Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по информатике

Вариант №68

Выполнил:

Эрбаев Ильдус

Группа: P3112

Приняла: Рудникова

Тамара Владимировна

???

?



Санкт-Петербург, 2021

Оглавление

[I. Построить схему декодирования классического кода Хэмминга (7;4). Показать, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение 3](#_Toc84324407)

[1. Вариант сообщения №50 3](#_Toc84324408)

[2. Вариант сообщения №87 4](#_Toc84324409)

[3. Вариант сообщения №12 5](#_Toc84324410)

[4. Вариант сообщения №91 6](#_Toc84324411)

[II. Построить схему декодирования классического кода Хэмминга (15;11). Показать, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение 6](#_Toc84324412)

[Вариант сообщения №68 6](#_Toc84324413)

[III. 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](#_Toc84324414)

[IV. 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 9](#_Toc84324415)

[V. Вывод 10](#_Toc84324416)

Вариант №68

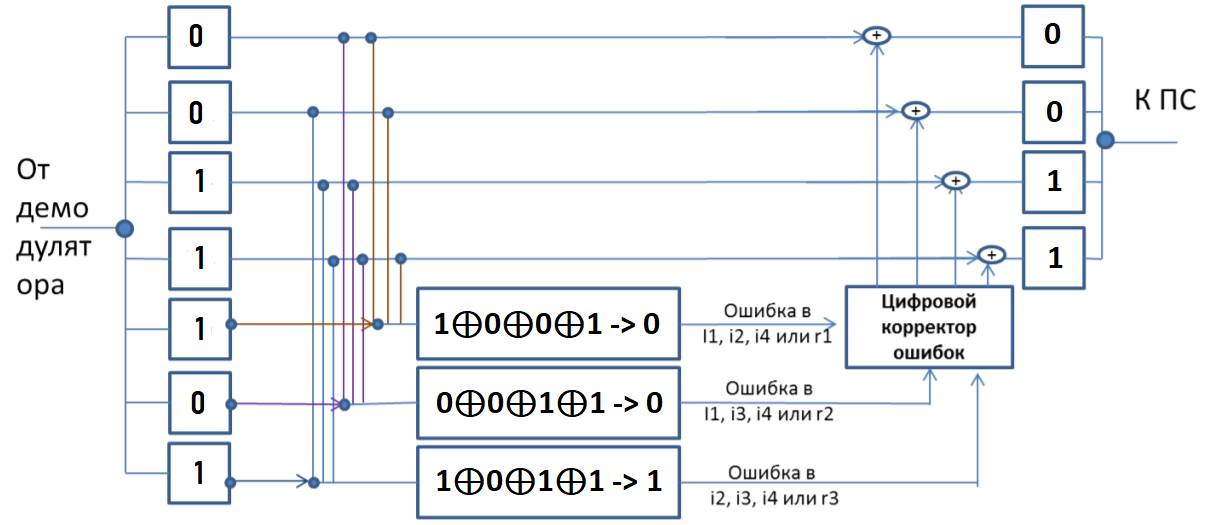
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | 1 | | | | 2 |
| 68 | 50 | 87 | 12 | 91 | 68 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ALT | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| r1 | r2 | i1 | r3 | i2 | i3 | i4 |
| 1 | 50 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 87 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 12 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 91 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ALT | | 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 |
| 2 | 68 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

I. Построить схему декодирования классического кода Хэмминга (7;4). Показать, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение:

1. Вариант сообщения №50:



Для начала находим 3 разных варианта суммы по модулю 2 по 4 битам:

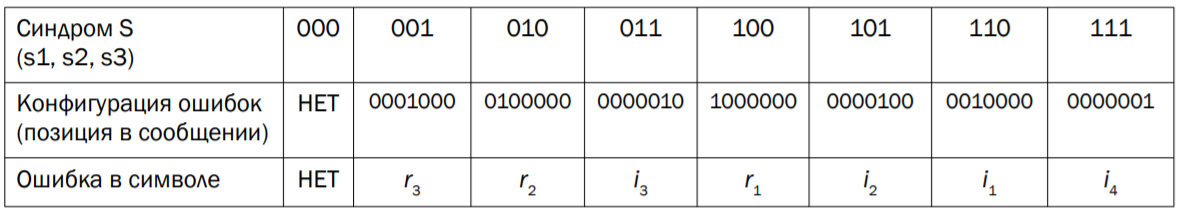
1) r1, i1, i2, i4

2) r2, i1, i3, i4

3) r3, i2, i3, i4

В схеме указаны вышеуказанные операции.

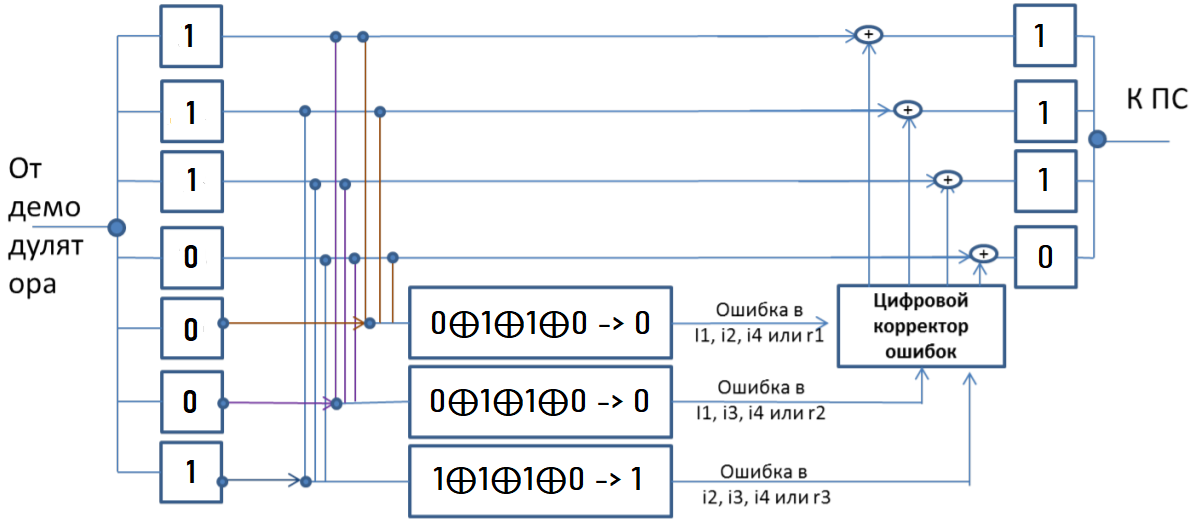
Затем находим синдром последовательности путем сложения сумм по модулю 2, в данном случае синдромом является «001».



Теперь благодаря таблице находим ошибку по синдрому последовательности, ошибкой является бит чётности r3.

Так как ошибки в информационных битах не обнаружены, то нам не нужно инвертировать их, соответственно, правильное сообщение: i1 = 0, i2 = 0, i3 = 1, i4 = 1.

