



Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки: 09.03.04 – Системное и прикладное программное обеспечение

Дисциплина «Информатика»

Отчёт по лабораторной работе №2

Синтез помехоустойчивого кода

Вариант №81

Выполнил: студентка группы Р3115

Галак Екатерина Анатольевна

Проверил:

Белокон Юлия Алексеевна

Санкт – Петербург, 2024

Оглавление

Задание.....	3
Порядок выполнения работы	3
Задачи	4
Основные этапы вычисления:	5
Задача 1	5
Задача 1.1	5
Задача 1.2	6
Задача 1.3	6
Задача 1.4	7
Задача 2	7
Задача 3	8
Дополнительное задание	9
Заключение.....	11
Список используемых источников	11

Задание

Порядок выполнения работы

1. Определить свой вариант задания с помощью номера в ISU (он же номер студенческого билета). Вариантом является комбинация 3-й и 5-й цифр. Мой номер ISU: 408417 \Rightarrow мой номер варианта для второй лабораторной работы 81
2. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
3. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
4. 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.
5. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
6. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
7. 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.
8. 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.
9. Дополнительное задание №1 (позволяет набрать от 86 до 100 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на любом языке программирования, которая на вход 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.

Задачи

1. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода (Таблица 1); построить схему декодирования классического кода Хэмминга (7; 4), которую представит в отчете в виде изображения; показать, исходя из выбранных вариантов сообщений, имеются ли в принятом сообщении ошибки, и, если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

Номер задания	Полученное сообщение
1 (65)	1010100
2 (10)	1010000
3 (37)	1001010
4 (77)	0111101

(Таблица 1)

2. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода (Таблица 2); построить схему декодирования классического кода Хэмминга (15; 11), которую представить в отчете в виде изображения; показать, исходя из выбранного варианта сообщений, имеются ли в принятом сообщении ошибки, и, если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

Номер задания	Полученное сообщение
80	001010100000101

(Таблица 2)

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.

Основные этапы вычисления:

Задача 1

Задача 1.1

r_1	r_2	i_1	r_3	i_2	i_3	i_4
1	0	1	0	1	0	0

	1	2	3	4	5	6	7	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	S
1	X		X		X		X	s_1
2		X	X			X	X	s_2
4				X	X	X	X	s_3

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$s = (s_1, s_2, s_3) = 110 \Rightarrow$ ошибка в символе i_4 (исходя из таблицы на Рисунке 1) \Rightarrow верное сообщение: 101010**1**

	1	2	3	4	5	6	7	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	S
1	X		X		X		X	s_1
2		X	X			X	X	s_2
4				X	X	X	X	s_3

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4$$

Синдром S (s_1, s_2, s_3)	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	r_3	r_2	i_3	r_1	i_2	i_1	i_4

(Рисунок 1 – код Хэмминга (7;4))

Ответ: 1010101

Задача 1.2

r ₁	r ₂	i ₁	r ₃	i ₂	i ₃	i ₄
1	0	1	0	0	0	0

	1	2	3	4	5	6	7	
2 ^x	r ₁	r ₂	i ₁	r ₃	i ₂	i ₃	i ₄	S
1	X		X		X		X	s ₁
2		X	X			X	X	s ₂
4				X	X	X	X	s ₃

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 0 \oplus 0 = 0$$

$s = (s_1, s_2, s_3) = 010 \Rightarrow$ ошибка в символе r_2 (исхода из таблицы на Рисунке 1) \Rightarrow
верное сообщение: 1**1**10000

Ответ: 1110000

Задача 1.3

r ₁	r ₂	i ₁	r ₃	i ₂	i ₃	i ₄
1	0	0	1	0	1	0

	1	2	3	4	5	6	7	
2 ^x	r ₁	r ₂	i ₁	r ₃	i ₂	i ₃	i ₄	S
1	X		X		X		X	s ₁
2		X	X			X	X	s ₂
4				X	X	X	X	s ₃

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$s = (s_1, s_2, s_3) = 110 \Rightarrow$ ошибка в символе i_1 (исходя из таблицы на Рисунке 1) \Rightarrow
верное сообщение: 10**1**1010

Ответ: 1011010

Задача 1.4

r_1	r_2	i_1	r_3	i_2	i_3	i_4
0	1	1	1	1	0	1

	1	2	3	4	5	6	7	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	S
1	X		X		X		X	s_1
2		X	X			X	X	s_2
4				X	X	X	X	s_3

$$r_1 = i_1 \oplus i_2 \oplus i_4$$

$$r_2 = i_1 \oplus i_3 \oplus i_4$$

$$r_3 = i_2 \oplus i_3 \oplus i_4$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$s = (s_1, s_2, s_3) = 111 \Rightarrow$ ошибка в символе i_4 (исходя из таблицы на Рисунке 1) \Rightarrow
верное сообщение: 01111**00**

Ответ: 0111100

Задача 2

r_1	r_2	i_1	r_3	i_2	i_3	i_4	r_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}
0	0	1	0	1	0	1	0	0	0	0	0	1	0	1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
2^x	r_1	r_2	i_1	r_3	i_2	i_3	i_4	r_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	S
1	X		X		X		X		X		X		X		X	s_1
2		X	X			X	X			X	X			X	X	s_2
4				X	X	X	X					X	X	X	X	s_3
8								X	X	X	X	X	X	X	X	s_4

$$r_1 = i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11}$$

$$r_2 = i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11}$$

$$r_3 = i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11}$$

$$r_4 = i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11}$$

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$s_4 = r_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$s = (s_1, s_2, s_3, s_4) = 1100 \Rightarrow \text{ошибка в символе } i_1 \Rightarrow \text{верное сообщение:}$$

000010100000101

Ответ: 000010100000101

Задача 3

Информационных разрядов в передаваемом сообщении: $i = (65 + 10 + 37 + 77 + 80)$

$$* 4 = 1076$$

Пусть r – количество проверочных разрядов, тогда всего бит в сообщении: $2^r - 1$;
информационных бит: $2^r - r - 1$. Требуется найти такое r , что

$$2^r \geq r + i + 1$$

$$2^r \geq r + 1076 + 1$$

$r = 11$, т. е. минимальное количество проверочных разрядов $r = 11$.

