримуємо

$$v = a \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = aG\ . \qquad (68)$$

Співмножник G у виразі (68) називається породжувальною матрицею (рис. 52).

Таким чином, кодована послідовність, що передається, на приймальній стороні відновлюється шляхом множення інформаційної послідовності на породжувальну матрицю.

Використавши лінійний код (59), опишемо члени α за зростанням їх степенів за допомогою векторів-стовпців. За векторним поданням лінійного коду у вигляді системи рівнянь (62) для контролю парності визначимо положення помилкових бітів

$$S = \begin{pmatrix} S_2 \\ S_1 \\ S_0 \end{pmatrix},$$

яке і є ***ознакою несправності.***

Отже, у матричній формі вираз (59) можна записати так:

$$S = \alpha^6 \cdot a_3 + \alpha^5 \cdot a_2 + \alpha^4 \cdot a_1 +$$

$$+ \alpha^3 \cdot a_0 + \alpha^2 \cdot c_2 + \alpha^1 \cdot c_1 + \alpha^0 \cdot c_0 =$$

$$= \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right) \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} =$$

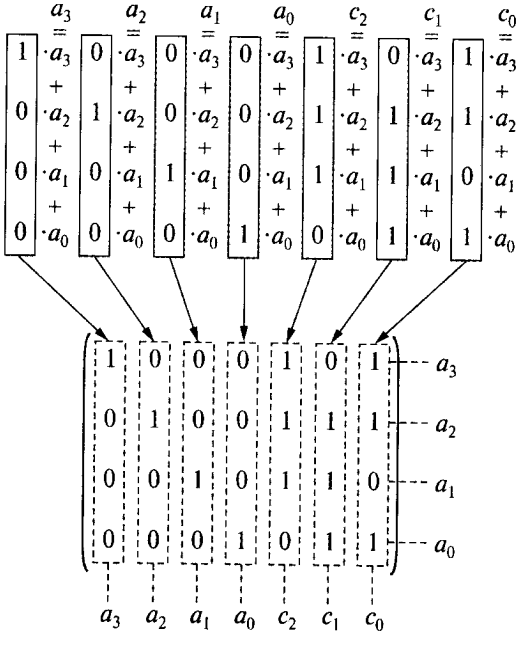


Рис. 52. Приклад породжувальної матриці коду Хеммінга (7, 4)

$$= \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right)\left(a_3, a_2, a_1, a_0, c_2, c_1, c_0\right)^T.$$

Введемо позначення:

$$H = \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right). \tag{69}$$

Тоді ознака несправності може бути подана так:

$$S = Hv^T. \tag{70}$$

Якщо тепер початкові елементи α із зростаючими їх степенями описати за допомогою векторів-стовпців, то отримаємо матрицю

$$H = \begin{bmatrix} 1\,1\,1\,0\,1\,0\,0 \\ 0\,1\,1\,1\,0\,1\,0 \\ 1\,1\,0\,1\,0\,0\,1 \end{bmatrix}. \tag{71}$$

Матриця H — ***матриця контролю парності***.

Виправлення помилок на приймальній стороні здійснюється з використанням ознаки несправності, що описується залежністю (70).

Нехай під час передачі даних по каналу зв'язку виникла помилка одного біта. Тоді в помилковій послідовності $e = (e_6, e_5, e_4, e_3, e_2, e_1, e_0)$ один з її членів буде дорівнювати 1.

Так, як і раніше, будемо вважати, що в каналі передачі інформації до передаваної кодованої послідовності v додається помилкова послідовність e і приймається послідовність u, тобто

$$u = v + e = \left(a_3 + e_6, a_2 + e_5, a_1 + e_4, a_0 + e_3, c_2 + e_2, c_1 + e_1, c_0 + e_0\right) \pmod 2. \tag{72}$$

За прийнятою послідовністю u за допомогою матриці H контролю парності виявляється ознака несправності:

$$S = Hu^T = H(v+e)^T = \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right)\begin{bmatrix} a_3 + e_6 \\ a_2 + e_5 \\ a_1 + e_4 \\ a_0 + e_3 \\ c_2 + e_2 \\ c_1 + e_1 \\ c_0 + e_0 \end{bmatrix} = \alpha^6(a_3 + e_6) + \tag{73}$$

$$+ \alpha^5(a_2 + e_5) + \alpha^4(a_1 + e_4) + \alpha^3(a_0 + e_3) + \alpha^2(c_2 + e_2) + \alpha^1(c_1 + e_1) + \alpha^0(c_0 + e_0) =$$

$$= \left(\alpha^6 a_3 + \alpha^5 a_2 + \alpha^4 a_1 + \alpha^3 a_0 + \alpha^2 c_2 + \alpha^1 c_1 + \alpha^0 c_0\right) +$$

$$+ \left(\alpha^6 e_6 + \alpha^5 e_5 + \alpha^4 e_4 + \alpha^3 e_3 + \alpha^2 e_2 + \alpha^1 e_1 + \alpha^0 e_0\right) =$$

$$= \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right) \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} + \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right) \begin{bmatrix} e_6 \\ e_5 \\ e_4 \\ e_3 \\ e_2 \\ e_1 \\ e_0 \end{bmatrix} = Hv^T + He^T. \quad (73)$$

Перша складова ознаки несправності Hv^T несе інформацію щодо несправності на передавальній стороні. Оскільки тут помилок немає, то $Hv^T = 0$. На приймальній стороні ознака несправності має вигляд

$$S = He^T = \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right)\left(e_6, e_5, e_4, e_3, e_2, e_1, e_0\right)^T =$$

$$= \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right) \begin{bmatrix} e_6 \\ e_5 \\ e_4 \\ e_3 \\ e_2 \\ e_1 \\ e_0 \end{bmatrix} = \quad (74)$$

$$= \alpha^6 \cdot e_6 + \alpha^5 \cdot e_5 + \alpha^4 \cdot e_4 + \alpha^3 \cdot e_3 + \alpha^2 \cdot e_2 + \alpha^1 \cdot e_1 + \alpha^0 \cdot e_0 \ .$$

З урахуванням того, що зі всіх помилкових бітів e_6—e_0 лише один біт дорівнює 1, можна встановити взаємно однозначну відповідність між ознакою несправності та помилковою послідовністю:

$e_6 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^6 \rightarrow (1\ 0\ 0\ 0\ 0\ 0\ 0)$

$e_5 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^5 \rightarrow (0\ 1\ 0\ 0\ 0\ 0\ 0)$

$e_4 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^4 \rightarrow (0\ 0\ 1\ 0\ 0\ 0\ 0)$

$e_3 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^3 \rightarrow (0\ 0\ 0\ 1\ 0\ 0\ 0)$

$e_2 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^2 \rightarrow (0\ 0\ 0\ 0\ 1\ 0\ 0)$

$e_1 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^1 \rightarrow (0\ 0\ 0\ 0\ 0\ 1\ 0)$

$e_0 = 1$, *інші біти дорівнюють* $0 \rightarrow S = \alpha^0 \rightarrow (0\ 0\ 0\ 0\ 0\ 0\ 1)$

$e_6 - e_1 = 0 \qquad \rightarrow S = 0 \rightarrow (0\ 0\ 0\ 0\ 0\ 0\ 0)$.

Зведемо отримані дані до табл. 12, що встановлює кореляцію між ознакою несправності та помилковою послідовністю.

За ознакою несправності і встановленою відповідністю помилковій послідовності легко відновити початкову кодовану послідовність.

Розглянемо процес декодування у випадку, наприклад, помилки третього біта (a_1), тобто $a_1 + 1 = \overline{a}_1 \pmod 2$. При цьому прийнята кодована послідовність має вигляд $u = (a_3, a_2, \overline{a}_1, a_0, c_2, c_1, c_0)$. Визначимо ознаку несправності

Таблиця 12. Ознака несправності та помилкова послідовність

Ознака несправності		Помилкова послідовність
Елемент $GF(2^3)$	Вектори-стовпці	
0	$(0\ 0\ 0)^T$	(0 0 0 0 0 0 0)
α^6	$(1\ 0\ 1)^T$	(1 0 0 0 0 0 0)
α^5	$(1\ 1\ 1)^T$	(0 1 0 0 0 0 0)
α^4	$(1\ 1\ 0)^T$	(0 0 1 0 0 0 0)
α^3	$(0\ 1\ 1)^T$	(0 0 0 1 0 0 0)
α^2	$(1\ 0\ 0)^T$	(0 0 0 0 1 0 0)
α^1	$(0\ 1\ 0)^T$	(0 0 0 0 0 1 0)
α^0	$(0\ 0\ 1)^T$	(0 0 0 0 0 0 1)

$$S = Hu^T = \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right)\left(a_3, a_2, \overline{a}_1, a_0, c_2, c_1, c_0\right)^T =$$

$$= \left(\alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0\right)\begin{bmatrix} a_3 \\ a_2 \\ \overline{a}_1 \\ a_0 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} =$$

$$= \alpha^6 \cdot a_3 + \alpha^5 \cdot a_2 + \alpha^4 \cdot \overline{a}_1 + \alpha^3 \cdot a_0 + \alpha^2 \cdot c_2 + \alpha^1 \cdot c_1 + \alpha^0 \cdot c_0 =$$

$$= \left(\alpha^6 \cdot a_3 + \alpha^5 \cdot a_2 + \alpha^4 \cdot a_1 + \alpha^3 \cdot a_0 + \alpha^2 \cdot c_2 + \alpha^1 \cdot c_1 + \alpha^0 \cdot c_0\right) + \alpha^4 \cdot a_1 + \alpha^4 \cdot \overline{a}_1.$$

У цій матриці вираз $\alpha^6 \cdot a_3 + \alpha^5 \cdot a_2 + \alpha^4 \cdot a_1 + \alpha^3 \cdot a_0 + \alpha^2 \cdot c_2 + \alpha^1 \cdot c_1 + \alpha^0 \cdot c = 0$, отже, ознака несправності може бути записана так:

$$S = Hu^T = \alpha^4\left(a_1 + \overline{a}_1\right) = \alpha^4 . \tag{75}$$

З табл. 12 знаходимо помилкову послідовність, що відповідає елементу α^4, — (0 0 1 0 0 0 0) . Якщо цю помилкову послідовність додати за модулем 2 до прийнятої кодованої послідовності, то помилка в ній буде виправлена, тобто

$$\left(a_3, a_2, \overline{a}_1, a_0, c_2, c_1, c_0\right) + \left(0\ 0\ 1\ 0\ 0\ 0\ 0\right) =$$

$$= \left(a_3 + 0,\ a_2 + 0,\ \overline{a}_1 + 1,\ a_0 + 0,\ c_2 + 0,\ c_1 + 0,\ c_0 + 0\right) = \tag{76}$$

$$= \left(a_3, a_2, a_1, a_0, c_2, c_1, c_0\right) = v.$$

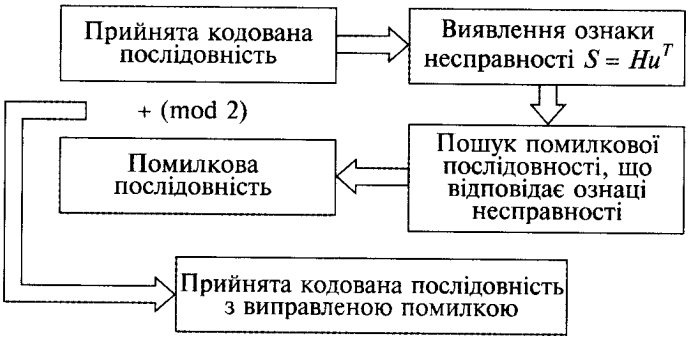


Рис. 53. Схема декодування лінійного коду

Такий процес декодування показано на рис. 53. Як бачимо, прийнята кодована послідовність після виявлення ознаки несправності та пошуку і установлення помилкової послідовності шляхом зіставлення з прийнятою послідовністю формує прийняту послідовність з виправленою помилкою.

У табл. 13 наведено відповідності між станами помилки коду Хеммінга (7, 4) і ознаками несправності, тобто станом контролю.

Проаналізуємо залежність між матрицею контролю парності і породжувальною матрицею. За ознакою несправності $S = Hv^T$ можна стверджувати: якщо v є кодованим словом, то несправність відсутня, тобто $S = Hv^T = 0$. У разі інформаційної послідовності a і породжувальної матриці G кодоване слово можна подати так (див. формулу (68)): $v = aG$. Оскільки транспонована матриця v має вигляд

$$v^T = (aG)^T = G^T a^T,$$

то

$$S = Hv^T = HG^T a^T = 0\,.$$

Таблиця 13. Відповідність між помилкою коду Хеммінга (7, 4) і станом контролю

Стан помилки		Стан контролю (ознака несправності)			
Помилковий біт	Помилкова послідовність	Елемент $GF(2^3)$	Многочлен за α	Вектори-рядки	Вектори-стовпці
Немає помилки	(0 0 0 0 0 0 0)	0	—	(0 0 0)	$(0\ 0\ 0)^T$
1	(1 0 0 0 0 0 0)	α^6	$\alpha^2 + 1$	(1 0 1)	$(1\ 0\ 1)^T$
2	(0 1 0 0 0 0 0)	α^5	$\alpha^2 + \alpha + 1$	(1 1 1)	$(1\ 1\ 1)^T$
3	(0 0 1 0 0 0 0)	α^4	$\alpha^2 + \alpha$	(1 1 0)	$(1\ 1\ 0)^T$
4	(0 0 0 1 0 0 0)	α^3	$\alpha + 1$	(0 1 1)	$(0\ 1\ 1)^T$
5	(0 0 0 0 1 0 0)	α^2	α^2	(1 0 0)	$(1\ 0\ 0)^T$
6	(0 0 0 0 0 1 0)	α^1	α	(0 1 0)	$(0\ 1\ 0)^T$
7	(0 0 0 0 0 0 1)	$\alpha^0 = 1$	1	(0 0 1)	$(0\ 0\ 1)^T$

З огляду на те, що дійсно величина $a^T \neq 0$, то $HG^T = 0$. Зазначимо також, що

$$HG^T = GH^T = 0 . \qquad (77)$$

Зі співвідношення (77), знаючи матрицю контролю парності, можна визначити породжувальну матрицю.

Тепер покажемо коректність використання терміну ***лінійний код***. Оскільки лінійний код згідно з (68) має вигляд $v = aG$, то легко встановити наступну залежність:

(***кодоване слово***) + (***кодоване слово***) = (***кодоване слово***) . (78)

Оскільки ця залежність є ***лінійною***, то і всі коди, що задовольняють співвідношення (77) — лінійні коди.

КОД ХЕММІНГА І ВИПРАВЛЕННЯ ПАКЕТНИХ ПОМИЛОК

На сьогодні існує багато кодів з виправленням помилок. Усі вони можуть бути поділені на два класи: ***блокові*** коди та коди ***згортки***.

Блоковим кодом називається конструкція, що містить у собі інформаційну частину (блок) і контрольну частину (блок) (рис. 54).

Тобто у разі використання блокового коду, кожний окремий блок можна незалежно кодувати і декодувати. Прикладом такого блокового коду є код Хеммінга.

Швидкість R_k блокового коду визначається відношенням

$$R_k = \frac{k}{n}. \tag{79}$$

Якщо повідомлення складається з великого числа бітів, то в принципі краще використовувати один кодовий блок великої довжини, ніж послідовність кодових слів з коротшого коду. Природа статистичних флуктуацій така, що випадкова конфігурація помилок, зазвичай, має вид пакетів помилок. Деякі сегменти цієї конфігурації містять у собі більше, ніж середнє число помилок, а деякі — менше. Отже, при одній і тій же швидкості довші кодові слова є у значній мірі менш чутливими до помилок, ніж коротші, але, зазвичай, відповідний кодер і декодер можуть бути складнішими.

У коді ***згортки***, що також належить до кодів Хеммінга, які є блоковими кодами, у контрольних бітах є інформація щодо попереднього блоку (рис. 55). Застосування кодів згортки дозволяє зменшити вірогідність помилок при передачі блоку даних, якщо навіть число помилок при цьому більше ніж одна.

Отже, попередній блок впливає на процес декодування поточного блоку. Діапазон, на якому позначається вплив попереднього блоку на поточний блок, називається довжиною зв'язку.

Залежно від характеру помилок, що виникають у каналі передачі, коди з виправленням помилок діляться на дві групи (рис. 56):

- коди з виправленням випадкових помилок;
- коди з виправленням пакетних помилок.

Наприклад, код Хеммінга — код з виправленням випадкових помилок.

Коди з виправленням помилок необхідно застосовувати з урахуванням того типу помилок, які ймовірніше за все можуть виникати в каналі передачі. Код БЧХ, що широко використовується в супутниковому зв'язку для передачі цифрових звукових сигналів, належить до кодів з виправленням випадкових помилок. У разі супутникового зв'язку довжина каналу передачі у вільному просторі вимірюється десятками тисяч кілометрів і тому необхідно враховувати можливість виникнення як випадкових, так і пакетних помилок.

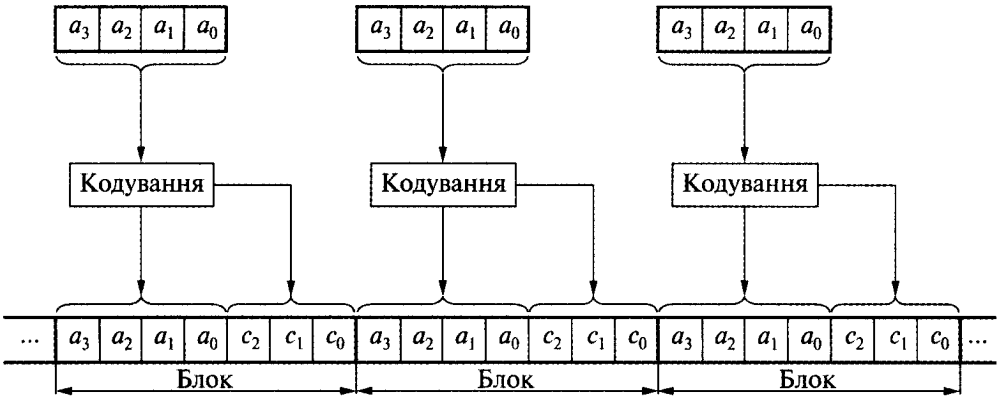


Рис. 54. Структура блокового коду

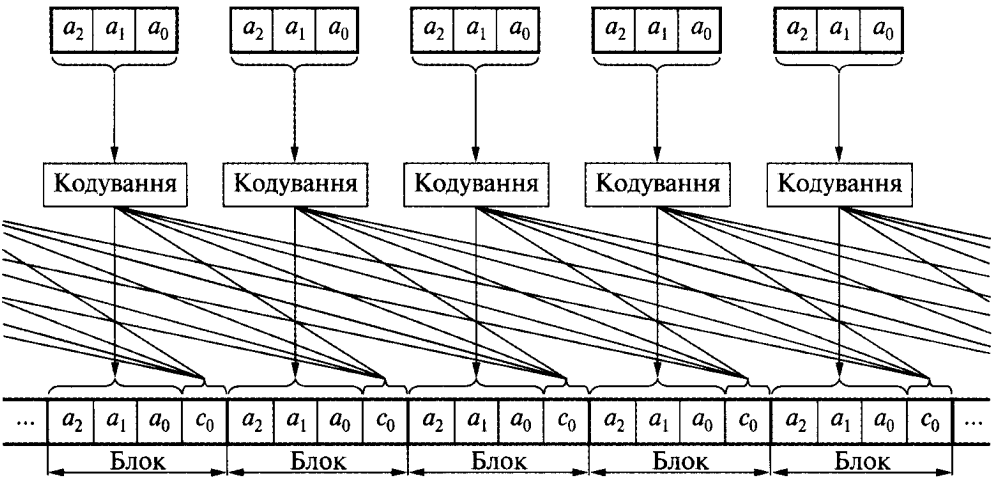


Рис. 55. Структура коду згортки

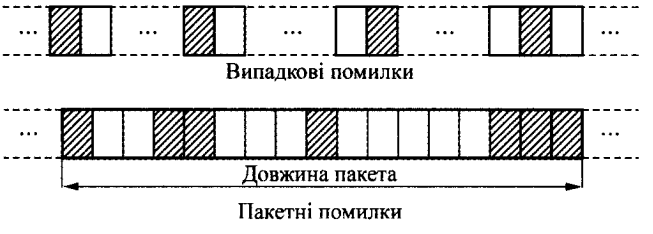


Рис. 56. Приклади помилок у каналі передачі сигналів

Для виправлення пакетних помилок при використанні коду з виправленням випадкових помилок необхідно перетворити пакетні помилки у випадкові. Це перетворення можна здійснити за допомогою операції переме-

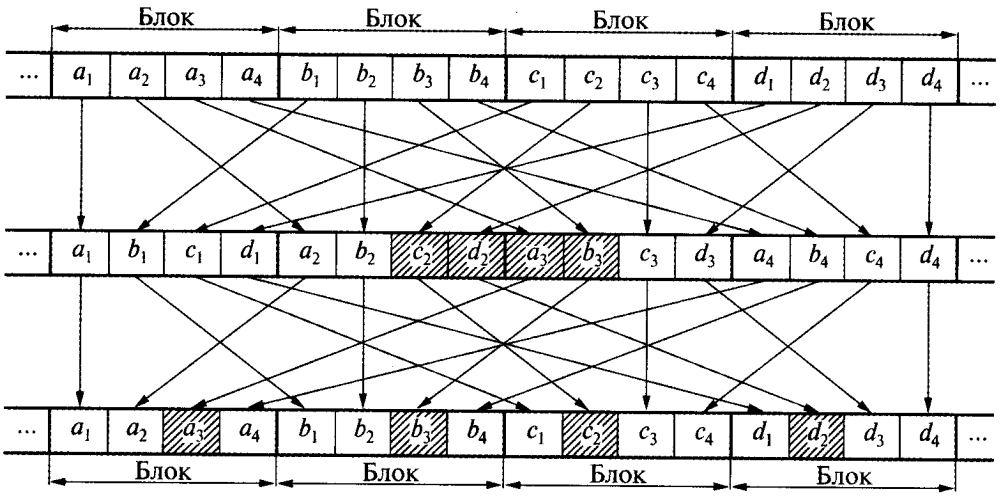


Рис. 57. Схема принципу перемежування (заштриховано помилковий біт)

жування (чергування). Суть цього методу полягає в тому, що при передачі кодові блоки розсіюють побітно, а при прийомі збирають воєдино кожен початковий блок (рис. 57).

Код Хеммінга хоч і є кодом з виправленням випадкової помилки, але з перемежуванням здатний виправляти і пакетні помилки. Взагалі код Хеммінга має безліч модифікацій. Він є базовим кодом для створення інших кодів. Знання цього коду дозволяє аналізувати будь-який інший блоковий код з виправленням помилки.

ДЕКОДУВАННЯ КОДУ БЧХ

Як приклад будемо використовувати коди БЧХ, що створюються породжувальним многочленом четвертого степеня $x^4 + x^2 + 1$. Для отримання кодів БЧХ необхідно обчислити мінімальні многочлени, що відповідають усім піднесеним до степеня кореням a первинного многочлена (рис. 58).

Степінь первинного многочлена — четвертий, тому значення коду БЧХ — $n = 2^4 - 1 = 15$. Множина побудована із залишків первинної формули $x^4 + x + 1$ є скінченною, містить 16 елементів, тобто $GF(2^4)$. Використовуючи корінь a первинного многочлена всі елементи множини $GF(2^4)$, за винятком нульового елемента, можна відобразити степенями кореня a.

Покажемо, що, якщо a послідовно підносити до степеня, то $a^{15} = a^0$, тобто період a дорівнюватиме 15.

Якщо a — корінь первинного многочлена, то

$$a^4 + a + 1 = 0,$$

тоді

$$a^4 = a + 1.$$

Таким чином, замість a^4 можна використовувати $a + 1$. Отже первинний многочлен можна записати у вигляді системи рівнянь його коренів:

$$\begin{cases} a^0 = 1 \\ a^1 = a \\ a^2 = a^2 \\ a^3 = a^3 \\ a^4 = a+1 \\ a^5 = a^2 + a \\ a^6 = a^3 + a^2 \\ a^7 = a^3 + a + 1 \\ a^8 = a^2 \\ a^9 = a^3 + a \\ a^{10} = a^2 + a + 1 \\ a^{11} = a^3 + a^2 + a \\ a^{12} = a^3 + a^2 + a + 1 \\ a^{13} = a^3 + a^2 + 1 \\ a^{14} = a^3 + 1 \\ a^{15} = 1 = a^0. \end{cases} \qquad (80)$$

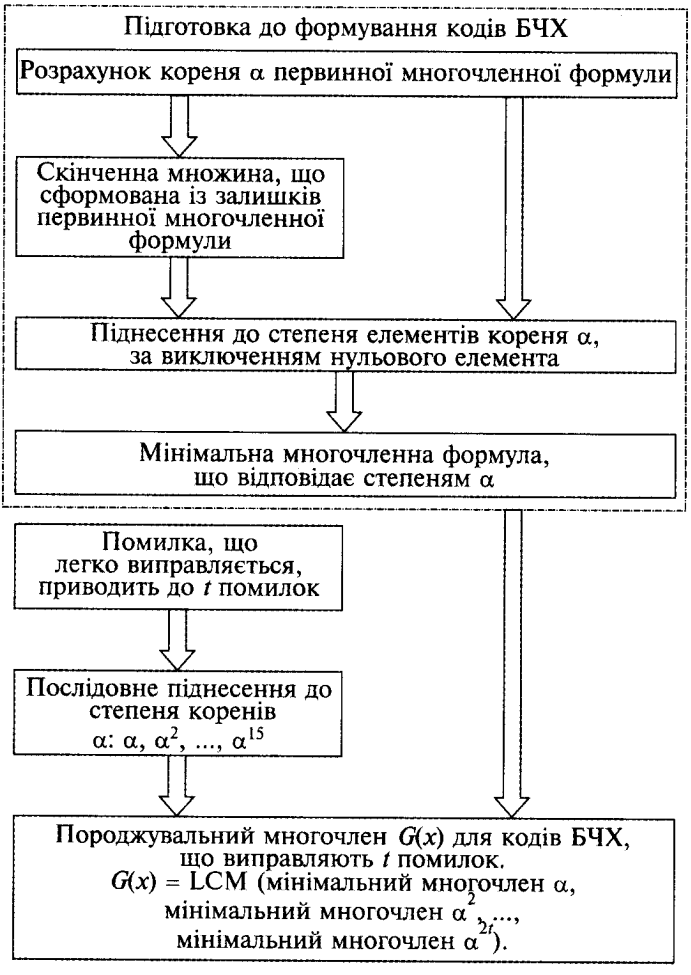


Рис. 58. Схема формування кодів БЧХ

Використовуючи систему рівнянь (80), елементи множини $GF(2^4)$ можна подати як у вигляді многочлена початкового елемента a, так і векторами у вигляді рядків чи стовпців (табл. 14).

До якого б степеня не підносився початковий елемент a, нульовий елемент подати як многочлен неможливо, тому для нього і не існує запису многочлена з коренем a. Для всіх інших елементів множини $GF(2^4)$ (за винятком нульового) мінімальний многочлен існує.

Спочатку знайдемо мінімальний многочлен для початкового елемента a. Запишемо ряд для елементів a: a, a^2, a^4, a^8, $a^{16} = a^{15}a = a$, $a^{32} = a^{15}a^{15}a^2 = a^2$, ... Тобто чотири елементи a, a^2, a^4, a^8 мають однаковий мінімальний многочлен. Його коренями є елементи a зі зростаючими степенями : a, a^2, a^4, a^8, тому визначити його можна так:

$$(x-a)(x-a^2)(x-a^4)(x-a^8) =$$

$$= (x+a)(x+a^2)(x+a^4)(x+a^8) = x^4 + Ax^3 + Bx^2 + Cx + D.$$

Коефіцієнти A, B, C, D обчислюються за допомогою системи рівнянь (80).

Таблиця 14. Подання елементів множини $GF(2^4)$ многочленом і вектором

Елементи $GF(2^4)$	Многочлен a	Вектор		Елементи $GF(2^4)$	Многочлен a	Вектор	
		Рядок	Стовпець			Рядок	Стовпець
0	—	0 0 0 0	0 0 0 0	a^8	$a^2 + 1$	0 1 0 1	0 1 0 1
a^0	1	0 0 0 1	0 0 0 1	a^9	$a^3 + a$	1 0 1 0	1 0 1 0
a^1	a	0 0 1 0	0 0 1 0	a^{10}	$a^2 + a + 1$	0 1 1 1	0 1 1 1
a^2	a^2	0 1 0 0	0 1 0 0	a^{11}	$a^3 + a^2 + a$	1 1 1 0	1 1 1 0
a^3	a^3	1 0 0 0	1 0 0 0	a^{12}	$a^3 + a^2 + a + 1$	1 1 1 1	1 1 1 1
a^4	$a + 1$	0 0 1 1	0 0 1 1	a^{13}	$a^3 + a^2 + 1$	1 1 0 1	1 1 0 1
a^5	$a^2 + a$	0 1 1 0	0 1 1 0	a^{14}	$a^3 + 1$	1 0 0 1	1 0 0 1
a^6	$a^3 + a^2$	1 1 0 0	1 1 0 0	a^{15}	1	0 0 0 1	0 0 0 1
a^7	$a^3 + a + 1$	1 0 1 1	1 0 1 1				

Розрахуємо значення коефіцієнта A біля x^3 з використанням елементів a:

$$A = a + a^2 + a^4 + a^8 = a + a^2 + (a+1) + (a^2+1) = 2a^2 + 2a + 2 = 0 .$$

Визначимо коефіцієнт B біля x^2 з використанням елементів a:

$$B = aa^2 + aa^4 + aa^8 + a^2a^4 + a^2a^8 + a^4a^8 = a^3 + a^5 + a^9 + a^6 + a^{10} + a^{12} =$$
$$= a^3 + (a^2 + a) + (a^3 + a) + (a^3 + a^2) + (a^2 + a + 1) + (a^3 + a^2 + a + 1) =$$
$$= 4a^3 + 4a^2 + 4a + 2 = 0.$$

Значення коефіцієнта C біля x описується рівнянням

$$C = aa^2a^4 + aa^2a^8 + aa^4a^8 + a^2a^4a^8 = a^7 + a^{11} + a^{13} + a^{14} =$$
$$= (a^3 + a + 1) + (a^3 + a^2 + a) + (a^3 + a^2 + 1) + (a^3 + 1) = 4a^3 + 2a^2 + 2a + 3 = 1 .$$

Коефіцієнт D біля $x^0 = 1$ можна розрахувати так:

$$D = aa^2a^4a^8 = a^{15} = 1 .$$

Як наслідок, отримаємо

$$(x - a)(x - a^2)(x - a^4)(x - a^8) = x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 1 = x^4 + x + 1 .$$

Отже, мінімальний многочлен для коренів a, a^2, a^4, a^8 має вигляд $x^4 + x + 1$. Тепер знайдемо могочлен для елемента ряду a^3:

$$a^3, a^6, a^{12}, a^{24} = a^{15}a^9, \ a^{48} = (a^{24})^2 = (a^9)^2 = a^{18} = a^{15}a^3 = a^3, \ ...$$

Коренями цього мінімального многочлена є елементи a зі степенями: a^3, a^6, a^{12}, a^9. Його можна визначити з виразу

$$(x - a^3)(x - a^6)(x - a^{12})(x - a^9) = (x + a^3)(x + a^6)(x + a^{12})(x + a^9) =$$
$$= x^4 + Ax^3 + Bx^2 + Cx + D .$$

Використавши систему рівнянь (80), розрахуємо чисельні значення коефіцієнтів A, B, C, D:

$$A = a^3 + a^6 + a^{12} + a^9 = a^3 + (a^3 + a^2) + (a^3 + a^2 + a + 1) + (a^3 + a) =$$
$$= 4a^3 + 2a^2 + 2a + 1 = 1 ;$$

$$B = a^3a^6 + a^3a^{12} + a^3a^9 + a^6a^{12} + a^6a^9 + a^9a^{12} = a^9 + a^{15} + a^{12} + a^{18} + a^{15} + a^{21} =$$
$$= (a^3 + a) + 1 + (a^3 + a^2 + a + 1) + a^3 + 1 + (a^3 + a^2) = 4a^3 + 2a^2 + 2a + 3 = 1 ;$$

$$C = a^3a^6a^{12} + a^3a^6a^9 + a^3a^9a^{12} + a^6a^9a^{12} = a^{21} + a^{18} + a^{24} + a^{27} =$$
$$= (a^3 + a^2) + (a^3 + a) + a^3 + (a^3 + a^2 + a + 1) = 4a^3 + 2a^2 + 2a + 1 = 1 ;$$

$$D = a^3a^6a^9a^{12} = a^{30} = 1 .$$

Отже, мінімальний многочлен для a^3, a^6, a^{12}, a^9 має вигляд

$$x^4 + x^3 + x^2 + x + 1 .$$

Аналогічно знайдемо ряд для елемента a^5:

$$a^5, a^{10}, a^{20} = a^{15}a^5 = a^5, a^{40} = (a^{20})^2 = (a^5)^2 = a^{10}, \ ...$$

Тобто ряд містить лише елементи a^5, a^{10}. Запишемо мінімальний многочлен для цих коренів:

$$(x - a^5)(x - a^{10}) = (x + a^5)(x + a^{10}) = x^2 + (a^5 + a^{10})x + a^{15} =$$
$$= x^2 + [(a^2 + a) + (a^2 + a + 1)]x + 1 = x^2 + x + 1 .$$

Після цього залишається визначити мінімальний многочлен для елементів a^7, a^{11}, a^{13}, a^{14}, a^{15}. Для елемента a^7 ряд має вигляд:

$$a^7, a^{14}, a^{28} = a^{15}a^{13} = a^{13}, a^{56} = \left(a^{28}\right)^2 = \left(a^{13}\right)^2 = a^{26} =$$

$$= a^{15}a^{11} = a^{11}, a^{112} = \left(a^{56}\right)^2 = \left(a^{11}\right)^2 = a^{22} = a^7, \ldots$$

Таким чином визначаємо ряд a^7, a^{14}, a^{13}, a^{11}, для якого мінімальний многочлен розраховується із застосуванням отриманих коренів за допомогою добутку співмножників:

$$\left(x - a^7\right)\left(x - a^{14}\right)\left(x - a^{13}\right)\left(x - a^{11}\right) = \left(x + a^7\right)\left(x + a^{14}\right)\left(x + a^{13}\right)\left(x + a^{11}\right) =$$

$$= x^4 + Ax^3 + Bx^2 + Cx + D \ .$$

Використовуючи систему рівнянь (80), розраховуємо значення коефіцієнтів A, B, C, D для многочлена з рядом коренів a^7, a^{14}, a^{13}, a^{11}:

$$A = a^7 + a^{14} + a^{13} + a^{11} = \left(a^{13} + a + 1\right) + \left(a^3 + 1\right) + \left(a^3 + a^2 + 1\right) + \left(a^3 + a^2 + a\right) =$$

$$= 4a^3 + 2a^2 + 2a + 3 = 1 \ ,$$

$$B = a^7a^{14} + a^7a^{13} + a^7a^{11} + a^{14}a^{13} + a^{14}a^{11} + a^{13}a^{11} = a^{21} + a^{20} + a^{18} + a^{27} + a^{25} + a^{24} =$$

$$= \left(a^3 + a^2\right) + \left(a^2 + a\right) + a^3 + \left(a^3 + a^2 + a + 1\right) + \left(a^2 + a + 1\right) + \left(a^3 + a\right) =$$

$$= 4a^3 + 4a^2 + 4a + 2 = 0 \ ;$$

$$C = a^7a^{14}a^{13} + a^7a^{14}a^{11} + a^7a^{13}a^{11} + a^{14}a^{13}a^{11} = a^{34} + a^{32} + a^{31} + a^{38} =$$

$$= (a + 1) + a^2 + a + \left(a^2 + 1\right) = 2a^2 + 2a + 2 = 0 \ ;$$

$$D = a^7a^{14}a^{13}a^{11} = a^{45} = 1 \ .$$

Отже, мінімальним многочленом у цьому разі є многочлен $x^4 + x^3 + 1$.

