ФАКУЛЬТЕТ УПРАВЛІННЯ ПРОЦЕСАМИ ПЕРЕВЕЗЕНЬ

Кафедра вищої математики

В.І. Храбустовський, Ю.С. Шувалова

ЕЛЕМЕНТИ ТЕОРІЇ ЗАВАДОСТІЙКОГО КОДУВАННЯ

Конспект лекцій

Харків - 2015

Храбустовський В.І., Шувалова Ю.С. Математичні основи теорії кодування: Конспект лекцій. – Харків: УкрДАЗТ,

2015.– 38 с.

Конспект лекцій призначено для самостійного вивчення елементів теорії завадостійкого кодування. Викладання матеріалу максимально спрощено, і для його засвоєння достатньо володіння лише деякими розділами вищої математики, які викладаються у першому семестрі студентам ВТНЗ. Для кращого засвоєння матеріалу конспект містить теоретичні питання та завдання для засвоєння теоретичного матеріалу. Для більш глибокого опрацювання матеріалу можна використати [1-12].

Рекомендовано для магістрів факультету АТЗ денної та заочної форм навчання.

Іл. 13, табл. 3, бібліогр.: 12 назв.

Конспект лекцій розглянуто та рекомендовано до друку на засіданні кафедри вищої математики 1 вересня 2014 р., протокол № 1.

Рецензент

проф. С.І. Приходько

В.І. Храбустовський, Ю.С. Шувалова

ЕЛЕМЕНТИ ТЕОРІЇ
ЗАВАДОСТІЙКОГО КОДУВАННЯ

Конспект лекцій

Відповідальний за випуск Шувалова Ю.С.

Редактор Решетилова В.В.

Підписано до друку 19.09.14 р.

Формат паперу 60х84 1/16. Папір писальний.

Умовн.-друк.арк. 2,00. Тираж 50. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

Кафедра вищої математики

В.І. Храбустовський, Ю.С. Шувалова

ЕЛЕМЕНТИ ТЕОРІЇ ЗАВАДОСТІЙКОГО КОДУВАННЯ

Конспект лекцій
для самостійної роботи студентів факультету АТЗ всіх форм навчання

Харків 2014

Храбустовський В.І., Шувалова Ю.С. Математичні основи теорії кодування: Конспект лекцій. – Харків: УкрДАЗТ, 2015.– 38 с.

Конспект лекцій призначено для самостійного вивчення елементів теорії завадостійкого кодування. Викладання матеріалу максимально спрощено, і для його засвоєння достатньо володіння лише деякими розділами вищої математики, які викладаються у першому семестрі студентам ВТНЗ. Для кращого засвоєння матеріалу конспект містить теоретичні питання та завдання для засвоєння теоретичного матеріалу. Для більш глибокого опрацювання матеріалу можна використати [1-12].

Рекомендовано для магістрів факультету АТЗ денної та заочної форм навчання.
Іл. 13, табл. 3, бібліогр.: 12 назв.

Конспект лекцій розглянуто та рекомендовано до друку на засіданні кафедри вищої математики 1 вересня 2014 р., протокол № 1.

Рецензент

проф. С.І. Приходько

ЗМІСТ

ВСТУП

Теорія кодування виникла в 1948-1950 рр. в роботах К. Шеннона та Р. Хеммінга. Зазвичай при викладанні теорії кодування використовується теорія Галуа скінченних полів. В конспекті ця теорія не використовується, що дозволяє читати його студенту з мінімальною математичною підготовкою. Конспект базується на навчальному посібнику [7], матеріал якого доповнено і перероблено.

1 НАЙПРОСТІШІ ПОНЯТТЯ ТЕОРІЇ КОДУВАННЯ

Часто постає завдання передачі *інформації*. Під *інформацією* мається на увазі сукупність будь-яких відомостей про події, явища, предмети. Інформацію передають у вигляді *повідомлень*. Найпростіша *схема зв'язку* наведена на рисунку 1.1.

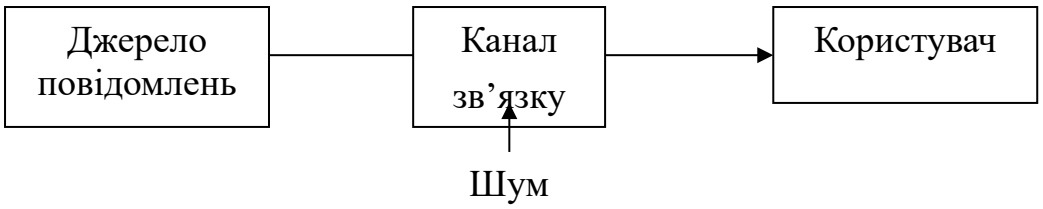


Рисунок 1.1

Повідомлення, які передаються каналом зв’язку можуть мати різну природу. Зараз, частіш за все, повідомлення передаються у цифровій формі. Однією з переваг цифрових повідомлень є можливість їх кодування для підвищення завадостійкості.

В найпростішому випадку для передачі цифрових повідомлень використовують лише *два символи*: «0» і «1».

Визначення. Упорядкована множина символів «0» і «1» називається *словом.* Число символів, з яких складене слово, зветься *довжиною слова*.

Приклад 1.1. Довжини слів 00, 001, 1100 дорівнюють відповідно 2, 3, 4.□

Послідовність великої кількості символів «0» і «1», яку видає джерело повідомлень, можна розбити на слова однакової

довжини. Для цього, можливо, до останнього слова доведеться додати кілька «0». Тому будемо вважати, що джерело повідомлень видає слова довжини k.

У реальних каналах зв'язку можуть виникати помилки, які, як правило, мають випадковий характер. Наприклад, якщо передано слово 10001, а прийнято 11001, то сталася помилка у другому символі, або якщо передано 11010, а прийнято 01000, то відбулися дві помилки у першому і четвертому символах. Бажано було б зменшити ймовірність помилки. Для цього до кожного переданого слова можна додати ще кілька символів «0» і «1», які називаються *перевірочними*, так, щоб можна було дізнатися, чи є в прийнятому слові помилки.

Визначення. Процес додавання перевірочних символів називається *завадостійким кодуванням*, а пристрій, який його здійснює – *кодером*.

Визначення. Процес виявлення або виправлення помилки в прийнятому слові називається *декодуванням*, а пристрій, який його здійснює – *декодером*.

Отже, на вхід кодера надходить слово a фіксованої довжини k. Кодер додає до нього перевірочні символи і отримує *кодове слово* b довжини $n \geq k$.

Схема зв'язку з рисунка 1.1 набуває вигляду, який наведено на рисунку 1.2.

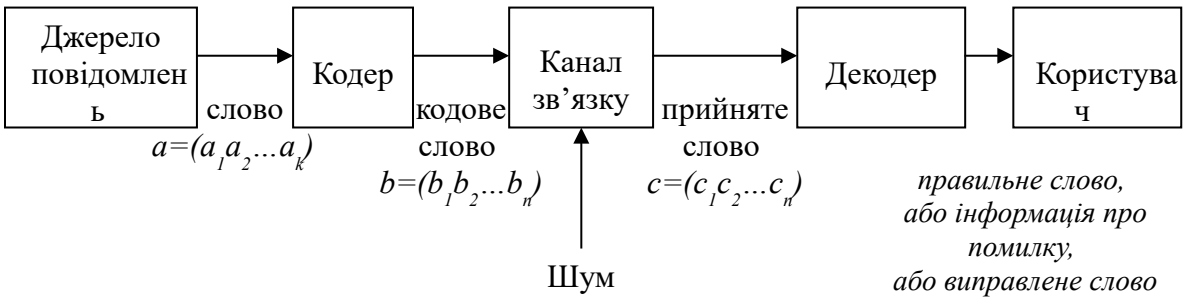


Рисунок 1.2

Визначення. *Кодом* називається множина всіх кодових слів довжини n. Число n називається *довжиною* кода.

З n символів кодового слова корисну інформацію містять k символів, що надійшли з джерела. Решта $n-k$ перевірочних символів служать для виправлення або виявлення помилок.

Найпростішим прикладом кода є *код з однією перевіркою на парність.*

Кодування й декодування кода з однією перевіркою на парність:

– ***кодування***. Кодер додає до слова довжини k, яке надійшло на його вхід, один символ «0» або «1» так, щоб одержане кодове слово мало парну кількість «1». Це кодове слово передається каналом зв'язку;

– ***декодування***. Декодер рахує кількість «1» в одержаному слові. Якщо ця кількість непарна, то вважається, що виявлена помилка; якщо парна, то вважається, що помилки не було.

Декодер перевіряє кожне отримане слово на парність суми його символів, звідки і назва кода.

Приклад 1.2. При передачі слова каналом зв’язку, що наведено на рисунку 1.3, трапилася помилка в першому символі.

