Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №1**

**«Синтез помехоустойчивого кода»**

**Вариант: 84**

Выполнил:

студент 1 курса

Садовой Григорий Владимирович

Группа: Р3107

Принял:

Белозубов Александр Владимирович

Отчёт принят «\_\_»\_\_\_\_\_2022 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2022

**Оглавление**

[Задание: 3](#_Toc115552758)

[Основные этапы вычисления: 4](#_Toc115552759)

[Код на Python: 12](#_Toc115552760)

[Заключение: 14](#_Toc115552761)

# Задание

1. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
2. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
3. 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.
4. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
5. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
6. 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.
7. 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.
8. 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 командной̆ 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.

# Выполнение работы

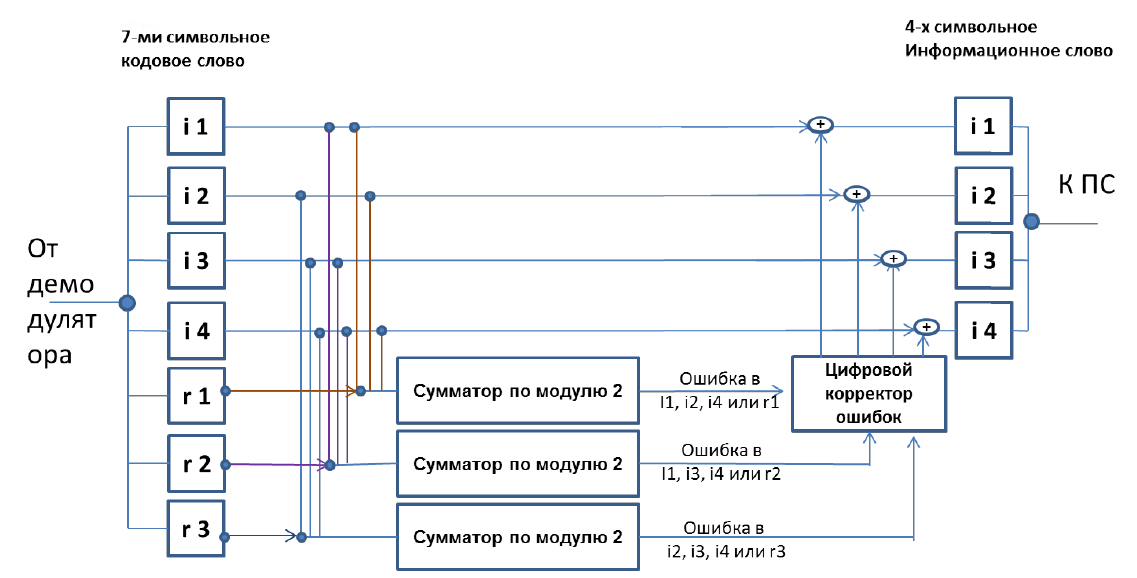


Рисунок 1: Схема декодирования классического кода Хэмминга (7;4).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2x | r1 | r2 | i1 | r3 | i2 | i3 | i4 | S |
| 1 | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x | S3 |

Таблица №1: Таблица кода Хэмминга (7,4).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Синдром S (S1, S2, S3) | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Конфигурация ошибок (позиция в сообщении) | нет | 0001000 | 0100000 | 0000010 | 1000000 | 0000100 | 0010000 | 0000001 |
| Ошибка в символе | нет | r3 | r2 | i3 | r1 | i2 | i1 | i4 |

Таблица №2: Таблица для поиска ошибочных битов.

По таблице №1 видно, какие информационные биты i контролируют проверочные биты r.

Таким образом:

* r1 = i1 ⊕ i2 ⊕ i4
* r2 = i1 ⊕ i3 ⊕ i4
* r3 = i2 ⊕ i3 ⊕ i4

Так же для декодирования сообщения применяют такое понятие, как синдром (S)– набор контрольных сумм информационных и проверочных разрядов.

Для таблицы 1 будут следующие синдромы:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4

На основе двух этих таблиц можно выяснить, какие ошибки возникли при передаче информации.

Теперь примени эти знания на практике.

