

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2
Синтез помехоустойчивого кода
Вариант 6

Выполнил:
Жгилев Иван Игоревич
Студент группы Р3130
Преподаватель:
Рыбаков Степан Дмитриевич

Санкт-Петербург
2025 г

Содержание

Содержание	2
Задание	3
Решение Задания	4
Решение задания на написание кода	7
Создание учетной записи на Gitlab	9
Заключение	10
Список литературы.....	10

Задание

1. Построить схему декодирования классического кода Хэмминга (7;4).
2. Показать, имеются ли ошибки в четырех принятых сообщениях, выбранных исходя из варианта, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.
3. Построить схему декодирования классического кода Хэмминга (15;11).
4. Показать, имеются ли ошибки в одном принятом сообщении, выбранном исходя из варианта задания, 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.
6. Дополнительное задание №1. Сделать себе учётную запись на <https://gitlab.se.ifmo.ru/>.
7. 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.

Сообщения для пункта 2 представлены в Таблица 1

Номер сообщения	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
29	0	0	0	0	0	1	0
41	1	1	1	0	0	1	0
63	0	1	1	0	1	0	0
85	0	0	0	0	1	1	0

Таблица 1 - сообщения в виде последовательности 7-символьного кода

Сообщение для пункта 4 представлено в Таблица 2

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

Таблица 2 – сообщение в виде последовательности 15-символьного кода

Решение Задания

На Рис 1 представлена схема декодирования классического кода Хемминга (7;4)

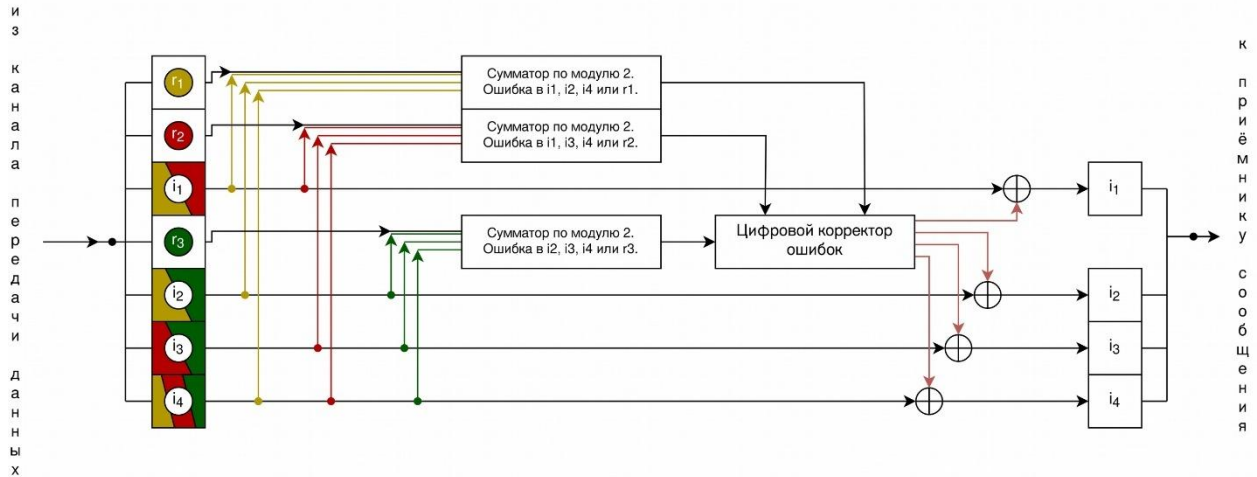


Рис 1 - схема декодирования кода Хемминга (7;4)

Рассмотрим сообщения, проверим есть ли в них ошибки и исправим.

