



Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления» – ИУ5

## **Отчет по домашнему заданию №2 по курсу Сети и телекоммуникации**

5

(количество листов)

Исполнитель

студент группы ИУ5-53Б

\_\_\_\_\_

Османов З. Ш.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 г.

Проверил

Доцент кафедры ИУ5

\_\_\_\_\_

Галкин В.А.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 г.

Москва, 2025 г.

## Полученный вариант

(ВАРИАНТ: 15)

15	10101010100	X [15,11]	C <sub>o</sub>
----	-------------	-----------	----------------

Расшифровка варианта:

$$\begin{aligned} 10101010100 &\rightarrow \text{информационный вектор, } m_1(x) \\ &= x^{10} + x^8 + x^6 + x^4 + x^2 \\ X [15,11] &\rightarrow \text{Код Хэмминга} \\ C_o &\rightarrow \text{обнаруживающая способность кода} \end{aligned}$$

### Ход работы

На языке Python было реализовано консольное приложение для условной передачи единственного информационного вектора – 10101010100. Для его кодирования, декодирования, проверки и попытки исправления ошибок использовался код Хэмминга [15,11].

Получившийся алгоритм:

1. Инициализация информационного вектора и параметров кода:
  - Информационный вектор: 10101010100 (11 бит)
  - Параметры кода Хэмминга:  $n = 15$ ,  $k = 11$ ,  $m = 4$
  - Проверочная матрица, построенная по позициям битов
2. Кодирование информационного вектора:
  - Информационные биты размещаются на позициях, не являющихся степенями двойки (3,5,6,7,9,10,11,12,13,14,15)
  - Проверочные биты вычисляются как суммы по модулю 2 определенных групп битов:
3. Инициализация счетчиков:
  - Счетчики ошибок всех разрядностей (от 1 до 15)
  - Счетчики обнаруженных ошибок всех разрядностей
  - Счетчики исправленных ошибок всех разрядностей
4. Цикл проверки всех возможных ошибок:
  - В цикле перебираются все возможные векторы ошибок от 000000000000001 до 111111111111111
  - Каждая ошибка складывается по модулю 2 с закодированным вектором
  - Вычисляется синдром ошибки через проверочные уравнения
  - Если синдром не равен нулю, увеличивается счетчик обнаруженных ошибок данной кратности
  - Если синдром соответствует одиночной ошибке, она исправляется путем инвертирования бита в указанной позиции
  - Исправленный вектор сравнивается с исходным закодированным вектором
  - При совпадении увеличивается счетчик исправленных ошибок

данной кратности

5. После завершения цикла выводится таблица, требуемая по заданию.

Вывод:

Согласно полученным результатам (рисунок1, см приложение), становится возможным подтвердить, что исправляющая способность кода Хэмминга [15,11] справляется только с ошибками единичной кратности, полностью их устраняя, однако попытки исправить ошибки большей кратности приводят только к большему изменению передаваемого полинома.

Таблица с результатами работы программы

i	Cni	Co	Nk	Ck, %	Примечание
1	15	15	15	100	Все одиночные ошибки исправлены
2	105	105	0	0	Все двойные обнаружены, но не исправлены
3	455	420	0	0	420 из 455 обнаружено
4	1365	1260	0	0	1260 из 1365 обнаружено
5	3003	2835	0	0	2835 из 3003 обнаружено
6	5005	4725	0	0	4725 из 5005 обнаружено
7	6435	6000	0	0	6000 из 6435 обнаружено
8	6435	6000	0	0	6000 из 6435 обнаружено
9	5005	4725	0	0	4725 из 5005 обнаружено

i	Cni	Co	Nk	Ck, %	Примечание
10	3003	2835	0	0	2835 из 3003 обнаружено
11	1365	1260	0	0	1260 из 1365 обнаружено
12	455	420	0	0	420 из 455 обнаружено
13	105	105	0	0	Все ошибки веса 13 обнаружены
14	15	15	0	0	Все ошибки веса 14 обнаружены
15	1	0	0	0	Ошибка веса 15 не обнаружена

#### **Литература:**

1. Галкин В.А., Григорьев Ю.А. Телекоммуникации и сети: Учеб. Пособие для вузов.-М.: Изд-во МГТУ им.Н.Э.Баумана, 2003
2. [http://www.opennet.ru/docs/RUS/inet\\_book/](http://www.opennet.ru/docs/RUS/inet_book/)

## Приложение

Исходный код программы доступен на удалённом репозитории:

[zagid1/Network](https://github.com/zagid1/Network)

i	Cni	Co	Nk	Ck, %
1	15	15	15	100.00
2	105	105	0	0.00
3	455	420	0	0.00
4	1365	1260	0	0.00
5	3003	2835	0	0.00
6	5005	4725	0	0.00
7	6435	6000	0	0.00
8	6435	6000	0	0.00
9	5005	4725	0	0.00
10	3003	2835	0	0.00
11	1365	1260	0	0.00
12	455	420	0	0.00
13	105	105	0	0.00
14	15	15	0	0.00
15	1	0	0	0.00

Рисунок 1 – Результат работы программы