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**

Синтез помехоустойчивого кода

Вариант №38

Группа: P3131

Выполнил: Воронин И.А.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2021г

Оглавление

[Задание 3](#_Toc84892090)

[Основные этапы вычисления 4](#_Toc84892091)

[Заключение 9](#_Toc84892092)

# 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 №1 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 №2 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.

# Основные этапы вычисления

Код Хэмминга(7:4) -

Вариант 61

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |

S1 = ( i1+i2+i4)%2 = ( 0 + 1 + 0 )%2 = 1 != r1:False

S2 = ( i1+i3+i4)%2 = ( 0 + 0 + 0 )%2 = 0 != r2: False

S3 = (i2+i3=i4)%2 = ( 1 + 0 + 0 )%2 = 1 != r3: False

S = 111 = 7

Меняем 7 бит и проверяем:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |

S1 = ( 0 + 1 + 1 )%2 = 0 == r1 : True

S2 = (0 + 0 + 1)%2 = 1 == r2: True

S3 = ( 1 + 0 + 1 )%2 = 0 == r3: True

После декодирования видно, что ошибка находится в 7 бите.

После исправления код имеет вид : 0100101

Вариант 102

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |

S1 = ( i1+i2+i4)%2 = ( 0 + 1 + 1 )%2 = 0 == r1:True

S2 = ( i1+i3+i4)%2 = ( 0 + 1 + 1 )%2 = 0 != r2: False

S3 = (i2+i3=i4)%2 = ( 1 + 1 + 1 )%2 = 1 != r3: False

S = 110 = 6

После декодирования видно, что ошибка находится в 6 бите.

Меняем 6 бит и проверяем:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |

S1 = ( 0 + 1 + 1 )%2 = 0 == r1 : True

S2 = (0 + 0 + 1)%2 = 1 == r2: True

S3 = ( 1 + 0 + 1 )%2 = 0 == r3: True

После исправления код имеет вид : 0100101

Вариант 60

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |

S1 = ( i1+i2+i4)%2 = ( 1 + 1 + 0 )%2 = 0 == r1:True

S2 = ( i1+i3+i4)%2 = ( 1 + 0 + 0 )%2 = 0 != r2: False

S3 = (i2+i3=i4)%2 = ( 1 + 0 + 0 )%2 = 1 == r3: True

S = 010 = 2

После декодирования видно, что ошибка находится в 2 бите.

Меняем 2 бит и проверяем:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |

S1 = (1 + 1 + 0 )%2 = 0 == r1 : True

S2 = (1 + 0 + 0)%2 = 1 == r2: True

S3 = ( 1 + 0 + 0 )%2 = 1 == r3: True

После исправления код имеет вид : 0111100

Вариант 24

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |

S1 = ( i1+i2+i4)%2 = ( 1 + 0 + 1 )%2 = 0 != r1:False

S2 = ( i1+i3+i4)%2 = ( 1 + 0 + 1 )%2 = 0 == r2: True

S3 = (i2+i3=i4)%2 = ( 0 + 0 + 1 )%2 = 1 != r3: False

S = 101 = 5

Меняем 5 бит и проверяем:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |

S1 = ( 1 + 1 + 1 )%2 = 1 == r1 : True

S2 = (1 + 0 + 1)%2 = 0 == r2: True

S3 = ( 1 + 0 + 1 )%2 = 0 == r3: True

После декодирования видно, что ошибка находится в 5 бите.

После исправления код имеет вид : 1010101

Код Хэмминга(15:11)

Вариант 42

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

S1 = (i1+i2+i4+i5+i7+i9+i11)%2 = 0%2 = 0 == r1:True

S2 = (i1+i3+i4+i6+i7+i10+i11)%2 = ( 0+1+0+1+0+1+0 ) % 2 = 1 == r2: True

S3 = (i2+i3+i4+i8+i9+i10+i11)%2 = (0+1+0+0+0+1+0)%2 = 0 != r3:False

S3 = (i5+i6+i7+i8+i9+i10+i11)%2= (0+1+0+0+0+1+0) %2 = 0 != r4:False

S = 1100 = 12

Меняем 12 бит:

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

S1 = (i1+i2+i4+i5+i7+i9+i11)%2 = 0%2 = 0 == r1:True

S2 = (i1+i3+i4+i6+i7+i10+i11)%2 = ( 0+1+0+1+0+1+0 ) % 2 = 1 == r2: True

S3 = (i2+i3+i4+i8+i9+i10+i11)%2 = (0+1+0+1+0+1+0)%2 = 1 == r3:True

S3 = (i5+i6+i7+i8+i9+i10+i11)%2= (0+1+0+1+0+1+0) %2 = 1 == r4:True

После исправления код имеет вид: 010101010101010

Пункт №8:

(60+24+61+102+42) \* 4 = 1156

r – количество контрольных разрядов

i – количество информационных разрядов

min r = 11

коэффициент избыточности равен: = = 0,009425

Пункт №9

h = [int(s) for s in input().strip()]

k = -1

if h[0] != (h[2]+h[4]+h[6])%2:

k += 1

if h[1] != (h[2]+h[5]+h[6])%2:

k += 2

if h[3] != (h[4]+h[5]+h[6])%2:

k += 4

if k != -1:

print('Wrong bit:'+str(k+1))

if h[k] == 0:

h[k] = 1

else:

h[k] = 0print('Correct code:' + str(h[2]) + str(h[4]) + str(h[5]) + str(h[6]))

# Заключение

В ходе выполнения данной лабораторной работы я научился кодировать и декодировать код Хэмминга и вычислять для него минимальное кол-во проверочных разрядов и коэфф. избыточности.