Посчитаем синдром последовательности для каждого сообщения.

$$29) S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 0 \oplus 0 \oplus 0 \oplus 0 = 0$$

$$S_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$S_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

Тогда синдром S получился $(0,1,1)$, значит ошибка в 6 бите, то есть в i_3 , так как только он одновременно присутствует в вычислениях S_1 и S_2 , которые показали, что в полученном сообщении контрольные биты отличаются от тех, которые были переданы вместе с сообщением, тогда его исходное значение 0.

$$41) S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$S_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$S_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

Синдром $S - (0,1,1)$, аналогично первому ошибка в бите i_3 , тогда его исходное значение 0.

$$63) S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$S_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$S_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

Синдром $S - 001$, значит ошибка в бите r_3 , так как он присутствует только в вычислении S_3 , где и возникло несовпадение отправленного и вычисленного снова контрольного бита, тогда его исходное значение 1.

$$85) S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$S_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$S_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

Синдром $S = (1,1,0)$, значит ошибка в бите i_1 , так как только он одновременно присутствует в вычислениях S_1 и S_2 , где возникло несовпадение отправленного и снова вычисленного контрольного бита, тогда его исходное значение 1.

Приведем все сообщения без ошибок в Таблица 3

Номер сообщения	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
29	0	0	0	0	0	0	0
41	1	1	1	0	0	0	0
63	0	1	1	1	1	0	0
85	0	0	1	0	1	1	0

Таблица 3 - 7-символьные сообщения с исправленными ошибками

На Рис 2 представлена схема декодирования классического кода Хемминга (15;11)

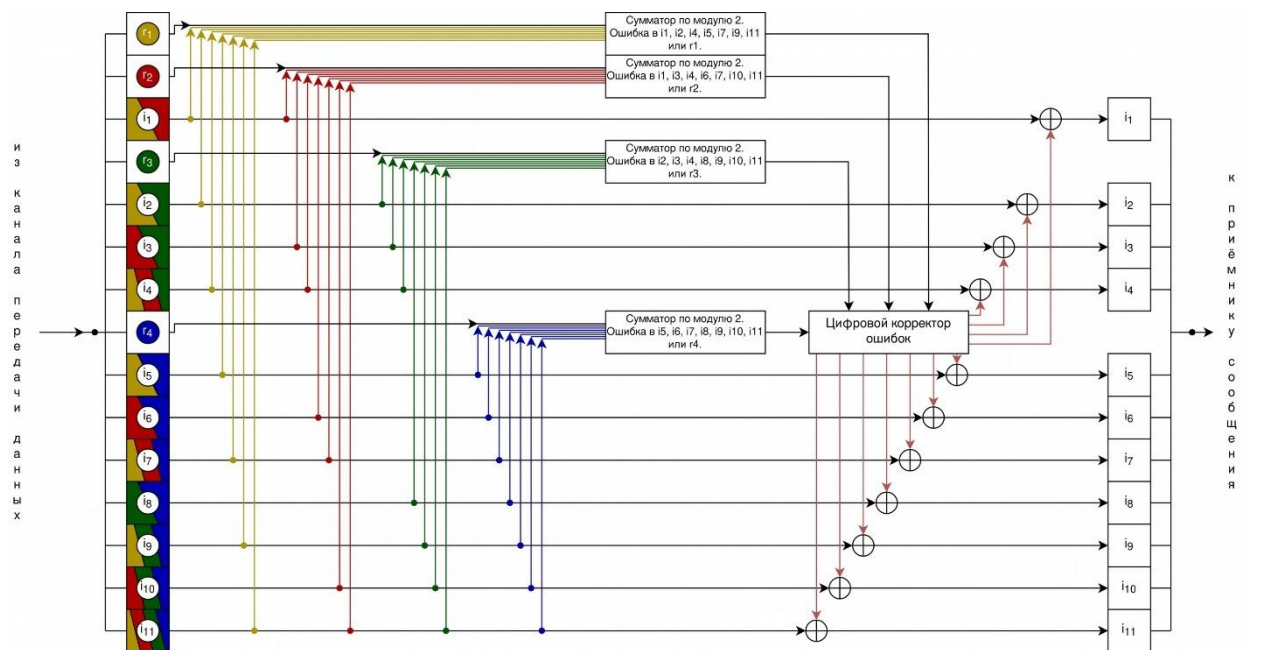


Рис 2 – схема декодирования кода Хемминга (15;11)

Рассмотрим сообщение, проверим есть ли в нем ошибка и исправим её.

$$\begin{aligned}
 7) S_1 &= r_1 \oplus i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} = \\
 &= 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1 \\
 S_2 &= r_2 \oplus i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} = \\
 &= 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0 \\
 S_3 &= r_3 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = \\
 &= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1 \\
 S_4 &= r_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = \\
 &= 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1
 \end{aligned}$$

Синдром $S = (1, 0, 1, 1)$, значит ошибка в бите i_9 , так как только он одновременно присутствует в вычислениях S_1 , S_3 и S_4 , которые и показали, что отправленные и снова вычисленные контрольные биты различаются, значит исходное значение этого бита – 1.

