Информатика

Лабораторная работа №2

Вариант №367422=72

Выполнил:

Мухамеджанов Артур Илдусович

P3114

Преподаватели:

Балакшин Б. В.

Машина Е. А.

Оглавление

[Задание 3](#_Toc115766419)

[Основные этапы вычисления 3](#_Toc115766420)

[1. 1101011 4](#_Toc115766421)

[2. 0111110 4](#_Toc115766422)

[3. 0001001 4](#_Toc115766423)

[4. 1010011 5](#_Toc115766424)

[5. 001110001110100 6](#_Toc115766425)

[6. (54 + 91 + 16 + 51 + 71) \* 4 = 1132 6](#_Toc115766426)

[7. Программа 6](#_Toc115766427)

[Вывод 7](#_Toc115766428)

[Список литературы 7](#_Toc115766429)

# Задание

1) 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.

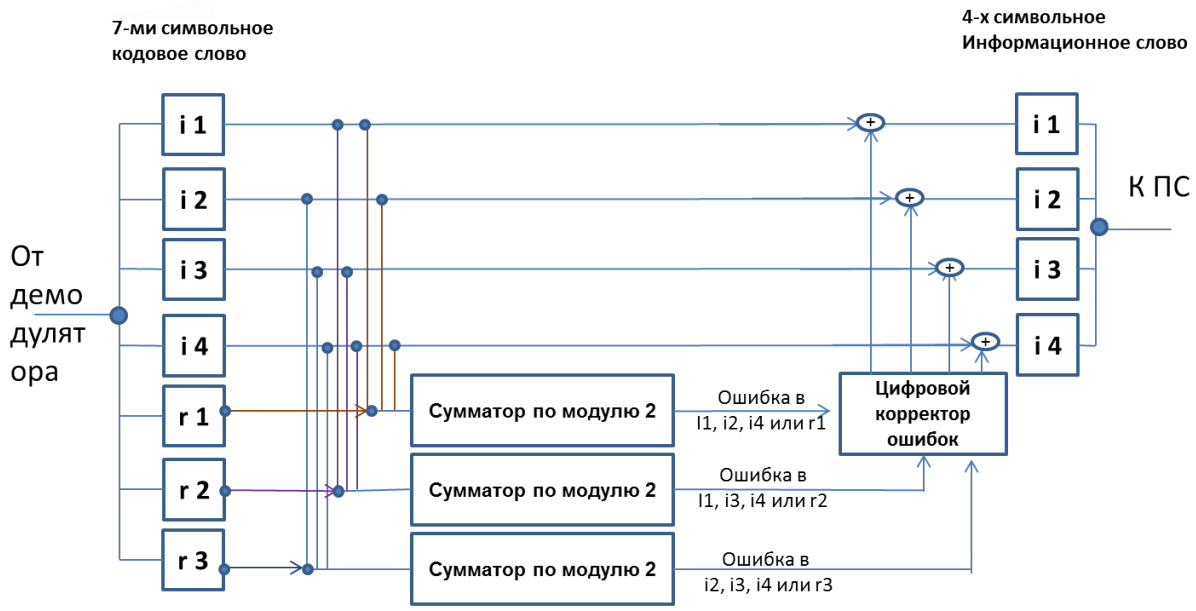
2) Показать, исходя из выбранного варианта сообщений (по 1 у каждого – часть №2 в варианте), имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение.

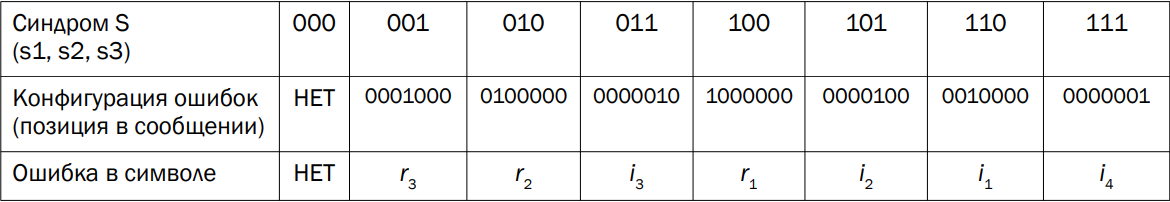
3) Сложить номера всех 5 вариантов заданий. Умножить полученное число на 4. Принять данное число как число информационных разрядов в передаваемом сообщении. Вычислить для данного числа минимальное число проверочных разрядов и коэффициент избыточности.

4) 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.

