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**

Синтез помехоустойчивого кода

Вариант №66

Группа: P3131

Выполнил: Хайкин О. И.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2021г

Оглавление

[Задание 3](#_Toc84925873)

[Основные этапы вычисления 4](#_Toc84925874)

[Схема декодирования классического кода Хэмминга (7;4): 4](#_Toc84925875)

[Задание №1 4](#_Toc84925876)

[Схема декодирования классического кода Хэмминга (15;11): 5](#_Toc84925877)

[Задание №2 6](#_Toc84925878)

[Задание №3 6](#_Toc84925879)

[Задание №4 6](#_Toc84925880)

[Заключение 8](#_Toc84925881)

# Задание

Порядок выполнения работы:

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 пердставить в отчёте в виде изображения.
3. Показать, исходя из выбранных вариантов сообщений, имеются ли в принятом сообщении ошибки, и если имеются, то какие.
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, имеются ли в принятом сообщении ошибки, и если имеются, то какие.
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.

# Основные этапы вычисления

## Схема декодирования классического кода Хэмминга (7;4):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Биты: | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |
|  | r1 | r2 | i1 | r3 | i2 | i3 | i4 | S |
| 1 | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x | S3 |

*Таблица 1.*

Для декодирования нужно посчитать S1, S2 и S3:

S1=r1⊕i1⊕i2⊕i4

S2=r2⊕i1⊕i3⊕i4

S3=r3⊕i2⊕i3⊕i4

(Для каждого S считается сумма по модулю 2 для всех битов отмеченных знаком “x” в соответствующей строчке)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S1,S2,S3 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Ошибка в бите: | нет | r3 (№4) | r2 (№2) | i3 (№6) | r1 (№1) | i2 (№5) | i1 (№3) | i4 (№7) |

*Таблица 2.*

Чтобы определить, в каком бите была допущена ошибка (и была ли она допущена), можно рассмотреть получившиеся результаты S1, S2, S3 и найти соответствующий им столбец в 1-ой таблице (пустая клетка соответствует 0, а x соответствует 1)

*Пример: если S1=0, S2=1, S3=0, то им соответствует столбец 2, т.е. ошибка была допущена в бите №2 (r2)*

## Задание №1

48.

0101011

S1=0⊕0⊕0⊕1=1

S2=1⊕0⊕1⊕1=1

S3=1⊕0⊕1⊕1=1

S1,S2,S3=111, т.е. ошибка допущена в бите №7 (i4).

Правильное сообщение:

0101010

(только информационные биты: 0010)

85.

0000110

S1=0⊕0⊕1⊕0=1

S2=0⊕0⊕1⊕0=1

S3=0⊕1⊕1⊕0=0

S1,S2,S3=110, т.е. ошибка допущена в бите №3 (i1).

Правильное сообщение:

0010110

(только информационные биты: 1110)

10.

1010000

S1=1⊕1⊕0⊕0=0

S2=0⊕1⊕0⊕0=1

S3=0⊕0⊕0⊕0=0

S1,S2,S3=010, т.е. ошибка допущена в бите №2 (r2).

Правильное сообщение:

1110000

(только информационные биты: 1000)

67.

1100100

S1=1⊕0⊕1⊕0=0

S2=1⊕0⊕0⊕0=1

S3=0⊕1⊕0⊕0=1

S1,S2,S3=011, т.е. ошибка допущена в бите №6 (i3).

Правильное сообщение:

1100110

(только информационные биты: 0110)

## Схема декодирования классического кода Хэмминга (15;11):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Биты: | 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 | S |
| 1 | x |  | x |  | x |  | x |  | x |  | x |  | x |  | x | S1 |
| 2 |  | x | x |  |  | x | x |  |  | x | x |  |  | x | x | S2 |
| 4 |  |  |  | x | x | x | x |  |  |  |  | x | x | x | x | S3 |
| 8 |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | S4 |

*Таблица 3.*

Для декодирования нужно посчитать S1, S2 и S3:

S1=r1⊕i1⊕i2⊕i4⊕i5⊕i7⊕i9⊕i11

S2=r2⊕i1⊕i3⊕i4⊕i6⊕i7⊕i10⊕i11

S3=r3⊕i2⊕i3⊕i4⊕i8⊕i9⊕i10⊕i11

S4=r4⊕i5⊕i6⊕i7⊕i8⊕i9⊕i10⊕i11

(Для каждого S считается сумма по модулю 2 для всех битов отмеченных знаком “x” в соответствующей строчке)

Чтобы определить, в каком бите была допущена ошибка (и была ли она допущена), можно рассмотреть получившиеся результаты S1, S2, S3, S4 и найти соответствующий им столбец в таблице (пустая клетка соответствует 0, а x соответствует 1)

*Пример: если S1=0, S2=1, S3=0, S4=1 то им соответствует столбец 10, т.е. ошибка была допущена в бите №10 (i6)*

## Задание №2

66.

001110000100100

S1=0⊕1⊕1⊕0⊕0⊕0⊕1⊕0=1

S2=0⊕1⊕0⊕0⊕1⊕0⊕0⊕0=0

S3=1⊕1⊕0⊕0⊕0⊕1⊕0⊕0=1

S4=0⊕0⊕1⊕0⊕0⊕1⊕0⊕0=0

S1,S2,S3,S4=1010, т.е. ошибка допущена в бите №5 (i2).

Правильное сообщение:

001100000100100

(только информационные биты: 10000100100)

## Задание №3

(48+85+10+67+66)\*4=1104

2r≥r+i+1, где r – число проверочных битов, i – число информационных битов.

Т.е. 2r-r-1≥1104. Минимальное подходящее r: 11.

Коэффициент избыточности: = ≈ 0,0099

## Задание №4

Написано на Python. Работает с классическими кодами Хэмминга (не только (7,4)). На вход должна подаваться строка из единиц и нулей правильной длины (7, 15 и т.д.)

def error():

print("Error!")

quit()

def swap(a):

if a==1:

return 0

else:

return 1

def tell():

print("Сообщение: ")

for i in range (0, len(a)+1):

i2=i

if i!=0 and i!=1:

infcheck=0

while i!=1:

if i%2==1:

infcheck=1

i=i//2

if infcheck==1:

print(a[i2-1], end=’’)

def name(b):

b2=b

count=0

infcheck=0

while b!=1:

if b%2==1:

infcheck=1

b=b//2

count=count+1

if infcheck==0:

return "r", count+1

else:

return "i", b2-count-1

a=list(input())

b=len(a)+1

count=0

while b!=1:

if b%2==1:

error()

b=b//2

count=count+1

for i in range (0, len(a)):

if a[i]!='0' and a[i]!='1':

error()

a[i]=int(a[i])

print(a)

s=list(range(count))

for i in range (0, count):

s[i]=0

c=2\*\*i-1

for j in range (0, 2\*\*(count-1)//2\*\*i):

for k in range (2\*\*i):

s[i]=s[i]+a[c]

c=c+1

c=c+2\*\*i

s[i]=s[i]%2

errbit=0

for i in range (0, count):

errbit=errbit+s[i]\*2\*\*i

if errbit!=0:

a[errbit-1]=swap(a[errbit-1])

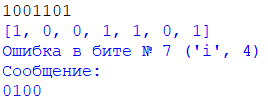
print("Ошибка в бите №", errbit, name(errbit))

else:

print("Ошибок нет")

tell()

Вывод:



*Рисунок 1*

# Заключение

В ходе выполнения данной работы я научился декодировать сообщения, закодированные кодом Хэмминга, рассчитывать коэффициент избыточности и использовать полученные навыки на практике.