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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

Синтез помехоустойчивого кода

Вариант 31

Выполнил:

Шалабодов Ярослав Дмитриевич, Р3110

Проверил:

к.п.н. доцент, доцент Авксентьева Елена Юрьевна

Санкт-Петербург 2025

**Содержание:**

[**1.** **Задания** 3](#_Toc210998487)

[**2.** **Основные этапы вычисления** 4](#_Toc210998488)

[**3.** **Дополнительное задание** 7](#_Toc210998489)

[**4.** **Заключение** 8](#_Toc210998490)

[**5.** **Список используемой литературы** 9](#_Toc210998491)

# **Задания**

1.Построить схему декодирования классического кода Хэмминга (7;4), которую представить в виде изображения. Показать, исходя из выбранных вариантов сообщений (варианты 27, 54, 81, 108), имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

2.Построить схему декодирования классического кода Хэмминга (15;11), которую представить в виде изображения. Показать, исходя из выбранного варианта сообщений (вариант 32), имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

3.Сложить номера всех 5 вариантов заданий (27+54+81+108=270). Умножить полученное число на 4 (270\*4=1080). Принять данное число как число информационных разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.

4.Дополнительное задание. 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**

**Задание 1**

***Вариант 27***

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

s1 = r1⊕i1⊕i2⊕i4 = 1⊕1⊕0⊕1 = 1

s2 = r2⊕i1⊕i3⊕i4 = 1⊕1⊕0⊕1 = 1

s3 = r3⊕i2⊕i3⊕i4 = 0⊕0⊕0⊕1 = 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2^n | 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 |

S = (s1, s2, s3) = 111 => ошибка в символе i4

Правильное сообщение: 1110000

***Вариант 54***

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

s1 = r1⊕i1⊕i2⊕i4 = 1⊕0⊕0⊕1 = 0

s2 = r2⊕i1⊕i3⊕i4 = 1⊕0⊕1⊕1 = 1

s3 = r3⊕i2⊕i3⊕i4 = 1⊕0⊕1⊕1 = 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2^n | 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 |

S = (s1, s2, s3) = 011 => ошибка в символе i3

Правильное сообщение: 1101001

***Вариант 81***

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

s1 = r1⊕i1⊕i2⊕i4 = 1⊕0⊕1⊕1 = 1

s2 = r2⊕i1⊕i3⊕i4 = 1⊕0⊕0⊕1 = 0

s3 = r3⊕i2⊕i3⊕i4 = 0⊕1⊕0⊕1 = 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2^n | 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 |

S = (s1, s2, s3) = 100 => ошибка в символе r1

Правильное сообщение: 0100101

***Вариант 108***

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

s1 = r1⊕i1⊕i2⊕i4 = 1⊕1⊕1⊕1 = 0

s2 = r2⊕i1⊕i3⊕i4 = 0⊕1⊕1⊕1 = 1

s3 = r3⊕i2⊕i3⊕i4 = 0⊕1⊕1⊕1 = 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2^n | 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 |

S = (s1, s2, s3) = 011 => ошибка в символе i3

Правильное сообщение: 1010101

**Задание 2**

***Вариант 32***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 | r4 | i5 | i6 | i7 | i8 | i9 | i10 | i11 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

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

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

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

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| 2^x | 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 |

S = (s1, s2, s3) = 0111 => ошибка в символе i10

Правильное сообщение: 010101000000000

**Задание 3**

Число информационных разрядов в передаваемом сообщении i = 270.

Имеем формулу 2r > r + 270 + 1.

Если взять r = 8: 256 > 8 + 270 + 1 – это неверно => r = 8 мало.

Если взять r = 9: 512 > 9 + 270 + 1 – верно => r = 9 подходит.

Коэффициент избыточности равен (r)/(r+i) = 9/279 ≈ 0.32258.

