

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет  
ИТМО»

Факультет программной инженерии и компьютерной техники

Отчёт о лабораторной работе №2  
Синтез помехоустойчивого кода  
Вариант №52

Выполнил:

Дядев Владислав Александрович, гр. Р3131

Проверила:

Авксентьева Е. Ю., к.п.н., доцент

Санкт-Петербург 2024

## Оглавление

Задание.....	3
Основные этапы вычисления .....	5
Схема декодирования кода Хэмминга (7;4).....	5
Задание №37 (Часть 1).....	5
Задание №69 (Часть 1).....	6
Задание №101 (Часть 1).....	7
Задание №21 (Часть 1).....	7
Схема декодирования кода Хэмминга (15;11) .....	8
Задание №52 (Часть 2).....	9
Вычисление минимального числа проверочных разрядов и коэффициента избыточности.....	10
Дополнительное задание .....	11
Заключение.....	13
Список использованных источников .....	14

## Задание

1. Определить свой вариант задания с помощью номера в ISU (он же номер студенческого билета). Вариантом является комбинация 3-й и 5-й цифр. Т. е. если номер в ISU = 12**3**4**5**6, то вариант = 35.
2. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
3. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
4. Показать, исходя из выбранных вариантов сообщений (по 4 у каждого – часть №1 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
5. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
6. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
7. Показать, исходя из выбранного варианта сообщений (по 1 у каждого – часть №2 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
8. Сложить номера всех 5 вариантов заданий. **Умножить полученное число на 4.** Принять данное число как число информационных разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.
9. **Дополнительное задание №1** (позволяет набрать от 86 до 100 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на любом языке программирования, которая на вход получает набор из 7 цифр «0» и «1», записанных подряд, анализирует это сообщение на основе классического кода Хэмминга (7;4), а затем выдаёт

правильное сообщение (только информационные биты) и указывает бит с ошибкой при его наличии.

Варианту №52 соответствуют задания №37, №69, №101, №21 для первой части, а также задание №52 для второй части.

## Основные этапы вычисления

### Схема декодирования кода Хэмминга (7;4)

Схема декодирования классического кода Хэмминга (7;4) представлена на рисунке 1.

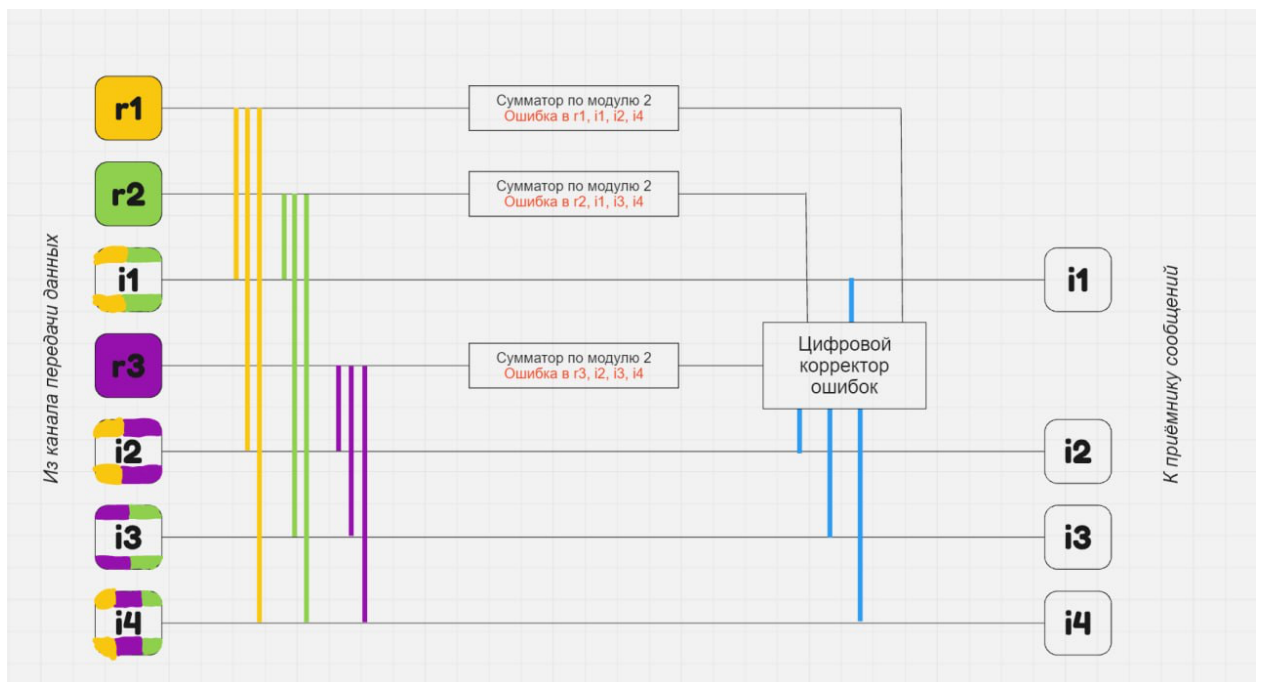


Рисунок 1 - Код Хэмминга (7;4)

### Задание №37 (Часть 1)

Дано сообщение: 1001010. Построим таблицу, в которой покажем, какие информационные биты покрывают проверочные биты (Таблица 1).

