Тема 25. Віддаль Геммінга. Умови надійності кодування в разі адитивних помилок. Лінійні коди. Коди Геммінга. Поняття про циклічні коди

План лекції

- **Віддаль Геммінга**
- **Умови надійності кодування в разі адитивних помилок**
- **Лінійні або групові коди. Коди Геммінга**
- **Поняття про циклічні коди**

У цій лекції ми узагальнимо міркування щодо кодів, стійких до завад, і розвинемо відповідну теорію. Тому запишемо систему перевірочних співвідношень, яку вивели в попередній лекції:

$$\begin{aligned} s_1 &= p_4 \oplus d_5 \oplus d_6 \oplus d_7 = 0, \\ s_2 &= p_2 \oplus d_3 \oplus d_6 \oplus d_7 = 0, \\ s_3 &= p_1 \oplus d_3 \oplus d_5 \oplus d_7 = 0, \end{aligned} \qquad (1)$$

Ця система дає змогу або визначити, що помилки немає, або однозначно визначити її місце.

Віддаль Геммінга

З методичних міркувань нам зараз зручно використовувати такі позначення. Елементи множини E_2^n – двійкові вектори довжини n – позначатимемо великими латинськими буквами X, Y, Z,…, а їхні компоненти – відповідними малими буквами з індексами. Зокрема, елементарні коди будуть позначатись як традиційно β_1, β_2, β_3, …, так і X, Y, Z, …, залежно від контексту.

Нормою $\|X\|$ двійкового вектора $X = x_1x_2\ldots x_n$ називають число, яке дорівнює кількості його одиничних компонент. Отже,

$$\|X\| = \sum_{i=1}^{n} x_i .$$

Припустимо, що в каналі зв'язку діє *джерело адитивних перешкод*, яке описують множиною $P(n, t)$. Елементи цієї множини - двійкові вектори-помилки $x_1x_2\ldots x_s$, у яких норма будь-якого фрагмента $x_ix_{i+1}\ldots x_{i+l-1}$ не більша ніж t, якщо довжина фрагмента $l \le n$, (тобто на n переданих поспіль двійкових символів припадає не більше ніж t помилок). Це означає, що коли на вході каналу зв'язку передано повідомлення α, то на виході може бути отримано будь-яке слово із множини $\{\alpha \oplus \gamma \mid \gamma \in P(n, t),\ l(\alpha)=l(\gamma)\}$.

Оскільки проблема локалізації інформації (тобто розділення закодованого повідомлення на елементарні коди) у моделі рівномірного кодування тривіальна, то виявлення помилок полягає в знаходженні незбігу локалізованої групи n символів ні з яким елементарним кодом. Якщо в результаті помилки елементарний код перейде в інший елементарний код, то помилку не буде виявлено. Іноді є можливим виправлення помилки. Якщо групу локалізовано правильно, то для цього необхідно й достатньо, щоб помилкова група була «синонімом» єдиного елементарного коду.

Канал зв'язку називають *надійним*, якщо будь-які помилки виявляються або виправляються відповідно до заданої мети декодування. Далі наведено головні положення побудови кодів, які забезпечують надійність найпростіших каналів зв'язку.

Віддаллю Геммінга називають функцію $\rho(X,Y)$ двох змінних визначену на множині E_2^n:

$$\rho(X,Y)=\sum_{i=1}^{n}|x_i-y_i|=\sum_{i=1}^{n}(x_i\oplus y_i) \tag{2}$$

(дорівнює кількості розрядів, у яких вектори X та Y не збігаються).

Скалярний добуток векторів $X,Y\in E_2^n$ визначають так:

$$\langle X,Y\rangle=\sum_{i=1}^{n}x_i y_i;$$

він дорівнює кількості розрядів, у яких X та Y збігаються й дорівнюють 1.

Легко перевірити такі співвідношення:

$$\rho(X,0)=\|X\|=\sum_{i=1}^{n}x_i \tag{3}$$

(тут 0 – n-вимірний вектор із нульовими компонентами),

$$\rho(X,Y)=\|X\oplus Y\| \tag{4}$$

($X\oplus Y$ – порозрядне додавання за mod 2),

$$\rho(X\oplus Z,Y\oplus Z)=\rho(X,Y), \tag{5}$$

$$\rho(X,Y)=\|X\|+\|Y\|-2\langle X,Y\rangle. \tag{6}$$

Для віддалі Геммінга виконуються *аксіоми метрики*:

$\rho(X,Y)\geq 0$, причому $\rho(X,Y)=0$ в тому й лише в тому випадку, якщо $X=Y$;

$\rho(X,Y)=\rho(Y,X)$;

$\rho(X,Y)+\rho(Y,Z)\geq\rho(X,Z)$ (нерівність трикутника).

Умови надійності кодування в разі адитивних помилок

Метрика Геммінга є зручним математичним поняттям для формулювання умов надійності кодування в разі адитивних помилок. Нехай схема $\sigma_{k,n}$: визначається кодом $V=\{\beta_1,\beta_2,\beta_3,\ldots,\beta_{2^k}\}$. Кодовою віддаллю для коду V називають величину

$$\rho(V)=min\{\rho(X,Y)\mid\ X,Y\in V,\ X\neq Y\}.$$

Теорема 1. Якщо в каналі зв'язку діє джерело адитивних перешкод $P(n, t)$, то

1) для виявлення будь-яких помилок необхідно й достатньо $\rho(V) > t$;

2) для виправлення будь-яких помилок необхідно й достатньо $\rho(V) > 2t$.

Зауваження. Іншими словами, код здатний виявляти будь-які комбінації з t і меншої кількості помилок тоді й лише тоді, коли його кодова віддаль є більшою ніж t; код здатний виправляти будь-які комбінації з t і меншої кількості помилок тоді й лише тоді, коли його кодова віддаль є більшою ніж $2t$.

Доведення. 1. Нехай $\rho(V) > t$. Якщо $X \in V$, $Y \in P(n, t)$, $Y \neq 0$, то, використовуючи спочатку (5), а потім (3), можемо записати $\rho(X, X \oplus Y) = \rho(0, Y) = \|Y\| \leq t$. Отже, $X \oplus Y \notin V$, і помилку виявлено.

