НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Информатика

Лабораторная работа № 2 " Синтез помехоустойчивого кода"

Вариант 21

Выполнил студент:

Маликов Глеб Игоревич

Группа № Р3124

Преподаватель: Болдырева Елена Александровна

г. Санкт-Петербург

2022

Оглавление

[Задание 3](#_Toc115780712)

[Дополнительное задание 3](#_Toc115780713)

[Решение 4](#_Toc115780714)

[Код Хэмминга (7,4) 4](#_Toc115780715)

[Первое сообщение 4](#_Toc115780716)

[Второе сообщение 5](#_Toc115780717)

[Третье сообщение 5](#_Toc115780718)

[Четвёртое сообщение 6](#_Toc115780719)

[Код Хэмминга (15,11) 6](#_Toc115780720)

[Пятое сообщение 7](#_Toc115780721)

[Минимальное число проверочных разрядов 8](#_Toc115780722)

[Исходный код 9](#_Toc115780723)

[Вывод программы 9](#_Toc115780724)

[Вывод 10](#_Toc115780725)

[Список литературы 11](#_Toc115780726)

# Задание

Построить схему декодирования классического кода Хэмминга (7;4). Показать, исходя из выбранных вариантов сообщений, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

Построить схему декодирования классического кода Хэмминга (15;11). Показать, исходя из выбранных вариантов сообщений, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

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.

## Дополнительное задание

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.

# Решение

## Код Хэмминга (7,4)

Ниже приведена схема декодирования кода Хэмминга (7,4).

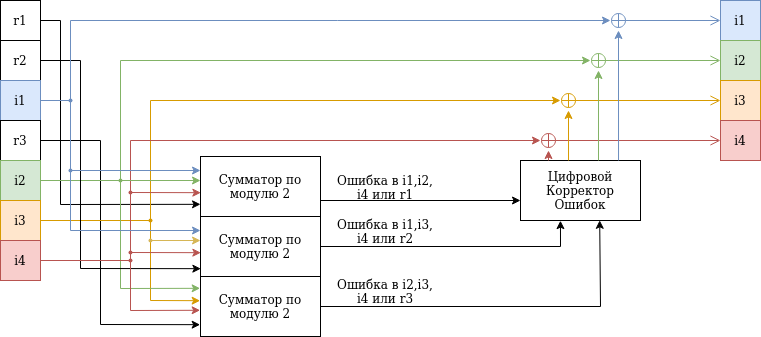


Рисунок 1 - Схема кода Хэмминга (7,4)

## Первое сообщение

Делится первое сообщение (34 = 0110010) в соответствии с таблицей кода Хэмминга (7,4):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| 2k | 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 |

Таблица 1 - Таблица кода Хэмминга (7,4)

r1 = 0; r2 = 1; i1 = 1; r3 = 0; i2 = 0; i3 = 1; i4 = 0;

Вычисляются значения контрольных бит результата:

r1 рез = i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0 ⊕ 0 = 1;

r2 рез = i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 0 = 0;

r3 рез = i2 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 = 1;

Рассчитываются синдромы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = r1 рез ⊕ r1 исх = 1 ⊕ 0 = 1;

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = r2 рез ⊕ r2 исх = 0 ⊕ 1 = 1;

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = r3 рез ⊕ r3 исх = 1 ⊕ 0 = 1;

Полученный синдром: S (1, 1, 1). Ошибочный бит седьмой (i4). Исправляется ошибочный бит и правильная последовательность равна 0110011.

## Второе сообщение

Делится второе сообщение (56 = 1111011) в соответствии с таблицей 1.

r1 = 1; r2 = 1; i1 = 1; r3 = 1; i2 = 0; i3 = 1; i4 = 1;

Вычисляются значения контрольных бит результата:

r1 рез = i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0 ⊕ 1 = 0;

r2 рез = i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 1 = 1;

r3 рез = i2 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 1 = 0;

Рассчитываются синдромы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = r1 рез ⊕ r1 исх = 0 ⊕ 1 = 1;

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = r2 рез ⊕ r2 исх = 1 ⊕ 1 = 0;

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = r3 рез ⊕ r3 исх = 0 ⊕ 1 = 1;

Полученный синдром: S (1, 0, 1). Ошибочный бит пятый (i2). Исправляется ошибочный бит и правильная последовательность равна 1111111.