Нарешті, залишився лише елемент a^{15}. Ряд для елемента a^{15} має вигляд $a^{15}(=1)$, $a^{30} = a^{15}a^{15} = a^{15}(=1)$. Тому елемент a^{15} буде єдиним членом ряду і мінімальним многочленом для нього є рівняння $x - a^{15} = x + 1$. Отримана вище відповідність між кожним елементом і його многочленом наведена в табл. 15.

Таблиця 15. Взаємозв'язок між мінімальними многочленами та відповідними їм елементами множини $GF(2^4)$

Елементи множини $GF(2^4)$	Мінімальний многочлен	Елементи множини $GF(2^4)$	Мінімальний многочлен
0	—	a^8	$x^4 + x + 1$
$a^0(=1)$	$x + 1$	a^9	$x^4 + x^3 + x^2 + x + 1$
a	$x^4 + x + 1$	a^{10}	$x^2 + x + 1$
a^2	$x^4 + x + 1$	a^{11}	$x^4 + x^3 + 1$

Закінчення табл. 15

Елементи множини $GF(2^4)$	Мінімальний многочлен	Елементи множини $GF(2^4)$	Мінімальний многочлен
a^3	$x^4 + x^3 + x^2 + x + 1$	a^{12}	$x^4 + x^3 + x^2 + x + 1$
a^4	$x^4 + x + 1$	a^{13}	$x^4 + x^3 + 1$
a^5	$x^2 + x + 1$	a^{14}	$x^4 + x^3 + 1$
a^6	$x^4 + x^3 + x^2 + x + 1$	$(a^{15} = a^0)$	$(x + 1)$
a^7	$x^4 + x^3 + 1$		

На цьому підготовка до формування кодів БЧХ з бітовим числом $n = 15$ завершена. Далі сформуємо код БЧХ для кодів TEC, SEC, DEC з бітовим числом $n = 15$ та розглянемо загальний спосіб декодування.

КОДИ БЧХ, ЩО ВИПРАВЛЯЮТЬ ПОМИЛКИ

§14.1. КОДИ БЧХ, ЩО ВИПРАВЛЯЮТЬ ОДИНИЧНІ ПОМИЛКИ

Для кодів БЧХ, що виправляють одиничні помилки, коренем первинного многочлена $x^4 + x + 1$ є початковий елемент a. Для первинного многочлена четвертого порядку кодове бітове число дорівнює 15. Для кодів SEC, якщо $t = 1$ (див. рис. 11), то коренями породжувального многочлена кодів БЧХ є такі зростаючі степені елемента a: a, a^2.

З табл. 15 визначають мінімальні многочлени: для кореня a — це многочлен $x^4 + x + 1$, для кореня a^2 — многочлен $x^4 + x + 1$.

Отже, породжувальний многочлен, що з'являється при послідовному піднесенні кореня a до першого та другого степенів, — це один і той самий многочлен $G(x) = x^4 + x + 1 = LCM$ (найменший загальний многочлен для a і найменший загальний многочлен для a^2).

Таким чином, у разі кодів SEC первинний многочлен буде породжувальним многочленом, тобто многочленом найменшого степеня, що належить циклічному коду. Контрольне бітове число є порядком породжувального многочлена і $m = 4$. Отже, ці коди БЧХ мають такі параметри:

- кодове бітове число $n = 15$;
- контрольне бітове число $m = 4$;
- інформаційне бітове число $k = 11$;
- можливість корекції — SEC;
- породжувальний многочлен — $x^4 + x + 1$,

і є кодами SEC (15, 11) БЧХ.

Структура інформаційного біта і контрольного біта (15, 11) БЧХ кодів наведена в табл. 16.

Оскільки код БЧХ є циклічним кодом, що використовується при послідовному зв'язку і не тільки для кодового захисту, а й для перетворення інформаційного слова з метою виключення довгих послідовностей нулів або одиниць, які погано передаються по лініях зв'язку, то спосіб отримання контрольних бітів, тобто процедуру кодування, можна здійснити на підставі способу, наведеного раніше. У кодах БЧХ, які є блоковими коректувальними кодами, кодові комбінації складаються з двох частин: інформаційної і перевірочної. Їх символи завжди займають одні і ті самі позиції, тобто розташовуються на певних місцях. Як правило, у таких кодах, усі кодові комбінації яких містять у собі n символів, перші k символів — інформаційні, а за ними $(n–k)$ — перевірочні (контрольні) символи. Відповідно до цього такі коди умовно позначили (n, k)-коди.

Таблиця 16. Структура інформаційного і контрольного бітів коду БЧХ типу (15, 11)

Інформаційні біти (11 бітів)											Контрольні біти (4 біта)			
a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	c_3	c_2	c_1	c_0
1 біт	2 біт	3 біт	4 біт	5 біт	6 біт	7 біт	8 біт	9 біт	10 біт	11 біт	11 біт	12 біт	13 біт	14 біт

Для того щоб дізнатися можливість корекції, слід з'ясувати кодову відстань. Останню можна визначити, розглянувши як змінюється кожен біт у всіх кодових словах при зміні інформаційного біта. Для цього з'ясуємо, як підставити інформаційні біти в контрольні біти, а також прослідкуємо, який біт змінюється при зміні кожного інформаційного біта.

Контрольні біти коду БЧХ отримують з інформаційних бітів з використанням наступних перетворень (рис. 59):

$$c_3 = a_{10} + a_9 + a_8 + a_7 + a_5 + a_3 + a_2 \pmod 2,$$

$$c_2 = a_9 + a_8 + a_7 + a_6 + a_4 + a_2 + a_1 \pmod 2,$$

$$c_1 = a_8 + a_7 + a_6 + a_5 + a_3 + a_1 + a_0 \pmod 2,$$

$$c_0 = a_{10} + a_9 + a_8 + a_6 + a_4 + a_3 + a_0 \pmod 2.$$

З цих формул та з даних рис. 59 видно, що особливістю кодів БЧХ є те, що перевірні (коректувальні) символи утворюються за допомогою лінійних операцій над інформаційними. Крім того, будь-яка дозволена кодова комбінація може бути отримана внаслідок лінійної операції над набором k лінійно незалежних кодових комбінацій. Зокрема, додавання за модулем 2 двох і більше дозволених комбінацій також дає дозволену кодову комбінацію. Оскільки теоретичною основою одержання таких комбінацій є математичний апарат лінійної алгебри, то коди назвали лінійними. А з урахуванням того, що перевірні символи формуються за визначеною системою (певними правилами), блокові рівномірні роздільні лінійні коди отримали назву систематичних. Використання апарату лінійної алгебри, в якій важливе значення має поняття "група", породило й іншу назву цих кодів – групові. Ці коди найбільш застосовні в системах передачі дискретної інформації.

Зрозуміло, що при зміні інформаційного біта, контрольний біт, який містить у собі цей інформаційний біт, також змінюється. Тому, для з'ясування, яким чином контрольний біт містить у собі інформаційний біт, звернемося до табл. 17.

З табл. 17 неважко виявити, як змінюються всі кодовані слова при зміні одного інформаційного біта. З неї також видно, що якщо один й той самий інформаційний біт (наприклад, a_0) входить до складу двох контрольних бітів (наприклад, c_1, c_0), то значення бітового числа, отриманого зміною інформаційного біта, дорівнює трьом. Якщо один і той самий інформаційний біт входить до складу трьох контрольних бітів, то значення бітового числа, одержа-

ного зміною інформаційного біта, дорівнює чотирьом, якщо ж той самий інформаційний біт входить до складу чотирьох контрольних бітів, то значення бітового числа — п'ять. Як наслідок, отримуємо мінімальну кодову відстань, що дорівнює трьом.

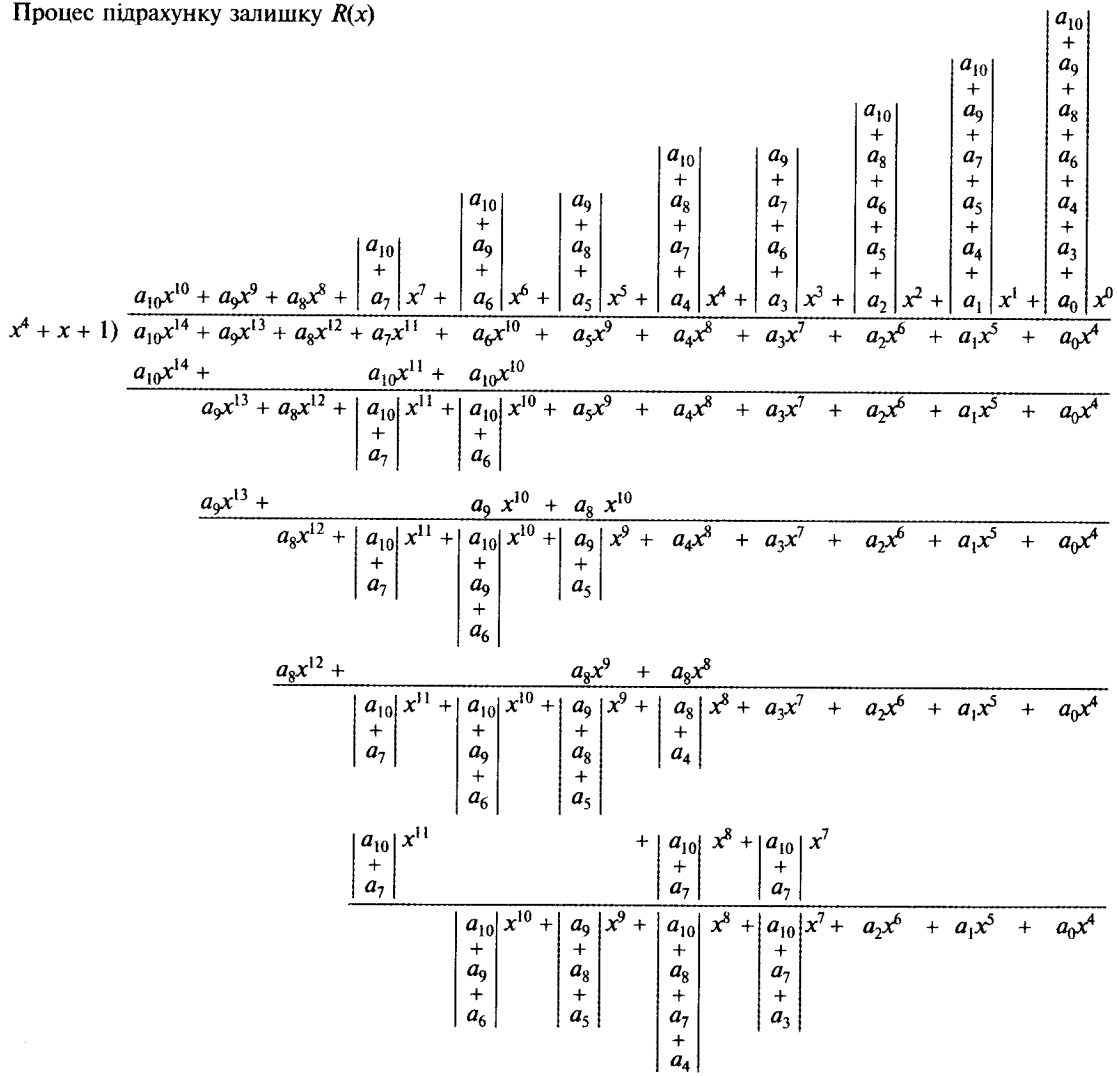
Отже, для кодів БЧХ (15, 11) можливий процес SEC.

Проте мало побудувати циклічний код. Треба уміти виділити з нього можливі помилкові розряди, тобто ввести деякі розпізнавачі помилок, які виділяли б помилковий блок зі всіх інших. Оскільки циклічні коди — блокові, то кожен блок повинен мати свій розпізнавач. І тут вирішальними є властивості породжувального многочлена. Методика побудови циклічного коду така, що породжувальний многочлен бере участь в утворенні кожної кодової комбінації, тому будь-який многочлен циклічного коду ділиться на породжувальний без залишку. Але без залишку діляться тільки ті многочлени, які належать даному коду, тобто породжувальний многочлен дозволяє вибрати дозволені комбінації зі всіх можливих. Якщо при діленні циклічного коду на породжувальний многочлен буде отримано залишок, то це означає наступне: або в коді відбулася помилка, або це комбінація якогось іншого коду (заборонена комбінація), що для декодувального пристрою не має принципової різниці. За залишком і виявляється наявність забороненої комбінації, тобто помилка. Залишки від ділення многочленів є розпізнавачами помилок циклічних кодів.

Таблиця 17. Взаємозв'язок інформаційних, контрольних бітів та бітового числа, отриманого зміною інформаційного біта

Інформаційні біти	Контрольні біти, що містять у собі інформаційний біт	Бітове число, отримане зміною інформаційного біта
a_{10}	c_3, c_0	3
a_9	c_3, c_2, c_0	4
a_8	c_3, c_2, c_1, c_0	5
a_7	c_3, c_2, c_1	4
a_6	c_2, c_1, c_0	4
a_5	c_3, c_1	3
a_3	c_3, c_1, c_0	4
a_2	c_3, c_2	3
a_1	c_2, c_1	3
a_0	c_1, c_0	3

Процес підрахунку залишку $R(x)$



Інформаційна послідовність
$a = (a_{10}, a_9, a_8, a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$

Інформаційний многочлен
$A(x) = a_{10}x^{10} + a_9x^9 + a_8x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

Породжувальний многочлен
$G(x) = x^4 + x + 1$

Многочлен $A(x)$ помножений на породжувальний многочлен четвертого степеня x^4

$A(x)\cdot x^4 = a_{10}x^{14} + a_9x^{13} + a_8x^{12} + a_7x^{11} + a_6x^{10} + a_5x^9 + a_4x^8 + a_3x^7 + a_2x^6 + a_1x^5 + a_0x^4$

$A(x)$ віднесено до залишку $R(x)$ від $G(x)$

$R(x) = c_3x^3 + c_2x^2 + c_1x + c_0$

$c_3 = a_{10} + a_9 + a_8 + a_7 + a_5 + a_3 + a_2 \bmod 2$
$c_2 = a_9 + a_8 + a_7 + a_4 + a_2 + a_1 \bmod 2$
$c_1 = a_8 + a_7 + a_6 + a_5 + a_3 + a_1 + a_0 \bmod 2$
$c_0 = a_{10} + a_9 + a_8 + a_6 + a_4 + a_3 + a_0 \bmod 2$

$A(x)\cdot x^4 + R(x)$

Породжувальний многочлен $V(x)$
$V(x) = a_{10}x^{14} + a_9x^{13} + a_8x^{12} + a_7x^{11} + a_6x^{10} + a_5x^9 + a_4x^8 + a_3x^7 + a_2x^6 + a_1x^5 + a_0x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$

Кодований многочлен $V(x)$

Передавана кодована послідовність
$v = (a_{10}, a_9, a_8, a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0, c_3, c_2, c_1, c_0)$

$$
\begin{array}{l}
\left|\begin{array}{l} a_{10} \\ + \\ a_9 \\ + \\ a_6 \end{array}\right| x^{\cdots} + \left|\begin{array}{l} a_{10} \\ + \\ a_9 \\ + \\ a_6 \end{array}\right| x^{\cdots} + \left|\begin{array}{l} a_{10} \\ + \\ a_9 \\ + \\ a_6 \end{array}\right| x^{\cdots} \\
\hline
\left|\begin{array}{l} a_9 \\ + \\ a_8 \\ + \\ a_5 \end{array}\right| x^9 + \left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_7 \\ + \\ a_4 \end{array}\right| x^8 + \left|\begin{array}{l} a_9 \\ + \\ a_7 \\ + \\ a_6 \\ + \\ a_3 \end{array}\right| x^7 + \left|\begin{array}{l} a_{10} \\ + \\ a_9 \\ + \\ a_6 \\ + \\ a_2 \end{array}\right| x^6 + a_1 x^5 + a_0 x^4 \\
\hline
\left|\begin{array}{l} a_9 \\ + \\ a_8 \\ + \\ a_5 \end{array}\right| x^9 + \left|\begin{array}{l} a_9 \\ + \\ a_8 \\ + \\ a_5 \end{array}\right| x^6 + \left|\begin{array}{l} a_9 \\ + \\ a_8 \\ + \\ a_5 \end{array}\right| x^5 + a_0 x^4 \\
\hline
\left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_7 \\ + \\ a_4 \end{array}\right| x^8 + \left|\begin{array}{l} a_9 \\ + \\ a_7 \\ + \\ a_6 \\ + \\ a_3 \end{array}\right| x^7 + \left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_6 \\ + \\ a_5 \\ + \\ a_2 \end{array}\right| x^6 + \left|\begin{array}{l} a_9 \\ + \\ a_8 \\ + \\ a_5 \\ + \\ a_1 \end{array}\right| x^5 + a_0 x^4 \\
\hline
\left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_7 \\ + \\ a_4 \end{array}\right| x^8 + \left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_7 \\ + \\ a_4 \end{array}\right| x^5 + \left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_7 \\ + \\ a_4 \end{array}\right| x^4 \\
\hline
\left|\begin{array}{l} a_9 \\ + \\ a_7 \\ + \\ a_6 \\ + \\ a_3 \end{array}\right| x^7 + \left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_6 \\ + \\ a_5 \\ + \\ a_2 \end{array}\right| x^6 + \left|\begin{array}{l} a_{10} \\ + \\ a_9 \\ + \\ a_7 \\ + \\ a_5 \\ + \\ a_4 \\ + \\ a_1 \end{array}\right| x^5 + \left|\begin{array}{l} a_{10} \\ + \\ a_8 \\ + \\ a_7 \\ + \\ a_4 \\ + \\ a_0 \end{array}\right| x^4
\end{array}
$$

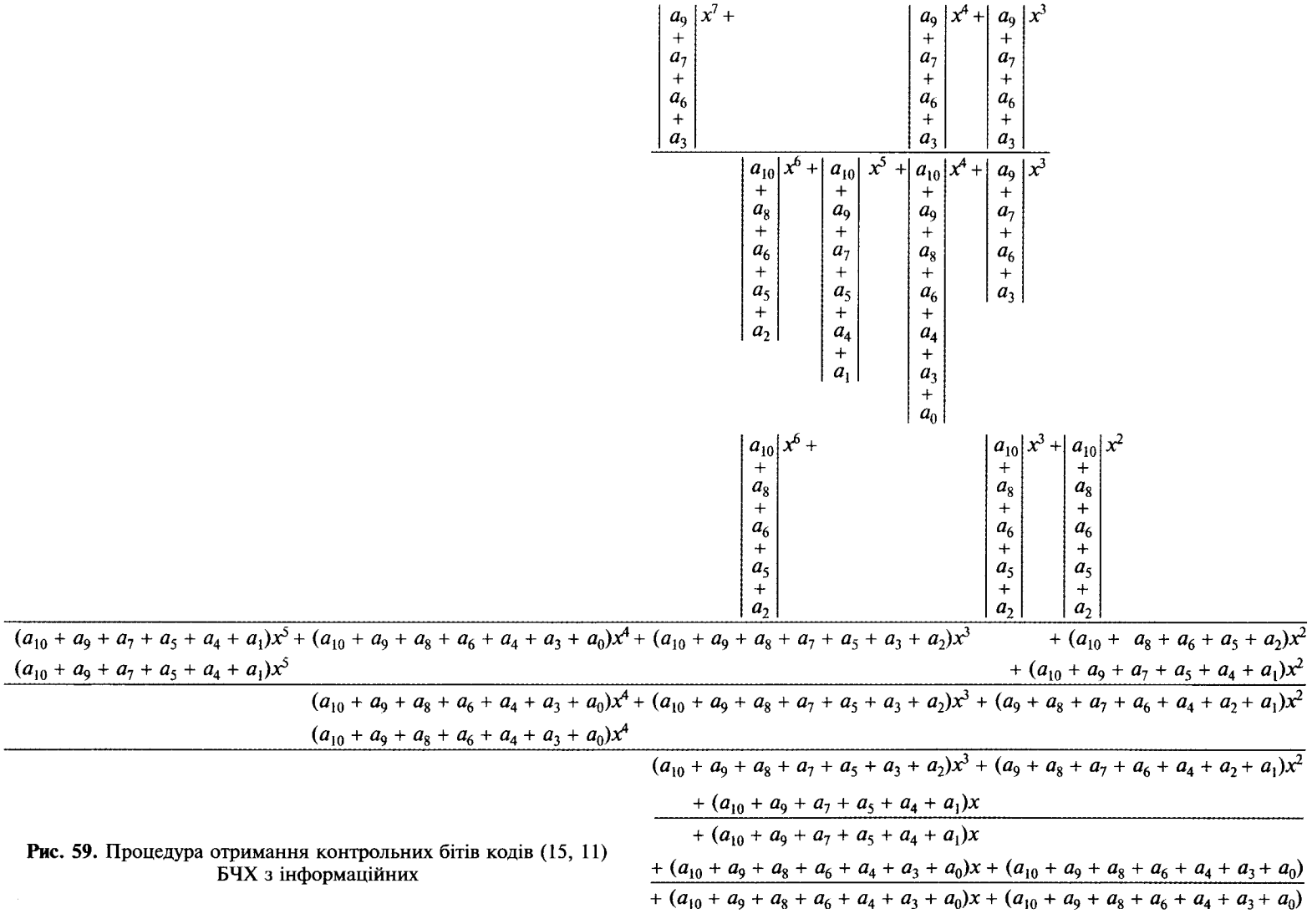


Рис. 59. Процедура отримання контрольних бітів кодів (15, 11) БЧХ з інформаційних

З іншого боку, залишки від ділення одиниці з нулями на породжувальний многочлен використовуються для побудови циклічних кодів. У разі такого ділення слід пам'ятати, що довжина залишку повинна бути не менша, ніж число контрольних розрядів, тому за відсутності розрядів у залишку до нього (залишку) справа дописують необхідне число нулів.

Послідовність контрольних бітів $m=n-k$ називається синдромом S. У ньому поєднуються ознаки, характерні певній відмінності від заданої (реальної) конфігурації. У даному випадку синдром характеризує певну конфігурацію помилок. Обчислення синдрому помилки виконується синдромним декодером, який ділить кодове слово на породжувальний многочлен. Якщо при діленні виникає залишок, то в слові є помилка. Залишок від ділення є синдромом помилки. Кількість можливих синдромів визначається за формулою

$$S = 2^m.$$

При числі перевірочних символів $m = 3$ є вісім можливих синдромів ($2^3 = 8$). Нульовий синдром (000) указує на те, що помилки при прийомі відсутні або не виявлені. Всякому ненульовому синдрому відповідає певна конфігурація помилок, яка і виправляється. Зокрема, кількість синдромів у класичних кодах Хеммінга є точно необхідною (що дозволяє виправити всі одиничні помилки в будь-якому інформативному і перевірному символах), і ці коди містять у собі один нульовий синдром.

Оскільки коди БЧХ є також і лінійними, то, зіставивши положення помилок і синдром помилки, що характеризується залишком, який виникає при діленні кодового слова на породжувальний многочлен, отримаємо методику виявлення положення помилок (табл. 18). Видно взаємно однозначну відповідність положення помилки (наприклад, помилка в першому біті (a_{10})) і синдрому (параметр a^{14} в матриці контролю парності, див. нижче).

У лінію передачі надходить кодова послідовність v:

$$v = [a_{10}, a_9, a_8, a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0, c_3, c_2, c_1, c_0].$$

Нехай помилка, що з'являється в лінії передачі, буде одиничною. Тобто помилкову послідовність запишемо так:

$$e = [e_{14}, e_{13}, e_{12}, e_{11}, e_{10}, e_9, e_8, e_7, e_6, e_5, e_4, e_3, e_2, e_1, e_0],$$

і в проміжку e_{14}—e_0 максимально можна зафіксувати не більше ніж одну помилку.

Додаючи при передачі до кодової послідовності v помилкову послідовність, отримуємо послідовність u, що передається:

$$u = v + e = [a_{10} + e_{14}, \ldots, c_0 + e_0]. \qquad (81)$$

Таблиця 18. Процедура виявлення помилкових бітів

Синдром	Положення помилки	Синдром	Положення помилки
0	Помилки немає	a^7	Помилка в 8-му біті (a_3)
a^{14}	Помилка в 1-му біті (a_{10})	a^6	Помилка в 9-му біті (a_2)
a^{13}	Помилка в 2-му біті	a^5	Помилка в 10-му біті (a_1)
a^{12}	Помилка в 3-му біті (a_8)	a^4	Помилка в 11-му біті (a_0)
a^{11}	Помилка в 4-му біті (a_7)	a^3	Помилка в 12-му біті (c_3)
a^{10}	Помилка в 5-му біті (a_6)	a^2	Помилка в 13-му біті (c_2)
a^9	Помилка в 6-му біті (a_5)	a^1	Помилка в 14-му біті (c_1)
a^8	Помилка в 7-му біті (a_4)	a^0	Помилка в 15-му біті (c_0)

Якщо тепер за допомогою передаваної кодової послідовності u і матриці контролю парності H обчислити синдром помилки S:

$$S = Hu^T, \tag{82}$$

то стає зрозумілим місцезнаходження помилкових бітів в інформаційній послідовності, що надійшла в лінію передачі сигналів.

Відповідно до табл. 18 матриця контролю парності H запишеться так:

$$H = \left[a^{14}, a^{13}, a^{12}, a^{11}, a^{10}, a^9, a^8, a^7, a^6, a^5, a^4, a^3, a^2, a^1, a^0\right]. \tag{83}$$

Степені коренів a відображаються чотирибітовими векторами (див. табл. 14), тому, якщо їх подати векторами-стовпцями, то матриця матиме такий вигляд:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{84}$$

Матриці контролю парності (83) і (84) ідентичні, але для того, щоб уникнути складності обчислення за формулою (82), матрицю H доцільно записати у вигляді (84).

Якщо підставити послідовність u, що передається, у формулу (82), то синдром S можна подати так:

$$S = Hu^T = H(v + e)^T = Hv^T + He^T.$$

Оскільки v є передаваною кодованою послідовністю, то

$$Hv^T = 0.$$

Отже, синдром S визначається тільки моделлю помилок, а саме:

$$S = He^T. \tag{85}$$

Розкладемо синдром помилки S за допомогою формули (83):

$$S = \left[a^{14}, a^{13}, a^{12}, a^{11}, a^{10}, a^9, a^8, a^7, a^6, a^5, a^4, a^3, a^2, a^1, a^0\right] \begin{bmatrix} e_{14} \\ e_{13} \\ e_{12} \\ e_{11} \\ e_{10} \\ e_9 \\ e_8 \\ e_7 \\ e_6 \\ e_5 \\ e_4 \\ e_3 \\ e_2 \\ e_1 \\ e_0 \end{bmatrix} = \tag{86}$$

$$= a^{14}e_{14} + a^{13}e_{13} + a^{12}e_{12} + a^{11}e_{11} + a^{10}e_{10} + a^{9}e_{9} + a^{8}e_{8} + a^{7}e_{7} + a^{6}e_{6} +$$

$$+ a^{5}e_{5} + a^{4}e_{4} + a^{3}e_{3} + a^{2}e_{2} + a^{1}e_{1} + a^{0}e_{0} .$$

Наприклад, якщо помилка відбулася в сьомому біті, то помилкова послідовність запишеться так: $e = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$.

При цьому тільки сьомий біт дорівнює одиниці, а решта значень помилкової послідовності — нулю. Відповідно до матриці (86) синдром S має вигляд: $S = a^8$. Стає зрозумілим процес виникнення помилки в сьомому біті.

Таким чином, здійснюється декодування лінійного коду.

Для декодування кодів БЧХ та визначення положення помилкових бітів за допомогою синдрому S виводиться рівняння помилок, після чого визначаються його корені. У разі кодів SEC допускається тільки один помилковий біт, для відновлення якого необхідно, щоб число залишків від ділення удвічі перевершувало число розрядів інформаційного слова. При цьому рівняння, що описує положення помилкового біта, — рівняння першого степеня, а саме:

$$S(x) = x + S . \qquad (87)$$

Отже, синдром S, наведений в табл. 18, є елементом множини $GF(2^4)$. Іншими словами, елементи множини $GF(2^4)$ показують положення помилкового біта. Відповідність елемента множини $GF(2^4)$ положенню помилкового біта відображена в табл. 19, із якої видно прямий зв'язок кожного елемента множини $GF(2^4)$ (наприклад, елемента a^{14}) з положенням помилкового біта (наприклад, для першого біта значення елемента помилкової послідовності — (e_{14})).

Параметри табл. 19 та 18 у другому стовпці повністю збігаються. Проте, безумовно, існує необхідність розрізняти синдром помилки S і елементи множини $GF(2^4)$: вони збігаються тільки у разі коду SEC.

Тому спочатку з передаваної кодованої послідовності за формулою (86) обчислюється синдром помилки S. Наприклад, у разі помилки в сьомому біті $S = a^8$. З обчисленого синдрому S виводиться рівняння, що описує положення помилок:

$$S(x) = x + S \qquad \left(S = a^8\right). \qquad (88)$$

Таблиця 19. Відповідність елемента множини $GF(2^4)$ положенню помилкового біта

Елементи $GF(2^4)$	Положення помилкового біта	Елементи $GF(2^4)$	Положення помилкового біта
0	Помилкового біта немає	a^7	8-й біт (e_7)
a^{14}	1-й біт (e_{14})	a^6	9-й біт (e_6)
a^{13}	2-й біт (e_{13})	a^5	10-й біт (e_5)
a^{12}	3-й біт (e_{12})	a^4	11-й біт (e_4)
a^{11}	4-й біт (e_{11})	a^3	12-й біт (e_3)
a^{10}	5-й біт (e_{10})	a^2	13-й біт (e_2)
a^9	6-й біт (e_9)	a^1	14-й біт (e_1)
a^8	7-й біт (e_8)	a^0	15-й біт (e_0)

У рівняння (88) по черзі підставляються елементи множини $GF(2^4)$

$$0,\ a^{14}, a^{13}, \ldots, a^{1}, a^{0}.$$

У разі, якщо

$$S(0) = 0, \tag{89}$$

то рівняння можна розв'язати безпомилково.

Якщо

$$S(x) \neq 0, \tag{90}$$

то виникає помилка. Підставляючи елементи множини, що залишилися:

$$S(a^{i}) = 0 \qquad (i = 14,\ldots,0), \tag{91}$$

за допомогою табл. 19 можна отримати кожен з п'ятнадцяти бітів, що характеризують положення помилкового біта.

Таким чином, враховуючи рівняння $S(x) = x + a^8$, маємо

$$S(a^{14}) = a^{14} + a^{8} = 0,$$

$$S\left(a^{13}\right) = a^{13} + a^{8} = 0,$$

.....

$$S(a^{8}) = a^{8} + a^{8} = 0.$$

Зрозуміло, що елемент a^8 є коренем рівняння $S(x)$. Знаючи положення помилкового біта, можливо виправити помилку, якщо інвертувати цей біт.

Остаточно послідовність декодування кодів БЧХ має такий вигляд:

1) обчислення синдрому з передаваної кодованої послідовності і матриці контролю парності;

2) виведення рівняння, що описує положення помилок, через синдром S;

3) обчислення коренів рівняння, що описує положення помилок, шляхом підстановки в нього значень елементів множини $GF(2^4)$;

4) визначення характеру помилки;

5) виправлення помилкового біта, положення якого визначене коренем рівняння.

Все це стосується тільки простого випадку кодів SEC. Далі розглянуті більш складні приклади з кодами DEC і TEC, для розуміння яких, проте, не існує ніяких завад.

§14.2. КОДИ БЧХ, ЩО ВИПРАВЛЯЮТЬ ПОДВІЙНІ ПОМИЛКИ

Методика побудови циклічних кодів, що виправляють одиничні помилки, відрізняється від методики побудови циклічних кодів, що виправляють багатократні помилки, тільки вибором породжувального многочлена. Побу-

дова породжувального многочлена виконується за допомогою так званих мінімальних многочленів, які є простими незвідними многочленами, і залежить в основному від двох параметрів: довжини кодового слова n і числа помилок s, що виправляються. Породжувальний многочлен — це добуток непарних мінімальних многочленів. Отже, він є їх найменшим спільним кратним (НСК). Решта параметрів, які використовують при побудові породжувального многочлена, залежно від заданих умов може бути визначена за допомогою таблиць і допоміжних співвідношень.