Навпаки, нехай $\rho(X, Y) \leq t$ та $X, Y \in V$, $X \neq Y$. Тоді, використовуючи (4), маємо $\|X \oplus Y\| = \rho(X, Y) \leq t$, отже, $Z = X \oplus Y \in P(n, t)$. Звідси випливає, що $X \oplus Z = Y$, тобто помилку в елементарному коді Y не можна виявити.

2. Нехай $\rho(V) > 2t$. Якщо $X \in V$, $Z \in P(n, t)$, то X є єдиним елементарним кодом з V, який міг перейти внаслідок помилки в $X \oplus Z$. Справді, припустимо, що існує $Y \neq X$, такий, що $Y \in V$ та $Y \oplus Z_1 = X \oplus Z$ для деякого $Z_1 \in P(n, t)$. Додамо до обох частин останньої рівності $X \oplus Z_1$, тоді отримаємо $X \oplus Y = Z_1 \oplus Z$. Але, використовуючи (4), можемо записати $\|X \oplus Y\| = \rho(X, Y) > 2t$, а $\|Z_1 \oplus Z\| \leq \|Z_1\| + \|Z\| \leq 2t$. Суперечність.

Навпаки, нехай $\rho(X, Y) \leq 2t$ для деяких різних $X, Y \in V$. Тоді $\|X \oplus Y\| \leq 2t$ та існують такі Z_1, Z_2, що $\|Z_1\| \leq t$, $\|Z_2\| \leq t$ (тобто вони містяться в множині векторів-помилок $P(n, t)$) і $X \oplus Y = Z_1 \oplus Z_2$. До обох частин останньої рівності додамо $Y \oplus Z_1$, тоді матимемо $X \oplus Z_1 = Y \oplus Z_2 = W$. Отже, у разі отримання спотвореного елементарного коду W неможливо визначити, X чи Y було передано насправді.

Лінійні або групові коди. Коди Геммінга

Вектор, який відповідає елементарному коду, будемо називати *кодовим*.

Розглянуті в попередній лекції два приклади кодів мають таку особливість: сума двох кодових векторів є кодовим вектором.

Справді, для коду V із загальною перевіркою на парність, нехай $X, Y \in V$, тоді з урахуванням (4) та (6) можемо записати:

$$\|X \oplus Y\| = \|X\| + \|Y\| - 2\langle X, Y\rangle = 0 \pmod 2, \text{тобто } X \oplus Y \in V.$$

Код Геммінга, який задано системою перевірочних співвідношень (1), також має цю властивість: якщо вектори $(x_1, x_2, \ldots, x_n)$ та $(y_1, y_2, \ldots, y_n)$ – кодові, а, отже, вони – розв'язки системи (1), то їхня сума $(x_1 \oplus y_1, x_2 \oplus y_2, \ldots, x_n \oplus y_n)$ також розв'язок цієї системи, а тому – кодовий вектор.

Коди із зазначеною властивістю називають *лінійними* (або *груповими*).

Теорема 2. Для кодової віддалі лінійного коду виконується рівність

$$\rho(V) = \min\{\|X\| \mid X \in V,\ X \neq 0\}.$$

Доведення. Зазначимо, що нульовий вектор 0 міститься в будь-якому лінійному коді. Використовуючи спочатку (5), а потім – (4), можемо записати

$$\rho(V) = \min\{\rho(Y, Z) \mid Y, Z \in V,\ Y \neq Z\} =$$
$$= \min\{\rho(0, Y \oplus Z) \mid Y \oplus Z \in V,\ Y \oplus Z \neq 0\} = \min\{\|X\| \mid X \in V,\ X \neq 0\}.$$

Рівномірне кодування $\sigma_{k,n}$: $\alpha_i \to \beta_i$ (i=1, 2, 3, …, 2^k) називають *систематичним*, якщо можна виділити множину k розрядів $I=\{j_1, \ldots, j_k\} \subset \{1, 2, \ldots, n\}$, які називають *інформаційними*, так, що коли $\beta_i = x_1 \ldots x_n$ (i=1, 2, 3,…, 2^k), то $\alpha_i = x_{j_1} \ldots x_{j_k}$. Решту розрядів у такому разі називають *контрольними*. Усі лінійні коди – систематичні.

Коди, розглянуті нами раніше як приклади, – лінійні. Є дуже багато причин, з яких лінійні коди найважливіші в теорії кодування. Одна з них, як ми бачили, пов'язана із зручностями виявлення та виправлення помилок. Інша причина – це можливість компактного подання коду.

Справді, для лінійного коду непотрібно виписувати всі елементарні коди, адже вони повністю визначені системою лінійних рівнянь або матрицею цієї системи (*цю матрицю називають перевірочною матрицею*).

Повертаючись до розглянутих вище кодів, легко знайти їх перевірочні матриці. Так, для коду із загальною перевіркою на парність (наш перший приклад) маємо одне перевірочне рівняння

$$x_1 \oplus x_2 \oplus \ldots \oplus x_n = 0.$$

Відповідно до цього перевірочна матриця складається з одного рядка і має вигляд

$$H = [111 \ldots 1].$$

За теоремою 2 для коду v із загальною перевіркою на парність $\rho(V) = 2$, отже, можна виявити будь-яку помилку в каналі з джерелом перешкод $P(n,1)$

Перевірочна матриця двійкового коду Геммінга довжиною 7, тобто (7, 4)-коду Геммінга, як це випливає з рівнянь (1) (ми їх нижче продублювали)

$$\begin{aligned} s_1 &= p_4 \oplus d_5 \oplus d_6 \oplus d_7 = 0, \\ s_2 &= p_2 \oplus d_3 \oplus d_6 \oplus d_7 = 0, \\ s_3 &= p_1 \oplus d_3 \oplus d_5 \oplus d_7 = 0, \end{aligned} \qquad (1)$$

