

Федеральное государственное автономное образовательное
учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и Компьютерной техники

Лабораторная работа по дисциплине «Информатика» №2

«Синтез помехоустойчивого кода»

Вариант №42

Выполнила:

Джунь Александра Васильевна

Группа Р3106

Проверил:

Балакшин Павел Валерьевич

Доцент, кандидат технических наук

Г. Санкт-Петербург 2025г.

Оглавление

Задание	3
Основные этапы вычисления:	4
Рисунок 1.	4
Задание 1 - №38:	4
Рисунок 2.	4
Рисунок 3.	4
Задание 2 - №65:	4
Рисунок 4.	5
Рисунок 5.	5
Задание 3 - №92:	5
Рисунок 6.	5
Рисунок 7.	6
Задание 4 - №7:	6
Рисунок 8.	6
Рисунок 9.	6
Задание 5 - №43:	6
Рисунок 10.	7
Рисунок 11.	7
Задание 6:.....	7
Дополнительное задание №1:	8
Рисунок 12.	8
Список использованных источников:	9

Задание:

1. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения для задания №38, где $r_1=1$, $r_2=0$, $i_1=1$, $r_3=0$, $i_2=0$, $i_3=1$, $i_4=0$.
2. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения для задания №65, где $r_1=0$, $r_2=0$, $i_1=1$, $r_3=0$, $i_2=1$, $i_3=0$, $i_4=0$.
3. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения для задания №92, где $r_1=1$, $r_2=0$, $i_1=0$, $r_3=0$, $i_2=1$, $i_3=1$, $i_4=0$.
4. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения для задания №7, где $r_1=0$, $r_2=1$, $i_1=1$, $r_3=1$, $i_2=0$, $i_3=0$, $i_4=0$.
5. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения для задания №43, где $r_1=0$, $r_2=1$, $i_1=0$, $r_3=1$, $i_2=0$, $i_3=1$, $i_4=0$, $i_5=1$, $i_6=0$, $i_7=1$, $i_8=1$, $i_9=0$, $i_{10}=0$, $i_{11}=1$.
6. Дополнительное задание №1. 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:

Схема декодирования классического кода Хэмминга (7;4) представлена на [Рисунок 1](#).

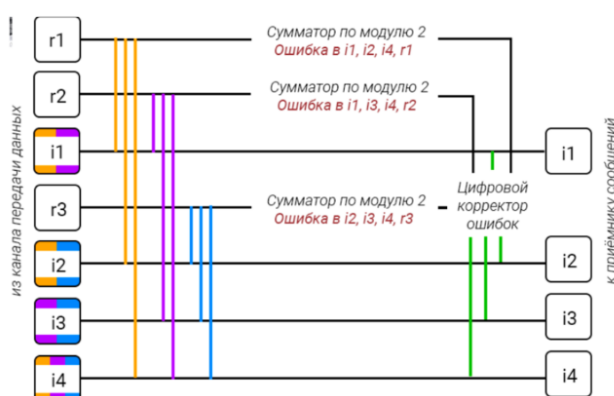


Рисунок 1.

Задание 1 - №38:

Полученные данные отображены в таблице ([Рисунок 2](#)).

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

Рисунок 2.

$$S1 = r1 \oplus i1 \oplus i2 \oplus i4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0 \text{ (чётно)}$$

$$S2 = r2 \oplus i1 \oplus i3 \oplus i4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0 \text{ (чётно)}$$

$$S3 = r3 \oplus i2 \oplus i3 \oplus i4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1 \text{ (нечётно)}$$

Полученный результат можно отобразить в таблице для более удобного поиска неверно записанного бита ([Рисунок 3](#)).

	1	2	3	4	5	6	7	
2^x	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

Рисунок 3.

Ответ: ошибка в бите r3, правильный код: 1011010

Задание 2 - №65:

Полученные данные отображены в таблице ([Рисунок 4](#)).

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

Рисунок 4.

$$S1 = r1 \oplus i1 \oplus i2 \oplus i4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0 \text{ (чётно)}$$

$$S2 = r2 \oplus i1 \oplus i3 \oplus i4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1 \text{ (нечётно)}$$

$$S3 = r3 \oplus i2 \oplus i3 \oplus i4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1 \text{ (нечётно)}$$

Полученный результат можно отобразить в таблице для более удобного поиска неверно записанного бита ([Рисунок 5.](#)).

	1	2	3	4	5	6	7	
2^X	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

Рисунок 5.

Ответ: ошибка в бите i3, правильный код: 0010110

Задание 3 - №92:

Полученные данные отображены в таблице ([Рисунок 6.](#)).

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

Рисунок 6.

$$S1 = r1 \oplus i1 \oplus i2 \oplus i4 = 1 \oplus 0 \oplus 1 \oplus 0 = 0 \text{ (чётно)}$$

$$S2 = r2 \oplus i1 \oplus i3 \oplus i4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1 \text{ (нечётно)}$$

$$S3 = r3 \oplus i2 \oplus i3 \oplus i4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0 \text{ (чётно)}$$

Полученный результат можно отобразить в таблице для более удобного поиска неверно записанного бита ([Рисунок 7.](#)).

	1	2	3	4	5	6	7	
2^X	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

Рисунок 7.

