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

Факультет программной инженерии и компьютерной техники

Лабораторная работа No2. "Синтез помехоустойчивого кода"

Номер Варианта: 20

Выполнил: Бурейко Роман Олегович

Группа: p3115

ФИО преподавателя: Авксентьева Елена Юрьевна

Санкт-Петербург, 2023

Оглавление:

Задания ……………………………………………………………………..3

Решение заданий ……………………………………………………..........4

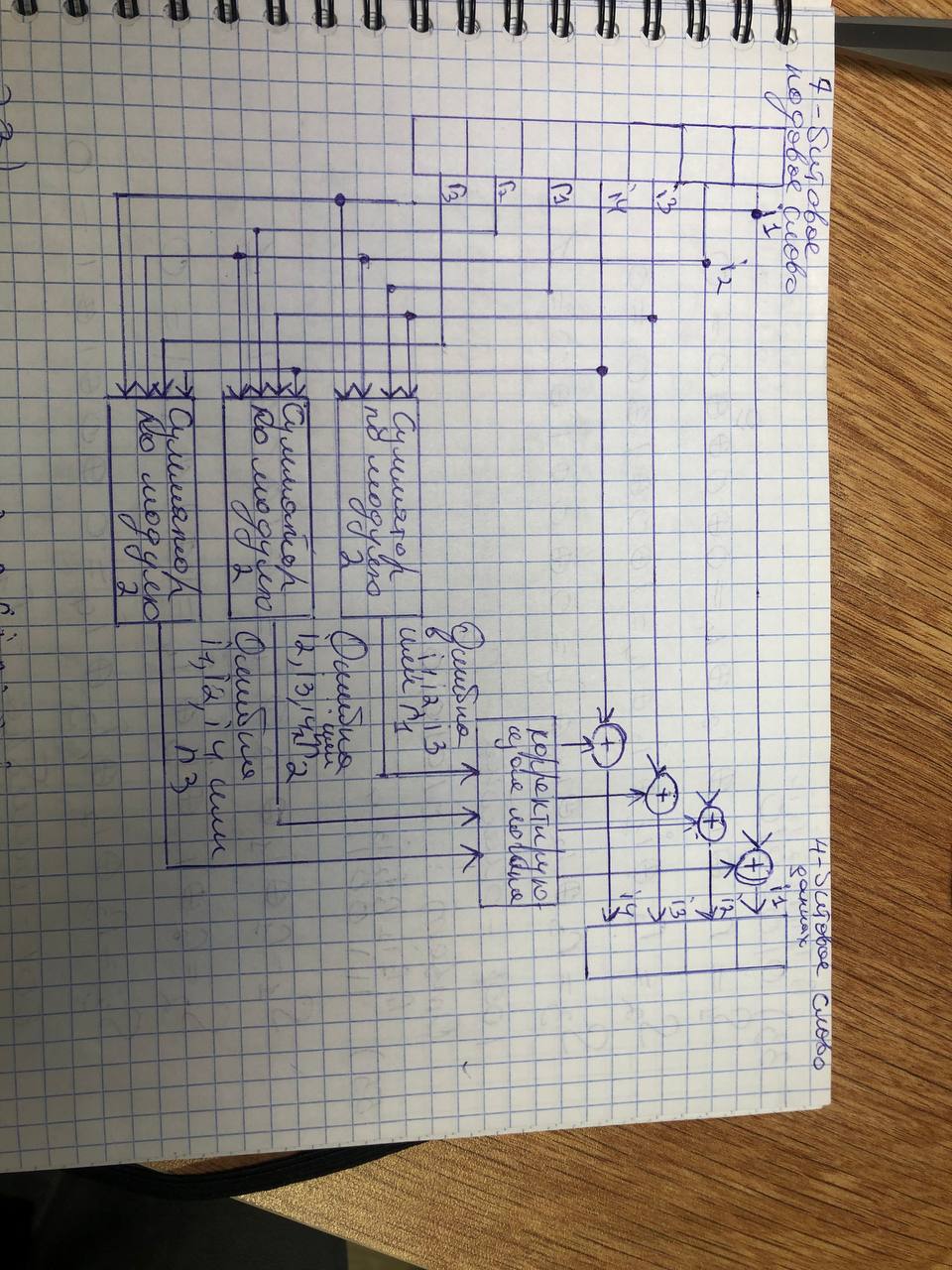
Дополнительное задание …………………………………………………..8

Вывод ………………………………………………………………………9

Список литературы ………………………………………………………..10

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 No1 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
3. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
4. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
5. Показать, исходя из выбранного варианта сообщений (по 1 у каждого – часть No2 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. **Подробно прокомментировать** и записать правильное сообщение.
6. 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.
7. Дополнительное задание No1 (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.

