

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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа по информатике №2

Вариант 72

Выполнил: Снагин Станислав Максимович
Проверил: Белокон Юлия Алексеевна
Группа: Р3115

Санкт Петербург 2024 г.

Содержание

Задание.....	3
Основные этапы вычисления.....	4
Номер варианта.....	4
Набор сообщений.....	4
Схема декодирования кода Хэмминга (7, 4).....	5
Проверяем на наличие ошибок.....	5
Вариант 54.....	5
Вариант 91.....	6
Вариант 16.....	6
Вариант 51.....	7
Одиннадцатисимвольное сообщение.....	7
Схема декодирования кода Хэмминга (15, 11).....	8
Проверяем на наличие ошибок, 71 вариант.....	8
Вариант 848*.....	9
Дополнительное задание.....	10
Вывод.....	12

Задание

1. Определить свой вариант задания с помощью номера в ISU (он же номер студенческого билета). Вариантом является комбинация 3-й и 5-й цифр. Т.е. если номер в ISU = 123456, то вариант = 35.
2. На основании номера варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода.
3. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения.
4. 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.
5. На основании номера варианта задания выбрать 1 полученное сообщение в виде последовательности 11-символьного кода.
6. Построить схему декодирования классического кода Хэмминга (15;11), которую представить в отчёте в виде изображения.
7. 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.
8. 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.
9. Дополнительное задание №1 (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.

Основные этапы вычисления

Номер варианта

Мой номер ИСУ 467525. Берем 3 и 5 цифры от этого числа, и получаем номер варианта: **72**.

Набор сообщений

Берём значения согласно варианту **72**: 54, 91, 16, 51.

ALT	1	2	3	4	5	6	7
	r1	r2	i1	r3	i2	i3	i4
54	1	1	0	1	0	1	1
91	0	1	1	1	1	1	0
16	0	0	0	1	0	0	1
51	1	0	1	0	0	1	1

Схема декодирования кода Хэмминга (7, 4)

