

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**

по дисциплине

«Информатика»

Вариант №75

Выполнил:

Студент группы Р3113

Султанов А.Р.

Проверил:

Белозубов А.В.

г. Санкт-Петербург

2022г.

## Оглавление

<b>Оглавление</b>	<b>2</b>
<b>Задание</b>	<b>3</b>
<b>Решение</b>	<b>6</b>
<b>Исходный код</b>	<b>14</b>
<b>Заключение</b>	<b>16</b>
<b>Литература</b>	<b>17</b>

## Задание

1. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
2. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
3. 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.
4. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
5. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
6. 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.
7. 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.
8. Необязательное задания для получения оценки «5» (позволяет набрать от 86 до 100 процентов от максимального числа баллов БаРС за данную лабораторную). 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.

Вариант	1				2
50	35	67	99	19	50
51	36	68	100	20	51
52	37	69	101	21	52
53	38	70	102	22	53
54	39	71	3	23	54
55	41	72	104	24	55
56	40	73	105	25	56
57	42	74	106	26	62
58	43	75	107	27	58
59	44	76	108	28	59
60	45	77	109	29	110
61	43	80	5	42	61
62	44	81	6	43	57
63	45	82	7	46	63
64	46	83	8	51	64
65	47	84	9	58	65
66	48	85	10	67	66
67	49	86	11	78	67
68	50	87	12	91	68
69	51	88	13	10	69
70	52	89	14	11	20
71	53	90	15	30	70
72	54	91	16	51	71
73	55	92	17	74	72
74	56	93	18	99	73
75	57	94	19	10	74

Рисунок 1. Номера вариантов заданий на основе варианта работы

1	57	0	0	0	0	1	0	0
---	----	---	---	---	---	---	---	---

Рисунок 2. Вариант сообщения №57

1	94	1	0	1	0	1	1	0
---	----	---	---	---	---	---	---	---

*Рисунок 3. Вариант сообщения №94*

1.	19	0	1	0	1	0	0	1
----	----	---	---	---	---	---	---	---

*Рисунок 4. Вариант сообщения №19*

1.	10	1	0	1	0	0	0	0
----	----	---	---	---	---	---	---	---

*Рисунок 5. Вариант сообщения №10*

2.	74	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0
----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

*Рисунок 6. Вариант сообщения №74*

## Решение

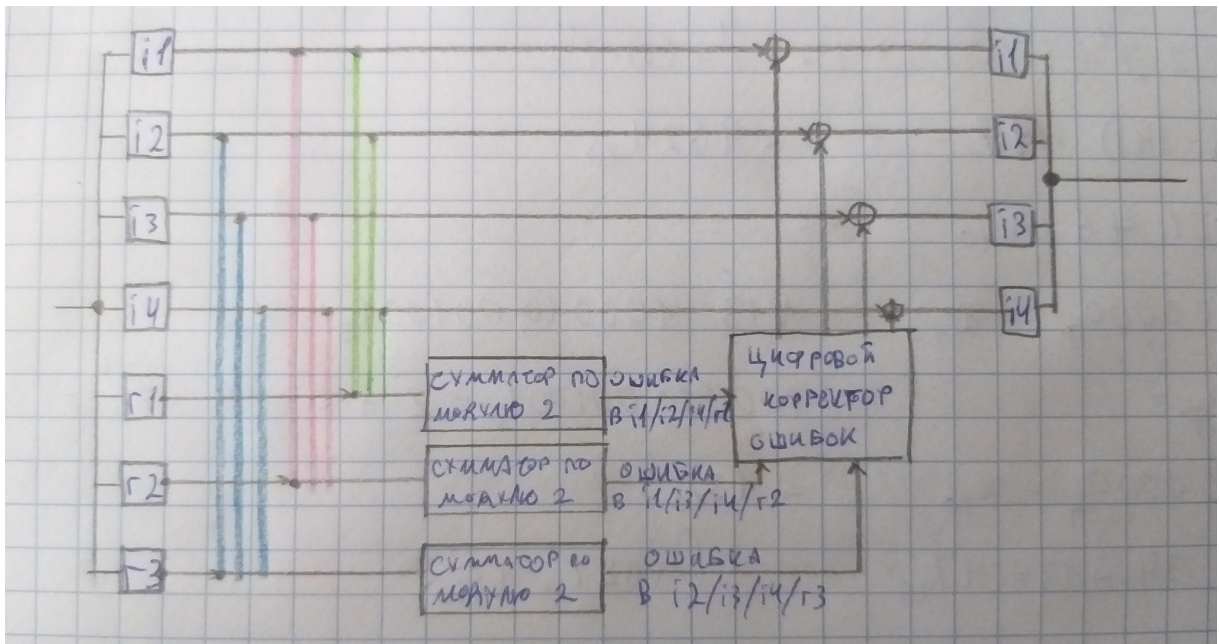


Рисунок 7. Схема декодирования классического кода Хэмминга (7;4)

По очереди рассмотрим все варианты сообщений (часть №1), каждое сообщение подставим в таблицу кода Хэмминга (7;4). Посчитаем контрольные суммы, на их основании найдем синдромы этих сообщений. Синдром (если он не равен 0) показывает номер бита, в котором допущена ошибка. После нахождения синдрома запишем исправленное сообщение (см. рисунки 8-11).

