**Федеральное государственное автономное образовательное учреждение высшего образования Национальный Исследовательский**   
**Университет ИТМО**

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.01

Информатика и Вычислительная Техника

(Компьютерные системы и технологии)

Дисциплина «Информатика»

**Отчет**

**По лабораторной работе №2**

**“Синтез помехоустойчивого кода”**

**Вариант №1**

Студент

Карташев Владимир Сергеевич,   
группа P3131

Преподаватель

Белозубов Александр Владимирович

г. Санкт-Петербург, 2022 г.

# Оглавление

[**Оглавление**](#_heading=h.517o5g44k1xd) **2**

[**Задание**](#_heading=h.jumkf6fdv3zq) **3**

[**Выполнение задания**](#_heading=h.99bd3cwgq6kf) **4**

[Задание №1](#_heading=h.owp5m1l0n3pf) 4

[Задание №2](#_heading=h.oyx6k33es0rq) 4

[Задание №3](#_heading=h.tlgy155gizga) 5

[Пример 1](#_heading=h.kros65yshsga) 6

[Пример 2](#_heading=h.h1u23emzv22p) 7

[Пример 3](#_heading=h.651h8r2s84n6) 8

[Пример 4](#_heading=h.o7420tm0mrnw) 9

[Задание №4](#_heading=h.dg8ny3rb20dp) 10

[Задание №5](#_heading=h.f0ro4f7076dl) 10

[Задание №6](#_heading=h.vcg0ct1o8tqv) 11

[Задание №7](#_heading=h.vxkkw9sd5qlm) 12

[**Код программы**](#_heading=h.125v74wr1jhk) **12**

# 

# 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.

1. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
2. 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.
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.

# Выполнение задания

## Задание №1

| № | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 4 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

таблица 1

## Задание №2

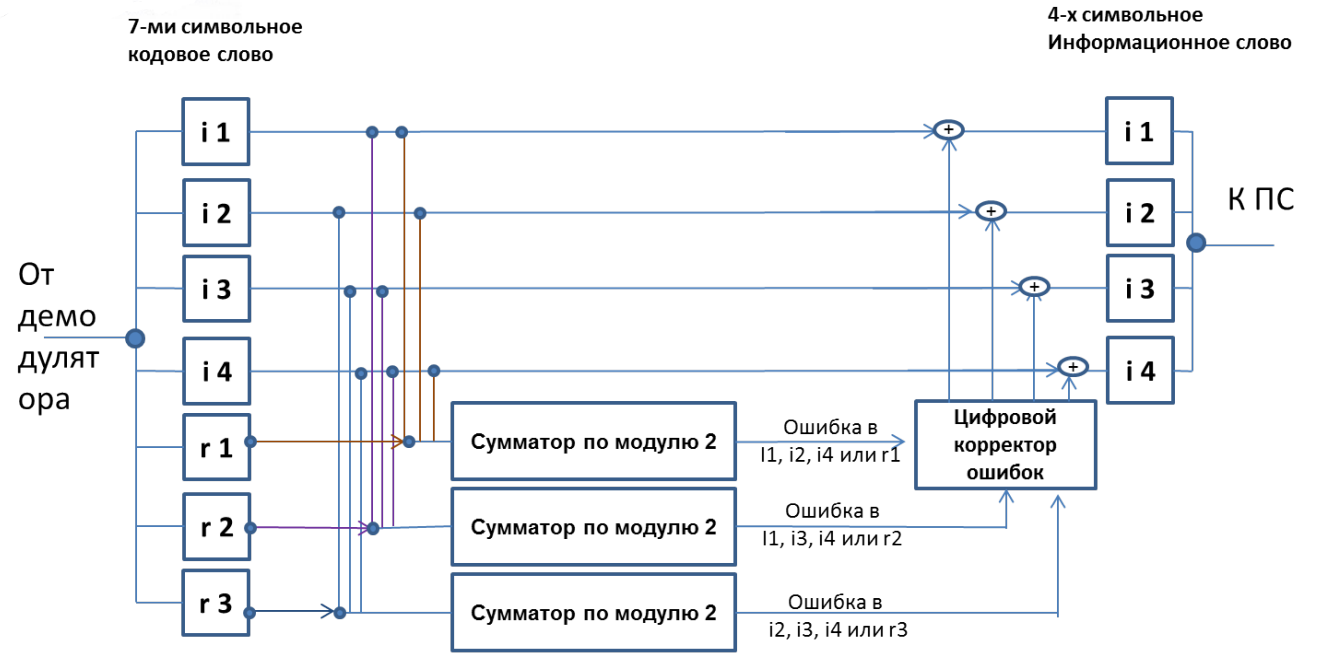
****

рисунок 1

## Задание №3

Для построения схемы декодирования классического кода Хэмминга (7;4) обратимся к таблице Хэмминга.