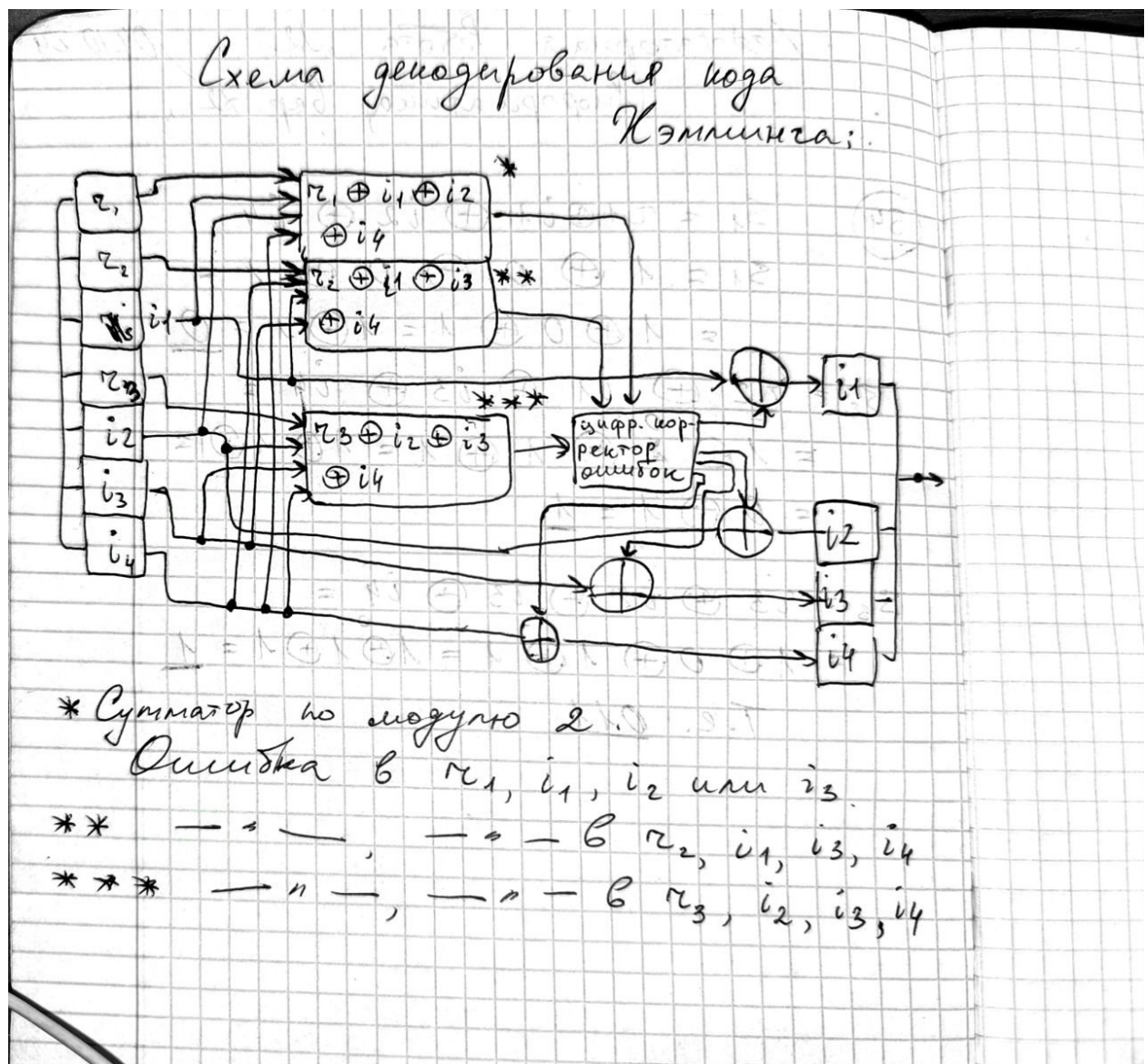


Рисунок 1: Схема декодирования кода Хэмминга (7, 4)

Проверяем на наличие ошибок

Далее считаем синдром от s_1, s_2, s_3 , делаем таблицы и находим различия (если таковые имеются):

Вариант 54

$$s_1 = r_1 \underline{\vee} i_1 \underline{\vee} i_2 \underline{\vee} i_4 = 1 \underline{\vee} 0 \underline{\vee} 0 \underline{\vee} 1 = 0$$

$$s_2 = r_2 \underline{\vee} i_1 \underline{\vee} i_3 \underline{\vee} i_4 = 1 \underline{\vee} 0 \underline{\vee} 1 \underline{\vee} 1 = 1$$

$$s_3 = r_3 \underline{\vee} i_2 \underline{\vee} i_3 \underline{\vee} i_4 = 1 \underline{\vee} 0 \underline{\vee} 1 \underline{\vee} 1 = 1$$

Синдром : 011

	r1	r2	i1	r3	i2	i3	i4	
S1						#		0
S2						#		1
S3						#		1

Сопоставив маску со значением, я увидел ошибку в i3, поэтому инвертируем бит в этой клетке.

Итого правильное сообщение целиком:

1	1	0	1	0	0	1
---	---	---	---	---	---	---

Вариант 91

$$s1 = r1 \vee i1 \vee i2 \vee i4 = 0 \vee 1 \vee 1 \vee 0 = 0$$

$$s2 = r2 \vee i1 \vee i3 \vee i4 = 1 \vee 1 \vee 1 \vee 0 = 1$$

$$s3 = r3 \vee i2 \vee i3 \vee i4 = 1 \vee 1 \vee 1 \vee 0 = 1$$

Синдром: 011

	r1	r2	i1	r3	i2	i3	i4	
S1						#		0
S2						#		1
S3						#		1

Сопоставив маску со значением, я увидел ошибку в i3, поэтому инвертируем бит в этой клетке.