Позначимо корінь початкового многочлена $x^4 + x + 1$ як α. Якщо код БЧХ — це код з виправленням подвійних помилок, то параметр t дорівнює 2. Отже, породжувальний многочлен для такого коду БЧХ як корені включає послідовні степені кореня α, тобто α, α^2, α^3, α^4. Тоді мінімальні многочлени для кожного із степенів α (див. табл. 15) записують так:

$$\begin{aligned} &для \quad \alpha \rightarrow x^4 + x + 1 , \\ &для \quad \alpha^2 \rightarrow x^4 + x + 1 , \\ &для \quad \alpha^3 \rightarrow x^4 + x^3 + x^2 + x + 1 , \\ &для \quad \alpha^4 \rightarrow x^4 + x + 1 . \end{aligned}$$

Тому породжувальний многочлен $G(x)$ дорівнює НСК цих мінімальних многочленів, тобто подається у вигляді добутку:

$$G(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1. \qquad (92)$$

Визначивши породжувальний многочлен, можна знайти кількість перевірних (контрольних) бітів. У даному коді БЧХ:

- число кодових бітів $n = 15$;
- число контрольних бітів $m = 8$;
- число інформаційних бітів $k = 7$;
- здібність до виправлення помилок — виправлення подвійних помилок;
- породжувальний многочлен — $x^8 + x^7 + x^6 + x^4 + 1$.

Таким чином, це код (15, 7) БЧХ. Структура інформаційних і контрольних бітів коду (15, 7) БЧХ наведена в табл. 20, із якої видно їх послідовність.

Питання, пов'язані з кодуванням і перевіркою здатності коду до виправлення помилок, розглянуті в наступному розділі.

Код (15, 7) БЧХ стосовно стану контролю помилок містить два контрольні ряди:

1. Контрольний ряд для $x^4 + x + 1$ (ряд від α):

$$\alpha^{14}, \alpha^{13}, \alpha^{12}, \alpha^{11}, \alpha^{10}, \alpha^9, \alpha^8, \alpha^7, \alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0 .$$

Таблиця 20. Структура інформаційних і контрольних бітів коду (15, 7) БЧХ

Інформаційні біти (7 бітів)							Контрольні біти (8 бітів)							
a_6	a_5	a_4	a_3	a_2	a_1	a_0	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0
1 біт	2 біт	3 біт	4 біт	5 біт	6 біт	7 біт	8 біт	9 біт	10 біт	11 біт	12 біт	13 біт	14 біт	15 біт

2. Контрольний ряд для $x^4 + x^3 + x^2 + x + 1$ (ряд від α^3):

$$(\alpha^3)^{14}, (\alpha^3)^{13}, (\alpha^3)^{12}, (\alpha^3)^{11}, (\alpha^3)^{10}, (\alpha^3)^9, (\alpha^3)^8, (\alpha^3)^7, (\alpha^3)^6, (\alpha^3)^5, (\alpha^3)^4, (\alpha^3)^3, (\alpha^3)^2, (\alpha^3)^1, (\alpha^3)^0.$$

Ряд від α^3 не зміниться, якщо його записати так:

$$(\alpha^{14})^3, (\alpha^{13})^3, (\alpha^{12})^3, (\alpha^{11})^3, (\alpha^{10})^3, (\alpha^9)^3, (\alpha^8)^3, (\alpha^7)^3, (\alpha^6)^3, (\alpha^5)^3, (\alpha^4)^3, (\alpha^3)^3, (\alpha^2)^3, (\alpha^1)^3, (\alpha^0)^3.$$

За допомогою цих двох рядів можна визначити стан контролю, тобто синдром помилки. Відповідність між знайденим таким чином синдромом і станом помилки наведена в табл. 21. Видно взаємно однозначну відповідність між станом помилки (наприклад, помилка в п'ятнадцятому біті (c_0)) і бітами синдрому $S_1 = [(\alpha^0)^3 (= \alpha^0)]$ та $S_0 = \alpha^0$.

Згідно з табл. 21, матриця контролю парності має вигляд

$$H = \begin{bmatrix} (\alpha^{14})^3, (\alpha^{13})^3, \ldots, (\alpha^1)^3, (\alpha^0)^3 \\ \alpha^{14}, \alpha^{13}, \ldots, \alpha^1, \alpha^0 \end{bmatrix} =$$

$$= \begin{bmatrix} \alpha^{12}, \alpha^9, \alpha^6, \alpha^3, \alpha^0, \alpha^{12}, \alpha^9, \alpha^6, \alpha^3, \alpha^0, \alpha^{12}, \alpha^9, \alpha^6, \alpha^3, \alpha^0 \\ \alpha^{14}, \alpha^{13}, \alpha^{12}, \alpha^{11}, \alpha^{10}, \alpha^9, \alpha^8, \alpha^7, \alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0 \end{bmatrix}. \quad (93)$$

Якщо подати степені α стовпцями, то матриця контролю парності H запишеться так:

$$H = \begin{pmatrix} 111101111011110 \\ 101001010010100 \\ 110001100011000 \\ 100011000110001 \\ 111101011001000 \\ 011110101100100 \\ 001111010110010 \\ 111010110010001 \end{pmatrix}. \quad (94)$$

Код БЧХ (15, 7) є кодом з виправленням подвійних помилок, тому припустимо, що число помилкових бітів у помилковій послідовності $e = (e_{14}, e_{13}, e_{12}, e_{11}, e_9, e_8, e_7, e_6, e_5, e_4, e_3, e_2, e_1, e_0)$, що складається з інформаційною послідовністю при передачі по каналу зв'язку, не перевищує двох. У цьому випадку серед бітів e_{14}—e_0 число одиниць не перевищує двох. На приймальній стороні за матрицею контролю парності H і помилковою послідовністю e обчислюється синдром S. Матриця контролю парності (див. (93)) складається з двох рядків, тому синдром помилки S можна розкласти на два біти синдромів S_1, S_0, які відповідають різним рядкам матриці:

$$S = \begin{pmatrix} S_1 \\ S_0 \end{pmatrix} = He^T = \begin{pmatrix} (\alpha^{14})^3, (\alpha^{13})^3, \ldots, (\alpha^0)^3 \\ \alpha^{14}, \alpha^{13}, \ldots, \alpha^0 \end{pmatrix} e^T. \quad (95)$$

Таблиця 21. Відповідність між синдромом і станом помилки

Синдром		Стан помилки
S_1	S_0	
0	0	Помилки відсутні
$(\alpha^{14})^3$ (= α^{12})	α^{14}	Помилка в першому біті (a_6)
$(\alpha^{13})^3$ (= α^{9})	α^{13}	Помилка в другому біті (a_5)
$(\alpha^{12})^3$ (= α^{6})	α^{12}	Помилка в третьому біті (a_4)
$(\alpha^{11})^3$ (= α^{3})	α^{11}	Помилка в четвертому біті (a_3)
$(\alpha^{10})^3$ (= α^{0})	α^{10}	Помилка в п'ятому біті (a_2)
$(\alpha^{9})^3$ (= α^{12})	α^{9}	Помилка в шостому біті (a_1)
$(\alpha^{8})^3$ (= α^{9})	α^{8}	Помилка в сьомому біті (a_0)
$(\alpha^{7})^3$ (= α^{6})	α^{7}	Помилка у восьмому біті (c_7)
$(\alpha^{6})^3$ (= α^{3})	α^{6}	Помилка в дев'ятому біті (c_6)
$(\alpha^{5})^3$ (= α^{0})	α^{5}	Помилка в десятому біті (c_5)
$(\alpha^{4})^3$ (= α^{12})	α^{4}	Помилка в одинадцятому біті (c_4)
$(\alpha^{3})^3$ (= α^{9})	α^{3}	Помилка в дванадцятому біті (c_3)
$(\alpha^{2})^3$ (= α^{6})	α^{2}	Помилка в тринадцятому біті (c_2)
$(\alpha^{1})^3$ (= α^{3})	α^{1}	Помилка в чотирнадцятому біті (c_1)
$(\alpha^{0})^3$ (= α^{0})	α^{0}	Помилка в п'ятнадцятому біті (c_0)

З урахуванням матриці (95) відповідні біти синдромів S_1, S_0 записують так:

$$S_1 = \left[\left(\alpha^{14}\right)^3, \left(\alpha^{13}\right)^3, ..., \left(\alpha^{1}\right)^3, \left(\alpha^{0}\right)^3\right] e^T = \left[\left(\alpha^{14}\right)^3, \left(\alpha^{13}\right)^3, ..., \left(\alpha^{1}\right)^3, \left(\alpha^{0}\right)^3\right] \begin{bmatrix} e_{14} \\ e_{13} \\ . \\ . \\ . \\ e_1 \\ e_0 \end{bmatrix} =$$

$$= \left(\alpha^{14}\right)^3 \cdot e_{14} + \left(\alpha^{13}\right)^3 \cdot e_{13} + ... + \left(\alpha^{1}\right)^3 \cdot e_1 + \left(\alpha^{0}\right)^3 \cdot e_0 . \qquad (96)$$

$$S_0 = \left(\alpha^{14}, \alpha^{13}, ..., \alpha^{1}, \alpha^{0}\right) e^T = \alpha^{14} \cdot e_{14} + \alpha^{13} \cdot e_{13} + ... + \alpha^{1} \cdot e_1 + \alpha^{0} \cdot e_0 . \qquad (97)$$

Якщо позначити елементи групи $GF(2^4)$, що відображають положення двох помилкових бітів, як α^i і α^j, то рівняння розташування помилок набуде вигляду

$$S(x) = (x - \alpha^i)(x - \alpha^j) = x^2 + Ax + B . \qquad (98)$$

Якщо виразити коефіцієнти A і B цього рівняння через два біти синдрому S_1 і S_0, то надалі можна знайти його корені, послідовно підставляючи елементи групи $GF(2^4)$ замість аргументу x. Це, у свою чергу, дозволяє виправити помилкові біти, що знаходяться в позиціях, позначених коренями.

Наприклад, розглянемо випадок, коли помилки виникли у восьмому і дванадцятому бітах. Подамо послідовність помилок e:

$$e = (0000000010001000),$$

а саме: $e_7 = 1$, $e_3 = 1$, решта бітів дорівнюють нулю. Тоді, згідно з рівняннями (96) і (97), можна записати два біти синдромів:

$$S_1 = (\alpha^7)^3 + (\alpha^3)^3, \tag{99}$$

$$S_0 = \alpha^7 + \alpha^3. \tag{100}$$

Восьмий і дванадцятий біти є помилковими, тому елементи групи, що відповідають положенню цих помилкових бітів (див. табл. 19), приймають значення пари чисел: α^7 і α^3. Отже, рівняння розташування помилок має вигляд

$$S(\alpha^7) = 0,$$

$$S(\alpha^3) = 0.$$

З урахуванням рівняння розташування помилок (98) отримуємо

$$(\alpha^7)^2 + A(\alpha^7) + B = 0\ , \tag{101}$$

$$(\alpha^3)^2 + A(\alpha^3) + B = 0\ . \tag{102}$$

Виконавши піднесення виразу (100) до квадрата, прийдемо до виразу

$$S_0^{\,2} = (\alpha^7 + \alpha^3)^2 = (\alpha^7)^2 + (\alpha^3)^2. \tag{103}$$

Додавши формули (101) і (102), одержимо $\{(\alpha^7)^2 + (\alpha^3)^2\} + A\{\alpha^7 + \alpha^3\} + 2B = 0$. Відповідно до формул (100) і (103) $S_0^2 + A \cdot S_0 = 0$. Тоді $S_0 + A = 0$, $A = S_0$. Отже, знайдено значення коефіцієнта A. Для визначення коефіцієнта B помножимо формулу (101) на α^7, а формулу (102) на α^3, далі результати додамо:

$$\{(\alpha^7)^3 + A(\alpha^7)^2 + B\alpha^7\} + \{(\alpha^3)^3 + A(\alpha^3)^2 + B\alpha^3\} =$$

$$= \{(\alpha^7)^3 + (\alpha^3)^3\} + A\{(\alpha^7)^2 + (\alpha^3)^2\} + B(\alpha^7 + \alpha^3) = 0.$$

Використовуючи співвідношення (101)—(103), неважко отримати $S_1 + AS_0^2 + BS_0 = 0$. Підставивши в цей вираз значення $A = S_0$, знайдемо коефіцієнт B:

$$S_1 + S_0^3 + BS_0 = 0,$$

$$B = (S_1 + S_0^3)/S_0 = S_0^2 + (S_1/S_0)\ .$$

Отже, рівняння розташування помилок набуває вигляду

$$S(x) = x^2 + S_0 x + S_0^2 + S_1/S_0\ . \tag{104}$$

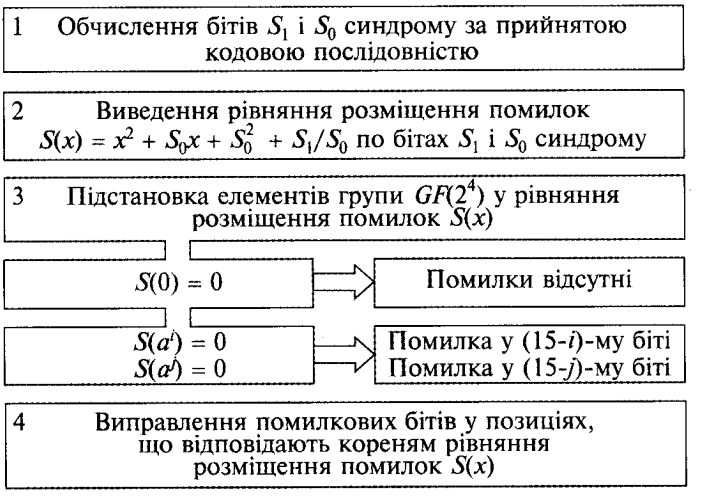


Рис. 60. Схема декодування коду (15, 7) БЧХ з виправленням подвійних помилок

Це рівняння можна розкласти на множники:

$$S_1 = (\alpha^7)^3 + (\alpha^3)^3 = (\alpha^7 + \alpha^3)\left\{(\alpha^7)^2 + (\alpha^7)(\alpha^3) + (\alpha^3)^2\right\},$$

$$S_0^2 + S_1 / S_0 = (\alpha^7)^2 + (\alpha^3)^2 + \left\{(\alpha^7)^2 + (\alpha^7)(\alpha^3) + (\alpha^3)^2\right\} = (\alpha^7)(\alpha^3).$$

Як наслідок, рівняння розташування помилок можна записати у вигляді добутку двох множників:

$$S(x) = (x + \alpha^7)(x + \alpha^3).$$

Підставивши в нього елементи групи $GF(2^4)$, отримаємо: $S(\alpha^7) = 0$, $S(\alpha^3) = 0$.

Таким чином, положення помилкових бітів можна вважати визначеними. Це дає можливість виправити помилки, інвертуючи (додаючи одиницю за модулем 2) помилкові біти, розташовані в тих позиціях, які позначені коренями α^7 і α^3. Повністю процес виправлення подвійних помилок наведено на рис. 60.

§14.3. КОДИ БЧХ, ЩО ВИПРАВЛЯЮТЬ ПОТРІЙНІ ПОМИЛКИ

Сигнал, що передається, розбивають на блоки певної довжини і фіксують середню вірогідність помилки в блоці. У цьому разі частка помилок, які потрібно виправляти, зменшується при зростанні довжини блока. Сказане свідчить про резерви поліпшення характеристик при усереднюванні шуму і про те, що ці резерви зростають при збільшенні довжини блока. Таким чином, довгі блокові коди ефективніші, ніж короткі. При збільшенні довжини блока частка помилкових символів в ньому наближається до середньої частоти помилок у каналі. А також, що дуже важливо, частка блоків, кількість помилок в яких істотно відрізняється від цього середнього значення, стає дуже малою.

Корінь початкового многочлена $x^4 + x + 1$ позначимо α. Породжувальний многочлен коду БЧХ з виправленням потрійних помилок відповідає випадку, коли параметр t дорівнює 3. Отже, він як корені містить у собі послідовно зростаючі степені кореня α. Відповідно до табл. 15 мінімальні многочлени для кожного із степенів α можна записати так:

для $\alpha \rightarrow x^4 + x + 1,$

для $\alpha^2 \rightarrow x^4 + x + 1,$

для $\alpha^3 \rightarrow x^4 + x^3 + x^2 + x + 1,$

для $\alpha^4 \rightarrow x^4 + x + 1,$

для $\alpha^5 \rightarrow x^2 + x + 1,$

для $\alpha^6 \rightarrow x^4 + x^3 + x^2 + x + 1.$

Тому породжувальний многочлен $G(x)$ для такого коду БЧХ дорівнює НСК мінімальних многочленів, тобто добутку трьох мінімальних многочленів:

$$G(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) =$$

$$= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1. \qquad (105)$$

Степінь породжувального многочлена — 10, тому і число контрольних бітів — 10. Отже, цей код є кодом (15, 5) БЧХ з виправленням потрійних помилок, і має такі характеристики:

- число кодових бітів $n = 15$;
- число контрольних бітів $m = 10$;
- число інформаційних бітів $k = 5$;
- здібність до виправлення помилок — виправлення потрійних помилок;
- породжувальний многочлен — $x^{10} + x^8 + x^{15} + x^4 + x^2 + x + 1$.

Структура інформаційних і контрольних бітів цього коду БЧХ, із якої видно їх послідовність, наведена в табл. 22. Питання, пов'язані з кодуванням і перевіркою здатності коду (15, 5) БЧХ до виправлення помилок, будуть розглянуті в наступному розділі.

Код (15, 5) БЧХ стосовно стану контролю містить три контрольні ряди:

1. Контрольний ряд для $x^4 + x + 1$ (ряд від α):

$$\alpha^{14}, \alpha^{13}, \alpha^{12}, \alpha^{11}, \alpha^{10}, \alpha^9, \alpha^8, \alpha^7, \alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, \alpha^0.$$

Таблиця 22. Структура інформаційних і контрольних бітів коду (15, 5) БЧХ

Інформаційні біти (5 бітів)					Контрольні біти (10 бітів)									
a_4	a_3	a_2	a_1	a_0	c_9	c_8	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0
1 біт	2 біт	3 біт	4 біт	5 біт	6 біт	7 біт	8 біт	9 біт	10 біт	11 біт	12 біт	13 біт	14 біт	15 біт

2. Контрольний ряд для $x^4 + x^3 + x^2 + x + 1$ (ряд від α^3):

$$(\alpha^{14})^3,(\alpha^{13})^3,(\alpha^{12})^3,(\alpha^{11})^3,(\alpha^{10})^3,(\alpha^{9})^3,(\alpha^{8})^3,(\alpha^{7})^3,(\alpha^{6})^3,(\alpha^{5})^3,(\alpha^{4})^3,(\alpha^{3})^3,(\alpha^{2})^3,(\alpha^{1})^3,(\alpha^{0})^3.$$

3. Контрольний ряд для $x^2 + x + 1$ (ряд від α^5):

$$(\alpha^{14})^5,(\alpha^{13})^5,(\alpha^{12})^5,(\alpha^{11})^5,(\alpha^{10})^5,(\alpha^{9})^5,(\alpha^{8})^5,(\alpha^{7})^5,$$
$$(\alpha^{6})^5,(\alpha^{5})^5,(\alpha^{4})^5,(\alpha^{3})^5,(\alpha^{2})^5,(\alpha^{1})^5,(\alpha^{0})^5.$$

За допомогою цих трьох рядів можна визначити стан контролю, тобто синдром помилки. Відповідність між синдромом, який побудовано на основі цих трьох рядів, і станами помилки наведена у табл. 23. Спостерігається взаємно однозначна відповідність між станом помилки (наприклад, помилка в п'ятнадцятому біті (c_0)) і бітами синдрому $S_2 = [(\alpha^0)^5(= \alpha^0)]$, $S_1 = [(\alpha^0)^3(= \alpha^0)]$ і $S_0 = \alpha^2$.

Згідно з табл. 23 матриця контролю парності запишеться так:

$$H = \begin{pmatrix} (\alpha^{14})^5,(\alpha^{13})^5,\ldots,(\alpha^{0})^5 \\ (\alpha^{14})^3,(\alpha^{13})^3,\ldots,(\alpha^{0})^3 \\ \alpha^{14},\ \alpha^{13},\ \ldots\ ,\ \alpha^{0} \end{pmatrix} =$$

$$= \begin{pmatrix} \alpha^{10},\alpha^{5},\alpha^{0},\alpha^{10},\alpha^{5},\alpha^{0},\alpha^{10},\alpha^{5},\alpha^{0},\alpha^{10},\alpha^{5},\alpha^{0},\alpha^{10},\alpha^{5},\alpha^{0} \\ \alpha^{12},\alpha^{9},\alpha^{6},\alpha^{3},\alpha^{0},\alpha^{12},\alpha^{9},\alpha^{6},\alpha^{3},\alpha^{0},\alpha^{12},\alpha^{9},\alpha^{6},\alpha^{3},\alpha^{0} \\ \alpha^{14},\alpha^{13},\alpha^{12},\alpha^{11},\alpha^{10},\alpha^{9},\alpha^{8},\alpha^{7},\alpha^{6},\alpha^{5},\alpha^{4},\alpha^{3},\alpha^{2},\alpha^{1},\alpha^{0} \end{pmatrix}. \qquad (106)$$

Таблиця 23. Відповідність між синдромом, побудованим на основі трьох рядів, і станами помилки

Синдром			Стан помилки
S_2	S_1	S_0	
0	0	0	Помилки відсутні
$(\alpha^{14})^5(=\alpha^{10})$	$(\alpha^{14})^3(=\alpha^{12})$	α^{14}	Помилка в першому біті (a_4)
$(\alpha^{13})^5(=\alpha^{5})$	$(\alpha^{13})^3(=\alpha^{9})$	α^{13}	Помилка в другому біті (a_3)
$(\alpha^{12})^5(=\alpha^{0})$	$(\alpha^{12})^3(=\alpha^{6})$	α^{12}	Помилка в третьому біті (a_2)
$(\alpha^{11})^5(=\alpha^{10})$	$(\alpha^{11})^3(=\alpha^{3})$	α^{11}	Помилка в четвертому біті (a_1)
$(\alpha^{10})^5(=\alpha^{5})$	$(\alpha^{10})^3(=\alpha^{0})$	α^{10}	Помилка в п'ятому біті (a_0)
$(\alpha^{9})^5(=\alpha^{0})$	$(\alpha^{9})^3(=\alpha^{12})$	α^{9}	Помилка в шостому біті (c_9)
$(\alpha^{8})^5(=\alpha^{10})$	$(\alpha^{8})^3(=\alpha^{9})$	α^{8}	Помилка в сьомому біті (c_8)
$(\alpha^{7})^5(=\alpha^{5})$	$(\alpha^{7})^3(=\alpha^{6})$	α^{7}	Помилка у восьмому біті (c_7)
$(\alpha^{6})^5(=\alpha^{0})$	$(\alpha^{6})^3(=\alpha^{3})$	α^{6}	Помилка в дев'ятому біті (c_6)
$(\alpha^{5})^5(=\alpha^{10})$	$(\alpha^{5})^3(=\alpha^{0})$	α^{5}	Помилка в десятому біті (c_5)
$(\alpha^{4})^5(=\alpha^{5})$	$(\alpha^{4})^3(=\alpha^{12})$	α^{4}	Помилка в одинадцятому біті (c_4)
$(\alpha^{3})^5(=\alpha^{0})$	$(\alpha^{3})^3(=\alpha^{9})$	α^{3}	Помилка в дванадцятому біті (c_3)
$(\alpha^{2})^5(=\alpha^{10})$	$(\alpha^{2})^3(=\alpha^{6})$	α^{2}	Помилка в тринадцятому біті (c_2)
$(\alpha^{1})^5(=\alpha^{5})$	$(\alpha^{1})^3(=\alpha^{3})$	α^{1}	Помилка в чотирнадцятому біті (c_1)
$(\alpha^{0})^5(=\alpha^{0})$	$(\alpha^{0})^3(=\alpha^{0})$	α^{0}	Помилка в п'ятнадцятому біті (c_0)

Код (15, 5) БЧХ припускає виправлення потрійних помилок, тому число помилкових бітів у послідовності

$$e = (e_{14}, e_{13}, e_{12}, e_{11}, e_{10}, e_9, e_8, e_7, e_6, e_5, e_4, e_3, e_2, e_1, e_0)$$

не перевищує трьох. Отже, серед бітів послідовності e_{14} ... e_0 число одиниць (помилок) не більше трьох. На приймальній стороні, за матрицею контролю парності H і помилковою послідовністю e, обчислюється синдром S. Матриця контролю парності (106) складається з трьох рядків, тому синдром S можна розкласти на три біти синдрому S_2, S_1, S_0, які відповідають різним рядкам матриці:

$$S = \begin{pmatrix} S_2 \\ S_1 \\ S_0 \end{pmatrix} = He^T = \begin{pmatrix} (\alpha^{14})^5, (\alpha^{13})^5, ..., (\alpha^0)^5 \\ (\alpha^{14})^3, (\alpha^{13})^3, ..., (\alpha^0)^3 \\ \alpha^{14}, \alpha^{13}, \alpha^{12}, ..., \alpha^0 \end{pmatrix} e^T. \qquad (107)$$

Згідно з формулою (107), відповідні біти синдрому помилки набувають такого вигляду:

$$S_2 = \left\{(\alpha^{14})^5, (\alpha^{13})^5, ..., (\alpha^0)^5\right\} e^T = (\alpha^{14})^5 \cdot e_{14} + (\alpha^{13})^5 \cdot e^{13} + ... + (\alpha^0)^5 \cdot e_0, \qquad (108)$$

$$S_1 = \left\{(\alpha^{14})^3, (\alpha^{13})^3, ..., (\alpha^0)^3\right\} e^T = (\alpha^{14})^3 \cdot e_{14} + (\alpha^{13})^3 \cdot e_{13} + ... + (\alpha^0)^3 \cdot e_0, \qquad (109)$$

$$S_0 = \left\{\alpha^{14}, \alpha^{13}, ...\alpha^0\right\} e^T = \alpha^{14} \cdot e_{14} + \alpha^{13} \cdot e_{13} + ... + \alpha^0 \cdot e_0. \qquad (110)$$

Якщо позначити елементи групи $GF(2^4)$, що відображають положення трьох помилкових бітів, як α^i, α^j, α^k, то рівняння розташування помилок запишеться так:

$$S(x) = (x - \alpha^i)(x - \alpha^j)(x - \alpha^k) = x^3 + Ax^2 + Bx + C. \qquad (111)$$

Виразивши коефіцієнти A, B, C цього рівняння через три біти синдрому S_2, S_1, S_0, аналогічно випадку, що описує положення подвійних помилок, можна знайти його корені, підставляючи елементи групи $GF(2^4)$. Далі можна виправити помилки шляхом інвертування бітів, що відповідають цим кореням.

Як приклад розглянемо випадок, коли помилка виникла в п'ятому, десятому і дванадцятому бітах. Подамо послідовність помилок так:

$$e = (000010000101000),$$

а саме: $e_{10} = 1$, $e_5 = 1$, $e_3 = 1$, решта бітів дорівнюють нулю. При цьому відповідні біти синдрому помилки мають вигляд:

$$S_2 = (\alpha^{10})^5 + (\alpha^5)^5 + (\alpha^3)^5, \qquad (112)$$

$$S_1 = (\alpha^{10})^3 + (\alpha^5)^3 + (\alpha^3)^3, \qquad (113)$$

$$S_0 = \alpha^{10} + \alpha^5 + \alpha^3. \qquad (114)$$

П'ятий, десятий і дванадцятий біти є помилковими, тому їм, згідно з табл. 19, відповідають наступні елементи групи $GF(2^4)$: п'ятому біту $\to\alpha^{10}$, десятому біту $\to\alpha^5$, дванадцятому біту $\to\alpha^3$. Отже, підставивши в рівняння розташування помилок (111) ці елементи, отримаємо вирази, що дорівнюють нулю:

$$S(\alpha^{10}) = (\alpha^{10})^3 + A(\alpha^{10})^2 + B(\alpha^{10}) + C = 0, \tag{115}$$

$$S(\alpha^5) = (\alpha^5)^3 + A(\alpha^5)^2 + B(\alpha^5) + C = 0, \tag{116}$$

$$S(\alpha^3) = (\alpha^3)^3 + A(\alpha^3)^2 + B(\alpha^3) + C = 0. \tag{117}$$

Встановивши відповідність між формулами (112)—(114) і формулами (115)—(117), можна визначити відповідність між бітами синдрому помилки S_2, S_1, S_0 і коефіцієнтами A, B, C. Оскільки

$$S(\alpha^{10}) + S(\alpha^5) + S(\alpha^3) = 0,$$

то і

$$\{(\alpha^{10})^3 + (\alpha^5)^3 + (\alpha^3)^3\} + A\{(\alpha^{10})^2 + (\alpha^5)^2 + (\alpha^3)^2\} + B\{(\alpha^{10}) + (\alpha^5) + (\alpha^3)\} + 3C = 0.$$

З урахуванням (113), (114) та співвідношення

$$(\alpha^{10})^2 + (\alpha^5)^2 + (\alpha^3)^2 = \{(\alpha^{10}) + (\alpha^5) + (\alpha^3)\}^2 = S_0{}^2, \tag{118}$$

отримаємо

$$S_1 + S_0^2 A + S_0 B + C = 0. \tag{119}$$

Аналогічно, оскільки

$$\alpha^{10} \cdot S(\alpha^{10}) + \alpha^5 \cdot S(\alpha^5) + \alpha^3 \cdot S(\alpha^3) = 0,$$

то і

$$\{(\alpha^{10})^4 + (\alpha^5)^4 + (\alpha^3)^4\} + A\{(\alpha^{10})^3 + (\alpha^5)^3 + (\alpha^3)^3\} +$$
$$+ B\{(\alpha^{10})^2 + (\alpha^5)^2 + (\alpha^3)^2\} + C\{(\alpha^{10}) + (\alpha^5) + (\alpha^3)\} = 0.$$

Виконавши перетворення з використанням рівнянь (113), (114), (118) і співвідношення

$$(\alpha^{10})^4 + (\alpha^5)^4 + (\alpha^3)^4 = \{(\alpha^{10})^2 + (\alpha^5)^2 + (\alpha^3)^2\}^2 = S_0{}^4, \tag{120}$$

одержимо

$$S_0^4 + S_1 A + S_0^2 B + S_0 C = 0. \tag{121}$$

На підставі того, що

$$(\alpha^{10})^2 \cdot S(\alpha^{10}) + (\alpha^5)^2 \cdot S(\alpha^5) + (\alpha^3)^2 \cdot S(\alpha^3) = 0,$$

маємо

$$\{(\alpha^{10})^5 + (\alpha^5)^5 + (\alpha^3)^5\} + A\{(\alpha^{10})^4 + (\alpha^5)^4 + (\alpha^3)^4\} +$$
$$+ B\{(\alpha^{10})^3 + (\alpha^5)^3 + (\alpha^3)^3\} + C\{(\alpha^{10})^2 + (\alpha^5)^2 + (\alpha^3)^2\} = 0.$$

Виконавши нескладні перетворення з урахуванням співвідношень (113), (114), (118) і (120), отримаємо

$$S_2 + S_0^4 A + S_1 B + S_0^2 C = 0. \tag{122}$$

Об'єднавши три рівняння з бітами синдрому в єдину систему, одержимо

$$\begin{cases} S_0^2 A + S_0 B + \quad C = S_1, \\ S_1 A + S_0^2 B + S_0 C = S_0^4, \\ S_0^4 A + S_1 B + S_0^2 C = S_2. \end{cases} \tag{123}$$

Якщо тепер визначити біти синдрому S_2, S_1, S_0, то можна знайти A, B, C, тобто розв'язати рівняння розташування помилок.

Спростимо вирази для бітів синдрому за допомогою співвідношень (80).

Для біта синдрому S_2 після перетворень маємо:

$$S_2 = (\alpha^{10})^5 + (\alpha^5)^5 + (\alpha^3)^5 = \alpha^{50} + \alpha^{25} + \alpha^{15} = \alpha^5 + \alpha^{10} + 1 =$$
$$= (\alpha^2 + \alpha)(\alpha^2 + \alpha + 1) + 1 = 2\alpha^2 + 2\alpha + 2 = 0.$$

Для біта синдрому S_1 отримаємо вираз

$$S_1 = (\alpha^{10})^3 + (\alpha^5)^3 + (\alpha^3)^3 = \alpha^{30} + \alpha^{15} + \alpha^9 = 1 + 1 + \alpha^9 = \alpha^9.$$

Тут для спрощення подальших обчислень використано α^9 в його безпосередньому вигляді.

