

Содержание

Содержание	1
Задание.....	2
Схемы декодирования кода Хэмминга.....	3
Основные этапы вычисления	4
Задание 1.1	5
Задание 1.2	6
Задание 1.3	7
Задание 1.4.....	8
Задание 2	9
Задание 3	10
Дополнительное задание.....	10
Заключение.....	11

Задание

Часть 1. На основании варианта задания выбрать набор из 4 полученных сообщений в виде последовательности 7-символьного кода. Построить схему декодирования классического кода Хэмминга (7;4), которую представить в отчёте в виде изображения. Показать, исходя из выбранных вариантов сообщений, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

Часть 2. На основании номера варианта задания выбрать 1 полученное сообщение в виде 15-символьного кода. Построить схему декодирования классического кода Хэмминга (15; 11), которую представить в отчёте в виде изображения. Показать, исходя из выбранного варианта сообщений, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

Часть 3. Сложить номера всех 5 вариантов заданий. Умножить полученное число на 4. Получить данное число как число разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.

Дополнительное задание. Написать программу на любом языке программирования, которая на вход получает набор из 7 цифр от “0” до “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.

Схемы декодирования кода Хэмминга

Ниже представлены схемы декодирования кода Хэмминга (7,4) (Рисунок 1) и кода Хэмминга (15,11) (Рисунок 2).