	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
	1	0	0	1	0	1	0
1	X		X		X		X
2		X	X			X	X
4				X	X	X	X

Таблица 1 - Задание №37

Посчитаем контрольные суммы:

$$1) \quad s1 = r1 \oplus i1 \oplus i2 \oplus i4 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$2) \quad s2 = r2 \oplus i1 \oplus i3 \oplus i4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$3) \quad s3 = r3 \oplus i2 \oplus i3 \oplus i4 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

Получим синдром:

$$S = (s1, s2, s3) = (1, 1, 0) \Rightarrow 011_2 = 3 \Rightarrow \text{ошибка в } i1$$

Правильное сообщение: 10**1**1010 (**1**1010 – без проверочных битов)

### Задание №69 (Часть 1)

Дано сообщение: 1110100. Построим таблицу, в которой покажем, какие информационные биты покрывают проверочные биты (Таблица 2).

	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
	1	1	1	0	1	0	0
1	X		X		X		X
2		X	X			X	X
4				X	X	X	X

Таблица 2 - Задание №69

Посчитаем контрольные суммы:

$$1) \quad s1 = r1 \oplus i1 \oplus i2 \oplus i4 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$2) \quad s2 = r2 \oplus i1 \oplus i3 \oplus i4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$3) \quad s3 = r3 \oplus i2 \oplus i3 \oplus i4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

Получим синдром:

$$S = (s1, s2, s3) = (1, 0, 1) \Rightarrow 101_2 = 5 \Rightarrow \text{ошибка в } i2$$

Правильное сообщение: 1110000 (1000 – без проверочных битов)

### Задание №101 (Часть 1)

Дано сообщение: 0011111. Построим таблицу, в которой покажем, какие информационные биты покрывают проверочные биты (Таблица 3).

	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
	0	0	1	1	1	1	1
1	X		X		X		X
2		X	X			X	X
4				X	X	X	X

Таблица 3 - Задание №101

Посчитаем контрольные суммы:

- 1)  $s1 = r1 \oplus i1 \oplus i2 \oplus i4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$
- 2)  $s2 = r2 \oplus i1 \oplus i3 \oplus i4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$
- 3)  $s3 = r3 \oplus i2 \oplus i3 \oplus i4 = 1 \oplus 1 \oplus 1 \oplus 1 = 0$

Получим синдром:

$$S = (s1, s2, s3) = (1, 1, 0) \Rightarrow 011_2 = 3 \Rightarrow \text{ошибка в } i1$$

Правильное сообщение: 0001111 (0111 – без проверочных битов)

### Задание №21 (Часть 1)

Дано сообщение: 0111001. Построим таблицу, в которой покажем, какие информационные биты покрывают проверочные биты (Таблица 4).

	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
	0	1	1	1	0	0	1
1	X		X		X		X
2		X	X			X	X
4				X	X	X	X

Таблица 4 - Задание №21

Посчитаем контрольные суммы:

- 1)  $s1 = r1 \oplus i1 \oplus i2 \oplus i4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$
- 2)  $s2 = r2 \oplus i1 \oplus i3 \oplus i4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$
- 3)  $s3 = r3 \oplus i2 \oplus i3 \oplus i4 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$

Получим синдром:

$$S = (s1, s2, s3) = (0, 1, 0) \Rightarrow 010_2 = 2 \Rightarrow \text{ошибка в } r2$$

Правильное сообщение: 0011001 (1001 – без проверочных битов)

### Схема декодирования кода Хэмминга (15;11)

Схема декодирования классического кода Хэмминга (15;11) представлена на рисунке 2.



