

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

Факультет программной инженерии и компьютерной техники

## **Лабораторная работа №2**

"Синтез помехоустойчивого кода"

Вариант №73

Выполнил:  
Студент группы Р3118  
Зыков Дмитрий  
Андреевич  
Преподаватель:  
Рыбаков Степан  
Дмитриевич

г. Санкт-Петербург

2022 год

## СОДЕРЖАНИЕ

Стр.

1	Задания .....	3
2	Основные этапы вычисления .....	4
3	Вывод .....	9
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	10

## 1 Задания

Проверить двоичный код на ошибочность, если есть ошибки – исправить:

№ 55: 1110011 (классический код Хэмминга (7,4))

№ 92: 1000110 (классический код Хэмминга (7,4))

№ 17: 0010001 (классический код Хэмминга (7,4))

№ 74: 0011101(классический код Хэмминга (7,4))

№ 72: 0011100100001 (классический код Хэмминга (15,11))

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.

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.

## 2 Основные этапы вычисления

№55

	1	2	3	4	5	6	7	
Полученное сообщение	1	1	1	0	0	1	1	
$2^x$	$r_1$	$r_2$	$i_1$	$r_3$	$i_2$	$i_3$	$i_4$	$s$
1	x		x		x		x	$s_1$
2		x	x			x	x	$s_2$
4				x	x	x	x	$s_3$

Рисунок 2.1 — схема декодирования кода Хэмминга (7,4)

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 1 \oplus 1 = 0$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$$

Так как первый синдром показал ошибку, следовательно ошибка в бите с номером 1

№92

	1	2	3	4	5	6	7	
Полученное сообщение	1	0	0	0	1	1	0	
$2^x$	$r_1$	$r_2$	$i_1$	$r_3$	$i_2$	$i_3$	$i_4$	$s$
1	x		x		x		x	$s_1$
2		x	x			x	x	$s_2$
4				x	x	x	x	$s_3$

Рисунок 2.2 — схема декодирования кода Хэмминга (7,4)

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

Так как второй синдром показал ошибку, следовательно ошибка в бите с номером 2

№17

	1	2	3	4	5	6	7	
Полученное сообщение	0	0	1	0	0	0	1	
$2^x$	$r_1$	$r_2$	$i_1$	$r_3$	$i_2$	$i_3$	$i_4$	$s$
1	x		x		x		x	$s_1$
2		x	x			x	x	$s_2$
4				x	x	x	x	$s_3$

Рисунок 2.3 — схема декодирования кода Хэмминга (7,4)

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

Так как третий синдром показал ошибку, то ошибка в бите с номером 4

№74

	1	2	3	4	5	6	7	
Полученное сообщение	0	0	1	1	1	0	1	
$2^x$	$r_1$	$r_2$	$i_1$	$r_3$	$i_2$	$i_3$	$i_4$	$s$
1	x		x		x		x	$s_1$
2		x	x			x	x	$s_2$
4				x	x	x	x	$s_3$

Рисунок 2.4 — схема декодирования кода Хэмминга (7,4)

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

Так как первый и третий синдром показал ошибку, то ошибка в бите с номером  $1+4=5$ .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Полученное сообщение	0	0	1	1	1	0	0	1	0	0	0	0	1	0	0	
$2^x$	$r_1$	$r_2$	$i_1$	$r_3$	$i_2$	$i_3$	$i_4$	$r_4$	$i_5$	$i_6$	$i_7$	$i_8$	$i_9$	$i_{10}$	$i_{11}$	$s$
1	x		x		x		x		x		x		x		x	$s_1$
2		x	x			x	x			x	x			x	x	$s_2$
4				x	x	x	x					x	x	x	x	$s_3$
8								x	x	x	x	x	x	x	x	$s_4$

Рисунок 2.5 — схема декодирования кода Хэмминга (15,11)

$$s_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$s_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$s_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$s_4 = r_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$$

Так как первый, второй и третий синдромы показали ошибку, то ошибка в бите с номером  $1 + 2 + 4 = 7$ .