Варианты: 54 91 16 51 71

# Основные этапы вычисления





## 1101011

В)54

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r1 | r2 | i1 | r3 | i2 | i3 | i4 | Синдром |  |
|  | 1 | 1 | 0 | 1 | 0 | 1 | 1 | S |  |
| 1 | x |  | x |  | x |  | x | S1 | 0 |
| 2 |  | x | x |  |  | x | x | S2 | 1 |
| 4 |  |  |  | x | x | x | x | S3 | 1 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = 1 ⊕ 0 ⊕ 0 ⊕ 1 = 0

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 1 = 1

S1 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = 1 ⊕ 0 ⊕ 1 ⊕ 1 = 1

S = 011 (за ошибочный бит отвечает r3 и r2)

Ошибка в i3

Верное сообщение: 1101001

Отбрасываем контрольные биты: 0001

## 2. 0111110

В)91

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r1 | r2 | i1 | r3 | i2 | i3 | i4 | Синдром |  |
|  | 0 | 1 | 1 | 1 | 1 | 1 | 0 | S |  |
| 1 | x |  | x |  | x |  | x | S1 | 0 |
| 2 |  | x | x |  |  | x | x | S2 | 1 |
| 4 |  |  |  | x | x | x | x | S3 | 1 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = = 0 ⊕ 1 ⊕ 1 ⊕ 0 = 0

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1

S1 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = = 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1

S = 011 (за ошибочный бит отвечает r2 и r3)

Ошибка в i3

Верное сообщение: 0111100

Отбрасываем контрольные биты: 1100

## 3. 0001001

В)16

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r1 | r2 | i1 | r3 | i2 | i3 | i4 | Синдром |  |
|  | 0 | 0 | 0 | 1 | 0 | 0 | 1 | S |  |
| 1 | x |  | x |  | x |  | x | S1 | 1 |
| 2 |  | x | x |  |  | x | x | S2 | 1 |
| 4 |  |  |  | x | x | x | x | S3 | 0 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1

S1 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = = 1 ⊕ 0 ⊕ 0 ⊕ 1 = 0

S = 110 (за ошибочный бит отвечает r1 и r2)

Ошибка в i1

Верное сообщение: 0011001

Отбрасываем контрольные биты: 1001

## 4. 1010011

В)51

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r1 | r2 | i1 | r3 | i2 | i3 | i4 | Синдром |  |
|  | 1 | 0 | 1 | 0 | 0 | 1 | 1 | S |  |
| 1 | x |  | x |  | x |  | x | S1 | 1 |
| 2 |  | x | x |  |  | x | x | S2 | 1 |
| 4 |  |  |  | x | x | x | x | S3 | 0 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 = = 1 ⊕ 1 ⊕ 0 ⊕ 1 = 1

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 = = 0 ⊕ 1 ⊕ 1 ⊕ 1 = 1

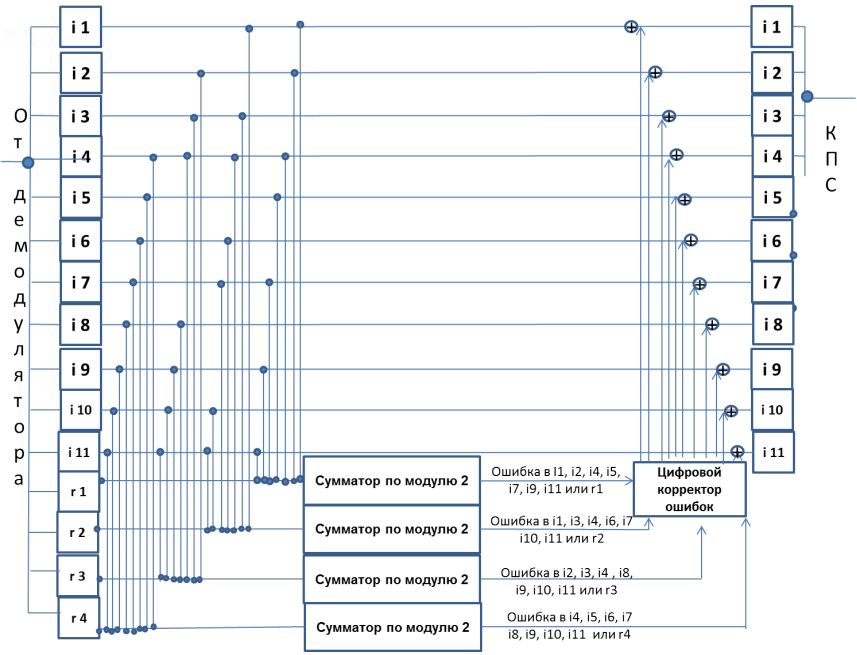
S1 = r3 ⊕ i2 ⊕ i3 ⊕ i4 = = 0 ⊕ 0 ⊕ 1 ⊕ 1 = 0

