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

Факультет программной инженерии и компьютерной техники

Лабораторная работа по информатике №2

Вариант №69

Выполнил:  
Студент группы P3106  
Мельник Фёдор Александрович

Проверил:

Балакшин П.В.,

Кандидат технических наук, доцент ФПиКТ

Санкт-Петербург, 2024

**Оглавление**

[Задание 3](#_Toc178970413)

[Основные этапы вычисления 4](#_Toc178970414)

[Часть №1 4](#_Toc178970415)

[Часть №2 6](#_Toc178970416)

[Часть №3 7](#_Toc178970417)

[Дополнительное задание 8](#_Toc178970418)

[Заключение 10](#_Toc178970419)

[Список использованных источников 11](#_Toc178970420)

# 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 = 123456, 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 выдает правильное сообщение (только информационные биты) и указывает бит с ошибкой при его наличии.

# Основные этапы вычисления

Для решения задач использовались материалы изданий «Помехоустойчивое кодирование»[[1]](#Литр1) и Помехоустойчивые коды[[2]](#Литр2)

## Часть №1

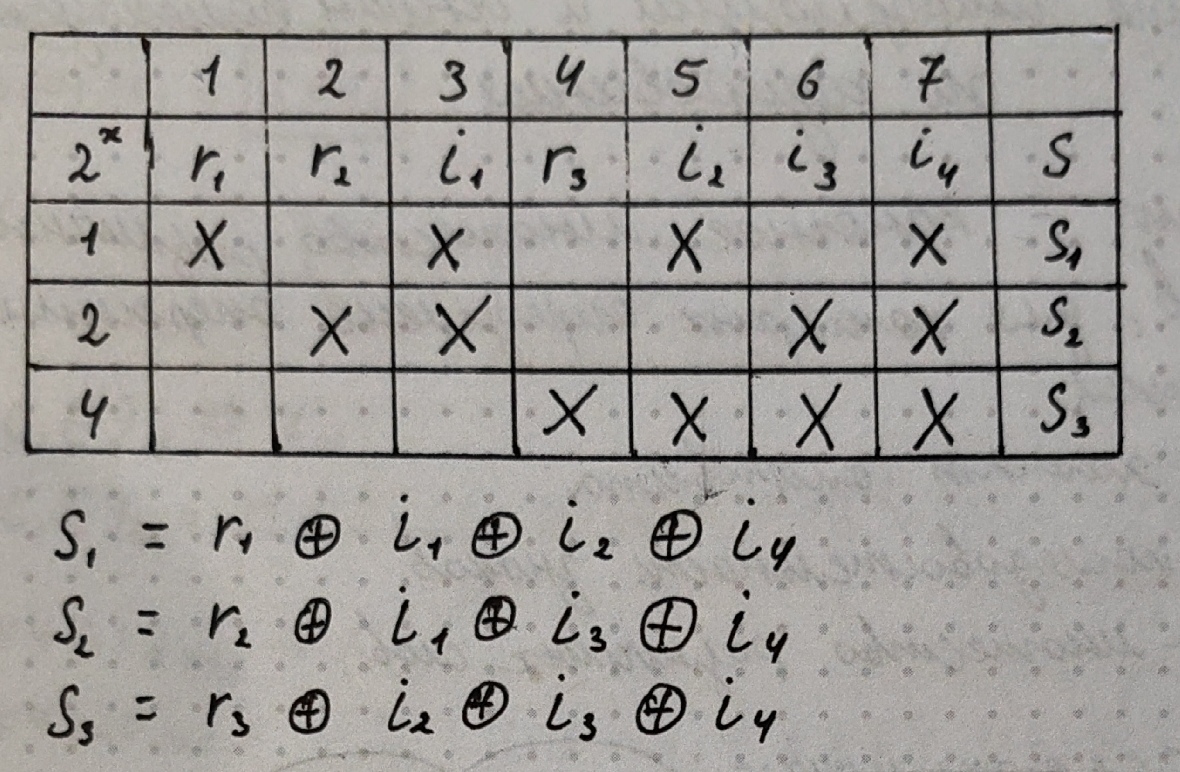
Схема декодирования и выполнения классического кода Хэмминга (7;4) представлена на [рисунке №1](#Рисунок1) и [рисунке №2](#Рисунок1_2) соотвественно.