Итого правильное сообщение целиком:

0	1	1	1	1	0	0
---	---	---	---	---	---	---

Вариант 16

$$s1 = r1 \vee i1 \vee i2 \vee i4 = 0 \vee 0 \vee 0 \vee 1 = 1$$

$$s2 = r2 \vee i1 \vee i3 \vee i4 = 0 \vee 0 \vee 0 \vee 1 = 1$$

$$s3 = r3 \vee i2 \vee i3 \vee i4 = 1 \vee 0 \vee 0 \vee 1 = 0$$

Синдром: 110

	r1	r2	i1	r3	i2	i3	i4	
S1			#					1
S2			#					1
S3			#					0

Сопоставив маску со значением, я увидел ошибку в i1, поэтому инвертируем бит в этой клетке.

Итого правильное сообщение целиком:

0	0	1	1	0	0	1
---	---	---	---	---	---	---

Вариант 51

$$s1 = r1 \vee i1 \vee i2 \vee i4 = 1 \vee 1 \vee 0 \vee 1 = 1$$

$$s2 = r2 \vee i1 \vee i3 \vee i4 = 0 \vee 1 \vee 1 \vee 1 = 1$$

$$s3 = r3 \vee i2 \vee i3 \vee i4 = 0 \vee 0 \vee 1 \vee 1 = 0$$

Синдром: 110

	r1	r2	i1	r3	i2	i3	i4	
S1			#					1
S2			#					1
S3			#					0

Сопоставив маску со значением, я увидел ошибку в i1, поэтому инвертируем бит в этой клетке.

Итого правильное сообщение целиком:

1	0	0	0	0	1	1
---	---	---	---	---	---	---

Одиннадцатисимвольное сообщение

ALT	r1	r2	i1	r3	i2	i3	i4	r4	i5	i6	i7	i8	i9	i10	i11
71	0	0	1	1	1	0	0	0	1	1	1	0	1	0	0

Схема декодирования кода Хэмминга (15, 11)