виглядає так:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (8)$$

Зазначимо, що (n, k)-коди Геммінга – лінійні, у загальному випадку мають довжину $n = 2^m–1$ ($m = 2, 3, \ldots$), виправляють одиночні помилки та обходяться мінімально можливою для цієї мети кількістю перевірок (ця кількість дорівнює m). Отже, перевірочна матриця коду Геммінга має розмір $m\times(2^m - 1)$. При цьому всі стовпці цієї матриці мають бути ненульовими й різними. Кожний стовпець – двійковий вектор довжиною m; усього є 2^mтаких векторів. Тому для побудови перевірочної матриці коду Геммінга довжиною $n = 2^m–1$ потрібно виписати (як стовпці цієї матриці) всі ненульові двійкові вектори довжиною m. Порядок цих стовпців довільний, але частіше їх упорядковують так, щоб кожний стовпець h_i був двійковим записом його номеру i (старші розряди зверху). Наприклад, для $m = 4$одержимо $n = 15$; перевірочна матриця (15, 11)-коду Геммінга виглядає так:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (9)$$

Контрольні (паритетні) розряди тут – степені двійки: 1, 2, 4, 8; їх чотири. Решта 11 розрядів – інформаційні, тому код і називають (15, 11)-кодом Геммінга.

За допомогою перевірочної матриці код Геммінга для виправлення помилок у каналі зв'язку з джерелом помилок $P(n,1)$ будують, використовуючи систему перевірочних рівнянь. Для (7, 4)-коду Геммінга – це система (1), яку можна записати так:

$$p_1\begin{pmatrix}0\\0\\1\end{pmatrix} \oplus p_2\begin{pmatrix}0\\1\\0\end{pmatrix} \oplus d_3\begin{pmatrix}0\\1\\1\end{pmatrix} \oplus p_4\begin{pmatrix}1\\0\\0\end{pmatrix} \oplus d_5\begin{pmatrix}1\\0\\1\end{pmatrix} \oplus d_6\begin{pmatrix}1\\1\\0\end{pmatrix} \oplus d_7\begin{pmatrix}1\\1\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix}. \qquad (10)$$

Задавши конкретні значення інформаційних бітів $d_3d_5d_6d_7$, одержимо систему трьох лінійних рівнянь відносно невідомих p_1, p_2, p_4 – контрольних розрядів коду.

Приклад. Закодуємо повідомлення 1001 за допомогою (7, 4)-коду Геммінга. Для цього запишемо *макет коду*, беручи до уваги розміщення контрольних бітів: $p_1p_2 1 p_4 001$. Для знаходження значень контрольних бітів використаємо систему (10) (доданки з нульовими коефіцієнтами випущені):

$$p_1\begin{pmatrix}0\\0\\1\end{pmatrix}\oplus p_2\begin{pmatrix}0\\1\\0\end{pmatrix}\oplus\begin{pmatrix}0\\1\\1\end{pmatrix}\oplus p_4\begin{pmatrix}1\\0\\0\end{pmatrix}\oplus\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix},$$

звідки $p_4\oplus 1=0$, $p_2\oplus 1\oplus 1=0$, $p_1\oplus 1\oplus 1=0$. Отже, $p_1=0$, $p_2=0$, $p_4=1$, і отримаємо такий код заданого повідомлення: 0011001.

Виправлення помилки. Нехай помила виникла, скажімо, у третьому розряді (7, 4)-коду Геммінга, тобто замість $p_1p_2d_3p_4d_5d_6d_7$ ми отримали $p_1p_2(d_3\oplus 1)p_4d_5d_6d_7$. Тоді систему перевірочних рівнянь запишемо так:

$$p_1\begin{pmatrix}0\\0\\1\end{pmatrix}\oplus p_2\begin{pmatrix}0\\1\\0\end{pmatrix}\oplus(d_3\oplus 1)\begin{pmatrix}0\\1\\1\end{pmatrix}\oplus p_4\begin{pmatrix}1\\0\\0\end{pmatrix}\oplus d_5\begin{pmatrix}1\\0\\1\end{pmatrix}\oplus d_6\begin{pmatrix}1\\1\\0\end{pmatrix}\oplus d_7\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}0\\1\\1\end{pmatrix},$$

бо

$$p_1\begin{pmatrix}0\\0\\1\end{pmatrix}\oplus p_2\begin{pmatrix}0\\1\\0\end{pmatrix}\oplus d_3\begin{pmatrix}0\\1\\1\end{pmatrix}\oplus p_4\begin{pmatrix}1\\0\\0\end{pmatrix}\oplus d_5\begin{pmatrix}1\\0\\1\end{pmatrix}\oplus d_6\begin{pmatrix}1\\1\\0\end{pmatrix}\oplus d_7\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix}.$$

Легко побачити, що отримали стовпчик перевірочної матриці, який відповідає помилковому розряду. Якщо розглянути цей стовпчик як двійковий запис числа, то отримаємо три – номер помилкового розряду, і цей розряд потрібно інвертувати. Ці міркування мають загальний характер – аналогічно виправляють помилку у будь-якому розряді.

Приклад. Нехай під час декодування повідомлення в (7, 4)-коді Геммінга одержано послідовність 0011011. Знаходимо

$$\begin{pmatrix}0\\1\\1\end{pmatrix}\oplus\begin{pmatrix}1\\0\\0\end{pmatrix}\oplus\begin{pmatrix}1\\1\\0\end{pmatrix}\oplus\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}1\\1\\0\end{pmatrix}.$$

Звідси доходимо висновку, що помилка виникла в шостому розряді коду; виправляємо її (інвертуємо помилковий розряд): $0011\overline{0}11=0011001$. Із відкоректованого коду виділяємо інформаційні розряди (третій, п'ятий, шостий і сьомий). У результаті отримаємо 1001.

Для загального випадку (n, k)-код Геммінга для каналу зв'язку з джерелом помилок $P(n,1)$ будують аналогічно. Наприклад, (15, 11)-код Геммінга можна побудувати скориставшись макетом $p_1p_2d_3p_4d_5d_6d_7p_8d_9d_{10}d_{11}d_{12}d_{12}d_{14}d_{15}$ і перевірочною матрицею (9).