Сумма номеров всех 5 вариантов заданий равна  $55+92+17+74+72 = 310$  А сумма, умноженная на 4, равна 1240 По формуле определения минимального числа контрольных разрядов  $2r \geq r + i + 1$ , где  $r$  – минимальное кол-во проверочных битов,  $i$  – кол-во информационных битов, мы получим, что  $2r - r \geq 1240 + 1$  Из этого следует, что минимальное кол-во проверочных битов ( $r$ ) для 1240 битов информации равняется 11, то есть коэффициент избыточности равен  $r/(r + i) = 11/(11 + 1240) \approx 0,0088$

```

1  def ff(x):
2      if x=='1':
3          return '0'
4      else:
5          return '1'
6  print('Введите набор из 7 цифр "0" и "1" ')
7  s = input()
8  if (s.count("1")+s.count("0"))!=7 or (s.count("1")+s.count("0"))!=len(s) :
9      print('Вы ввели неправильный набор')
10 else:
11     nb=0
12     k=0
13     r1=s[0]
14     r2=s[1]
15     r3=s[3]
16     sind1 = (r1+s[2]+s[4]+s[6])
17     sind2 = (r2+s[2]+s[5]+s[6])
18     sind3 = (r3+s[4]+s[5]+s[6])
19     if sind1.count("1")%2!=0:
20         k+=1
21     if sind2.count("1")%2!=0:
22         k+=2
23     if sind3.count("1")%2!=0:
24         k+=4
25     t=(s[:k-1]+ff(s[k-1])+s[k:])
26     print(t[2]+t[4]+t[5]+t[6])
27     if k==0:
28         print('Ошибка не обнаружена')
29     else:
30         print('Ошибка в бите № ' + str(k))

```

Рисунок 2.6 — Программа для проверки кода Хэмминга в Python: <https://clck.ru/32GBEx>

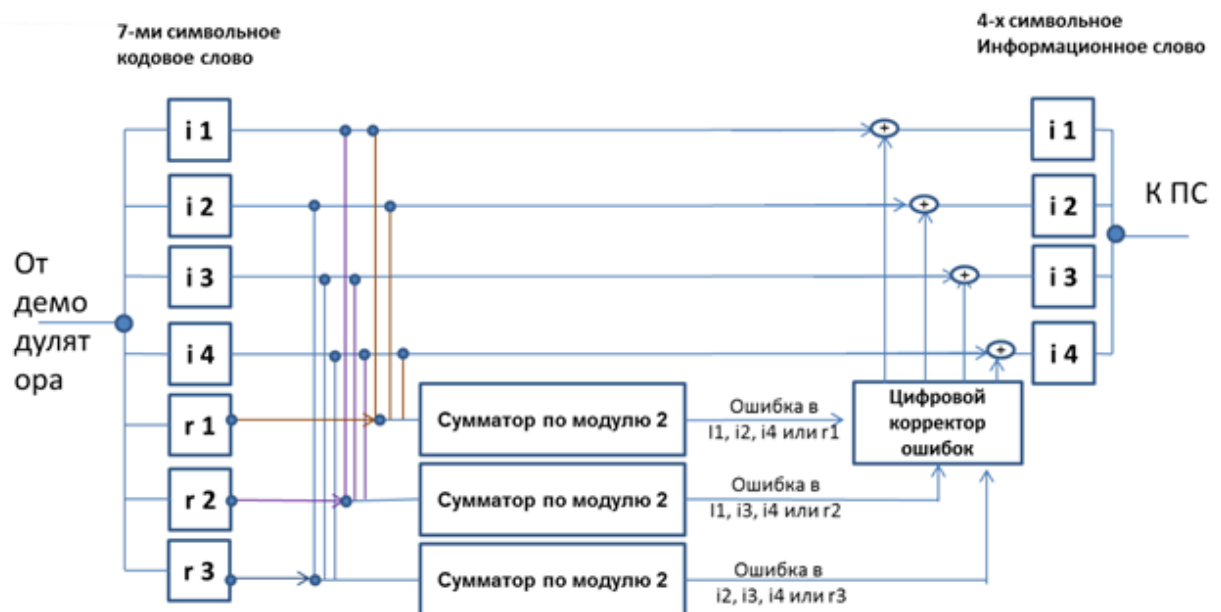


Рисунок 2.7 — схема декодирования кода Хэмминга (7,4)