Ответ: ошибка в бите r2, правильный код: 1100110

Задание 4 - №7:

Полученные данные отображены в таблице (Рисунок 8.).

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

Рисунок 8.

$$S1 = r1 \oplus i1 \oplus i2 \oplus i4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1 \text{ (нечётно)}$$

$$S2 = r2 \oplus i1 \oplus i3 \oplus i4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0 \text{ (чётно)}$$

$$S3 = r3 \oplus i2 \oplus i3 \oplus i4 = 1 \oplus 0 \oplus 0 \oplus 0 = 1 \text{ (нечётно)}$$

Полученный результат можно отобразить в таблице для более удобного поиска неверно записанного бита (Рисунок 9.).

	1	2	3	4	5	6	7	
2^X	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

Рисунок 9.

Ответ: ошибка в бите i2, правильный код: 0111100

Задание 5 - №43:

Полученные данные отображены в таблице (Рисунок 10.).

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

Рисунок 10.

$S1 = r1 \oplus i1 \oplus i2 \oplus i4 \oplus i5 \oplus i7 \oplus i9 \oplus i11 = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$
(нечётно)

$S2 = r2 \oplus i1 \oplus i3 \oplus i4 \oplus i6 \oplus i7 \oplus i10 \oplus i11 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1$
(нечётно)

$S3 = r3 \oplus i2 \oplus i3 \oplus i4 \oplus i8 \oplus i9 \oplus i10 \oplus i11 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 1$
(нечётно)

$S4 = r4 \oplus i5 \oplus i6 \oplus i7 \oplus i8 \oplus i9 \oplus i10 \oplus i11 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$
(чётно)

	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

Рисунок 11.

Ответ: ошибка в бите i4, правильный код: 010101110110010

Задание 6:

Информационных сообщений, передаваемых в сообщении: $(38 + 65 + 92 + 7 + 43) * 4 = 980$

Если число проверочных разрядов равно r , то общее количество бит в сообщении вычисляется по формуле 2^{r-1} , информационных бит – $2^r - 1 - r$.

Найдем r , для которого выполняется неравенство $2^{r-1} - 1 - (r-1) < 980 < 2^r - 1 - r$. Если подставить $r=10$, то неравенство выполняется, значит $r=10$ подходит:

$$2^9 - 1 - 10 = 1013 < 980 < 2036 = 2^{10} - 1 - 11$$

Найду коэффициент избыточности: $r/(i+r) = 10/(10+980) \approx 0,01(!)$

Ответ: $r = 10$, коэффициент избыточности $\approx 0,01(!)$

Дополнительное задание №1:

1. 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.

Для ввода кода, который нужно проверить использую input(), далее создаю массив symbols. Если введенная строка больше 7 символов или содержит символы кроме 0 и 1, то выводится предложение «Строка должна быть набором из 7 символов и содержать только 0 или 1».

Если условие соблюдается, то дальше собираются синдромы последовательности, в моем случае их 3, потом из них составляется один синдром. Далее находится бит, в котором содержится ошибка, а после выводится правильное сообщение, состоящее из информационных битов.

```
1 string input()#принимает значение с консоли
2 symbols=['r1','r2','i1','r3','i2','i3','i4']#создаем список с проверочными и изначальными битами
3 if len(string)!=7 or bool(set(string)-{'1','0'}):#условие на длину
4     print("Строка должна быть набором из 7 символов и содержать только 0 или 1")
5 else:
6     array= list(map(int,list(string)))#членим введенное число на биты
7     s1= (array[0] + array[2] + array[4] + array[6])%2#сумма 1
8     s2= (array[1] + array[2] + array[5] + array[6])%2#сумма 2
9     s3= (array[3] + array[4] + array[5] + array[6])%2#сумма 3
10    syndrome=[s1,s2,s3]#из трех сумм формируется синдром для проверки наличия ошибки
11    if syndrome != [0,0,0]:#проверка базового наличия ошибки (если 000 - ошибки нет)
12        num=int(''.join(map(str,syndrome[::-1])),2)#переводим в 10сс, чтобы найти индекс ошибки +1, тк начинается с нуля
13        print("Найдена ошибка в символе", symbols[num-1])#выводим ошибку
14        array[num-1]=1-array[num-1]#замена неверного бита
15        rez=''.join(map(str,array))
16        print(f"Правильное сообщение: {rez[2]}{rez[4]}{rez[5]}{rez[6]}")#выдает верный ответ без проверочных битов
17        print(f"Правильное сообщение: {rez[0]}{rez[1]}{rez[2]}{rez[3]}{rez[4]}{rez[5]}{rez[6]}")#выдает верный ответ с проверочными битами
18    else: print("Сообщение передано без ошибки")
```

Рисунок 12.

Список использованных источников:

- 1) Информатика. Лекция 2 / Балакшин П. В. [Электронный ресурс] - Режим доступа: <https://cloud.mail.ru/public/nxcb/kfFLJ5kdA> (дата обращения 20.09.2025)
- 2) Информатика / Балакшин П. В. Соснин В. В. Машина Е. А. [Электронный ресурс] — Режим доступа: [..\Черновик методического пособия Информатика.pdf](#) (дата обращения: 20.09.2025)