

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 P3115

Галак Екатерина Анатольевна

Проверил:

Белокон Юлия Алексеевна

Санкт – Петербург, 2024

Оглавление

[Задание 3](#_Toc178969645)

[Порядок выполнения работы 3](#_Toc178969646)

[Задачи 4](#_Toc178969647)

[Основные этапы вычисления: 5](#_Toc178969648)

[Задача 1 5](#_Toc178969649)

[Задача 1.1 5](#_Toc178969650)

[Задача 1.2 6](#_Toc178969651)

[Задача 1.3 6](#_Toc178969652)

[Задача 1.4 7](#_Toc178969653)

[Задача 2 7](#_Toc178969654)

[Задача 3 8](#_Toc178969655)

[Дополнительное задание 9](#_Toc178969656)

[Заключение 11](#_Toc178969657)

[Список используемых источников 11](#_Toc178969658)

# 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 ⇒ 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 отчёте 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**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

r1 = i1 ⊕ i2 ⊕ i4

r2 = i1 ⊕ i3 ⊕ i4

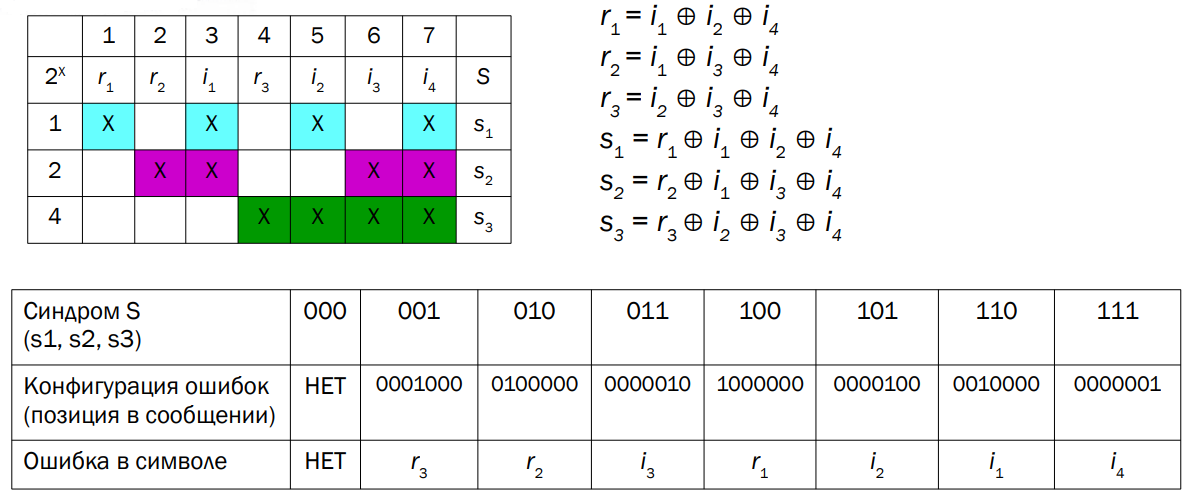
r3 = i2 ⊕ i3 ⊕ i4

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

s = (s1, s2, s3) = 110 ⇒ ошибка в символе i4 (исходя из таблицы на Рисунке 1) ⇒ верное сообщение: 1010101



(Рисунок 1 – код Хэмминга (7;4))

Ответ: 1010101

### **Задача 1.2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

r1 = i1 ⊕ i2 ⊕ i4

r2 = i1 ⊕ i3 ⊕ i4

r3 = i2 ⊕ i3 ⊕ i4

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 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0

s = (s1, s2, s3) = 010 ⇒ ошибка в символе r2 (исхода из таблицы на Рисунке 1) ⇒ верное сообщение: 1110000

Ответ: 1110000

### **Задача 1.3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

r1 = i1 ⊕ i2 ⊕ i4

r2 = i1 ⊕ i3 ⊕ i4

r3 = i2 ⊕ i3 ⊕ i4

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 0 = 0

s = (s1, s2, s3) = 110 ⇒ ошибка в символе i1 (исходя из таблицы на Рисунке 1) ⇒ верное сообщение: 1011010

Ответ: 1011010

### **Задача 1.4**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

r1 = i1 ⊕ i2 ⊕ i4

r2 = i1 ⊕ i3 ⊕ i4

r3 = i2 ⊕ i3 ⊕ i4

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 1 ⊕ 1 ⊕ 1 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1

s = (s1, s2, s3) = 111 ⇒ ошибка в символе i4 (исходя из таблицы на Рисунке 1) ⇒ верное сообщение: 0111100

Ответ: 0111100

## Задача 2

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

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 = 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 ⊕ i6 ⊕ i7 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 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 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 = 0

s = (s1, s2, s3, s4) = 1100 ⇒ ошибка в символе i1 ⇒ 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: 2r – 1; информационных бит: 2r – r –1. Требуется найти такое r, что

2r ≥ r + i + 1

2r ≥ r + 1076 + 1

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

Коэффициент избыточности:

k = = ≈ 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 << 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**;**

**}**

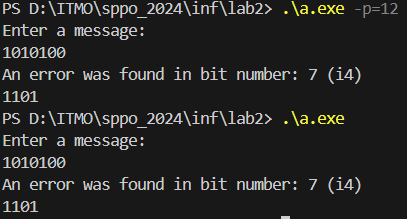
std**::**cout **<<** message**[**i **-** **1];**

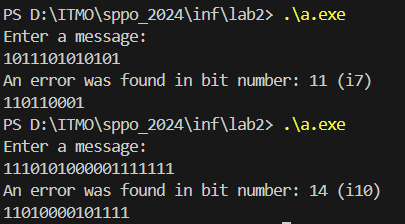
**}**

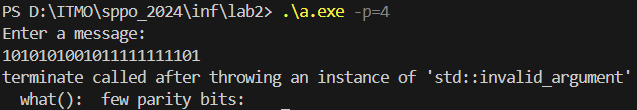
std**::**cout **<<** std**::**endl**;**

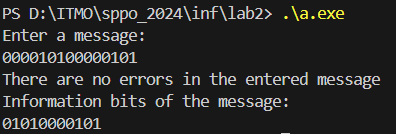
**}**

Результаты работы программы (Рисунки 1 – 5):

 (Рисунок 2 – Результат работы программы)

 (Рисунок 3 – Результат работы программы)

 (Рисунок 4 – Результат работы программы)

 (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 c.
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.