Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Образовательная программа

«Системное и прикладное программное обеспечение»

Дисциплина «Информатика»

**Отчет**

**Лабораторная работа №2**

**Синтез помехоустойчивого кода**

**Вариант №57**

Выполнил:

Ерохин Егор Геннадьевич

Группа: P3111

Проверил:

Доцент факультета ПИиКТ

Малышева Татьяна Алексеевна

Г. Санкт-Петербург, 2024 г.

Оглавление

[Задание 3](#_Toc179378221)

[Основные этапы вычисления 4](#_Toc179378222)

[Схема декодирование классического кода Хэмминга (7;4) 4](#_Toc179378223)

[№1.42 4](#_Toc179378224)

[№1.74 5](#_Toc179378225)

[№1.106 6](#_Toc179378226)

[№1.26 6](#_Toc179378227)

[Схема декодирование классического кода Хэмминга (15;11) 7](#_Toc179378228)

[№2.62 8](#_Toc179378229)

[№(42 + 74 + 106 + 26 + 62)\*4 8](#_Toc179378230)

[Программное решение 9](#_Toc179378231)

[Результат выполнения программы 11](#_Toc179378232)

[Заключение 11](#_Toc179378233)

[Список литературы 12](#_Toc179378234)

# **Задание**

1. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
2. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
3. 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.
4. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
5. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
6. 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.
7. 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.
8. 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.

Таблица 1. Вариант 57

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | | | | **2** |
| 42 | 74 | 106 | 26 | 62 |

# Основные этапы вычисления

## Схема декодирование классического кода Хэмминга (7;4)

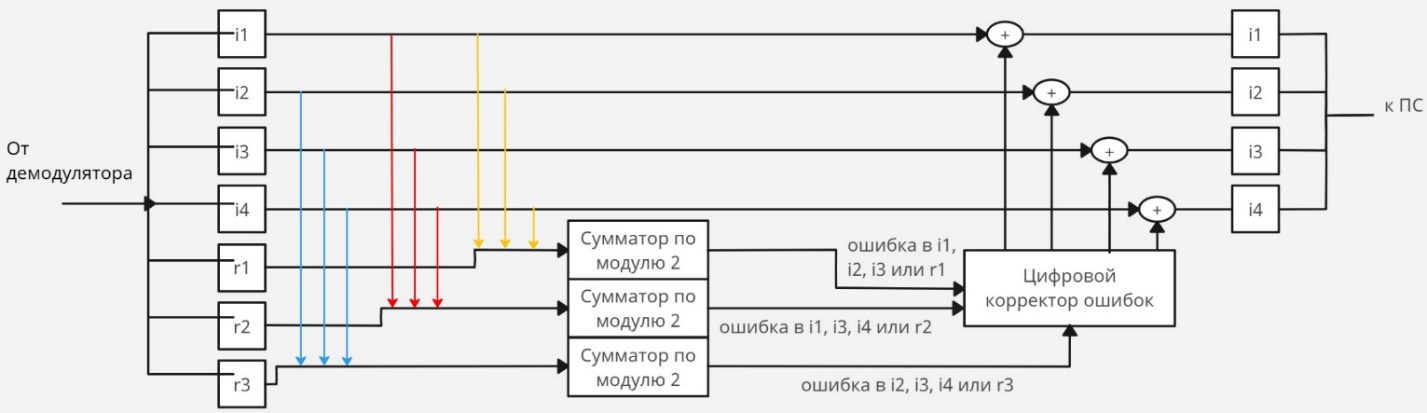


Рисунок 1. Схема декодирования классического кода Хэмминга (7;4)

## №1.42

Таблица 2. Таблица кода Хэмминга №1.42

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N -> | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 1 | 1 | 1 | 1 | 0 | 1 | 0 |  |
|  |  |  |  |  |  |  |  | S |
| 1 | X |  | X |  | X |  | X |  |
| 2 |  | X | X |  |  | X | X |  |
| 4 |  |  |  | X | X | X | X |  |

S = 010, чтобы найти позицию ошибки возьмем синдром S, перевернем его и переведем в десятичную систему счисления. Получаем, что ошибка в позиции 2, то есть ошибка в бите

Верное сообщение (информационные биты): 1010

Верное сообщение (все биты): 1011010

## №1.74

Таблица 3. Таблица кода Хэмминга №1.74

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N -> | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 0 | 0 | 1 | 1 | 1 | 0 | 1 |  |
|  |  |  |  |  |  |  |  | S |
| 1 | X |  | X |  | X |  | X |  |
| 2 |  | X | X |  |  | X | X |  |
| 4 |  |  |  | X | X | X | X |  |