**Задание 1.1**Полученное сообщение – 0100110  
Составим таблицу:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Полученное сообщение | 0 | 1 | 0 | 0 | 1 | 1 | 0 |  |
| 2x | r1 | r2 | i1 | r3 | i2 | i3 | i4 | S |
| 1 | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x | S3 |

Таблица №3

* 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

S1S2S3 = 100. Таким образом, исходя из таблицы №2, ошибка заключается в r1. Чтобы получить правильное сообщение, нужно инвертировать r1.

Получившееся правильное сообщение: 1100110  
**Ответ: 1100110**

**Задание 1.2**Полученное сообщение – 1011000  
Составим таблицу:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Полученное сообщение | 1 | 0 | 1 | 1 | 0 | 0 | 0 |  |
| 2x | r1 | r2 | i1 | r3 | i2 | i3 | i4 | S |
| 1 | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x | S3 |

Таблица №4

* 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 = 1 ⊕ 0⊕ 0⊕ 0 = 1

S1S2S3 = 011. Таким образом, исходя из таблицы №2, ошибка заключается в i3. Чтобы получить правильное сообщение, нужно инвертировать i3.

Получившееся правильное сообщение: 1011010  
**Ответ: 1011010**

**Задание 1.3**

Полученное сообщение – b  
Составим таблицу:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Полученное сообщение | 1 | 1 | 0 | 1 | 0 | 1 | 0 |  |
| 2x | r1 | r2 | i1 | r3 | i2 | i3 | i4 | S |
| 1 | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x | S3 |

Таблица №5

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0⊕ 0⊕ 0 = 1
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0⊕ 1⊕ 0 = 0
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0⊕ 1⊕ 0 = 0

S1S2S3 = 100. Таким образом, исходя из таблицы №2, ошибка заключается в r1. Чтобы получить правильное сообщение, нужно инвертировать r1.

Получившееся правильное сообщение: 0101010  
**Ответ: 0101010**

**Задание 1.4**

Полученное сообщение – 0110001  
Составим таблицу:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Полученное сообщение | 0 | 1 | 1 | 0 | 0 | 0 | 1 |  |
| 2x | r1 | r2 | i1 | r3 | i2 | i3 | i4 | S |
| 1 | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x | S3 |

Таблица №6

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 1⊕ 0⊕ 1 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1⊕ 0⊕ 1 = 1
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 0⊕ 0⊕ 1 = 1

S1S2S3 = 011. Таким образом, исходя из таблицы №2, ошибка заключается в i3. Чтобы получить правильное сообщение, нужно инвертировать i3.

Получившееся правильное сообщение: 0110011  
**Ответ: 0110011**

**Задание 2**

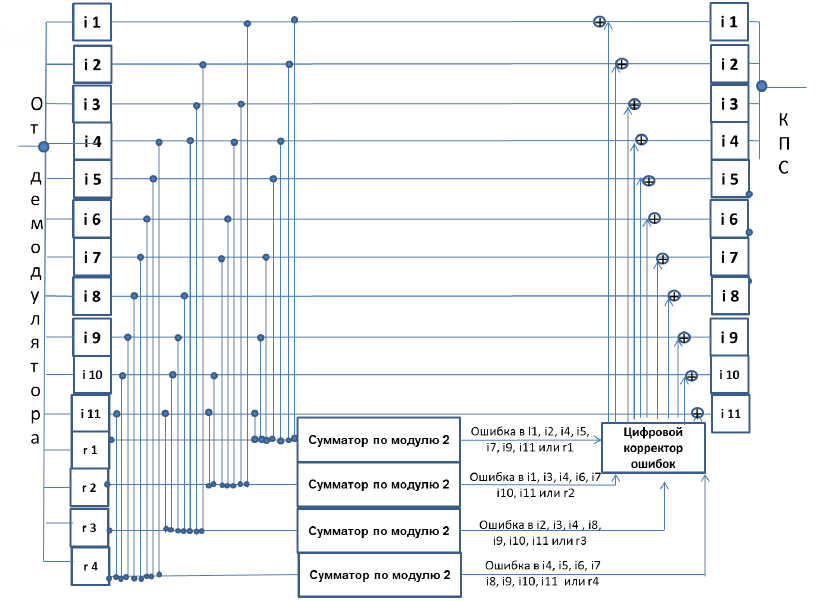
****

Рисунок 2: Схема декодирования кода Хэмминга (15;11).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| Полученное сообщение | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |  |
| 2x | r1 | r2 | i1 | r3 | i2 | i3 | i4 | r4 | i5 | i6 | i7 | i8 | i9 | i10 | i11 | S |
| 1 | x |  | x |  | x |  | x |  | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x |  |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x |  |  |  |  | x | x | x | x | S3 |
| 8 |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | S4 |