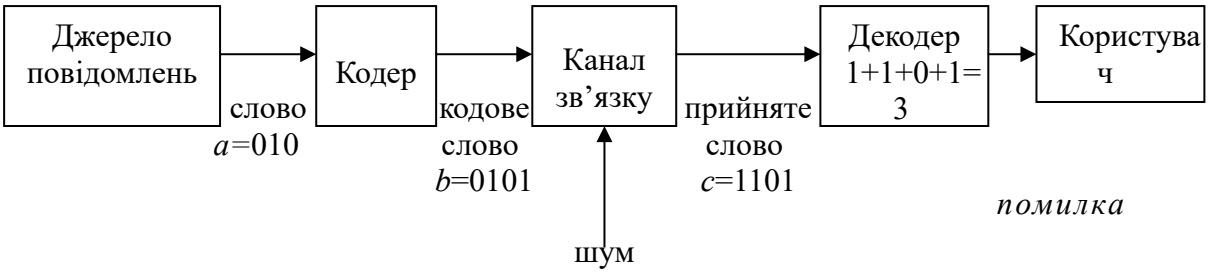


Рисунок 1.3

Декодер дійсно виявив помилку: сума символів «1» одержаного слова 1+1+0+1=3 – непарна.□

Очевидно, що код з однією перевіркою на парність може лише виявити, що кількість помилок у слові непарна, але не може жодну з них виправити. Якщо кількість помилок парна, то код з однією перевіркою на парність не може навіть їх виявити. Оскільки апаратурна реалізація цього кода вкрай проста, то даний код застосовують в каналах зв'язку зі слабкими завадами.

Кодове слово можна розглядати як вектор. Тому можна говорити про його координати. Наприклад, перші три координати слова 11100 дорівнюють «1», четверта і п'ята – «0».

Визначення. *Відстанню Хеммінга* $\rho(a,b)$ між словами a і b однакової довжини називається число їх розбіжних координат.

Приклад 1.3. Нехай $a=000,\ b=100,\ c=010$, тоді $\rho(a,b)=1,\ \rho(b,c)=2,\ \rho(a,c)=1$. □

Відстань Хеммінга є аналогом звичайної відстані між точками та має такі самі властивості:

1) $\rho(a,b)\geq 0$;

2) $\rho(a,b)=0 \Leftrightarrow a=b$;

3) $\rho(a,b)=\rho(b,a)$;

4) нерівність трикутника: $\rho(a,c)\leq \rho(a,b)+\rho(b,c)$.

Відстань між двома словами дорівнює числу помилок, які треба зробити в одному слові, щоб отримати інше. Відстань між переданим і прийнятим словом дорівнює числу помилок, що відбулися при передачі каналом зв'язку.

Нехай p ймовірність помилки у довільній координаті кодового слова, відповідно $1-p$ ймовірність правильної передачі цієї координати. Ймовірність того, що відбудеться помилка, у конкретних m координатах дорівнює $p^m(1-p)^{n-m}$. Наприклад, при $p=0,25$ ймовірність помилки у 3 конкретних координатах з 5 дорівнює 0,008978906. При $p<0,5$ $(\Rightarrow 1-p>p)$ маємо $(1-p)^n > p(1-p)^{n-1} > p^2(1-p)^{n-2} > \ldots$. Отже, чим більше помилок, тим менша ймовірність. Це дозволяє вважати, що велика кількість помилок малоймовірна, та будувати теорію виправлення невеликої кількості помилок.

Оскільки помилки малоймовірні, то, як правило, відстань між переданим та одержаним словом мала. Тому при виправленні та виявленні помилок використовують *принцип максимальної правдоподібності.* А саме:

– для *виявлення помилок* декодер лише перевіряє, чи є отримане слово кодовим. Якщо отримане слово не є кодовим, то він сигналізує про помилку. Якщо отримане слово є кодовим, то

декодер вважає, що це слово і було передано, тобто, що помилок не було;

– для *виправлення помилок* декодер повинен знайти кодове слово, яке є найближчим до одержаного.

Мірою здатності коду виявляти і виправляти помилки є його *мінімальна відстань*.

Визначення. *Мінімальною відстанню* кода V називається мінімум відстані між його словами, які відрізняються одне від одного:

$$d(V) = \min_{x,y \in V} \rho(x,y) \quad (x \neq y).$$

Приклад 1.4. З множини всіх слів довжини 2 отримано код довжини 3 з однією перевіркою на парність. Він складається з чотирьох слів: $a_1 = 000, a_2 = 011, a_3 = 110, a_4 = 101$. Відстань між двома кодовими словами дорівнює 2 і, отже, $d = 2$. □

Приклад 1.5. Код, що складається із слів 000, 100, 111, має мінімальну відстань $d = 1$. □

Мінімальна відстань кода дозволяє *оцінити кількість помилок, які код гарантовано може виявити, а також виправити*.

Нижче, згідно з принципом максимуму правдоподібності, слова «*код може виявити (виправити)* t *помилок*» означають, що в t-околі* кожного кодового слова a кодових слів, які не дорівнюють a, немає (і для будь-якого слова, яке не дорівнює a, найближчим кодовим є a**).

Теорема 1.1. Код V може виявити t помилок, тоді і тільки тоді, коли $d(V) \geq t+1$.

Доведення. Якщо код V може виявити всі t помилок, то в t-околі кожного кодового слова немає інших кодових слів. Тому $d(V) > t \Rightarrow d(V) \geq t+1$.

У зворотний бік: нехай $d(V) \geq t+1$. Якщо в t-околі кодового слова a є кодове слово, яке не дорівнює a, то $d(V) \leq t$, що

---

* t- околом слова називають множину слів, які знаходяться на відстані не більше t від цього слова.

** Це означає, що якщо при передачі кодового слова сталося не більше t помилок (одержане слово опинилося у t- околі слова a), то декодер замінить одержане слово словом a, і помилку буде виправлено.

суперечить умові $d(V) \geq t+1$. Отже, в t-околі слова a немає кодових слів, які не дорівнюють a. □

Наприклад, код з $d(V) = 2$ може виявити тільки одну помилку (у кожному слові), а код з $d(V) = 3$ – дві помилки. Зокрема, код, розглянутий у прикладі 1.4, може виявити одну помилку. Код із прикладу 1.5 не здатний виявляти помилки.

Теорема 1.2. Код V може виправити t помилок, тоді і тільки тоді, коли $d(V) \geq 2t+1$.

Доведення. Згідно з поясненнями перед теоремою 1.1, якщо код V може виправити всі t помилок, то в t-околі кожного кодового слова a немає кодових слів, які не дорівнюють a, околи кодових слів не перетинаються. Тому $d(V) > 2t \Rightarrow d(V) \geq 2t+1$.

У зворотний бік: нехай $d(V) \geq 2t+1 \Rightarrow$ t-околи кодових слів не перетинаються, а значить, код V може виправити t помилок. □

Наприклад, для виправлення однієї помилки має виконуватися нерівність $d(V) \geq 3$, а для виправлення двох помилок – нерівність $d(V) \geq 5$.

Розв'язання задачі вибору коду оптимального за тим чи іншим критерієм складає суть теорії кодування.

Теоретичні питання

1 Дати визначення слова і його довжини.

2 Яке завдання вирішує теорія кодування ?

3 У чому полягає кодування і декодування для коду з однією перевіркою на парність? Скільки помилок виявляє цей код?

4 Дати визначення коду і його довжини.

5 Що називається відстанню Хеммінга? Які її властивості?

6 Що таке принцип максимальної правдоподібності?

7 Що таке мінімальна відстань коду? Скільки помилок може виявити (виправити) код?

Завдання для засвоєння теоретичного матеріалу

1 Використовуючи код з однією перевіркою на парність , закодувати слова: 11010 , 00101 , 11000.

2 При кодуванні використано код з однією перевіркою на парність. Нехай у кожному з отриманих слів 01010, 10110, 00100 міститься не більше однієї помилки . Вказати слова, що містять помилки.

3 Для коду { 1010, 0101, 1111, 0000 } знайти його довжину , мінімальну відстань, число помилок, які код виявляє і виправляє.

4 Дати відповідь на ті ж питання, що і в попередній вправі, для коду { 101101, 010011, 000000 }. При передачі слова 101101 сталася помилка в першій координаті. Яке слово буде отримано? Як декодер її виправить ?

5 Довести, що код V може виправити t або менше помилок, та виявити t' або менше помилок ($t' \geq t$), якщо $d(V) \geq t + t' + 1$.

2 ЛІНІЙНІ КОДИ

На множині символів {0, 1} введемо операції додавання та множення таким чином:

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Єдина відмінність від дій з числами 0, 1 полягає в тому, що тут 1 +1 = 0.

Нехай V_n – це множина всіх слів довжини n – впорядкований набір з n символів «0» та «1». Зі словами можна здійснити дії аналогічні діям зі звичайними векторами, координати яких є звичайні числа. А саме, додавання двох слів з V_n здійснюється покоординатно (з урахуванням 1+1=0). Наприклад, $010 + 100 = 110$, $0111 + 1100 = 1011$, $101 + 101 = 000$.

Множення слів $a \in V_n$ на символи «0», «1» здійснюється так: $1 \cdot a = a$, $0 \cdot a = \overline{0}$, де. $\overline{0}$ – слово з нульовими координатами.

Властивості операцій додавання та множення на символи «0», «1» аналогічні лінійним операціям з векторами у звичайному n - вимірному векторному просторі $\mathbb{R}_n$.

Тобто якщо $a,b,c \in V_n$, $\alpha,\beta \in \{0,1\}$, то

1) $a+b=b+a$,
2) $(a+b)+c=a+(b+c)$,
3) $\alpha(\beta a)=(\alpha\beta)a$,
4) $(\alpha+\beta)a=\alpha a+\beta a$,
5) $\alpha(a+b)=\alpha a+\alpha b$.
6) $1\cdot a=a$.

