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**

Синтез помехоустойчивого кода

Вариант №78

Выполнил:

Агаев Хамза Рустам оглы

Группа: P3234

Проверил:

Балакшин Павел Валерьевич

г. Санкт-Петербург, 2023

**СОДЕРЖАНИЕ**

[Задание 3](#_Toc148643440)

[Основные этапы вычисления 5](#_Toc148643441)

[Задание 1 5](#_Toc148643442)

[Задание 2 6](#_Toc148643443)

[Задание 3 10](#_Toc148643444)

[Задание 4 11](#_Toc148643445)

[Задание 5. 13](#_Toc148643446)

[Дополнительное задание. 14](#_Toc148643447)

[Заключение 16](#_Toc148643448)

[Список использованных источников 17](#_Toc148643449)

Задание

Обязательное задание.

На основании номера варианта задания был выбран набор из 4 полученных сообщений в виде последовательности 7-символьного кода и 1 полученное сообщение в виде последовательности 11-символьного кода. Сам набор представлен в таблице 1, а задания в таблицах 2 и 3.

Таблица - Набор заданий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант № | Часть 1 | | | | Часть 2 |
| 78 | 85 | 97 | 22 | 10 | 77 |

Таблица - Сообщения 1-ой части задания

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Примера | Сообщение | | | | | | |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 |
| 85 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 97 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 22 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

Таблица - Сообщения 2-ой части задания

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № Примера | Сообщение | | | | | | | | | | | | | | |
| R1 | R2 | I1 | R3 | I2 | I3 | I4 | R4 | I5 | I6 | I7 | I8 | I9 | I10 | I11 |
| 77 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

Для полученных сообщений необходимо выполнить следующие задания:

1. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
2. 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.
3. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
4. 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.
5. 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 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

Схема декодирования классического кода Хэмминга (7; 4) представлена ниже на рисунке 1.

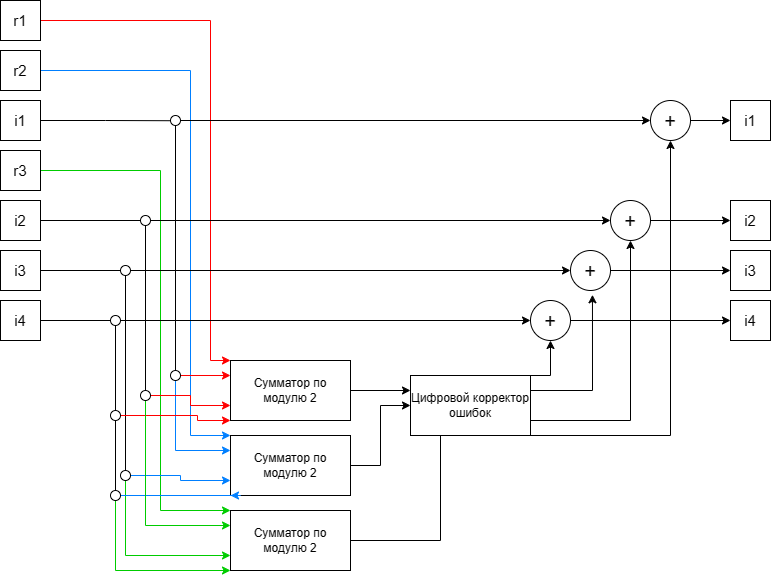


Рисунок - Схема декодирования классического кода Хэмминга (7;4)

Задание 2

Для каждого сообщения будем строить таблицу кодов Хэмминга. Затем вычислим синдром S из s1, s2, s3, сложив отмеченные напротив биты в таблице по модулю 2. Если S равен 0, то ошибки нет, иначе найдём бит с ошибкой, сопоставив неправильные синдромы и значения R, так мы найдем, какие именно R отвечали за неправильный бит и изменим этот неправильный бит.

***Пример 85.***

Таблица кодов Хэмминга (7;4) с рассматриваемым сообщением представлена в виде таблицы 4.

Таблица - Таблица кодов Хэмминга сообщения 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 0 | 0 | 0 | 0 | 1 | 1 | 0 |  |
|  | 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 |

Вычислим синдром S:

Получился синдром S(1, 1, 0). Исходя из этого, можно понять, что при передаче сообщения произошла ошибка в бите, за который отвечают только R1 и R2. Этот бит – I1. Изменим его значение с 0 на 1, чтобы исправить ошибку. В итоге, получаем сообщение 0010110.

Ответ: Ошибка в символе I1 (3-ий бит сообщения), правильное сообщение – 0010110.

***Пример 97.***

Таблица кодов Хэмминга (7;4) с рассматриваемым сообщением представлена в виде таблицы 5.

Таблица - Таблица кодов Хэмминга сообщения 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |
|  | 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 |

Вычислим синдром S:

Получился синдром S(1, 1, 0). Исходя из этого, можно понять, что при передаче сообщения произошла ошибка в бите, за который отвечают только R1 и R2. Этот бит – I1. Изменим его значение с 1 на 0, чтобы исправить ошибку. В итоге, получаем сообщение 1100110.