S = 101, чтобы найти позицию ошибки возьмем синдром S, перевернем его и переведем в десятичную систему счисления. Получаем, что ошибка в позиции 5, то есть ошибка в бите

Верное сообщение (информационные биты): 1001

Верное сообщение (все биты): 0011001

## №1.106

Таблица 4. Таблица кода Хэмминга №1.106

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N -> | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 1 | 0 | 0 | 0 | 1 | 1 | 1 |  |
|  |  |  |  |  |  |  |  | S |
| 1 | X |  | X |  | X |  | X |  |
| 2 |  | X | X |  |  | X | X |  |
| 4 |  |  |  | X | X | X | X |  |

S = 101, чтобы найти позицию ошибки возьмем синдром S, перевернем его и переведем в десятичную систему счисления. Получаем, что ошибка в позиции 5, то есть ошибка в бите

Верное сообщение (информационные биты): 0011

Верное сообщение (все биты): 1000011

## №1.26

Таблица 5. Таблица кода Хэмминга №1.26

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N -> | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 1 | 1 | 0 | 0 | 0 | 0 | 1 |  |
|  |  |  |  |  |  |  |  | S |
| 1 | X |  | X |  | X |  | X |  |
| 2 |  | X | X |  |  | X | X |  |
| 4 |  |  |  | X | X | X | X |  |

S = 001, чтобы найти позицию ошибки возьмем синдром S, перевернем его и переведем в десятичную систему счисления. Получаем, что ошибка в позиции 4, то есть ошибка в бите

