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:***Рязанов Никита Сергеевич

студент группы P3107

***Проверил:***Балакшин Павел Валерьевич

кандидат технических наук, доцент факультета ПИиКТ

**Содержание**

[Задание 3](#_Toc179544660)

[Основные этапы вычисления 4](#_Toc179544661)

[Заключение 9](#_Toc179544662)

[Список литературы 10](#_Toc179544663)

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, 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, 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: 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.

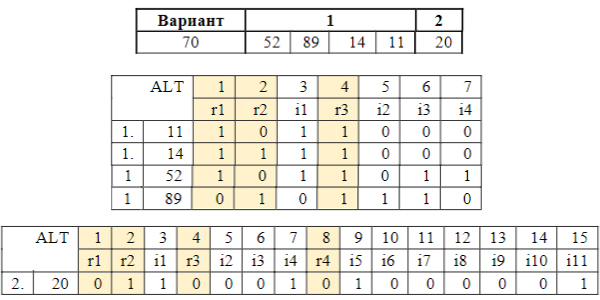


Рисунок . Сообщения в соответствии с вариантом

Основные этапы вычисления

Код Хэмминга (7, 4)

Схема декодирования классического кода Хэмминга (7, 4) изображена ниже (рис. 2).

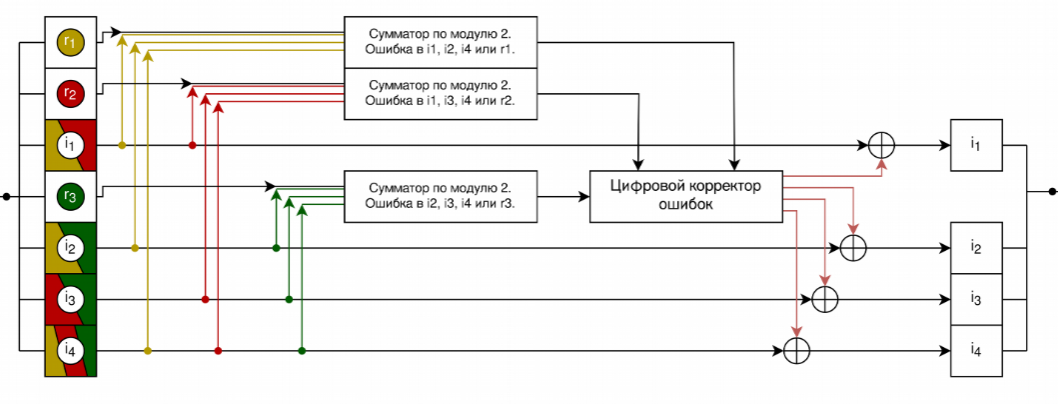


Рисунок 2. Схема декодирования классического кода Хэмминга (7, 4)

* Сообщение под номером 1.52

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2n | r1  **1** | r2  **0** | i1  **1** | r3  **1** | i2  **0** | i3  **1** | i4  **1** | s |
| 1 | X |  | X |  | X |  | X | s1 |
| 2 |  | X | X |  |  | X | X | s2 |
| 4 |  |  |  | X | X | X | X | s3 |

Таблица 1. Таблица кода Хэмминга для сообщения №1.52

**Исходное сообщение***:* 1011011

**Синдром:**

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

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

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

s = (s1,s2,s3) = (1, 1, 1)

Из полученного синдрома ясно, что ошибка находится в бите i4.

**Исправленное сообщение:** 1011010

* Сообщение под номером 1.89

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2n | r1  **0** | r2  **1** | i1  **0** | r3  **1** | i2  **1** | i3  **1** | i4  **0** | s |
| 1 | X |  | X |  | X |  | X | s1 |
| 2 |  | X | X |  |  | X | X | s2 |
| 4 |  |  |  | X | X | X | X | s3 |

Таблица 2. Таблица кода Хэмминга для сообщения №1.89

**Исходное сообщение***:* 0101110

**Синдром:**

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

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

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

s = (s1,s2,s3) = (1, 0, 1)

Из полученного синдрома ясно, что ошибка находится в бите i2.

**Исправленное сообщение:** 0101010

* Сообщение под номером 1.14

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2n | r1  **1** | r2  **1** | i1  **1** | r3  **1** | i2  **0** | i3  **0** | i4  **0** | s |
| 1 | X |  | X |  | X |  | X | s1 |
| 2 |  | X | X |  |  | X | X | s2 |
| 4 |  |  |  | X | X | X | X | s3 |

Таблица 3. Таблица кода Хэмминга для сообщения №1.14

**Исходное сообщение***:* 1111000

**Синдром:**

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

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

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

s = (s1,s2,s3) = (0, 0, 1)

Из полученного синдрома ясно, что ошибка находится в бите r3.

**Исправленное сообщение:** 1110000

* Сообщение под номером 1.11

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2n | r1  **1** | r2  **0** | i1  **1** | r3  **1** | i2  **0** | i3  **0** | i4  **0** | s |
| 1 | X |  | X |  | X |  | X | s1 |
| 2 |  | X | X |  |  | X | X | s2 |
| 4 |  |  |  | X | X | X | X | s3 |

Таблица 4. Таблица кода Хэмминга для сообщения №1.11

**Исходное сообщение***:* 1011000

**Синдром:**

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 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1

s = (s1,s2,s3) = (0, 1, 1)

Из полученного синдрома ясно, что ошибка находится в бите i3.