Надлишковість у разі використання (n, k)-коду Геммінга зі зростанням n зменшується:

$$R=\frac{n}{k}-1=\frac{2^m-1}{2^m-m-1}-1=\frac{m}{2^m-m-1}.$$

Покажемо, що для (n, k)-коду Геммінга V кодова віддаль $\rho(V)\geq 3$. Справді, якщо $X\neq 0$ та $X\in V$ (тобто X – елементарний код), то $\|X\|\neq 1$, оскільки всі стовпці перевірочної матриці ненульові. Далі, $\|X\|\neq 2$, бо якщо X має одиниці тільки в двох розрядах, скажімо $x_i=x_j=1$, $i\neq j$, то $h_i\oplus h_j=0$ (тут h_i та h_j – стовпці перевірочної матриці H, 0 – вектор з нульовими компонентами), що рівносильно рівності $h_i=h_j$, а всі стовпці перевірочної матриці різні. Отже, найменша вага ненульового вектора з V не менша 3 і, за теоремою 2, $\rho(V)\geq 3$.

Тому для (7, 4)-коду Геммінга використовують також запис (7, 4, 3)-код, де перше число вказує загальну кількість бітів, друге – кількість інформаційних бітів, а третє – кодову віддаль. Аналогічно, для (15, 11)-кодом Геммінга використовують запис (15, 11, 3)-код Геммінга, явно вказуючи кодову віддаль.

Отже, коди Геммінга мають кодову віддаль 3, що означає, що код може виявити та виправити одну помилку (на n переданих поспіль бітів), але помилка в двох бітах нерозрізнена від коду з однією помилкою. Тому вони можуть виявляти помилки в двох бітах, тільки якщо виправлення не передбачено. Це означає, що для виправлення однієї помилки має бути впевненість, що це справді одна помилка, а не дві.

Додаванням іще одного контрольного біта кодову віддаль можна збільшити до 4. Це дає змогу виявляти та виправляти одну помилку, і в той самий час виявляти (але не виправляти!) дві помилки. (Також можна виявляти до 3 помилок, але не виправляти жодну з них.)

Розглянемо це питання докладніше для (7, 4)-коду Геммінга. Цей додатковий контрольний біт p_8 дописують у кінці коду для загальної перевірки парності, тобто елементарний код виглядатиме так: $p_1 p_2 d_3 p_4 d_5 d_6 d_7 p_8$, де $p_8 = p_1 \oplus p_2 \oplus d_3 \oplus p_4 \oplus d_5 \oplus d_6 \oplus d_7$. Неважко одержати й перевірочну матрицю цього коду; для цього до перевірочної матриці (8) справа приписуємо стовпець з нулів, після чого зверху приписуємо рядок з 1:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \qquad (11)$$

Можна показати, що кодова віддаль збільшилась з трьох у (7, 4)-коді до чотирьох у (8, 4)-коді, тому вживають запис (8, 4, 4)-код Геммінга (нагадаємо, що останнє число – це кодова віддаль).

Для виявлення та виправлення помилки у цьому коді обчислюють, як і раніше, вектор s, який дорівнює сумі добутків компонент коду на відповідні стовпці матриці (11).

Тут можливі дві основні ситуації: або цей вектор співпадає з одним із стовпців перевірочної матриці (11), або ні. Перша ситуація відповідає непарній кількості помилок в отриманому коді, а друга – парній. У першому випадку вважаємо, що відбулась помилка в одному біті, і її положення визначається номером стовпця, з яким співпав вектор s. У другому випадку, якщо $s \neq 0$, то вважаємо, що відбулося дві помилки; якщо ж $s = 0$, то вважаємо, що помилок при передаванні повідомлення не було.

Приклад. Нехай на виході каналу зв'язку одержано повідомлення у (8, 4)-коді Геммінга: а) 01100100, б) 01101100. Що можна твердити щодо повідомлень а) та б)?

а) Обчислимо вектор s:

$$s = \begin{pmatrix}1\\0\\1\\0\end{pmatrix} \oplus \begin{pmatrix}1\\0\\1\\1\end{pmatrix} \oplus \begin{pmatrix}1\\1\\1\\0\end{pmatrix} = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}.$$

Отже, помилка відбулася у сьомому розряді, зробимо корекцію коду

$$01100\underline{1}00 = 01100110$$

і виділимо інформаційні розряди, одержимо 1011.

б) Обчислимо вектор s:

$$s = \begin{pmatrix}1\\0\\1\\0\end{pmatrix} \oplus \begin{pmatrix}1\\0\\1\\1\end{pmatrix} \oplus \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \oplus \begin{pmatrix}1\\1\\1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

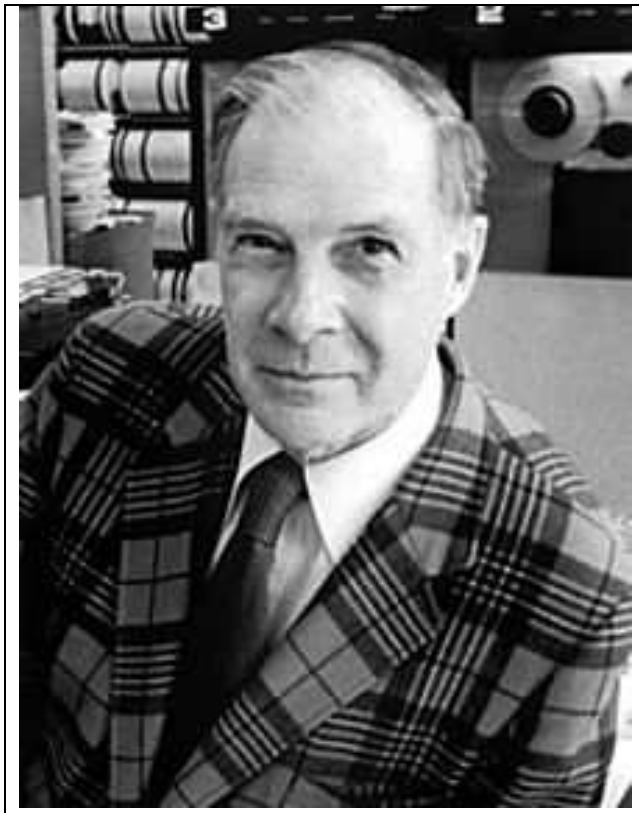
Такого стовпця в перевірочній матриці (11) немає; звідси доходимо висновку, що відбулося дві помилки, але їх положення знайти неможливо.