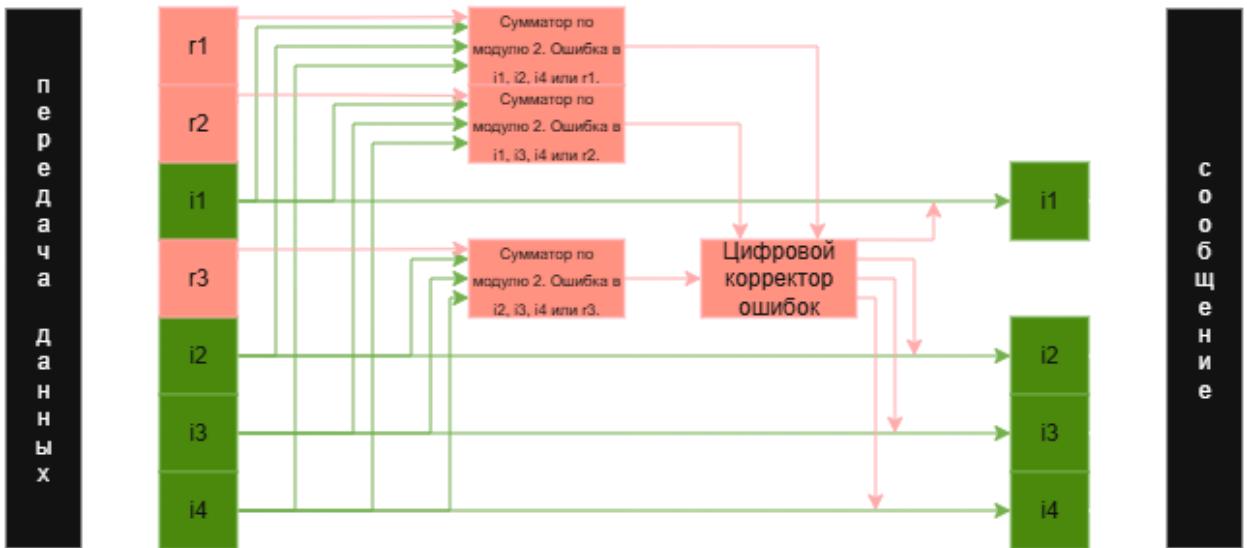


Рисунок 1

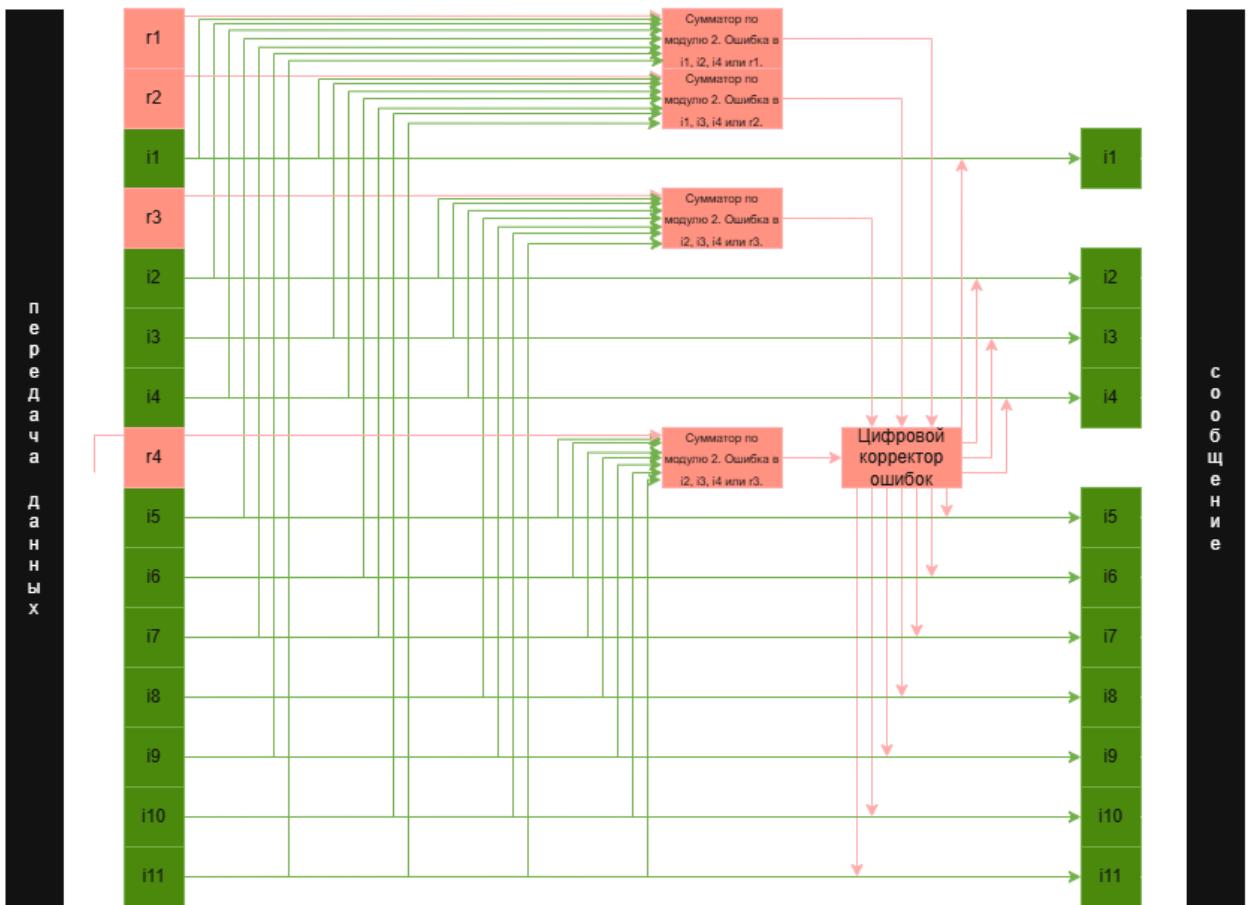


Рисунок 2

Основные этапы вычисления

Задание 1.1

Необходимо построить схему декодирования кода Хэмминга (7;4) для полученного сообщения и найти ошибки при их наличии. Для построения данной схемы нам нужно вычислить синдромы для каждого проверочного бита, а после на основании полученных синдромов найти ошибочный бит.

Тогда вычислим синдромы:

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

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

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

Тогда в соответствии с полученными синдромами заполним таблицу по следующему правилу: если синдром равен 1, то ошибка в битах, за которые этот синдром отвечает; если же синдром равен 0, то ошибка в битах, за которые этот синдром не отвечает.