Приведем сообщение без ошибок в Таблица 4

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

Таблица 4 – 15-символьное сообщение с исправленной ошибкой

Выполним пункт 5.

$(29 + 41 + 63 + 85 + 7) * 4 = 900$. Значит в нашем передаваемом сообщении 900 информационных разрядов. Вычислим минимальное число проверочных разрядов.

$2^r \geq r + i + 1$, где r – число контрольных разрядов, а i – число информационных разрядов.

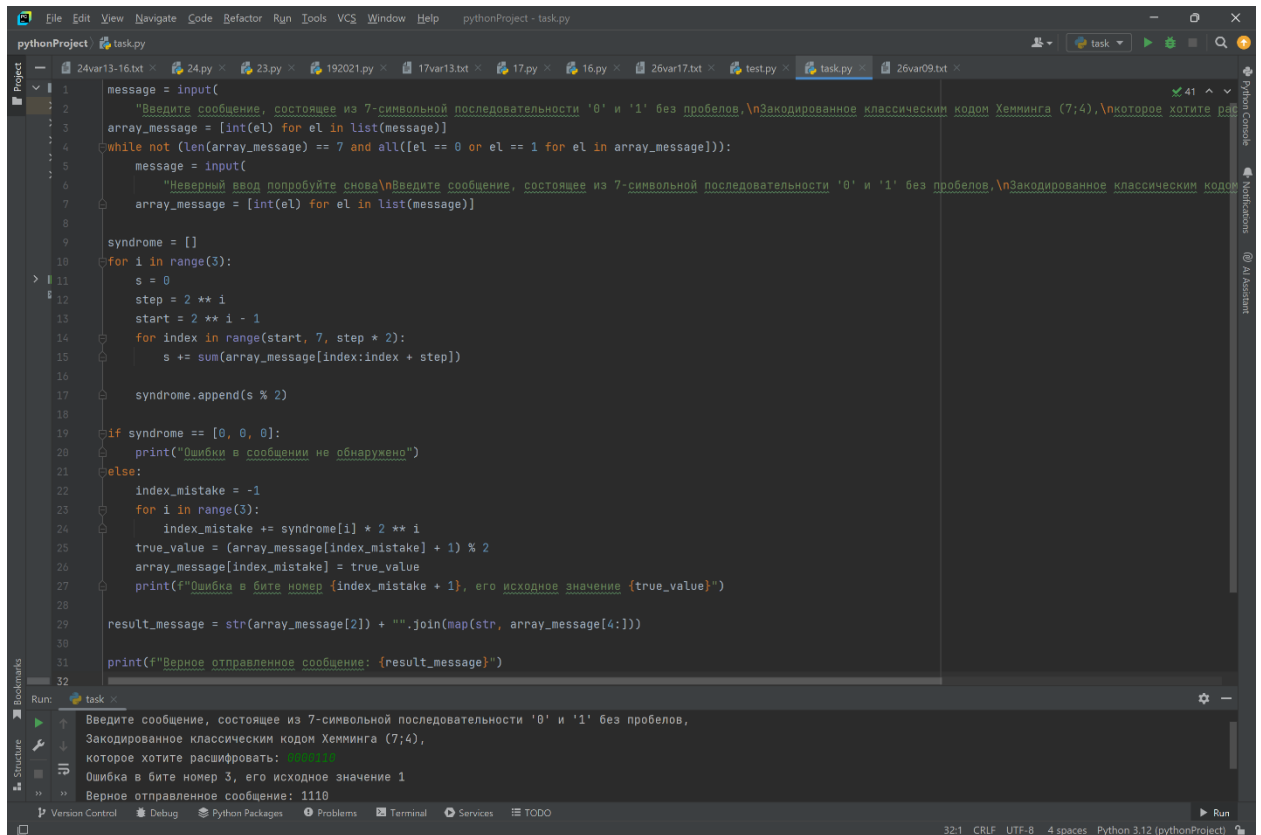
Тогда если $r = 10$, то $2^r = 2^{10} = 1024 \geq 10 + 900 + 1 = 911$, значит $r = 10$ подходит, проверим что оно является минимальным.