1.

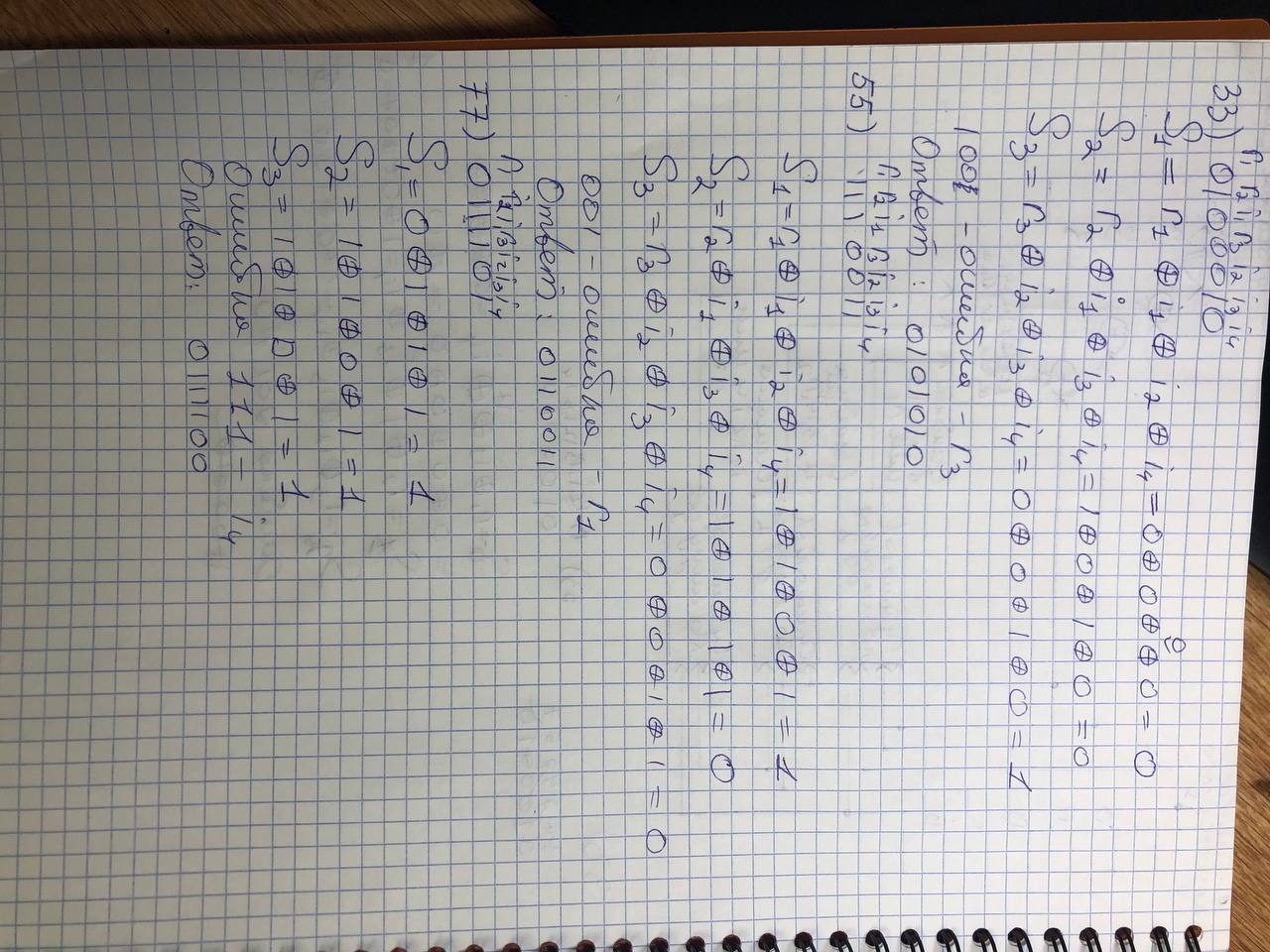
2. Для начала пронумеруем биты четности и информационные биты в виде r1, r2, i1, r3, i2, i3, i4. Посчитаем значения сумм с помощью сложения по модулю 2 (исключающее или). Первая сумма это каждый один элемент через один элемент, начиная с первого. Вторая сумма – это каждые два элемента через два элемента, начиная со второго. Третья сумма – это каждые 4 элемента через 4 элемента, начиная с четвертого.

