**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО”**

Факультет программной инженерии и компьютерной техники (ПИКТ)

Направление подготовки (специальность) – 09.03.04 (Нейротехнологии и программная инженерия)

Информатика

Синтез помехоустойчивого кода

Лабораторная работа № 2

Выполнил студент

Слудная Виктория Евгеньевна

Группа №P3122

Преподаватель: Болдырева Елена Александровна

г. Санкт-Петербург

2024 г.

Оглавление

[**Задание:** 3](#_Toc177928336)

[**Отчет:** 3](#_Toc177928337)

[***Задание 1*** 3](#_Toc177928338)

[***Задание 2*** 3](#_Toc177928339)

[***Задание 3*** 4](#_Toc177928340)

[***Задание 4*** 4](#_Toc177928341)

[***Задание 5*** 5](#_Toc177928342)

[***Задание 6*** 5](#_Toc177928343)

[***Задание 7*** 6](#_Toc177928344)

[***Задание 8*** 6](#_Toc177928345)

[***Задание 9*** 7](#_Toc177928346)

[***Задание 10*** 7](#_Toc177928347)

[***Задание 12*** 9](#_Toc177928348)

[***Задание 13*** 9](#_Toc177928349)

[**Вывод:** 10](#_Toc177928350)

[**Список литературы:** 10](#_Toc177928351)

**Вариант: 476011 = 61**

# **Задание:**



*Рис. 1. Выбранный вариант.*

1. На основании номера варианта выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода. Построить схему декодирования классического кода Хэмминга (7;4), выявить ошибки в принятом сообщении и исправить.
2. На основании номера варианта выбрать 1 полученное сообщение в виде последовательности 11-символьного кода. Построить схему декодирования классического кода Хэмминга (15;11), выявить ошибки в принятом сообщении и исправить.
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.
4. 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.

# 

# **Отчет:**

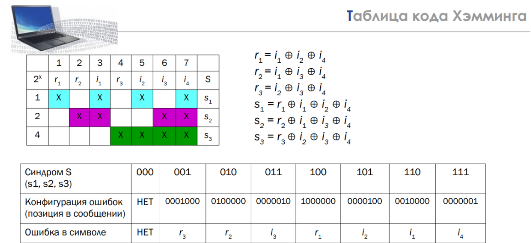
## ***Задание 1. Основные этапы вычисления.***

Для выполнения задания 1 воспользуемся формулами для r = 3.

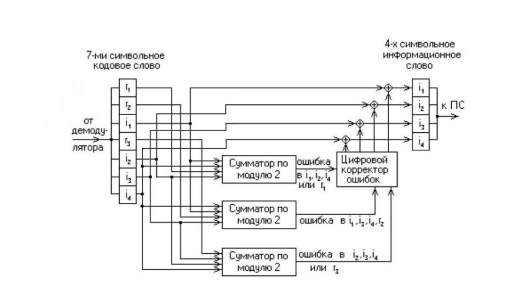
s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4;

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4;

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4;



*Рис. 2. Таблица кода Хэмминга*

**

*Рис. 3. Схема декодирования кода Хэминга.*

## ***Задание 1.1***

*Таблица 1.*

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

Полученное сообщение: 0000011.

S1 = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1

S2 = 0 ⊕ 0 ⊕ 1 ⊕ 1 = 0

S3 = 0 ⊕ 0 ⊕ 1 ⊕ 1 = 0

Полученный синдром S = 100, ошибка находится в r1 (3 бит). Правильное сообщение: 1000011

## ***Задание 1.2***

*Таблица 2.*

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

Полученное сообщение: 1011101.

S1 = 1 ⊕ 1 ⊕ 1 ⊕ 1 = 0

S2 = 0 ⊕ 1 ⊕ 0 ⊕ 1 = 0

S3 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1

Полученный синдром S = 001, ошибка находится в r3 (4 бит). Правильное сообщение: 1010101.

## ***Задание 1.3***

*Таблица 3.*

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

Полученное сообщение: 0101000.

S1 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0

S2 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

S3 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

Полученный синдром S = 011, ошибка находится в i3 (6 бит). Правильное сообщение: 0101010.

## ***Задание 1.4***

*Таблица 4.*

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

Полученное сообщение: 1111010.

S1 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

S2 = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1

S3 = 1 ⊕ 0 ⊕ 1 ⊕ 0 = 0

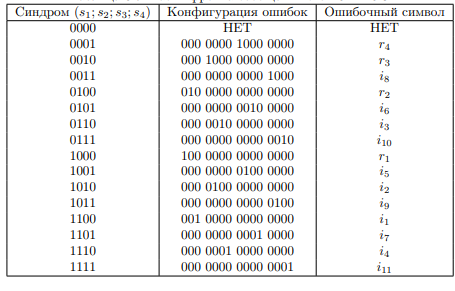
Полученный синдром S = 010, ошибка находится в r2 (2 бит).