**Исправленное сообщение:** 1011010

Код Хэмминга (15, 11)

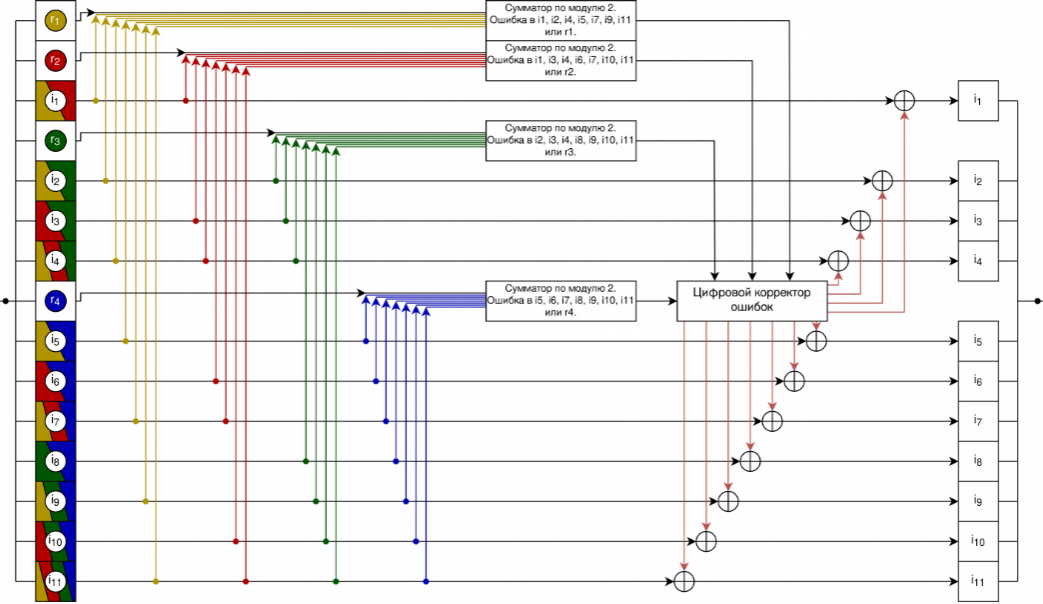
Схема декодирования классического кода Хэмминга (15,11) показана ниже (рис. 3).

Рисунок 3. Схема декодирования классического кода Хэмминга (15, 11)

* Сообщение под номером 2.20

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| 2n | r1  **0** | r2  **1** | i1  **1** | r3  **0** | i2  **0** | i3  **0** | i4  **1** | r4  **0** | i5  **1** | i6  **0** | i7  **0** | i8  **0** | i9  **0** | i10  **0** | i11  **1** | 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 |

Таблица 5. Таблица кода Хэмминга для сообщения №2.20

**Исходное сообщение***:* 011000101000001

**Синдром:**

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 ⊕ i5 ⊕ i7 ⊕ i9 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 = 0

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

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 = 0

s4 = r4 ⊕ i5 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 = 0

s = (s1,s2, s3, s4) = (0, 0, 0, 0)

Из полученного синдрома ясно, что ошибок в сообщении нет.

**Исправленное сообщение:** 011000101000001

Расчет кода Хэмминга

Вычисление минимального числа проверочных разрядов и коэффициента избыточности для *i* = (11 + 14 + 52 + 89 + 20) \* 4 = 744 информационных разрядов:

* Минимальное число проверочных разрядов *r*:

2r ≥ r + i + 1 → 2r ≥ r + 745 → rmin = 10

* Коэффициент избыточности *k*:

k = r / (i + r) = 10 / 754 ≈ 0.0133

Дополнительное задание №1

Для выполнения дополнительного задания №1 была написана программа на языке программирования Python, которая на вход получает набор цифр «0» и «1», записанных подряд, анализирует это сообщение на основе кода Хэмминга, а затем выдает правильное сообщение (только информационные биты) и указывает бит с ошибкой при его наличии. Пример работы программы представлен ниже (рис. 4).

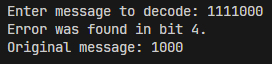


Рисунок 4. Пример работы программы

Исходный код программы:

* <https://github.com/s4dnex/itmo-labs/blob/main/informatics/lab3/src/hamming_decoder.py>

Заключение

В ходе выполнения лабораторной работы был изучен принцип работы кода Хэмминга. Был написан собственный декодировщик, который исправляет поврежденные в процессе передачи бинарные сообщения.

Список литературы

1. Python Docs. Документация по языку программирования Python. Версия 3.10. [Электронный ресурс] – Режим доступа: <https://docs.python.org/3.10/>
2. Балакшин П.В., Соснин В.В. Информатика. Методическое пособие. – Режим доступа: [https://picloud.pw/media/resources/posts/2018/02/19/%D0%9C%D0%B5%D1%82%  
   D0%BE%D0%B4%D0%B8%D1%87%D0%BA%D0%B0.pdf](https://picloud.pw/media/resources/posts/2018/02/19/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%BA%D0%B0.pdf)