Рисунок 1 - схема декодирования классического кода Хэмминга (7;4)

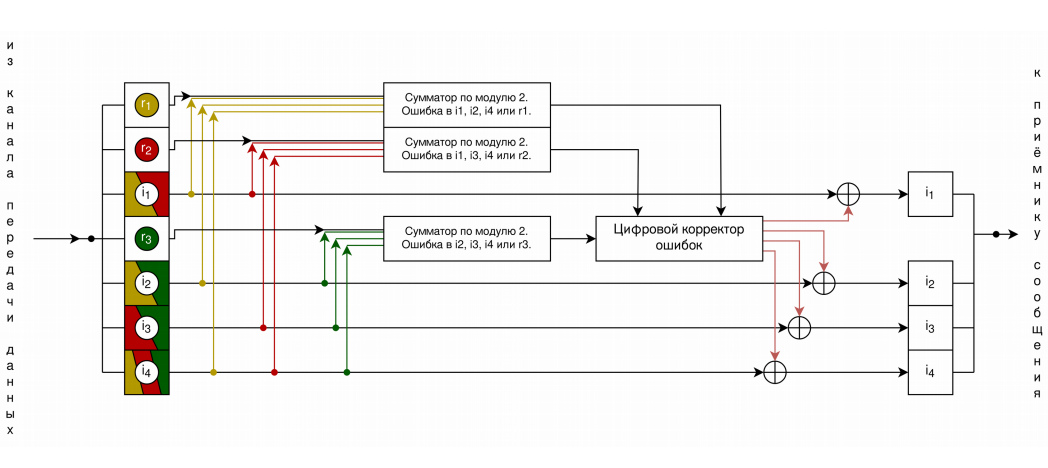


Рисунок 2 - схема выполнения классического кода Хэмминга (7;4)

Сообщение №1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 1 ⊕ 1 = 1

S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 1 = 0

Синдром S(S1, S2, S3) = 110 ⇒ ошибка в i1

Исправленный вариант: 1000011

Исходное сообщение без ошибки: 0011

Сообщение №2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 0 = 0

S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 1 ⊕ 0 = 0

Синдром S(S1, S2, S3) = 100 ⇒ ошибка в r1

Исправленный вариант: 1100110

Исходное сообщение без ошибки: 0110

Сообщение №3:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

Синдром S(S1, S2, S3) = 111 ⇒ ошибка в i4

Исправленный вариант: 1101001

Исходное сообщение без ошибки: 0001

Сообщение №4:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *r1* | *r2* | *i1* | *r3* | *i2* | *i3* | *i4* |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0

Синдром S(S1, S2, S3) = 010 ⇒ ошибка в r2

Исправленный вариант: 1110000

Исходное сообщение без ошибки: 1000

## Часть №2

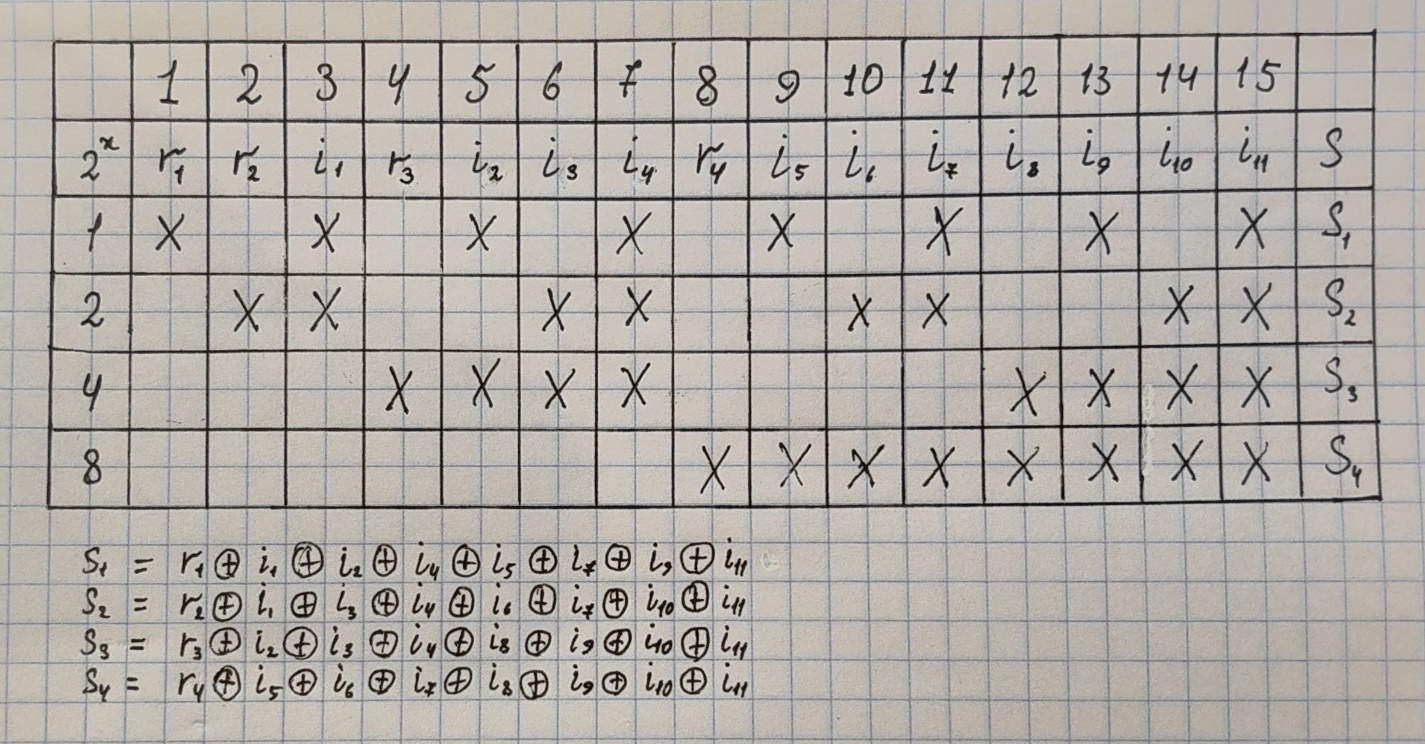
Схема декодирования и выполнения классического кода Хэмминга (15;11) представлена на [рисунке №3](#Рисунок2) и [рисунке №4](#Рисунок4) соответственно.