Верное сообщение (информационные биты): 0001

Верное сообщение (все биты): 1101001

## Схема декодирование классического кода Хэмминга (15;11)

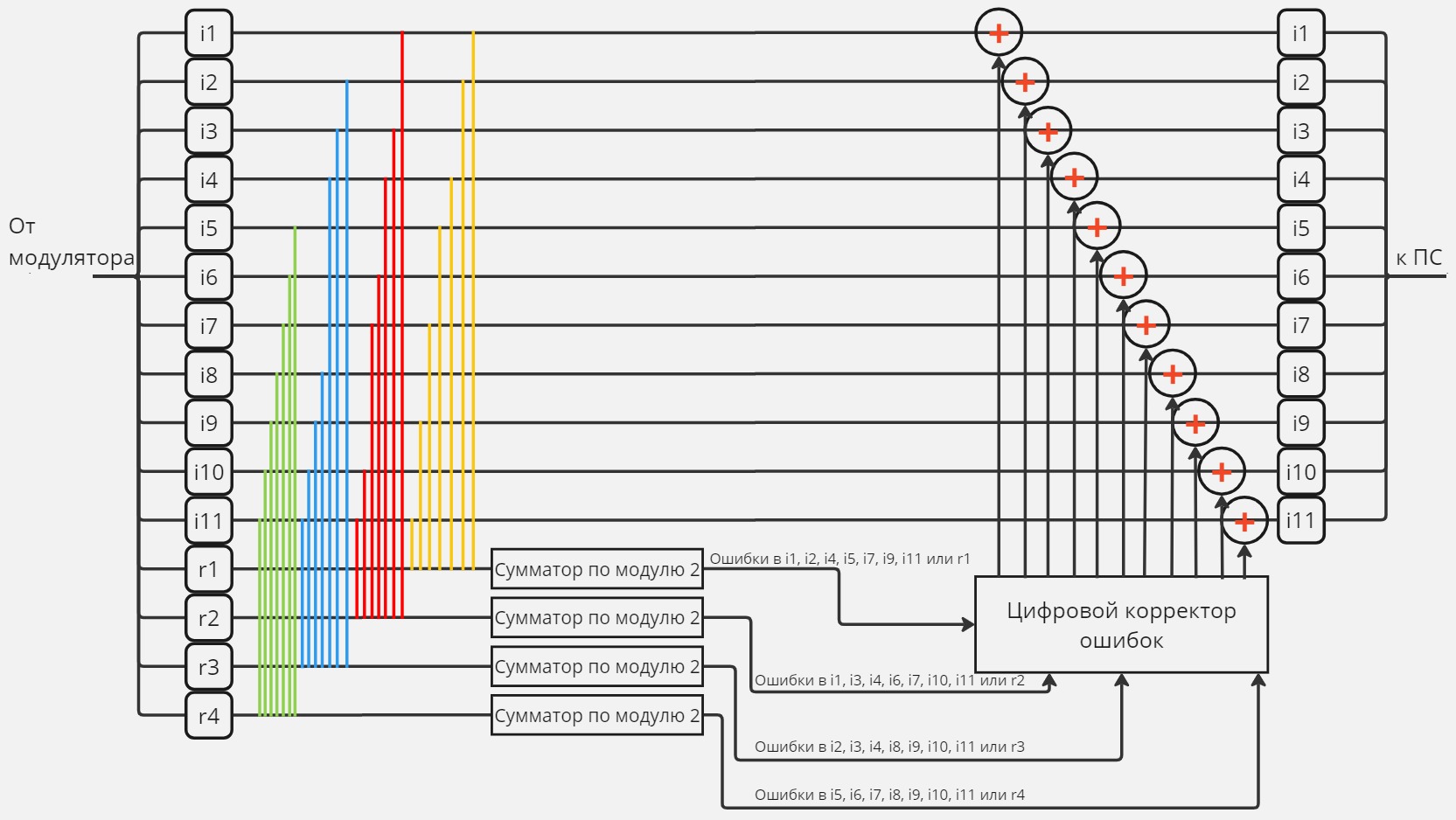


Рисунок 2. Схема декодирования классического кода Хэмминга (15;11)

## №2.62

Таблица 6. Таблица Хэмминга №2.62

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| С. | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | S |
| 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 |  |

S = 1001, чтобы найти позицию ошибки возьмем синдром S, перевернем его и переведем в десятичную систему счисления. Получаем, что ошибка в позиции 9, то есть ошибка в бите

Верное сообщение (информационные биты): 00110100011

Верное сообщение (все биты): 010001110100011

## №(42 + 74 + 106 + 26 + 62)\*4

i = (42 + 74 + 106 + 26 + 62) \* 4 = 1240

Минимальное число контрольных разрядов:

Значит наименьшее

Коэффициент избыточности — отношение числа проверочных разрядов (r) к общему числу разрядов (n = i + r), т.е.

Коэффициент избыточности

# Программное решение

class NumberAlphabetError(Exception):  
 pass  
  
  
class MessageLengthError(Exception):  
 pass  
  
  
def verification(message: str) -> str | int:  
 *"""  
 Проверка входного сообщения на корректность  
 """* try:  
 if len(message) != 7:  
 raise MessageLengthError  
 if set(message) != {"0", "1"}:  
 raise NumberAlphabetError  
 return 1  
 except NumberAlphabetError:  
 return "Неправильный формат сообщения (содержит символы отличные от цифр 0 и 1)"  
 except MessageLengthError:  
 return "Длина сообщения не равна 7"  
  
  
def message\_analysis(data: list[int]) -> str:  
 names\_index = ["r1", "r2", "i1", "r3", "i2", "i3", "i4"]  
 s1 = str(data[0] ^ data[2] ^ data[4] ^ data[6])  
 s2 = str(data[1] ^ data[2] ^ data[5] ^ data[6])  
 s3 = str(data[3] ^ data[4] ^ data[5] ^ data[6])  
 s = int(s3 + s2 + s1, 2)  
 full\_mes = f"\n{data[0]}{data[1]}{data[2]}{data[3]}{data[4]}{data[5]}{data[6]} - полное сообщение (с битами четности)"  
 if s != 0:  
 err = names\_index[s - 1]  
 correction = f"\nИсправлена ошибка в бите {'{'}{err}{'}'}"  
 if "r" in err:  
 mes = f"{data[2]}{data[4]}{data[5]}{data[6]} - сообщение (информационные биты)"  
 else:  
 if err == "i1":  
 mes = f"{(data[2] + 1) % 2}{data[4]}{data[5]}{data[6]} - сообщение (информационные биты)"  
 elif err == "i2":  
 mes = f"{data[2]}{(data[4] + 1) % 2}{data[5]}{data[6]} - сообщение (информационные биты)"  
 elif err == "i3":  
 mes = f"{data[2]}{data[4]}{(data[5] + 1) % 2}{data[6]} - сообщение (информационные биты)"  
 elif err == "i4":  
 mes = f"{data[2]}{data[4]}{data[5]}{(data[6] + 1) % 2} - сообщение (информационные биты)"  
 return mes + full\_mes + correction  
 else:  
 mes = f"{data[2]}{data[4]}{data[5]}{data[6]} - сообщение (информационные биты)"  
 correction = f"\nОшибок в сообщении не было"  
 return mes + full\_mes + correction  
  