Нагадаємо, що непорожня підмножина M звичайного векторного простору $\mathbb{R}_n$ називається *підпростором*, якщо M сама є векторним простором, тобто $\vec{a},\vec{b} \in M \Rightarrow \vec{a}+\vec{b} \in M$, $\alpha \in \mathrm{R}$, $\vec{a} \in M \Rightarrow \alpha\vec{a} \in M$.

Приклад 2.1. Вектори, які лежать на площині, що знаходиться в $\mathbb{R}_3$, утворюють підпростір (див. рисунок 2.1).

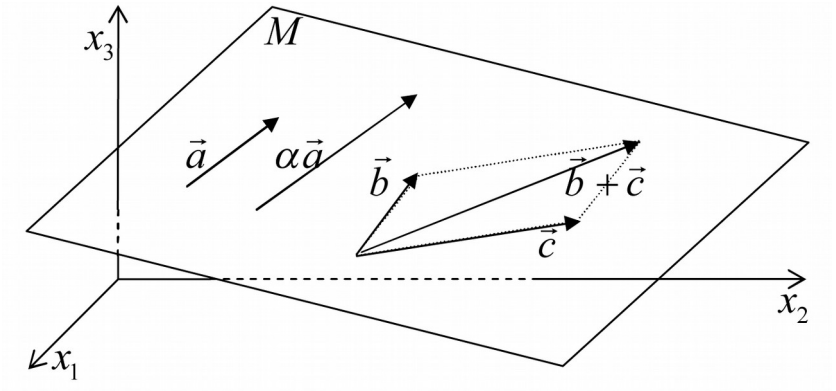


Рисунок 2.1

□

Аналогічно визначається підпростір у V_n. Нехай V – підпростір в V_n. Зокрема V може збігатися з V_n.

Визначення. Підмножина $B \subset V$ називається *базисом* підпростору V, якщо кожне слово $a \in V$ єдиним чином подається у вигляді лінійної комбінації деяких слів з B.

Оскільки у V_n коефіцієнтами лінійної комбінації можуть бути лише символи «0» та «1», то визначення базису можна переформулювати так.

Визначення. Підмножина $B \subset V$ називається *базисом* підпростору V, якщо кожне слово $a \in V$ єдиним чином подається у вигляді суми деяких слів з B.

Всі базиси підпростору містять однакове число векторів. Це число називається *вимірністю* підпростору і позначається $\dim V$.

Теорема 2.1. Підпростір $V \subset V_n$ містить 2^k слів, де k – вимірність V.

Визначення. *Лінійним кодом* називається підпростір простору V_n.

Наведемо кілька прикладів *лінійних кодів*

Приклад 2.2. З $2^3 = 8$ можливих слів довжини 3 обрано підпростір $V = \{000, 101, 011, 110\}$, який є лінійним кодом. $\dim V = 2$. Кількість слів у коді $2^2 = 4$. □

Приклад 2.3. З $2^4 = 16$ можливих слів довжини 4 обрано підпростір $V = \{0000, 0101, 1010, 1111\}$, який є лінійним кодом. $\dim V = 2$. Кількість слів у коді $2^2 = 4$. □

Визначення. *Вагою* слова називається кількість одиниць, які воно містить.

Приклад 2.4. Розглянемо коди з прикладів 2.2 та 2.3. Всі ненульові слова коду $V = \{000, 101, 011, 110\}$ мають вагу 2. В коді $V = \{0000, 0101, 1010, 1111\}$ слово 1111 має вагу 4, слова 0101,1010 мають вагу 2. □

Зауважимо, що для будь-якого слова a

$$\textit{вага } a = \rho\left(a, \bar{0}\right).$$

Теорема 2.2. Для лінійного коду V його мінімальна відстань $d(V)$ дорівнює найменшій з ваг його ненульових слів:

$$d(V) = \min_{a \in V} \textit{вага } a, \quad a \neq \bar{0}.$$

Доведення. $d(V) = \min\limits_{\substack{b,c \in V, \\ b \neq c}} \rho(b,c) = \min\limits_{b \neq c} \rho(b+c, 0) = \min\limits_{\substack{a=b+c \\ a \neq 0}} \textit{вага } a.$ □

Приклад 2.5. Коди з прикладу 2.4 мають мінімальну вагу 2, а отже, мінімальна відстань $d(V) = 2$. □

Далі ми будемо вивчати лише *лінійні коди*, тому замість *лінійний код* будемо говорити просто *код*. Код, в якого *довжина* дорівнює n, *вимірність* дорівнює k, *мінімальна відстань* дорівнює d, будемо називати $[n, k, d]$ - *кодом*.

Визначення. Два коди називаються *еквівалентними*, якщо слова одного коду можна одержати зі слів іншого за допомогою однієї і тієї ж перестановки координат.

Приклад 2.6. Після перестановки 1-ї і 2-ї координат коду $V_1 = \{0000, 1001, 0111, 1110\}$ маємо еквівалентний код $V_2 = \{0000, 0101, 1011, 1110\}$, параметри n, k, d цих кодів збігаються. □

Нехай V це $[n, k, d]$ - код (базис містить k векторів довжини n).

Визначення. *Матрицею, яка породжує* код V, називається матриця G розміру $k \times n$, рядками якої є базисні слова коду.

Назва матриці G обумовлена тим, що всі слова коду V можна отримати так:

$$aG, \tag{2.1}$$

де a – всі можливі слова довжини k.

Приклад 2.7. Матриця, яка породжує код з прикладу 2.2, має вигляд $G = \begin{pmatrix} 101 \\ 110 \end{pmatrix}$.

Матриця, яка породжує код з прикладу 2.3, має вигляд $G = \begin{pmatrix} 0101 \\ 1010 \end{pmatrix}$. □

Задавати код зручніше за допомогою матриці, яка його породжує, а не шляхом перерахування його слів. Наприклад, якщо вимірність коду $k = 20$, то код містить $2^{20} = 1048576$ слів, і завдання 20 базисних слів визначає більше мільйона кодових слів.

Взагалі кажучи, код має не одну матрицю, яка породжує код. Додаючи рядки і переставляючи стовпці матриці, яка породжує код, можна привести її до *канонічного вигляду*

$$G = (E_K A), \tag{2.2}$$

де E_K – одинична $k \times k$ – матриця, A – деяка матриця.

Якщо матриця G, яка породжує код, має канонічний вигляд (2.2), то за формулою (2.1) перші k координат кодового слова b збігаються з координатами слова a, яке передається.

Зауваження. Між параметрами $[n, k, d]$ - *коду* існує такий зв'язок:

$$\boxed{d \le n - k + 1,}$$

який називається *межею Сінглтона* (ця нерівність є наслідком формул (2.1), (2.2)).

З визначення матриці, яка породжує код, випливає, що перестановка її стовпців приводить до тієї ж перестановки координат всіх кодових слів, тобто приводить до еквівалентного коду.

Приклад 2.8

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \uparrow \sim \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \uparrow \sim \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \uparrow \sim \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ \underbrace{0 \quad 0 \quad 1}_{E_3} & & & \underbrace{1}_{A} \end{pmatrix} = \left(E_3 A \right).$$

переставлені 1 та 3 стовпці — *до 3–го рядка додано 2–й рядок* — *переставлено 3 та 4 сповці*

□

Визначення. $(n-k) \times n$ матриця H називається *перевірочною* для коду V, якщо $Ha^T = \overline{0}$ для $\forall a \in V$.*

Якщо матриця G, яка породжує код, має канонічний вигляд (2.2), то перевірочна матриця $H = \left(A^T E_{n-k} \right)$.

Часто поряд з лінійним кодом V використовують *дуальний* (*двоїстий, ортогональний*) код $V^\perp$, який складається з усіх слів (векторів), які перпендикулярні кожному слову (вектору) з кода V. Породжуюча (перевірочна) матриця коду $V^\perp$ збігається з перевірочною (породжуючою) матрицею коду V.

Теорема 2.3. Про зв'язок перевірочної матриці та мінімальної відстані. Мінімальна відстань коду $d(V) = d$, тоді і

---

*Матриця A^T називається транспонованою до матриці A, якщо її рядки є стовпцями матриці A. Наприклад, Якщо a – слово-рядок, то a^T – слово-стовпець.

тільки тоді, коли у матриці H, яка породжує цей код, будь-які $d-1$ стовпців лінійнонезалежні та існує d лінійно залежних стовпців.

Доведення цієї теореми пропонуємо читачеві.

Приклад 2.9. Для коду з однією перевіркою на парність матриця, яка породжує код, має вигляд $G=\begin{pmatrix} 1 \\ E_K \vdots \\ 1 \end{pmatrix}$ (розмір матриці G дорівнює $k\times n,\ n=k+1$), перевірочна матриця $H=\begin{pmatrix}1 & \dots & 1\end{pmatrix}$ (розмір матриці H дорівнює $1\times k$). □

Код з перевіркою на парність може виявляти одну помилку, але виправити помилку не вдасться. Хотілося б побудувати код, який виправляє хоча б одну помилку.

Приклад 2.10. **Код Хеммінга**. Стовпцями перевірочної матриці H коду Хеммінга є всі ненульові вектори довжини r. Порядок розташування стовпців довільний.

Розмір матриці H дорівнює $r\times\left(2^r-1\right)\ \Rightarrow n=2^r-1$, $k=n-r=2^r-1-r$. За теоремою 2.3 мінімальна відстань коду Хеммінга $d=3$. Код Хеммінга є $\left[2^r-1, 2^r-1-r, 3\right]$–кодом. Цей код може виявити дві помилки та виправити одну.