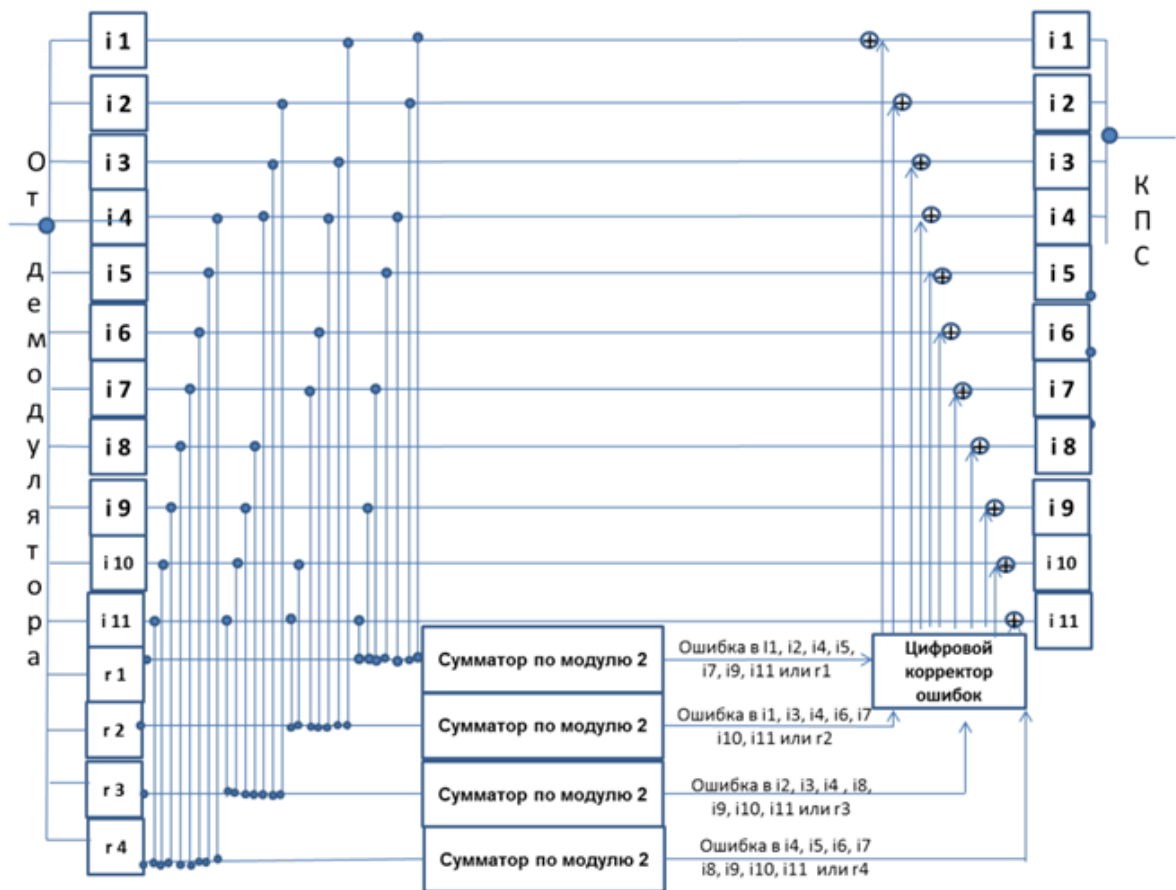


Рисунок 2.8 — схема декодирования кода Хэмминга (15,11)



### **3 Вывод**

В ходе выполнения данной лабораторной работы я познакомился с понятием помехоустойчивых кодов, наиболее подробно изучил код Хэмминга. То есть научился находить ошибки при передачи в подобных кодах, а также декодировать их. Подобные алгоритмы имеют множество применений, они значительно повышают надежность хранения информации, что и является их основным применением.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки - 1976 год [Электронный ресурс]. – URL:  
<https://clck.ru/32Edig>
2. Презентация «Код Хэмминга» Балакшин П.В – 2021-2022 год [Электронный ресурс]. – URL:  
<https://clck.ru/32EdYJ>