Коэффициент избыточности:

$$k = \frac{r}{i+r} = \frac{11}{1076+11} \approx 0,0101196$$

Ответ: 11; 0,0101196

Дополнительное задание

Код написан на языке программирования C++. Работает для любой длины сообщения (не только для длины 7). Также программа поддерживает аргумент командной строки (опцию) для ввода вручную количества контрольных разрядов (например, “-p=4”, т.е. код отработает для 4 контрольных разрядов, если этого количества достаточно для покрытия введенных информационных разрядов, в случае, если введенного количества контрольных разрядов недостаточно, пользователь получает сообщение об этом, генерируется исключение). Использованы поразрядные операции:

1) $S \& (S - 1)$ – проверка числа на принадлежность степеням двойки (выражение равно нулю, если S – степень двойки)

2) $(1 \ll r)$ – возведение 2 в степень r .

```
#include <iostream>
#include <string>
#include <cstring>
#include <cmath>

int countSyndrom(std::string& message) {
    int numberControlDig = 0;
    while ((1 << numberControlDig) < message.size() + 1) {
        ++numberControlDig;
    }

    int syndrom = 0;

    for (int r = 0; r < numberControlDig; ++r) {
        int parity = 0;
        for (int i = (1 << r) - 1; i < message.size(); i += (1 << (r + 1))) {
            for (int j = i; j < i + (1 << r) && j < message.size(); ++j)
            {
                parity ^= (message[j] - '0');
            }

            syndrom += parity * (1 << r);
        }

        return syndrom;
    }
}

int parsingArgv(int argc, char** argv) {
    std::string ansString = "";
    for (std::size_t i = 1; i < argc; ++i) {
        for (std::size_t j = 0; j < strlen(argv[i]); ++j) {
            if (argv[i][j] == '-' || argv[i][j] == 'p' || argv[i][j] == '=') {
                continue;
            }
            ansString += argv[i][j];
        }
    }
}
```

```

        return stoi(ansString);
    }

main(int argc, char** argv) {
    std::string message;
    std::cout << "Enter a message: " << std::endl;
    std::cin >> message;

    int messegeSize = message.size();

    if (argc > 1) {
        int parity = parsingArgv(argc, argv);

        if (message.size() > (1 << parity) - 1) {
            throw std::invalid_argument("Too few parity bits: " +
parity);
        }

        while (message.size() < (1 << parity) - 1) {
            message += '0';
        }

        int S = countSyndrom(message);

        if (!S) {
            std::cout << "There are no errors in the entered message" <<
std::endl;
            std::cout << "Information bits of the message: " << std::endl;
        } else {
            std::cout << "An error was found in bit number: " << S;
            message[S - 1] = (message[S - 1] == '0') ? '1' : '0';

            if (!(S & (S - 1))) {
                std::cout << " (r" << (int)log2(S) + 1 << ")" << std::endl;
            } else {
                std::cout << " (i" << S - (int)log2(S) - 1 << ")" <<
std::endl;
            }
        }

        for (int i = 1; i <= message.size() && i <= messegeSize; ++i) {
            if (!(i & (i - 1))) { // проверка на степень двойки
                continue;
            }

            std::cout << message[i - 1];
        }

        std::cout << std::endl;
    }
}

```

Результаты работы программы (Рисунки 1 – 5):

```
PS D:\ITMO\sppo_2024\inf\lab2> .\a.exe -p=12
Enter a message:
1010100
An error was found in bit number: 7 (i4)
1101
PS D:\ITMO\sppo_2024\inf\lab2> .\a.exe
Enter a message:
1010100
An error was found in bit number: 7 (i4)
1101
```

(Рисунок 2 – Результат работы программы)

```
PS D:\ITMO\sppo_2024\inf\lab2> .\a.exe
Enter a message:
1011101010101
An error was found in bit number: 11 (i7)
110110001
PS D:\ITMO\sppo_2024\inf\lab2> .\a.exe
Enter a message:
111010100000111111
An error was found in bit number: 14 (i10)
11010000101111
```

(Рисунок 3 – Результат работы программы)

```
PS D:\ITMO\sppo_2024\inf\lab2> .\a.exe -p=4
Enter a message:
101010100101111111101
terminate called after throwing an instance of 'std::invalid_argument'
what(): few parity bits:
```

(Рисунок 4 –

Результат работы программы)

```
PS D:\ITMO\sppo_2024\inf\lab2> .\a.exe
Enter a message:
000010100000101
There are no errors in the entered message
Information bits of the message:
01010000101
```

(Рисунок 5 – Результат работы программы)

Заключение

Во время выполнения данной лабораторной работы я научилась работать с кодом Хэмминга, анализировать помехоустойчивые коды, находить в них ошибки и исправлять их.

Список используемых источников

1. Кудряшов Б. Д. Основы теории кодирования / Б. Д. Кудряшов — Санкт-Петербург: БХВ-Петербург, 2016 — 400 с.
2. Ишмухаметов Ш. Т., Латыпов Р. Х., Рубцова Р. Г., Столов Е. Л. Введение в теорию кодирования и криптографию: Учебное пособие / Ш. Т.

Ишмухаметов, Р. Х. Латыпов, Р. Г. Рубцова, Е. Л. Столов — Казань: Казанский университет, 2021 — 211 с.