При $r=2$ маємо $\left[3, 1, 3\right]$–*код*. Перевірочну матрицю можна взяти, наприклад, у вигляді $H=\begin{pmatrix}1 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}$. Тоді відповідна матриця, яка породжує код, дорівнює $G=\begin{pmatrix}1 & 1 & 1\end{pmatrix}$.

При $r=3$ маємо $\left[7, 4, 3\right]$–*код*. Перевірочну матрицю можна взяти, наприклад, у вигляді

$$H=\begin{pmatrix}1 & 1 & 1 & 0 & 1 & 0 & 0\\ 0 & 1 & 1 & 1 & 0 & 1 & 0\\ 0 & 0 & 1 & 1 & 1 & 0 & 1\end{pmatrix}. \qquad (2.3)$$

Переставляючи стовпці у (2.3), отримаємо перевірочну матрицю у канонічному вигляді

$$H_1 = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \qquad (2.4)$$

Тоді відповідна матриця, яка породжує код, дорівнює

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \square$$

2.1 Кодування і декодування лінійного коду

Нехай каналом зв'язку передається слово a довжини k:

– *кодування*. Слово a множиться на матрицю G, яка породжує код. Одержуємо кодове слово $b = aG$ довжини n;

– *декодування*. Нехай c – прийняте слово. Множимо перевірочну матрицю H на c^T. Одержуємо слово $s = Hc^T$, яке називається синдромом. Якщо $s = \overline{0}$, то декодер вважає, що помилок не було*. Якщо $s \neq \overline{0}$, то виявлена помилка. Для її виправлення треба знайти кодове слово, яке є найближчим до отриманого слова c.

Зауваження. *Декодування за синдромом, якщо у каналі зв'язку може відбутися максимум одна помилка.* Нехай в перевірочній матриці всі слова ненульові та попарно різні, і відомо, що в слові може статися максимум одна помилка. Тоді ненульовий синдром завжди збігається з якимось стовпцем перевірочної матриці, і позиція цього стовпця вказує позицію помилки в прийнятому слові.

---

*Дійсно, нехай матриця G, яка породжує код, має канонічний вигляд (2.1), і прийняте слово є кодовим, тобто дорівнює $c = \tilde{a}G$. Тоді

$$Hc^T = H(\tilde{a}G)^T = HG^T\tilde{a}^T = \left(A^T E_{n-k}\right)\begin{pmatrix} E_k \\ A^T \end{pmatrix}\tilde{a}^T = \underbrace{\left(A^T + A^T\right)}_{\| \\ 0}\tilde{a}^T = 0$$

.

Приклад 2.11. Матриця $G=\begin{pmatrix}1 & 0 & 0 & 1\\ 0 & 1 & 1 & 1\end{pmatrix}$, яка породжує код V, має розмір (2×4). Отже, $k=2,\ n=4$. Код V містить $2^k=2^2=4$ слова, які можна одержати за формулою (2.1)

$$V=\{0000,1001,0111,1110\}. \qquad (2.5)$$

Мінімальна відстань коду $d=2$. Отже, за теоремою 1.2 код (2.5) може виявити одну помилку.

Матриця G, яка породжує код (2.5), має канонічний вигляд $G=(E_2 A)$, де $A=\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}$. Тому перевірочна матриця $H=\left(A^T E_{4-2}\right)=\begin{pmatrix}0 & 1 & 1 & 0\\ 1 & 1 & 0 & 1\end{pmatrix}$.

Нехай передається слово $a=(0\ \ 1)$. Відповідне кодове слово $b=aG=(0\ \ 1\ \ 1\ \ 1)$.

а) Якщо при передачі кодового слова b помилок не було, то буде прийнято слово $c=(0111)=b$. Декодування: $H\cdot(0111)^T=\overline{0}$ (рисунок 2.2);

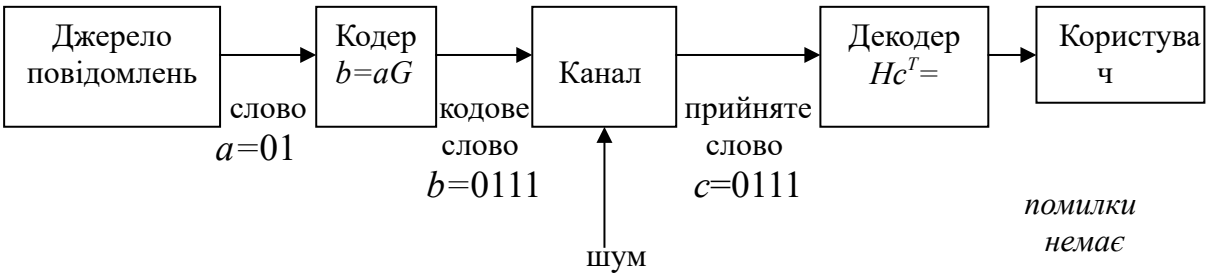


Рисунок 2.2

б) Якщо при передачі кодового слова b сталася помилка, наприклад, в 4-й координаті, то буде прийняте $c=(0110)$. Декодування: $H(0110)^T=\begin{pmatrix}0\\ 1\end{pmatrix}\neq\overline{0}$ (рисунок 2.3).

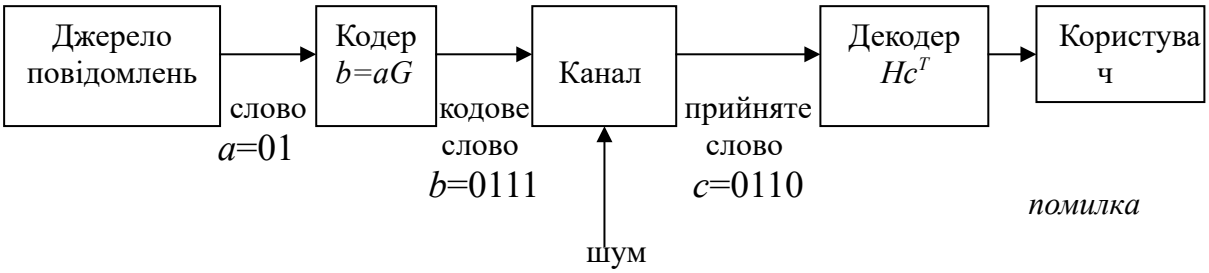


Рисунок 2.3

Найближчими до $c=(0110)$ є два слова $b=(0111)$ та (1110), і тому декодер не може виправити помилку.

Синдром s збігається з першим та четвертим стовпцем перевірочної матриці, отже, декодування за синдромом також неможливо.□

Приклад 2.12. Матриця $G=\begin{pmatrix}1&0&0&0&1&1\\0&1&0&1&0&1\\0&0&1&1&1&0\end{pmatrix}$, яка породжує код V, має розмір (3×6). Отже, $n=6,\ k=3$. Код V містить $2^k=2^3=8$ слів довжини $n=6$, які можна одержати за формулою (2.1): $a_1=\overline{0}$, $a_2=(100011)$, $a_3=(010101)$, $a_4=(001110)$, $a_5=(110110)$, $a_6=(101101)$, $a_7=(011011)$, $a_8=(111000)$.

Мінімальна відстань коду $d=3$, тому за теоремою 1.2 код виправляє одну або за теоремою 1.1 виявляє дві помилки.

Матриця G, яка породжує код V, має канонічний вигляд $G=(E_3A)$, де $A=\begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}$. Тому перевірочна матриця $H=\left(A^TE_{6-3}\right)=\begin{pmatrix}0&1&1&1&0&0\\1&0&1&0&1&0\\1&1&0&0&0&1\end{pmatrix}$.

Нехай передається слово $a=(010)$. Відповідне кодове слово $b=aG=(0\,1\,0\,1\,0\,1)$.

а) Якщо при передачі кодового слова b помилок не було, то буде прийнято слово $c=(0\,1\,0\,1\,0\,1)=b$. Декодування: $H(0\,1\,0\,1\,0\,1)^T=\overline{0}$ (рисунок 2.4);

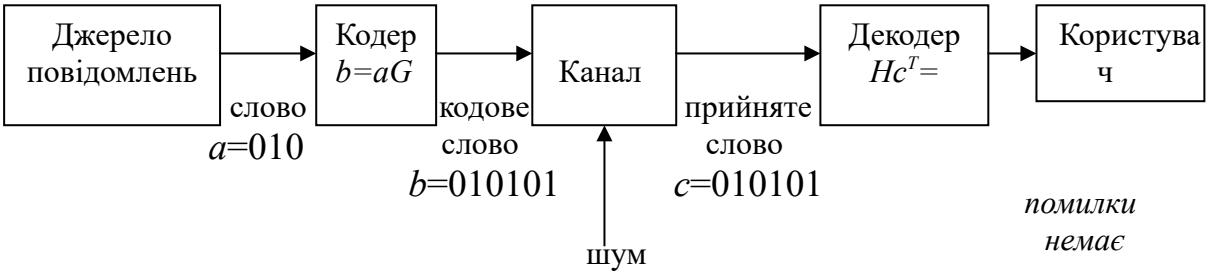


Рисунок 2.4

б) Якщо при передачі кодового слова b, наприклад, сталася помилка в останній координаті, то буде прийняте слово $c=(0\,1\,0\,1\,0\,0)$. Декодування: $H(0\,1\,0\,1\,0\,0)^T=(001)^T\neq\overline{0}$ (рисунок 2.5).

