Министерство науки и высшего образования Российской Федерации

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

*Факультет программной инженерии и компьютерной техники*

**Отчёт по лабораторной работе №2**

**на тему:**

**«Синтез помехоустойчивого кода»**

Вариант №78

Выполнил:

Студент группы P3134

Баянов Р.Д.

Проверил:

Балакшин П. В.

Доцент ф-та ПИиКТ

Санкт-Петербург

2022г

Оглавление

[**Задание** 3](#_Toc116846882)

[**Ход работы** 4](#_Toc116846883)

[**Вывод** 10](#_Toc116846884)

[**Источник информации** 10](#_Toc116846885)

# **Задание**

* 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. Необязательное задания для получения оценки «5» (позволяет набрать от 86 до 100 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на любом языке программирования, которая на вход из командной строки получает набор из 7 цифр «0» и «1», записанных подряд, анализирует это сообщение на основе классического кода Хэмминга (7,4), а затем выдает правильное сообщение (только информационные биты) и указывает бит с ошибкой при его наличии.

**Вариант 78**

Задание 1 - 

Задание 2 - 

# **Ход работы**

**Задание 1.1 (85)**

Полученное сообщение: 0 0 0 0 1 1 0

r1 = 0

r2 = 0

i1 = 0

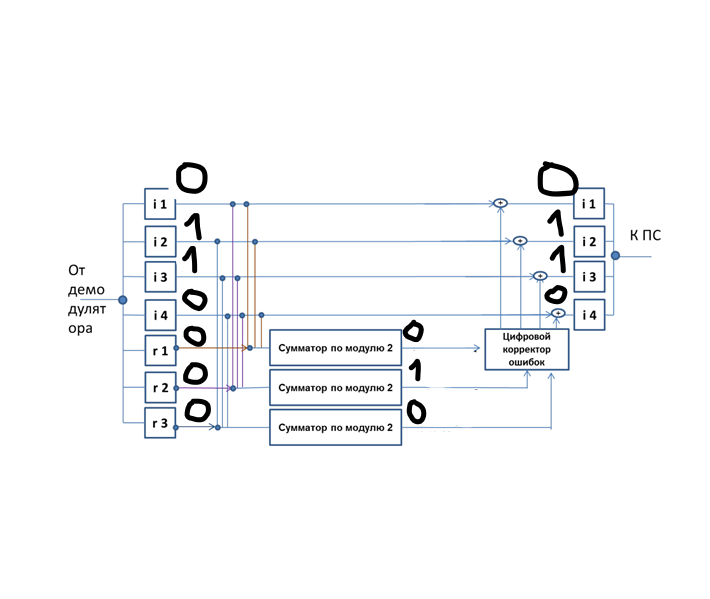
r3 = 0

i2 = 1

i3 = 1

i4 = 0

Строим схему кода Хэмминга:



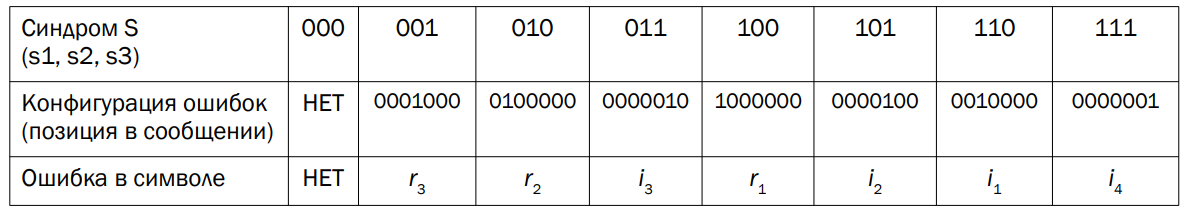
Найдём контрольный суммы s1, s2, s3:

s1 = r1 + i1 + i3 + i4 = 0 + 0 + 1 + 0 = 1

s2 = r2 + i1 + i2 + i4 = 0 + 0 + 1 + 0 = 1

s3 = r3 + i2 + i3 + i4 = 0 + 1 + 1 + 0 = 0

Синдром S = 010



По таблице можем понять, что ошибка находится в i1 (во втором проверочном бите)