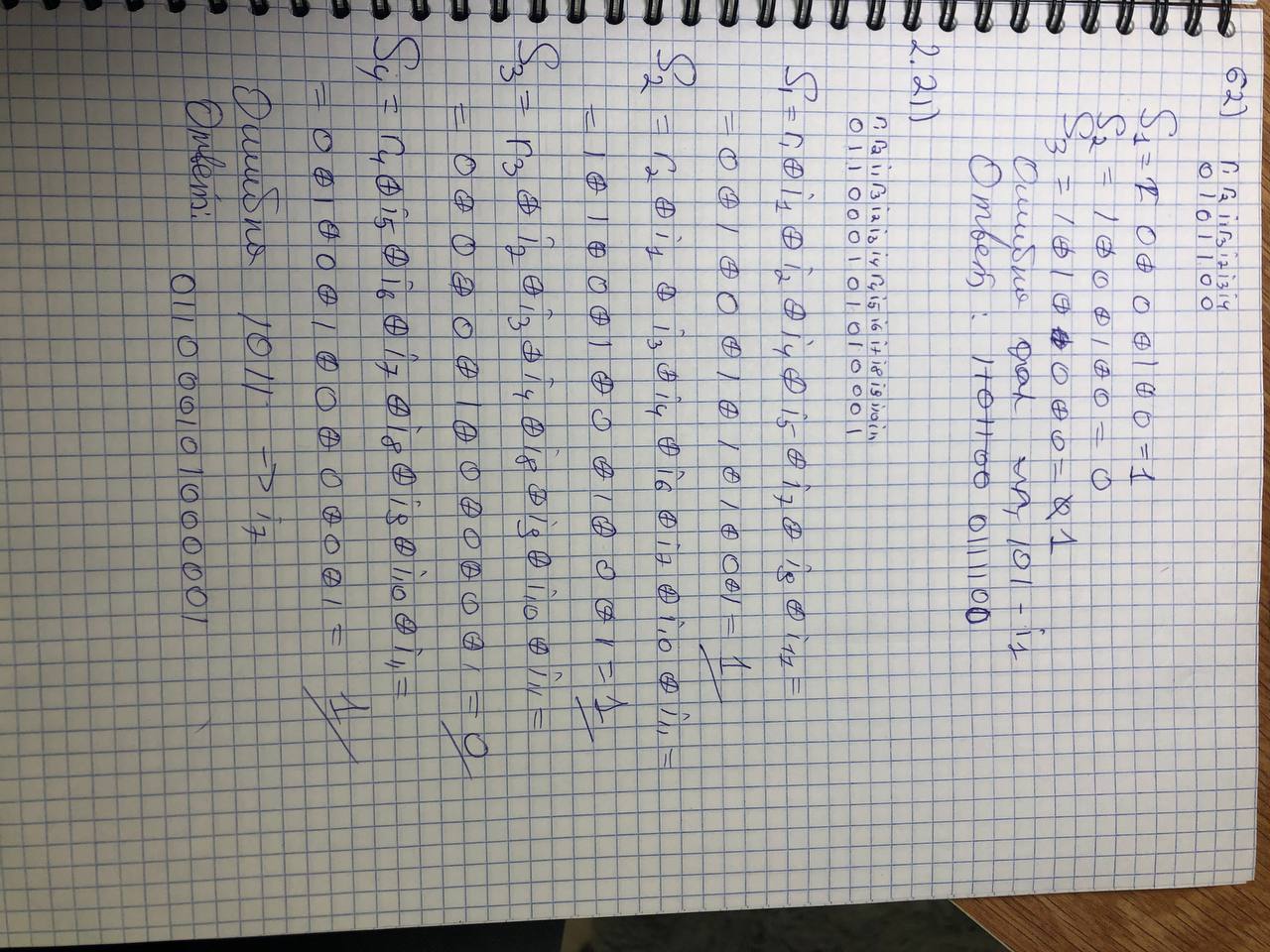
S1 = r1 ^ i1 ^ i2 ^ i4

S2 = r2 ^ i1 ^ i3 ^ i4

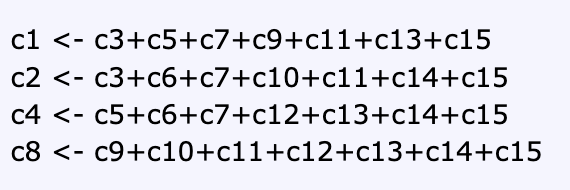
S3 = r3 ^ i2 ^ i3 ^ i4

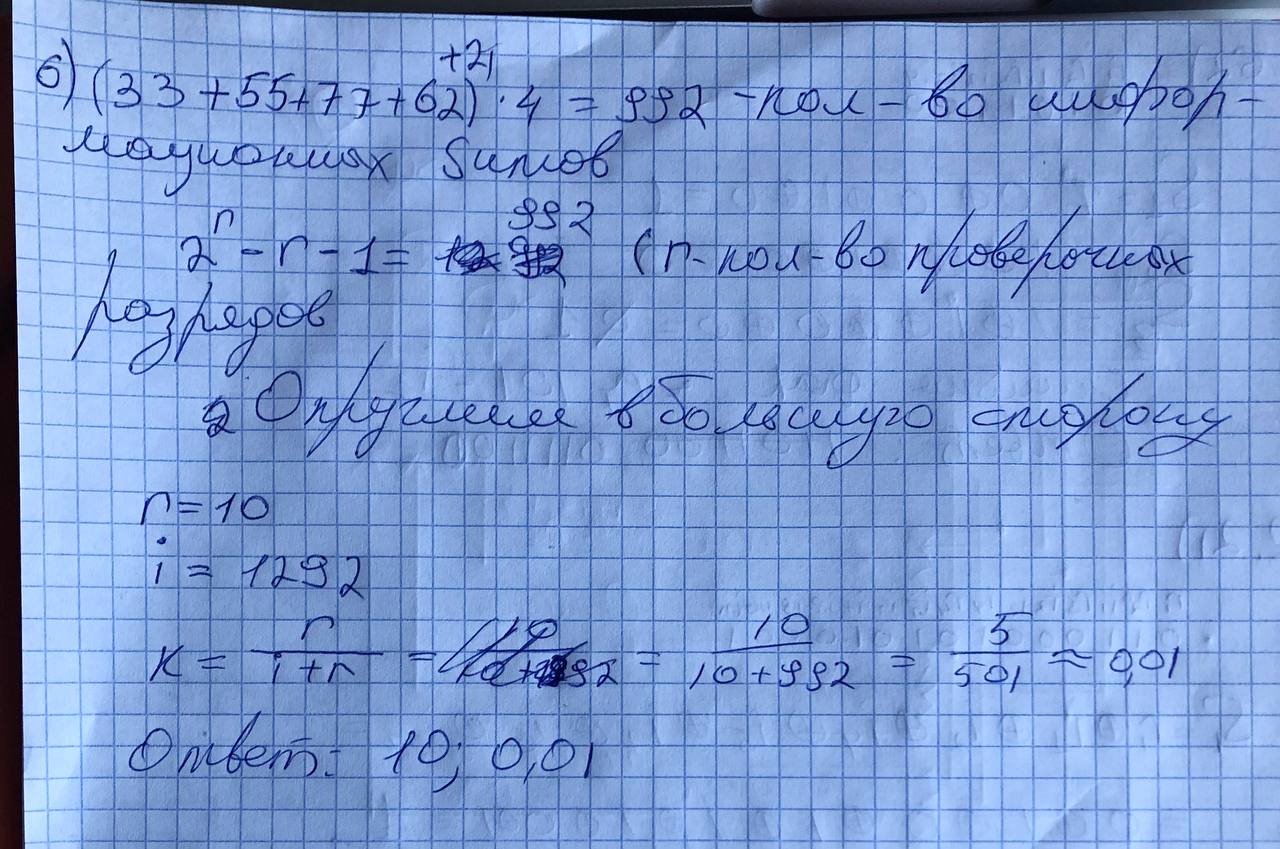
Затем необходимо склеить полученные числа в формате S3S2S1 – то есть в обратном порядке. Полученное число переводим в 10 сс, например s1 = 0, s2 = 1, s3 = 1. Получаем число 1102 = 610 => ошибка в 6 номере кода, меняем значение на противоположное. Если полученное число = 000, то в коде нет ошибок.





4.



6. 

Доп. Задание

num = str(input())

r1 = int(num[0])

r2 = int(num[1])

i1 = int(num[2])

r3 = int(num[3])

i2 = int(num[4])

i3 = int(num[5])

i4 = int(num[6])

s1 = str(r1 ^ i1 ^ i2 ^ i4)

s2 = str(r2 ^ i1 ^ i3 ^ i4)

s3 = str(r3 ^ i2 ^ i3 ^ i4)

err = s3 + s2 + s1

match err:

case "000":

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Ошибок не допущено. Сообщение = {out}")

case "001":

r1 = r1^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в r1. Сообщение = {out}")

case "010":

r2 = r2^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в r2. Сообщение = {out}")

case "011":

i1 = i1^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в i1. Сообщение = {out}")

case "100":

r3 = r3^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в r3. Сообщение = {out}")

case "101":

i2 = i2^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в i2. Сообщение = {out}")

case "110":

i3 = i3^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в i3. Сообщение = {out}")

case "111":

i4 = i4^1

out = str(i1) + str(i2) + str(i3) + str(i4)

print(f"Была допущена ошибка в i4. Сообщение = {out}")

Вывод: Я научился работать с самокорректирующим кодом Хэмминга: считывать, записывать и исправлять.

Литература:

Википедия – URL: <https://ru.wikipedia.org/wiki/>

Хабр – URL: <https://habr.com/ru/articles/>