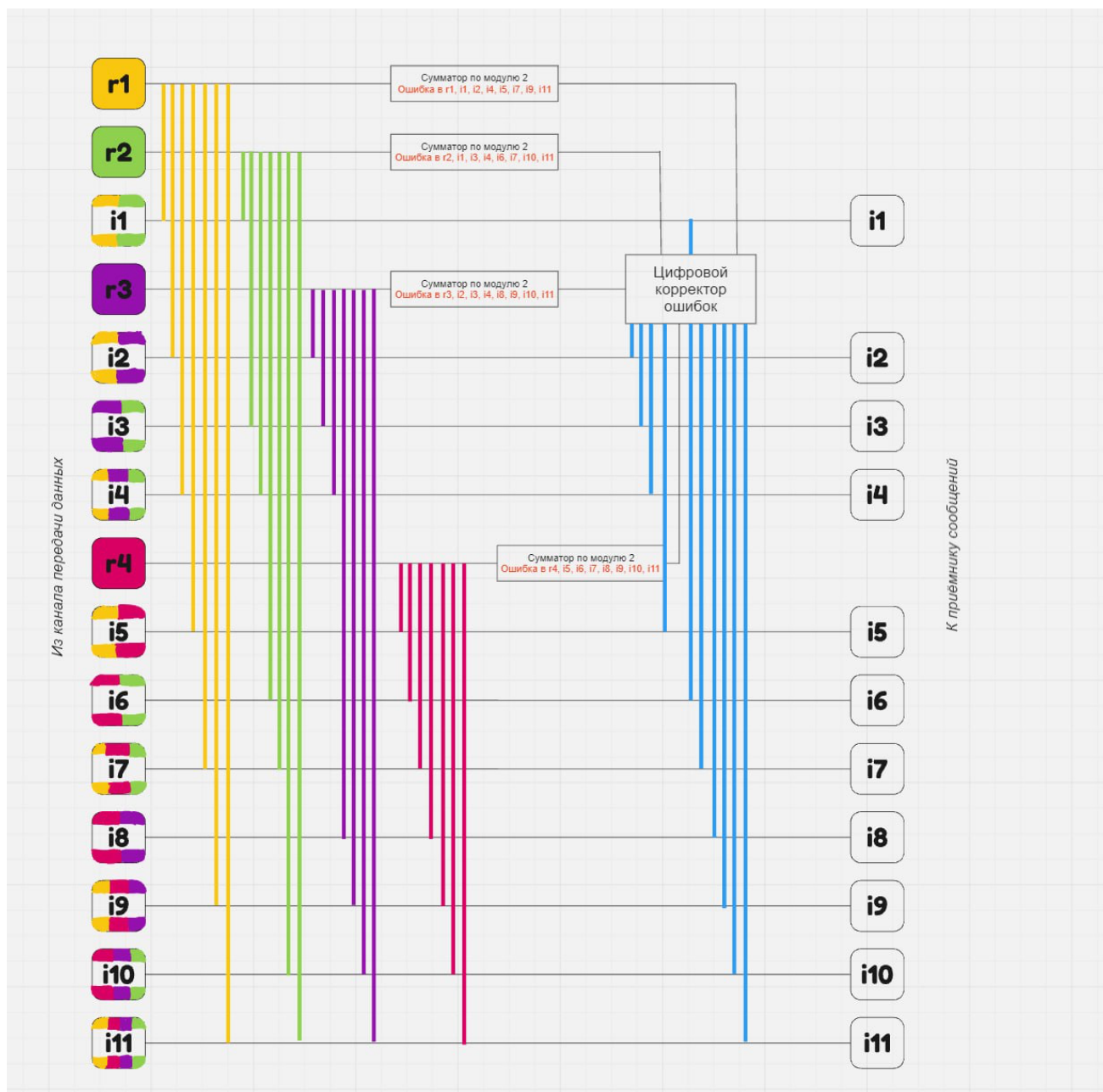


Рисунок 2 - Код Хэмминга (15;11)

### Задание №52 (Часть 2)

Дано сообщение: 010001101000011. Построим таблицу, в которой покажем, какие информационные биты покрывают проверочные биты (Таблица 5).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	r1	r2	i1	r3	i2	i3	i4	r4	i5	i6	i7	i8	i9	i10	i11
	0	1	0	0	0	1	1	0	1	0	0	0	0	1	1
1	X		X		X		X		X		X		X		X
2		X	X			X	X			X	X			X	X
4				X	X	X	X					X	X	X	X
8								X	X	X	X	X	X	X	X

Таблица 5 - Задание №52

Посчитаем контрольные суммы:

$$1) \quad s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$2) \quad s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$3) \quad s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$4) \quad s_4 = r_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$$

Получим синдром:

$$S = (s_1, s_2, s_3, s_4) = (1, 1, 0, 1) \Rightarrow 1011_2 = 11 \Rightarrow \text{ошибка в } i_7$$

Правильное сообщение: 010001101010011 (00111010011 – без проверочных битов)

**Вычисление минимального числа проверочных разрядов и коэффициента избыточности**

$$\text{Число информационных разрядов: } (37 + 69 + 101 + 21 + 52) * 4 = 1120$$

Минимальное число проверочных разрядов вычисляется по формуле:

$$2^r \geq r + i + 1 \leftrightarrow 2^r \geq r + 1121 \leftrightarrow \min r = 11$$

Коэффициент избыточности — это отношение числа проверочных разрядов к общему числу разрядов. В данном случае он равен  $11 / (11+1120) = 11 / 1131$  ( $\sim 0,0097$ ).