Аналогічно можна будувати й інші коди Геммінга з кодовою віддаллю чотири, наприклад, (16, 11, 4)-код.

Теорема. Для виправлення t (і меншої кількості) помилок і водночас виявлення s (і меншої кількості) помилок (причому $s \geq t$) необхідно й достатньо, щоб кодова віддаль задовольняла нерівність $\rho(V) > t + s$.

Якщо $n \neq 2^m - 1$, то будують *укорочений код Геммінга*. Його задають перевірочною матрицею, утвореною першими n стовпцями перевірочної матриці (n_1, k)-коду, де n_1 – найменше ціле із чисел, які більші ніж n, і мають вигляд $2^m - 1$. Очевидно, що всі попередні міркування правильні й для вкороченого коду Геммінга.

Така кодова система популярна в системах комп'ютерної пам'яті, де вона відома як SECDED (англ. «single error correction, double error detection» – «виправлення однієї помилки, виявлення двох помилок»). Особливо популярний (72, 64)-код – укорочений (127, 120)-код Геммінга плюс один додатковий контрольний біт.



Richard Wesley Hamming
11 лютого 1915 – 07 січня, 1998

Ричард Уеслі Геммінг (Хеммінг) – американський математик, чиї роботи мають велике значення для обчислювальної техніки й телекомунікацій. Народився в Чикаго, навчався в Чиказькому університеті, Університеті штату Небраска та Університеті штату Іллінойс в Урбана-Шампейн, де він написав докторську дисертацію з математики.

У квітні 1945 року він вступив у Манхеттенський проект у лабораторії в Лос-Аламосі, де він програмував машини IBM для обчислення розв'язків диференціальних рівнянь для фізичних проектів. У 1946 році перейшов до Bell Telephone Laboratories. Протягом наступних п'ятнадцяти років він був залучений майже до всіх важливих досягнень лабораторії.

Після відходу з Bell Labs 1976 року, Хеммінг працював в аспірантурі ВМС США в Монтереї, штат Каліфорнія, як ад'юнкт-професор і доцент з інформатики; він присвятив себе викладанню й написанню книг. З останньою лекцією він виступив у грудні 1997 року, усього за кілька тижнів до смерті від серцевого нападу 7 січня 1998 р.

Поняття про циклічні коди

Побудова циклічного коду.

Серед лінійних кодів особливо важливе значення мають так звані *циклічні коди*. По-перше, вони допускають ще більш компактний опис, ніж довільні лінійні коди. По-друге, наявні для лінійних кодів алгоритми кодування й декодування при застосуванні до циклічних кодів можна значно спростити. Більше того, для циклічних кодів існують свої особливі методи декодування, незастосовні до інших лінійних кодів. Нарешті, за своєю структурою ці коди ідеально пристосовані до реалізації в сучасних технічних пристроях.

Циклічні коди поширені як у техніці зв'язку, так і в комп'ютерних засобах зберігання інформації. У зарубіжних джерелах циклічні коди зазвичай називають перевіркою циклічним надлишковим кодом (CRC, Cyclic Redundancy Check).

Тут ми коротко розглянемо головні положення щодо циклічних кодів.

Термін *циклічний код* зумовлений тим, що кожна кодова комбінація, одержана циклічною перестановкою символів, також належить коду. Так, наприклад, циклічні перестановки комбінації 1000101 будуть також кодовими комбінаціями 0001011, 0010110, 0101100 і т.д.

Подання кодових комбінацій у вигляді поліномів $F(x)$ дає змогу задати однозначну відповідність між ними й звести дії над комбінаціями до дії над поліномами.

Приклад. Якщо кодове слово циклічного коду

$$101101001,$$

то йому відповідає поліном

$$F(x) = x^8 + x^6 + x^5 + x^3 + 1,$$

ми це будемо умовно записувати як

$$F(x) = x^8 + x^6 + x^5 + x^3 + 1 = x^8 + 0 + x^6 + x^5 + 0 + x^3 + 0 + 0 + 1,$$ що відповідає 101101001.

Важливі зауваження. *Побудова циклічних кодів ґрунтується на використанні незвідних у полі двійкових чисел поліномів. Множина елементів належить полю, якщо над ними можна виконувати операції додавання і множення за правилами даного поля, при цьому додавання й множення підлягають дистрибутивному закону. Для будь-яких поліномів у полі двійкових чисел виконуються такі правила.*

1. Додавання двійкових поліномів зводиться до додавання за mod 2 коефіцієнтів при однакових степенях змінної x. Звідси, зокрема, випливає можливість переносити доданки з однієї частини рівності в іншу без зміни знака – див далі формулу (*).

2. Множення здійснюється за звичним правилом множення степеневих функцій, проте одержані коефіцієнти при даному степені додаються за mod 2. Ділення здійснюється як звичне ділення поліномів, але операцію віднімання замінюють операцією додавання за mod 2.

3. Циклічна перестановка кодової комбінації еквівалентна множенню полінома $F(x)$ на x із заміною на одиницю змінної зі степенем, що перевищує степінь полінома.

Особливу роль у теорії циклічних кодів відіграють *незвідні поліноми* $G(x)$, тобто поліноми, які **не можуть** бути подані у вигляді добутку поліномів нижчих степенів. Вибирають поліноми $G(x)$ зі спеціальних таблиць з умови, щоби їхній степінь був не меншим ніж кратність можливих помилок.

Ідея побудови циклічного коду (n, m) зводиться до того, що поліном $Q(x)$, який подає інформаційну частину кодової комбінації, потрібно перетворити в поліном $F(x)$ степеня не більшого ніж $(n-1)$, який без остачі ділиться на поліном $G(x)$ (незвідний поліном) степеня $k = n - m$. Розглянемо послідовність операцій побудови циклічного коду.

