МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

«ИНФОРМАТИКА»

Синтез помехоустойчивого кода

Вариант 70

***Выполнил:***Студент группы P3107  
 Пшеничников Артём Дмитриевич

***Проверил:***Балакшин Павел Валерьевич

кандидат технических наук, доцент факультета ПИиКТ

Оглавление

[Задание 3](#_Toc179632978)

[Основные этапы вычисления 4](#_Toc179632979)

[Заключение 7](#_Toc179632980)

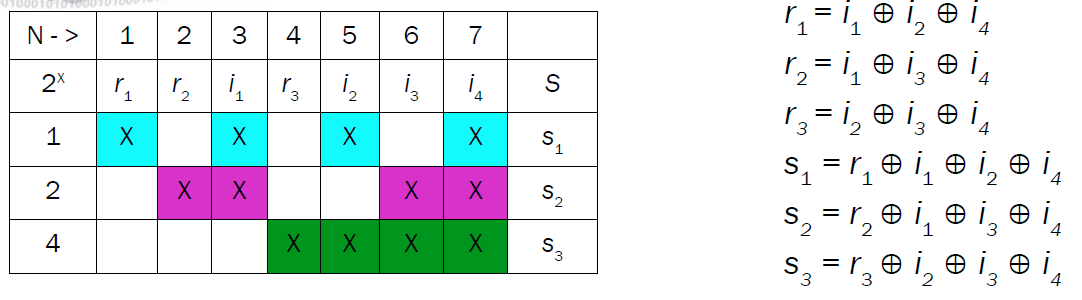
# Задание

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 = 12**3**4**5**6, 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. 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.

# Основные этапы вычисления

Часть 1

Схема декодирования классического кода хемминга (7,4)



Пример 52

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2­x | R1 | R2 | I1 | R3 | I2 | I3 | I4 |
| D | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | X |  | X |  | X |  | X |
| 2 |  | X | X |  |  | X | X |
| 4 |  |  |  | X | X | X | X |

Ошибка в бите 7, исходное сообщение 1010

Пример 89

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2­x | R1 | R2 | I1 | R3 | I2 | I3 | I4 |
| D | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | X |  | X |  | X |  | X |
| 2 |  | X | X |  |  | X | X |
| 4 |  |  |  | X | X | X | X |

Ошибка в бите 5, исходное сообщение 0010

Пример 14

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2­x | R1 | R2 | I1 | R3 | I2 | I3 | I4 |
| D | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | X |  | X |  | X |  | X |
| 2 |  | X | X |  |  | X | X |
| 4 |  |  |  | X | X | X | X |

Ошибка в бите 4, исходное сообщение 1000

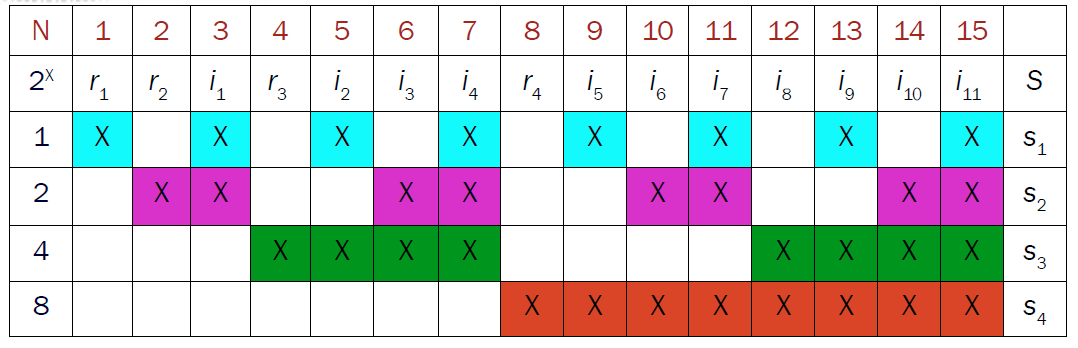
Пример 11

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2­x | R1 | R2 | I1 | R3 | I2 | I3 | I4 |
| D | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | X |  | X |  | X |  | X |
| 2 |  | X | X |  |  | X | X |
| 4 |  |  |  | X | X | X | X |