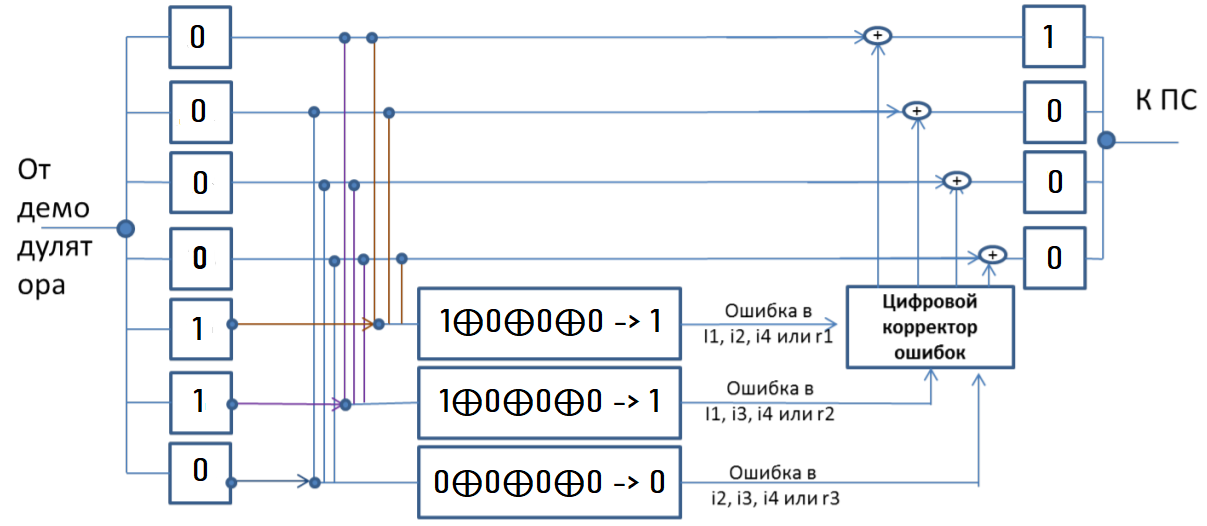
2. Вариант сообщения №87:



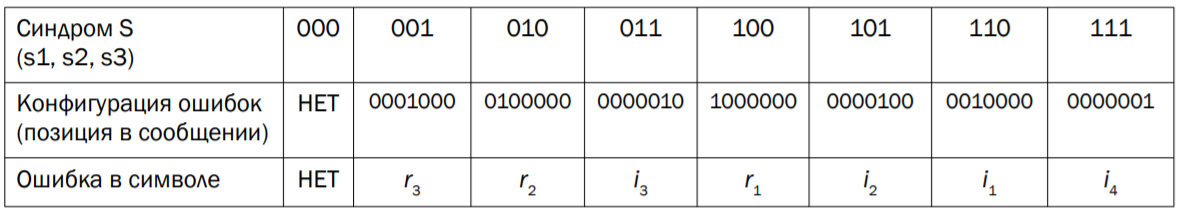
Повторяем действия, как в предыдущем варианте и находим синдром последовательности, в этом случае она идентична предыдущей «001». Сверяемся с таблицей и понимаем, что ошибка в бите r3.

Правильное сообщение: i1 = 1, i2 = 1, i3 = 1, i4 = 0.

3. Вариант сообщения №12:

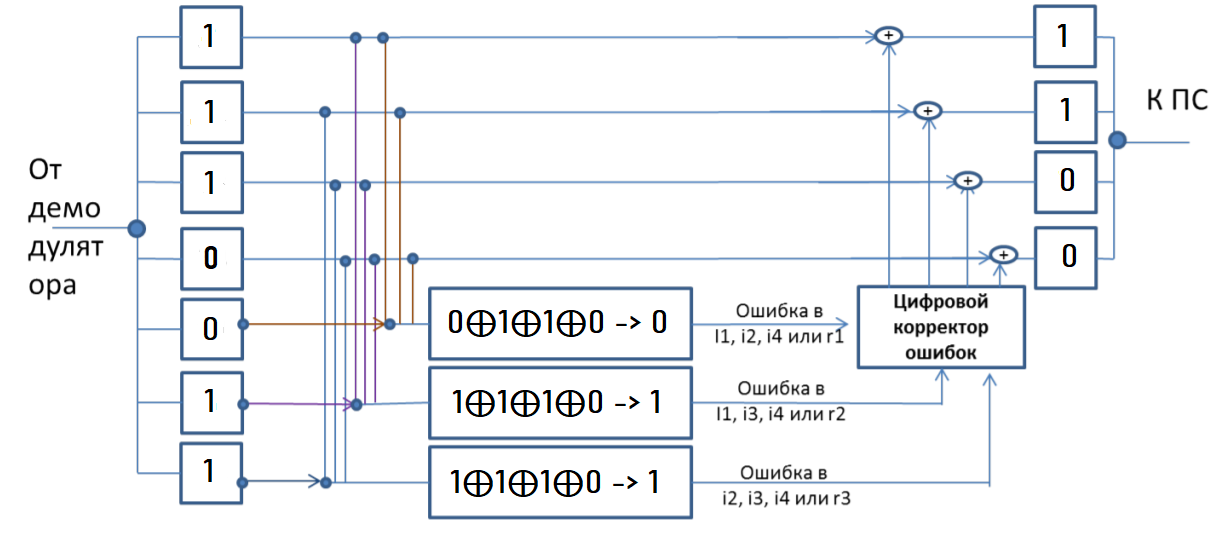


Здесь, можно заметить, что в принятом сообщения все информационные биты нули, но первые два бита четности равны единице, тогда, не считая, можно определить синдром последовательности «110».



По таблице ошибкой является информационный бит i1. Так как нам необходимо получить правильное сообщение, инвертируем ошибочный бит: i1 = 1, i2 = 0, i3 = 0, i4 = 0.

4. Вариант сообщения №91:



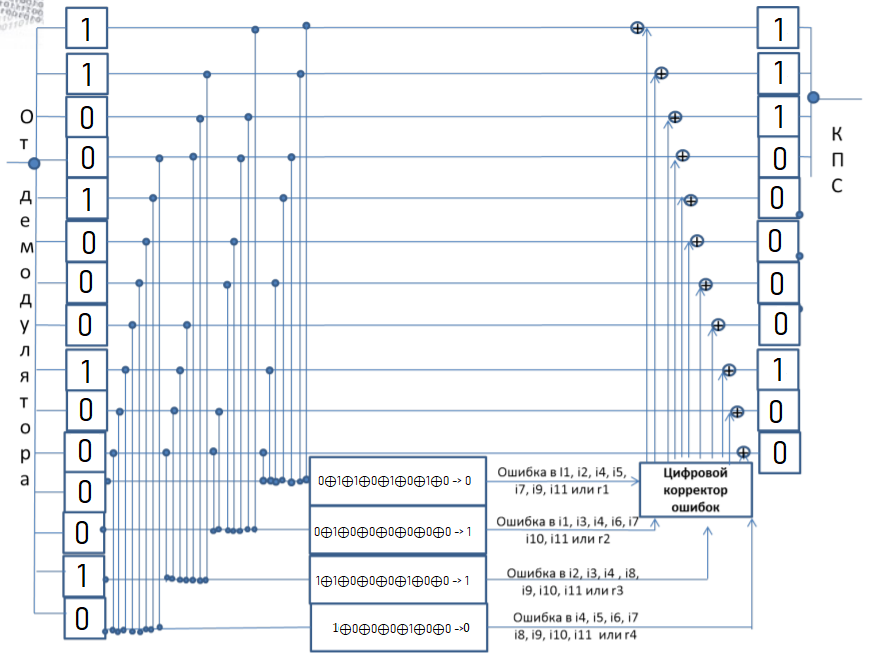
Проделываем точно такие же операции и получаем синдром последовательности «011», по таблице ошибкой является информационный бит i3, инвертируем его и получаем правильное сообщение: i1 = 1, i2 = 1, i3 = 0, i4 = 0.

II. Построить схему декодирования классического кода Хэмминга (15;11). Показать, имеются ли в принятом сообщении ошибки, и если имеются, то какие. Подробно прокомментировать и записать правильное сообщение:

Вариант сообщения №68:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ALT | | 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 |
| 2 | 68 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

Схема декодирования:



Алгоритм дешифровки кода (15;11) очень схож с кодом (7;4), но теперь у нас добавляется бит четности и 7 информационных битов.