### Дополнительное задание

Код программы по обнаружению ошибки с помощью классического кода Хэмминга (7;4), написанный на языке Python, представлен на рисунке 3.

A screenshot of a code editor window with a dark background and light-colored text. The code is a Python function named 'hamming\_analyze(s)' that takes a string 's' as input. It first checks if 's' contains only '0' and '1' and has a length of 7. If not, it raises an AssertionError. Then, it defines symbols for the 7 bits: 'r1', 'r2', 'i1', 'r3', 'i2', 'i3', 'i4'. It converts the string into a list of integers 'digits'. Next, it calculates three syndrome bits: s1 (bits 0, 2, 4, 6), s2 (bits 1, 2, 5, 6), and s3 (bits 3, 4, 5, 6). These are combined into a syndrome tuple. If the syndrome is not (0, 0, 0), it finds the position of the error, flips the bit, and prints the corrected message. Otherwise, it prints the original message. Finally, it handles the AssertionError by printing a message about the input length and format. The script concludes by taking user input and calling the 'hamming\_analyze' function.

```
def hamming_analyze(s):
    try:
        # проверяем, что пользователь ввёл строку из 0 и 1
        assert all(x in "01" for x in s) and len(s) == 7

        # вводим обозначения для проверочных и информационных разрядов
        symbols = ("r1", "r2", "i1", "r3", "i2", "i3", "i4")

        # разделяем строку на отдельные цифры
        digits = [int(x) for x in s]

        # считаем контрольные суммы
        s1 = (digits[0] + digits[2] + digits[4] + digits[6]) % 2
        s2 = (digits[1] + digits[2] + digits[5] + digits[6]) % 2
        s3 = (digits[3] + digits[4] + digits[5] + digits[6]) % 2

        # составляем синдром и проверяем сообщение на ошибки
        syndrome = (s1, s2, s3)
        if syndrome != (0, 0, 0):
            # находим позицию ошибки
            num = int("".join(list(map(str, syndrome))[::-1]), 2)
            digits[num - 1] = int(not digits[num - 1])
            print("Исходное сообщение неверно!")
            print(f"Ошибка в бите {symbols[num - 1]}")
        else:
            print("Исходное сообщение верно!")

        # выводим правильное сообщение
        print(f"Правильное сообщение: {digits[2]}{digits[4]}{digits[5]}{digits[6]}\n")

    except AssertionError:
        # обрабатываем исключение
        print("Можно ввести только строку длиной 7, состоящую из 0 и 1.\n")

s = input("Введите сообщение: ")
hamming_analyze(s)
```

Рисунок 3 – Исходный код программы

Также исходный код программе представлен в репозитории на github (<https://github.com/Alvas07/ITMO/blob/main/1%20Informatics/Labs/Lab2/lab2.py>)

Результат работы программы представлен на рисунке 4.

```
Введите сообщение: 123
Можно ввести только строку длиной 7, состоящую из 0 и 1.

Введите сообщение: 1001010
Исходное сообщение неверно!
Ошибка в бите i1
Правильное сообщение: 1010

Введите сообщение: 1110100
Исходное сообщение неверно!
Ошибка в бите i2
Правильное сообщение: 1000

Введите сообщение: 0011111
Исходное сообщение неверно!
Ошибка в бите i1
Правильное сообщение: 0111

Введите сообщение: 0111001
Исходное сообщение неверно!
Ошибка в бите r2
Правильное сообщение: 1001
```

Рисунок 4 – Результат работы программы

## Заключение

В ходе лабораторной работы мы научились искать ошибки в сообщениях с помощью классических кодов Хэмминга (7;4) и (15;11), нарисовали схемы декодирования для них. Также мы вспомнили, как рассчитывать минимальное количество проверочных разрядов и коэффициент избыточности. Кроме того, нами была написана программа для анализа сообщения на ошибки с помощью классического кода Хэмминга (7;4).

### **Список использованных источников**

1. Коды и устройства помехоустойчивого кодирования информации / сост. Королев А. И.: 2002. с. 286
2. Основы цифровой радиосвязи. Помехоустойчивое кодирование: метод. Указания / сост. Д. В. Пьянзин. – Саранск: Изд-во Мордов. ун-та, 2009 – 16с.
3. Ромащенко А. Е., Румянцев А. Ю., Шень А. Заметки по теории кодирования. – 2-е изд., испр. и доп. – М.: МЦНМО, 2017. – 88 с.