Получаем схему:

N	1	2	3	4	5	6	7
Биты	r1	r2	i1	r3	i2	i3	i4
Полученное сообщение	0	0	0	0	1	0	0
s1	X		X		X		X
s2		X	X			X	X
s3				X	X	X	X

Тогда из схемы видно, что ошибка в бите 5. Заменим повреждённый бит на ему обратный, получив исправленное сообщение “0000000”. Убрав проверочные биты, получим сообщение “0000”.

Задание 1.2

Необходимо построить схему декодирования кода Хэмминга (7;4) для полученного сообщения и найти ошибки при их наличии. Для построения данной схемы нам нужно вычислить синдромы для каждого проверочного бита, а после на основании полученных синдромов найти ошибочный бит.

Тогда вычислим синдромы:

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

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

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

Тогда в соответствии с полученными синдромами заполним таблицу по следующему правилу: если синдром равен 1, то ошибка в битах, за которые этот синдром отвечает; если же синдром равен 0, то ошибка в битах, за которые этот синдром не отвечает.

Получаем схему:

N	1	2	3	4	5	6	7
Биты	r1	r2	i1	r3	i2	i3	i4
Полученное сообщение	0	1	1	1	0	1	1
s1	X		X		X		X
s2		X	X			X	X
s3				X	X	X	X

Тогда из схемы видно, что ошибка в бите 4. Заменим повреждённый бит на ему обратный, получив исправленное сообщение “0110011”. Убрав проверочные биты, получим сообщение “1011”.

Задание 1.3

Необходимо построить схему декодирования кода Хэмминга (7;4) для полученного сообщения и найти ошибки при их наличии. Для построения данной схемы нам нужно

вычислить синдромы для каждого проверочного бита, а после на основании полученных синдромов найти ошибочный бит.

Тогда вычислим синдромы:

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

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

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

Тогда в соответствии с полученными синдромами заполним таблицу по следующему правилу: если синдром равен 1, то ошибка в битах, за которые этот синдром отвечает; если же синдром равен 0, то ошибка в битах, за которые этот синдром не отвечает.

Получаем схему:

N	1	2	3	4	5	6	7
Биты	r1	r2	i1	r3	i2	i3	i4
Полученное сообщение	0	0	0	0	1	0	1
s1	X		X		X		X
s2		X	X			X	X
s3				X	X	X	X

Тогда из схемы видно, что ошибка в бите 2. Заменим повреждённый бит на ему обратный, получив исправленное сообщение “0100101”. Убрав проверочные биты, получим сообщение “0101”.

Задание 1.4

Необходимо построить схему декодирования кода Хэмминга (7;4) для полученного сообщения и найти ошибки при их наличии. Для построения данной схемы нам нужно вычислить синдромы для каждого проверочного бита, а после на основании полученных синдромов найти ошибочный бит.

Тогда вычислим синдромы:

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

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

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

Тогда в соответствии с полученными синдромами заполним таблицу по следующему правилу: если синдром равен 1, то ошибка в битах, за которые этот синдром отвечает; если же синдром равен 0, то ошибка в битах, за которые этот синдром не отвечает.

Получаем схему:

N	1	2	3	4	5	6	7
Биты	r1	r2	i1	r3	i2	i3	i4
Полученное сообщение	1	0	0	1	1	1	0
s1	X		X		X		X
s2		X	X			X	X
s3				X	X	X	X

Тогда из схемы видно, что ошибка в бите 6. Заменим повреждённый бит на ему обратный, получив исправленное сообщение “1001110”. Убрав проверочные биты, получим сообщение “0100”.

Задание 2

Необходимо построить схему декодирования кода Хэмминга (15;11) для полученного сообщения и найти ошибки при их наличии. Для построения данной схемы нам нужно

вычислить синдромы для каждого проверочного бита, а после на основании полученных синдромов найти ошибочный бит.