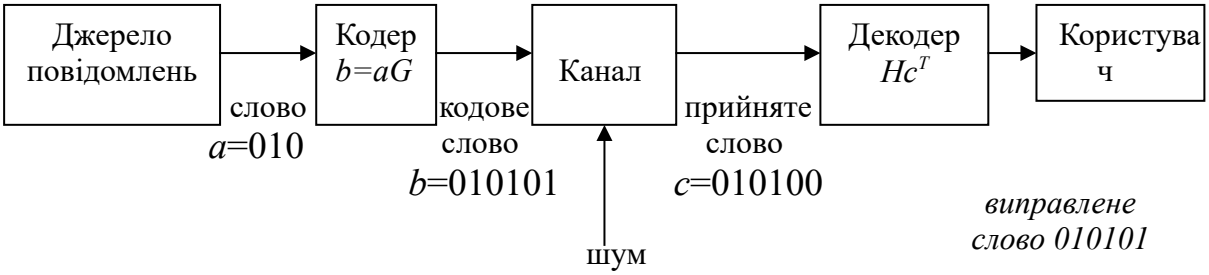


Рисунок 2.5

Декодер замінює слово c найближчим, яке дійсно дорівнює b.

Згідно із зауваженням, оскільки $s=(001)^T$ збігається з останнім стовпчиком матриці H, то помилка дійсно сталась у останній координаті;

в) Якщо при передачі кодового слова b сталося дві помилки, наприклад, в 3-й та 4-й координатах, то буде прийняте слово $c=(011001)$. Декодування: $H(0\,1\,1\,0\,0\,1)^T=(010)^T\neq\overline{0}$ (рисунок 2.6).

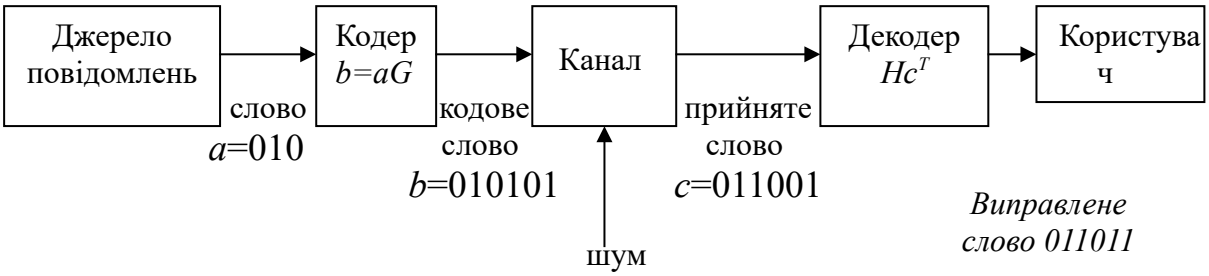


Рисунок 2.6

Декодер виправив помилку неправильно, оскільки найближче до c кодове слово a_7 не збігається зі словом a_3, яке передавалося.

Помилку буде виправлено неправильно також і при декодуванні за синдромом.

Приклад в) показує, що декодування «у найближче слово» або «за синдромом» приведе до правильного результату, тільки якщо число помилок дійсно не більше t (умову про «максимум одну помилку» неможливо відкинути). □

Приклад 2.13. Розглянемо $[7, 4, 3]$-*код* Хеммінга з перевірочною матрицею (2.4). За визначенням перевірочна матриця коду Хеммінга не містить нульових та однакових стовпців. Отже, для цих кодів у каналах зв'язку, де «можлива максимум одна помилка», завжди можливо декодування за синдромом згідно із зауваженням.

Нехай $a = (0011)$, тоді кодове слово $b = aG = (0\ 0\ 1\ 1\ 0\ 1\ 0)$.

Якщо прийняте слово $c = (0\ 0\ 1\ 1\ 0\ 1\ 0)$, то синдром $s = Hc^T = \overline{0}$, отже, помилки дійсно немає.

Якщо прийняте слово $c = (0\ 0\ 1\ 0\ 0\ 1\ 0) \neq b$, то синдром $s = Hc^T = (101)^T \neq \overline{0}$ збігається з 4 стовпчиком перевірочної матриці H, отже, помилка дійсно сталася у четвертій координаті. □

Теоретичні питання

1 Дати визначення ваги слова

2 Дати визначення лінійного кода.

3 Сформулювати теорему про мінімальну відстань лінійного коду.

4 Дати визначення матриці, яка породжує лінійний код, та перевірочної матриці лінійного коду. Навести їх канонічний вигляд.

5 Чому лінійний код зручніше задавати матрицею, яка його породжує, а не шляхом перерахування всіх його слів?

6 Дати визначення коду Хеммінга.

7 Дати визначення синдрому. Коли і як виходячи із синдрому визначається помилка у прийнятому слові?

Завдання для засвоєння теоретичного матеріалу

1 Перевірити, виписавши всі кодові слова за допомогою матриці, яка породжує код, що [7,4,3]-код Хеммінга має по одному слову ваги 0 та 7, по сім слів ваги 3 та 4.

2 Задано матрицю G, яка породжує код, та слова, які передаються.

Виписати всі кодові слова. Знайти параметри, коди, число помилок, які виявляються та виправляються. Закодувати наведені слова, які передаються. Показати декодування за синдромом та знаходження найближчого до прийнятого кодового слова: при відсутності помилок; при помилці в першій координаті; при помилці в 3-й координаті.

а) $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$. Передаються слова 011, 101;

б) $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$. Передаються слова 101, 001, 111.

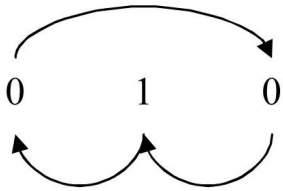
3 Довести теорему 2.3 про зв’язок перевірочної матриці та мінімальну відстань коду.

3 ЦИКЛІЧНІ КОДИ

Циклічні коди – це важливий клас лінійних кодів. Вони відрізняються достатньо добре вивченою математичною структурою та простотою реалізації.

Визначення. *Циклічним зсувом* слова називається переміщення всіх його символів на одну позицію ліворуч, а першого символа – на останню позицію.

Приклад 3.1.



$\longrightarrow$ 100; 0111 $\longrightarrow$ 1110

□

Визначення. Код називається *циклічним*, якщо циклічний зсув будь-якого кодового слова є кодовим словом.

Приклад 3.2

Циклічні коди: $\{00,11\}$, $\{000,110,101,011\}$.

Нециклічні коди: $\{00,01,11\}$ і $\{000,001,010,110\}$.□

Зауважимо, що код, еквівалентний циклічному, взагалі кажучи, може не бути циклічним.

Для роботи з циклічними кодами введемо операції множення та ділення слів.

Множення слів проводиться в «стовпчик» аналогічно звичайним числам (з урахуванням 1+1=0).

Приклад 3.3

$$\begin{array}{r} 1101 \\ \times\quad 101 \\ \hline 1101 \\ 1101 \\ \hline 111001 \end{array}.\ \square$$

Зауваження 3.1. Якщо на «порожніх» позиціях і перед першим множником дописати відповідну кількість нулів

$$\begin{array}{r} 001101 \\ \times\quad 101 \\ \hline 001101 \\ +\ 110100 \\ \hline 111001, \end{array}$$

то ми побачимо, що результат множення можна одержати, виконуючи операції циклічного зсуву і додавання. □

Ділення слів проводиться «кутом» аналогічно звичайним числам, але при цьому віднімання замінюється додаванням.

Ділення слів можна здійснювати, коли степінь діленого більша або дорівнює степеню дільника.

Визначення. *Степенем* слова a називається число його символів, що стоять праворуч від першої «1». Позначається $\deg a$.

Приклад 3.4. $\deg(0101)=2,\ \deg(1000)=3,\ \deg(11111)=4$. □

Приклад 3.5. Розділимо 100101 на 101:

$$\begin{array}{rl|l} & 100101 & 101 \\ \hline + & 101 & 1011 \\ \hline & 1101 & \\ + & 101 & \\ \hline & 111 & \\ + & 101 & \\ \hline & 010, & \end{array}$$

тобто $\dfrac{100101}{101}=1011+\dfrac{010}{101}$.

Ділення припиняється тоді, коли степінь залишку стане менше степеня дільника. □

Визначення. Ненульове слово g циклічного кода, яке має найменший степінь, називається *твірним*.

Циклічний код, як і будь-який інший лінійний код, можна описати за допомогою матриці, яка породжує код. Зазначимо, що дуальний (з точністю до еквівалентності) до циклічного кода також буде циклічним кодом.

Матрицю, яка породжує циклічний $[n,k,d]$- код, можна побудувати, взявши першим рядком будь-яке базисне слово цього коду, другим рядком його циклічний зсув, …., k-м рядком циклічний зсув k-1-го рядка. Аналогічно можна побудувати перевірочну матрицю, в які як перший рядок беруть базисне слово дуального коду.

Приклад 3.6. Зрозуміло, що циклічний код $\{00,11\}$ – це [2,1,2]-код. Слово (11) цього коду має найменший степінь, отже, воно твірне. Тому твірна матриця цього коду $G=(1\,1)$.

Циклічний код $\{000,110,101,011\}$ – це [4,2,2]-код. Слово (011) цього коду має найменший степінь, тому $G=\begin{pmatrix}0&1&1\\1&1&0\end{pmatrix}$.