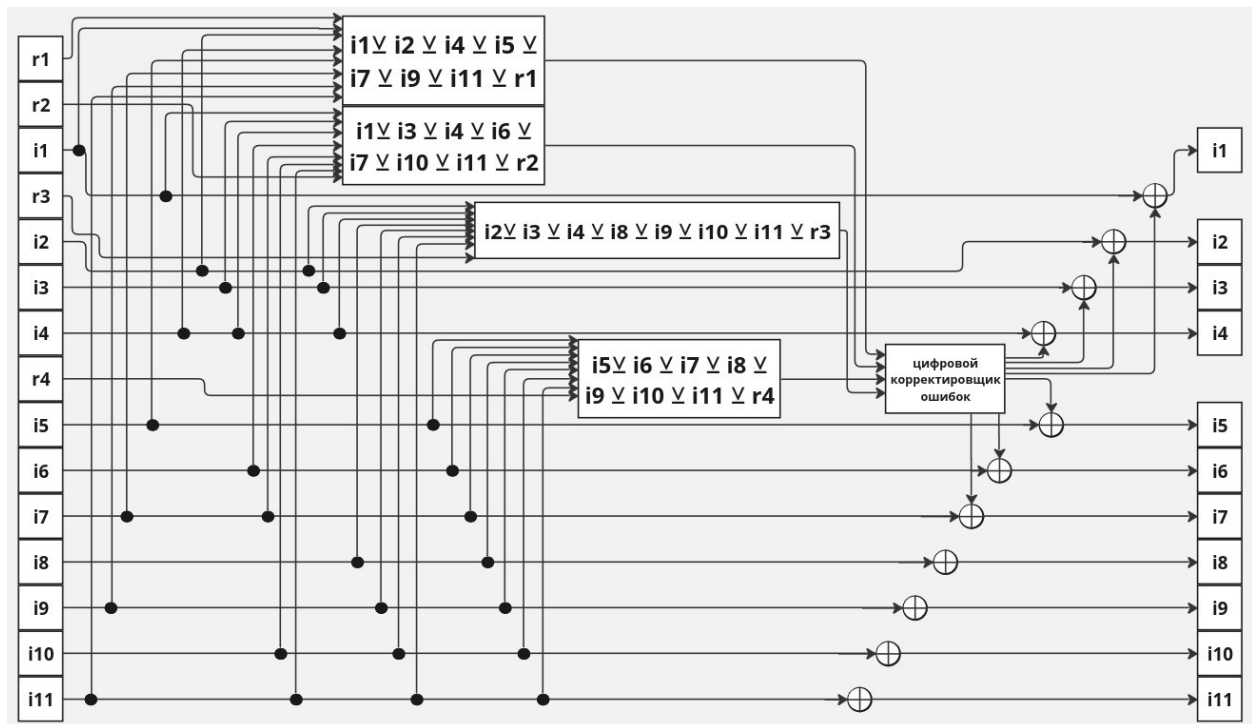


Рисунок 2: Схема декодирования кода Хэмминга (15, 11). Выполнена в Miro.

Проверяем на наличие ошибок, 71 вариант

Схема похожа на ту, что использовалась для кода (7, 4), но теперь синдром состоит из 4 бит:

$$s1 = i1 \vee i2 \vee i4 \vee i5 \vee i7 \vee i9 \vee i11 \vee r1 = 1 \vee 1 \vee 0 \vee 1 \vee 1 \vee 1 \vee 0 \vee 0 = 1$$

$$s2 = i1 \vee i3 \vee i4 \vee i6 \vee i7 \vee i10 \vee i11 \vee r2 = 1 \vee 0 \vee 0 \vee 1 \vee 1 \vee 0 \vee 0 \vee 0 = 1$$

$$s3 = i2 \vee i3 \vee i4 \vee i8 \vee i9 \vee i10 \vee i11 \vee r3 = 1 \vee 0 \vee 0 \vee 0 \vee 1 \vee 0 \vee 0 \vee 1 = 1$$

$$s4 = i5 \vee i6 \vee i7 \vee i8 \vee i9 \vee i10 \vee i11 \vee r4 = 1 \vee 1 \vee 1 \vee 0 \vee 1 \vee 0 \vee 0 \vee 0 = 0$$

Синдром: 1110

	r1	r2	i1	r3	i2	i3	i4	r4	i5	i6	i7	i8	i9	i10	i11	
S1							#									1
S2							#									1
S3							#									1
S4							#									0

Сопоставив маску со значением, я увидел ошибку в i4, поэтому инвертируем бит в этой клетке.

Итого правильное сообщение целиком:

	r1	r2	i1	r3	i2	i3	i4	r4	i5	i6	i7	i8	i9	i10	i11
71	0	0	1	1	1	0	1	0	1	1	1	0	1	0	0

Вариант 848*

$*(54 + 91 + 16 + 51) * 4 = 848$ — число информационных разрядов в передаваемом сообщении.

Воспользуемся неравенством минимального числа контрольных разрядов: $2^r \geq r + i + 1$, где r — минимальное число контрольных разрядов, i — диапазон информационных разрядов, а так же формулой коэффициента избыточности: $n = r / (i + r)$.

$i = 848$; $2^r \geq r + 848 + 1$
 $2^r - r - 1 \geq 848$

Нам нужно подобрать такое r , чтобы для нашего i (диапазона) оно удовлетворяло заданным ограничениям (верхним и нижним):

$\begin{cases} 848 > 2^{r-1} - (r-1) - 1 \\ 848 < 2^r - r - 1 \end{cases}$

$\begin{cases} 848 > 2^{r-1} - r + 1 \\ 848 < 2^r - r - 1 \end{cases}$

Методом перебора (подбора) получаем, что $r = 10$ удовл. нашей системе.