Если $r = 9$, то $2^r = 2^9 = 512 < 900 + 9 + 1 = 910$, значит меньше, чем $r = 10$ мы взять не можем, тогда минимальное число проверочных разрядов – 10.

Найдем коэффициент избыточности, который равен отношению проверочных разрядов к общему числу разрядов. Тогда $\frac{10}{900+10} \frac{10}{900+10} \approx 0,011$ – коэффициент избыточности.

Решение задания на написание кода

На Рис 3 представлена программа и результат её работы



```
1 message = input(
2     "Введите сообщение, состоящее из 7-символьной последовательности '0' и '1' без пробелов,\nЗакодированное классическим кодом Хемминга (7;4),\nкоторое хотите расшифровать: "
3 )
4 array_message = [int(el) for el in list(message)]
5 while not (len(array_message) == 7 and all([el == 0 or el == 1 for el in array_message])):
6     message = input(
7         "Неверный ввод попробуйте снова\nВведите сообщение, состоящее из 7-символьной последовательности '0' и '1' без пробелов,\nЗакодированное классическим кодом Хемминга (7;4),\nкоторое хотите расшифровать: "
8     )
9     array_message = [int(el) for el in list(message)]
10
11 syndrome = []
12 for i in range(3):
13     s = 0
14     step = 2 ** i
15     start = 2 ** i - 1
16     for index in range(start, 7, step * 2):
17         s += sum(array_message[index:index + step])
18     syndrome.append(s % 2)
19
20 if syndrome == [0, 0, 0]:
21     print("Ошибки в сообщении не обнаружено")
22 else:
23     index_mistake = -1
24     for i in range(3):
25         index_mistake += syndrome[i] * 2 ** i
26     true_value = (array_message[index_mistake] + 1) % 2
27     array_message[index_mistake] = true_value
28     print(f"Ошибка в бите номер {index_mistake + 1}, его исходное значение {true_value}")
29
30 result_message = str(array_message[2]) + "".join(map(str, array_message[4:]))
31
32 print(f"Верное отправленное сообщение: {result_message}")
```

Run: task

```
Введите сообщение, состоящее из 7-символьной последовательности '0' и '1' без пробелов,
Закодированное классическим кодом Хемминга (7;4),
которое хотите расшифровать: 1111111
Ошибка в бите номер 3, его исходное значение 1
Верное отправленное сообщение: 1110
```

Рис 3 – результат работы программы

Листинг кода

```
1. message = input(
2.     "Введите сообщение, состоящее из 7-символьной последовательности '0'
и '1' без пробелов,\nЗакодированное классическим кодом Хемминга
(7;4),\nкоторое хотите расшифровать: ")
3. array_message = [int(el) for el in list(message)]
4. while not (len(array_message) == 7 and all([el == 0 or el == 1 for el in
array_message])):
5.     message = input(
6.         "Неверный ввод попробуйте снова\nВведите сообщение, состоящее из
7-символьной последовательности '0' и '1' без пробелов,\nЗакодированное
классическим кодом Хемминга (7;4),\nкоторое хотите расшифровать: ")
7.     array_message = [int(el) for el in list(message)]
8.
9. syndrome = []
10. for i in range(3):
11.     s = 0
12.     step = 2 ** i
13.     start = 2 ** i - 1
14.     for index in range(start, 7, step * 2):
15.         s += sum(array_message[index:index + step])
16.
17.     syndrome.append(s % 2)
18.
19. if syndrome == [0, 0, 0]:
20.     print("Ошибки в сообщении не обнаружено")
21. else:
22.     index_mistake = -1
23.     for i in range(3):
24.         index_mistake += syndrome[i] * 2 ** i
25.     true_value = (array_message[index_mistake] + 1) % 2
26.     array_message[index_mistake] = true_value
27.     print(f"Ошибка в бите номер {index_mistake + 1}, его исходное
значение {true_value}")
28.
29. result_message = str(array_message[2]) + "".join(map(str,
array_message[4:]))
30.
31. print(f"Верное отправленное сообщение: {result_message}")
32.
```