$[7,4,3]$-код Хеммінга є еквівалентним циклічному коду, який породжує матриця

$$G=\begin{pmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&0&0&1&1&0&1\end{pmatrix}, \qquad (3.1)$$

оскільки матрицю (3.1) можна одержати з матриці G_1 (див. приклад 2.10) шляхом додавання рядків. Очевидно, що тут твірне слово $g=0001101$. □

Теорема про структуру циклічного коду. Лінійний циклічний код V складається з усіх слів, які діляться на твірне слово g цього коду.

Доведення. Нехай V лінійний циклічний код, a - довільне кодове слово, g - твірне слово коду V. За визначенням $\deg a \geq \deg g$. Розділимо a на g, маємо $\frac{a}{g}=b+\frac{r}{g}$, де $\deg d < \deg g$. Звідки

$$a=bg+r \quad \Rightarrow \quad r=a+bg.$$

Оскільки V циклічний код, то разом зі словом g у V входять всі його циклічні зсуви. Далі, оскільки V лінійний код, то він містить будь-які суми таких зсувів. Але в силу зауваження 3.1 добуток $gb=bg$ дорівнює сумі таких зсувів, тому $bg\in V$. Оскільки $a\in V$, то $a+bg\in V$, то $r\in V$. Оскільки степінь r менший степеня g, то з визначення g випливає, що $r=\overline{0}$. □

Розглянемо циклічний код довжини n, призначений для кодування слів a довжини k, і виправлення t помилок. Візьмемо якесь твірне слово $g = \underbrace{0\ldots01}_{k}\underbrace{*\ldots*}_{n-k}$ довжини n степеня $n-k$.

Кодове слово $b = \underbrace{a}_{\text{інформаційні символи}}\underbrace{*\ldots*}_{\text{перевірочні символи}}$ повинно ділитися на g. Один з способів цього досягти, це ділення $b' = \underbrace{a}_{k}\underbrace{0\ldots0}_{n-k}$ на g та обчислення залишку r. Тоді кодове слово

$$b = b' + r = \underbrace{a}_{k}\underbrace{r}_{n-k}. \quad (3.2)$$

Перевіримо, що отримане слово b дійсно є кодовим. Для цього за теоремою про структуру циклічного коду достатньо перевірити, що слово b ділиться на g. Дійсно

$$\frac{b'}{g} = e + \frac{r}{g} \Rightarrow b' = eg + r \Rightarrow eg = b' + r = b \Rightarrow b \text{ ділиться на } g.$$

3.1 Кодування і декодування циклічного коду

Нехай каналом зв'язку передається слово a довжини k:

– ***кодування***. Приписуємо до a праворуч нулі, щоб отримати слово $b' = \underbrace{a}_{k}\underbrace{0\ldots0}_{n-k}$ довжини n. Знаходимо залишок r ділення b' на g. Одержуємо кодове слово $b = b' + r$, яке відповідає слову a;

– ***декодування***. Нехай c прийняте слово. Ділимо c на g, обчислюємо залишок s. Якщо $s = \bar{0}$, то декодер вважає, що помилок не було. Якщо $s \neq \bar{0}$, то виявлена помилка. Для її виправлення декодер застосовує такий алгоритм:

вага $s \le t$	*вага* $s > t$
виправлене	1) здійснюємо J циклічних зсувів і ділення на g доти, доки не отримуємо вагу залишку

слово $c+s$	$s_j \le t$; 2) вектор c_j, отриманий при останньому зсуві, складаємо з його залишком від ділення на g $c_j + s_j$; 3) здійснюємо j зворотних циклічних зсувів праворуч.

Зауваження 3.2. Усі ділення в алгоритмі потрібні лише для знаходження залишків. Тому частки можна не шукати.

Знаходження залишку від ділення нескладно реалізувати. Ця простота – головна причина широкого застосування *циклічних кодів*. Схеми кодерів і декодерів для циклічних кодів дивись, наприклад, у [3, 8].

Приклад 3.7. Розглянемо циклічний код, еквівалентний [7,4,3]-коду Хеммінга (див. приклад 3.6). Цей код виправляє одну помилку $3 \ge 2t+1 \Rightarrow t=1$. Візьмемо твірне слово $g = 0001101$.

Нехай передане слово $a = 1100$. Допишемо праворуч нулі, щоб отримати слово $b' = 1100000$ довжини $n = 7$. Знаходимо залишок r від ділення b' на g:

$$\begin{array}{rl|l} & 1100000 & 1101 \\ + & 1101 & \\ \hline & 1000 & \\ + & 1101 & \\ \hline & 101 & = r. \end{array}$$

Кодове слово $b = 1100000 + 101 = 1100101$.

а) Якщо при передачі кодового слова b помилок не було, то буде прийняте слово $c = 1100101 = b$. Декодування: ділимо прийняте слово c на g і знаходимо залишок s (рисунок 3.1).

$$\begin{array}{rl|l} & 1100101 & 1101 \\ + & 1101 & \\ \hline & 1101 & \\ + & 1101 & \\ \hline & 0 & = s. \end{array}$$

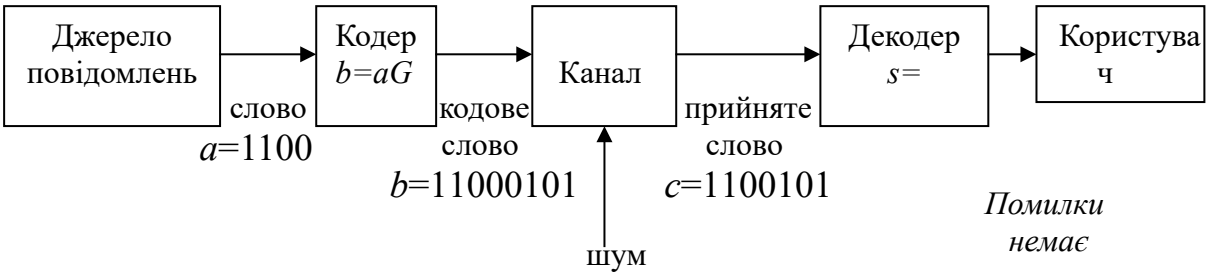


Рисунок 3.1

б) Якщо при передачі кодового слова b сталася помилка, наприклад, у 6-й координаті, то буде прийняте слово $c = 1100111$. Декодування: ділимо прийняте слово c на g і знаходимо залишок s (рисунок 3.2).

$$\begin{array}{r|l} 1100111 & 1101 \\ \underline{+\ 1101} & \\ 1111 & \\ \underline{+\ 1101} & \\ 10 & \end{array} = s \neq \overline{0}.$$

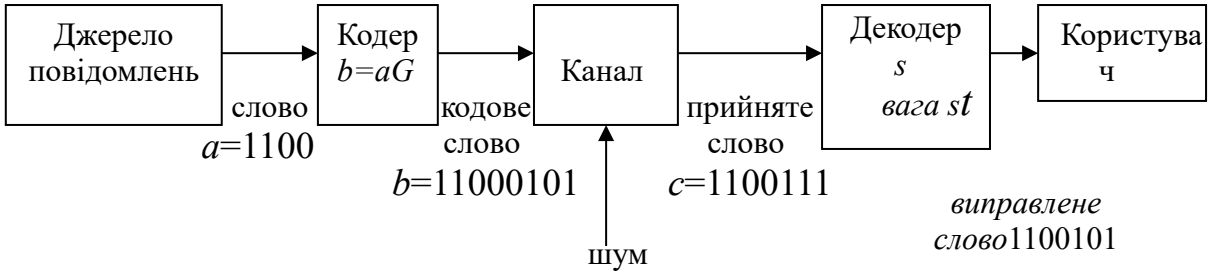


Рисунок 3.2

Оскільки *вага* $s = 1 \le t = 1$, то декодер замінить прийняте слово на $c + s = 1100111 + 10 = 1100101$. Помилку дійсно виправлено;

в) Якщо при передачі кодового слова b сталася помилка, наприклад, у першій координаті, то буде прийняте слово $c = 0100101$. Декодування: ділимо прийняте слово c на g і знаходимо залишок s (рисунок 3.3).

$$\begin{array}{r|l} + \ 0100101 & 1101 \end{array}$$

$$
\begin{array}{r|l}
1101 & \\
\hline
+\ 10001 & \\
\quad\ 1101 & \\
\hline
+\ 1011 & \\
\quad\ 1101 & \\
\hline
110 = s. &
\end{array}
$$

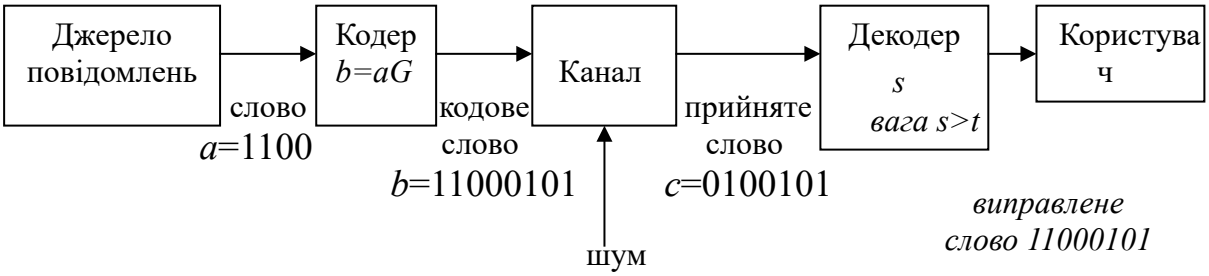


Рисунок 3.3

Декодер буде здійснювати циклічні зсуви і ділення на g доти, доки не отримуємо вагу залишку не більше $t=1$ ($вага\, s = 2 > t = 1$).