Подаємо інформаційну частину кодової комбінації завдовжки m у вигляді полінома $Q(x)$. Для одержання k розрядів під контрольні символи множимо $Q(x)$ на одночлен x^k і одержуємо $Q(x)\cdot x^k$.

Ділимо поліном $Q(x)\cdot x^k$ на поліном $G(x)$ (незвідний поліном) степеня $k = n - m$, при цьому одержуємо результат ділення $C(x)$ такого ж степеня, що й $Q(x)$.

$$\frac{Q(x)\cdot x^k}{G(x)} = C(x) + \frac{R(x)}{G(x)},$$

де $R(x)$ – остача від ділення $Q(x)\cdot x^k$ на $G(x)$.

Множимо обидві частини на $G(x)$ і одержимо

$$Q(x)\cdot x^k = C(x)\cdot G(x) + R(x),$$

звідки

$$C(x)\cdot G(x) = Q(x)\cdot x^k + R(x). \qquad (*)$$

(Зверніть увагу, що при отриманні цієї формули використано попереднє зауваження щодо додавання двійкових поліномів.)

Тоді

$$F(x) = C(x)\cdot G(x) = Q(x)\cdot x^k + R(x).$$

Поліном $F(x)$ ділиться без остачі на $G(x)$, тобто є кодовим вектором циклічного (n, m) коду.

Приклад циклічного (7, 4)-коду з незвідним поліномом (вибирається зі спеціальної таблиці):

$$G(x) = x^3 + x + 1,\ k = 3.$$

Довжина кодового вектора дорівнює семи, з них чотири розряди інформаційні.

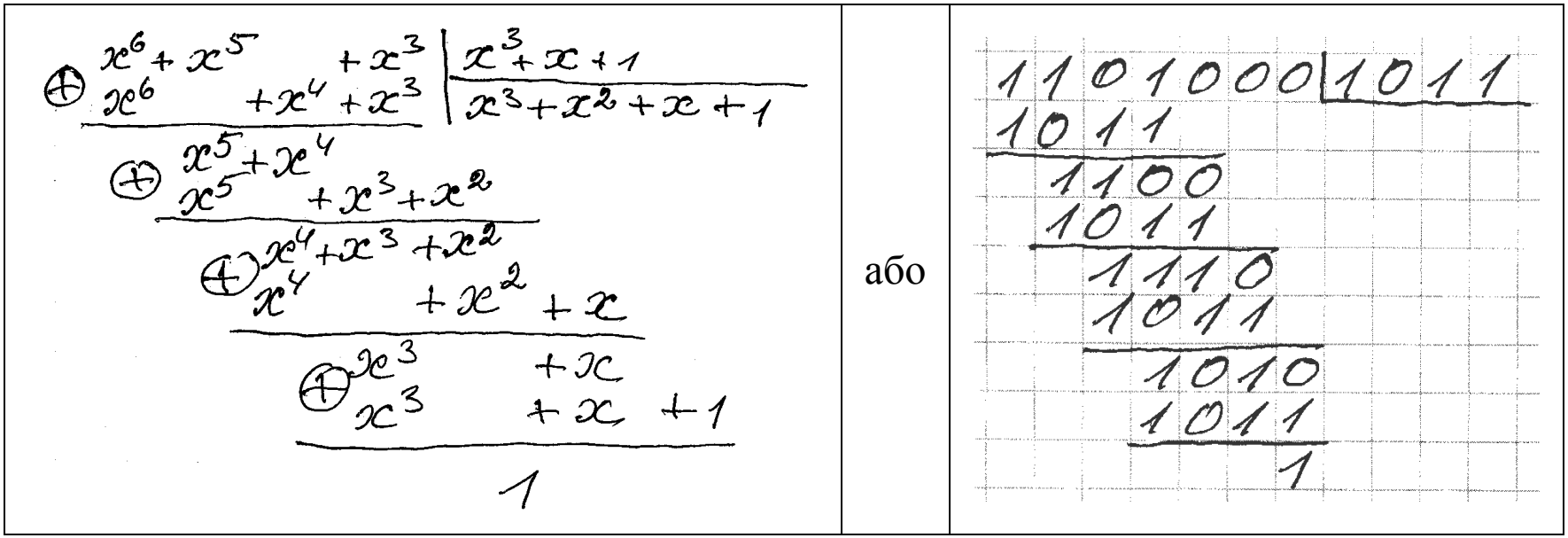
Як інформаційну частину кодової комбінації візьмемо 1101, її зобразимо поліномом

$$Q(x) = x^3 + x^2 + 1, \text{ що відповідає } 1101.$$

Помножимо цей поліном на x^k, де $k = 7 - 4 = 3$, це еквівалентне підвищенню степеня полінома на три:

$$Q(x)\cdot x^k = \left(x^3 + x^2 + 1\right)\cdot x^3 = x^6 + x^5 + x^3, \text{ що відповідає } 1101000.$$

Розділимо $Q(x)\cdot x^k = \left(x^3 + x^2 + 1\right)\cdot x^3 = x^6 + x^5 + x^3$ на $G(x) = x^3 + x + 1$ і сформуємо перевірочну частину кодової комбінації.



Унаслідок ділення одержуємо частку

$$C(x) = x^3 + x^2 + x + 1$$

і остачу – перевірочну частину кодової комбінації:

$$R(x) = 1, \text{ що відповідає } 001.$$

Тепер можемо отримати кодову комбінацію

$$F(x) = Q(x) \cdot x^k + R(x) = x^6 + x^5 + x^3 + 1, \text{ що відповідає } 1101001.$$

Ця комбінація 1101001 відправляється в канал зв'язку.

Приклад циклічного (9, 5)-коду з незвідним поліномом (вибирається зі спеціальної таблиці):

$$G(x) = x^4 + x + 1, \; k = 9 - 5 = 4.$$

Довжина кодового вектора – дев'ять розрядів, з них п'ять розрядів інформаційні.