Создание учетной записи на Gitlab

Созданная учетная запись представлена на Рис 4

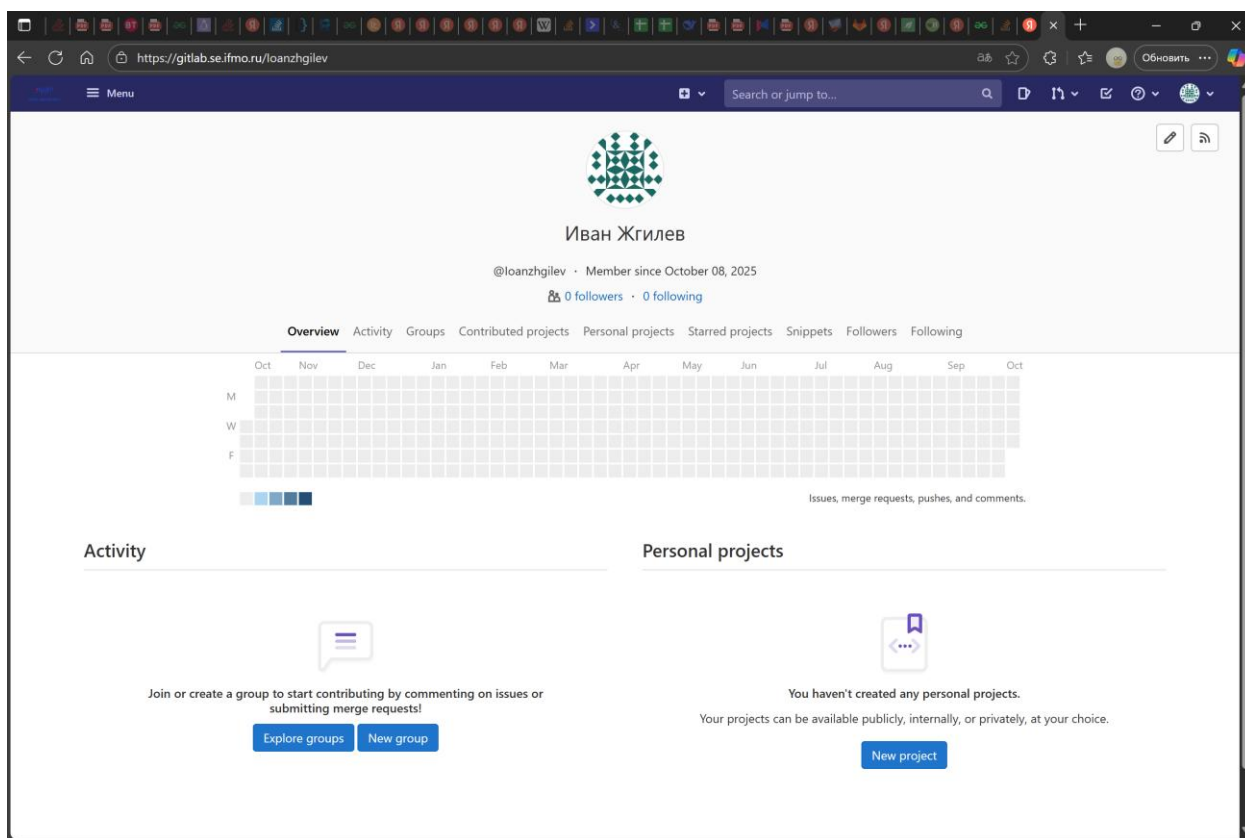


Рис 4 - учетная запись на Gitlab

Заключение

Я изучил информацию, связанную с устройством классического кода Хемминга. А именно как происходит его кодирование и декодирование. А также применил знания на практике и расшифровал сообщения, а также исправил в них ошибки.

Список литературы

1. Код Хэмминга. Пример работы алгоритма / [Электронный ресурс] // Хабр: [сайт]. — URL: <https://habr.com/ru/articles/140611/> (дата обращения: 08.10.2025).
2. Помехоустойчивое кодирование. Понятие избыточности кода. Коэффициент избыточности. Принцип обнаружения ошибок в кодах с избыточностью. / [Электронный ресурс] // Studfiles: [сайт]. — URL: <https://studfile.net/preview/12302149/page:18/> (дата обращения: 08.10.2025).