$$c = 0100101 \to 1001010 = c_1 \Rightarrow$$

$$
\begin{array}{r|l}
1001010 & 110 \\
+\quad\ \ & 1 \\
\quad\ 1101 & \\
\hline
+\ 100010 & \\
\quad\ 1101 & \\
\hline
+\ 10110 & \\
\quad\ 1101 & \\
\hline
+\ 1100 &
\end{array}
$$

$$\begin{array}{r}\hline 1101 \\ \hline 1\end{array} = s_1,$$

$$вага\, s_1 = 1 \le t = 1 \Rightarrow c_1 + s_1 = 1001010+1=1001011.$$

Декодер здійснить $j = 1$ зворотних циклічних зсувів праворуч $1001011 \to 1100101$. Помилку дійсно виправлено;

г) Якщо в каналі зв'язку сталося більше t помилок, то декодер циклічного кода може видати виправлене кодове слово, яке не збігається з відправленим. Дійсно, якщо при передачі кодового слова b сталася помилка, наприклад, у двох останніх координатах, то буде прийняте слово $c = 1100110$. Декодування: ділимо прийняте слово c на g і знаходимо і знаходимо залишок s (рисунок 3.4).

$$\begin{array}{rl|l} & 1100110 & 1101 \\ + & 1101 & \\ \hline & 1110 & \\ + & 1101 & \\ \hline & 11 & = s. \end{array}$$

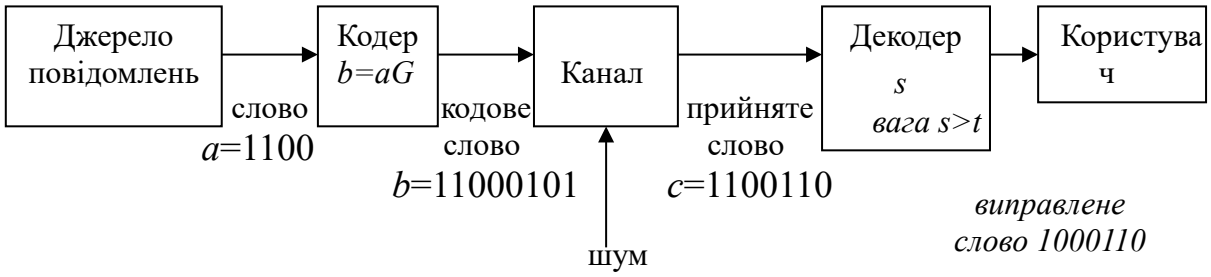


Рисунок 3.4.

Виправлене слово 1000110 є кодовим, але воно не збігається з відправленим. □

Приклад 3.8. Дуальний до [7,4,3]-коду Хеммінга з перевірочною матрицею (2.3) є циклічний код з породжуючою матрицею $G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$. Очевидно, що це [7,3,4]-код.

Теоретичні питання

1 Що таке циклічний зсув слова?

2 Дати визначення циклічного коду.
3 Як визначається степінь слова?
4 Дати визначення твірного слова.
5 Сформулювати теорему про структуру циклічного коду.
6 Алгоритм декодування циклічного коду.

Завдання для засвоєння теоретичного матеріалу

1 Помножити слова: 110×11; 1101×101.

2 Знайти залишок ділення: 101011:110; 110100:101.

3 Закодувати кодом, побудованим у прикладі 3.7, слова 1100, 1010. Декодувати прийняті слова: 1) за відсутності помилок; 2) при помилці в першій координаті; 3) при помилці у другій координаті.

4 Розв'язати запитання завдання 3 для коду, що відрізняється від коду, побудованого у прикладі 3.7, лише твірним словом: $g = 0001011$.

4 ЦИКЛІЧНІ КОДИ ТА МНОГОЧЛЕНИ

Для дослідження властивостей циклічних кодів виявилося можливим застосувати потужні алгебраїчні методи, зокрема теорію скінченних полів Галуа. Це привело до побудови багатьох класів циклічних кодів з достатньо простими алгоритмами декодування. Крім того, пошук кодів з «хорошими» параметрами в класі циклічних кодів виявився найуспішнішим. В цьому розділі ми наводимо деякі результати у цьому напрямку (доведення яких спирається на теорію полів Галуа, див. наприклад [3, 8]).

Поставимо у відповідність довільному слову довжини n многочлен степеня $\le n-1$ (з коефіцієнтами з $\{0,1\}$):

$$a_1 a_2 \dots a_n \to a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_n .$$

Приклад 4.1. $1011 \to x^3 + x + 1$, $0101 \to x^2 + 1$. □

Додавання і множення таких многочленів проводиться з урахуванням того, що $1+1=0$.

Приклад 4.2. $x^2 + x^2 = x^2(1+1) = 0,$

$$(x^2+1)+(x+1) = x^2 + x, \qquad (4.1)$$

$$(x+1)^2 = (x+1)(x+1) = x^2 + 1. \qquad (4.2)$$

□

Додаванню і множенню слів відповідає додавання і множення відповідних многочленів.

Приклад 4.3. Наприклад, замінивши многочлени (4.1) і (4.2) словами, отримаємо

$$\begin{array}{r} 101 \\ +\ 011 \\ \hline 110 \end{array} \rightarrow \quad x^2 + x \qquad \begin{array}{r} 11 \\ \times\ 11 \\ \hline 11 \\ 11 \\ \hline 101 \end{array} \rightarrow \quad x^2 + 1. \ \square$$

Якщо g – *твірне* слово циклічного коду, то відповідний многочлен $g(x)$ також називається твірним. Цей циклічний код будемо позначати V_g.

Теорема 4.1. Многочлен $g(x)$ породжує циклічний код довжини n тоді і тільки тоді, коли $x^n + 1$ ділиться на $g(x)$.

Зауважимо, що при доведенні цієї теореми та інших результатів цього напрямку важливу роль відіграє те, що множення на x многочлена відповідає циклічному зсуву відповідного слова.

Приклад 4.4. Оскільки

$$x^7 + 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1), \qquad (4.3)$$

то для кодів довжини $n = 7$ можна брати, наприклад, $g(x) = x^3 + x^2 + 1$ або $g(x) = (x+1)(x^3 + x + 1)$. □

З'ясуємо, які дільники многочлена $x^n + 1$ породжують код з більшою мінімальною відстанню.

Теорема 4.2. Нехай многочлен $g_1(x)$ породжує $V_{g_1} - [n, k_1, d_1]$-код, а многочлен $g_2(x)$ породжує $V_{g_2} - [n, k_2, d_2]$-код.

Якщо многочлен $g_2(x)$ ділиться на многочлен $g_1(x)$ ($g_1(x) \neq g_2(x)$), то

а) $k_2 < k_1$, б) $d_2 \geq d_1$.

Доведення. В силу теореми про структуру циклічного коду всі слова кода V_{g_2} діляться на g_2. Тому вони діляться також на g_1, отже,

$$V_{g_2} \subset V_{g_1}. \qquad (4.4)$$

З (4.4) та визначення мінімальної відстані коду отримуємо, що $d_2 \geq d_1$.

Оскільки код V_{g_2} містить 2^{k_2} слів, а код V_{g_1} містить 2^{k_1} слів, то, враховуючи (4.4), $k_2 < k_1$. □

Приклад 4.5. Нехай $n = 7$. З формули (4.3) та теореми 4.1 випливає, що многочлен $g_1(x) = x^3 + x^2 + 1$ породжує код V_{g_1} довжини 7. Твірний многочлен $g_1(x)$ відповідає твірному слову для коду з прикладу 3.6 $\Rightarrow$ $V_{g_1} - [7, 4, 3]$ - код, еквівалентний коду Хеммінга.

Нехай $g_2(x) = (x+1)\, g_1(x) = x^4 + x^2 + x + 1$. Твірний многочлен $g_2(x)$ відповідає твірному слову 10111 $\Rightarrow$ $V_{g_2} - [7, 3, 4]$-код, який еквівалентний коду з прикладу 3.8.

Тут $k_1 = 4 > 3 = k_2$, $d_1 = 3 < 4 = d_4$, що узгоджується з теоремою 4.2. □

Для вибору многочлена, який породжує циклічний код довжини n, треба знати розкладання на множники многочлена $x^n + 1$.

Як відомо, розкладання цілого числа на множники можна знайти, ділячи його на всі прості числа, які менші за нього. Для многочленів аналогом простих чисел є *незвідні* многочлени.

Визначення. Многочлен з коефіцієнтами з $\{0;1\}$ називається *незвідним*, якщо він ділиться тільки на себе та на 1. В противному разі многочлен називається *звідним*.

Приклад 4.6. Многочлен x^2+1 є звідним, оскільки, згідно з (4.2), він ділиться на $x+1$. □

Розкладання x^n+1 на незвідні множники наведено у таблиці 4.1.

