Санкт-Петербургский государственный университет ИТМО

Лабораторная работа №2.

"Синтез помехоустойчивого кода"

Вариант 11.

Выполнила: студентка группы P3112

Берелехис Светлана Михайловна

Преподаватель: Рудникова Тамара Владимировна

Санкт-Петербург, 2021

Оглавление

[Задание. 2](#_Toc84321820)

[Основная часть. 3](#_Toc84321821)

[Программа. 6](#_Toc84321822)

[Вывод. 7](#_Toc84321823)

# Задание.

1. Определить свой вариант задания с помощью номера в ISU (он же номер студенческого билета). Вариантом является комбинация 3-й и 5-й цифр. Т.е. если номер в ISU = 123456, то вариант = 35.

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. 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.

# Основная часть.

Последовательности по варианту (числа):

54 46 68 90

Схема декодирования классического кода Хемминга.

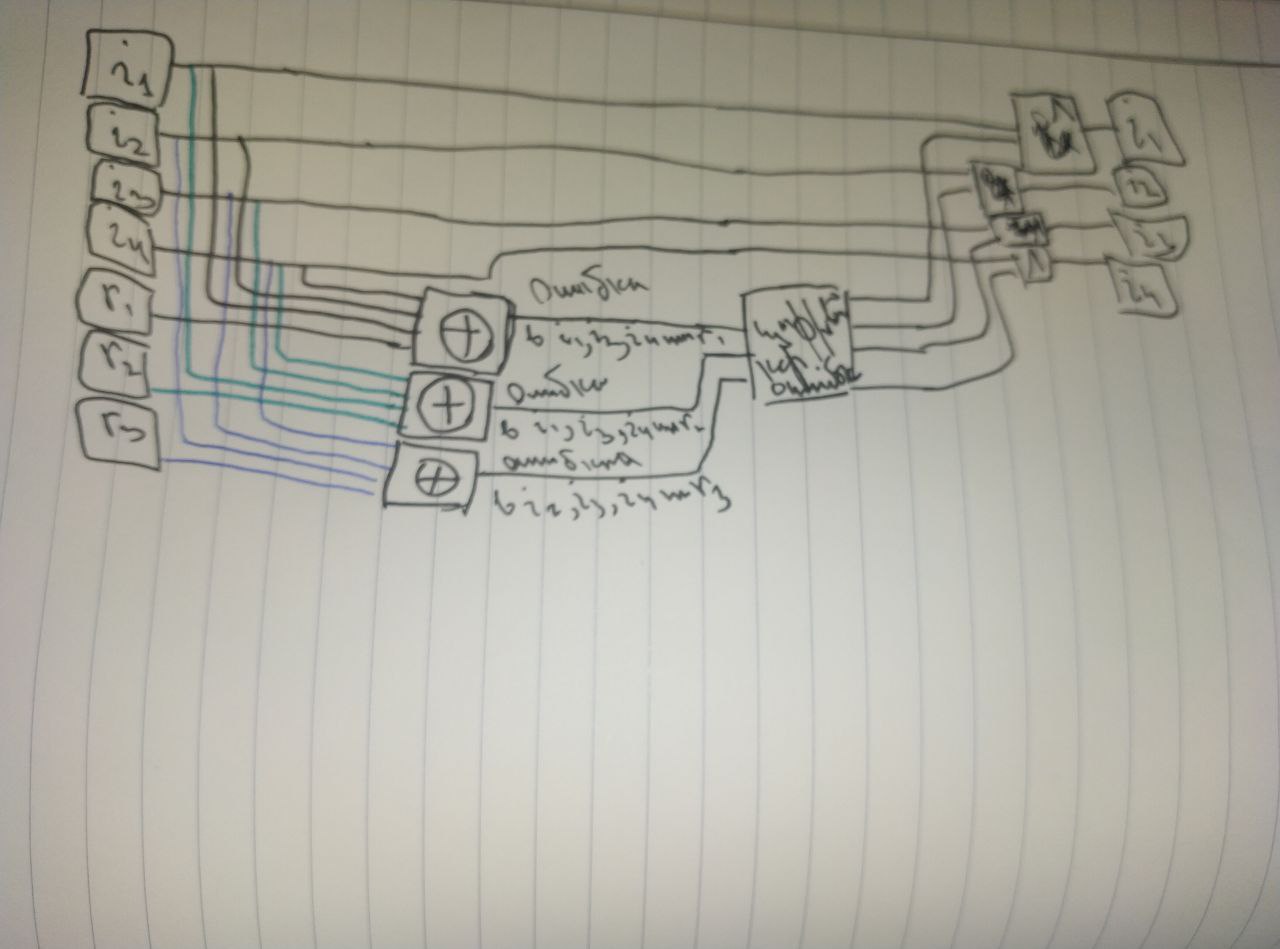


Рисунок . Схема декодирования классического кода Хемминга(7, 4)

Перевод в последовательности:

1 1 0 1 0 1 1

0 0 1 1 0 1 1

1 1 0 1 1 0 0

0 1 1 0 1 1 0

Для 1 1 0 1 0 1 1 таблица:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 |  |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 |  |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

Код (0, 1, 1) – ошибка в символе 6 т.е. в i3

Сообщение 0001

Для 0 0 1 1 0 1 1 таблица:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 |  |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

Код (0, 1, 1) ошибка в символе 6 т.е в i3. Тогда сообщение 1001

Для 1 1 0 1 1 0 0 таблица

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 |  |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |  |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

Код (0, 1, 0) ошибка во 2 бите. Сообщение 0100.

Для 0 1 1 0 1 1 0 таблица.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 |  |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |  |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

Код (0, 1, 0) ошибка во 2 символе это r2 . Сообщение 1110.