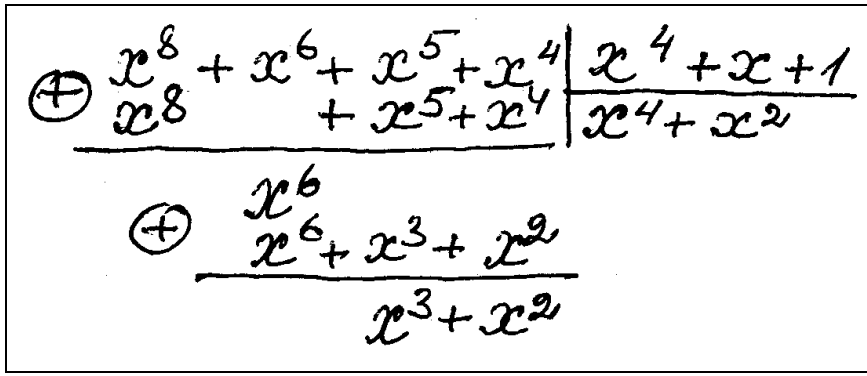
Як інформаційну частину кодової комбінації візьмемо 10111, тобто поліном

$$Q(x) = x^4 + x^2 + x + 1, \text{ що відповідає } 10111.$$

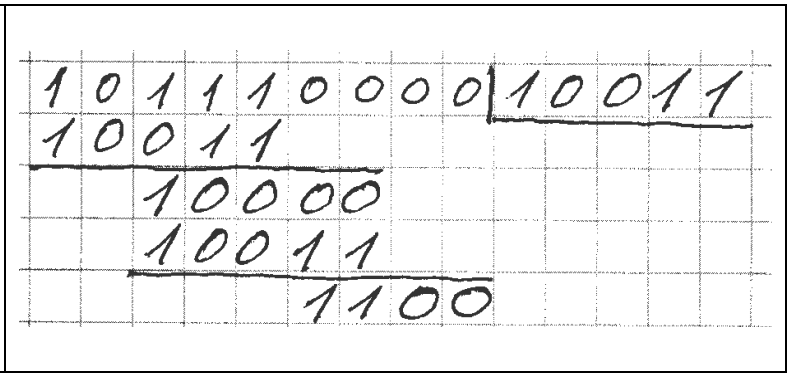
Множення $Q(x)$ на x^k еквівалентне підвищенню степеня полінома на k:

$$Q(x)\cdot x^k = \left(x^4 + x^2 + x + 1\right)\cdot x^4 = x^8 + x^6 + x^5 + x^4,$$ що відповідає 101110000.

Формування перевірочної частини кодової комбінації здійснюється в процесі ділення $Q(x)\cdot x^k$ на $G(x)$.



або



Унаслідок ділення одержуємо частку

$$C(x) = x^4 + x^2$$

і остачу – перевірочну частину кодової комбінації:

$$R(x) = x^3 + x^2,$$ що відповідає 1100.

Тепер можемо отримати кодову комбінацію

$$F(x) = Q(x)\cdot x^k + R(x) = x^8 + x^6 + x^5 + x^4 + x^3 + x^2,$$ що відповідає 101111100.

Ця комбінація 101111100 відправляється в канал зв’язку. Аналогічні операції виконуються для інших інформаційних комбінацій.

Виявлення і виправлення помилок у циклічному коді.

Виявлення і виправлення помилок здійснюють за остачами від ділення прийнятої кодової комбінації $F(x)$ на той же незвідний поліном $G(x)$, який використовувався для кодування. Нульова остача вказує на те, що прийнята комбінація є правильною. Остача від ділення свідчить про помилку, але не вказує, яку саме. Щоб знайти помилковий розряд і виправити його в циклічних кодах, потрібно зробити наступне.

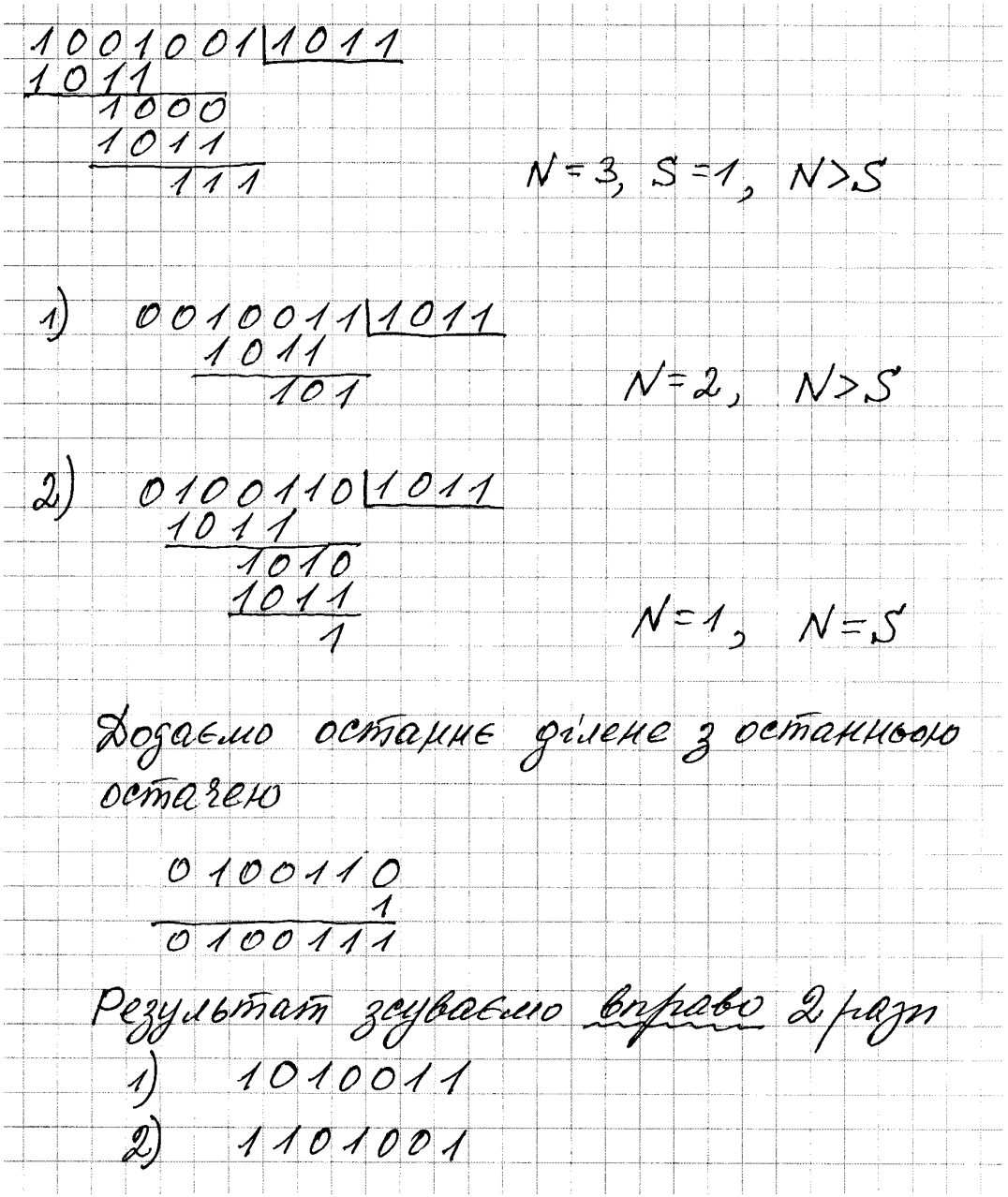
1. Прийняту комбінацію ділять на незвідний поліном, який використовувався для кодування.