Вариант сообщения	57						
	1	2	3	4	5	6	7
Сообщение	0	0	0	0	1	0	0
2^x	r1	r2	i1	r3	i2	i3	i4
1	x		x		x		x
2		x	x			x	x
4				x	x	x	x
s1=r1⊕i1⊕i2⊕i4	0⊕0⊕1⊕0=1						
s2=r2⊕i1⊕i3⊕i4	0⊕0⊕0⊕0=0						
s3=r3⊕i2⊕i3⊕i4	0⊕1⊕0⊕0=1						
S = concat(s3, s2, s1)	101 (= 5 в десятичной)						
Ошибка в (бит с номером S)	i2						
Исправленное сообщение	0	0	0	0	0	0	0

Рисунок 8. Таблица кода Хэмминга для сообщения №57 и его исправление

Вариант сообщения	94						
	1	2	3	4	5	6	7
Сообщение	1	0	1	0	1	1	0
2^x	r1	r2	i1	r3	i2	i3	i4
1	x		x		x		x
2		x	x			x	x
4				x	x	x	x
s1=r1⊕i1⊕i2⊕i4	1⊕1⊕1⊕0=1						
s2=r2⊕i1⊕i3⊕i4	0⊕1⊕1⊕0=0						
s3=r3⊕i2⊕i3⊕i4	0⊕1⊕1⊕0=0						
S = concat(s3, s2, s1)	001 (= 1 в десятичной)						
Ошибка в (бит с номером S)	r1						
Исправленное сообщение	1	0	1	0	1	1	0

Рисунок 9. Таблица кода Хэмминга для сообщения №94 и его исправление



Вариант сообщения	19						
	1	2	3	4	5	6	7
Сообщение	0	1	0	1	0	0	1
2^x	r1	r2	i1	r3	i2	i3	i4
1	x		x		x		x
2		x	x			x	x
4				x	x	x	x
s1=r1⊕i1⊕i2⊕i4	0⊕0⊕0⊕1=1						
s2=r2⊕i1⊕i3⊕i4	1⊕0⊕0⊕1=0						
s3=r3⊕i2⊕i3⊕i4	1⊕0⊕0⊕1=0						
S = concat(s3, s2, s1)	001 (= 1 в десятичной)						
Ошибка в (бит с номером S)	r1						
Исправленное сообщение	1	1	0	1	0	0	1

Рисунок 10. Таблица кода Хэмминга для сообщения №19 и его исправление

Вариант сообщения	10						
	1	2	3	4	5	6	7
Сообщение	1	0	1	0	0	0	0
$2^x$	r1	r2	i1	r3	i2	i3	i4
1	x		x		x		x
2		x	x			x	x
4				x	x	x	x
$s1=r1\oplus i1\oplus i2\oplus i4$	$1\oplus 1\oplus 0\oplus 0=0$						
$s2=r2\oplus i1\oplus i3\oplus i4$	$0\oplus 1\oplus 0\oplus 0=1$						
$s3=r3\oplus i2\oplus i3\oplus i4$	$0\oplus 0\oplus 0\oplus 0=0$						
$S = \text{concat}(s3, s2, s1)$	010 (= 2 в десятичной)						
Ошибка в (бит с номером S)	r2						
Исправленное сообщение	1	1	1	0	0	0	0