Для біта синдрому S_0 одержимо

$$S_0 = \alpha^{10} + \alpha^5 + \alpha^3 =$$
$$= (\alpha^2 + \alpha + 1) + (\alpha^2 + \alpha) + \alpha^3 = \alpha^3 + 2\alpha^2 + 2\alpha + 1 = \alpha^3 + 1 = \alpha^{14},$$
$$S_0^2 = (\alpha^{14})^2 = \alpha^{28} = \alpha^{13}, \quad S_0^4 = (\alpha^{13})^2 = \alpha^{26} = \alpha^{11}.$$

Підставивши вирази для бітів синдромів S_2, S_1, S_0 у формулу (123), запишемо

$$\alpha^{13} \cdot A + \alpha^{14} \cdot B + \quad C = \alpha^9, \tag{124}$$

$$\alpha^9 \cdot A + \alpha^{13} \cdot B + \alpha^{14} \cdot C = \alpha^{11}, \tag{125}$$

$$\alpha^{11} \cdot A + \alpha^9 \cdot B + \alpha^{13} \cdot C = 0. \tag{126}$$

Розв'яжемо рівняння першого порядку, що задані формулами (124)—(126), скоротивши кількість змінних. Знову скористаємося співвідношеннями (80). З рівняння (126) випливає, що

$$B = (\alpha^{11} \cdot A + \alpha^{13} \cdot C) / \alpha^9 = \alpha^2 \cdot A + \alpha^4 \cdot C.$$

Якщо тепер підставимо цей вираз B у формулу (124), то

$$\alpha^{13} \cdot A + \alpha^{14}(\alpha^2 \cdot A + \alpha^4 \cdot C) + C = \alpha^9,$$

$$(\alpha^{13} + \alpha^{16}) \cdot A + (\alpha^{18} + 1) \cdot C = \alpha^9.$$

Оскільки

$$\alpha^{13} + \alpha^{16} = (\alpha^3 + \alpha^2 + 1) + \alpha = \alpha^{12},$$

$$\alpha^{18} + 1 = \alpha^3 + 1 = \alpha^{14},$$

то

$$\alpha^{12} \cdot A + \alpha^{14} \cdot C = \alpha^9.$$

Поділивши обидві сторони цієї рівності на α^9, отримаємо

$$\alpha^3 \cdot A + \alpha^5 \cdot C = 1. \qquad (127)$$

Підставивши вираз для B у формулу (125) і виконавши прості алгебраїчні перетворення, запишемо

$$\alpha^9 \cdot A + \alpha^{13}(\alpha^2 \cdot A + \alpha^4 \cdot C) + \alpha^{14} \cdot C = \alpha^{11},$$

$$(\alpha^9 + \alpha^{15}) \cdot A + (\alpha^{17} + \alpha^{14}) \cdot C = \alpha^{13}.$$

Оскільки

$$\alpha^9 + \alpha^{15} = (\alpha^3 + \alpha) + 1 = \alpha^7,$$

$$\alpha^{17} + \alpha^{14} = \alpha^2 + (\alpha^3 + 1) = \alpha^{18},$$

то

$$\alpha^7 \cdot A + \alpha^{13} \cdot C = \alpha^{11}.$$

Поділивши обидві сторони цієї рівності на α^7, маємо

$$A + \alpha^6 \cdot C = \alpha^4.$$

Звідки

$$A = \alpha^6 \cdot C + \alpha^4. \qquad (128)$$

Підставивши цей вираз для A у формулу (127), одержимо

$$\alpha^3(\alpha^6 \cdot C + \alpha^4) + \alpha^5 \cdot C = 1,$$

$$(\alpha^9 + \alpha^5) \cdot C = \alpha^7 + 1.$$

Оскільки

$$\alpha^9 + \alpha^5 = (\alpha^3 + \alpha) + (\alpha^2 + \alpha) = \alpha^6,$$

$$\alpha^7 + 1 = (\alpha^3 + \alpha + 1) + 1 = \alpha^9,$$

то

$$C = \alpha^3.$$

Далі легко визначити коефіцієнти A і B:

$$A = \alpha^6 \cdot \alpha^3 + \alpha^4 = \alpha^9 + \alpha^4 = (\alpha^3 + \alpha) + (\alpha + 1) = \alpha^3 + 1 = \alpha^{14},$$

$$B = \alpha^2 \cdot \alpha^{14} + \alpha^4 \cdot \alpha^3 = \alpha^{16} + \alpha^7 = \alpha + (\alpha^3 + \alpha + 1) = \alpha^3 + 1 = \alpha^{14}.$$

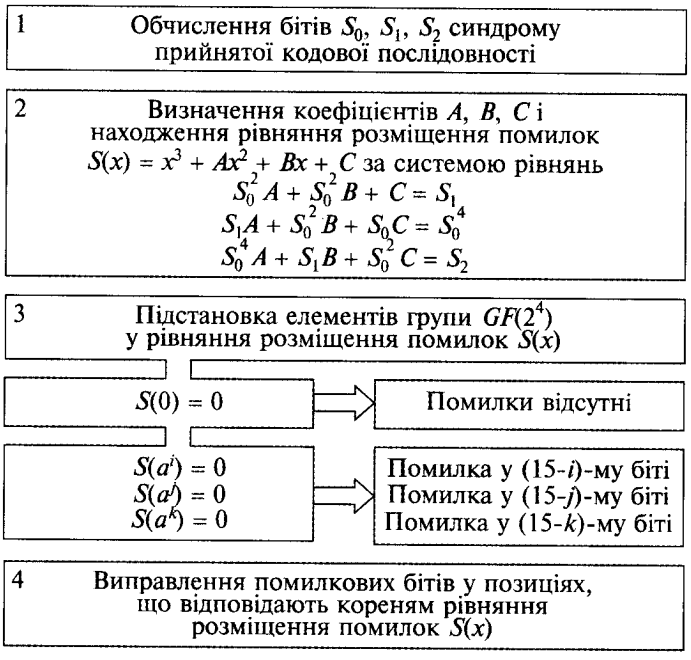


Рис. 61. Схема декодування коду (15, 5) БЧХ з виправленням потрійних помилок

Тоді рівняння розташування помилок матиме вигляд

$$S(x) = x^3 + \alpha^{14}x^2 + \alpha^{14}x + \alpha^3. \qquad (129)$$

Підставивши в рівняння розташування помилок (129) елементи групи $GF(2^4)$, отримаємо

$$S(\alpha^{10}) = (\alpha^{10})^3 + \alpha^{14}\cdot(\alpha^{10})^2 + \alpha^{14}\cdot\alpha^{10} + \alpha^3 = \alpha^{30} + \alpha^{34} + \alpha^{24} + \alpha^3 = 1 + \alpha^4 + \alpha^9 + \alpha^3 =$$

$$= 1 + (\alpha + 1) + (\alpha^3 + \alpha) + \alpha^3 = 0,$$

$$S(\alpha^5) = (\alpha^5)^3 + \alpha^{14}\cdot(\alpha^5)^2 + \alpha^{14}\cdot\alpha^5 + \alpha^3 = \alpha^{15} + \alpha^{24} + \alpha^{19} + \alpha^3 = 1 + \alpha^9 + \alpha^4 + \alpha^3 = 0,$$

$$S(\alpha^3) = (\alpha^3)^3 + \alpha^{14}\cdot(\alpha^3)^2 + \alpha^{14}\cdot\alpha^3 + \alpha^3 = \alpha^9 + \alpha^{20} + \alpha^{17} + \alpha^3 = \alpha^9 + \alpha^5 + \alpha^2 + \alpha^3 =$$

$$= (\alpha^3 + \alpha) + (\alpha^2 + \alpha) + \alpha^2 + \alpha^3 = 0.$$

Звідси видно, що елементи α^{10}, α^5, α^3 є коренями. Отже, виходячи з відповідності: елемент $\alpha^{10} \rightarrow$ п'ятий біт, елемент $\alpha^5 \rightarrow$ десятий біт, елемент $\alpha^3 \rightarrow$ дванадцятий біт, встановлюється положення помилкових бітів, після чого вони виправляються шляхом їх інвертування. Описаний процес декодування з виправленням потрійних помилок продемонстровано у вигляді схеми на рис. 61.

ОПЕРАЦІЯ РОЗБИТТЯ НА ГРУПИ ПОВНОЇ МНОЖИНИ ЦІЛИХ ЧИСЕЛ ТА МЕТОД ПОБУДОВИ СТАНДАРТНОЇ КОНФІГУРАЦІЇ ДЛЯ ЛІНІЙНОГО КОДУ

Розглянемо принципові міркування, на яких ґрунтуються методи декодування лінійних кодів. Суть цих методів — операції обчислення синдрому і підсумовування помилок, що відповідають синдрому, з прийнятою кодовою послідовністю. Які ж помилки можна виявити, а які не піддаються виявленню?

Якщо позначити число інформаційних бітів k, то обсяг інформації, який може бути поданий за допомогою цих k бітів, буде дорівнювати 2^k. Якщо позначити число перевірних бітів m, то число перевірних станів, які можуть бути записані за допомогою цих m бітів, буде становити 2^m. Крім того, інформаційна послідовність, що складається з k бітів, до якої приєднана перевірна послідовність, що складається з m бітів, є кодове слово довжиною $n = m + k$ (табл. 24).

Кодове слово містить у собі n бітів, тому кількість різних кодових послідовностей, до складу яких входять n бітів, дорівнює 2^n. Проте кодові слова будуються з інформаційних послідовностей на основі взаємно однозначної відповідності, тому число різних кодових слів збігається з числом різних інформаційних послідовностей, які містять у собі k бітів, і становить 2^k (рис. 62).

У математиці сукупність з n символів називається вектором. Тому цілком природно сукупність з n бітів, що складає кодове слово, називати ***кодовим вектором***. Як кодове слово, так і кодовий вектор містить у собі n бітів, проте, очевидно, що число комбінацій, яке можна отримати в тому чи іншому випадку, різне. А саме, число кодових слів = 2^k, число кодових векторів = 2^n, тобто число кодових слів пов'язано тільки з інформаційними бітами, а число кодових векторів — із загальною кількістю бітів кодової послідовності, що передається. Число бітів n у кодовому векторі є сумою інформаційних бітів k і перевірних бітів m, тому число різних комбінацій, яке може бути подано за допомогою кодового вектора, описується таким співвідношенням:

$$2^n = 2^{k+m} = 2^k \cdot 2^m, \tag{130}$$

а саме:

$$\textit{число кодових векторів} = (\textit{число кодових слів}) \cdot (\textit{число перевірних слів}). \tag{131}$$

Таблиця 24. Склад кодового слова

Кодове слово (n бітів)											
Інформаційна послідовність (k бітів)						Перевірна послідовність (m бітів)					
a_{k-1}	a_{k-2}	a_{k-3}	. . .	a_1	a_0	c_{m-1}	c_{m-2}	c_{m-3}	. . .	c_1	c_0

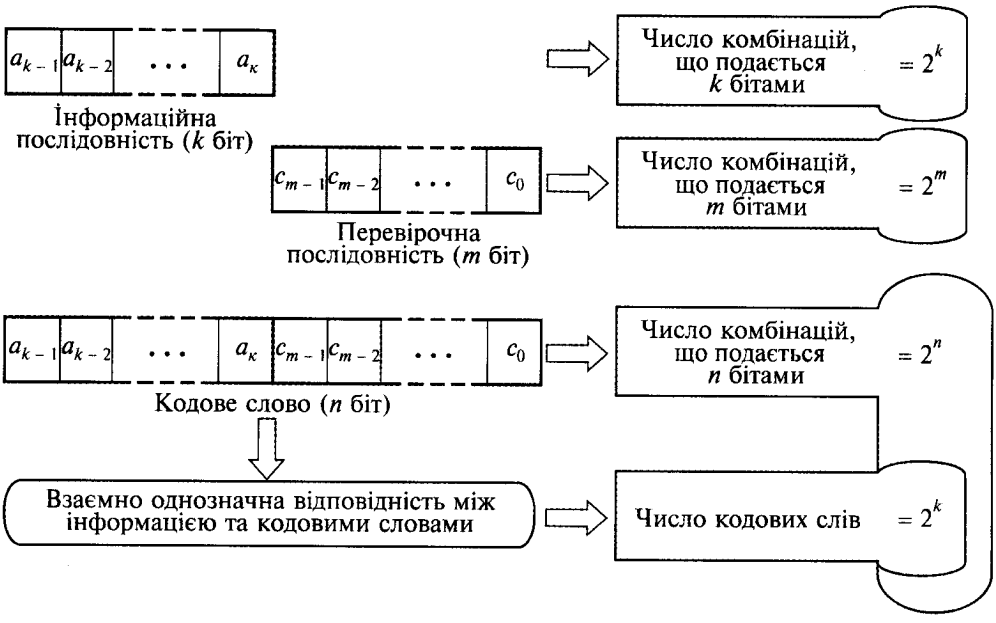


Рис. 62. Схема визначення числа бітових комбінацій, що подані параметрами лінійного коду (n, k)

Прийнята кодова послідовність є результатом підсумовування в каналі передачі даних кодового слова і послідовності помилок. Отже, серед n-бітових кодових векторів число кодових слів дорівнює 2^k. Проте кількість різних варіантів прийнятих кодових послідовностей не залежить від числа інформаційних комбінацій, оскільки до кодового слова додається помилкова послідовність. Таким чином, число різних варіантів прийнятих кодових послідовностей дорівнює 2^n. Тобто

число прийнятих кодових послідовностей $= 2^n =$ *число кодових векторів*.

Методи декодування лінійних кодів безпосередньо пов'язані з співвідношенням (131).

§15.1. РОЗБИТТЯ І ГРУПИ

Розглянемо операцію розбиття на конкретному прикладі розподілу учнів, що навчаються в школі, по класах у навчальному році. На початок кожного нового навчального року склад учнів у класах переглядається: визначається новий склад класів. Розглянувши принципи визначення складу класів, можна дійти до наступних висновків.

1. Усі учні, що навчаються в якому-небудь навчальному році в школі, обов'язково належать до якогось із класів.

2. Один і той же учень не може належати до двох і більше класів.

Ці дві умови є обов'язковими умовами розбиття. Якщо розбиття виконувати, додержуючись цих умов, то воно можливе. Отже, для розбиття необхідно мати *правила* цього розбиття.

Що стосується розподілу учнів по класах у навчальному році, то і зарахування учнів у той або інший клас здійснюється за деякими правилами. Підставами для розбиття по класах можуть бути, наприклад, прізвища (в алфавітному порядку), день, місяць і рік народження, успішність або навіть вибір за жеребкуванням тощо.

Якщо розглядати всіх учнів школи в певному навчальному році як одну множину, то можна визначити такі відповідності:

всі учні даного навчального року $\Rightarrow$ *множина (повна),*

учень $\Rightarrow$ *елемент,*

клас $\Rightarrow$ *підмножина.*

Далі розглянемо множини з математичної точки зору і спробуємо виконати розбиття.

Відмітним моментом математичного поняття розбиття від звичайного є застосування правил розбиття.

Як приклад розглянемо розбиття безлічі цілих чисел за модулем 16. Залишки від ділення за модулем 16 можуть приймати одне з наступних 16 значень: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15. При діленні будь-якого цілого числа на 16 отриманий залишок обов'язково прийме одне з шістнадцяти наведених вище значень. Розглянемо, як виконується розбиття цілих чисел за модулем 16.

Безліч цілих чисел, залишок від ділення яких на 16 *дорівнює* 0:

$\{0, \pm 16 + 0, \pm 32 + 0, \pm 48 + 0, \ldots\}$.

Безліч цілих чисел, залишок від ділення яких на 16 *дорівнює* 1:

$\{1, \pm 16 + 1, \pm 32 + 1, \pm 48 + 1, \ldots\}$.

Безліч цілих чисел, залишок від ділення яких на 16 *дорівнює* 2:

$\{2, \pm 16 + 2, \pm 32 + 2, \pm 48 + 2, \ldots\}$.

.....

Безліч цілих чисел, залишок від ділення яких на 16 *дорівнює* 14:

$\{14, \pm 16 + 14, \pm 32 + 14, \pm 48 + 14, \ldots\}$.

Безліч цілих чисел, залишок від ділення яких на 16 *дорівнює* 15:

$\{15, \pm 16 + 15, \pm 32 + 15, \pm 48 + 15, \ldots\}$.

Таким чином, всю множину цілих чисел можна розбити на шістнадцять підмножин за модулем 16 (рис. 63). У цьому випадку дві умови розбиття матимуть такий вигляд:

1. Будь-яке ціле число належить одній з підмножин залишків.

2. Одне і те саме ціле число не може належати двом і більше підмножинам залишків.

Очевидно, що обидві ці умови виконуються.

Друга умова розбиття показує, що після розбиття всіх цілих чисел на підмножини ці підмножини взаємно не перетинаються, тобто не мають спільних областей.

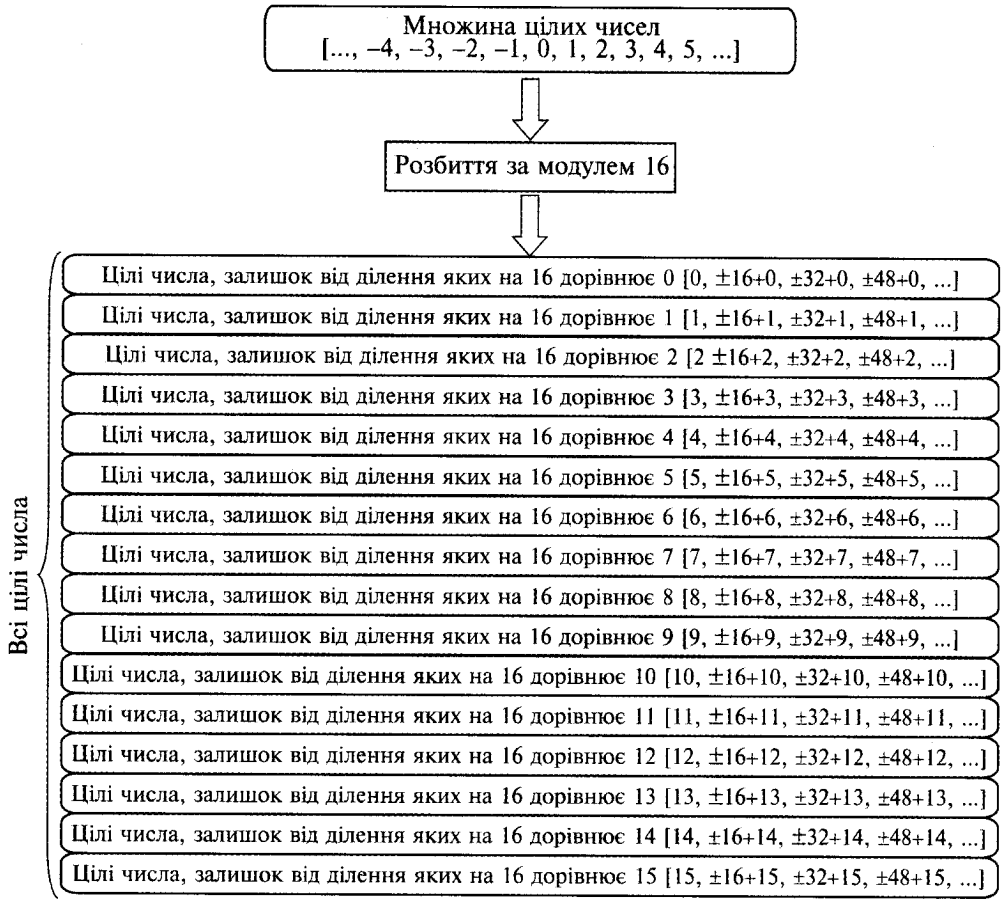


Рис. 63. Схема розбиття цілих чисел за модулем 16

Підмножини, побудовані на основі залишків, називаються угрупування за залишками.

Таким чином, виявляється, що всі цілі числа за модулем 16 можна розбити на шістнадцять угрупувань за залишками.

Далі розглянемо ще один метод розбиття повної множини цілих чисел, відмінний від методу розбиття за модулем 16. Цього разу спочатку розглянемо множину цілих чисел, що складається з чисел, кратних 16. За модулем 16 всі числа, які кратні 16, дорівнюють нулю. Виявляється, що множину, яка складається з чисел, кратних 16, можна подати таким чином: угрупування за залишками {0} = {0, ±16 + 0, ±32 + 0, ±48 + 0, ...} = {*цілі числа*, *кратні* 16} = {16 *n*} (*n* – ціле число). Множина цілих чисел, що містить у собі числа, кратні 16, тобто угрупування за залишками {0}, — це підмножина повної множини цілих чисел.

Розіб'ємо цілі числа на окремі підмножини, взявши за еталон цю підмножину повної множини цілих чисел {16 *n*}.

Вибравши з повної множини цілих чисел довільне число і додавши його до вказаної підмножини, знову можна отримати деяку підмножину повної множини цілих чисел. Далі можна побудувати ще одну, нову підмножину, вибравши ціле число, що не міститься в побудованій раніше підмножині, і додати його до підмножини, яка прийнята за еталон. Розбиття повної множини цілих чисел на підмножини можна здійснити, повторюючи цю операцію доти, поки не отримаємо висновок, що будь-яке ціле число уже належить якійсь із побудованих підмножин.

Спочатку як довільне ціле число виберемо 0 і додамо його до підмножини повної множини цілих чисел {16 *n*}.

{16 *n*} + 0 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16} = *угрупування за залишками* {0}.

Далі виберемо число, що не належить знову створеній підмножині повної множини цілих чисел (в угрупуванні за залишками {0}), наприклад 1, і додамо його до підмножини повної множини цілих чисел, що утворюється з чисел, кратних 16.

{16 *n*} + 1 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16 *плюс* 1} = *угрупування за залишками* {1}.

А тепер виберемо ціле число, що не міститься в двох визначених раніше підмножинах (угрупуваннях за залишками {0} і {1}), наприклад 2, і додамо його до підмножини, прийнятої за еталон:

{16 *n*} + 2 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16 *плюс* 2} = *угрупування за залишками* {2}.

Виконуючи цю операцію далі, отримуємо

{16 *n*} + 3 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16 *плюс* 3} = *угрупування за залишками* {3}.

{16 *n*} + 4 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16 *плюс* 4} = *угрупування за залишками* {4}.

. . . . . .

{16 *n*} + 14 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16 *плюс* 14} = *угрупування за залишками* {14}.

{16 *n*} + 15 = {*підмножина повної множини цілих чисел,*
що утворюється із чисел, кратних 16 *плюс* 15} = *угрупування за залишками* {15}.

Неважко виявити, що кожне ціле число належить до однієї із знову побудованих підмножин. Тому вся сукупність побудованих таким чином підмножин, тобто вся сукупність угрупувань за залишками, еквівалентна повній множині цілих чисел. Тут дуже важливу роль відіграє те, яким чином виконувалося це розбиття.

Вище повна множина цілих чисел розбита за модулем 16 і побудовано шістнадцять угрупувань за залишками. Проте повна множина цілих чисел є нескінченною множиною, тому виявляється, що шістнадцять підмножин, побудованих на основі цієї множини, знову-таки є нескінченними множинами. Очевидно, що цією концепцією нескінченності надзвичайно важко оперувати,

якщо мислити в поняттях повсякденного обмеженого світу. Справа у тому, що майже всі множини, з якими ми щодня маємо справу, наприклад розподіл учнів по класах протягом навчального року, — скінченні множини.

Вибравши з шістнадцяти угрупувань за залишками, взятими за модулем 16 від всіх цілих чисел:

{0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {13}, {14}, {15},

по одному залишку з кожного угрупування, побудуємо множину, що складається з цих елементів:

$$R(16) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}.$$

Число елементів множини $R(16)$ дорівнює шістнадцяти, тому вона є скінченною. Як і в повній множині цілих чисел можна вільно, без будь-яких обмежень, виконувати додавання, так само можна додавати один з одним елементи множини $R(16)$, використовуючи операцію додавання, визначену як додавання за модулем 16. Наприклад,

$$10 + 14 \equiv 8 \pmod{16},\ 8 + 9 \equiv 1 \pmod{16},\ \ldots$$

Додаючи між собою аналогічним чином різні пари елементів множини $R(16)$, отримуємо правило додавання (табл. 25).

Ретельно вивчивши цю таблицю додавання, можна встановити, що множина $R(16)$ має наступні властивості:

1. Якщо додати між собою за модулем 16 два елементи множини $R(16)$, то отриманий результат буде одним із елементів, що належать цій самій множині $R(16)$. А саме, результат додавання за mod 16 над елементами множини $R(16)$ неодмінно належить множині $R(16)$. Математично це виражається так: множина

Таблиця 25. Правило додавання множини $R(16)$, побудованої із залишків за mod 16

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1
3	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2
4	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3
5	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4
6	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5
7	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8
10	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9
11	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10
12	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11
13	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12
14	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13
15	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

$R(16)$ замкнена щодо додавання за mod 16. Бути замкненим щодо додавання за mod 16 означає, що операція додавання за mod 16 між елементами множини $R(16)$ може здійснюватися вільно, без будь-яких обмежень. Тобто якщо припустити, що a і b є елементами множини $R(16)$, то і $a + b \pmod{16}$ є елементом цієї множини. Належність до множини в математиці прийнято позначати значком $\in$. Таким чином, можна записати

$$\text{якщо } a, b \in R(16), \text{ то } a + b \in R(16) \pmod{16}.$$

2. Якщо з множини $R(16)$ довільним чином вибрати три елементи a, b і c та знайти суму елементів a і b ($a + b \pmod{16}$), а потім до неї додати елемент c, або знайти суму b і c ($b + c \pmod{16}$) та додати a, то отримаємо один і той самий результат. Наприклад,

$$(13 + 2) + 4 \equiv 13 + (2 + 4) \pmod{16}.$$

Так, завжди виконується співвідношення

$$(a + b) + c \equiv a + (b + c) \pmod{16}.$$

У таких випадках говорять, що виконується ***асоціативний*** закон.

3. Зазначимо, що при додаванні до елемента 0 будь-якого елемента множини $R(16)$ одержимо той самий складовий елемент. Наприклад,

$$0 + 13 \equiv 13 \pmod{16},\ 8 + 0 \equiv 8 \pmod{16},\ \ldots$$

А саме: якщо позначити довільний елемент множини $R(16)$ як a, то завжди виконується співвідношення

$$a + 0 \equiv 0 + a \pmod{16}.$$

Цей елемент 0 називається одиничним елементом щодо адитивного закону, або просто — нульовий елемент.

4. Для будь-якого довільно взятого елемента a множини $R(16)$ обов'язково існує такий елемент x, при додаванні якого (елемента x) до початкового елемента a отримуємо нульовий елемент (одиничний елемент щодо адитивного закону). А саме:

$$a + x \equiv 0 \pmod{16}.$$

Цей елемент x називається протилежним до елемента a, тобто

$$15 + 1 \equiv 0 \pmod{16},\ 14 + 2 \equiv 0 \pmod{16},\ 13 + 3 \equiv 0 \pmod{16}.$$

Тому число 1 є елементом, протилежним до 15, число 2 — до 14, а число 3 — до 13 і т. д.

5. Із табл. 25 видно, що при додаванні двох елементів множини $R(16)$ отримуємо однаковий результат незалежно від того, в якій послідовності виконується операція додавання. Наприклад,

$$13 + 8 \equiv 8 + 13 \pmod{16},\ 2 + 3 \equiv 3 + 2 \pmod{16},\ \ldots$$

А саме: якщо позначити два довільно взятих елементи множини $R(16)$ як a і b, то виконуватиметься співвідношення

$$a + b \equiv b + a \pmod{16}.$$

У такому разі говорять, що виконується ***комутативний*** закон.

Отже, постульована операція додавання за mod 16 елементів множини $R(16)$ має п'ять властивостей. У математиці всі множини, яким притаманні чотири наведені властивості, називаються ***групами***. В описаному прикладі для додавання елементів запропоновано додавання за mod 16, тому для точності можна відзначити, що ця група є адитивною. Крім того, коли додатково до властивостей 1—4 група має ще й п'яту властивість, то вона називається ***комутативною групою***.

Виходячи з таких міркувань, можна сформулювати визначення групи.

Визначення групи. Якщо існує множина G і між двома довільно взятими елементами, що належать цій множині G, задано операцію додавання (двомісна операція додавання), що має наведені нижче властивості, то ця множина G називається ***групою***.

• ***Властивість G1***: для будь-яких двох довільно взятих елементів a і b множини G має місце

$$a + b \in G,$$

тобто множина G замкнена щодо операції додавання.

• ***Властивість G2***: для будь-яких трьох довільно взятих елементів a, b і c множини G виконується співвідношення

$$(a + b) + c = a + (b + c),$$

тобто виконується асоціативний закон.

• ***Властивість G3***: для будь-якого довільно взятого елемента a множини G існує такий елемент e, що виконується співвідношення

$$a + e = e + a = a.$$

Елемент e — одиничний елемент.

• ***Властивість G4***: для будь-якого довільно взятого елемента a множини G існує такий елемент x, що виконується співвідношення

$$a + x = x + a = e.$$

Елемент x — протилежний до елемента a.

У наведеному визначенні групи як операція між двома елементами (в двомісній операції) використовувалося додавання (+). У такому разі G — адитивна група. Сенс наведеної в даному контексті операції додавання (+) є цілком зрозумілим, і не важливо, що ця операція не збігається з операцією над дійсними числами. Що ж до операції додавання, визначеної як операція між двома елементами множини G, то з викладеного вище випливає, що вона повинна бути підпорядкована властивостям $G1$—$G4$.

Як двомісну операцію можна також задати операцію множення (×). Група, в якій як двомісну операцію постульовано множення (×), називається ***мультиплікативною групою***. І у цьому випадку немає потреби, щоб така операція множення збігалася із звичайною операцією множення дійсних чисел.

Коли груповою операцією визначено додавання (+), елемент e (одиничний елемент), який задано за властивостями групи $G3$, називається нульовим, і, якщо не виникає різночитання, він може записуватися як "0". А у випадку, коли груповою операцією визначено множення (×), елемент називається одиничним, і якщо не виникає різночитання, він може записуватися як "1". Таким чином,

$$a + 0 = 0 + a = a \text{ (адитивна група)},$$

$$a \times 1 = 1 \times a = a \text{ (мультиплікативна група)}.$$

Елемент x, протилежний до a, який задається за властивостями групи $G4$, у разі адитивної групи може бути поданий як $-a$, а у разі мультиплікативної групи — як a^{-1}. Тобто

$$a + (-a) = (-a) + a = 0 \text{ (адитивна група)},$$

$$a \times a^{-1} = a^{-1} \times a = 1 \text{ (мультиплікативна група)}.$$

Таким чином, виходячи з того, що в групі для кожного елемента існує протилежний, очевидною є справедливість наступних тверджень:

- *в адитивній групі можливі операції додавання та віднімання;*
- *у мультиплікативній групі можливі операції множення та ділення.*

Отже, залишилася нерозглянутою ще одна властивість.

- ***Властивість AG5***: якщо для будь-яких двох довільно взятих елементів a і b групи G виконуються наступні співвідношення:

$$a + b = b + a \text{ (в адитивній групі)},$$

$$a \times b = b \times a \text{ (у мультиплікативній групі)},$$

то група називається ***комутативною***.

Ця властивість $AG5$ завжди виконується в адитивних групах, але цілком можливо, що в мультиплікативній групі вона виконуватися не буде.

Отже, безліч залишків від ділення цілих чисел за mod 16 щодо операції додавання за mod 16 є групою, точніше кажучи, адитивною групою. У такому разі з'ясуємо стан справ щодо повної множини цілих чисел. Проаналізуємо це питання, прийнявши додавання як операцію з цілими числами.

- Властивість $G1$: якщо додати два будь-які довільно взяті цілі числа m і n, то отримаємо, що $m + n$ належить ($\in$) множині цілих чисел. Таким чином, ця властивість виконується.
- Властивість $G2$: для будь-яких трьох довільно взятих цілих чисел l, m і n завжди виконується співвідношення $(l + m) + n = l + (m + n)$.
- Властивість $G3$: множина цілих чисел містить у собі число 0, і для будь-якого довільно взятого цілого числа n має місце співвідношення: $n + 0 = 0 + n = n$, тобто 0 є нульовим елементом (одиничним елементом щодо додавання).
- Властивість $G4$: для будь-якого довільно взятого елемента n завжди існує протилежний елемент $(-n)$, такий, що виконується рівність $n + (-n) = 0$.

Отже, повна множина цілих чисел є групою. У такому разі виникає питання, якою є множина цілих чисел, кратних 16. Повна множина цілих чисел задовольняє чотири групові властивості, тому, очевидно, що навіть, якщо при збереженні колишньої операції додавання виконати заміну:

$$\textit{ціле число} \rightarrow 16 \times \textit{ ціле число},$$

то знову будуть виконуватися властивості $G1$ – $G4$. Таким чином, множина цілих чисел, кратних 16, також є групою. Множина цілих чисел, кратних 16, — це підмножина повної множини цілих чисел. Тому, якщо підмножина множини, що є групою, також має властивості, характерні для групи, то ця підмножина називається ***підгрупою***. Отже,

множина цілих чисел → *група* G,

множина цілих чисел, кратних 16 → *підгрупа* SG.

Той факт, що повна множина цілих чисел може бути розбита на підмножини, які утворюються з цілих чисел, кратних 16, свідчить, що група G може бути розбита на підгрупи SG.

Множина $R(16)$ є групою щодо додавання за модулем 16. Спробуємо розбити множину $R(16)$ на підгрупи. Перш за все необхідно знайти відповідні підгрупи цієї множини:

$$R(16) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}.$$

Питання, чи є множина $R(16)$ групою, було з'ясовано раніше (див. табл. 25) шляхом додавання за mod 16, що охоплює всі можливі пари елементів множини $R(16)$. Таким чином, очевидно, що не перевіряючи, чи є підмножина множини $R(16)$ групою, можна, проаналізувавши виконання групових властивостей $G1—G4$, відразу визначити таку групу, побудувавши таблицю додавання за mod 16, що охоплює всі елементи даної підмножини.

Спробуємо побудувати таку таблицю додавання для підмножини $S = \{0, 1, 2, 3, 4, 5\}$, що складається з шести елементів, 0, 1, 2, 3, 4, 5, які взяті з елементів множини. Результати додавання для всіх можливих пар елементів цієї підмножини S наведено в табл. 26.

Унаслідок додавання деяких пар елементів з підмножини S отримаємо елементи, що не містяться у початковій підмножині S. Це означає, що умова замкненості щодо заданої операції не виконується, тому дана підмножина S не є групою.

Можливо, такий метод, при якому елементи вибирають будь-як, утворюють з них підмножину, потім будують таблицю додавання для цієї підмножини і визначають, чи є вона групою, в принципі і можна застосовувати для пошуку підгруп. Проте його навряд чи можна назвати інтелектуальним. У разі повної множини цілих чисел виявилось, що підгрупою є множина цілих чисел,

Таблиця 26. Результати додавання за mod 16 для підмножини S

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	6
2	2	3	4	5	6	7
3	3	4	5	6	7	8
4	4	5	6	7	8	9
5	5	6	7	8	9	10

кратних 16. Спробуємо і в цьому випадку аналогічним чином побудувати підмножину з елементів, пов'язаних між собою співвідношеннями кратності. Зазначимо, що до складу кожної з цих підмножин обов'язково вводитиметься 0, оскільки в групі неодмінно повинен бути одиничний елемент.