Все сообщение 0001 1001 0100 1110.

Число по варианту 12. Последовательность

0 1 1 1 0 0 0 1 1 0 0 0 0 0 0

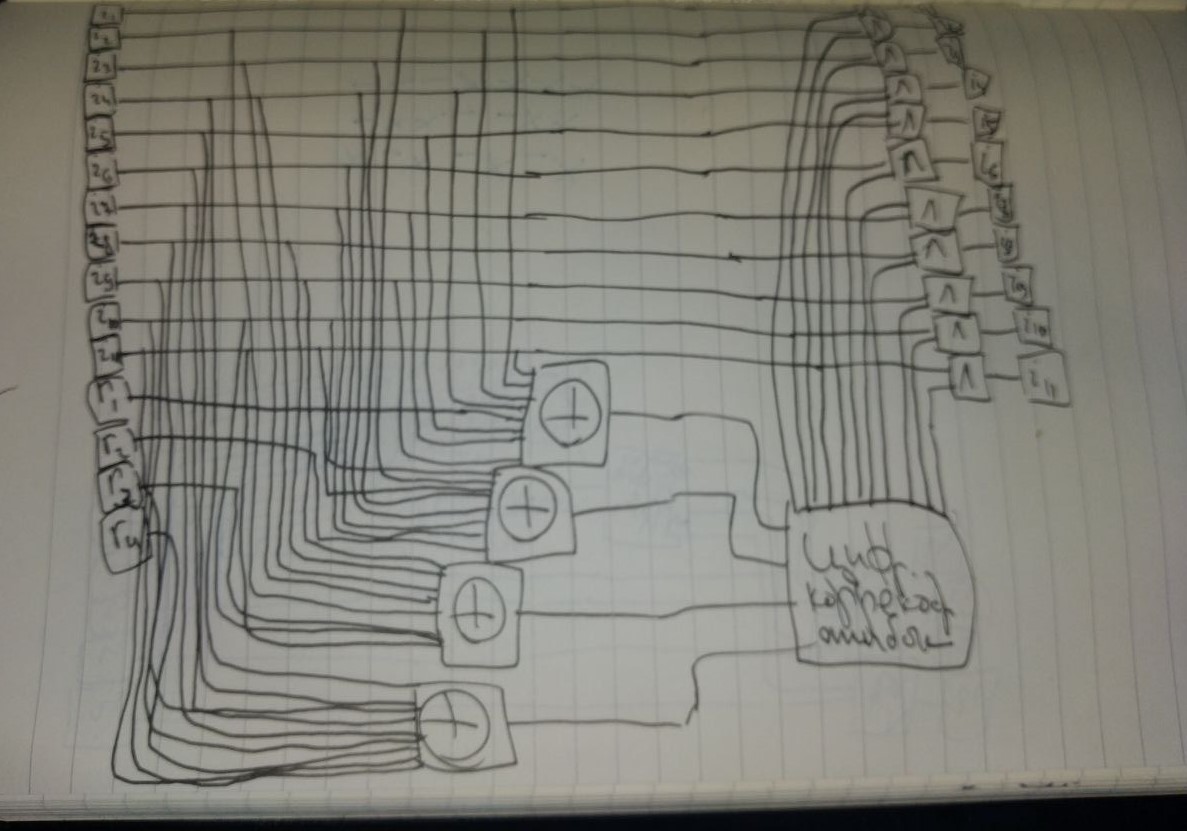


Рисунок . Схема декодирования классического кода Хемминга (15, 11)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 | R4 | I5 | I6 | I7 | I8 | I9 | I10 | I11 |  |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Ошибка в 4 символе. Это r3 . Тогда сообщение 10001000000

Тогда проверочных разрядов должно быть

Тогда r = 11 – Минимальное число проверочных разрядов.

# Программа.

import java.util.Scanner;  
  
public class DecHem {  
 public static void main(String[] args){  
 String str;  
 Scanner scanner = new Scanner(System.*in*);  
 str = scanner.next();  
 int r1, r2, r3, i1, i2, i3, i4, s1, s2, s3, res;  
 char[] strToCharArr = str.toCharArray();  
 r1 = Integer.*parseInt*(String.*valueOf*(strToCharArr[0]));  
 r2 = Integer.*parseInt*(String.*valueOf*(strToCharArr[1]));  
 i1 = Integer.*parseInt*(String.*valueOf*(strToCharArr[2]));  
 r3 = Integer.*parseInt*(String.*valueOf*(strToCharArr[3]));  
 i2 = Integer.*parseInt*(String.*valueOf*(strToCharArr[4]));  
 i3 = Integer.*parseInt*(String.*valueOf*(strToCharArr[5]));  
 i4 = Integer.*parseInt*(String.*valueOf*(strToCharArr[6]));  
 s1 = (r1 + i1 + i2 + i4)%2;  
 s2 = (r2 + i1 + i3 + i4)%2;  
 s3 = (r3 + i2 + i3 + i4)%2;  
 res = s1 + s2\*2 + s3\*4;  
 if (res == 3){  
 i1 = (i1 + 1) % 2;  
 } else if (res == 5){  
 i2 = (i2 + 1) % 2;  
 } else if (res == 6){  
 i3 = (i3 + 1) % 2;  
 } else if (res == 7){  
 i4 = (i4 + 1) % 2;  
 }  
 if (res != 0){  
 System.*out*.println("Ошибка в " + res + " бите");  
 }  
 System.*out*.println("Сообщение" + i1+ "" +i2 + "" + i3 + "" + i4);  
 }  
}

# Вывод.

В процессе выполнения лабораторной работы я на практике поработала с кодом Хемминга.