Полученное сообщение – 001110011000100  
Составим таблицу:

Таблица 7

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 ⊕ i5 ⊕ i7 ⊕ i9 ⊕ i11 = 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 1 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 ⊕ i6 ⊕ i7 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1
* S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i­8 ⊕ i9 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 = 0
* S4 = r4 ⊕ i5 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = 0 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 = 0

S1S2S3S4 = 0100. Для определения ошибки можно составить таблицу, аналогичную таблице 3, или воспользоваться следующим методом:  
Нужно «перевернуть» выражение S1S2S3S4 и перевести его из 2-ной в 10-ную систему счисления.  
00102 = 0 \* 23 + 0 \* 22 + 1 \* 21  + 0 \* 20  = 2 – таким образом, ошибка заключается в 2 разряде, т.е. в r2.   
Чтобы получить правильное сообщение, нужно инвертировать r2.

Получившееся правильное выражение – 011010100110101  
**Ответ: 011010100110101**

**Задание 3**

i = (88 + 11 + 40 + 20 + 83) \* 4 = 968 – число информационных разрядов в передаваемом сообщении.  
Чтобы определить минимальное количество проверочных разрядов (r) для данного количества информационных разрядов, воспользуемся неравенством:

2r >= r + i + 1  
2r >= r + 968 + 1  
Определим минимальное r с помощью подбора

|  |  |
| --- | --- |
| r | 2r >= r + 969 |
| 9 | 512 >= 978 (ложь) |
| 10 | 1024 >= 979 (истина) |

Таблица 8

Исходя из таблицы 8, получаем, что минимальное количество проверочных разрядов r = 10

**Ответ: 10**

# Код на Python

print('Введите семизначное число в двоичной системе Состоязее из нулей и едениц')

s = input()

n = str(s)

g = list(s)

tmp = s[2]+s[4]+s[5] ]+s[6]

print("Исходный код:")

print(tmp)

s\_1 = (int(s[0]) + int(s[2]) + int(s[4]) + int(s[6])) % 2

s\_2 = (int(s[1]) + int(s[2]) + int(s[5]) + int(s[6])) % 2

s\_3 = (int(s[3]) + int(s[4]) + int(s[5]) + int(s[6])) % 2

res = str(s\_3) + str(s\_2) + str(s\_1)

res = int(res,2)

promej = str(int(res))

match promej:

case "0":

print("Исходный код правильный")

case "1":

print("Ошибка в первом бите")

if g[res - 1] == "0":

g[res - 1] = '1'

else:

g[res - 1] = '0'

case "2":

print("Ошибка во втором бите")

if g[res - 1] == "0":

g[res - 1] = "1"

else:

g[res - 1] = "0"

case "3":

print("Ошибка во третьем бите")

if g[res - 1] == "0":

g[res - 1] = "1"

else:

g[res - 1] = "0"

case "4":

print("Ошибка во четвертом бите")

if g[res - 1] == "0":

g[res - 1] = "1"

else:

g[res - 1] = "0"

case "5":

print("Ошибка в пятом бите")

if g[res - 1] == "0":

g[res - 1] = "1"

else:

g[res - 1] = "0"

case "6":

print("Ошибка в шестом бите")

if g[res - 1] == "0":

g[res - 1] = "1"

else:

g[res - 1] = "0"

case "7":

print("Ошибка в седьмом бите")

if g[res - 1] == "0":

g[res - 1] = "1"

else:

g[res - 1] = "0"

tmp = "".join(g[2] + g[4] + g[5] + g[6])

print("Правильный код: " + tmp)

# Заключение

В ходе выполнения данной лабораторной работы я узнал о помехоустойчивых кодах, их видах и применении. На примере кода Хемминга научился с ними работать. Написал программу, анализирующую вводимое сообщение на основе кода Хемминга(7,4) и выдает правильное сообщение.