

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ СИСТЕМ УПРАВЛЕНИЯ И РОБОТОТЕХНИКИ

## ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине  
«Информатика»

Вариант № 73

**Выполнил:**

Студент группы Р3116

Злобин Николай Андреевич

**Преподаватель:**

Балакшин Павел Валерьевич

Санкт-Петербург, 2021

## Задание:

На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.

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.

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.

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.

**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.

## Решение:

1)

55:

r1	r2	i1	r3	i2	i3	i4
1	1	1	0	0	1	1
X		X		X		X
	X	X		X	X	
			X	X	X	X

$$s1 = \text{xor}(r1, i1, i2, i4) = 1$$

$$s2 = \text{xor}(r2, i1, i2, i3) = 1$$

$$s3 = \text{xor}(r3, i2, i3, i4) = 0$$

011(2) = 3(10) => ошибка в 3 бите. Правильно: **1100011**

92:

r1	r2	i1	r3	i2	i3	i4
1	0	0	0	1	1	0
X		X		X		X
	X	X		X	X	
			X	X	X	X

$$s1 = \text{xor}(r1, i1, i2, i4) = 0$$

$$s2 = \text{xor}(r2, i1, i2, i3) = 0$$

$$s3 = \text{xor}(r3, i2, i3, i4) = 0$$

ошибок нет (Матрица синдромов нулевая)

17:

r1	r2	i1	r3	i2	i3	i4
0	0	1	0	0	0	1
X		X		X		X
	X	X		X	X	
			X	X	X	X

$$s1 = \text{xor}(r1, i1, i2, i4) = 0$$

$$s2 = \text{xor}(r2, i1, i2, i3) = 1$$

$$s3 = \text{xor}(r3, i2, i3, i4) = 1$$

110(2) = 6(10) => ошибка в 6 бите. Правильно: **0010011**

74:

r1	r2	i1	r3	i2	i3	i4
0	0	1	1	1	0	1
X		X		X		X
	X	X		X	X	
			X	X	X	X

$$s1 = \text{xor}(r1, i1, i2, i4) = 1$$

$$s2 = \text{xor}(r2, i1, i2, i3) = 0$$

$$s3 = \text{xor}(r3, i2, i3, i4) = 1$$

101(2) = 5(10) => ошибка в 5 бите. Правильно: **0011001**

2)

72:

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

0	0	1	1	1	0	0	1	0	0	0	0	1	0	0
X		X		X		X		X		X		X		X
	X	X		X	X			X	X			X	X	
			X	X	X	X					X	X	X	X
							X	X	X	X	X	X	X	X

$s1 = \text{xor}(r1, i1, i2, i4, i5, i7, i9, i11) = 1$

$s2 = \text{xor}(r2, i1, i2, i3, i5, i6, i9, i10) = 1$

$s3 = \text{xor}(r3, i2, i3, i4, i8, i9, i10, i11) = 1$

$s4 = \text{xor}(r4, i5, i6, i7, i8, i9, i10, i11) = 0$

$0111(2) = 7(10) \Rightarrow$  ошибка в 7 бите. Правильно: **001110110000100**

### 3)

$m = 4 * (55 + 92 + 17 + 74 + 72) = 1240$

$r \geq \log_2(1240 + r + 1) \Rightarrow r = 11$

коэф. избыточности =  $11 / (11 + 1240) = 0,00879297$

### 4)

```
#include <iostream>
```

```
#include <bitset>
```

```
#include <string>
```

```
#define __check(what, n) (what[n] == '1' || what[n] == '0')
```

```
#define _check(what)( __check(what, 0) && __check(what, 1) && __check(what, 2) && \
```

```
__check(what, 3) && __check(what, 4) && __check(what, 5) && \
```

```
__check(what, 6))
```

```
typedef std::bitset<7> Word;
```

```
std::string reverse(std::string const& s) noexcept;
```

```
int16_t check_error(Word&) noexcept;
```

```
signed main() {
```

```
    std::string buff;
```

```
    std::cin >> buff;
```

```
    if(buff.length() != 7 || !_check(buff)) {
```

```
        std::cout << "incorrect input\n";
```

```
        return 0;
```

```

    }

    Word word(reverse(buff));

    int16_t error = check_error(word);

    if(error) {
        std::cout << "Error in " << error << " bit\n";
    } else {
        std::cout << "No error\n";
    }

    std::cout << "Correct is " << reverse(word.to_string()) << "\n";

    return 0;
}

std::string reverse(std::string const& s) noexcept {
    return std::string(s.rbegin(), s.rend());
}

int16_t check_error(Word& word) noexcept {
    std::string syndrome = "";
    syndrome.append(word[3] ^ word[4] ^ word[5] ^ word[6] ? "1" : "0");
    syndrome.append(word[1] ^ word[2] ^ word[4] ^ word[5] ? "1" : "0");
    syndrome.append(word[0] ^ word[2] ^ word[4] ^ word[6] ? "1" : "0");
    if(syndrome == "000") {
        return 0;
    } else {
        int16_t error_bit = std::stoi(syndrome, nullptr, 2);
        word.flip(error_bit - static_cast<int64_t>(1));
        return error_bit;
    }
}

```

## Вывод:

Во время выполнения работы я разобрался с кодом Хэмминга. Методом его кодирования и декодирования. А также разработал программу для проверки классического кода Хэмминга (7,4).