S = 110 (за ошибочный бит отвечает r1 и r2)

Ошибка в i1

Верное сообщение: 1000011

Отбрасываем контрольные биты: 0011



## 5. 001110001110100

В)71

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | r1 | r2 | i1 | r3 | i2 | i3 | i4 | r4 | i5 | i6 | i7 | i8 | i9 | i10 | i11 |  |  |
|  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | S |  |
| 1 | x |  | x |  | x |  | x |  | x |  | x |  | x |  | x | S1 | 1 |
| 2 |  | x | x |  |  | x | x |  |  | x | x |  |  | x | x | S2 | 1 |
| 4 |  |  |  | x | x | x | x |  |  |  |  | x | x | x | x | S3 | 1 |
| 8 |  |  |  |  |  |  |  | x | x | x | x | x | x | x | x | S4 | 0 |

S1 = r1 ⊕ i1 ⊕ i2 ⊕ i4 ⊕ i5 ⊕ i7 ⊕ i9 ⊕ i11 = 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 = 1

S2 = r2 ⊕ i1 ⊕ i3 ⊕ i4 ⊕ i6 ⊕ i7 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 0 = 1

S3 = r3 ⊕ i2 ⊕ i3 ⊕ i4 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1

S4 = r4 ⊕ i5 ⊕ i6 ⊕ i7 ⊕ i8 ⊕ i9 ⊕ i10 ⊕ i11 = 0 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 ⊕ 1 ⊕ 0 ⊕ 0 = 0

S = 1110 (за ошибочный бит отвечает r1, r2, r3)

Ошибочный бит – i4

Верное сообщение: 001110101110100

Отбрасываем контрольные биты: 11011110100

## 6. (54 + 91 + 16 + 51 + 71) \* 4 = 1132

1132 бита информации

1132 + 1 +11 < -> минимальное число проверочных битов - 11 битов

Коэффициент избыточности:

r = 11 / (1132 + 11) ≈ 0,0096238

## 7. Программа

Пример программы на языке Python:

s = input()

while len(s) != 7 or (s.count('1') + s.count('0') != 7):  
 print("Сообщение введено неверно, повторите попытку")  
 s = input()  
r1 = int(s[0])  
r2 = int(s[1])  
i1 = int(s[2])  
r3 = int(s[3])  
i2 = int(s[4])  
i3 = int(s[5])  
i4 = int(s[6])  
s1 = str((r1 + i1 + i2 + i4) % 2)  
s2 = str((r2 + i1 + i3 + i4) % 2)  
s3 = str((r3 + i2 + i3 + i4) % 2)  
s = s1 + s2 + s3  
if s == '000':  
 print("Ошибок нет")  
 print("Верное сообщение:", i1,i2,i3,i4, sep = '')  
if s == '001':  
 print("Ошибка в r3")  
 print("Верное сообщение:", i1,i2,i3,i4, sep = '')  
if s == '010':  
 print("Ошибка в r2")  
 print("Верное сообщение:", i1,i2,i3,i4, sep = '')  
if s == '011':  
 print("Ошибка в i3")  
 print("Верное сообщение:", i1,i2,(i3 + 1) % 2,i4, sep = '')  
if s == '100':  
 print("Ошибка в r1")  
 print("Верное сообщение:", i1,i2,i3,i4, sep = '')  
if s == '101':  
 print("Ошибка в i2")  
 print("Верное сообщение:", i1,(i2+1)%2,i3,i4, sep = '')  
if s == '110':  
 print("Ошибка в i1")  
 print("Верное сообщение:", (i1+1)%2,i2,i3,i4, sep = '')  
if s == '111':  
 print("Ошибка в i4")  
 print("Верное сообщение:", i1,i2,i3,(i4+1)%2, sep = '')

# Вывод

Я научился применять код Хэмминга для поиска ошибок при передаче сообщения. Я научился высчитывать коэффициент избыточности. Смог реализовать работу кода Хэмминга для 7 бит информации на языке Python.

# Список литературы

1. Алексеев Е. Г., Богатырев С. Д. Информатика. Мультимедийный электронный учебник. Саранск: 2009.
2. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.: ил.