Таблиця 4.1

n	Розкладання x^n+1 на незвідні множники
7	$(x+1)\underline{(x^3+x+1)(x^3+x^2+1)}$
9	$(x+1)\underline{(x^2+x+1)}(x^6+x^3+1)$
15	$(x+1)\underline{(x^2+x+1)(x^4+x+1)(x^4+x^3+1)}(x^4+x^3+x^2+x+1)$
17	$(x+1)(x^8+x^5+x^4+x^3+x+1)(x^8+x^7+x^6+x^4+x^2+x+1)$
21	$(x+1)\underline{(x^2+x+1)(x^3+x+1)(x^3+x^2+1)}(x^6+x^4+x^2+x+1)(x^6+x^5+x^4+x^2+1)$
23	$(x+1)(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)$
25	$(x+1)(x^4+x^3+x^2+x+1)(x^{20}+x^{15}+x^{10}+x^5+1)$
27	$(x+1)\underline{(x^2+x+1)}(x^6+x^3+1)(x^{18}+x^9+1)$
31	$(x+1)\underline{(x^5+x^3+1)(x^5+x^2+1)(x^5+x^4+x^3+x^2+1)(x^5+x^4+x^3+x+1)\cdot}$ $\underline{\cdot(x^5+x^4+x^2+x+1)(x^5+x^3+x^2+x+1)}$

У таблиці відсутні розкладання для парних n, оскільки за формулою $(x^m+1)(x^m+1)=x^{2m}+1$ будь-який многочлен парного степеня можна звести до добутку многочленів непарних степенів. Для $n=3,\ 5,\ 11,\ 13,\ 19,\ 29$ многочлени розкладаються за формулою $x^n+1=(x+1)(x^{n-1}+x^{n-2}+\ldots+x+1)$, тому такі розкладання теж не наведені у таблиці 4.1. Роль підкреслених многочленів з'ясується далі.

Теорема 4.3. Нехай α - корінь незвідного многочлена $f(x)$ степеня k, тобто $f(\alpha)=0$, і нехай $n=2^k-1$. Тоді:

1) $\alpha^n=1$;

2) якщо m - найменше ціле додатне число таке, що $\alpha^m=1$, то n ділиться на m.

Доведення. Доведемо теорему на прикладах.

1) нехай α корінь незвідного многочлена $f(x) = x^2 + x + 1$ степеня $k = 2$, тобто

$$\alpha^2 + \alpha + 1 = 0, \tag{4.5}$$

і нехай $n = 2^k - 1$=3. Використовуючи (4.5), можна скласти таблицю 4.2 відповідності степенів α многочленам з коефіцієнтами з $\{0,1\}$.

Таблиця 4.2

Степінь	Многочлен	Слово
-	0	00
α^0	1	01
α^1	α	10
α^2	$\alpha^2 = \alpha + 1$	11

Продовжуючи таблицю 4.2, одержимо $\alpha^n = \alpha^3 = \alpha^2 + \alpha = 1$, і п. 1 доведено.

Зауважимо, що з таблиці 4.2 видно, що для $0 < m < 3$ $\alpha^m \neq 1$;

2) нехай α корінь незвідного многочлена $f(x) = x^4 + x^3 + x^2 + x + 1$ степеня $k = 4$, і нехай $n = 2^k - 1$=15. Тоді $\alpha^5 = \alpha^4 + \alpha^3 + \alpha^2 + \alpha = 1$. Аналогічно зауваженню наприкінці першої частини доведення можна показати, що для $0 < m < 5$ $\alpha^m \neq 1$. Отже, найменше ціле m таке, що $\alpha^m = 1$, дорівнює 5, і п. 2 доведено.□

Наслідок. Нехай $f(x)$ незвідний многочлен степеня k, і нехай $n = 2^k - 1$, тоді, $x^n + 1$ ділиться на $f(x)$.

Доведення. Нехай α корінь многочлена $f(x)$, тоді, згідно з п. 2 теореми 4.3, $\alpha^n = \left(\alpha^m\right)^{n/m} = 1^{n/m} = 1 \Rightarrow \alpha^n + 1 = 0$. Тобто, α - корінь многочлена $x^n + 1$. Отже, будь-який корінь α многочлена $f(x)$ є коренем для $x^n + 1$. Тому $x^n + 1$ ділиться на $f(x)$. □

Визначення. Незвідний многочлен $f(x)$ степеня k називається *примітивним*, якщо для будь-якого його кореня α маємо $\alpha^m \neq 1$ при $0 < m < 2^k - 1$.

Можна довести, що незвідний многочлен $f(x)$ степеня k буде примітивним, якщо умова $\alpha^m \neq 1$ при $0 < m < 2^k - 1$ виконується хоча б для одного з його коренів.

Приклад 4.7. Нехай α корінь $f(x) = x^3 + x + 1$. Згідно з таблицею 4.1 $f(x)$ є незвідним многочленом степеня $k = 3$ $\Rightarrow n = 2^k - 1 = 7$, отже, за теоремою 4.3, $\alpha^7 = 1$. З таблиці 4.3 бачимо, що для $m < 7$ $\alpha^m \neq 1$.

Таблиця 4.3

Степінь	Многочлен	Слово
-	0	000
$1=\alpha^0$	1	001
α	α	010
α^2	α^2	100
α^3	$\alpha + 1$	011
α^4	$\alpha^2 + \alpha$	110
α^5	$\alpha^2 + \alpha + 1$	111
α^6	$\alpha^2 + 1$	101

Отже, $x^3 + x + 1$ − примітивний многочлен. □

З доведення теореми 4.3 випливає, що многочлен $x^2 + x + 1$ є примітивний, а многочлен $x^4 + x^3 + x^2 + x + 1$ не є таким.

Теорема 4.4. Нехай $f(x)$ примітивний многочлен степеня k, і нехай $n = 2^k - 1$.

1) многочлен $g(x) = \dfrac{x^n + 1}{f(x)}$ породжує циклічний $\left[2^k - 1; k; 2^{k-1}\right]$ - код;

2) всі ненульові слова цього кода є зсувами слова g, яке його утворює. І тому всі ненульові слова цього кода мають однакову вагу.

Приклад 4.8. Згідно з прикладом 4.7, многочлен $f(x) = x^3 + x + 1$ є примітивним, тому за теоремою 4.4 код, який

породжується $g(x)=\dfrac{x^7+1}{x^3+x+1}=x^4+x^2+x+1$ (тобто словом $g=0010111$), є $[7,3,4]$ - кодом.

Всі його ненульові слова одержується шляхом зсуву слова g. Наприклад, сума зсуву g та його подвійного зсува

$$\begin{array}{r} 0101110 \\ +\ 1011100 \\ \hline 1110010. \end{array}$$

знову є зсувом слова g. □

Наслідок. Можна побудувати циклічні коди, які знаходять і виправляють будь-яку, наперед задану, кількість помилок.

Дійсно, оскільки існують примітивні многочлени якого завгодно великого степеня k [8], то за схемою, наведеною у теоремі 4.4, можна побудувати циклічний код з мінімальною відстанню $d=2^k-1$, більшою за будь-яке наперед задане число. □

В таблиці 4.1 підкреслені примітивні многочлени.

Наостанок зауважимо, що наведені у таблицях 4.2, 4.3 відповідності між коренями незвідних многочленів і словами відіграють важливу роль у теорії циклічних кодів (див., наприклад, [3, 8, 10]).

Теоретичні питання

1 Який многочлен називається незвідним?

2 Який многочлен називається примітивним?

3 Як будується код, який знаходить і виправляє будь-яку, наперед задану, кількість помилок?

Завдання для засвоєння теоретичного матеріалу

1 Довести, що многочлен є $f(x)=x^4+x^3+1$ примітивним, а $f(x)=x^6+x^3+1$ таким не є.

2 За допомогою теореми 4.4 та таблиці 4.1 побудувати [15, 4, 8] - код.

СПИСОК ЛІТЕРАТУРИ

1 Аршинов Н.М. Коды и математика (рассказы о кодировании) / Н.М. Аршинов, Л.Е. Садовский. – М.: Наука, 1983. – 144 с.

2 Берлекэмп Э. Алгебраическая теорія кодирования / Э. Берлекэмп. – М.: Мир, 1971. – 479 с.

3 Блэйхут Р. Теория и практика кодов, контролирующих ошибки / Р. Блэйхут. – М.: Мир, 1986. – 576 с.

4 Бородин Л.Ф. Введение в теорию помехоустойчивого кодирования / Л.Ф. Бородин. – М.: Советское радио, 1968. – 408 с.

5 Бояринов И.М. Помехоустойчивое кодирование числовой информации / И.М. Бояринов. – М.: Наука, 1983. – 196 с.

6 Васильев К.К. Основы теории помехоустойчивых кодов / К.К. Васильев, Л.Я. Новосельцев, В.Л. Смирнов. – Ульяновск: УлГТУ, 2000. – 91 с.

7 Вишневецкий А.Л. Математические основы теории кодирования / А.Л. Вишневецький. – Харьков: ХВВАУРЭ, 1990.

8 Мак-Вильямс Ф. Дж. Теория кодов,исправляющих ошибки / Ф. Дж. Мак-Вильямс, Н Дж. А. Слоэн. – М.: Связь, 1979. – 744 с.

9. Основи дискретної математики / Ю.В. Капітонова, С.Л. Кривий, О.А. Летичевський, Г.М. Луцикий, М.К. Песурін. – К.: Наукова думка, 2002. – 579 с.

10. Питерсон У. Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон. – М.: Мир, 1976. – 595 с.

11. Теория электрической связи / К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко; Под. общ. ред. К.К. Васильева. – Ульяновск: УлГТУ, 2008. – 452 с.

12. Яглом А.М. Вероятность и информация / А.М. Яглом, И.М. Яглом. – 3-е изд. – М.: Наука, 1973. – 512 с.