

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Отчёт по лабораторной работе №2**

Тема: Синтез помехоустойчивого кода

Выполнил**:** студент группы Р3132 Гузалов Тимур Павлович

Проверил**:** Белокон Юлия Алексеевна

Дата сдачи: 14.10.2024

2024

Итак, мой вариант 56, мне даны следующие коды сообщений:

40, 73,105, 25

40

1

1

0

1

0

1

0

73

0

0

1

0

1

0

1

105

0

1

1

1

1

1

1

25

1

0

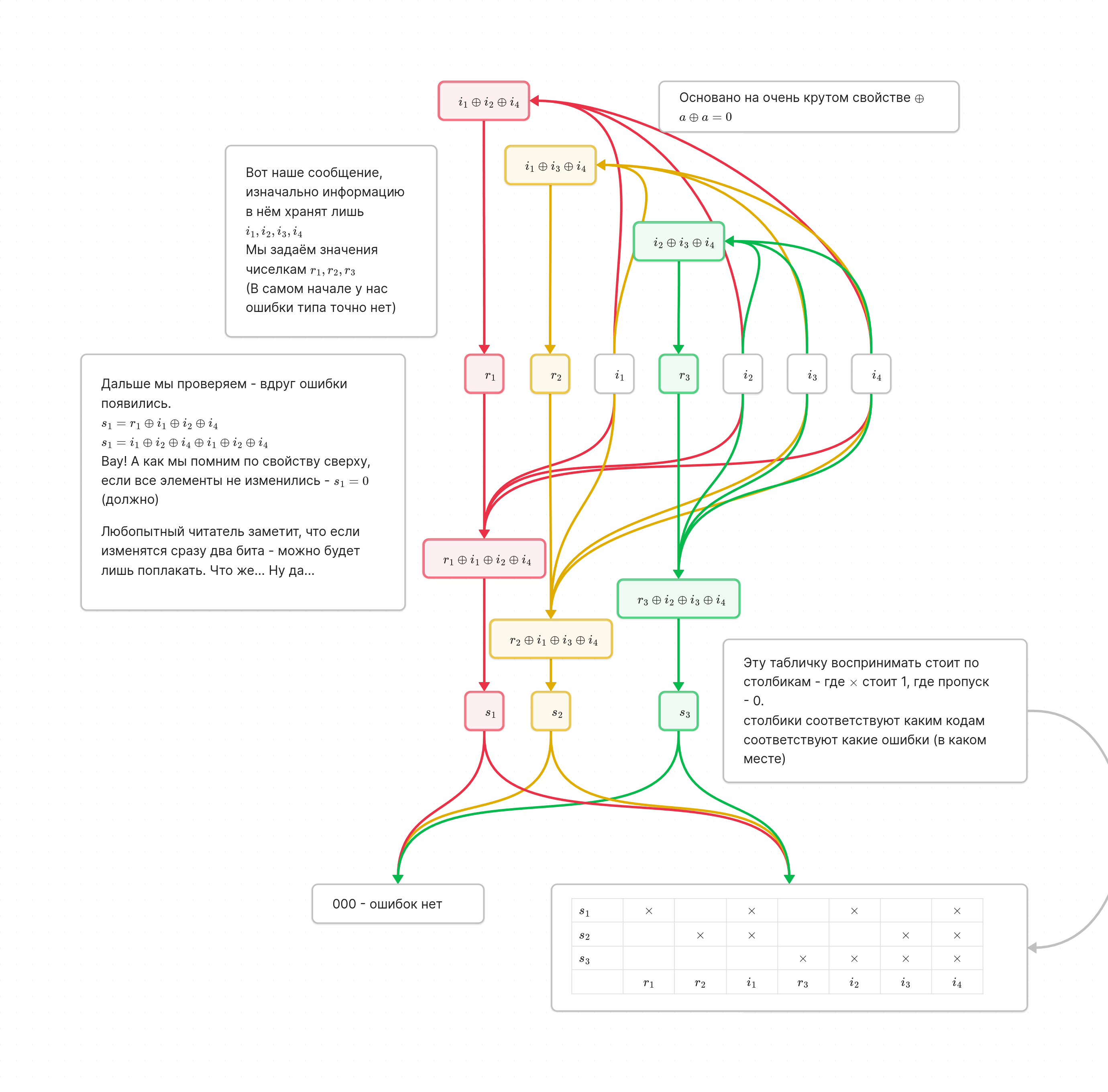
1

1

0

0

1

Построим схему декодирования:

Теперь наконец-то рассмотрим наши коды:

1)

40

1

1

0

1

0

1

0

s1 = 1 ^ 0 ^ 0 ^ 0 = 1

s2 = 1 ^ 0 ^ 1 ^ 0 = 0

s3 = 1 ^ 0 ^ 1 ^ 0 = 0

Следовательно ошибка находится в r1

Следовательно верное сообщение вероятно таково:

0 1 0 1 0 1 0

2)

73

0

0

1

0

1

0

1

s1 = 0 ^ 1 ^ 1 ^ 1 = 1

s2 = 0 ^ 1 ^ 0 ^ 1 = 0

s3 = 0 ^ 1 ^ 0 ^ 1 = 0

Следовательно ошибка в r1

Следовательно верное сообщение вероятно таково:

1 0 1 0 1 0 1

3)

105

0

1

1

1

1

1

1

s1 = 0 ^ 1 ^ 1 ^ 1 = 1

s2 = 1 ^ 1 ^ 1 ^ 1 = 0

s3 = 1 ^ 1 ^ 1 ^ 1 = 0

Следовательно ошибка опять в r1…

Следовательно верное сообщение вероятно таково:

1 1 1 1 1 1 1

4)

25

1

0

1

1

0

0

1

s1 = 1 ^ 1 ^ 0 ^ 1 = 1

s2 = 0 ^ 1 ^ 0 ^ 1 = 0

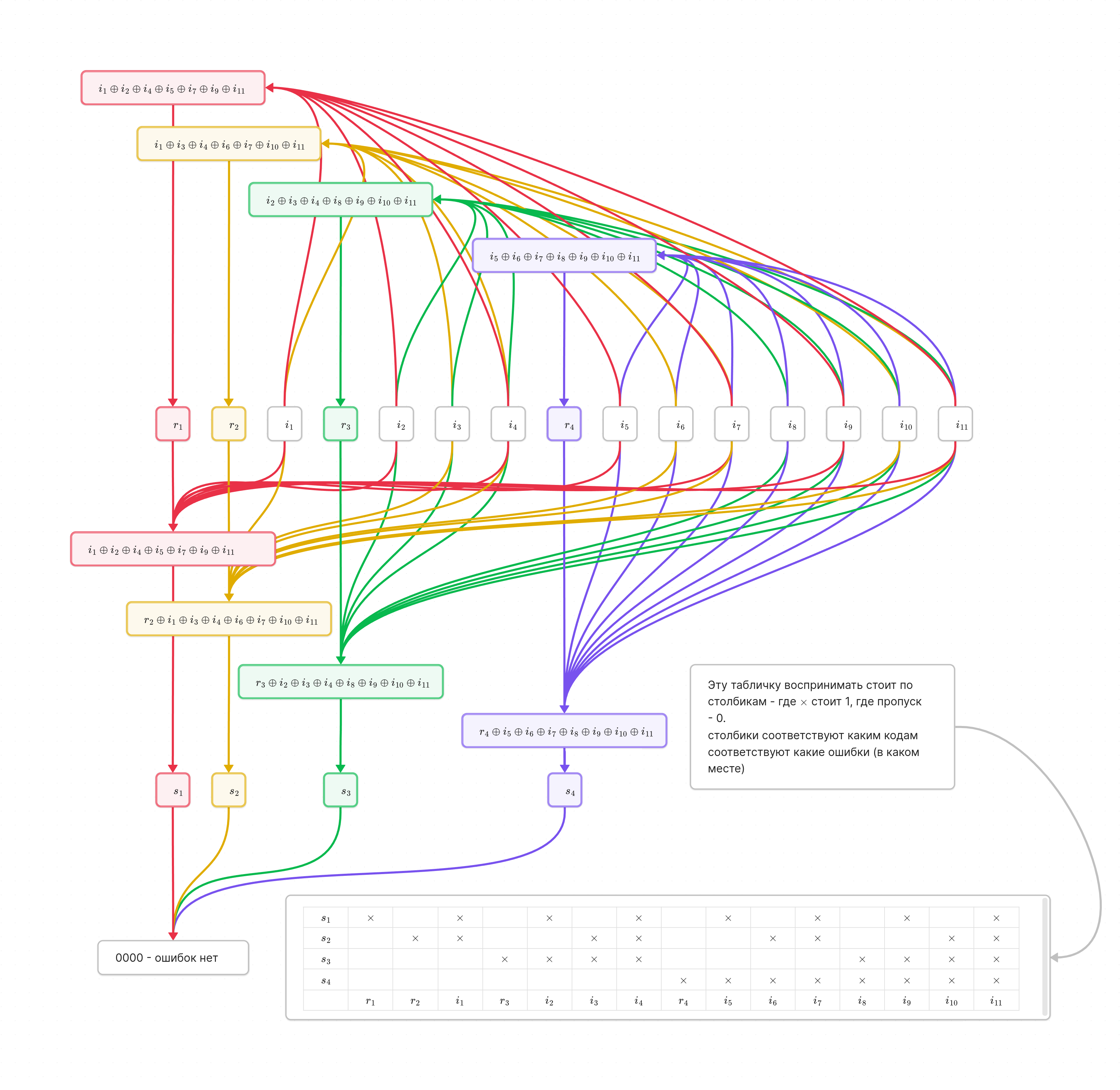
s3 = 1 ^ 0 ^ 0 ^ 1 = 0

И ошибка вновь в r1…

Следовательно верное сообщение вероятно таково:

0 0 1 1 0 0 1

5) Мой 11-символьный код:



56

0

1

0

0

0

1

1

1

0

0

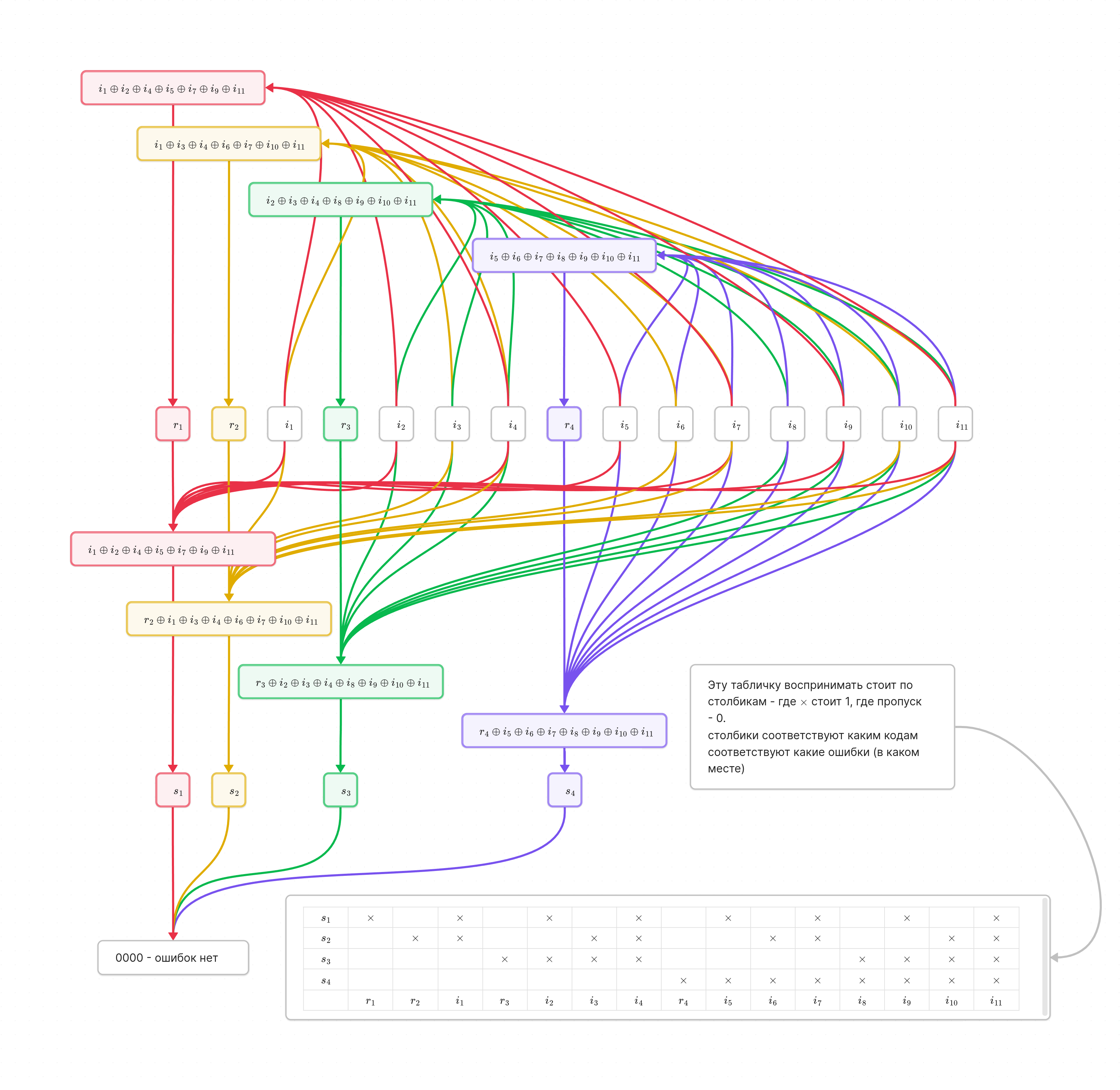
0

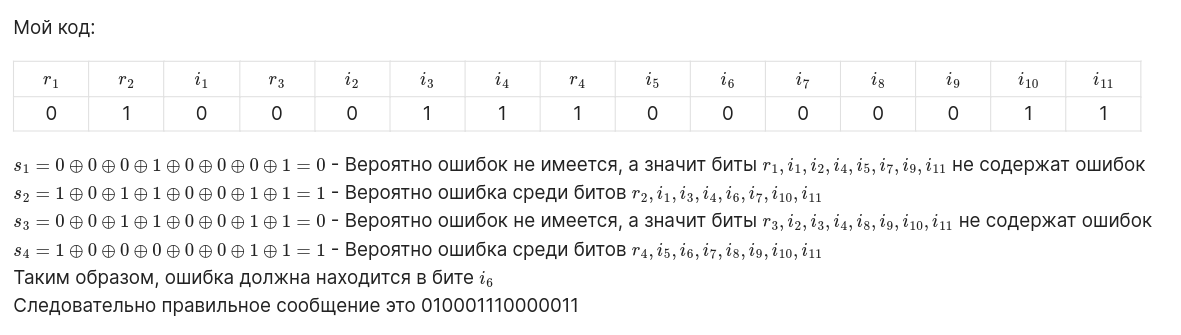
0

0

1

1

6) Алгоритм:

7)

8)

(40+73+105+25+56) \*4 = 1196

210 ⋡ 10 + 1196 + 1

211 ≥ 11 + 1196 + 1

Следовательно минимальное число проверочных разрядов — 11 штук

Коэфициент избыточности: 11/1207 = 0.009113505

9)

def is\_binary(bin\_str):

res = True

for i in bin\_str:

if (i != '0' or i != '1'):

res = False

break

return res

while True:

num = input("Input a binary Hamming code (7,4): ")

if is\_binary(num) or len(num)!= 7:

print("Чёт не то, товарищ!")

continue

r1 = int(num[0])

r2 = int(num[1])

i1 = int(num[2])

r3 = int(num[3])

i2 = int(num[4])

i3 = int(num[5])

i4 = int(num[6])

s1 = r1 ^ i1 ^ i2 ^ i4

s2 = r2 ^ i1 ^ i3 ^ i4

s3 = r3 ^ i2 ^ i3 ^ i4

errorin = 0

if(s1 == 1 and s2 == 0 and s3 == 0):

r1 = 1-r1

errorin = 1

if(s1 == 0 and s2 == 1 and s3 == 0):

r2 = 1-r2

errorin = 2

if(s1 == 1 and s2 == 1 and s3 == 0):

i1 = 1-i1

errorin = 3

if(s1 == 0 and s2 == 0 and s3 == 1):

r3 = 1-r3

errorin = 4

if(s1 == 1 and s2 == 0 and s3 == 1):

i2 = 1-i2

errorin = 5

if(s1 == 0 and s2 == 1 and s3 == 1):

i3 = 1-i3

errorin = 6

if(s1 == 1 and s2 == 1 and s3 == 1):

i4 = 1-i4

errorin = 7

print("Correct code: ", i1, i2, i3, i4)

if(errorin == 0):

print("No errors ^\_^")

else:

print("Error found in byte №", errorin, ". :(( \n")

Вывод: я ознакомился с базовыми принципами работы помехоустойчивых кодов, узнал как работает код Хэммингтона и реализовал простой алгоритм для кода (7,4) на практике.