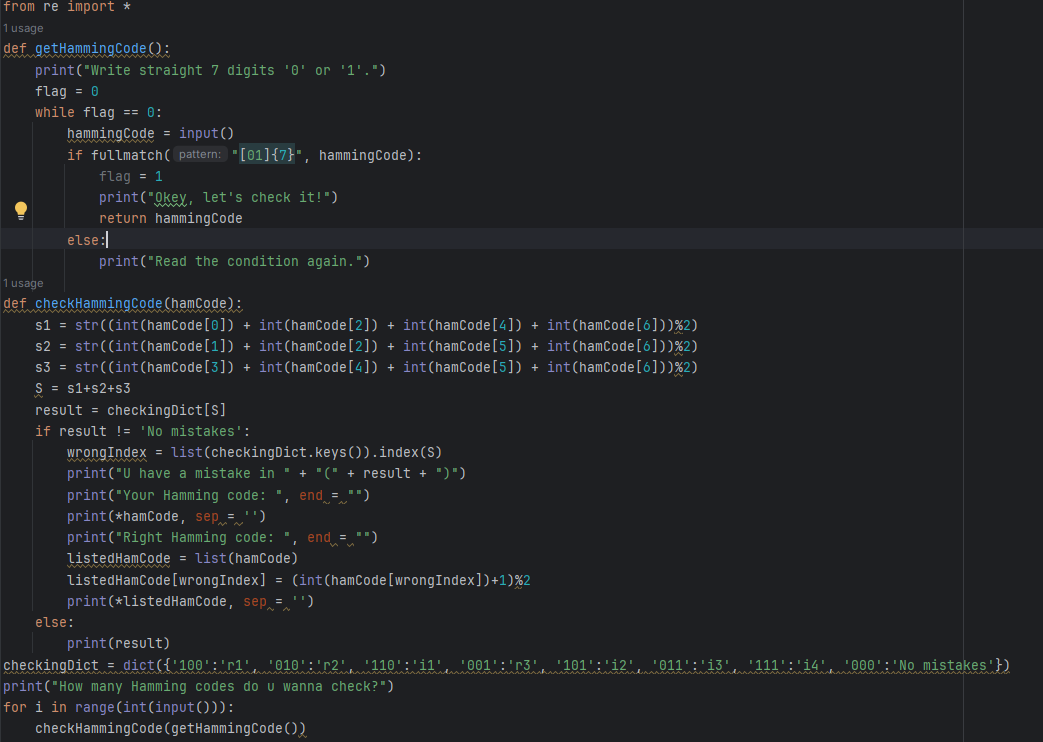
Ответ: r = 9; коэффициент избыточности ≈ 0.32258.

# **Дополнительное задание**

Функция getHammingCode() проверяет: длинна введённого кода равна 7 и он состоит только из «0» и «1».

Функция checkHammingCode(hamCode) проверяет код Хэмминга на верность в соответствии с правилом. Если он неверен, то выводится место, где была ошибка и исправленный код. Если всё верно, то выводится «No mistakes».

Остальные выводы сделаны в декоративных целях для того, чтобы пользователь понимал, что от него требуется.

*Рисунок 3.1 – Листинг кода программы на языке программирования Python*

# **Заключение**

В результате выполнения лабораторной работы №2 по информатике по теме «Синтез помехоустойчивого кода»:

1. Изучен код Хэмминга, синдром последовательности S.
2. Изучены таблицы кода Хэмминга и диаграммы Вена для кода Хэмминга.
3. Получены знания об определении минимального числа контрольных разрядов: 2r > r+i+1 и нахождении коэффициента избыточности, равного отношению числа контрольных разрядов к общему числу разрядов.
4. Выполнены практические задания для закрепления полученных знаний.

# **Список используемой литературы**

1. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.: ил., Приложение А «Арифметические основы вычислительных машин». URL: <https://bit.ly/4dzgo3u>.

2. Балакшин П.В., Соснин В.В. Информатика: методическое пособие. Санкт-Петербург. URL: [https://books.ifmo.ru/file/pdf/2464.pdf](https://books.ifmo.ru/file/pdf/2464.pdf%20)