Перш за все спробуємо побудувати підмножину S_2, вибравши з множини $R(16)$ елементи, які кратні двом: $S_2 = \{0, 2, 4, 6, 8, 10, 12, 14\}$. Результати додавання для всіх можливих пар елементів підмножини S_2 наведено в табл. 27.

Як бачимо з табл. 27, результати додавання всіх можливих пар елементів підмножини S_2 належать цій самій підмножині S_2. Тобто підмножина S_2 замкнена щодо тієї самої операції додавання, яка задана на множині $R(16)$, тому ця підмножина є групою і водночас підгрупою множини $R(16)$.

Далі, спробуємо побудувати підмножину, вибравши елементи, що кратні трьом: $S_3 = \{0, 3, 6, 9, 12, 15\}$. Результати додавання для всіх можливих пар елементів підмножини S_3 подано в табл. 28.

Як видно з табл. 28, підмножина S_3 не є групою, оскільки в ній з'являються нові елементи, які відсутні в початковій підмножині.

А тепер спробуємо побудувати підмножину, вибравши з множини $R(16)$ елементи, що кратні чотирьом: $S_4 = \{0, 4, 8, 12\}$. Результати додавання для всіх можливих пар елементів підмножини S_4 наведено в табл. 29.

З табл. 29 випливає, що результати додавання всіх можливих пар елементів підмножини S_4 належать цій самій підмножині S_4. Отже, підмножина S_4 є групою і водночас підгрупою множини $R(16)$.

Таблиця 27. Результати додавання за mod 16 для підмножини S_2

+	0	2	4	6	8	10	12	14
0	0	2	4	6	8	10	12	14
2	2	4	6	8	10	12	14	0
4	4	6	8	10	12	14	0	2
6	6	8	10	12	14	0	2	4
8	8	10	12	14	0	2	4	6
10	10	12	14	0	2	4	6	8
12	12	14	0	2	4	6	8	10
14	14	0	2	4	6	8	10	12

Таблиця 28. Результати додавання за mod 16 для підмножини S_3

+	0	3	6	9	12	15
0	0	3	6	9	12	15
3	3	6	9	12	15	2
6	6	9	12	15	2	5
9	9	12	15	2	5	8
12	12	15	2	5	8	11
15	15	2	5	8	11	14

Таблиця 29. Результати додавання за mod 16 для підмножини S_4

+	0	4	8	12
0	0	4	8	12
4	4	8	12	0
8	8	12	0	4
12	12	0	4	8

Таким чином, навіть при побудові підмножин шляхом вибору елементів, пов'язаних між собою співвідношеннями кратності, виникають підмножини, що є підгрупами і не є ними. Виявилось, що підмножини, які утворюються з елементів, кратних двом і чотирьом, є групами. Загальне число елементів множини $R(16)$ дорівнює шістнадцяти, причому як два, так і чотири є дільниками шістнадцяти. Аналогічно, число елементів множини S_2, що складається з елементів, кратних двом, дорівнює восьми, а елементів множини S_4, що утворюється з елементів, кратних чотирьом, — чотирьом. Тому ці числа є дільниками шістнадцяти, тобто всіх чисел, що входять до складу загального числа елементів множини $R(16)$. Таким чином, з'ясувалося, що число елементів підмножини, які є підгрупою, — це дільник числа елементів початкової групи.

Дільниками числа елементів множини $R(16)$, що дорівнює шістнадцяти, є вісім, чотири і два. Зрозуміло, що підмножина S_2 з числом елементів, що дорівнює восьми, і підмножина S_4, з числом елементів, що дорівнює чотирьом, є групами. Залишається з'ясувати, чи є групою підмножина з числом елементів, що дорівнює двом. Дослідимо підмножину S_8, що складається з елементів, кратних восьми:

$$S_8 = \{0, 8\}.$$

Результати аналізу додавання для всіх можливих пар елементів підмножини S_8 наведено в табл. 30. Як і передбачалося, підмножина S_8 є групою. Отже, підмножина S_8 є підгрупою множини $R(16)$.

Загальне число підгруп множини $R(16)$ дорівнює трьом:

$$S_2 = \{0, 2, 4, 6, 8, 10, 12, 14\},\ S_4 = \{0, 4, 8, 12\},\ S_8 = \{0, 8\}.$$

Той факт, що інші підгрупи, крім вказаних, відсутні, з'ясовується на підставі того, що число елементів множини, яке дорівнює шістнадцяти, не має інших дільників, крім трьох: 8, 4 і 2.

А тепер спробуємо розбити групу $R(16)$, використовуючи підгрупи S_4. Перш за все, побудуємо клас залишків, вибравши з множини $R(16)$ один елемент, наприклад 0. Додамо його до елементів підгрупи S_4 (застосувавши операцію додавання, визначену в групі):

$S_4 + 0$	0	4	8	12

Таблиця 30. Результати додавання за mod 16 для підмножини S_8

+	0	0
0	0	8
8	8	0

Потім побудуємо клас залишків $S_4 + 1$, вибравши елемент множини $R(16)$, що не міститься в $S_4 + 0$, наприклад 1:

$S_4 + 0$	0	4	8	12
$S_4 + 1$	1	5	9	13

Аналогічно побудуємо клас залишків, вибравши елемент множини $R(16)$, що не міститься в класах залишків $S_4 + 0$ та $S_4 + 1$, наприклад 2:

$S_4 + 0$	0	4	8	12
$S_4 + 1$	1	5	9	13
$S_4 + 2$	2	6	10	14

Залишилися чотири елементи множини $R(16)$, які не потрапили в цю таблицю. Виберемо з них, наприклад, 3 і побудуємо клас залишків $S_4 + 3$:

$S_4 + 0$	0	4	8	12
$S_4 + 1$	1	5	9	13
$S_4 + 2$	2	6	10	14
$S_4 + 3$	3	7	11	15

До останньої таблиці увійшли всі елементи множини $R(16)$. Отже, вдалося розбити групу $R(16)$ з використанням підгрупи S_4 і задовольнити умови розбиття.

Відповідно до викладеного вище, метод розбиття групи G за її підгрупою SG можна подати у вигляді табл. 31.

Нехай група G має наступний вигляд: $G = \{g_0, g_1, g_2, ..., g_i, g_{n-1}\}$, а підгрупа SG групи G — $SG = \{e, a_1, a_2, ..., a_{m-1}\}$ (e — одиничний елемент). Порядок розбиття групи G на підгрупи наступний.

1. Вибравши з групи G довільний елемент g_0 і додавши його до елементів підгрупи SG (за допомогою операції додавання, заданої в групі G), будуємо клас залишків $SG + g_0$.

2. Вибравши довільний елемент g_1 групи G, що не міститься в класі залишків $SG + g_0$, і додавши його до елементів підгрупи SG, будуємо клас залишків $SG + g_1$.

3. Вибравши довільний елемент g_2 групи G, що не міститься в класах залишків $SG + g_0$ та $SG + g_1$, і додавши його до елементів підгрупи SG, будуємо клас залишків $SG + g_2$.

4. Аналогічна процедура повторюється доти, поки в класи залишків не будуть включені всі елементи групи G.

Елементи $g_0, g_1, g_2, ..., g_{l-1}$, вибрані з метою побудови класу залишків, називаються ***представницькими елементами*** (або ядром класу залишків). Що ж до методу вибору представницького елемента, то не має значення, який елемент буде узятий на тому або іншому етапі, лише б він був елементом групи G. У табл. 32—34 наведено розбиття групи $R(16)$, яке виконано з використанням підгрупи S_4.

Таблиця 31. Розбиття групи G за підгрупою SG

Клас залишків	Представницький елемент	Елементи підгрупи SG				
		e	a_1	a_2		a_{m-1}
$SG + g_0$	g_0	$e + g_0$	$a_1 + g_0$	$a_2 + g_0$	·	$a_{m-1} + g_0$
$SG + g_1$	g_1	$e + g_1$	$a_1 + g_1$	$a_2 + g_1$	·	$a_{m-1} + g_1$
$SG + g_2$	g_2	$e + g_2$	$a_1 + g_2$	$a_2 + g_2$	·	$a_{m-1} + g_2$
	·	·	·	·	·	·
	·	·	·	·	·	·
	·	·	·	·	·	·
$SG + g_{l-1}$	g_{l-1}	$e + g_{l-1}$	$a_1 + g_{l-1}$	$a_2 + g_{l-1}$	·	$a_{m-1} + g_{l-1}$

Таблиця 32. Результати першого розбиття

Клас залишків	Представницький елемент	$S_4(SG)$			
		0	4	8	12
$SG + 0$	0	0	4	8	12
$SG + 1$	1	1	5	9	13
$SG + 2$	2	2	6	10	14
$SG + 3$	3	3	7	11	15

Таблиця 33. Результати другого розбиття

Клас залишків	Представницький елемент	$S_4(SG)$			
		0	4	8	12
$SG + 3$	3	3	7	11	15
$SG + 6$	6	6	10	14	2
$SG + 9$	9	9	13	1	5
$SG + 12$	12	12	0	4	8

Таблиця 34. Результати третього розбиття

Клас залишків	Представницький елемент	$S_4(SG)$			
		0	4	8	12
$SG + 0$	0	0	4	8	12
$SG + 5$	5	5	9	13	1
$SG + 10$	10	10	14	2	6
$SG + 15$	15	15	3	7	11

Таблиця 35. Розбиття групи $R(16)$ за підгрупою S_2

Клас залишків	Представницький елемент	$S_2(SG)$							
		0	2	4	6	8	10	12	14
S_2+0.....	0	0	2	4	6	8	10	12	14
S_2+1.....	1	1	3	5	7	9	11	13	15

Таблиця 36. Розбиття групи $R(16)$ за підгрупою S_8

Клас залишків	Представницький елемент	$S_8(SG)$	
		0	8
S_8+0.....	0	0	8
S_8+1.....	1	1	9
S_8+2.....	2	2	10
S_8+3.....	3	3	11
S_8+4.....	4	4	12
S_8+5.....	5	5	13
S_8+6.....	6	6	14
S_8+7.....	7	7	15

Отже, як представницькі елементи можна вибирати найприйнятніші, враховуючи при цьому характерні особливості даної множини.

У табл. 35 та 36 наведено розбиття групи $R(16)$, яке виконано з використанням підгруп S_2 і S_8.

Як представницькі елементи вибрані елементи з мінімальними числовими значеннями. Між числом елементів у групі G, числом елементів у підгрупі SG і числом класів залишків існує залежність:

$$\textit{число елементів у групі } G = \\ = (\textit{число елементів у підгрупі } SG) \times (\textit{число класів залишків}). \qquad (132)$$

На її підставі при виборі підгрупи, на основі якої виконуватиметься розбиття групи, як критерій вибору можна використовувати дільники числа елементів групи G.

§15.2. СТАНДАРТНА КОНФІГУРАЦІЯ ДЛЯ ЛІНІЙНОГО КОДУ

Як уже зазначалося, залежність між числом кодових векторів і числом кодових слів, а також числом станів перевірки може бути описана за допомогою співвідношення (131). Проведемо зіставлення цієї залежності з залежністю між числом елементів у групі G і числом елементів у підгрупі SG, а також числом класів залишків, поданих з використанням формули (132) і рис. 64.

При аналізі рис. 64 асоціативно виникає думка щодо таких відповідностей:

число кодових векторів ⇔ число елементів групи G,

число кодових слів ⇔ число елементів підгрупи SG,

число станів перевірки ⇔ число класів залишків.

Очевидно, має сенс, розглядаючи кодовий вектор, тобто сукупність бітів довжиною n, що утворюють лінійний код, вважати його групою і розбивати за допомогою кодового слова, що є його підгрупою. Стани перевірки лінійного коду тісно пов'язані з картинами помилок, а вони, у свою чергу, можуть бути

Рис. 64. Відповідність між лінійними кодами і групами

поставлені у відповідність представницьким елементам, які відіграють таку важливу роль при розбитті групи G за допомогою підгрупи SG:

картина помилок ⇔ представницький елемент.

Запишемо векторне подання кодового вектора довжиною n бітів:

$$[b_{n-1}, b_{n-2}, b_{n-3}, \ldots, b_1, b_0], \; b_i \in \{0,1\}.$$

Тут кожний з елементів b_i є елементом кінцевого поля $GF(2)$.

Перш за все, виникає питання, чи є групою множина, подана кодовим вектором, який є сукупністю з n елементів кінцевого поля $GF(2)$.

Наприклад, розглянемо множину, що є трибітовим кодовим вектором:

$$[b_2, b_1, b_0], \; b_i \in \{0,1\}.$$

За операцію над цим кодовим вектором приймемо операцію додавання, що застосовувалася при підсумовуванні кодового слова і картини помилок, а саме:

$$[b_2, b_1, b_0] + [d_2, d_1, d_0] = [f_2, f_1, f_0], \qquad (133)$$

де $f_2 = b_2 + d_2 \pmod 2$, $f_1 = b_1 + d_1 \pmod 2$, $f_0 = b_0 + d_0 \pmod 2$.

У цілому безліч трибітових кодових векторів G містить всього $2^3 = 8$ елементів:

$$G = \{[000], [001], [010], [011], [100], [101], [110], [111]\}.$$

З'ясувати, чи є ця множина G групою, можна, побудувавши таблицю додавання, на основі операції додавання, визначеної формулою (133), що охоплює всі пари елементів множини (табл. 37).

З цієї таблиці видно, що результати додавання по всіх парах є елементами множини G. Отже, можна зробити висновок, що множина G — група.

Крім того, з табл. 37 випливає, що множина $[b_{n-1}, b_{n-2}, b_{n-3}, \ldots, b_1, b_0]$, $b_i \in \{0,1\}$, що являє собою кодовий вектор, який є сукупністю з n елементів кінцевого поля $GF(2)$, також — група. Операцію додавання, що застосовується у цьому випадку, можна визначити так:

$$[b_{n-1}, b_{n-2}, b_{n-3}, \ldots, b_1, b_0] + [d_{n-1}, d_{n-2}, d_{n-3}, \ldots, d_1, d_0] = [f_{n-1}, f_{n-2}, f_{n-3}, \ldots, f_1, f_0], \qquad (134)$$

де $f_{n-1} = b_{n-1} + d_{n-1} \pmod 2$, $f_{n-2} = b_{n-2} + d_{n-2} \pmod 2$, ..., $f_0 = b_0 + d_0 \pmod 2$.

Таблиця 37. Результати додавання для множини *G*

+	[000]	[001]	[010]	[100]	[011]	[101]	[110]	[111]
[000]	[000]	[001]	[010]	[100]	[011]	[101]	[110]	[111]
[001]	[001]	[000]	[011]	[101]	[010]	[100]	[111]	[110]
[010]	[010]	[011]	[000]	[110]	[001]	[111]	[100]	[101]
[100]	[100]	[101]	[110]	[000]	[111]	[001]	[010]	[011]
[011]	[011]	[010]	[001]	[111]	[000]	[110]	[101]	[100]
[101]	[101]	[100]	[111]	[001]	[110]	[000]	[011]	[010]
[110]	[110]	[111]	[100]	[010]	[101]	[011]	[000]	[001]
[111]	[111]	[110]	[101]	[011]	[100]	[010]	[001]	[000]

Оскільки, множина G — група, то спробуємо визначити підгрупи цієї групи G. У цьому разі критерієм достовірності можуть бути дільники числа елементів групи G. Число елементів групи G становить 8, тому дільники числа елементів цієї групи G дорівнюють 2 та 4. Перш за все спробуємо знайти підгрупу з числом елементів, що дорівнює 2. Нагадаємо, що вона повинна обов'язково містити у собі одиничний елемент.

З аналізу табл. 37 бачимо, що одиничним елементом є комбінація [000]. Отже, необхідно знайти ще один елемент, що залишився, яким може бути будь-який елемент, крім [000]. Розглянемо приклад.

Якщо вибрати [001]

+	[000]	[001]
[000]	[000]	[001]
[001]	[001]	[000]

Якщо вибрати [011]

+	[000]	[011]
[000]	[000]	[011]
[011]	[011]	[000]

Якщо вибрати [101]

+	[000]	[101]
[000]	[000]	[101]
[101]	[101]	[000]

Як ще один одиничний елемент спробуємо вибрати елемент [111], що найбільшою мірою відрізняється від елемента [000]. При цьому підгрупа SG_2 матиме вигляд

$$SG_2 = \{[000], [111]\}.$$

Результати додавання для підгрупи SG_2 занесено до табл. 38.

Виконавши розбиття групи G з використанням підгрупи SG_2, отримаємо результати щодо класів залишків і відповідних їм представницьких елементів (табл. 39).

Розглянемо ці елементи групи G як кодові слова, а також встановимо відповідність між інформацією, що передається, і кодовими словами:

Інформація		Кодове слово
A	$\Leftrightarrow$	[000]
B	$\Leftrightarrow$	[111]

Таблиця 38. Результати додавання для підмножини SG_2

+	[000]	[111]
[000]	[000]	[111]
[111]	[111]	[000]

Таблиця 39. Розбиття групи G за підгрупами SG_2

Клас залишків	Представницький елемент	SG_2	
		[000]	[111]
SG_2 + [000]...	[000]	[000]	[111]
SG_2 + [001]...	[001]	[001]	[110]
SG_2 + [010]...	[010]	[010]	[101]
SG_2 + [011]...	[011]	[011]	[100]

Якщо прийняти викладене вище трактування, то слід вважати розбиття, наведене в табл. 39, розбиттям трибітового кодового вектора за допомогою кодових слів. Зрозуміло, що в такому розбитті відстань між кодовими словами дорівнює трьом. Таким чином, виявляється, що цей код є кодом з виправленням одиничних помилок.

Розбиття кодового вектора було виконано за кодовими словами і, отже, розбиття групи G за допомогою підгрупи SG можна назвати розкладанням на класи залишків.

Таким чином, цей код, що складається з двох кодових слів: [000] та [111], є кодом з виправленням одиничних помилок, тому він дозволяє скорегувати наступні конфігурації помилок: [000], [001], [010], [100].

Конфігурація [000], в якому відсутні помилкові біти, також розглядається як одне із слів помилок: воно відповідає одиничному елементу.

Крім відповідності між класами залишків і представницькими елементами, далі з'ясуємо, як може бути встановлена відповідність між представницькими елементами і конфігураціями помилок:

Клас залишків	*Представницький елемент*		*Конфігурація помилок*
SG_2 + [000]...	[000]	<----->	[000]
SG_2 + [001]...	[001]	<----->	[001]
SG_2 + [010]...	[010]	<----->	[010]
SG_2 + [011]...	[011]	?	[100]

Як бачимо, має місце взаємно однозначна відповідність між представницькими елементами і конфігураціями помилок усіх класів залишків SG_2, крім останнього SG_2+[011]. Відповідність між представницьким елементом [011] класу залишків SG_2+[011] і картиною помилок [100] безпосередньо встановити неможливо. Проте, якщо уважно розглянути рядок класу залишків SG_2+[011], то можна побачити, що в ньому є елемент [100]. Спробуємо побудувати клас залишків SG_2+[011], прийнявши цей елемент [100] за представницький:

Клас залишків	*Представницький елемент*		*Конфігурація помилок*
SG_2 + [000]...	[000]	<----->	[000]
SG_2 + [001]...	[001]	<----->	[001]
SG_2 + [010]...	[010]	<----->	[010]
SG_2 + [100]...	[100]	<----->	[100]

Маємо повну відповідність між представницькими елементами і конфігураціями помилок. Це означає, що якщо як представницькі елементи при розкладанні кодового вектора на класи залишків за допомогою кодових слів вибрати елементи з мінімальною кількістю одиниць, то можна розглядати ці елементи як конфігурації помилок.

Такі представницькі елементи можна, у переносному розумінні, уподібнити старості шкільного класу, тобто старостою може бути вільно обраний будь-який учень класу, що є в даному випадку класом залишків. Зрозуміло, що старостою краще вибирати найбільш гідного учня, але якщо мова йде про лінійні коди, то як представницькі елементи необхідно вибирати елементи з найменшою кількістю одиниць.

Таблиця 40. Стандартна конфігурація для коду {[000], [111]}

Інформація	*A*	*B*
Кодові слова	[000]	[111]
Комбінації помилок	Коди, що приймаються	
[000]	[000]	[111]
[001]	[001]	[110]
[010]	[010]	[101]
[100]	[100]	[011]

У тому випадку, коли представницькі елементи вибираються відповідно до описаних вище принципів, вони приводяться у взаємно однозначну відповідність з комбінаціями (конфігураціями) помилок. Крім того, розміщення, отримане в результаті розкладання на класи залишків, за умови вибору як представницькі елементи конфігурацій з мінімальною кількістю одиниць, називається ***стандартною конфігурацією.***

Розклавши лінійний код у стандартну конфігурацію, можна відразу визначити здатність коду до виправлення помилок (табл. 40).

Якщо побудовано стандартну конфігурацію, то конфігураціями помилок, які можуть бути виправлені, є тільки ті, які обрані як представницькі елементи.

Побудувавши для коду стандартну конфігурацію, на зразок наведеної в табл. 40, в подальшому її можна розглядати як структуру, що ілюструє процес виникнення помилок і методи їх виправлення.

Проаналізуємо ці завдання, використавши результати, подані в табл. 40.

1. Перш за все побудуємо кодові слова, що взаємно однозначно відповідають кодовій інформації. Ці кодові слова являють собою ланцюжки кодових бітів, що відправляються в лінію передачі з боку передавача.

2. У кодових словах, що передаються, під впливом шумів та інших збурних чинників у каналі передачі виникають помилки коду. Ці помилки можна розглядати як результат підсумовування в каналі передачі даних конфігурації помилок, що виникають внаслідок дестабілізувальних чинників, та кодового слова. Таку ситуацію можна трактувати як процес додавання з кодовим словом, що виконує роль підгрупи, конфігурації помилок, що виконує роль представницького елемента, тобто як процес розкладання на класи залишків кодового вектора за кодовим словом, де представницькими елементами є конфігурації помилок.

3. На приймальну сторону надходить та кодова бітова послідовність, що приймається. Вона являє собою кодове слово, до якого додається комбінація помилок:

$$\underset{\text{(кодове слово)}}{[b_2, b_1, b_0]} + \underset{\substack{\text{(конфігурація}\\ \text{помилок)}}}{[e_2, e_1, e_0]} = \underset{\substack{\text{(кодова бітова послідовність,}\\ \text{що приймається)}}}{[b_2 + e_2, b_1 + e_1, b_0 + e_0]}.$$

За цією кодовою бітовою послідовністю, що приймається, можна визначити синдром помилки з використанням матриці перевірки на парність

та комбінацію помилок, що відповідає синдрому. Така комбінація помилок — послідовність бітів, що вибрана як представницький елемент. Шляхом підсумовування цієї комбінації помилок з кодовою бітовою послідовністю, що приймається, можна виправити помилки і отримати початкове (вихідне) кодове слово:

$$\underset{\substack{\text{(кодова бітова послідовність,}\\ \text{що приймається)}}}{[b_2 + e_2,\ b_1 + e_1,\ b_0 + e_0]} + \underset{\substack{\text{(комбінація помилок, що відпо-}\\ \text{відає синдрому помилок)}}}{[e_2,\ e_1,\ e_0]} = \underset{\text{(кодове слово)}}{[b_2,\ b_1,\ b_0]}.$$

4. Отже, якщо до коду додається конфігурація помилок, що не входить до числа комбінації помилок, вибраних як представницькі елементи, то виправити ці помилки буде неможливо. В цьому випадку матиме місце така ситуація, коли помилка, що виникла в коді, перевершуватиме здатність коду до виправлення помилок.

Таким чином, є ще одна підгрупа, кількість елементів якої дорівнює чотирьом. Це означає, що потрібно знайти і цю підгрупу, тобто кодові слова. Як описано вище, при пошуці підгруп на підставі групи, побудованої із залишків від ділення за модулем 16, перш за все потрібно звернути увагу на дільники числа елементів.

Пригадаємо, як було побудовано кодові слова, тобто кодові бітові послідовності, що передаються. У разі циклічного коду кодові слова будували так, щоб вони ділилися без залишку на породжувальний многочлен. А саме: кодові слова виявляються многочленами, які кратні породжувальному многочлену. У випадку лінійного коду кодові слова будувалися у вигляді добутку інформаційної бітової послідовності на породжувальну матрицю. Тут використовується не простий добуток, і кодове слово приймається у формі кратного вектору породжувальної матриці.

Отже, не обмежуючи часу, потрібно виконати перевірку методу, за яким кодові слова визначаються так, що спочатку будується породжувальна матриця, а потім їм у відповідність ставляться інформаційні бітові послідовності.

Для побудови підгрупи, кількість елементів якої дорівнює чотирьом, тобто що складається з чотирьох кодових слів, можна вважати, що як інформаційна бітова послідовність використовується послідовність довжиною два біти. Отже, як кодове слово повинна бути використана бітова послідовність у формі

$$[a_1, a_0, c_0],$$

де a_1, a_0 — інформаційні біти; c_0 — перевірний біт.

Оскільки як перевірний біт вибрано тільки один біт, то згідно з табл. 1 код, що виправляє помилки, побудувати неможливо. З'ясуємо, чи можна побудувати код, що виявляє помилки.

Найпростішим способом побудувати код, що виявляє помилки, є додавання одного біта зведеної перевірки на парність. У кодовому слові саме і є один біт — перевірний, тому спробуємо використати цей біт для зведеної перевірки на парність.

Взаємозв'язок між інформаційними бітами a_1, a_0 і перевірним бітом c_0 можна записати так:

$$c_0 = a_1 + a_0 \pmod 2.$$

Співвідношення між кожним з бітів a_1, a_0, c_0 кодового слова і інформаційною кодовою послідовністю a_1, a_0 приймають наступний вигляд:

$$a_1 = 1 \cdot a_1 + 0 \cdot a_0$$
$$a_0 = 0 \cdot a_1 + 1 \cdot a_0$$
$$c_0 = 1 \cdot a_1 + 1 \cdot a_0 .$$

Виходячи з цих співвідношень, згідно з процедурою, наведеною на рис. 52, можна вивести породжувальну матрицю

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} .$$

Отже, кодове слово v з використанням цієї породжувальної матриці можна побудувати так:

$$v = [a_1, a_0] \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad (135)$$

а саме: отримаємо відповідність між інформаційною послідовністю та кодовим словом:

Інформаційна бітова послідовність		Кодове слово
[00]	--------->	[000]
[01]	--------->	[011]
[10]	--------->	[101]
[11]	--------->	[110]

Отже, чотири кодові слова визначено. Такий код — це код *SED* (3,2) з виявленням одиничних помилок.

Підгрупа SG_4, кількість елементів якої дорівнює чотирьом, — множина, чотирма елементами якої є ці чотири кодові слова:

$$SG_4 = \{[000], [011], [101], [110]\} .$$

Між кодовими словами, побудованими на підставі цієї породжувальної матриці, виконується наступне співвідношення:

[*кодове слово*] + [*кодове слово*] = [*кодове слово*].

З цього співвідношення випливає, що для заданого коду завжди справедлива формула

[*кодове слово*] + [*кодове слово*] ∈ [*множині кодових слів*].

Тобто гарантується замкненість такої множини щодо операції додавання, тому виявляється, що безліч кодових слів, без жодного сумніву, є підгрупою. З метою додаткової перевірки використовують таблицю додавання для кодових слів цього лінійного коду (табл. 41).

Отже, з'ясовано, що в даному випадку виконуються співвідношення

кодовий вектор <-----> *група,*

кодове слово <-----> *підгрупа.*

Таблиця 41. Результат додавання кодових слів лінійного коду (3,2)

+	[000]	[011]	[101]	[110]
[000]	[000]	[011]	[101]	[110]
[011]	[011]	[000]	[110]	[101]
[101]	[101]	[110]	[000]	[011]
[110]	[110]	[101]	[011]	[000]

Спробуємо розкласти групу G на залишки з використанням підгрупи SG_4, кількість елементів якої дорівнює чотирьом. За представницькі елементи, як і раніше, приймали елементи з мінімальним числом одиниць. Перш за все, вибравши елемент [000] як представницький, побудуємо клас залишків SG_4 + [000] (рис. 65).

Далі, з числа елементів групи G, що не входять до складу класу залишків SG_4 + [000], виділимо елементи з найменшим числом одиниць, яких є три: [100], [010], [001]. З них виберемо елемент [001] і спробуємо побудувати клас залишків (рис. 66). У цих двох класах залишків містяться всі елементи групи G.

Видами помилок, які можуть бути виправлені, є тільки конфігурації помилок, вибрані як представницькі елементи. Отже, в наведеному коді, у разі його використання для виправлення помилок, може бути виправлено тільки вид помилок [001]. Проте в тому, що зі всіх можливих одиничних помилок код дозволяє виправити тільки одну, мало сенсу.

Елементи класу залишків SG_4 + [001] суттєво відрізняються від кодових слів. Крім того, в його класах залишків, наприклад,

[001], [010], [100], [111]

містяться всі конфігурації, отримані унаслідок додавання кодових слів та комбінацій помилок з одиничними помилками. Тому при виявленні елементів цього класу залишків стає очевидно, що виникла помилка, а саме: за наявності непарного числа одиниць у кодовій бітовій послідовності, яка приймається, можна зробити висновок, що виникла одинична помилка.

Даний код є кодом з перевіркою на парність, тому можна сказати, що наведений факт випливає з його властивостей.

Класи залишків	Представницькі елементи	SG_4			
		[000]	[011]	[101]	[110]
SG_4 + [000] -----	[000]	[000]	[011]	[101]	[110]

Рис. 65. Схема розкладання на класи залишків за підгрупою SG_4 (у процесі виконання)

Класи залишків	Представницькі елементи	SG_4			
		[000]	[011]	[101]	[110]
SG_4 + [000]-----	[000]	[000]	[011]	[101]	[110]
SG_4 + [001]-----	[001]	[001]	[010]	[100]	[111]

Рис. 66. Схема розкладання на класи залишків за підгрупою SG_4

Вважатимемо, що викладеного матеріалу достатньо, щоб з'ясувати методи побудови стандартної конфігурації для лінійного коду і оцінити їх важливість. Якщо стандартна конфігурація побудована, то є очевидними такі відповідності:

(виправлена) комбінація помилок ----------> представницький елемент,

число станів перевірки ----------> число класів залишків.

А тепер, оскільки досі не розглянуто приклад коду, який за своєю стандартною конфігурацією є типовим лінійним кодом, спробуємо побудувати стандартну конфігурацію ще для одного лінійного коду — коду (6,3). Цей код містить у собі ланцюжок перевірних бітів довжиною три біти. З табл. 1 бачимо, що кодом, який можна побудувати з використанням перевірних бітів, кількість яких $m = 3$, є код з такими характеристиками:

- число кодових бітів $n = 7$;
- число перевірних бітів $m = 3$;
- число інформаційних бітів $k = 4$.

Як було з'ясовано, у разі, якщо число перевірних бітів дорівнює m, можна побудувати лінійний код (n', k'), що задовольняє співвідношення:

$$n' \le n,$$

$$k' \le k,$$

$$n' = m + k'.$$

Якщо прийняти число кодових бітів (n')

$$n' = 6,$$

то число інформаційних бітів (k')

$$k' = 3$$

і очевидно, що можна побудувати лінійний код (6,3).

Це показує, що лінійний код (6,3) — це код, отриманий унаслідок зменшення числа інформаційних бітів у лінійному коді (7,4) з чотирьох бітів до трьох. Отже, перш за все необхідно визначити лінійний код (7,4). І як цей лінійний код (7,4) використаємо код Хеммінга (7,4), описаний в розд. 11.

Код Хеммінга (7,4) — код, який будується на підставі породжувальної матриці

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Якщо позначити інформаційну бітову послідовність a як

$$a = [a_3, a_2, a_1, a_0],$$

то кодове слово v можна побудувати шляхом обчислення за такою формулою:

$$v = [a_3, a_2, a_1, a_0] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [a_3, a_2, a_1, a_0, c_2, c_1, c_0], \qquad (136)$$

де $c_2 = a_3 + a_2 + a_1 \pmod 2$, $c_1 = a_2 + a_1 + a_0 \pmod 2$, $c_0 = a_3 + a_2 + a_0 \pmod 2$.

Введемо обмеження: один інформаційний біт цього коду Хеммінга (7,4), наприклад перший біт (a_3), завжди буде дорівнювати нулю. За такої умови перепишемо формулу (136):

$$v = [0, a_2, a_1, a_0] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [0, a_2, a_1, a_0, c_2, c_1, c_0],$$

де $c_2 = 0 + a_2 + a_1 \pmod 2$, $c_1 = a_2 + a_1 + a_0 \pmod 2$, $c_0 = 0 + a_2 + a_0 \pmod 2$.

На рис. 67 наведено відповідність між інформаційними бітовими послідовностями і кодовими словами для коду Хеммінга (7,4), на який накладено обмеження, що перший біт a_3 завжди дорівнює нулю. Виходячи з обмеження очевидно, що немає ніякої необхідності передавати його по каналах зв'язку. Отже, унаслідок ухваленого рішення цей інформаційний біт a_3 був із самого початку виключений з вживання. Тому інформаційна бітова послідовність a матиме вигляд

$$a = [a_2, a_1, a_0] .$$

Оскільки інформаційний біт a_3 відсутній, то немає необхідності здійснювати його прив'язку до позиції a_3 породжувальної матриці коду Хеммінга (7,4). З інформаційним бітом a_3 пов'язані перший стовпець і перший рядок породжувальної матриці коду Хеммінга (7,4).