$2^{10} - 10 - 1 \geq 848 > 2^9 - 10$
 $1013 \geq 848 > 502$

$r = 10$:
 $2^9 = 512$ (нижн. гр.)
 $512 - 10 = 502$ (нижн. гр.)
 $2^{10} = 1024$ (верх. гр.)
 $1024 - 10 - 1 = 1013$ (верх. гр.)

$r = 11$:
 $2^{10} - 11 \neq 848$

$r = 9$:
 $2^8 - 9 = 256 - 9 = 247$
 $2^9 - 10 = 502 \neq 848$

Рисунок 3: Поиск значения r ($r = 10$)

$$\text{коэф. избыточности:} \\ h = \frac{r}{i+r} = \frac{10}{848+10} \approx 0,11655012...$$

Рисунок 4: Значение коэф. избыточности

Ответ: $r = 10$, $n = 0.11655012...$

Дополнительное задание

Код программы (доступен на github --

https://raw.githubusercontent.com/ssnagin/ITMO_SPPO/refs/heads/main/ICT/1sem/lab2/hamming_code.py):

```
ALLOWED_DIGITS = [{"0", "1"}, {"1", "0"}, {"1", "1"}, {"0", "0"}]
NAMES = ["r1", "r2", "i1", "r3", "i2", "i3", "i4"]
```

```
def intro() -> str:
    return ("ssngn | Hamming code ver. 1.0\nINFO | Запись делать след. образом: r1 r2\ni1 r3 i2 i3 i4")
```

```
def regular_check(binary_number: str) -> bool:
    global ALLOWED_DIGITS
    if len(binary_number) != 7: return False

    for element in ALLOWED_DIGITS:
        if set(binary_number) == element:
            return True

    return False
```

```
def parse_bits(number: str) -> []:
    """
    output: [r1, r2, i1, r3, i2, i3, i4]
    """
    return [int(x) for x in number]
```

```
def parse_payload_bits(bits: []) -> []:
    """
    output: [i1, i2, i3, i4]
    """
    parsed = parse_bits(number)
    return [bits[2], bits[4], bits[5], bits[6]]
```

```
def hamming_7_4(number: str) -> str:
    global NAMES

    # Our response to the console:
    response = " ... | \n"
```

```

# Count syndromes:
bits = parse_bits(number)

syndromes = [
    bits[0] ^ bits[2] ^ bits[4] ^ bits[6],
    bits[1] ^ bits[2] ^ bits[5] ^ bits[6],
    bits[3] ^ bits[4] ^ bits[5] ^ bits[6],
]

if syndromes == [0, 0, 0]:
    response = "      | " + str(bits[2]) + str(bits[4]) + str(bits[5]) +
str(bits[6])
    return response

# wrong_bit_place = int(str(syndromes[0]) + str(syndromes[1]) + str(syndromes[2]),
2)
wrong_bit_place = (syndromes[2] << 2) | (syndromes[1] << 1) | syndromes[0]
response += "ERROR | Wrong bit place: " + str(wrong_bit_place) + f"
({NAMES[wrong_bit_place - 1]})\n" + \
"WAS      | Replaced with " + str(int(not bits[wrong_bit_place - 1])) +
"\n"

bits[wrong_bit_place - 1] = int(not bits[wrong_bit_place - 1])

payload_bits = parse_payload_bits(bits)

response += "FIXED | Payload is " + "".join(str(x) for x in payload_bits)

return response

print(intro())

while True:
    number = input("Input | Binary number (7 bits): ")

    if not regular_check(number):
        print("Error | check if the number was written correctly.")
        continue

    print(hamming_7_4(number))
    break

```

Вывод

Благодаря этой лабораторной работе я узнал о коде Хэмминга, понял принцип работы и смог даже написать программу по этому алгоритму. Я уверен, что эти знания помогут мне в дальнейшей профессиональной деятельности.