НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Отчет по Лабораторной работе №2    
по курсу “Информатика”

“Помехоустойчивое кодирование. Код Хэмминга”  
   
Вариант №86

Выполнил:

Студент группы P3118

Шульга Артём Игоревич

Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург

2021

Оглавление

[Задания для Варианта 86 2](#_Toc83853766)

[Выполнение вычислений 3](#_Toc83853814)

[Вывод 9](#_Toc83853816)

[Список литературы 9](#_Toc83853817)

# 

# Задания для Варианта 86

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | 1 | | | | 2 |
| 86 | 89 | 11 | 42 | 24 | 85 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Задание | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 89 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 42 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 24 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Задание | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 | r4 | i5 | i6 | i7 | i8 | i9 | i10 | i11 |
| 85 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

# Выполнение вычислений

# Задание 89

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 | s |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| X |  | X |  | X |  | X | s1 |
|  | X | X |  |  | X | X | s2 |
|  |  |  | X | X | X | X | s3 |

Рассчитаем s1, s2, s3 по информационным и битам чётности:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 0 ⊕ 1 ⊕ 0 = 1 ⊕ 0 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 0 = 1 ⊕ 1 ⊕ 0 = 0 ⊕ 0 = 0

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 0 ⊕ 1 ⊕ 0 = 1 ⊕ 0 = 1

Из данных вычислений следует, что синдром последовательности равняется 101. Значит ошибочный бит – i2, так как за него отвечают контрольные суммы s1 и s3, которые равны 1.

**Задание 11**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 | s |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| X |  | X |  | X |  | X | s1 |
|  | X | X |  |  | X | X | s2 |
|  |  |  | X | X | X | X | s3 |

Рассчитаем контрольные суммы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0 ⊕ 0 ⊕ 0 = 0 ⊕ 0 = 0

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1 ⊕ 0 ⊕ 0 = 1 ⊕ 0 = 1

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1 ⊕ 0 ⊕ 0 = 1 ⊕ 0 = 1

Из данных вычислений следует, что синдром последовательности равен 011. Значит ошибочный бит – i3, так как контрольные суммы, которые за него отвечают, равны 1.

**Задание 42**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | s |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| X |  | X |  | X |  | X | s1 |
|  | X | X |  |  | X | X | s2 |
|  |  |  | X | X | X | X | s3 |

Рассчитаем контрольные суммы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0 ⊕ 0 ⊕ 0 = 0 ⊕ 0 = 0

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 0 ⊕ 1 ⊕ 0 = 1 ⊕ 0 = 1

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 0 = 1 ⊕ 1 ⊕ 0 = 0 ⊕ 0 = 0

Из данных вычислений, что синдром последовательности s = 010, значит ошибочный бит находится под номером 0102 (синдром инвертирован) = 2, то есть бит r2 является ошибочным.

**Задание 24**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | s |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| X |  | X |  | X |  | X | s1 |
|  | X | X |  |  | X | X | s2 |
|  |  |  | X | X | X | X | s3 |

Рассчитаем контрольные суммы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 0 ⊕ 0 ⊕ 1 = 0 ⊕ 1 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 0 ⊕ 1 ⊕ 0 ⊕ 1 = 1 ⊕ 0 ⊕ 1 = 1 ⊕ 1 = 0

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 0 ⊕ 0 ⊕ 1 = 0 ⊕ 1 = 1

Из данных вычислений следует, что синдром последовательности равен s = 101. Следовательно, ошибочный бит находится под номером 1012 (инвертированный синдром последовательности) = 5, то есть бит i2 является ошибочным.