  
def main(message: str) -> str:  
 verdict = verification(message)  
 if verdict == 1:  
 array = list(map(int, message))  
 return message\_analysis(array)  
 else:  
 return verdict  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 try:  
 print("Программа для анализа сообщения на основе классического кода Хэмминга (7;4)")  
 while True:  
 inp = input("Введите сообщение длиной 7 символов (для завершения работы программы напишите Ex): ")  
 if inp == "Ex":  
 print("Пока-пока")  
 break  
 else:  
 print(main(inp))  
 print("-" \* 94)  
 except EOFError:  
 print("Ручное завершение программы")  
 except KeyboardInterrupt:  
 print("\nРучное завершение программы")  
 finally:  
 print("Приходите ещё ^\_^")

# Результат выполнения программы

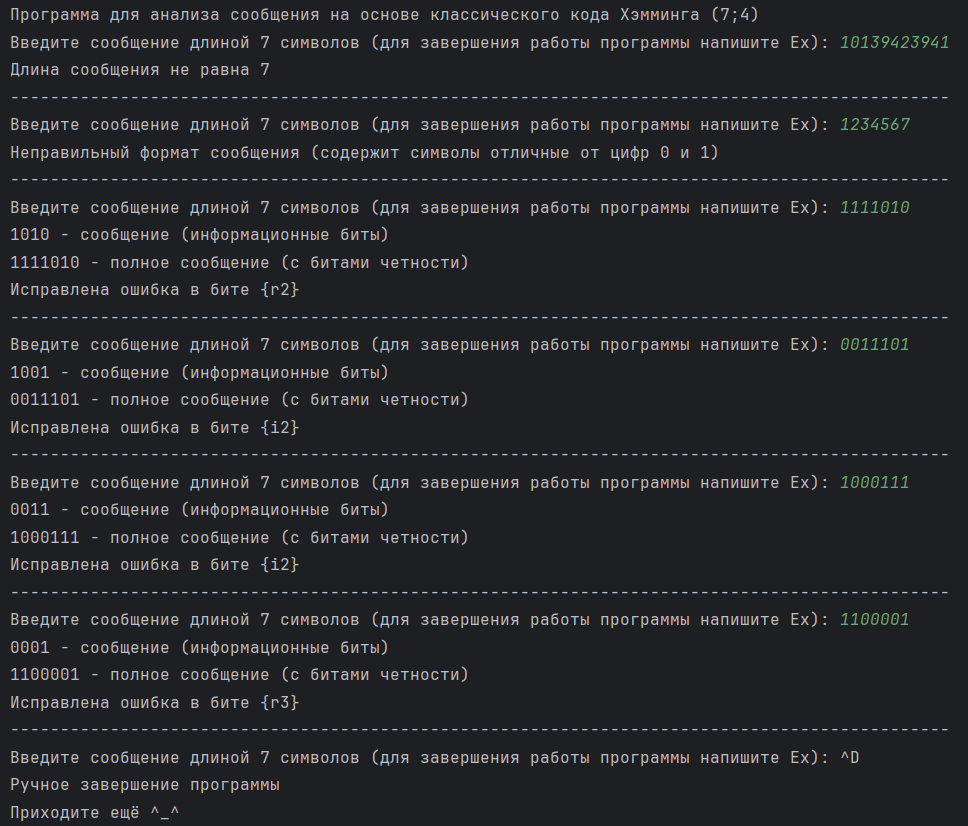


Рисунок 3. Результаты работы программы

# Заключение

В ходе выполнения лабораторной работы я познакомился со схемами Хэмминга (7;4) и (15;11). Научился находить поврежденные биты и исправлять возникшие ошибки.

# Список литературы

1. Wikipedia – Код Хэмминга:

<https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A5%D1%8D%D0%BC%D0%BC%D0%B8%D0%BD%D0%B3%D0%B0>

1. Книга «The\_Art\_of\_Doing\_Science\_and\_Engineering»:

<http://worrydream.com/refs/Hamming-TheArtOfDoingScienceAndEngineering.pdf>

1. Презентация: Информатика\_2024\_2025\_Лекция\_2.pdf