Рисунок 11. Таблица кода Хэмминга для сообщения №10 и его исправление

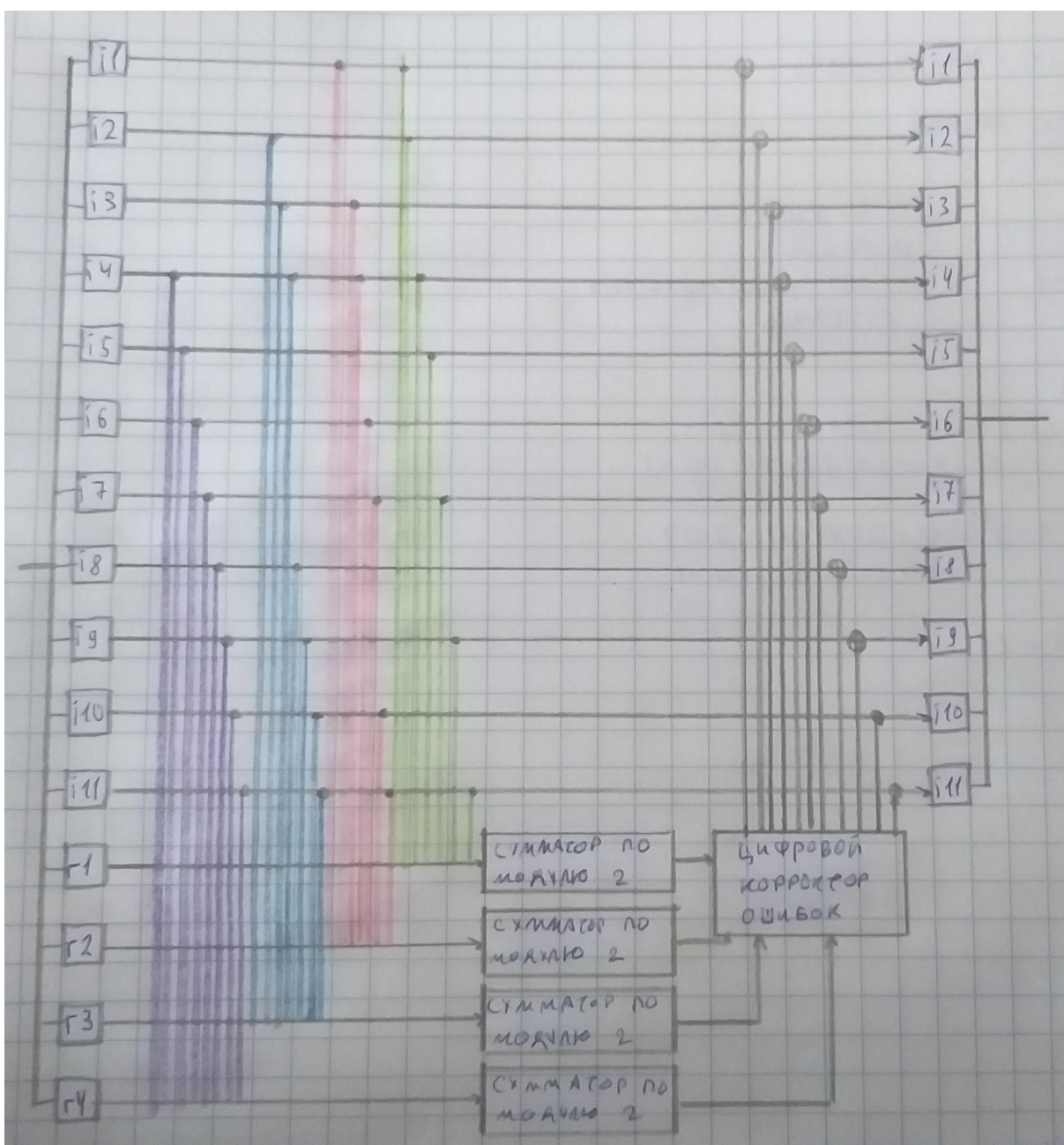


Рисунок 12. Схема декодирования классического кода Хэмминга (15;11)

Далее, необходимо проверить сообщение (№74, часть 2) на наличие ошибок, а после исправить их и записать верное сообщение. Занесем сообщение в таблицу кода Хэмминга (15;11), посчитаем контрольные суммы и на их основании найдем синдром, по которому определим наличие ошибки и номер ошибочного бита, если таковой имеется. В конце запишем исправленное сообщение (см. рисунок 13).

Вариант сообщения	74																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
Сообщение	0	0	1	1	1	0	0	1	0	1	0	0	1	0	0				
2^x	r1	r2	i1	r3	i2	i3	i4	r4	i5	i6	i7	i8	i9	i10	i11				
1	x		x		x		x		x		x		x		x				
2		x	x			x	x			x	x			x	x				
4				x	x	x	x					x	x	x	x				
8								x	x	x	x	x	x	x	x				
s1=r1⊕i1⊕i2⊕i4⊕i5⊕i7⊕i9⊕i11	0⊕1⊕1⊕0⊕0⊕0⊕0⊕1⊕0=1																		
s2=r2⊕i1⊕i3⊕i4⊕i6⊕i7⊕i10⊕i11	0⊕1⊕0⊕0⊕0⊕1⊕0⊕0⊕0=0																		
s3=r3⊕i2⊕i3⊕i4⊕i8⊕i9⊕i10⊕i11	1⊕1⊕0⊕0⊕0⊕0⊕1⊕0⊕0=1																		
s4=r4⊕i5⊕i6⊕i7⊕i8⊕i9⊕i10⊕i11	1⊕0⊕1⊕0⊕0⊕0⊕1⊕0⊕0=1																		
S = concat(s4, s3, s2, s1)	1101 (= 13 в десятичной)																		
Ошибка в (бит с номером S)	i9																		
Исправленное сообщение	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0				