**Задание 85**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | s |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 | r4 | i6 | i7 | i8 | i9 | i10 | i11 | i12 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| X |  | X |  | X |  | X |  | X |  | X |  | X |  | X | 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 |

Рассчитаем контрольные суммы:

s1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 ⊕ i6 ⊕ i8 ⊕ i10 ⊕ i12 = 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕1 ⊕ 1 ⊕ 1 ⊕ = 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 = 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 = 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 1 = 0 ⊕ 1 ⊕ 1 ⊕ 1 = 1 ⊕ 1 ⊕ 1 = 0 ⊕ 1 = 1

s2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 ⊕ i7 ⊕ i8 ⊕ i11 ⊕ i12 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1; Так как единиц чётное количество, то s2 = 0.

s3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i9 ⊕ i10 ⊕ i11 ⊕ i12 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1; Так как количество единиц чётно, то s3 = 0.

s4 = r4 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 ⊕ i12 = 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 1; Так как количество единиц чётно, то s4 = 0.

Синдром последовательности равняется s = 1000. Значит ошибочный бит находится под номером 00012 (инвертированный синдром) = 1, то есть бит r1, так как за него отвечает только контрольная сумма s1.

Сумма вариантов: (89+11+42+24+84)\*4 = 250 \* 4 = 1000 информационных битов. 2r >= r+i+1, 2r >= r+1001, 210 >= 10+1001 => 1024 >= 1011. Значит r = 10.

Коэффициент избыточности: k = r/(i+r) = 10/(10+1000) = 10/1010 = 0,0099009901.

**Дополнительное задание**

Исходный код:

#include <iostream>

#include <cmath>

using namespace std;

void print\_message(const int\* *a*,const int *N*)

{

    // печать ТОЛЬКО информационных битов, то есть не степени двойки

    for(int i=2;i<*N*;i++)

        if(i != 7 && i != 3)

            cout << *a*[i];

}

int main()

{

    const int N = 7;

    const int COUNT\_S = 3;

    string str;

    cin >> str;

    int a[N];

    // превращение строки в цифры кода

    for(int i=0;i<N;i++)

        a[i] = str[i] - '0';

    int s[COUNT\_S];

    for(int i=0;i<COUNT\_S;i++)

    {

        s[i] = 0;

        // начинаем от степени двойки и берём элементы в количестве этой степени

        // и делаем ещё один отступ от них в количестве степени двойки

        for(int j=(int)pow(2,i)-1;j<N;j+=(int)pow(2,i)\*2)

        {

            for(int z=0;z<(int)pow(2,i);z++)

                // делаем xor с выбранными элементами

                s[i] ^= a[j+z];

        }

    }

    // если синдром последовательности равен 0

    if(s[0] == s[1] && s[1] == s[2] && s[0] == 0)

    {

        cout << "MESSAGE IS CORRECT.\nMESSAGE: ";

        print\_message(a,N);

    }

    else

    {

        int incorrect\_i=0;

        // считаем некорректный бит путём инверсии синдрома последовательности

        for(int i=COUNT\_S-1;i>=0;i--)

            incorrect\_i += s[i]\*pow(2,i);

        // отнимаем 1, так как первый бит - 1

        incorrect\_i--;

        // инверсия некорректного бита

        if(a[incorrect\_i] == 0)

            a[incorrect\_i] = 1;

        else

            a[incorrect\_i] = 0;

        cout << "INCORRECT ";

        switch(incorrect\_i)

        {

            // по номеру бита определяем его название

            case 0:

                cout << "R1";

                break;

            case 1:

                cout << "R2";

                break;

            case 2:

                cout << "I1";

                break;

            case 3:

                cout << "R3";

                break;

            case 4:

                cout << "I2";

                break;

            case 5:

                cout << "I3";

                break;

            case 6:

                cout << "I4";

                break;

            default:

                break;

        }

        cout << endl;

        cout << "MESSAGE WAS INCORRECT.\nMESSAGE: ";

        print\_message(a,N);

    }

}

# Вывод

В ходе выполнения лабораторной работы я изучил классический код Хэмминга и научился находить ошибки при передаче данных при помощи данного кода.

# Список литературы

**“Код Хэмминга. Пример работы алгоритма”.** – Текст электронный // https://habr.com – URL <https://habr.com/ru/post/140611/>

**“Избыточное кодирование, код Хэмминга”.** – Текст электронный // neerc.ifmo.ru – URL https://bit.ly/3kSFXnN