Ошибка в бите 6, исходное сообщение 1010

Часть 2

Схема декодирования классического кода хемминга (15,11)



Пример 20

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 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 |
| D | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | X |  | X |  | X |  | X |  | X |  | X |  | X |  | X |
| 2 |  | X | X |  |  | X | X |  |  | X | X |  |  | X | X |
| 4 |  |  |  | X | X | X | X |  |  |  |  | X | X | X | X |
| 8 |  |  |  |  |  |  |  | X | X | X | X | X | X | X | X |

Ошибок либо нет, либо более одной, если ошибок нет, исходное сообщение 10011000001

Часть 3

Полученное число – 744.

По формуле



Получаем 2r >= 745

Отсюда мин r = 10

Коэффициент избыточности 10 / 745 = 0.0134228

**Дополнительное задание 1:**

Исходный код:

def t\_input*()*:  
 try:  
 n = input*(*"Введите Ваш код Хемминга: "*)* if len*(*n*)* < 3:  
 raise BaseException  
 if n.count*(*"0"*)* + n.count*(*"1"*)* != len*(*n*)*:  
 raise BaseException  
 except BaseException:  
 print*(*"Плохое число, попробуйте ещё раз"*)* n = t\_input*()* return n  
  
  
def calculate\_col\_ctrl*(*data*)*:  
 from math import log2, ceil  
  
 return ceil*(*log2*(*len*(*data*)* + 1*))*def add\_to\_classic*(*data, col\_ctrl*)*:  
 return data + "0" \* *(*2 \*\* col\_ctrl - len*(*data*)* - 1*)*def find\_error*(*data, col\_ctrl*)*:  
 d = ""  
 for q in range*(*col\_ctrl*)*:  
 ind = 2 \*\* q - 1  
 step = 2 \*\* q  
 ot = 0  
 cntr = step  
 for w in range*(*ind, len*(*data*))*:  
 if cntr > 0:  
 ot ^= int*(*data*[*w*])* cntr -= 1  
 if cntr == -step:  
 cntr = step  
 d += str*(*ot*)* d = d*[*::-1*]* return int*(*d, 2*)*def get\_corr\_bits*(*data, pos\_error, corr\_bits*)*:  
 f = *[]* for q in range*(*corr\_bits*)*:  
 f.append*(*2 \*\* q - 1*)* f = f*[*::-1*]* g = list*(*data*)* if pos\_error != 0:  
 g*[*pos\_error - 1*]* = str*(*int*(*g*[*pos\_error - 1*])* ^ 1*)* for q in f:  
 g.pop*(*q*)* return g  
  
  
def print\_inf\_bits*(*corr\_inf\_bits, pos\_error*)*:  
 if pos\_error == 0:  
 print*(*"Ошибок либо нет, либо более одной"*)* else:  
 print*(*"Ошибка в бите номер", pos\_error*)* print*(*"Исходное сообщение:", "".join*(*corr\_inf\_bits*))*def main*()*:  
 data = t\_input*()* col\_ctrl = calculate\_col\_ctrl*(*data*)* data = add\_to\_classic*(*data, col\_ctrl*)* pos\_error = find\_error*(*data, col\_ctrl*)* corr\_inf\_bits = get\_corr\_bits*(*data, pos\_error, col\_ctrl*)* print\_inf\_bits*(*corr\_inf\_bits, pos\_error*)*if \_\_name\_\_ == "\_\_main\_\_":  
 main*()*

Реализована конвертация в исходное сообщение из любого кода хемминга.

# Заключение

В ходе проделанной лабораторной работы, я познакомился с различными кодами хемминга и другими способами помехоустойчивого кодирования, научился ими пользоваться и написать программу для расшифровки повреждённых сообщений с кодом хемминга.