Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет**

по лабораторной работе №2

“ Синтез помехоустойчивого кода”

**Вариант: 89**

Выполнил:

Тахватулин Михаил Витальевич

Группа: P3107

Преподаватель:

Белозубов Александр Владимирович

г. Санкт-Петербург

2022

Оглавление

[Задание 1. 2](#_Toc117823351)

[Задание 2. 4](#_Toc117823352)

[Задание 3. 5](#_Toc117823353)

[Задание 4. 5](#_Toc117823354)

[Заключение. 6](#_Toc117823355)

# 

# **Задание 1.**

Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения. 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.

Изображение выглядит как стол

Автоматически созданное описание

Первая строка таблицы – это номер сообщения и само сообщение (по заданию). Следующая строка – это нумерация битов сообщения (номер неправильного бита выделен темным цветом). Первый столбец показывает проверочный бит (степень двойки).

Чтобы определить ошибочный бит мы должны выполнить операцию сложения по модулю 2 для всех проверочных битов (они выделены оранжевым цветом). Результат записан в ячейки желтого цвета.

Изображение выглядит как текст

Автоматически созданное описание

Запишем синдромы в обратном порядке: 0012 = 110. Значит ошибка в 1 бите. Исправим и запишем ответ:



Аналогичные операции применяем для остальных сообщений.

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как текст

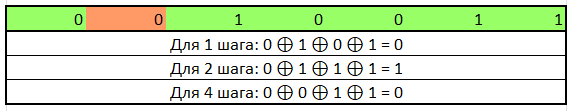
Автоматически созданное описание

Запишем синдромы в обратном порядке: 1112 = 710. Значит ошибка в 7 бите. Исправим и запишем ответ:



Изображение выглядит как стол

Автоматически созданное описание

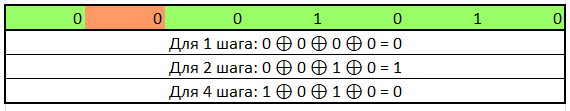


Запишем синдромы в обратном порядке: 0102 = 210. Значит ошибка в 2 бите. Исправим и запишем ответ:



Изображение выглядит как стол

Автоматически созданное описание

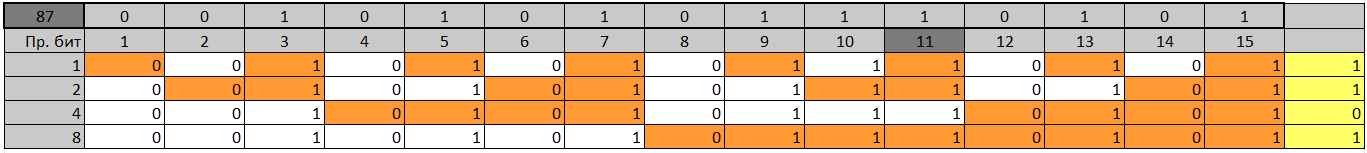


Запишем синдромы в обратном порядке: 0102 = 210. Значит ошибка в 2 бите. Исправим и запишем ответ:



# **Задание 2.**

На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения. 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.



Первая строка таблицы – это номер сообщения и само сообщение (по заданию). Следующая строка – это нумерация битов сообщения (номер неправильного бита выделен темным цветом). Первый столбец показывает проверочный бит (степень двойки).

Чтобы определить ошибочный бит мы должны выполнить операцию сложения по модулю 2 для всех проверочных битов (они выделены оранжевым цветом). Результат записан в ячейки желтого цвета.

Изображение выглядит как текст

Автоматически созданное описание



Запишем синдромы в обратном порядке: 10112 = 1110. Значит ошибка в 11 бите. Исправим и запишем ответ:



# **Задание 3.**

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.

Число информационных разрядов:

(73 + 3 + 45 + 30 + 87) \* 4 = 952

Подставим 952 в формулу 2r ≥ r + i + 1:

2r ≥ r + 953

Отсюда r = 10.

Чтобы найти коэффициент избыточности необходимо воспользоваться формулой

Коэффициент избыточности ≈ 0,01.

Ответ: Минимальное число проверочных разрядов – 10. Коэффициент избыточности ≈ 0,01.

# **Задание 4.**

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.

message = input("Введите набор из 7 цифр без разделительных знаков: ")

zeroone = ["0","1"]

*if* len(message) == 7:

*for* i *in* message:

*if* i not in zeroone:

            print("Вы ошиблись...")

            quit()

    message = int(message)

*else*:

    print("Вы ошиблись...")

    quit()

messageArr = []

messageArr.append((message//1000000)%10)

messageArr.append((message//100000)%10)

messageArr.append((message//10000)%10)

messageArr.append((message//1000)%10)

messageArr.append((message//100)%10)

messageArr.append((message//10)%10)

messageArr.append(message%10)

totalxor = ""

totalxor = totalxor + str( messageArr[3] ^ messageArr[4] ^ messageArr[5] ^ messageArr[6] )

totalxor = totalxor + str( messageArr[1] ^ messageArr[2] ^ messageArr[5] ^ messageArr[6] )

totalxor = totalxor + str( messageArr[0] ^ messageArr[2] ^ messageArr[4] ^ messageArr[6] )

*if* totalxor == "000":

    print("Код верный!")

*else*:

    totalxor = int(totalxor, 2)

    print("У вас была ошибка в " + str(totalxor) + " бите")

    messageArr[totalxor-1] = abs(1 - messageArr[totalxor-1])

    a = "".join(str(i) *for* i *in* messageArr)

    print("Ваш исправленный набор: " + a)

# **Заключение.**

В ходе проведения данной лабораторной работы я научился определять ошибки в сообщениях, представленных в бинарном виде, и при помощи кода Хэмминга исправлять данные ошибки, получая правильные сообщения. Также дополнительно я реализовал программу на языке Python, которая представляет алгоритм в действии.

# **Список литературы**

**@tltshnik Denis** Код Хэмминга. Пример работы алгоритма [В Интернете] // Хабр. - https://habr.com/ru/post/140611/.

**В. Балакшин П.** Лекция №2 [Конференция].

**Морелос-Сарагоса Р.** Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение [Книга].