Правильный код изначально выглядел так:

**0 1 0 0 1 1 0**

**Задание 1.2 (97)**

Полученное сообщение: 1 1 1 0 1 1 0

r1 = 1

r2 = 1

i1 = 1

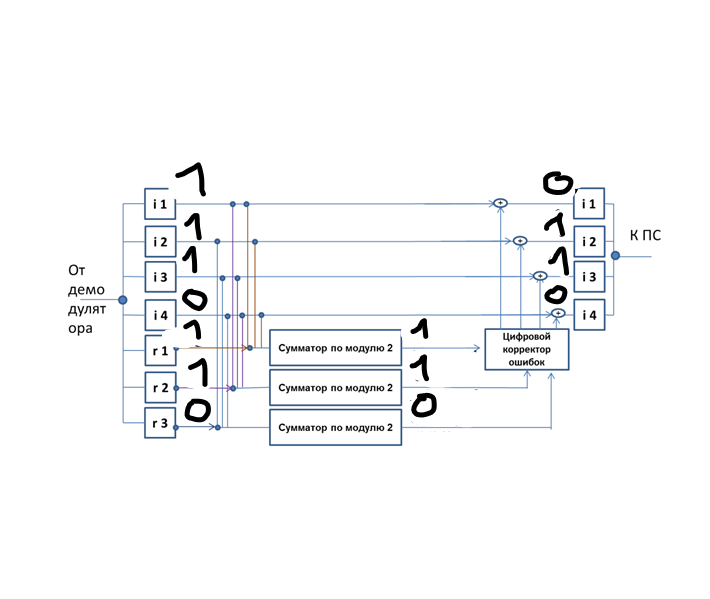
r3 = 0

i2 = 1

i3 = 1

i4 = 0

Строим схему кода Хэмминга:



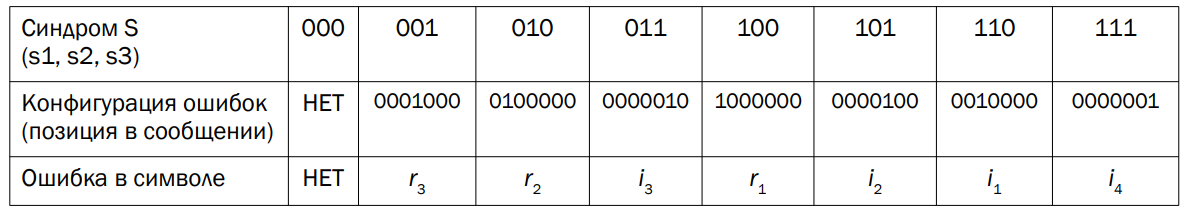
Найдём контрольный суммы s1, s2, s3:

s1 = r1 + i1 + i3 + i4 = 1 + 1 + 1 + 0 = 1

s2 = r2 + i1 + i2 + i4 = 1 + 1 + 1 + 0 = 1

s3 = r3 + i2 + i3 + i4 = 0 + 1 + 1 + 0 = 0

Синдром S = 110



По таблице можем понять, что ошибка находится в i1 (в первом информационном бите)

Правильный код изначально выглядел так:

**1 1 0 0 1 1 0**

**Задание 1.3 (22)**

Полученное сообщение: 1 0 0 0 0 0 1

r1 = 1

r2 = 0

i1 = 0

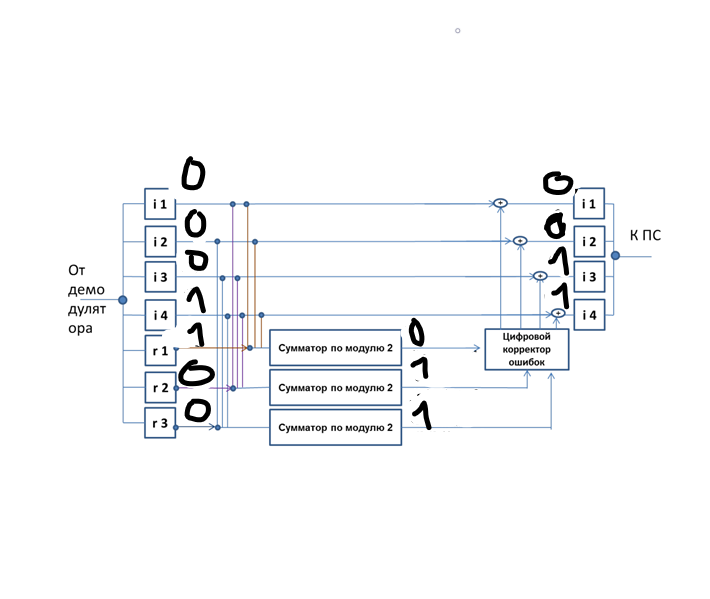
r3 = 0

i2 = 0

i3 = 0

i4 = 1

Строим схему кода Хэмминга:



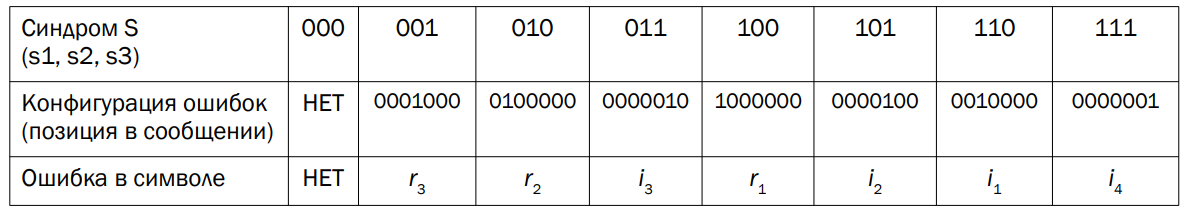
Найдём контрольный суммы s1, s2, s3:

s1 = r1 + i1 + i3 + i4 = 1 + 0 + 0 + 1 = 0

s2 = r2 + i1 + i2 + i4 = 0 + 0 + 0 + 1 = 1

s3 = r3 + i2 + i3 + i4 = 0 + 0 + 0 + 1 = 1

Синдром S = 011



По таблице можем понять, что ошибка находится в i3 (в третьем информационном бите)

Правильный код изначально выглядел так:

**1 0 0 0 0 1 1**

**Задание 1.4 (10)**

Полученное сообщение: 1 0 1 0 0 0 0

r1 = 1

r2 = 0

i1 = 1

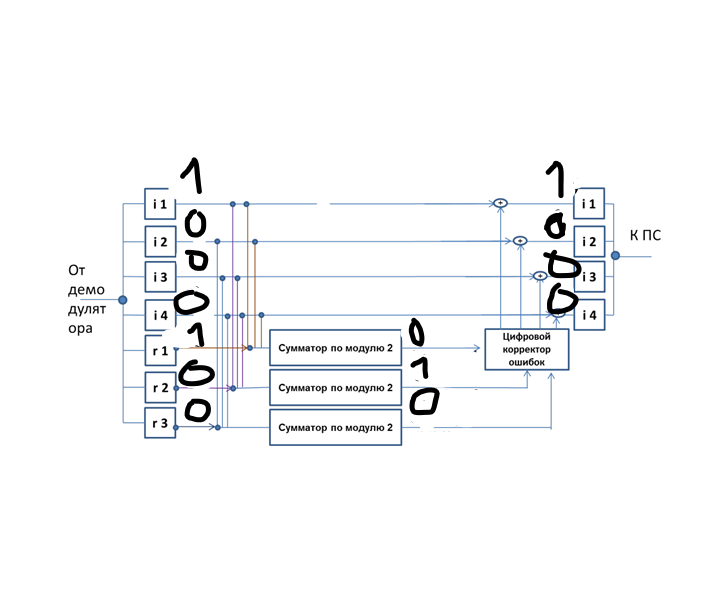
r3 = 0

i2 = 0

i3 = 0

i4 = 0

Строим схему кода Хэмминга:



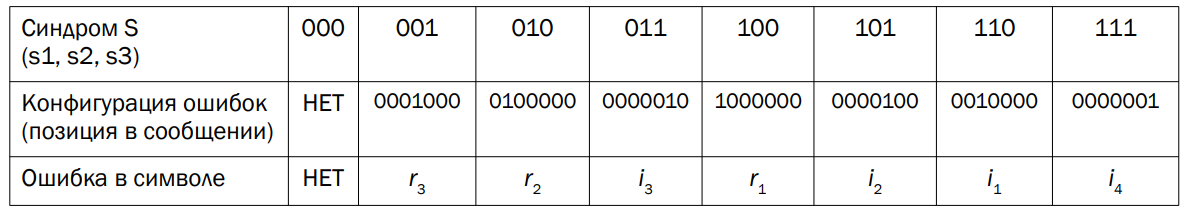
Найдём контрольный суммы s1, s2, s3:

s1 = r1 + i1 + i3 + i4 = 1 + 1 + 0 + 0 = 0

s2 = r2 + i1 + i2 + i4 = 0 + 1 + 0 + 0 = 1

s3 = r3 + i2 + i3 + i4 = 0 + 0 + 0 + 0 = 0

Синдром S = 010



По таблице можем понять, что ошибка находится в r2 (во втором проверочном бите)

Правильный код изначально выглядел так:

**1 1 1 0 0 0 0**

**Задание 2 (77)**

Полученное сообщение: 0 0 1 1 1 0 0 1 1 0 1 0 1 0 0

r1 = 0

r2 = 0

i1 = 1

r3 = 1

i2 = 1

i3 = 0

i4 = 0

r4 = 1

i5 = 1

i6 = 0

i7 = 1

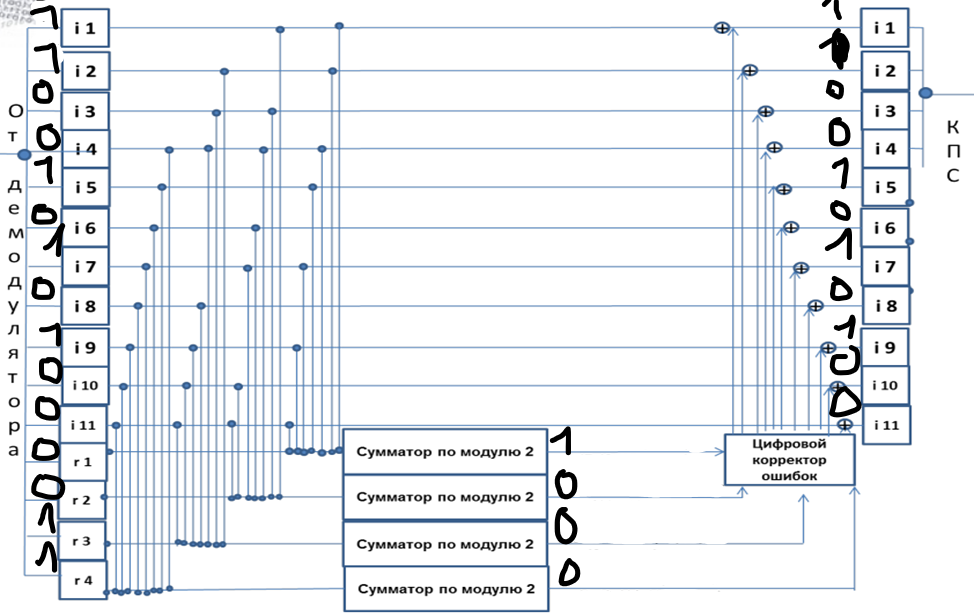
i8 = 0

i9 = 1

i10 = 0

i11 = 0

Строим схему кода Хэмминга:



Найдём контрольный суммы s1, s2, s3, s4:

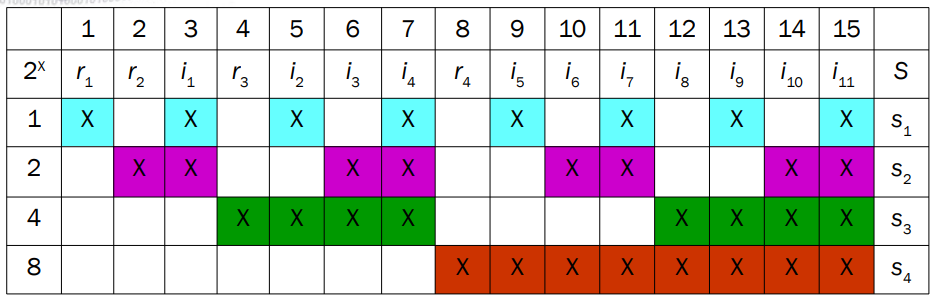
s1 = r1 + i1 + i2 + i4 + i5 + i7 + i9 + i11 = 0 + 1 + 1 + 0 + 1 + 1 + 1 + 0 = 1

s2 = r2 + i1 + i3 + i4 + i6 + i7 + i10 + i11 = 0 + 1 + 0 + 0 + 0 + 1 + 0 + 0 = 0

s3 = r3 + i2 + i3 + i4 + i8 + i9 + i10 + i11 = 1 + 1 + 0 + 0 + 0 + 1 + 0 + 0 = 1

s4 = r4 + i5 + i6 + i7 + i8 + i9 + i10 + i11 = 1 + 1 + 0 + 1 + 0 + 1 + 0 + 0 = 0

Синдром S = 1000



Проверяем, за какой бит отвечают только r1.

Им является r1 (первый проверочный бит).

Правильный код изначально выглядел так:

**0 0 1 1 1 0 0 1 1 0 1 0 1 0 0**

**Задание 3**

(85 + 97 + 22 + 10 + 77) \* 4 = 1164

Диапазон информационных разрядов, i = 1164

Определение минимального числа контрольных разрядов:

2r >= r + i + 1

2r >= r + 1165

1. r = 10

1024 >= 1175 (не выполняется)

1. r = 11

2048 >= 1176

Следовательно, минимальное количество проверочных разрядов r = 11

Коэффициент избыточности r/n, где (n = i + r):

11/1175 = 0,0093617

Ответ: r = 11, r/n = 0,0093617

**Задание 4**

Код программы:

def hamming\_code(message):  
 r1, r2, i1, r3, i2, i3, i4 = [int(x) for x in message]  
 return int((str((r1+i1+i2+i4) % 2) + str((r2+i1+i3+i4) % 2) + str((r3+i2+i3+i4) % 2))[::-1], 2)  
  
message = input()  
if len(message) == 7 and (message.count('0') + message.count('1') == 7):  
 wrong\_m = hamming\_code(message)  
 if wrong\_m == 0:  
 print('В сообщении нет ошибок')  
 else:  
 m = ''  
 for i in range(len(message)):  
 m += str((int(message[i]) + (i == (wrong\_m - 1)) \* 1) % 2)  
 print('В сообщении была ошибка в ' + str(wrong\_m) + ' бите. Правильное сообщение:' + m)  
  
  
else:  
 print('Неправильный формат ввода')

Примеры работы программы:

1)



2)



3)



# **Вывод**

В процессе выполнения лабораторной работы я познакомился с кодом Хэмминга. Научился пользоваться алгоритмом по определению ошибок в передаче информации и исправлять их. Узнал как находить коэффициент избыточности. Также я реализовал классический код Хэмминга (7,4) на Pyhton.

# **Источник информации**

1. Код Хэмминга. Пример работы алгоритма

<https://habr.com/ru/post/140611/>

1. Код Хэмминга – Википедия.

<https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A5%D1%8D%D0%BC%D0%BC%D0%B8%D0%BD%D0%B3%D0%B0>