З'ясуємо тип матриці, якщо викреслити вказані перший стовпець і перший рядок цієї матриці:

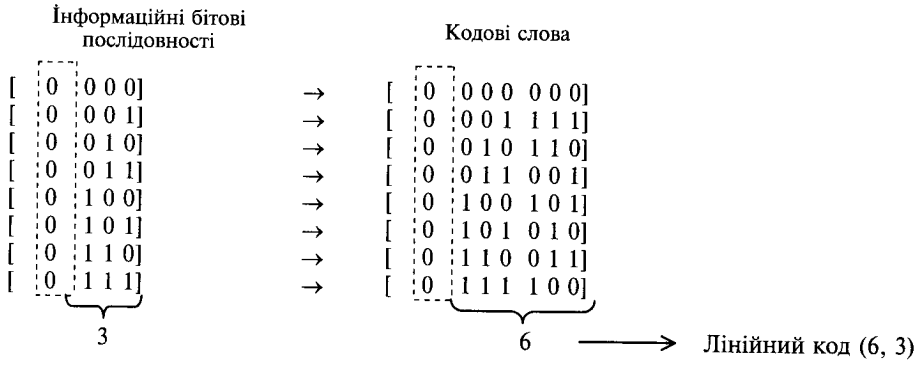


Рис. 67. Відповідність між інформаційними бітовими послідовностями і кодовими словами для коду Хеммінга (7,4)

Перший рядок

$$\left[\begin{array}{|c|cccccc|}\hline 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline\end{array}\right] \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Перший стовпець

Матриця, отримана після викреслення першого стовпця і першого рядка з породжувальної матриці коду Хеммінга (7,4), є породжувальною матрицею лінійного коду (6,3), побудованого унаслідок вилучення інформаційного біта a_3 з коду Хеммінга (7,4). Отже, породжувальна матриця лінійного коду (6,3) має вигляд

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Якщо позначити інформаційну бітову послідовність a так:

$$a = [a_2, a_1, a_0] ,$$

то кодове слово v можна буде побудувати шляхом обчислення за формулою

$$v = [a_2, a_1, a_0] \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \qquad (137)$$

Цей код отримано унаслідок введення обмеження, а саме: один інформаційний біт коду Хеммінга (7,4) завжди дорівнює нулю. Тому він за своєю здатністю до виправлення помилок так само, як і код Хеммінга (7,4), є кодом з виправленням одиничних помилок. Проте для певності виконаємо перевірку. Здійснивши розрахунки за формулою (137), знайдемо інформацію, що міститься в перевірних бітах:

$$\begin{aligned} v = [a_2 \cdot 1 + a_1 \cdot 0 + a_0 \cdot 0, &\quad \text{перший біт } (a_2) \\ a_2 \cdot 0 + a_1 \cdot 1 + a_0 \cdot 0, &\quad \text{другий біт } (a_1) \\ a_2 \cdot 0 + a_1 \cdot 0 + a_0 \cdot 1, &\quad \text{третій біт } (a_0) \\ a_2 \cdot 1 + a_1 \cdot 1 + a_0 \cdot 0, &\quad \text{четвертий біт } (c_2) \\ a_2 \cdot 1 + a_1 \cdot 1 + a_0 \cdot 1, &\quad \text{п'ятий біт } (c_1) \\ a_2 \cdot 1 + a_1 \cdot 0 + a_0 \cdot 1] =, &\quad \text{шостий біт } (c_0) \end{aligned}$$

$$= [a_2, a_1, a_0, a_2 + a_1, a_2 + a_1 + a_0, a_2 + a_0] = [a_2, a_1, a_0, c_2, c_1, c_0], \qquad (138)$$

де $c_2 = a_2 + a_1 \pmod 2$, $c_1 = a_2 + a_1 + a_0 \pmod 2$, $c_0 = a_2 + a_0 \pmod 2$.

Таблиця 42. Інформаційні біти і перевірні біти, що їх включають

Інформаційні біти	Перевірні біти	Кодова відстань
a_2	c_2, c_1, c_0	4
a_1	c_2, c_1	3
a_0	c_1, c_0	3

У табл. 42 відображено, як здійснюється включення інформаційних бітів до складу перевірних бітів. Згідно з цією таблицею, мінімальна кодова відстань дорівнює трьом. Тобто вдалося довести, що за своєю здатністю до виправлення помилок цей код є кодом з виправленням одиничних помилок.

Для лінійного коду (6,3) виконуються наступні закономірності:

- *число кодових векторів дорівнює* $2^6 = 64$

----------> *група з числом елементів, що дорівнює* 64;

- *число кодових слів дорівнює* $2^3 = 8$

----------> *підгрупа з числом елементів, що дорівнює* 8;

- *число станів, які можна відобразити за допомогою перевірних бітів, дорівнює* $2^3 = 8$

----------> 8 *класів залишків.*

Отже встановлено, що цей код є кодом з виправленням одиничних помилок, тому можна вважати доведеним, що серед усіх можливих комбінацій помилок як представницькі елементи можна використовувати наступні сім комбінацій помилок:

[000000] *комбінація відсутності помилок*

[100000]
[010000]
[001000]
[000100]
[000010]
[000001] } *комбінації одиничних помилок*

У табл. 43 подана стандартна конфігурація для досліджуваного лінійного коду (6,3). Число класів залишків дорівнює восьми, тому можна вибрати ще одну комбінацію помилок, крім перелічених вище. У табл. 43 з числа комбінацій помилок, що залишилися, як представницький елемент вибрано комбінацію помилок [110000].

Таблиця 43. Стандартна конфігурація для лінійного коду (6, 3)

Інформаційна бітова послідовність	000	001	010	011	100	101	110	111
Кодові слова	000000	001111	010110	011001	100101	101010	110011	111100
Комбінації помилок	Коди, що приймаються							
000000	000000	001111	010110	011001	100101	101010	110011	111100
100000	100000	101111	110110	111001	000101	001010	010011	011100
010000	010000	011111	000110	001001	110101	111010	100011	101100
001000	001000	000111	011110	010001	101101	100010	111011	110100
000100	000100	001011	010010	011101	100001	101110	110111	111000
000010	000010	001101	010100	011011	100111	101000	110001	111110
000001	000001	001110	010111	011000	100100	101011	110010	111101
110000	110000	111111	100110	101001	010101	011010	000011	001100

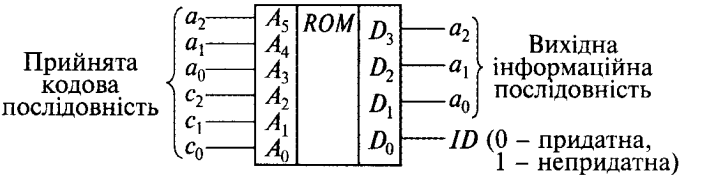


Рис. 68. Схема декодування лінійного коду (6, 3) з використанням ПЗП

Комбінація помилок, що вибрана як представницький елемент, може бути виправлена. Так, можуть бути виправлені всі одиничні помилки, а з числа подвійних помилок може бути виправлена тільки комбінація, що вибрана як представницький елемент [110000]. Проте, хоча з числа всіх можливих подвійних помилок і може бути виправлена певна подвійна помилка, в цьому особливого сенсу немає. Отже, можна вважати, що клас залишків, в якому як представницький елемент вибрана конфігурація [110000], не використовується для виправлення помилок. У зв'язку з цим виникає питання, чи можливе виявлення подвійних помилок. На жаль, уважно розглянувши табл. 43, дійшли висновку, що картини подвійних помилок входять у решту класів залишків, тому неможливо виявити жодної подвійної помилки. А саме: клас залишків, побудований на основі вибору як представницьких елементів комбінації [110000], не може використовуватися при декодуванні цього коду. Таким чином, виявляється, що для декодування можуть застосовуватися тільки ті класи залишків, які побудовані на основі вибору як представницьких елементів комбінації відсутності помилок і комбінацій одиничних помилок, для яких забезпечується виправлення помилок.

Як пристрій декодування лінійного коду розглянемо схему (рис. 68), що виконана з використанням постійного запам'ятовувального пристрою (ПЗП) (див. розд. 8).

У цій схемі декодування прийнята кодова бітова послідовність використовується як адреса, а вихідними даними є інформаційна бітова послідовність та ідентифікаційна інформація (*ID*), що вказує, придатною чи ні є вихідна інформація для використання. Що стосується перевірних бітів, то необхідність їх подачі на вихід, мабуть, відсутня.

З усіх можливих прийнятих кодових бітових послідовностей, які надходять на вхід, для декодування не можуть використовуватися ті, які входять у класи залишків табл. 43, розташовані нижче від першого. Отже, біт *ID* для даних, що відповідають такій прийнятій кодовій бітовій послідовності (адресі), встановлюється в одиницю, і це вказує, що дані непридатні для використання.

§15.3. ПОДАННЯ ЛІНІЙНОГО КОДУ У ВИГЛЯДІ ЦИКЛІЧНОГО КОДУ

Раніше з коду Хеммінга (7, 4), сформованого як циклічний код, було побудовано код Хеммінга (7, 4), що є лінійним кодом. А саме: кодування коду Хеммінга (7, 4) здійснювалося з використанням процедури кодування циклічного коду, будувалися кодові слова (кодові бітові послідовності, що передаються в лінію передачі), з яких виводилася породжувальна матриця лінійного коду. Очевидно, що при кодуванні з використанням як породжувального

многочлена, так і породжувальної матриці утворюються абсолютно однакові кодові слова. Таким чином, той факт, що циклічний код можна побудувати з використанням процедури кодування лінійного коду, вказує на те, що циклічний код — це лінійний код.

Отже, підкласом лінійних кодів є циклічні коди. Вони характеризуються тим, що всі набори, утворені циклічною перестановкою будь-якої кодової комбінації, є також кодовими комбінаціями. Ця властивість дозволяє значною мірою спростити кодувальний і декодувальний пристрої, особливо при виявленні помилок і виправленні одиничної помилки. Прикладами циклічних кодів є коди Хеммінга, Боуза—Чоудхурі—Хоквінгема (БЧХ-коди) тощо.

Циклічним кодом є лінійний код, для якого властиве наступне: якщо він є кодовим словом, то його циклічна перестановка також є кодовим словом. Слова циклічного коду зручно записувати у вигляді многочленів.

Серед роздільних кодів розрізняють лінійні і нелінійні. До лінійних належать коди, в яких порозрядна сума за модулем 2 будь-яких двох кодових слів також є кодовим словом. Лінійний код називається систематичним, якщо його перші k символів будь-якої кодової комбінації є інформаційними, інші $(n–k)$ символів — перевірними.

Серед лінійних систематичних кодів найпростішій код $(n, n–k)$, що містить у собі один перевірний символ, який дорівнює сумі за модулем 2 усіх інформаційних символів. Цей код — код з перевірки на парність — дозволяє виявити всі поєднання помилок непарної кратності. Вірогідність невиявленої помилки у першому наближенні можна визначити як вірогідність спотворення двох символів.

Проте у протилежному випадку, тобто, якщо потрібно з'ясувати, чи є лінійний код циклічним, неминуче виявляється, що результат може бути іншим. Наприклад, вище з використанням інформаційної бітової послідовності $a = [a_3, a_2, a_1, a_0]$ і породжувальної матриці

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{139}$$

було побудовано код Хеммінга (7, 4):

$$v = aG = [a_3, a_2, a_1, a] \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} =$$

$$= [a_3, a_2, a_1, a_0, c_2, c_1, c_0]. \tag{140}$$

Тут $c_2 = a_3 + a_2 + a_1 \pmod 2$, $c_1 = a_2 + a_1 + a_0 \pmod 2$, $c_0 = a_3 + a_2 + a_0 \pmod 2$.

Етапи проведення обчислень, які необхідно виконати для визначення кодового слова з інформаційної бітової послідовності і породжувальної матриці, наведено на рис. 69.

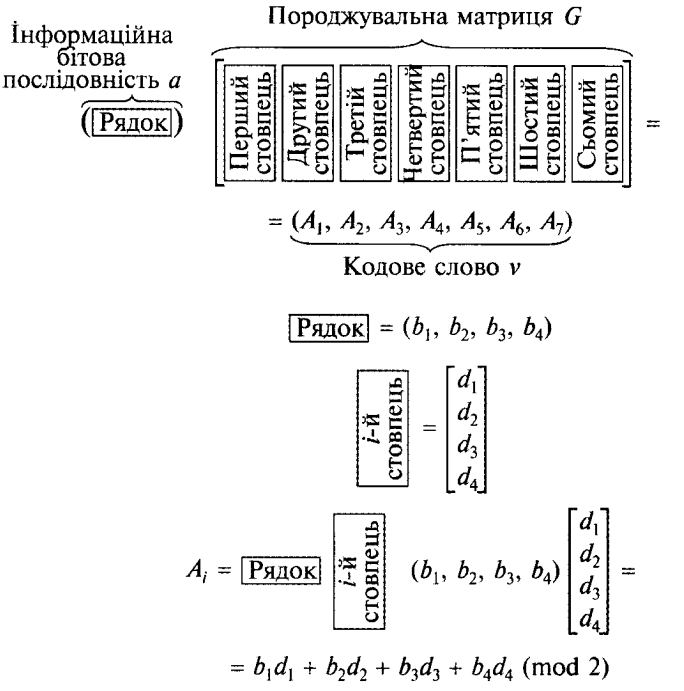


Рис. 69. Схема отримання лінійного коду

З'ясуємо, що відбудеться при невеликій зміні порядку розташування стовпців у цій породжувальній матриці. Наприклад, спробуємо переставити місцями четвертий і п'ятий стовпці матриці:

$$G = \begin{pmatrix} 1 & 0 & 0 & |0| & |1| & 0 & 1 \\ 0 & 1 & 1 & |0| & |1| & 1 & 1 \\ 0 & 0 & 1 & |0| & |1| & 1 & 0 \\ 0 & 0 & 0 & |1| & |0| & 1 & 1 \end{pmatrix}$$

$$G' = \begin{pmatrix} 1 & 0 & 0 & |1| & |0| & 0 & 1 \\ 0 & 1 & 0 & |1| & |0| & 1 & 1 \\ 0 & 0 & 1 & |1| & |0| & 1 & 0 \\ 0 & 0 & 0 & |0| & |1| & 1 & 1 \end{pmatrix}$$

Навіть використовуючи породжувальну матрицю, в якій змінено порядок розташування стовпців, отримуємо

$$v = aG' = [a_3, a_2, a_1, a] \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} = [a_3, a_2, a_1, c_2, a_0, c_1, c_0].$$

Тут $c_2 = a_3 + a_2 + a_1 \pmod 2$, $c_1 = a_2 + a_1 + a_0 \pmod 2$, $c_0 = a_3 + a_2 + a_0 \pmod 2$.

Отже, і у цьому випадку можна побудувати код Хеммінга (7,4) (лінійний код). Створити лінійний код можна за допомогою породжувальної матриці, тому навіть такий код, що побудований з використанням породжувальної матриці, в якій змінений порядок розташування стовпців, є лінійним.

Якщо лінійний код будується з використанням породжувальної матриці G', що сформована на основі породжувальної матриці G, в якій переставлені місцями стовпці, то в кодовому слові, отриманому на її основі, інформаційні біти і перевірні біти поміняються місцями. У циклічному коді перевірні біти записуються після інформаційних бітів і розташовані окремо один від одного, а саме:

кодове слово: [*інформаційні біти*, *перевірні біти*].

Кодове слово, в якому переставлені місцями інформаційні і перевірні біти, неможливо побудувати за допомогою процедури отримання циклічного коду. Побудова лінійного коду здійснюється з використанням породжувальної матриці. Тому, якщо порядок розташування в ній стовпців змінено, то надалі не можна чітко розпізнати, де знаходяться інформаційні біти, а де перевірні біти. З використанням такої матриці можна будувати кодові слова, що мають таку форму, де інформаційні біти і перевірні біти переставлені місцями. Іншими словами, виявляється, що ступінь вільності при побудові лінійного коду більший ніж у разі циклічного коду.

Оскільки циклічний код можна подати у вигляді лінійного коду, то для побудови коду БЧХ використовують породжувальні многочлени, а для декодування — матриці перевірки на парність.

Таким чином, той факт, що циклічний код може бути поданий у вигляді лінійного коду, дозволяє розраховувати на те, що при кодуванні з використанням лінійного і циклічного кодів та їх декодуванні можна встановити взаємно однозначні відповідності (табл. 44).

Цікаво зіставити код Хеммінга (7, 4), тобто код з виправленням одиничних помилок (7, 4) БЧХ, і код з виправленням подвійних помилок (15, 7) БЧХ, використовуючи співвідношення взаємно однозначної відповідності (див. табл. 44).

Таблиця 44. Співвідношення між циклічним і лінійним кодами

Процес	Циклічний код		Лінійний код
Кодування	Породжувальний многочлен	⇔	Породжувальна матриця
	Схема кодування	⇔	[Інформаційна бітова послідовність] × [Породжувальна матриця]
Декодування	Схема ділення на породжувальний многочлен	⇔	Матриця перевірки на парність
	Залишок породжувального многочлена	⇔	Синдром

15.3.1. Породжувальний многочлен і породжувальна матриця

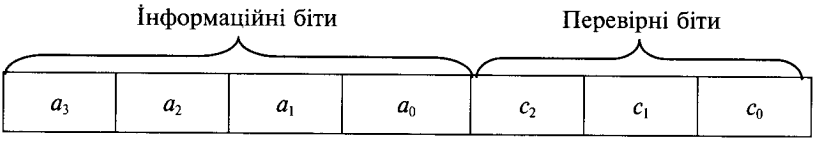
Код (7, 4) БЧХ — це код, який будується з використанням породжувального многочлена $x^3 + x + 1$. Для цього коду раніше була виведена породжувальна матриця, як для коду Хеммінга (7, 4), і він був побудований як лінійний код.

Порівняємо породжувальний многочлен, який використовується, коли код (7, 4) БЧХ будується у вигляді циклічного коду, та породжувальну матрицю, що застосовується, коли цей код будується у вигляді лінійного коду:

Породжувальний многочлен для випадку, коли код будується у вигляді циклічного коду

Породжувальна матриця для випадку, коли код будується у вигляді лінійного коду

$$x^3 + x + 1 \leftrightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$



$$c_2 = a_3 + a_2 + a_1 \pmod 2,$$
$$c_1 = a_2 + a_1 + a_0 \pmod 2,$$
$$c_0 = a_3 + a_2 + a_0 \pmod 2.$$

Рис. 70. Приклад кодового слова в коді (7, 4) БЧХ

У цьому разі незалежно від того, використовується чи породжувальний многочлен, чи породжувальна матриця, з інформаційної бітової послідовності можна побудувати кодове слово в коді (7, 4) БЧХ (рис. 70).

Тепер розглянемо код (15, 7) БЧХ, який будується з використанням породжувального многочлена $x^8 + x^7 + x^6 + x^4 + 1$. Процедура кодування в цьому коді наведена на рис. 71. Як бачимо, між інформаційними бітами a_6—a_0 і перевірними бітами c_7—c_0 та інформаційною бітовою послідовністю a_6—a_0 виконуються такі співвідношення:

кодове слово ↔ інформаційна бітова послідовність

$$\left.\begin{array}{lcccccccc}
a_6 & = & a_6 & & & & & & \\
a_5 & = & & a_5 & & & & & \\
a_4 & = & & & a_4 & & & & \\
a_3 & = & & & & a_3 & & & \\
a_2 & = & & & & & a_2 & & \\
a_1 & = & & & & & & a_1 & \\
a_0 & = & & & & & & & a_0 \\
c_7 & = & a_6 & & & & + a_2 & & + a_0 \\
c_6 & = & a_6 & + a_5 & & & + a_2 & + a_1 & + a_0 \\
c_5 & = & a_6 & + a_5 & + a_4 & & + a_2 & + a_1 & \\
c_4 & = & & a_5 & + a_4 & + a_3 & & + a_1 & + a_0 \\
c_3 & = & a_6 & & + a_4 & + a_3 & & & \\
c_2 & = & & a_5 & & + a_3 & + a_2 & & \\
c_1 & = & & & a_4 & & + a_2 & + a_1 & \\
c_0 & = & & & & a_3 & & + a_1 & + a_0
\end{array}\right\}. \qquad (141)$$

З формули (141) випливає, що породжувальна матриця G, яка призначена для побудови коду (15, 7) БЧХ у вигляді лінійного коду, має вигляд (рис. 72):

$$G = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}. \qquad (142)$$

Якщо прийняти, що інформаційна бітова послідовність має вигляд $a = [a_6, a_5, a_4, a_3, a_2, a_1, a_0]$, то використовуючи породжувальну матрицю (142), отримуємо

$$v = aG = [a_6, a_5, a_4, a_3, a_2, a_1, a_0]$$

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix} =$$

$$= [a_6, a_5, a_4, a_3, a_2, a_1, a_0, c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0]. \qquad (143)$$

Тут при кодуванні в коді (15, 7) БЧХ можуть виконуватися відповідності:

$$c_7 = a_6 + a_2 + a_0 \pmod 2,$$
$$c_6 = a_6 + a_5 + a_2 + a_1 + a_0 \pmod 2,$$
$$c_5 = a_6 + a_5 + a_4 + a_2 + a_1 \pmod 2,$$
$$c_4 = a_5 + a_4 + a_3 + a_1 + a_0 \pmod 2,$$
$$c_3 = a_6 + a_4 + a_3 \pmod 2,$$
$$c_2 = a_5 + a_3 + a_2 \pmod 2,$$
$$c_1 = a_4 + a_2 + a_1 \pmod 2,$$
$$c_0 = a_3 + a_1 + a_0 \pmod 2.$$

Таким чином, у коді (15, 7) БЧХ мають місце наступні взаємно-однозначні відповідності:

Породжувальний многочлен для випадку, коли код будується у вигляді циклічного коду

Породжувальна матриця для випадку, коли код будується у вигляді лінійного коду

$$x^8 + x^7 + x^6 + x^4 + 1 \longleftrightarrow \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

Тобто, незалежно від того, чи виконується кодування з використанням цього породжувального многочлена, чи цієї породжувальної матриці, можна отримати одне і те саме кодове слово в коді (15, 7) БЧХ (рис. 73).

Код БЧХ — це циклічний код. Отже, у цьому разі кодування можна здійснювати з використанням породжувального многочлена. У випадку, коли код БЧХ розглядається як лінійний, кодування можна виконувати з використанням породжувальної матриці.

За своєю здатністю до виправлення помилок код (15, 7) БЧХ є код з виправленням подвійних помилок. Перевіримо здатність цього коду до виправлення помилок. У табл. 45 показано, в яких перевірних бітах містяться інформаційні біти коду (15, 7) БЧХ.

Таблиця 45. Інформаційні біти і перевірні біти, що містять їх

Інформаційні біти	Перевірні біти, що містять інформаційні біти	Загальне число бітів, що піддаються зміні при зміні інформаційних бітів
a_6	c_7, c_6, c_5, c_3	5
a_5	c_6, c_6, c_4, c_2	5
a_4	c_5, c_4, c_3, c_1	5
a_3	c_4, c_3, c_2, c_0	5
a_2	c_7, c_6, c_2, c_1	5
a_1	c_6, c_5, c_4, c_1, c_0	6
a_0	c_7, c_6, c_4, c_0	5

1. Інформаційна бітова послідовність
   $a = (a_6, a_5, a_4, a_3, a_2, a_1, a_0)$
2. Породжувальний многочлен $G(x)$
   $G(x) = x^8 + x^7 + x^6 + x^4 + 1$
3. $A(x) \cdot x^8 \,/\, G(x)$ залишок $R(x)$
   як результат ділення
   $A(x) \cdot x^8$ на $G(x)$

Інформаційний многочлен $A(x)$
$A(x) = a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

$A(x)x^8$ (степінь породжувального многочлена)
$A(x) \cdot x^8 = a_6x^{14} + a_5x^{13} + a_4x^{12} + a_3x^{11} + a_2x^{10} + a_1x^9 + a_0x^8$

$$a_6x^6 + \binom{a_0}{+}{a_5}x^5 + \binom{a_5}{+}{a_4}x^4 + \binom{a_6}{+}{a_4}{+}{a_3}x^3 + \binom{a_5}{+}{a_3}{+}{a_2}x^2 + \binom{a_4}{+}{a_2}{+}{a_1}x + \binom{a_3}{+}{a_1}{+}{a_0}$$

$x^8 + x^7 + x^6 + x^4 + 1)$ $\quad a_6x^4 + a_5x^{13} + a_4x^{12} + a_3x^{11} + a_2x^{10} + a_1x^9 + a_0x^8$

$$a_6x^4 + a_6x^{13} + a_6x^{12} \quad + a_6x^{10} \quad + a_6x^6$$

$$\binom{a_6}{+}{a_5}x^{13} + \binom{a_6}{+}{a_4}x^{12} + a_3x^{11} + \binom{a_6}{+}{a_2}x^{10} + a_1x^9 + a_0x^8 + a_6x^6$$

$$\binom{a_6}{+}{a_5}x^{13} + \binom{a_6}{+}{a_5}x^{12} + \binom{a_6}{+}{a_5}x^{11} \quad + \binom{a_6}{+}{a_5}x^9 \quad + \binom{a_6}{+}{a_5}x^5$$

$$\binom{a_5}{+}{a_4}x^{12} + \binom{a_6}{+}{a_5}{+}{a_3}x^{11} + \binom{a_6}{+}{a_2}x^{10} \quad + \binom{a_6}{-}{a_5}{-}{a_1}x^9 + a_0x^8 \quad + a_0x^6 + \binom{a_6}{+}{a_5}x^5$$

$$\binom{a_5}{+}{a_4}x^{12} + \binom{a_5}{+}{a_4}x^{11} + \binom{a_5}{+}{a_4}x^{10} \quad + \binom{a_5}{+}{a_4}x^8 \quad + \binom{a_5}{+}{a_4}x^4$$

$$\binom{a_6+a_4+a_3}{}x^{11} + \binom{a_6+a_5+a_4+a_4+a_2}{}x^{10} + \binom{a_6+a_5+a_5+a_1}{}x^{9} + \binom{a_5+a_4+a_0}{}x^{8} + \binom{a_6+a_5}{}x^{7} + a_6x^6 + \binom{a_6+a_5}{}x^{5} + \binom{a_5+a_4}{}x^{4} + \binom{a_6+a_4+a_3}{}x^{3}$$

$$\binom{a_6+a_4+a_3}{}x^{11} + \binom{a_6+a_4+a_3}{}x^{10} + \binom{a_6+a_4+a_3}{}x^{9} + \binom{a_6+a_4+a_3}{}x^{7} + \binom{a_6+a_4+a_3}{}x^{3}$$

$$\binom{a_5+a_3+a_2}{}x^{10} + \binom{a_5+a_4+a_3+a_1}{}x^{9} + \binom{a_5+a_4+a_0}{}x^{8} + \binom{a_6+a_4+a_3}{}x^{7} + a_6x^6 + \binom{a_6+a_5}{}x^{5} + \binom{a_5+a_4}{}x^{4} + \binom{a_6+a_4+a_3}{}x^{3}$$

$$\binom{a_5+a_3+a_2}{}x^{10} + \binom{a_5+a_3+a_2}{}x^{9} + \binom{a_5+a_3+a_2}{}x^{8} + \binom{a_5+a_3+a_2}{}x^{6} + \binom{a_5+a_3+a_2}{}x^{2}$$

$$\binom{a_4+a_2+a_1}{}x^{9} + \binom{a_4+a_3+a_2+a_0}{}x^{8} + \binom{a_6+a_4+a_3}{}x^{7} + \binom{a_6+a_5+a_3+a_2}{}x^{6} + \binom{a_6+a_5}{}x^{5} + \binom{a_5+a_4}{}x^{4} + \binom{a_6+a_4+a_3}{}x^{3} + \binom{a_5+a_3+a_2}{}x^{2}$$

$$\binom{a_4+a_2+a_1}{}x^{9} + \binom{a_4+a_2+a_1}{}x^{8} + \binom{a_4+a_2+a_1}{}x^{7} + \binom{a_4+a_2+a_1}{}x^{5} + \binom{a_4+a_2+a_1}{}x$$

$$\begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^8 + \begin{pmatrix} a_6 \\ + \\ a_3 \\ + \\ a_2 \\ + \\ a_1 \end{pmatrix} x^7 + \begin{pmatrix} a_6 \\ + \\ a_5 \\ + \\ a_3 \\ + \\ a_2 \end{pmatrix} x^6 + \begin{pmatrix} a_6 \\ + \\ a_5 \\ + \\ a_4 \\ + \\ a_2 \\ + \\ a_1 \end{pmatrix} x^5 + \begin{pmatrix} a_5 \\ + \\ a_4 \end{pmatrix} x^4 + \begin{pmatrix} a_6 \\ + \\ a_4 \\ + \\ a_3 \end{pmatrix} x^3 + \begin{pmatrix} a_5 \\ + \\ a_3 \\ + \\ a_2 \end{pmatrix} x^2 + \begin{pmatrix} a_4 \\ + \\ a_2 \\ + \\ a_1 \end{pmatrix} x$$

$$\begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^8 + \begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^7 + \begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^6 + \begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^4 + \begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix}$$

$$\begin{pmatrix} a_6 \\ + \\ a_2 \\ + \\ a_0 \end{pmatrix} x^7 + \begin{pmatrix} a_6 \\ + \\ a_5 \\ + \\ a_2 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^6 + \begin{pmatrix} a_6 \\ + \\ a_5 \\ + \\ a_4 \\ + \\ a_2 \\ + \\ a_1 \end{pmatrix} x^5 + \begin{pmatrix} a_5 \\ + \\ a_4 \\ + \\ a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix} x^4 + \begin{pmatrix} a_6 \\ + \\ a_4 \\ + \\ a_3 \end{pmatrix} x^3 + \begin{pmatrix} a_5 \\ + \\ a_3 \\ + \\ a_2 \end{pmatrix} x^2 + \begin{pmatrix} a_4 \\ + \\ a_2 \\ + \\ a_1 \end{pmatrix} x + \begin{pmatrix} a_3 \\ + \\ a_1 \\ + \\ a_0 \end{pmatrix}$$

$\longrightarrow$ $R(x) = c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$

$c_7 = a_6 + a_2 + a_0 \pmod 2$
$c_6 = a_6 + a_5 + a_2 + a_1 + a_0 \pmod 2$
$c_5 = a_6 + a_5 + a_4 + a_2 + a_1 \pmod 2$
$c_4 = a_6 + a_4 + a_3 + a_1 + a_0 \pmod 2$
$c_3 = a_6 + a_4 + a_3 \pmod 2$
$c_2 = a_5 + a_3 + a_2 \pmod 2$
$c_1 = a_4 + a_2 + a_1 \pmod 2$
$c_0 = a_3 + a_1 + a_0 \pmod 2$

4. $A(x) \cdot x^8 + R(x)$ $\longrightarrow$ Кодовий многочлен $V(x)$

$V(x) = a_6x^{14} + a_5x^{13} + a_4x^{12} + a_3x^{11} + a_2x^{10} + a_1x^9 + a_0x^8 + c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$

5. Кодовий многочлен $V(x)$ $\longrightarrow$ Кодова послідовність, що передається

$v = [a_6, a_5, a_4, a_3, a_2, a_1, a_0, c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0]$

Рис. 71. Процедура кодування в коді (15, 7) БЧХ

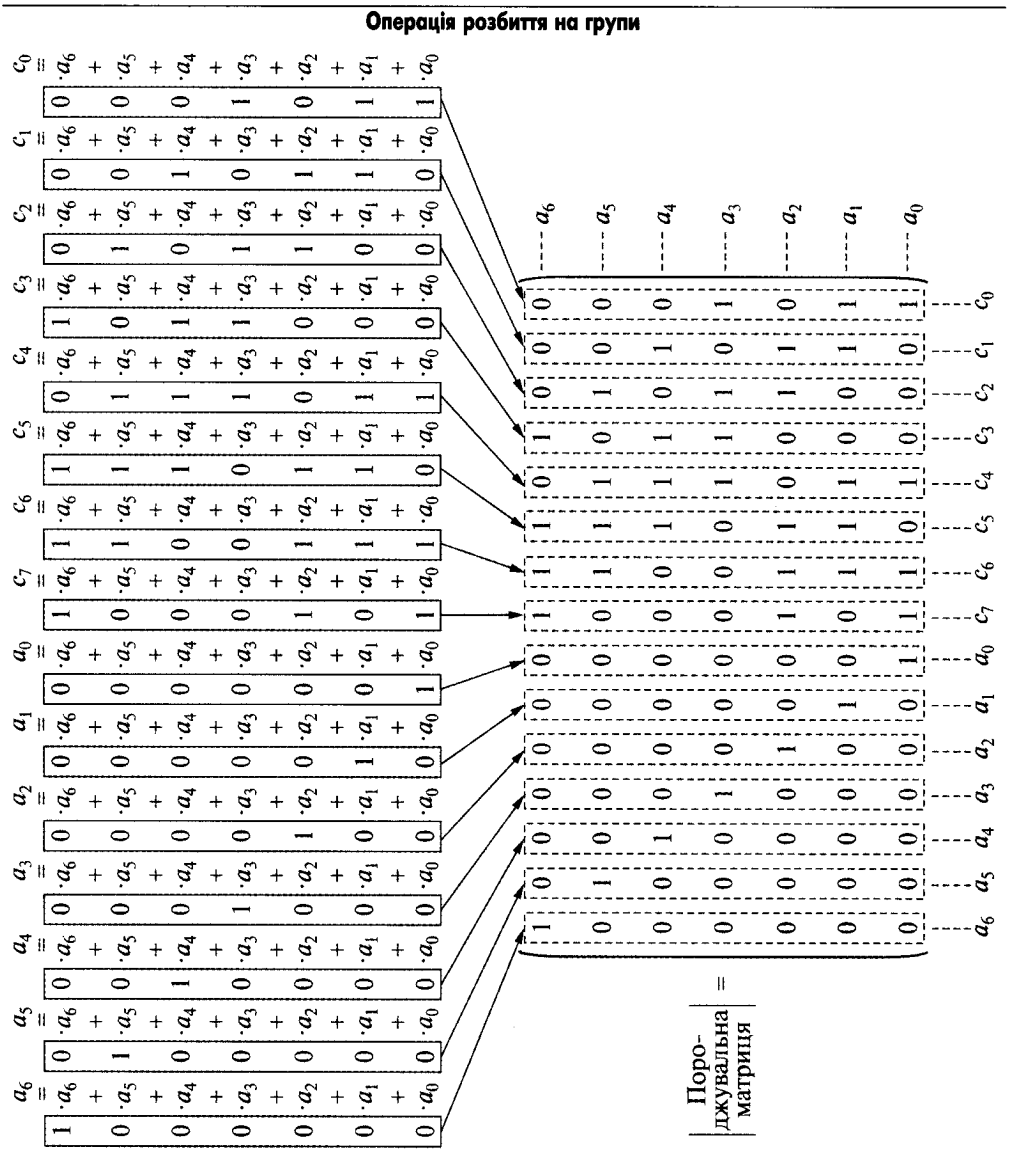


Рис. 72. Приклад породжувальної матриці коду (15, 7) БЧХ

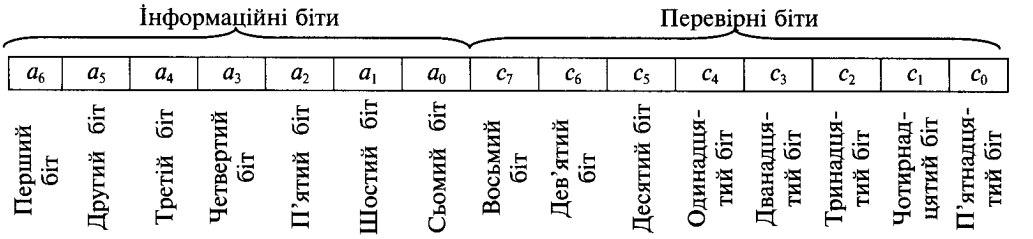


Рис. 73. Приклад кодового слова в коді (15, 7) БЧХ

Як видно з табл. 45, кодова відстань становить 5 або більше, отже, доведено, що за своєю спроможністю до виправлення помилок код (15, 7) БЧХ — це код з виправленням подвійних помилок.