Правильное сообщение: 1011010.

## ***Задание 2***



*Рис 4. Таблица кода Хэмминга(15;11).*

**

*Рис. 5. Расшифровка собщений.*

Полученное сообщение: 010001111010011.

r1 = i1 ⊕ i2 ⊕ i4 ⊕ i5 ⊕ i7 ⊕ i9 ⊕ i11;

r2 = i1 ⊕ i3 ⊕ i4 ⊕ i6 ⊕ i7 ⊕ i10 ⊕ i11;

r3 = i2 ⊕ i3 ⊕ i4 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11;

r4 = i5 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11;

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 ⊕ i5 ⊕ i7 ⊕ i9 ⊕ i11;

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 ⊕ i6 ⊕ i7 ⊕ i10 ⊕ i11;

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11;

s4 = r4 ⊕ i5 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11;

S1 = 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 = 0

S2 = 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 = 1

S3 = 0 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 = 0

S4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 = 1

0101 - i6, 10 бит

Полученный синдром S = 0101, ошибка находится в i6 (10 бит).

Правильное сообщение: 010001111110011.

*Таблица 5.*

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

## ***Задание 3***

i = (43 + 80 + 5 + 42 + 61) \* 4 = 924

Определение минимального количества контрольных разрядов: 2 k ≥ r + i + 1

r + i +1 = + 925

r = 10

Помехоустойчивый код характеризуется:

• i – числом информационных разрядов;

• r – числом проверочных разрядов;

• n – общим числом разрядов (n = i + r).

Коэффициент избыточности: КИ = r / n

КИ = 10 / (10+924) = 0,0107066381

Ответ: 10; 0,0107066381.

## ***Задание 7***

*Исходный код:*

#include <iostream>

#include <string>

#include <map>

#include <vector>

std::vector<std::string> DICHARGE = { "r1", "r2", "i1", "r3", "i2", "i3", "i4" };

std::map<std::string, int> PartsingMessage(const std::string& code) {

std::map <std::string, int> temp;

for (int i = 0; i < code.size(); ⊕⊕i) {

temp[DICHARGE[i]] = int(code[i] - '0');

}

return temp;

}

int CalculatingTheErrorCode(std::map<std::string, int> mapCode) {

int s1 = (mapCode["r1"] ⊕ mapCode["i1"] ⊕ mapCode["i2"] ⊕ mapCode["i4"])%2;

int s2 = (mapCode["r2"] ⊕ mapCode["i1"] ⊕ mapCode["i3"] ⊕ mapCode["i4"])%2;

int s3 = (mapCode["r3"] ⊕ mapCode["i2"] ⊕ mapCode["i3"] ⊕ mapCode["i4"])%2;

std::string s = std::to\_string(s1) ⊕ std::to\_string(s2) ⊕ std::to\_string(s3);

return stoi(s);

}

std::string SearchErrors(const std::string& code, std::map<std::string, int> mapCode) {

auto temp\_s = CalculatingTheErrorCode(mapCode);

switch (temp\_s) {

case 0:

return "none";

break;

case 1:

return "r3";

break;

case 10:

return "r2";

break;

case 11:

return "i3";

break;

case 100:

return "r1";

break;

case 101:

return "i2";

break;

case 110:

return "i1";

break;

case 111:

return "i4";

break;

}

return "error";

}

void MakeRightCode(const std::string& code) {

std::cout << "Correct code: ";

auto mapCode = PartsingMessage(code);

auto nameError = SearchErrors(code, mapCode);

if (nameError == "none" or nameError == "error") {

std::cout << code;

}

else {

if (mapCode.count(nameError)) {

if (mapCode[nameError] == 0) {

mapCode[nameError] = 1;

}

else {

mapCode[nameError] = 0;

}

}

for (const auto& name\_ : DICHARGE) {

std::cout << mapCode[name\_];

}

std::cout << std::endl << "error in " ⊕ nameError;

}

}

int main()

{

std::string s;

std::cin >> s;

if(s.size() == 7) {

MakeRightCode(s);

}

}



*Рис. 6. Пример ввода и вывода программы.*

# **Вывод:**

В ходе данной лабораторной работы я изучила алгоритм декодирования кода Хэмминга, научилась анализировать помехоустойчивый код, находить в нем ошибки и исправлять их.

# **Список литературы:**

1. Балакшин П.В., Соснин В.В., Машина Е.А. Информатика.– СПб: Университет ИТМО, 2020.– 64 с.
2. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 651 с.: ил.
3. Алексеев Е.Г., Богатырев С.Д. Информатика. Мультимедийный электронный учебник.