Ответ: Ошибка в символе I1 (3-ий бит сообщения), правильное сообщение – 1100110.

***Пример 22.***

Таблица кодов Хэмминга (7;4) с рассматриваемым сообщением представлена в виде таблицы 6.

Таблица - Таблица кодов Хэмминга сообщения 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 1 | 0 | 0 | 0 | 0 | 0 | 1 |  |
|  | 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 |

Вычислим синдром S:

Получился синдром S(0, 1, 1). Исходя из этого, можно понять, что при передаче сообщения произошла ошибка в бите, за который отвечают только R2 и R3. Этот бит – I3. Изменим его значение с 0 на 1, чтобы исправить ошибку. В итоге, получаем сообщение 1000011.

Ответ: Ошибка в символе I3 (6-ой бит сообщения), правильное сообщение – 1000011.

***Пример 10.***

Таблица кодов Хэмминга (7;4) с рассматриваемым сообщением представлена в виде таблицы 7.

Таблица - Таблица кодов Хэмминга сообщения 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
| Сообщение | 1 | 0 | 1 | 0 | 0 | 0 | 0 |  |
|  | 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 |

Вычислим синдром S:

Получился синдром S(0, 1, 0). Исходя из этого, можно понять, что при передаче сообщения произошла ошибка в бите, за который отвечают только R2. Этот бит – R2. Изменим его значение с 1 на 0, чтобы исправить ошибку. В итоге, получаем сообщение 1110000.

Ответ: Ошибка в символе R2 (2-ой бит сообщения), правильное сообщение – 1110000.

Задание 3

Схема декотирования классического кода Хэмминга (15; 11) представлена ниже на рисунке 2.

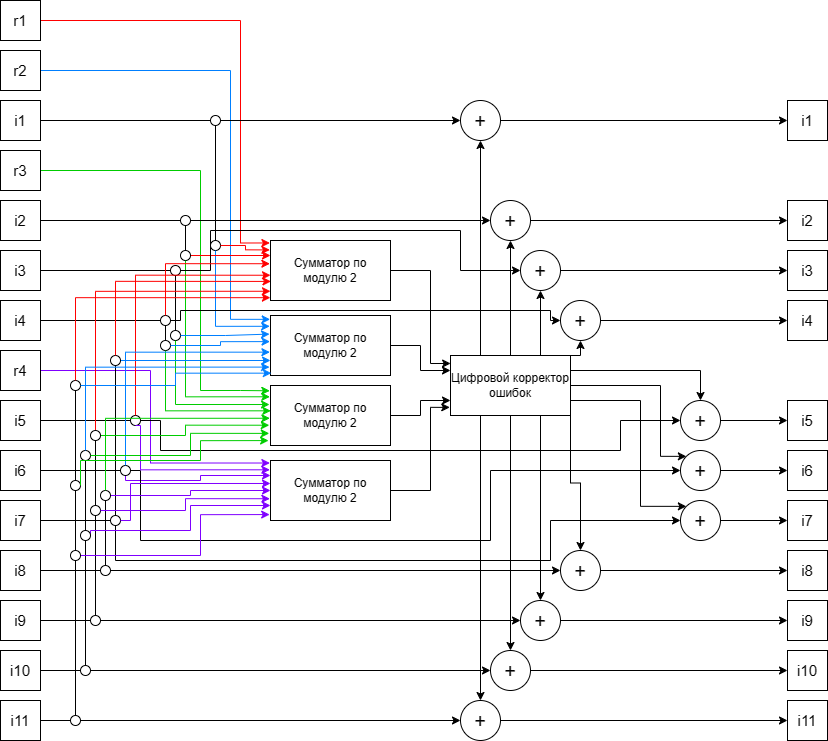


Рисунок - Схема декодирования классического кода Хэмминга (15;11)

Задание 4

Построим таблицу кодов Хэмминга (15;11). Затем вычислим синдром S из s1, s2, s3, s4, сложив отмеченные напротив биты в таблице по модулю 2. Если S равен 0, то ошибки нет, иначе найдём бит с ошибкой, сопоставив неправильные синдромы и значения R, так мы найдем, какие именно R отвечали за неправильный бит и изменим этот неправильный бит.

***Пример 77.***

Таблица кодов Хэмминга (15;11) с рассматриваемым сообщением представлена в виде таблицы 8.

Таблица - Таблица кодов Хэмминга сообщения 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| Сообщение | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |  |
|  | 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 |

Вычислим синдром S:

Получился синдром S(1, 0, 1, 0). Исходя из этого, можно понять, что при передаче сообщения произошла ошибка в бите, за который отвечают только R1 и R3. Этот бит – R2. Изменим его значение с 1 на 0, чтобы исправить ошибку. В итоге, получаем сообщение 001100011010100.

Ответ: Ошибка в символе I2 (5-ый символ сообщения), правильное сообщение – 001100011010100.

Задание 5.