15.3.2. Породжувальна матриця та матриця перевірки на парність

Як було з'ясовано, між породжувальною матрицею G і матрицею перевірки на парність H лінійного коду існують наступні співвідношення

$$GH^T = 0, \tag{144}$$

або

$$HG^T = 0.$$

Тут "0" — матриця, в якій всі елементи дорівнюють нулю (нульова матриця).

Перевіримо формули (144) на прикладі коду (7, 4) БЧХ. З табл. 10 і табл. 11 випливає, що для коду (7, 4) БЧХ співвідношення між синдромом і розташуванням помилкових бітів мають вигляд, наведений у табл. 46. Згідно з цією таблицею, матрицю перевірки на парність H коду (7, 4) БЧХ можна записати так:

$$H = (a^6, a^5, a^4, a^3, a^2, a^1, a^0) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{145}$$

Крім того, відповідно до викладеного вище, породжувальна матриця коду (7, 4) БЧХ може бути подана так:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{146}$$

Обчислення добутку породжувальної матриці G, і транспонованої матриці перевірки на парність, тобто H^T, виконуються за методом, наведеним на рис. 74.

Спробуємо тепер обчислити добуток породжувальної матриці G і транспонованої матриці перевірки на парність H^T коду (7, 4) БЧХ за методом обчислення матриць (див. рис. 74):

$$GH^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}^T =$$

Таблиця 46. Синдром і положення помилкових бітів у коді (7, 4) БЧХ

Синдром			Розміщення помилкових бітів
Відображення елементів групи $GF(2^3)$	Відображення горизонтального вектора	Відображення вертикального вектора	
0	[0 0 0]	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	Помилкові біти відсутні
a^6	[1 0 1]	$\begin{bmatrix}1\\0\\1\end{bmatrix}$	Перший біт (e_6)
a^5	[1 1 1]	$\begin{bmatrix}1\\1\\1\end{bmatrix}$	Другий біт (e_5)
a^4	[1 1 0]	$\begin{bmatrix}1\\1\\0\end{bmatrix}$	Третій біт (e_4)
a^3	[0 1 1]	$\begin{bmatrix}0\\1\\1\end{bmatrix}$	Четвертий біт (e_3)
a^2	[1 0 0]	$\begin{bmatrix}1\\0\\0\end{bmatrix}$	П'ятий біт (e_2)
a	[0 1 0]	$\begin{bmatrix}0\\1\\0\end{bmatrix}$	Шостий біт (e_1)
a^0	[0 0 1]	$\begin{bmatrix}0\\0\\1\end{bmatrix}$	Сьомий біт (e_0)

$$= \begin{pmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix}\begin{pmatrix}1&0&1\\1&1&1\\1&1&0\\0&1&1\\1&0&0\\0&1&0\\0&0&1\end{pmatrix} = \begin{pmatrix}q_{11}&q_{12}&q_{13}\\q_{21}&q_{22}&q_{23}\\q_{31}&q_{32}&q_{33}\\q_{41}&q_{42}&q_{43}\end{pmatrix}.$$

З цієї формули маємо:

$$q_{11} = (\text{Перший рядок})\begin{pmatrix}1\\1\\1\\0\\1\\0\\1\end{pmatrix} = (1000101)\begin{pmatrix}1\\1\\1\\0\\1\\0\\1\end{pmatrix} = 1\cdot1+0\cdot1+0\cdot1+0\cdot0+1\cdot1+0\cdot0+1\cdot1=0 \pmod 2$$

(Перший стовпець)

і далі.

Отже, отримуємо

$$GH^T = \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\end{pmatrix} = 0.$$

За формулою (144) нескладно визначити з породжувальної матриці G матрицю перевірки на парність H і, навпаки, з матриці перевірки на парність H — породжувальну матрицю G.

Перш за все звернемо увагу на те, в якій формі записані породжувальна матриця G і матриця перевірки на парність H. У коді (7, 4) БЧХ породжувальна матриця G набуває вигляду

$$G = \begin{pmatrix}1&0&0&0&a_1&a_2&a_3\\0&1&0&0&b_1&b_2&b_3\\0&0&1&0&c_1&c_2&c_3\\0&0&0&1&d_1&d_2&d_3\end{pmatrix} = (I_4, P). \qquad (147)$$

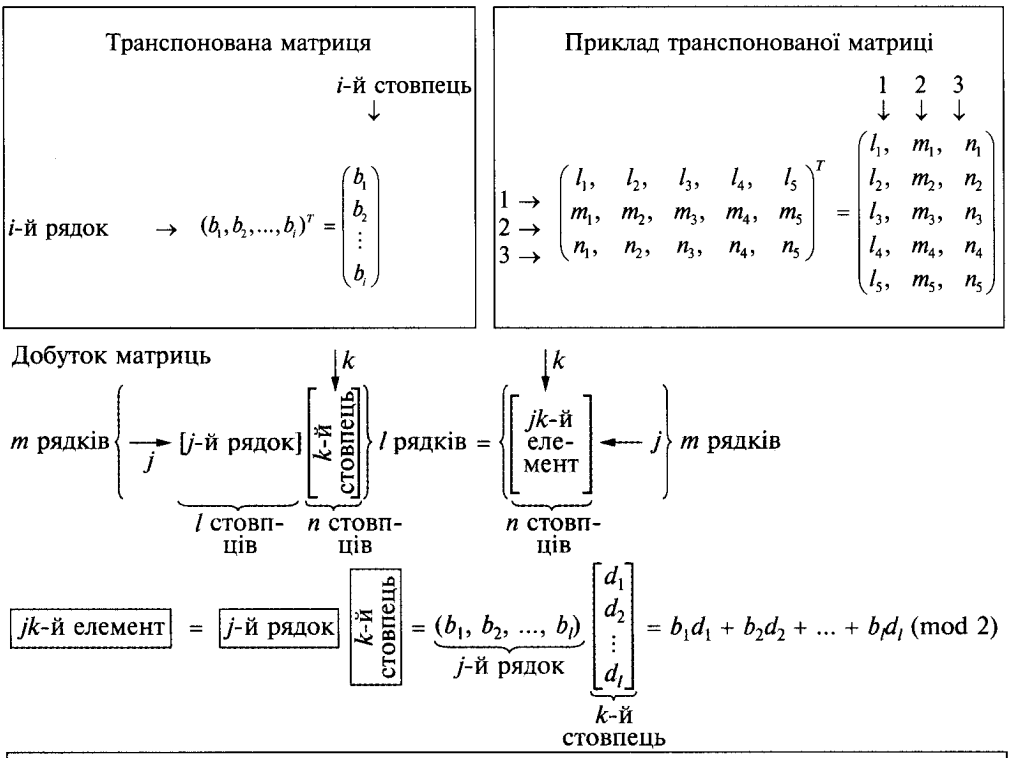
Тут I_4 і P відповідно:

$$I_4 = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$ — одинична матриця: 4 рядки на 4 стовпці;

$$P = \begin{pmatrix}a_1&a_2&a_3\\b_1&b_2&b_3\\c_1&c_2&c_3\\d_1&d_2&d_3\end{pmatrix}$$ — матриця розміром 4 рядки на 3 стовпці.

Матриця перевірки на парність H записується так:

$$H = \begin{pmatrix}k_1&l_1&m_1&n_1&1&0&0\\k_2&l_2&m_2&n_2&0&1&0\\k_3&l_3&m_3&n_3&0&0&1\end{pmatrix} = (Q, I_3). \qquad (148)$$



$$\begin{pmatrix} a_1, & a_2, & a_3, & a_4, & a_5 \\ b_1, & b_2, & b_3, & b_4, & b_5 \end{pmatrix} \begin{pmatrix} l_1, & m_1, & n_1 \\ l_2, & m_2, & n_2 \\ l_3, & m_3, & n_3 \\ l_4, & m_4, & n_4 \\ l_5, & m_5, & n_5 \end{pmatrix} = \begin{pmatrix} r_1, & r_2, & r_3 \\ s_1, & s_2, & s_3 \end{pmatrix}$$

$$r_1 = (a_1, a_2, a_3, a_4, a_5) \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{pmatrix} = \qquad s_1 = (b_1, b_2, b_3, b_4, b_5) \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{pmatrix} =$$

$= a_1l_1 + a_2l_2 + a_3l_3 + a_4l_4 + a_5l_5 \pmod 2$ $\quad = b_1l_1 + b_2l_2 + b_3l_3 + b_4l_4 + b_5l_5 \pmod 2$

$r_2 = a_1m_1 + a_2m_2 + a_3m_3 + a_4m_4 + a_5m_5 \pmod 2$ $\quad s_2 = b_1m_1 + b_2m_2 + b_3m_3 + b_4m_4 + b_5m_5 \pmod 2$

$r_3 = a_1n_1 + a_2n_2 + a_3n_3 + a_4n_4 + a_5n_5 \pmod 2$ $\quad s_2 = b_1n_1 + b_2n_2 + b_3n_3 + b_4n_4 + b_5n_5 \pmod 2$

Рис. 74. Транспонована матриця і добуток матриць

Тут I_3 і Q відповідно:

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ — одинична матриця: 3 рядки на 3 стовпці;

$$Q = \begin{pmatrix} k_1 & l_1 & m_1 & n_1 \\ k_2 & l_2 & m_2 & n_2 \\ k_3 & l_3 & m_3 & n_3 \end{pmatrix}$$ — матриця розміром 3 рядки на 4 стовпці.

Таким чином, породжувальна матриця G і матриця перевірки на парність H мають вигляд (147) і (148), а саме: $G = (I_4,P)$, $H = (Q,I_3)$.

Оскільки I_4 і I_3 — одиничні матриці, то їх форма визначена. Отже, якщо з'ясувати, які співвідношення пов'язують між собою матриці P і Q, то можна знайти, якими співвідношеннями пов'язані породжувальна матриця G і матриця перевірки на парність H. Визначимо співвідношення між матрицями P і Q за допомогою формули (144):

$$GH^T = (I_4,P)(Q,I_3)^T = (I_4,P)\begin{pmatrix} Q^T \\ I_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & a_1 & a_2 & a_3 \\ 0 & 1 & 0 & 0 & b_1 & b_2 & b_3 \\ 0 & 0 & 1 & 0 & c_1 & c_2 & c_3 \\ 0 & 0 & 0 & 1 & d_1 & d_2 & d_3 \end{pmatrix}\begin{pmatrix} k_1 & k_2 & k_3 \\ l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 0. \quad (149)$$

Тому

$$\text{(Перший рядок)} \left(\text{Перший стовпець}\right) = k_1 + a_1 = 0 \to k_1 = -a_1$$

$$\text{(Перший рядок)} \left(\text{Другий стовпець}\right) = k_2 + a_2 = 0 \to k_2 = -a_2$$

$$\text{(Перший рядок)} \left(\text{Третій стовпець}\right) = k_3 + a_3 = 0 \to k_3 = -a_3$$

Аналогічно

$$\left.\begin{matrix} l_1 + b_1 = 0 \\ l_2 + b_2 = 0 \\ l_3 + b_3 = 0 \end{matrix}\right\} \to \left\{\begin{matrix} l_1 = -b_1, & m_1 + c_1 = 0 \\ l_2 = -b_2, & m_2 + c_2 = 0 \\ l_3 = -b_3, & m_3 + c_3 = 0 \end{matrix}\right\} \to \left\{\begin{matrix} m_1 = -c_1, & n_1 + d_1 = 0 \\ m_2 = -c_2, & n_2 + d_2 = 0 \\ m_3 = -c_3, & n_3 + d_3 = 0 \end{matrix}\right\} \to \left\{\begin{matrix} n_1 = -d_1, \\ n_2 = -d_2, \\ n_3 = -d_3. \end{matrix}\right.$$

А саме:

$$Q^T = \begin{pmatrix} k_1 & k_2 & k_3 \\ l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{pmatrix} = \begin{pmatrix} -a_1 & -a_2 & -a_3 \\ -b_1 & -b_2 & -b_3 \\ -c_1 & -c_2 & -c_3 \\ -d_1 & -d_2 & -d_3 \end{pmatrix} = -\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \\ d_1 & d_2 & d_3 \end{pmatrix} = -P.$$

Елементи матриці — це елементи групи $GF(2)$, тому має місце співвідношення:

$$Q^T = P. \tag{150}$$

Отже,

$$H^T = (Q, I_3)^T = \begin{pmatrix} Q^T \\ I_3 \end{pmatrix} = \begin{pmatrix} P \\ I_3 \end{pmatrix} = \left(P^T, I_3\right)^T.$$

Тоді отримаємо

$$H = \left(P^T, I_3\right)^T. \tag{151}$$

А тепер визначимо матрицю перевірки на парність з породжувальної матриці G коду (7, 4) БЧХ (формула (146)) за формулами (147) і (151). Відповідно до формули (147) матриця P, отримана на підставі породжувальної матриці G (146), має такий вигляд:

$$P = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

Отже, можна визначити матрицю, транспоновану щодо матриці P:

$$P^T = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}^T = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

Тому, на підставі формули (151), матриця перевірки на парність H записується так:

$$H = \left(P^T, I_3\right) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Очевидно, що ця матриця абсолютно аналогічна поданій формулою (145) матриці перевірки на парність, яка отримана на підставі табл. 46.

Узагальнивши співвідношення між породжувальною матрицею G і матрицею перевірки на парність H (див. формули (147) і (151)), отримаємо такі співвідношення між породжувальною матрицею G і матрицею перевірки на парність H для лінійного коду (n, k) (рис. 75):

$$\left.\begin{aligned} G &= \left(I_k, P\right) \\ H &= \left(P^T, I_m\right) \end{aligned}\right\}. \tag{152}$$

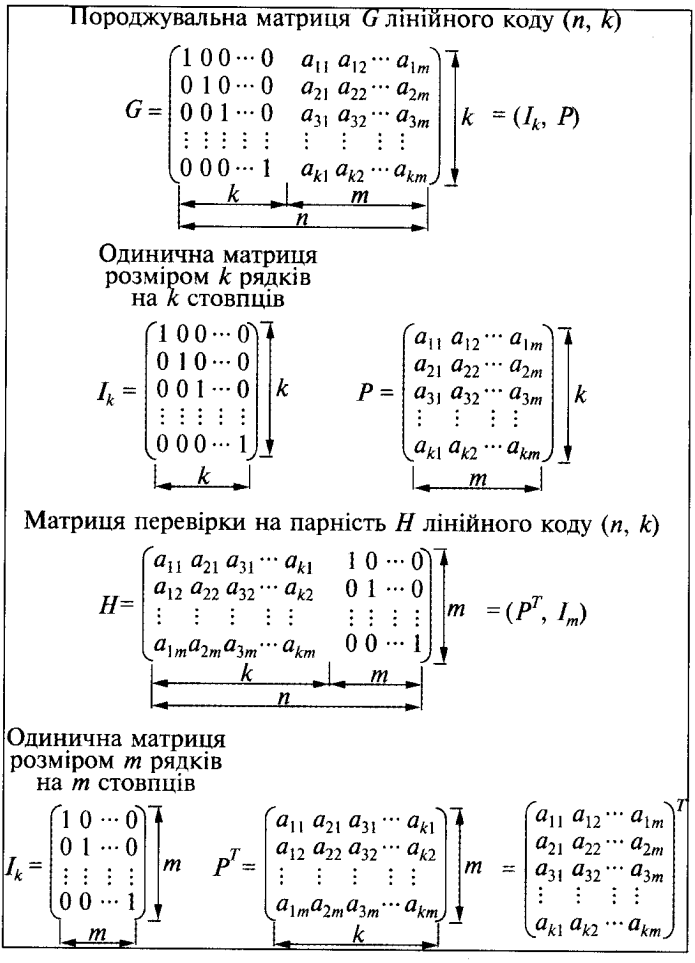


Рис. 75. Приклад породжувальної матриці та матриці перевірки на парність лінійного коду (n, k)

Тут

$$P = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{km} \end{pmatrix},$$

$$P^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{k1} \\ a_{12} & a_{22} & \cdots & a_{k2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{km} \end{pmatrix}$$

А тепер для коду (15, 7) БЧХ з використанням формули (152) визначимо матрицю перевірки на парність з породжувальної матриці.

Породжувальна матриця G для коду (15, 7) БЧХ задана за допомогою формули (142):

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} = (I_7, P).$$

Тут I_7 є одиничною матрицею розміром 7 рядків на 7 стовпців, а

$$P = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Матриця P^T, транспонована щодо матриці P, набуває вигляду

$$P^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix},$$

звідки випливає, що, виходячи з формули (152), матрицю перевірки на парність H для коду (15, 7) БЧХ можна подати так:

$$H = (P^T, I_8) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (153)$$

Раніше ця матриця перевірки на парність H для коду (15, 7) БЧХ була записана у вигляді формули (94), отриманої на підставі табл. 21:

$$H_S = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (154)$$

Очевидно, що подана у формулі (153) матриця перевірки на парність H, яку отримали з породжувальної матриці, відрізняється від записаної у формулі (154) матриці перевірки на парність H_S, одержаної раніше. Виявляється, що для однієї і тієї самої породжувальної матриці існують дві матриці перевірки на парність. Проте виникає питання: чи відрізняються насправді матриці перевірки на парність, задані формулами (153) і (154).

Розглянемо ще раз співвідношення між породжувальною матрицею і матрицею перевірки на парність.

Той факт, що співвідношення між породжувальною матрицею G і матрицею перевірки на парність H було отримано на підставі формули (144), означає, що для будь-яких i, j завжди виконується залежність

$$\begin{pmatrix} i\text{-й рядок породжувальної} \\ \text{матриці } G \end{pmatrix} \begin{pmatrix} j\text{-й рядок матриці} \\ \text{перевірки на парність } H \end{pmatrix}^T = 0. \qquad (155)$$

Залежність, що задана формулою (155), має виключно важливий сенс, а саме: вона означає, що якщо залишити в своєму незмінному вигляді породжувальну матрицю і розглядати тільки матрицю перевірки на парність, то співвідношення, яке задане формулою (144), виконуватиметься, навіть якщо будуть переставлені місцями довільно взяті рядки матриці перевірки на парність, або до елементів будь-якого довільно взятого рядка матриці будуть додаватися елементи іншого рядка. Наприклад, якщо до третього рядка матриці перевірки на парність H коду (7, 4) БЧХ (формула (145))

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \end{matrix}$$

додати перший і другий рядки, то отримаємо

Перший рядок	$\rightarrow$		1110100
Другий рядок	$\rightarrow$	+	0111010
Третій рядок	$\rightarrow$		1101001
Новий третій рядок	$\rightarrow$		0100111

Тому матрицю перевірки на парність можна записати так:

$$H_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2. \\ \end{matrix}$$

Крім цього, помінявши місцями перший і другий рядок, як показано нижче:

$$H_2 = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

одержимо нову матрицю перевірки на парність H_2. Безпосередньо з процесу перетворення випливає, що між новою матрицею перевірки на парність H_2, яка отримана шляхом підсумовування довільно взятого рядка матриці перевірки на парність H та іншого рядка і перестановки місцями будь-яких довільно взятих рядків та породжувальною матрицею G коду (7, 4) БЧХ (див. формула (146)), виконується співвідношення

$$GH_2^T = 0.$$

З викладеного вище маємо, що навіть якщо над матрицею перевірки на парність виконуватимуться наступні дві операції:

1) перестановка місцями довільно взятих рядків,

2) підсумовування довільно взятого рядка та іншого рядка,

то все одно буде отримана матриця перевірки на парність, еквівалентна вихідній, а саме: виконуватиметься співвідношення, яке задане формулою (144). Щодо породжувальної матриці, то і над нею, так само, як і над матрицею перевірки на парність, можна виконувати аналогічні операції.

Отже, якщо виконувати ці дві операції над заданою формулою (154) матрицею перевірки на парність H_S і перетворити її, отримавши матрицю перевірки на парність H, що задана формулою (153), то ці дві матриці перевірки на парність H і H_S, еквівалентні, і розрізняються тільки зовнішнім виглядом.

Процес перетворення матриці перевірки на парність H_S показано на рис. 76.

У ході перетворень, наведених на рис. 76, матриця перевірки на парність спочатку набуває вигляду

$$\begin{pmatrix} * & * & \cdots & * & 1 & 0 & 0 & 0 & 0 \\ * & * & \cdots & * & * & 1 & 0 & 0 & 0 \\ * & * & \cdots & * & * & * & 1 & 0 & 0 \\ * & * & \cdots & * & * & * & * & 1 & 0 \\ * & * & \cdots & * & * & * & * & * & 1 \end{pmatrix}.$$

$$H_s = \begin{pmatrix} 1&1&1&1&0&1&1&1&1&0&1&1&1&1&0 \\ 1&0&1&0&0&1&0&1&0&0&1&0&1&0&0 \\ 1&1&0&0&0&1&1&0&0&0&1&1&0&0&0 \\ 1&1&0&0&1&1&1&0&0&1&1&1&0&0&1 \\ 1&1&1&1&0&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&1&1&0&1&0&1&1&0&0&1&0&0 \\ 0&0&1&1&1&1&0&1&0&1&1&0&0&1&0 \\ 1&1&1&0&1&0&1&1&0&0&1&0&0&0&1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 6 \\ \leftarrow 7 \\ \leftarrow 8 \end{matrix}$$

1. Восьмий + четвертий = новий четвертий

1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 ← 8
+ 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 ← 4
0 0 1 0 0 1 0 1 0 1 0 1 0 0 0 ← новий 4

2. Шостий + другий = новий другий

0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 ← 6
+ 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 ← 2
1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 ← новий 2

3. П'ятий + шостий + сьомий + перший = новий перший

1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 ← 5
0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 ← 6
0 0 1 1 1 1 0 1 0 1 1 0 0 1 0 ← 7
+ 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 ← 1
0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 ← новий 1

4. П'ятий + третій = новий третій

1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 ← 5
+ 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 ← 3
0 0 1 1 0 0 1 1 1 0 1 0 0 0 0 ← новий 3

$$H_1 = \begin{pmatrix} 0&1&0&0&0&1&0&1&1&0&0&0&0&0&0 \\ 1&1&0&1&1&1&1&1&1&1&1&0&0&0&0 \\ 0&0&1&1&0&0&1&1&1&0&1&0&0&0&0 \\ 0&1&1&0&0&1&1&1&0&1&0&0&0&0&0 \\ 1&1&1&1&0&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&1&1&0&1&0&1&1&0&0&1&0&0 \\ 0&0&1&1&1&1&0&1&0&1&1&0&0&1&0 \\ 1&1&1&0&1&0&1&1&0&0&1&0&0&0&1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 6 \\ \leftarrow 7 \\ \leftarrow 8 \end{matrix}$$

5. Третій + четвертий + другий = новий другий

0 0 1 1 0 0 1 1 1 0 1 0 0 0 0 ← 3
0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 ← 4
+ 1 1 0 1 1 1 1 1 1 1 1 0 0 0 0 ← 2
1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 ← новий 2

6. Перший + третій = новий третій

0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 ← 1
+ 0 0 1 1 0 0 1 1 1 0 1 0 0 0 0 ← 3
0 1 1 1 0 1 1 0 0 0 1 0 0 0 0 ← новий 3

$$H_2 = \begin{pmatrix} 0&1&0&0&0&1&0&1&1&0&0&0&0&0&0 \\ 1&0&0&0&1&0&1&1&0&0&0&0&0&0&0 \\ 0&1&1&1&0&1&1&0&0&0&1&0&0&0&0 \\ 0&1&1&0&0&1&1&1&0&1&0&0&0&0&0 \\ 1&1&1&1&0&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&1&1&0&1&0&1&1&0&0&1&0&0 \\ 0&0&1&1&1&1&0&1&0&1&1&0&0&1&0 \\ 1&1&1&0&1&0&1&1&0&0&1&0&0&0&1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 6 \\ \leftarrow 7 \\ \leftarrow 8 \end{matrix}$$

Перший → новий перший
Другий → новий другий
Третій → новий третій
Четвертий → новий четвертий

$$H_3 = \begin{pmatrix} 1&0&0&0&1&0&1&1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&1&0&1&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&1&1&0&1&0&0&0&0&0 \\ 0&1&1&1&0&1&1&0&0&0&1&0&0&0&0 \\ 1&1&1&1&0&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&1&1&0&1&0&1&1&0&0&1&0&0 \\ 0&0&1&1&1&1&0&1&0&1&1&0&0&1&0 \\ 1&1&1&0&1&0&1&1&0&0&1&0&0&0&1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 6 \\ \leftarrow 7 \\ \leftarrow 8 \end{matrix}$$

7. Перший + другий = новий другий

```
  1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 ← 1
+ 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 ← 2
  1 1 0 0 1 1 1 0 1 0 0 0 0 0 0 ← новий 2
```

8. Перший + третій = новий третій

```
  1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 ← 1
+ 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 ← 3
  1 1 1 0 1 1 0 0 0 1 0 0 0 0 0 ← новий 3
```

9. Другий + п'ятий = новий п'ятий

```
  0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 ← 2
+ 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 ← 5
  1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 ← новий 5
```

10. Другий + третій + шостий = новий шостий

```
  0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 ← 2
  0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 ← 3
+ 0 1 1 1 1 0 1 0 1 1 0 0 1 0 0 ← 6
  0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 ← новий 6
```

11. Третій + четвертий + сьомий = новий сьомий

```
  0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 ← 3
  0 1 1 1 0 1 1 0 0 0 1 0 0 0 0 ← 4
+ 0 0 1 1 1 1 0 1 0 1 1 0 0 1 0 ← 7
  0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 ← новий 7
```

12. Перший + четвертий + восьмий = новий восьмий

$$
\begin{array}{l}
1\,0\,0\,0\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0 \leftarrow 1 \\
0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0 \leftarrow 4 \\
+\,\underline{1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,1 \leftarrow 8} \\
0\,0\,0\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1 \leftarrow \text{новий } 8
\end{array}
$$

$$
H_4 = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} = H \text{ (формула (153))}
$$

Рис. 76. Схема перетворення матриці перевірки на парність

Далі виконуються перетворення з метою звести матрицю до форми, що дозволяє досягти поставленої мети:

$$
\begin{pmatrix}
* & \cdots & * & 1 & 0 & 0 & 0 & 0 \\
* & \cdots & * & 0 & 1 & 0 & 0 & 0 \\
* & \cdots & * & 0 & 0 & 1 & 0 & 0 \\
* & \cdots & * & 0 & 0 & 0 & 1 & 0 \\
* & \cdots & * & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.
$$

Як видно з рис. 76, матриці перевірки на парність H і H_S еквівалентні, тому виникає припущення, що в процесі роботи можна використовувати таке подання матриці перевірки на парність, яке відповідає поставленій меті.

Напевно, відповідний аналог підібрати в повсякденні нелегко. Тому виникає питання, як використовувати різні подання матриці перевірки на парність. Наприклад, якщо якийсь артист грає в п'єсі роль з якоїсь епохи, то він відповідно до стилю цієї епохи змінює зачіску, одяг та інші аксесуари, а також поведінку і виходить на сцену, що обладнана певними декораціями. Так само і в кодуванні необхідно враховувати, що певні подання матриці перевірки на парність відповідають певним застосуванням.

15.3.3. Схема ділення на породжувальний многочлен і матриця перевірки на парність

Розглянемо співвідношення між використовуваними для циклічних кодів схемою ділення на породжувальний многочлен, що дозволяє визначити залишок, та матрицею перевірки на парність, яка призначена для визначення синдрому.

Якщо вдасться з'ясувати співвідношення між схемою ділення на породжувальний многочлен і матрицею перевірки на парність, то будуть зрозу-

мілими співвідношення між залишком, отриманим унаслідок ділення многочлена помилок (конфігурації помилок) на породжувальний многочлен, і синдромом, що одержали на підставі матриці перевірки на парність для випадку коду (7, 4) БЧХ.

Позначимо конфігурацію помилок $e = (e_6, e_5, e_4, e_3, e_2, e_1, e_0)$, де серед бітів e_6—e_0 не може бути більше ніж один одиничний (помилковий) біт.

При цьому відображення конфігурації помилок у вигляді многочлена, що є многочленом помилок $E(x)$, має вигляд

$$E(x) = e_6x^6 + e_5x^5 + e_4x^4 + e_3x^3 + e_2x^2 + e_1x^1 + e_0,$$

де серед бітів e_6—e_0 не може бути більше ніж один одиничний (помилковий) біт.

Залишок від ділення многочлена помилок $E(x)$ на породжувальний многочлен $G(x)$ коду (7, 4) БЧХ: $G(x) = x^3 + x + 1$, згідно з розрахунками запишеться так:

$$R(x) = E(x) \bmod G(x) =$$

$$= (e_6 + e_5 + e_4 + e_2) \cdot x^2 + (e_5 + e_4 + e_3 + e_1) \cdot x^1 + (e_6 + e_5 + e_3 + e_0).$$

У тому разі, коли в схему ділення на породжувальний многочлен $G(x)$ вводиться конфігурація помилок $E(x)$, залишок утворюється на регістрах схеми ділення до того моменту, коли повністю закінчується введення всієї конфігурації помилок. Визначивши залишок для таких випадків помилок у всіх бітах, починаючи від першого і закінчуючи сьомим, отримаємо результати, наведені на рис. 77.

На рис. 78 продемонстрована конфігурація регістрів $RG0$, $RG1$, $RG2$, що відповідають залишкам для відповідних бітів помилок, розташованих послідовно: від помилки в першому біті до помилки в сьомому біті. Як бачимо, порядок розташування регістрів, які відповідають залишкам, повністю аналогічний матриці перевірки на парність, наведеній у вигляді (145).

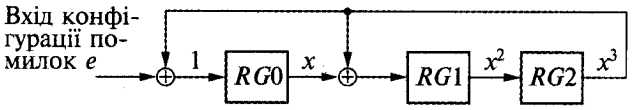
Мабуть, той факт, що впорядковані залишки бітів помилок, побудовані на регістрах схеми ділення на породжувальний многочлен, збігаються з матрицею перевірки на парність, свідчить про те, що залишки, отримані унаслідок введення конфігурації помилок у схему ділення на породжувальний многочлен і синдром, отриманий унаслідок обчислення з використанням конфігурації помилок і матриці перевірки на парність, повністю збігаються (рис. 79).

Перевіримо це припущення. Згідно з табл. 46, матриця перевірки на парність коду (7, 4) БЧХ має вигляд

$$H = (a^6, a^5, a^4, a^3, a^2, a^1, a^0) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

тому синдром S можна визначити так:

$$S = He^T =$$



Ділення на породжувальний многочлен $G(x) = x^3 + x + 1$

Залишок	Вихід регістра		
	*RG*0	*RG*1	*RG*2
До моменту, коли повністю закінчується введення конфігурації помилок	$e_6 + e_5 + e_3 + e_0$	$e_5 + e_4 + e_3 + e_1$	$e_6 + e_5 + e_4 + e_2$
За відсутності помилок (e_6—e_0)=0	0	0	0
За наявності помилки у першому біті ($e_1 = 1$, інші дорівнюють нулю)	1	0	1
За наявності помилки у другому біті ($e_2 = 1$, інші дорівнюють нулю)	1	1	1
За наявності помилки в третьому біті ($e_3 = 1$, інші дорівнюють нулю)	0	1	1
За наявності помилки в четвертому біті ($e_4 = 1$, інші дорівнюють нулю)	1	1	0
За наявності помилки в п'ятому біті ($e_5 = 1$, інші дорівнюють нулю)	0	0	1
За наявності помилки в шостому біті ($e_6 = 1$, інші дорівнюють нулю)	0	1	0
За наявності помилки в сьомому біті ($e_7 = 1$, інші дорівнюють нулю)	1	0	0

Рис. 77. Конфігурація залишків від ділення для всіх випадків, коли помилки відбулися в кожному з бітів

$$= (a^6, a^5, a^4, a^3, a^2, a^1, a^0)\begin{pmatrix} e_6 \\ e_5 \\ e_4 \\ e_3 \\ e_2 \\ e_1 \\ e_0 \end{pmatrix} = e_6 a^6 + e_5 a^5 + e_4 a^4 + e_3 a^3 + e_2 a^2 + e_1 a^1 + e_0 . \quad (156)$$

Виходячи з формули (156), у разі розрахунків синдрому для випадків, коли помилки з'явилися у всіх можливих бітах, отримаємо наступне:

- у разі відсутності помилок (e_0—$e_0 = 0$)

$$S = 0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix};$$

• у разі помилки в першому біті ($e_6 = 1$, інші дорівнюють нулю)

$$S = a^6 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix};$$

• у разі помилки в другому біті ($e_5 = 1$, інші дорівнюють нулю)

$$S = a^5 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix};$$

.
.
.