2. Підраховують кількість одиниць в остачі (норму остачі N). Якщо $N \le S$, де S – допустиме число помилок, що виправляються даним кодом, то прийняту комбінацію додають за mod2 з отриманою остачею. Сума дасть виправлену комбінацію. Якщо $N > S$, то виконують наступне.

3. Ділять отриману циклічним зсувом комбінацію на той же незвідний поліном $G(x)$. Якщо в остачі $N \le S$, то додають за mod2 ділене з остачею. Після цього отриману комбінацію циклічно зсуваємо *вправо*. Отримана комбінація вже не містить помилок. Якщо після першого циклічного зсуву і наступного ділення остача залишається такою, що $N > S$, то виконують наступне.

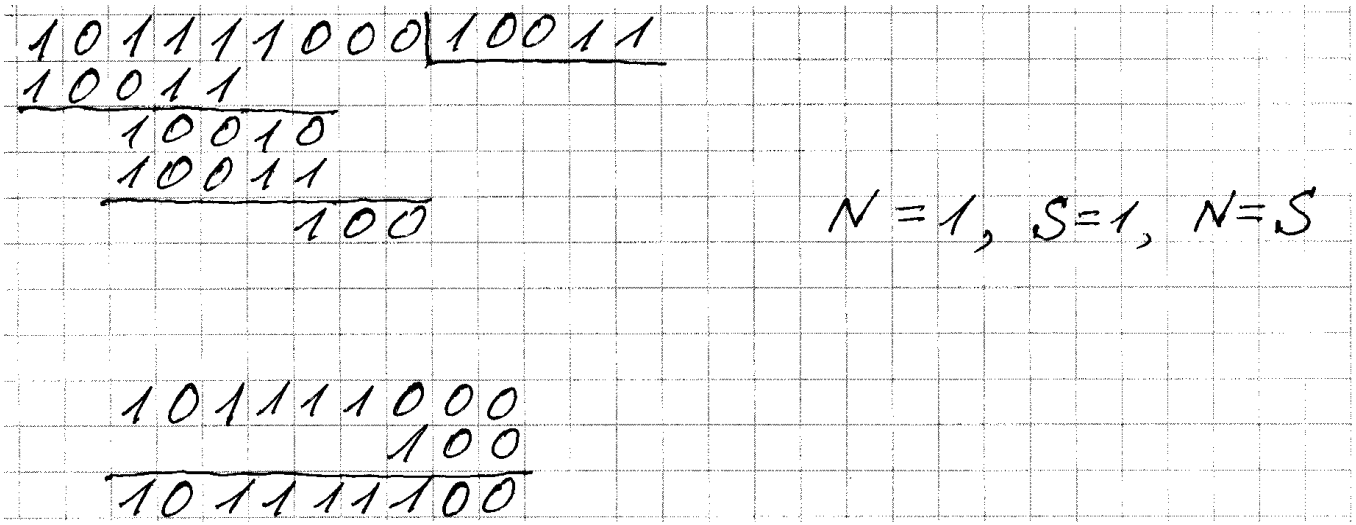
4. Процедуру пункту 3 повторюють, допоки не буде $N \le S$. У цьому випадку комбінацію, отриману як результат останнього циклічного зсуву, додають за mod2 до остачі, а після цього виконують циклічний зсув *вправо* на стільки розрядів, на скільки була зсунута підсумована з останньою остачею комбінація відносно прийнятої комбінації. Як результат отримаємо виправлену комбінацію.

Приклад. Розглянемо процес виправлення одиничної помилки у прийнятій кодовій комбінації на прикладі циклічного (7, 4)-коду: 1101001. Припустимо, що помилка відбулася у другому зліва символі, тобто прийнята комбінація виглядає так: 1001001. Далі діємо за описаним алгоритмом.



Остання комбінація відповідає переданій, тобто вже не містить помилки.

Приклад. Розглянемо процес виправлення одиничної помилки у прийнятій кодовій комбінації на прикладі циклічного (9, 5)-коду: 101111100. Припустимо, що помилка відбулася у третьому справа символі, тобто прийнята комбінація виглядає так: 101111000. Далі діємо за описаним алгоритмом.



Якщо норма остачі , отриманої як результат ділення прийнятої комбінації на незвідний поліном, є меншою чи рівною числу помилок, які виправляє код, то помилка виправляється шляхом безпосереднього додавання за mod2 остачі з прийнятою кодовою комбінацією.