Практичне завдання № 5

**ЛІНІЙНІ БЛОКОВІ КОДИ**

**Варіант 8**

**5.1.** Закодувати двійкову послідовність X кодами, що виявляють помилки. Виявити, у якому з отриманих повідомлень, закодованих першим та другим кодом є помилка.

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант | X | Коди | Послідовності для коду 1/2 |
| 8 | 00000000100 | ПрП, ІК | =0001100000111, =11000111010 /  =01101001000011, =0011010111001010 |

***ПрП (код з перевіркою на парність):***

***Code(Х)* = 000000001001**

=0001100000111; w(0001100000111) = 5; **помилки немає** =11000111010; w(11000111010) = 6; **є помилка**

***Інверсний код:* 0000000010011111111011**

***Інверсний код:***

**Y1 Y2**

01101001000011 0011010111001010

011**0**100 100**0**011 00110101 11001010

**є помилка помилки немає**

**5.2.** Закодувати двійкову послідовність X ітеративним кодом, здатним виявляти та виправляти однократні помилки, та визначити надлишковість коду. Показати процес виявлення та виправлення однократної помилки у прийнятій двійковій послідовності Y.

|  |  |  |
| --- | --- | --- |
| Варіант | Двійкова послідовність X | Прийнята послідовність Y |
| 8 | 011111000001 | 0111110001000001111101001 |

X = 011111000001

k = 12 → 3x4

Записуємо Х у вигляді матриці:

0111 | 1

1100 | 0

0001 | 1

1010 | 0

***Code(Х)*= 01110110000001010100**

ρ = = 0.4

**Виявлення та виправлення помилки:**

Y = 0111110001000001111101001

k = 25 → 5x5

Записуємо Y у вигляді матриці:

0111 | 1

1000 | 1

0000 | 0

1111 | 1

0100 | 1

Закодована послідовність має вигляд 0111110001000001**1**11101001,

де 1 - помилковий елемент, який заміняємо на 0 - 0111110001000001**0**11101001

**5.3.** Визначити, які з наведених двійкових послідовностей лінійного блокового (5,9)-коду містять помилку та виправити її, якщо відомо, що код побудований за твірною матрицею.

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант |  |  |  |
| 8 | 000111010 | 110101111 | 010101001 |

k = 5;

n = 9;

r = n – k = 4

Y = Code(X) = (, , , , , , , , ) = X・G

*Система перевірних рівнянь:*

Помилки немає:

**X = 11010**

Помилки є у :

**X = 10010**

Помилки є у :

**X = 00001**

**5.4.** Визначити, які з наведених двійкових послідовностей лінійного блокового (5,9)-коду містять помилку та виправити її, якщо відомо, що перевірна матриця коду має вигляд

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант |  |  |  |
| 8 | 001101010 | 110111100 | 011011001 |

(

(0101) помилка у - **01101**

001101010 001101010 001101010 001101010

000111000 011000100 101010010 110100001

000101000 001000000 001000010 000100000

(1001) помилка у - **11001**

110111100 110111100 110111100 110111100

000111000 011000100 101010010 110100001

000111000 010000100 100010000 110100000

(0000) помилок немає - **01101**

011011001 011011001 011011001 011011001

000111000 011000100 101010010 110100001

000011000 011000000 001010000 010000001

**5.5.** Закодувати кодами Хеммінга для заданих параметрів двійкову послідовність X, визначити надлишковість коду та показати процес виправлення однократної помилки (для коду з = 3) або виявлення будь якої двократної помилки (для коду з = 4 ) у прийнятих двійкових послідовностях Y.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Варіант |  | Y () |  | Y () |
| 8 | 0010 | 00000100011 | 100001 | 0110101000 |

**Закодовуємо кодом Хеммінга для послідовність X = 1010 :**

→ → → →

Будуємо матрицю 3x7:

**Code(Х) = 0101010**

**Закодовуємо кодом Хеммінга для послідовність X =** 100001**:**

→ → → → (+

Будуємо матрицю 4x10:

1010000101

**=** w() mod 2 = w(1010000101) mod 2 = 4 mod 2 = **0**

**Виправлення однократної помилки для коду з послідовності Y = 00000100011:**

→ → →

Будуємо матрицю 4x11:

Y =00000100011

Y \* ( = (0111) → помилка в :

Y = 00010100011;

X =001001

00000100011 00000100011 00000100011 00000100011

00000001111 00011110000 01100110011 10101010101

00000000011 00000100000 00000100011 00000000001

Dem 4

**Виправлення двократної помилки для коду з послідовності Y = 0110101000:**

Y = 0110101000

→ → →

**w() mod 2** = w(110101000) mod 2 = 4 mod 2 = 0 = → **OK**

\*( = (**0001**) → є дві помилки, бо на I етапі не було виявлено помилку

110101000

000000011

000000000

110101000

000111100

000101000

110101000

011001100

010001000

110101000

101010101

100000000