## Третье сообщение

Делится третье сообщение (78 = 1000101) в соответствии с таблицей 1.

r1 = 1; r2 = 0; i1 = 0; r3 = 0; i2 = 1; i3 = 0; i4 = 1;

Вычисляются значения контрольных бит результата:

r1 рез = i1 ⊕ i2 ⊕ i4 = 0 ⊕ 1 ⊕ 1 = 0;

r2 рез = i1 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 1 = 1;

r3 рез = i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 = 0;

Рассчитываются синдромы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = r1 рез ⊕ r1 исх = 0 ⊕ 1 = 1;

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = r2 рез ⊕ r2 исх = 1 ⊕ 0 = 1;

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = r3 рез ⊕ r3 исх = 0 ⊕ 0 = 0;

Полученный синдром: S (1, 1, 0). Ошибочный бит третий (i1). Исправляется ошибочный бит и правильная последовательность равна 1010101.

## Четвёртое сообщение

Делится четвёртое сообщение (63 = 0110100) в соответствии с таблицей 1.

r1 = 0; r2 = 1; i1 = 1; r3 = 0; i2 = 1; i3 = 0; i4 = 0;

Вычисляются значения контрольных бит результата:

r1 рез = i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 = 0;

r2 рез = i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 0 = 1;

r3 рез = i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 0 = 1;

Рассчитываются синдромы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = r1 рез ⊕ r1 исх = 0 ⊕ 0 = 0;

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = r2 рез ⊕ r2 исх = 1 ⊕ 1 = 0;

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = r3 рез ⊕ r3 исх = 1 ⊕ 0 = 1;

Полученный синдром: S (0, 0, 1). Ошибочный бит четвёртый (r3). Исправляется ошибочный бит и правильная последовательность равна 0111100.

## Код Хэмминга (15,11)

Ниже приведена схема декодирования кода Хэмминга (15,11).

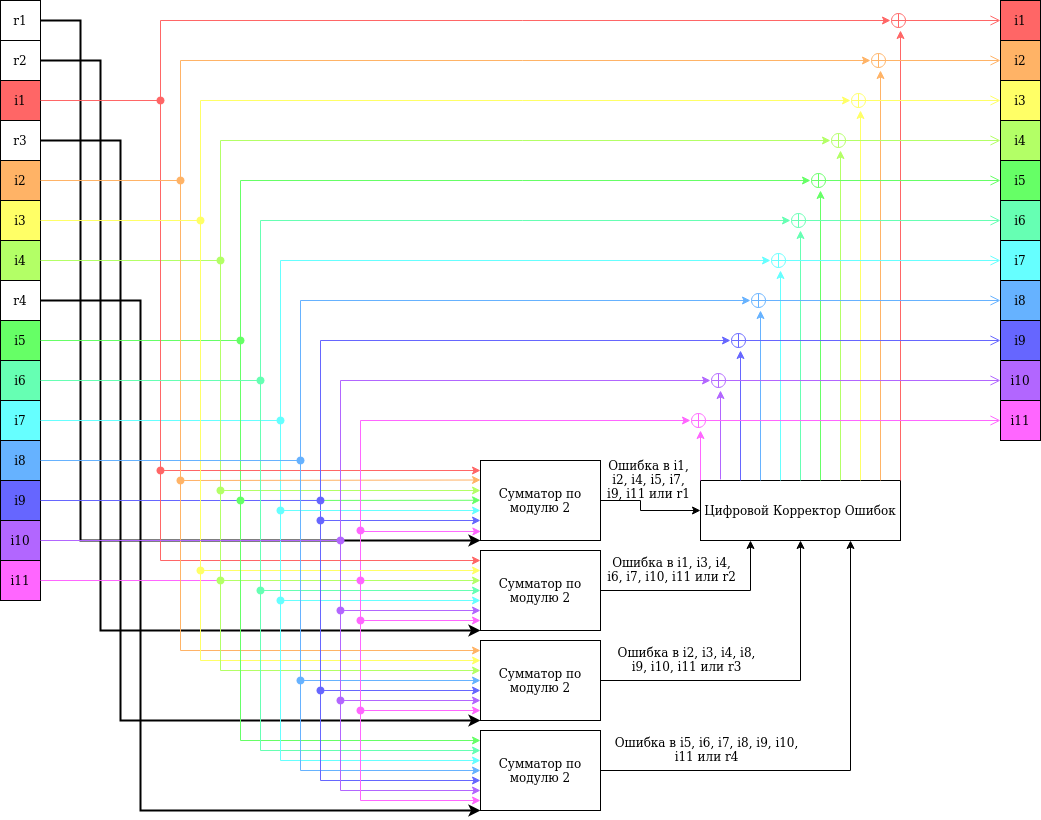


Рисунок 2 - Схема кода Хэмминга (15,11)

## Пятое сообщение

Полученное сообщение 22 = 011000101100001.