Сложим номера всех выполненных заданий и умножим сумму на 4 для получения необходимого для выполнения задания числа:

Получается, что передаваемое сообщение состояло из 1164 информационных разрядов. Определим минимальное количество контрольных разрядов для такого сообщения по формуле:

*,* где R – количество контрольных разрядов, R ∈ ℕ, I – количество информационных разрядов, I ∈ ℕ.

При переборе степеней R, получаем, что для:

R = 10, – неверно;

R = 11, – верно.

Получаем, что при R ≥ 11 выражение верно, а значит, для сообщения, состоящего из 1164 информационных символов нужно как минимум 11 проверочных разрядов.

Теперь вычислим коэффициент избыточности k как отношение числа проверочных разрядов R (при R = 11) к общему числу разрядов, равному R + I.

Ответ:

Дополнительное задание.

Для выполнения данного задания была написана программа на языке Python, которая из полученного сообщения длиной в 7 бит находит ошибки, используя классический код Хэмминга (7;4). Код данной программы представлен ниже.

import math

MSG\_LEN = 7

COUNT\_OF\_CHECK\_BITS = math.ceil(math.log2(MSG\_LEN))

full\_message = input("Введите все биты сообщения (информационные и проверочные): ")

if len(full\_message) != MSG\_LEN:

    raise ValueError(f"Длина собщения должна быть равна {MSG\_LEN}.")

elif set(full\_message) not in  ({"0"}, {"1"}, {"0", "1"}):

    raise ValueError("В сообщении есть лишние символы помимо 0 и 1.")

arr\_message = list(map(int, list(full\_message)))

SYNDROM = [0] \* COUNT\_OF\_CHECK\_BITS

for i in range(COUNT\_OF\_CHECK\_BITS):

    point = 2 \*\* i

    for j in range(point - 1, MSG\_LEN, 2 \* point):

        SYNDROM[i] = (SYNDROM[i] + sum(arr\_message[j:j+point])) % 2

if SYNDROM != [0] \* COUNT\_OF\_CHECK\_BITS:

    error\_index = 0

    for i in range(COUNT\_OF\_CHECK\_BITS):

        error\_index += SYNDROM[i] \* 2 \*\* i

    power = math.log2(error\_index)

    if power == int(power):

        print(f"Ошибка в бите R{int(power) + 1} ({error\_index} бит сообщения).")

    else:

        print(f"Ошибка в бите I{error\_index-int(power) - 1} ({error\_index} бит сообщения).")

    arr\_message[error\_index - 1] = 1 - arr\_message[error\_index - 1]

else:

    print("Сообщение было правильным!")

res\_info\_message = ""

for i in range(1, MSG\_LEN + 1):

    if math.log2(i) != int(math.log2(i)):

        res\_info\_message += str(arr\_message[i - 1])

print(f"Правильное сообщение (информационные биты): {res\_info\_message}.")

res\_all\_message = "".join(map(str, arr\_message))

print(f"Правильное сообщение (все биты): {res\_all\_message}.")

Далее представлен результат работы данной программы с заданиями из варианта (рисунок 3). Из него видно, что программа верно подсчитывает коды Хэмминга и работает корректно.

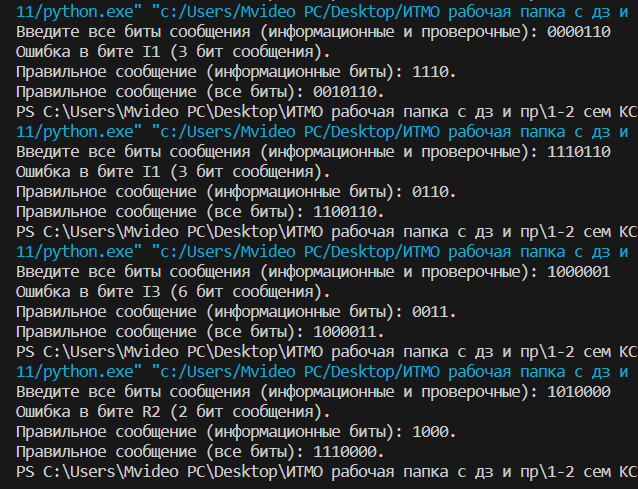


Рисунок - Результат работы программы

Заключение

В ходе лабораторной работы мы исследовали классические коды Хэмминга (7;4) и (15;11) для обнаружения и исправления ошибок в передаче данных. Мы также рассчитали коэффициент избыточности данных и разработали программу для подсчета кодов Хэмминга (7;4), что облегчает автоматизацию процесса работы с этими кодами.

Работа с кодами Хэмминга позволяет повысить надежность передачи данных и их сохранность. Полученные результаты и разработанная программа могут быть полезны в различных приложениях, где важна коррекция ошибок в передаваемой информации.

Список использованных источников

1. Балакшин Е.А., Соснин П.В., Машина В.В. Информатика. – СПб: Университет ИТМО, 2020.
2. Коды Хемминга // HiT [Электронный ресурс]. – Режим доступа: <https://all-ht.ru/inf/systems/p_0_14.html> (дата обращения: 19.10.2023).