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2^x | r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | x |  | x |  | x |  | x |
| 2 |  | x | x |  |  | x | x |
| 4 |  |  |  | x | x | x | x |

таблица 2

| Синдром S1  (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 3

По таблице (таблица 2) видно, какие информационные биты **i** контролируют проверочные биты **r**, так как значение **r** зависит от значений **i**:

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

Далее для декодирования сообщения мы применим синдром последовательности **S**.

**Синдром последовательности** (S) — набор контрольных сумм информационных и проверочных разрядов.

Для **таблицы 2** будет следующий набор синдромов:

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4
* S3 = r4 ⊕ i2 ⊕ i3 ⊕ i4

На основе набора синдромов **таблицы 3** можно выяснить, в каком элементе заложена ошибка при передаче информации.

Теперь найдем ошибки из наборов информации **таблицы 1**.

### Пример 1

Полученное сообщение - 0001010.

Составим таблицу:

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2^x | r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | x |  | x |  | x |  | x |
| 2 |  | x | x |  |  | x | x |
| 4 |  |  |  | x | x | x | x |

таблица 4

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1
* S3 = r4 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 0 = 0

S1S2S3 = 010. Благодаря **таблице 3** мы можем сравнить набор синдромов “010” и понять, в каком элементе заложена ошибка.

В данном случае ошибка заложена в элементе **r2**. Нужно инвертировать элемент **r2**.

Следовательно, правильное сообщение будет - **0101010**.

### Пример 2

Полученное сообщение - 0000100.

Составим таблицу:

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2^x | r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | x |  | x |  | x |  | x |
| 2 |  | x | x |  |  | x | x |
| 4 |  |  |  | x | x | x | x |

таблица 5

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0
* S3 = r4 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

S1S2S3 = 101. Благодаря **таблице 3** мы можем сравнить набор синдромов “101” и понять, в каком элементе заложена ошибка.

В данном случае ошибка заложена в элементе **i2**. Нужно инвертировать элемент **i2**.

Следовательно, правильное сообщение будет - **0000000**.

### Пример 3

Полученное сообщение - 1111101.

Составим таблицу:

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 2^x | r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | x |  | x |  | x |  | x |
| 2 |  | x | x |  |  | x | x |
| 4 |  |  |  | x | x | x | x |

таблица 6

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 1 = 0
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1
* S3 = r4 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1

S1S2S3 = 011. Благодаря **таблице 3** мы можем сравнить набор синдромов “011” и понять, в каком элементе заложена ошибка.

В данном случае ошибка заложена в элементе **i3**. Нужно инвертировать элемент **i3**.

Следовательно, правильное сообщение будет - **1111111**.

### Пример 4

Полученное сообщение - 1101111.

Составим таблицу:

|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 2^x | r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | x |  | x |  | x |  | x |
| 2 |  | x | x |  |  | x | x |
| 4 |  |  |  | x | x | x | x |

таблица 7

* S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 1 = 1
* S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 1 = 1
* S3 = r4 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 1 = 0

S1S2S3 = 110. Благодаря **таблице 3** мы можем сравнить набор синдромов “110” и понять, в каком элементе заложена ошибка.

В данном случае ошибка заложена в элементе **i1**. Нужно инвертировать элемент **i1**.

Следовательно, правильное сообщение будет - **1111111**.