Делится сообщение в соответствии с таблицей кода Хэмминга (15,11):

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

Таблица 2 - Таблица кода Хэмминга (15,11)

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

Вычисляются значения контрольных бит результата:

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

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

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

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

Рассчитываются синдромы:

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

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

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = r3 рез ⊕ r3 исх = 0 ⊕ 0 = 0;

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

Полученный синдром: S (0, 1, 0, 1). Ошибочный бит десятый, то есть i6. Исправляется ошибочный бит и правильная последовательность равна 011000101000001.

## Минимальное число проверочных разрядов

Сложение пяти вариантов:

34 + 56 + 78 + 63 + 22 = 253

Умножение на 4:

253 × 4 = 1012

Для кодового слова с 1012 информационными разрядами, необходимо иметь такое количество проверочных разрядов (r) которое удовлетворяет неравенство 2r ≥ r + 1012 + 1. Такое неравенство верно при:

r = 10, т.е. 210 ≥ 10 + 1012 + 1; 1024 ≥ 1023

Коэффициент избыточности рассчитывается с помощью деления числа проверочных разрядов (r), на общее число разрядов (n), которое соответствует сумме проверочных разрядов с информационными (i). В данном случае, r = 10 а n = r + i = 10 + 1012 = 1022. Соответственно, коэффициент избыточности = 10 / 1022 = 0.0097847358

# Исходный код

Для выполнения дополнительного задания был использован язык программирования Python.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | import re  binary\_7: str = input("Введите семизначный бинарный код: ")  while bool(re.search(r'[^10]', binary\_7)) or (len(binary\_7) != 7):  print("Вы ввели недопустимое значение")  binary\_7 = input("Введите семизначный бинарный код: ")  lst = []  for letter in binary\_7:  lst.append(int(letter))  print("Введённый бинарный код: ", lst)  print("Введённые информационные биты: ", lst[2], lst[4], lst[5], lst[6])  r1\_source, r2\_source, r3\_source = lst[0], lst[1], lst[3]  r1\_res = (lst[2] + lst[4] + lst[6]) % 2  r2\_res = (lst[2] + lst[5] + lst[6]) % 2  r3\_res = (lst[4] + lst[5] + lst[6]) % 2  s1 = (r1\_res + r1\_source) % 2  s2 = (r2\_res + r2\_source) % 2  s3 = (r3\_res + r3\_source) % 2  err\_bit: int = (s1 + s2 \* 2 + s3 \* 4) - 1  print("Неправильный бит: ", err\_bit + 1)  lst[err\_bit] = (lst[err\_bit] + 1) % 2  print("Правильный бинарный код: ", lst)  print("Правильные информационные биты: ", lst[2], lst[4], lst[5], lst[6]) |

Доступно в <https://github.com/glebmavi/HammingCode7-4.git>

## Вывод программы

Программа была запущена с значением: 0110010. Ожидаемая правильная последовательность: 0110011.

Введите семизначный бинарный код: 0110010

Введённый бинарный код: [0, 1, 1, 0, 0, 1, 0]

Введённые информационные биты: 1 0 1 0

Неправильный бит: 7

Правильный бинарный код: [0, 1, 1, 0, 0, 1, 1]

Правильные информационные биты: 1 0 1 1

Process finished with exit code 0

# Вывод

В данной работе рассматривались два вида классического кода Хэмминга (7,4) и (15,11) и были использованы для декодирования пяти сообщении. Код Хэмминга, имеет самокорректировку и является помехоустойчивым, часто применяется для корректировки сообщении имеющих не более одной ошибки, и проверочные биты занимают значительную часть сообщения, что увеличивает его коэффициент избыточности.

# Список литературы

*Балакшин П.В., Соснин В.В., Машина Е.А.* (2020) Информатика. – СПб: Университет ИТМО. – 122 с.

*Питерсон У., Уэлдон Э.* (1972) Коды, исправляющие ошибки: Пер. с англ. М.: Мир, 1976, 594 c.