Рисунок 3 - схема декодирования классического кода Хэмминга (15;11)

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Рисунок 4 - Схема выполнения классического кода Хэммингка (15;11)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 | r4 | i5 | i6 | i7 | i8 | i9 | i10 | i11 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

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 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

Синдром S(S1, S2, S3, S4) = 1011 => ошибка в i9

Исправленный вариант: 001110001010000

Исходное сообщение без ошибки: 11001010000

## Часть №3

Число информационных разрядов (i) = (51 + 88 + 13 + 10 + 69) \* 4 = 924

2r >= r + i + 1 ⇒ 1024 >= 935 ⇒ Минимальное число проверочных разрядов (r) = 10

Коэффициент избыточности = = = ≈ 0,010707

## Дополнительное задание

Решение задания представлено на языке программирования Python. Исходный код программы можно найти по ссылке: <https://github.com/ldpst/itmo/blob/main/labs/sem1/inf/lab2/main.py>

from random import randint  
  
  
def task(a):  
 def get\_s(s):  
 if int(a[s[0]] ^ a[s[1]] ^ a[s[2]] ^ a[s[3]]) == 1:  
 return s[0] + 1  
 return 0  
  
 error = get\_s([0, 2, 4, 6]) + get\_s([1, 2, 5, 6]) + get\_s([3, 4, 5, 6]) - 1  
 name = ["r1", "r2", "i1", "r3", "i2", "i3", "i4"]  
 if error == -1:  
 print("Ошибок нет")  
 return  
 a[error] = int(not a[error])  
 print(f"Ошибка в бите {name[error]}. Сообщение без ошибок: {"".join(list(map(str, [a[2], a[4], a[5], a[6]])))}")  
  
  
def test():  
 print("1. Исходное сообщение: 1010011")  
 task(list(map(int, list("1010011"))))  
 print("2. Исходное сообщение: 0100110")  
 task(list(map(int, list("0100110"))))  
 print("3. Исходное сообщение: 1101000")  
 task(list(map(int, list("1101000"))))  
 print("4. Исходное сообщение: 1010000")  
 task(list(map(int, list("1010000"))))  
 for i in range(5, 11):  
 s = [randint(0, 1), randint(0, 1), randint(0, 1), randint(0, 1), randint(0, 1), randint(0, 1), randint(0, 1)]  
 print(f"{i}. Исходное сообщение: {"".join(list(map(str, s)))}")  
 task(s)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 test()

Результат выполнения программы на 4 числах из [части №1](#_Часть_№1) и 6 случайных числах представлен на [рисунке №5](#Рисунок3).

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

Рисунок 5 - Результат выполнения программы

# Заключение

В процессе выполнения лабораторной работы я познакомился с помехоустойчивым кодированием, узнал, какими преимуществами и недостатками оно обладает, научился строить таблицы декодирования классического хода Хэмминга и написал собственную программу на языке программирования Python для проверки сообщения на ошибки и исправления их.

# Список использованных источников

1. Смирнов, В. А. Помехоустойчивое кодирование / В. А. Смирнов. — М.: БИНОМ. Лаборатория знаний, 2014. — 280 с.
2. Шевченко, А. А. Помехоустойчивые коды / А. А. Шевченко. — М.: Горячая линия — Телеком, 2018. — 350 с.