Тогда вычислим синдромы:

$$S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9 \oplus i_{11} = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$S_2 = r_2 \oplus i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus i_{10} \oplus i_{11} = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$S_3 = r_3 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 1 \oplus 0 = 1$$

$$S_4 = r_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus i_{10} \oplus i_{11} = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$$

Тогда в соответствии с полученными синдромами заполним таблицу по следующему правилу: если синдром равен 1, то ошибка в битах, за которые этот синдром отвечает; если же синдром равен 0, то ошибка в битах, за которые этот синдром не отвечает.

Получаем схему:

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Биты	r1	r2	i1	r3	i2	i3	i4	r4	i5	i6	i7	i8	i9	i10	i11
Полученное сообщение	0	1	1	1	0	0	0	1	1	1	1	0	0	0	0
s1	X		X		X		X		X		X		X		X
s2		X	X			X	X			X	X			X	X
s3				X	X	X	X					X	X	X	X
s4								X	X	X	X	X	X	X	X

Тогда из схемы видно, что ошибка в бите 5. Заменим повреждённый бит на ему обратный, получив исправленное сообщение “011110011110000”. Убрав проверочные биты, получим сообщение “11001110000”.

Задание 3

Необходимо сложить номера всех вариантов 5 выполненных ранее заданий ($57 + 49 + 71 + 93 + 15 = 285$), умножить полученное число на 4 ($285 * 4 = 1140$), и принять данное число как число информационных разрядов в передаваемом сообщении. В данном случае 1140 разрядов.

Для того, чтобы найти минимальное число проверочных разрядов для данного сообщения, необходимо найти для количества разрядов ближайшую минимальную степень двойки, которая будет больше числа разрядов, тогда эта степень и будет количеством проверочных разрядов. В данном случае получаем $1140 < 2048 = 2^{11}$, тогда число проверочных разрядов равно 11.

Для того, чтобы найти коэффициент избыточности, необходимо поделить количество проверочных разрядов на количество общего числа разрядов. В данном случае коэффициент равно $\frac{11}{11 + 1140} = \frac{11}{1151} \approx 0,009556907037358818 \dots$

Дополнительное задание.

Необходимо написать программу на любом языке программирования, которая, получая на вход код Хэмминга (7;4) с 1 или меньше ошибками, будет выводить информационные биты получаемого сообщения, при необходимости исправляя ошибки.

Код программы представлен ниже:

```
import sys

print("Введите набор цифр в формате 'rririii' (где r-проверочные биты; i-информационные")
```

```

биты):", end=" ")
while True:
    try:
        s = input().replace(" ", "")
        if len(s) != 7:
            print("Число не соответствует заданному формату")
            print("Пожалуйста, повторите ввод:", end=" ")
            continue
        correct = True
        for i in s:
            if i not in "01":
                print("Число не соответствует заданному формату")
                print("Пожалуйста, повторите ввод:", end=" ")
                correct = False
                break
        if correct: break
    except EOFError:
        print("Завершение работы.")
        sys.exit()
S = str()
S += str((int(s[0]) + int(s[2]) + int(s[4]) + int(s[6]))%2)
S += str((int(s[1]) + int(s[2]) + int(s[5]) + int(s[6]))%2)
S += str((int(s[3]) + int(s[4]) + int(s[5]) + int(s[6]))%2)
err_bit = int()
if S.count("1") == 0: err_bit = 7
elif S.count("1") == 1: err_bit = 2**int(S.index("1")) - 1
elif S.count("1") == 2: err_bit = int(S[::-1], 2)-1
else: err_bit = 6
if err_bit == 7: print("Передаваемое число - " + s[2] + s[4] + s[5] + s[6] + "! Ошибок нет!")
else:
    s = s[:err_bit] + str((int(s[err_bit]) + 1)%2) + s[err_bit+1:]
    print("Передаваемое число - " + s[2] + s[4] + s[5] + s[6] + "! Ошибка была в бите " +
str(err_bit + 1) + "!")

```

Заключение

В ходе данной лабораторной работы были изучены и освоены классический код Хэмминга, информационные и проверочные биты в нём, биты чётности и коэффициент избыточности.