Рисунок 13. Таблица кода Хэмминга для сообщения №74 и его исправление

Далее необходимо сложить номера всех 5 вариантов заданий, умножить на 4 и принять данное число как число информационных разрядов в передаваемом сообщении. Для вычисления минимального количества проверочных разрядов, необходимо воспользоваться следующей формулой:  $2^r \geq r + i + 1$ , где  $i$  - количества информационных разрядов, а  $r$  - количества проверочных разрядов. Нахождение минимального количества проверочных разрядов представлено на рисунке 14.

$$(57 + 94 + 19 + 10 + 74) \cdot 4 = 1016$$

КОМЕРА ВАРИАНТОВ ЗАДАНИЙ

$$2^{\Gamma} \geq \Gamma + i + 1 \quad \Gamma - \text{кол-во контрольных разрядов}$$

$$2^{\Gamma} \geq \Gamma + 1016 + 1 \quad i - \text{кол-во информационных разрядов}$$

$$2^{\Gamma} - \Gamma \geq 1017$$

(ПРОВЕРЯЕМ):

$\Gamma = 10: \quad 2^{\Gamma} - \Gamma = 2^{10} - 10 = 1024 - 10 = 1014$   
 $1014 < 1017$  НЕ ПОЯХОЖИТ.

$\Gamma = 11: \quad 2^{\Gamma} - \Gamma = 2^{11} - 11 = 2048 - 11 = 2037$   
 $2037 > 1017$  ПОЯХОЖИТ.

ОТВЕТ:  $\Gamma = 11$

Рисунок 14. Нахождение минимального количества проверочных разрядов для 1016 информационных разрядов

## Исходный код

```
MESSAGE_LENGTH = 7
BIT_NAMES = {
    1: 'r1',
    2: 'r2',
    3: 'i1',
    4: 'r3',
    5: 'i2',
    6: 'i3',
    7: 'i4',
}

def message_characters_are_valid(m: str) -> bool:
    return all(map(lambda x: x in '01', m))

def message_length_is_valid(m: str) -> bool:
    return len(m) == MESSAGE_LENGTH

def get_error_position(m: str) -> int:
    m_numbers = [int(c) for c in m]

    checksum1 = m_numbers[0] ^ m_numbers[2] ^ m_numbers[4] ^
m_numbers[6]
    checksum2 = m_numbers[1] ^ m_numbers[2] ^ m_numbers[5] ^
m_numbers[6]
    checksum3 = m_numbers[3] ^ m_numbers[4] ^ m_numbers[5] ^
m_numbers[6]

    error_syndrome = int(f'{checksum3}{checksum2}{checksum1}', 2)

    return error_syndrome

def get_information_bits(m: str) -> str:
    return f'{m[2]}{m[4]}{m[5]}{m[6]}'

def swap_message_bit(m: str, pos: int) -> str:
    if pos < 1 or pos > MESSAGE_LENGTH:
        return m
```

```

new_bit_value = '0' if m[pos - 1] == '1' else '1'
return m[:pos - 1] + new_bit_value + m[pos:]

def main() -> None:
    print('Enter the sequence of 7 bits (0/1s only):')
    message = input()

    if not message_characters_are_valid(message):
        print('Only 0s or 1s!')
        return

    if not message_length_is_valid(message):
        print('The message has to be of length 7!')
        return

    error_position = get_error_position(message)

    fixed_message = swap_message_bit(message, error_position)

    information_bits = get_information_bits(fixed_message)

    print('Information:')
    print(information_bits)

    if error_position == 0:
        print('No errors detected')
    else:
        bit_name = BIT_NAMES[error_position]
        print(f'Error in {bit_name} (position {error_position})')

if __name__ == '__main__':
    main()

```



## Заключение

В рамках этой работы было необходимо проверить сообщения на корректность с помощью кода Хэмминга, а также исправить эти ошибки. Данная лабораторная работа позволяет на практике ознакомиться с кодом Хэмминга, алгоритмом его декодирования и способом выявления ошибок.



## Литература

- Балакшин, Павел. *Информатика 2022-2023 Лекция №2*. 28 09 2022.  
*Telegram Web*, [https://t.me/balakshin\\_students/20](https://t.me/balakshin_students/20).

- Иванов, Артемий. “Код Хэмминга.” *YouTube*, 28 01 2020,  
<https://www.youtube.com/watch?v=ehuNcmE8S84>.