Для начала найдем синдром последовательности, который теперь состоит из четырех контрольных сумм:

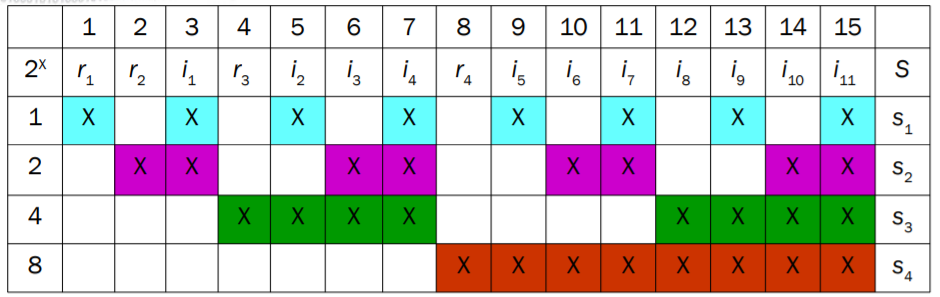
s1 = r1⊕i1⊕i2⊕i4⊕i5⊕i7⊕i9⊕i11

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

Смотря на схему, синдром последовательности равен «0110».



Теперь, используя таблицу, нам необходимо найти ошибочный бит. По последовательности мы видим, что ошибка находится в сумме s2 и s3. По таблице мы можем понять, что бит i3 задействован в этих суммах, но не задействован в других, соответственно, ошибочным битов является i3.

Правильное сообщение:

i1 = 1, i2 = 1, i3 = 1, i4 = 0, i5 = 0, i6 = 0, i7 = 0, i8 = 0, i9 = 1, i10 = 0, i11 = 0.

III. 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:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | 1 | | | | 2 |
| 68 | 50 | 87 | 12 | 91 | 68 |

Пусть количество информационных разрядов равняется i, тогда i = (50+87+12+91+68)\*4 = 1232.

Определение минимального числа проверочных разрядов:

2r ≥ r + i + 1 -> 2r – r ≥ 1233

При r = 10: 210-10 ≥ 1233 -> 1014 ≥ 1233

При r = 11: 211-11 ≥ 1233 -> 2037 ≥ 1233

Значит, нам понадобится r = 11 проверочных разрядов.

Коэффициент избыточности — отношение числа проверочных разрядов (r) к общему числу разрядов (n = i + r).

Коэффициент избыточности: r/(i+r) -> 11/(1232+11) -> 0.0088.

IV. 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:

r1, r2, i1, r3, i2, i3, i4 = map(int, input("Введите числа r1, r2, i1, r3, i2, i3, i4 через пробел: ").split())

s1 = (r1 + i1 + i2 + i4) % 2

s2 = (r2 + i1 + i3 + i4) % 2

s3 = (r3 + i2 + i3 + i4) % 2

syndrome\_s = str(s1) + str(s2) + str(s3)

cases = ['000', '001', '010', '011', '100', '101', '110', '111']

cases\_bits = ['Не найден', 'r3', 'r2', 'i3', 'r1', 'i2', 'i1', 'i4']

for i in range(len(cases)):

if syndrome\_s == cases[i]:

problem\_bit = cases\_bits[i]

break

if problem\_bit == 'i1':

i1 = (i1 + 1)%2

elif problem\_bit == 'i2':

i2 = (i2 + 1)%2

elif problem\_bit == 'i3':

i3 = (i3 + 1)%2

elif problem\_bit == 'i4':

i4 = (i4 + 1)%2

print('Правильное сообщение:', i1, i2, i3, i4)

print('Бит с ошибкой:', problem\_bit)

V. Вывод:

В результате выполнения лабораторной работы, я познакомился с кодом Хэмминга, научился дешифровывать сообщения по схеме, находить бит с ошибкой и реализовывать алгоритм дешифрования.