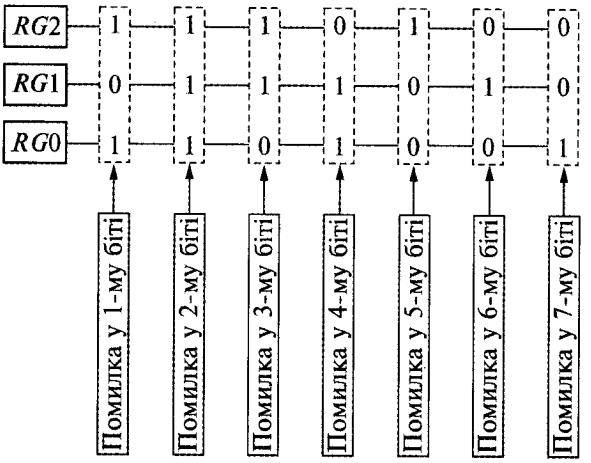


Рис. 78. Конфігурація регістрів

• у разі помилки в сьомому біті ($e_0 = 1$, інші дорівнюють нулю)

$$S = a^0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Очевидно, що це подання синдрому у вигляді вектора-стовпця і використання на рис. 78 конфігурацій регістрів, в якому ті, що відповідають залишкам, розташовані у вигляді вертикальних стовпців, повністю збігаються один з одним. А саме: залишки, отримані на основі синдрому S і з використанням схеми ділення на породжувальний многочлен, є однаковими, не дивлячись на те, що одержані з використанням різних процедур.

Як було вже показано (див. рис. 64), при декодуванні лінійного коду розрахунковим шляхом визначається синдром, і помилки виправляються унаслідок підсумовування з кодовою бітовою послідовністю конфігурації помилок, яка відповідає цьому синдрому, що приймається. Синдром S можна отримати без проведення безпосередніх обчислень, за допомогою схемного втілення обчислення матриці перевірки на парність і кодової бітової послідовності, що приймається, з використанням елемента додавання за модулем 2. Проте

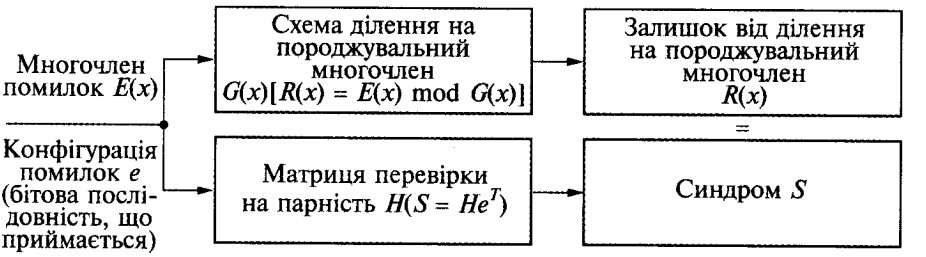


Рис. 79. Схема визначення залишку і синдрому

в даному контексті розрахунки виконуються на основі визначення синдрому шляхом безпосереднього обчислення.

У разі, коли циклічний код декодується за принципами лінійного коду, синдром виражається у вигляді залишку від ділення на породжувальний многочлен. Виявляється, що цей залишок слід використовувати як синдром. Тому замість того, щоб повністю закінчивши введення кодової бітової послідовності, що приймається, визначити синдром на основі обчислення з використанням матриці перевірки на парність, можна, ввівши кодову бітову послідовність, що приймається, в схему ділення на породжувальний многочлен, визначити синдром як залишок у той момент, коли введення такої кодової бітової послідовності повністю завершилося.

Розглянемо випадок коду (15, 7) БЧХ. Число кодових бітів у ньому 15, а за своєю здібністю до виправлення помилок він є кодом з виправленням одиничних помилок. Для опису конфігурації помилок e може використовуватися такий вираз: $e = (e_{14}, e_{13}, e_{12}, e_{11}, e_{10}, e_9, e_8, e_7, e_6, e_5, e_4, e_3, e_2, e_1, e_0)$, де число одиничних (помилкових) бітів серед бітів e_{14}—e_0 не може перевищувати двох.

Запишемо многочлен помилок $E(x)$, який можна отримати унаслідок подання цієї конфігурації помилок:

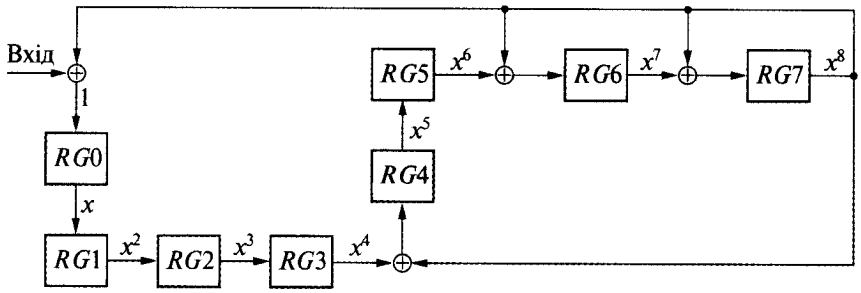
$$E(x) = e_{14}x^{14} + e_{13}x^{13} + e_{12}x^{12} + e_{11}x^{11} + e_{10}x^{10} + e_9x^9 +$$
$$+ e_8x^8 + e_7x^7 + e_6x^6 + e_5x^5 + e_4x^4 + e_3x^3 + e_2x^2 + e_1x^1 + e_0.$$

Залишок $R(x)$ від ділення многочлена помилок $E(x)$ на породжувальний многочлен $G(x)$ коду (15, 7) БЧХ, що подається виразом $G(x) = x^8 + x^7 + x^6 + x^4 + 1$, має вигляд

$$R(x) = E(x) \bmod G(x) = (e_{14} + e_{10} + e_8 + e_7)x^7 + (e_{14} + e_{13} + e_{10} + e_9 + e_8 + e_6)x^6 +$$
$$+ (e_{14} + e_{13} + e_{12} + e_{10} + e_9 + e_5)x^5 + (e_{13} + e_{12} + e_{11} + e_{10} + e_9 + e_8 + e_4)x^4 +$$
$$+ (e_{14} + e_{12} + e_{11} + e_3)x^3 + (e_{13} + e_{11} + e_{10} + e_2)x^2 +$$
$$+ (e_{12} + e_{10} + e_9 + e_1)x^1 + (e_{11} + e_9 + e_8 + e_0).$$

Визначивши конфігурацію помилок для цього залишку $R(x)$ шляхом ділення на породжувальний многочлен, отримаємо результати, наведені на рис. 80, для випадку, коли в схему ділення на такий многочлен введена конфігурація помилок на 15-му інтервалі синхронізації. Це той момент часу, коли повністю закінчується введення помилок (кодової бітової послідовності, що приймається), а на виходах регістрів схеми ділення з'являється залишок, що відповідає синдрому. Знайшовши цей залишок для випадків, коли помилка відбулася в тому або іншому біті, отримаємо результати, подані на рис. 81. Крім того, якщо (див. рис. 80) розташувати регістри, що відповідають залишкам для всіх випадків, коли помилка відбулася в тому або іншому біті, по порядку, починаючи від помилки в першому біті, то одержимо результати, наведені на рис. 82.

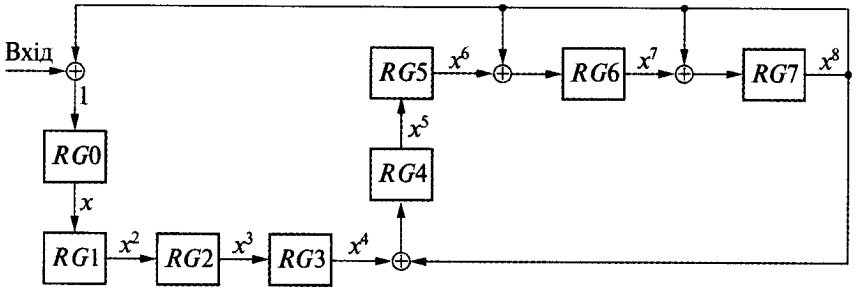
Матрицю перевірки на парність для коду (15, 7) БЧХ можна отримати з формули (153). Порівнявши цю матрицю перевірки на парність з впорядкованим розташуванням регістрів (див. рис. 82), де формуються залишки, можна побачити, що вони повністю збігаються. Отже, можна зробити висновок,



Синхронізація	Вхід	Вихід регістра							
		*RG*0	*RG*1	*RG*2	*RG*3	*RG*4	*RG*5	*RG*6	*RG*7
0	$x^{14} \to e_{14}$	0	0	0	0	0	0	0	0
1	$x^{13} \to e_{13}$	e_{14}	0	0	0	0	0	0	0
2	$x^{12} \to e_{12}$	e_{13}	e_{14}	0	0	0	0	0	0
3	$x^{11} \to e_{11}$	e_{12}	e_{13}	e_{14}	0	0	0	0	0
4	$x^{10} \to e_{10}$	e_{11}	e_{12}	e_{13}	e_{14}	0	0	0	0
5	$x^{9} \to e_{9}$	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	0	0	0
6	$x^{8} \to e_{8}$	e_{9}	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	0	0
7	$x^{7} \to e_{7}$	e_{8}	e_{9}	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	0
8	$x^{6} \to e_{6}$	e_{7}	e_{8}	e_{9}	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}
9	$x^{5} \to e_{5}$	$e_{14}+e_{6}$	e_{7}	e_{8}	e_{9}	$e_{14}+e_{10}$	e_{11}	$e_{14}+e_{12}$	$e_{14}+e_{12}$
10	$x^{4} \to e_{4}$	$e_{14}+e_{13}+e_{5}$	$e_{14}+e_{6}$	e_{7}	e_{8}	$e_{14}+e_{13}+e_{8}$	$e_{14}+e_{10}$	$e_{14}+e_{13}+e_{11}$	$e_{13}+e_{12}$
11	$x^{3} \to e_{3}$	$e_{13}+e_{12}+e_{4}$	$e_{14}+e_{13}+e_{5}$	$e_{14}+e_{6}$	e_{7}	$e_{13}+e_{12}+e_{8}$	$e_{14}+e_{13}+e_{8}$	$e_{14}+e_{13}+e_{12}+e_{10}$	$e_{14}+e_{12}+e_{11}$
12	$x^{2} \to e_{2}$	$e_{14}+e_{12}+e_{11}+e_{3}$	$e_{13}+e_{12}+e_{4}$	$e_{14}+e_{13}+e_{5}$	$e_{14}+e_{6}$	$e_{14}+e_{12}+e_{11}+e_{7}$	$e_{13}+e_{12}+e_{8}$	$e_{13}+e_{12}+e_{11}+e_{9}$	$e_{13}+e_{11}+e_{10}$
13	$x^{1} \to e_{1}$	$e_{13}+e_{11}+e_{10}+e_{2}$	$e_{14}+e_{12}+e_{11}+e_{3}$	$e_{13}+e_{12}+e_{4}$	$e_{14}+e_{13}+e_{5}$	$e_{14}+e_{13}+e_{11}+e_{10}+e_{6}$	$e_{14}+e_{12}+e_{11}+e_{7}$	$e_{12}+e_{11}+e_{10}+e_{8}$	$e_{12}+e_{10}+e_{9}$
14	$x^{0} \to e_{0}$	$e_{12}+e_{10}+e_{9}+e_{1}$	$e_{13}+e_{11}+e_{10}+e_{2}$	$e_{14}+e_{12}+e_{11}+e_{3}$	$e_{13}+e_{12}+e_{4}$	$e_{14}+e_{13}+e_{12}+e_{10}+e_{8}+e_{5}$	$e_{14}+e_{13}+e_{11}+e_{10}+e_{6}$	$e_{14}+e_{11}+e_{10}+e_{9}+e_{7}$	$e_{11}+e_{9}+e_{8}$
15	0	$e_{11}+e_{9}+e_{8}+e_{0}$	$e_{12}+e_{10}+e_{9}+e_{1}$	$e_{13}+e_{11}+e_{10}+e_{2}$	$e_{14}+e_{12}+e_{11}+e_{3}$	$e_{13}+e_{12}+e_{11}+e_{9}+e_{8}+e_{4}$	$e_{14}+e_{13}+e_{12}+e_{10}+e_{8}+e_{5}$	$e_{14}+e_{13}+e_{10}+e_{9}+e_{8}+e_{6}$	$e_{14}+e_{10}+e_{8}+e_{7}$
		(x^0)	(x^1)	(x^2)	(x^3)	(x^4)	(x^5)	(x^6)	(x^7)

Залишки

Рис. 80. Схема ділення на породжувальний многочлен $x^8+x^7+x^6+x^4+1$ і залишок



Залишок конфігурацій помилок	Вихід регістра							
	*RG*0	*RG*1	*RG*2	*RG*3	*RG*4	*RG*5	*RG*6	*RG*7
	$e_{11}+e_9+e_8+e_0$	$e_{12}+e_{10}+e_9+e_1$	$e_{13}+e_{11}+e_{10}+e_2$	$e_{14}+e_{12}+e_{11}+e_3$	$e_{13}+e_{12}+e_{11}+e_9+e_8+e_4$	$e_{14}+e_{13}+e_{12}+e_{10}+e_9+e_8$	$e_{14}+e_{13}+e_{10}+e_9+e_8+e_6$	$e_{14}+e_{10}+e_8+e_7$
За відсутності помилок ($e_{14} - e_0 = 0$)	0	0	0	0	0	0	0	0
За наявності помилки в першому біті ($e_{14} = 1$, інші дорівнюють нулю)	0	0	0	1	0	1	1	1
За наявності помилки в другому біті ($e_{13} = 1$,інші дорівнюють нулю)	0	0	1	0	1	1	1	0
За наявності помилки в третьому біті ($e_{12} = 1$, інші дорівнюють нулю)	0	1	0	1	1	1	0	0
За наявності помилки в четвертому біті ($e_{11} = 1$, інші дорівнюють нулю)	1	0	1	1	1	0	0	0
За наявності помилки в п'ятому біті ($e_{10} = 1$, інші дорівнюють нулю)	0	1	1	0	0	1	1	1
За наявності помилки в шостому біті ($e_9 = 1$, інші дорівнюють нулю)	1	1	0	0	1	1	1	0
За наявності помилки в сьомому біті ($e_8 = 1$, інші дорівнюють нулю)	1	0	0	0	1	0	1	1
За наявності помилки у восьмому біті ($e_7 = 1$, інші дорівнюють нулю)	0	0	0	0	0	0	0	1

Залишок конфігурацій помилок	Вихід регістра							
	*RG*0	*RG*1	*RG*2	*RG*3	*RG*4	*RG*5	*RG*6	*RG*7
	$e_{11}+e_9+e_8+e_0$	$e_{12}+e_{10}+e_9+e_1$	$e_{13}+e_{11}+e_{10}+e_2$	$e_{14}+e_{12}+e_{11}+e_3$	$e_{13}+e_{12}+e_{11}+e_9+e_8+e_4$	$e_{14}+e_{13}+e_{12}+e_{10}+e_9+e_8$	$e_{14}+e_{13}+e_{10}+e_9+e_8+e_6$	$e_{14}+e_{10}+e_8+e_7$
За наявності помилки в дев'ятому біті ($e_6 = 1$, інші дорівнюють нулю)	0	0	0	0	0	0	1	0
За наявності помилки в десятому біті ($e_5 = 1$, інші дорівнюють нулю)	0	0	0	0	0	1	0	0
За наявності помилки в одинадцятому біті ($e_4 = 1$, інші дорівнюють нулю)	0	0	0	0	1	0	0	0
За наявності помилки в дванадцятому біті ($e_3 = 1$, інші дорівнюють нулю)	0	0	0	1	0	0	0	0
За наявності помилки в тринадцятому біті ($e_2 = 1$, інші дорівнюють нулю)	0	0	1	0	0	0	0	0
За наявності помилки в чотирнадцятому біті ($e_1 = 1$, інші дорівнюють нулю)	0	1	0	0	0	0	0	0
За наявності помилки в п'ятнадцятому біті ($e_0 = 1$, інші дорівнюють нулю)	1	0	0	0	0	0	0	0

Рис. 81. Конфігурація залишків у схемі ділення для всіх випадків помилок у різних бітах

що і для цього коду синдром і залишки від ділення на породжувальний многочлен збігаються.

За своєю спроможністю до виправлення помилок код (15, 7) БЧХ — це код з виправленням подвійних помилок. Виникає питання, чи є однаковими залишки і синдром, якщо помилка відбулася в двох бітах у конфігурації помилок *e*. Виходячи з того, що у разі одиничної помилки залишки і синдром можна подати у вигляді суми помилкових бітів відповідних залишків або суми синдромів, можна зробити висновок, що результуючі залишки і синдром збігаються. Як приклад розглянемо залишки і синдром для випадку, коли помилка відбулася в четвертому і дев'ятому бітах. При цьому конфігурація помилок приймає вигляд: $e_{4,9}$ = (000100001000000). Ця конфігурація помилок є випадком, коли $e_{11} = 1$, $e_6 = 1$, решта бітів дорівнює нулю.

Синдром S приймає вигляд:

$$S = He_{4,9}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} (000100001000000)^T =$$

$$= \begin{pmatrix} 0 & + & 0 \\ 0 & + & 1 \\ 0 & + & 0 \\ 1 & + & 0 \\ 1 & + & 0 \\ 1 & + & 0 \\ 0 & + & 0 \\ 1 & + & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \leftarrow \text{Сума 4-го та 9-го стовпців } H$$

4-й стовпець H 9-й стовпець H

Показаний у складі синдрому четвертий стовпець матриці H — синдром для випадку помилки в четвертому біті. Крім того, дев'ятий стовпець матриці H є синдром для випадку помилки в дев'ятому біті. Отже, дійшли висновку, що синдром для випадку, коли помилка відбулася в четвертому і дев'ятому бітах, є сумою відповідних синдромів. Крім того, для залишків (див. рис. 82) випадок, коли помилка виникла в четвертому і дев'ятому бітах, можна подати як на рис. 83, *б*. Як бачимо, залишок — сума залишків для тих випадків, коли помилки відбулися у відповідних бітах (рис. 83, *в*).

Отже, у разі подвійних помилок і залишок, і синдром можна зобразити як суму залишків і синдромів для відповідних одиничних помилок, тому очевидно, що залишок і синдром збігатимуться.

Виходячи з наведеного вище, можна зробити висновок, що між поданням у вигляді циклічного коду, базисним поданням для циклічного коду і поданням у вигляді лінійного коду існує відповідність (див. табл. 44). А саме, у разі лінійного коду при кодуванні з використанням як породжувального многочлена, так і породжувальної матриці можна побудувати одне і те саме слово. Крім того, той факт, що залишок, отриманий у разі циклічного коду з використанням схеми ділення на породжувальний многочлен, і синдром, отриманий у разі лінійного коду з використанням матриці перевірки на парність, збігаються, свідчить, що при декодуванні синдром можна визначити без безпосередніх обчислень.

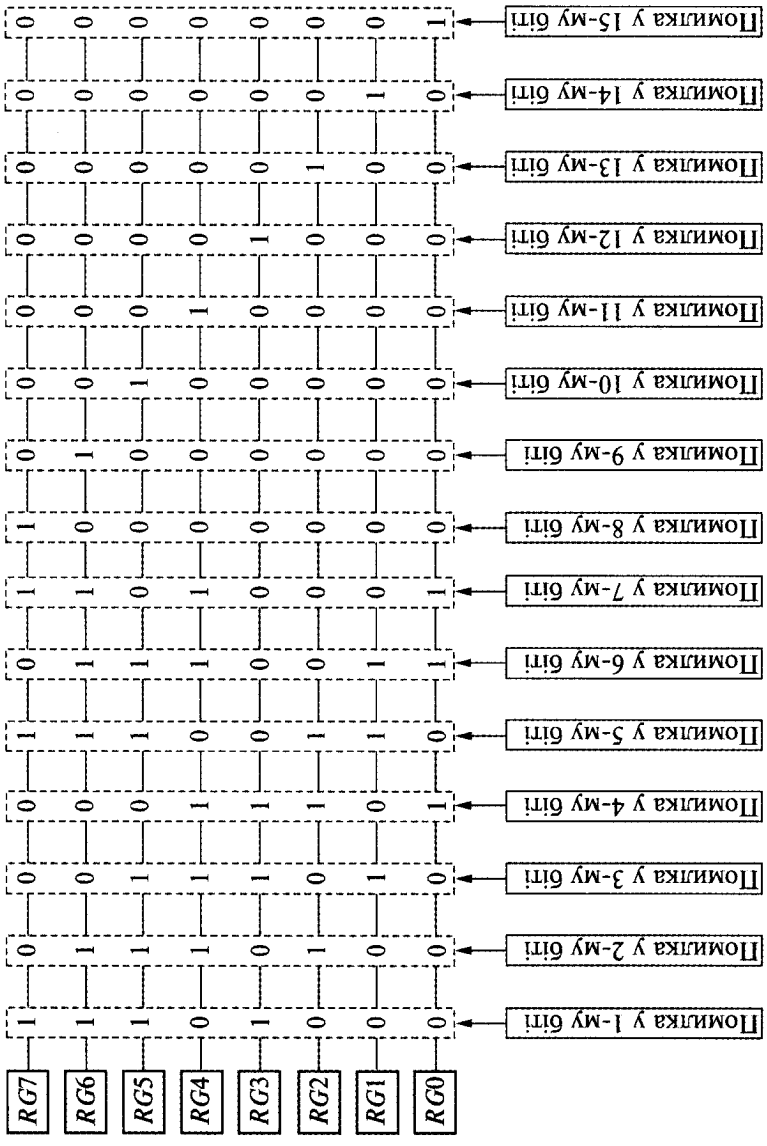
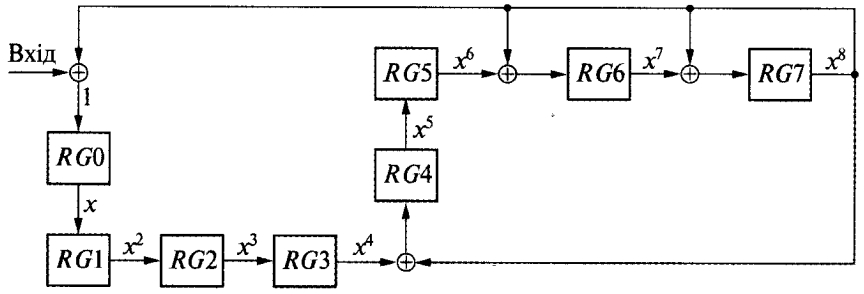


Рис. 82. Приклад визначення помилки в кожному з бітів і регістри, яким відповідають залишки



а

Залишок від конфігурації помилок	Вихід регістра							
	*RG*0	*RG*1	*RG*2	*RG*3	*RG*4	*RG*5	*RG*6	*RG*7
	$e_{11} + e_9 + e_8 + e_0$	$e_{12} + e_{10} + e_9 + e_1$	$e_{13} + e_{11} + e_{10} + e_2$	$e_{14} + e_{12} + e_{11} + e_3$	$e_{13} + e_{12} + e_{11} + e_9 + e_8 + e_4$	$e_{14} + e_{13} + e_{12} + e_{10} + e_9 + e_5$	$e_{14} + e_{13} + e_{10} + e_9 + e_8 + e_6$	$e_{14} + e_{10} + e_8 + e_7$
При подвійній помилці: в четвертому і дев'ятому бітах ($e_{11} = 11$, $e_6 = 1$), інші біти дорівнюють нулю	1	0	1	1	1	0	1	0

б

Залишок від конфігурації помилок	Вихід регістра							
	*RG*0	*RG*1	*RG*2	*RG*3	*RG*4	*RG*5	*RG*6	*RG*7
	$e_{11} + e_9 + e_8 + e_0$	$e_{12} + e_{10} + e_9 + e_1$	$e_{13} + e_{11} + e_{10} + e_2$	$e_{14} + e_{12} + e_{11} + e_3$	$e_{13} + e_{12} + e_{11} + e_9 + e_8 + e_4$	$e_{14} + e_{13} + e_{12} + e_{10} + e_9 + e_5$	$e_{14} + e_{13} + e_{10} + e_9 + e_8 + e_6$	$e_{14} + e_{10} + e_8 + e_7$
При помилці в четвертому біті, інші біти дорівнюють нулю	1	0	1	1	1	0	0	0
При помилці в дев'ятому біті, інші біти дорівнюють нулю	0	0	0	0	0	0	1	0
При одиничній помилці в четвертому біті і одиничній помилці в дев'ятому біті та сума залишків	1	0	1	1	1	0	1	0

в

Рис. 83. Конфігурації залишків у разі подвійних помилок:
а — схема ділення на породжувальний многочлен $x^8 + x^7 + x^6 + x^4 + 1$; *б* — залишок при подвійній помилці в четвертому і дев'ятому бітах; *в* — залишок при одиничній помилці в четвертому і дев'ятому бітах та сума залишків

ВИСНОВКИ

Історія кодування розпочалася публікацією в 1948 р. статті Клода Шеннона, в якій він показав залежність між інформаційною ємністю каналу зв'язку, смугою пропускання та відношенням сигнал/шум. Зі зростанням значення цього відношення і ширини смуги пропускання збільшується інформаційна ємність (пропускна спроможність) каналу. З теорії Шеннона випливає такий важливий висновок: побудова дуже якісних каналів є марнотратством; економічно вигідно використовувати кодування. Фактично у праці Шеннона стверджується, що потужність сигналу, шум у каналі і смуга частот обмежують лише швидкість передачі, а не її точність. Проте, як знайти відповідні коди, він не вказав, а лише довів їх існування.

У 50-ті роки XX ст. багато зусиль було витрачено на спроби побудувати в явному вигляді класи кодів, за яких вірогідність помилки була як завгодно малою, але результати були мізерними. Дослідники кодів проводили дослідження за двома основними напрямами. Перший напрям мав суто алгебраїчний характер, — розглядалися переважно блокові коди. Перші з них були введені в 1950 р., коли Хеммінг описав клас блокових кодів, що виправляють одиничні помилки. Коди Хеммінга були дуже слабкі порівняно з обіцяними Шенноном набагато сильнішими кодами. Кращого класу кодів до кінця 50-х років XX ст. не було створено. Протягом цього періоду без якої-небудь загальної теорії було побудовано багато кодів з малою довжиною блока. У 1959 р. Хоквінгем, а у 1960 р. Боуз і Рой-Чоудхурі знайшли великий клас кодів, що виправляють кратні помилки (коди БЧХ). Хоча ці коди і залишаються серед найважливіших класів кодів, проте загальна теорія блокових кодів, які контролюють помилки, з тих пір значно розвинулася. Відкриття кодів БЧХ зумовило пошук практичних методів побудови жорстких або м'яких реалізацій кодерів і декодерів.

Другий напрям досліджень мав швидше ймовірнісний характер. Початкові дослідження були пов'язані з оцінками вірогідності помилки для кращих сімей блокових кодів, хоча ці кращі коди не були відомі. З цими дослідженнями пов'язані спроби зрозуміти кодування і декодування з імовірнісної точки зору, і ці спроби зумовили появу послідовного декодування. Процедура корекції помилок припускає виявлення помилки і визначення її місцезнаходження. Після вирішення цих двох завдань виправлення тривіальне — потрібно інвертувати значення помилкового біта.

Для надійної передачі кодів було запропоновано два основні методи:

1) додати в передаваний блок даних декілька "зайвих" бітів так, щоб, аналізуючи отриманий блок, можна було визначити, чи є в переданому блоці помилки, чи ні. Це так звані *коди з виявленням помилок*;

2) внести надмірність настільки, щоб, аналізуючи отримані дані, можна було не тільки помітити помилки, а й вказати, де саме вони виникли. Це *коди, що виправляють помилки.*

Такий поділ умовний. Більш загальний варіант — це коди, що виявляють k помилок і що виправляють l помилок, де $l \leq k$.

У системах зв'язку можливі декілька стратегій боротьби з помилками:

- виявлення помилок у блоках даних і *автоматичний запит повторної передачі* пошкоджених блоків. Цей підхід застосовується в основному на канальному і транспортному рівнях;
- виявлення помилок у блоках даних і відкидання пошкоджених блоків. Такий підхід іноді застосовується в системах потокового мультимедіа, де важлива затримка передачі і немає часу на повторну передачу;
- виправлення помилок застосовується на фізичному рівні.

СПИСОК ЛІТЕРАТУРИ

1. *Блейхут Р.* Быстрые алгоритмы цифровой обработки сигналов. — М.: Мир, 1989. — 448 с.

2. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки: Пер. с англ. — М.: Мир, 1986. — 576 с.

3. *Берлекэмп Э.* Алгебраическая теория кодирования. — М.: Мир, 1971. — 478 с.

4. *Винберг Э.Б.* Курс алгебры. — М.: Факториал Прес, 2002. — 544 с.

5. *Галлагер Р.* Теория информации и надежная связь. — М.: Сов. радио, 1974.

6. *Глинченко А.С.* Цифровая обработка сигналов. В 2 ч. — Красноярск: Изд-во КГТУ, 2001. — 383 с.

7. *Гольденберг Л. М. и др.* Цифровая обработка сигналов. Справочник. — М.: Радио и связь, 1985. — 312 с.

8. *Даджион Д., Мерсеро Р.* Цифровая обработка многомерных сигналов. — М.: Мир, 1988. — 488 с.

9. *Зарисский О., Самюэль П.* Коммутативная алгебра: Пер. с англ.: В 2 т. — М.: Радио и связь, 1987. — Т. 1. — 287 с.; Т. 2. — 328 с.

10. *Злотник Б.М.* Помехоустойчивые коды в системах связи. — М.: Радио и связь, 1989. — 232 с.

11. *Касами Т., Токура Н., Ивадари Е., Инагаки Я.* Теория кодирования: Пер. с япон. — М.: Мир,1978. — 576 с.

12. *Кларк-мл. Дж., Кейн Дж.* Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ. — М.: Радио и связь, 1987. — 392 с.

13. *Коржик В.И., Финк Л.М.* Помехоустойчивое кодирование дискретных сообщений в каналах со случайной структурой. — М.: Связь, 1975. — 272 с.

14. *Мак-Вильямс Ф.Дж., Слоэн Н.Дж.А.* Теория кодов, исправляющих ошибки. — М.: Связь, 1979. — 331 с.

15. *Макс Ж.* Методы и техника обработки сигналов при физических измерениях: В 2 т. — М.: Мир, 1983. — Т. 1. — 234 с.; Т. 2. — 317 с.

16. *Марпл-мл. С. Л.* Цифровой спектральный анализ и его приложения. — М.: Мир, 1990. — 584 с.

17. *Морелос-Сарагоса Р.* Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. — М.: Техносфера, 2005. — 483 с.

18. *Муттер В.М.* Основы помехоустойчивой телепередачи информации. — Л.: Энергоатомиздат. Ленингр. отделение, 1990. — 288 с.

19. *Оппенгейм А., Шафер Р.* Цифровая обработка сигналов. Изд. 2-е, испр. — М.: Техносфера, 2007. — 856 с.

20. *Оппенгейм А. В., Шафер Р.В.* Цифровая обработка сигналов. — М.: Связь, 1979. — 416 с.

21. *Питерсон У.* Коды, исправляющие ошибки. — М.: Мир, 1964. — 278 с.

22. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. — М.: Мир, 1976. — 596 с.

23. *Рабинер Л., Гоулд Б.* Теория и применение цифровой обработки сигналов. — М.: Мир, 1978. — 848 с.

24. *Самойленко С.И.* Помехоустойчивое кодирование. — М.: Наука, 1966. — 240 с.

25. *Теория* прикладного кодирования: Учеб. пособие: В 2 т. / В.К. Конопелько, В.А. Липницкий, В.Д. Дворников и др.; Под ред. В.К. Конопелько. — Мн.: БГУИР, 2004. — Т. 1. — 285 с.; Т. 2. — 398 с.

26. *Хемминг Р.В.* Цифровые фильтры. — М.: Недра, 1987. — 221 с.

27. *Цифрова* обробка аудіо- та відеоінформації у мультимедійних системах: Навчальний посібник / О.В. Дробик, В.В. Кідалов, В.В. Коваль та ін. — К.: Наукова думка, 2008. — 144 с.

ЗМІСТ

Навчальне видання

ОЛЕКСЕНКО Павло Феофанович
КОВАЛЬ Валерій Вікторович
РОЗОРІНОВ Георгій Миколайович
СУКАЧ Георгій Олексійович

ТЕОРЕТИЧНІ ОСНОВИ ЗАВАДОСТІЙКОГО КОДУВАННЯ

ЧАСТИНА І

Підручник для вищих навчальних закладів

Київ, Науково-виробниче підприємство
«Видавництво "Наукова думка" НАН України», 2010

Художній редактор *І.Р. Сільман*
Оформлення художника *В.В. Кузьменка*
Технічний редактор *Г.М. Ковальова*
Коректор *Л.Г. Бузіашвілі*
Комп'ютерна верстка *Л.В. Багненко*

Підп. до друку 23.06.2010. Формат 70 × 100/16.
Офс. друк. Папір офс. № 1. Гарнітура Таймс.
Ум. друк. арк. 15,6. Ум. фарбо-відб. 16,25.
Обл.-вид. арк. 17,0. Тираж 300 прим. Зам. № 628

НВП «Видавництво "Наукова думка" НАН України»
Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру ДК № 2440 від 15.03.2006 р.
01601 Київ 1, вул. Терещенківська, 3

ЗАТ фірма "Віпол"
03151 Київ 151, вул. Волинська, 60
Свідоцтво про внесення до Державного реєстру
серія ДК № 752 від 27.12.2001