## Задание №4

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

таблица 8

## Задание №5

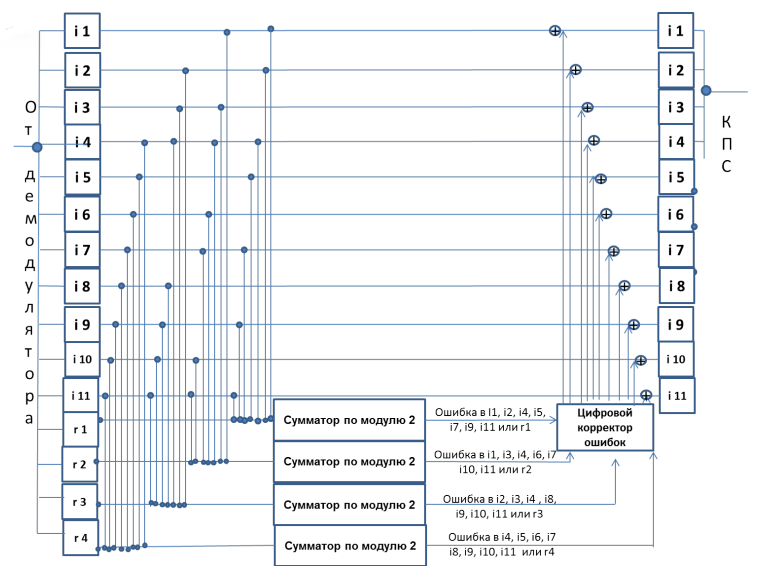


рисунок 2

## Задание №6

Полученное сообщение - 010101000110010.

Составим таблицу:

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

таблица 9

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

S1S2S3S4 = 1111

Чтобы понять, в каком разряде заключается ошибка - нужно перевернуть набор синдромов (S1S2S3S4 = S4S3S2S1). Следовательно “1111” станет “1111”. Теперь переведем число из 2-ной в 10-ную систему счисления:

11112 = 1 \* 23 + 1 \* 22 +1 \* 21 + 1 \* 20 = 8 + 4 + 2 + 0 = 14.

Следовательно, ошибка заключается в **14-ом** разряде или же в **i10**.

Нужно инвертировать элемент **i10**.

Следовательно, правильное сообщение будет - **010101000110000**.

## Задание №7

(30 + 57 + 84 + 111 + 35) \* 4 = 1268 – 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 + 1268 + 1

Определим минимальное **r** методом перебора.

| r | 2r >= r + 1268 |
| --- | --- |
| 10 | 1024 >= 1268 (ложь) |
| 11 | 2048 >= 1268 (истина) |

таблица 10

При помощи **таблицы 10** определим, что минимальное количество проверочных разрядов **r = 11**.

# 

# Код программы

#include <iostream>

using namespace std;

int main()

{

string code;

cout << "Введите ваше сообщение по типу: r1 r2 i1 r3 i2 i3 i4" << endl;

cin >> code;

int r1, r2, i1, r3, i2, i3, i4; // элементы сообщения

r1 = code[0] - '0'; r2 = code[1] - '0'; r3 = code[3] - '0';

i1 = code[2] - '0'; i2 = code[4] - '0'; i3 = code[5] - '0'; i4 = code[6] - '0';

int s1, s2, s3; // синдром

string syndrome[] = {"000", "001", "010", "011", "100", "101", "110", "111"};

string error[] = {"нет ошибки", "r3", "r2", "i3", "r1", "i2", "i1", "i4"};

s1 = r1 + i1 + i2 + i4;

s2 = r2 + i1 + i3 + i4;

s3 = r3 + i2 + i3 + i4;

if (s1 == 0 or s1 == 2 or s1 == 4) s1 = 0;

if (s2 == 0 or s2 == 2 or s2 == 4) s2 = 0;

if (s3 == 0 or s3 == 2 or s3 == 4) s3 = 0;

if (s1 == 1 or s1 == 3) s1 = 1;

if (s2 == 1 or s2 == 3) s2 = 1;

if (s3 == 1 or s3 == 3) s3 = 1;

string sum = to\_string(s1) + to\_string(s2) + to\_string(s3);

for (int i = 0; i < 8; i++){

if (sum == syndrome[i]) {

cout << "Ошибка в элементе: " << error[i];

return 0;

}

}

}

# Заключение

В ходе лабораторной работы я повторил ранее изученное представление информации в коде Хэммингтона (7/4 и 15/11), а также научился находить ошибки в переданных сообщениях, чтобы таким образом исправить поврежденные биты.

# Список литературы

1. Hamming, Richard W. *The Art of Doing SCIENCE and Engineering*. CRC Press, 1997.
2. Алексеев, Е. Г., and С. Д. Богатырев. *Системы Счисления*.