

Санкт-Петербургский Политехнический Университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

# Телекоммуникационные технологии

Отчет по лабораторной работе №7  
Помехоустойчивое кодирование

**Работу**  
**выполнил:**  
Соболь В.О.  
Группа: 33501/4  
**Преподаватель:**  
Богач Н.В.

Санкт-Петербург  
2017

# Содержание

<b>1. Цель и задачи</b>	<b>2</b>
1.1. Цель работы . . . . .	2
1.2. Постановка задачи . . . . .	2
<b>2. Теоретическая информация</b>	<b>2</b>
2.1. Кодирование . . . . .	2
2.2. Типы помехоустойчивого кодирования . . . . .	2
2.2.1. Кодирование Хэмминга . . . . .	2
2.2.2. Циклические коды . . . . .	3
2.2.3. Коды БЧХ . . . . .	3
2.2.4. Коды Рида-Соломона . . . . .	3
<b>3. Ход работы</b>	<b>3</b>
3.1. Коды Хэмминга . . . . .	3
3.2. Циклические коды . . . . .	3
3.3. Коды БЧХ . . . . .	3
3.4. Коды Рида-Соломона . . . . .	4
<b>4. Выводы</b>	<b>5</b>
<b>5. Листинг</b>	<b>6</b>

# 1. Цель и задачи

## 1.1. Цель работы

Изучение методов помехоустойчивого кодирования и сравнения их свойств.

## 1.2. Постановка задачи

Провести кодирование/декодирование сигнала, полученного с помощью функции `randerr` кодом Хэмминга 2-мя способами: с помощью встроенных функций `encode/decode`, а также через создание проверочной и генераторной матриц и вычисление синдрома. Оценить корректирующую способность кода.

Выполнить кодирование/декодирование циклическим кодом, кодом БЧХ, кодом Рида-Соломона. Оценить корректирующую способность кода.

# 2. Теоретическая информация

## 2.1. Кодирование

Физическое кодирование — преобразование двоичных данных, осуществляемое для их передачи по физическому каналу. Физическое кодирование может менять форму, ширину полосы частот и гармонический состав сигнала в целях осуществления синхронизации приёмника и передатчика, устранения постоянной составляющей или уменьшения аппаратных затрат передачи сигнала.

Обнаружение ошибок в технике связи — действие, направленное на контроль целостности данных при записи/воспроизведении информации или при её передаче по линиям связи. Исправление ошибок — процедура восстановления информации после чтения её из устройства хранения или канала связи.

Для обнаружения ошибок используют коды обнаружения ошибок, для исправления — корректирующие коды (коды, исправляющие ошибки, коды с коррекцией ошибок, помехоустойчивые коды).

## 2.2. Типы помехоустойчивого кодирования

### 2.2.1. Кодирование Хэмминга

Коды Хэмминга — линейные коды с минимальным расстоянием 3, то есть способные исправить одну ошибку. При кодировании используется порождающая матрица  $G$

$$code = msg * G \quad (1)$$

При декодировании используется проверочная матрица  $H$ , которая позволяет определить синдром  $S$ .

$$S = code * H^T \quad (2)$$

Синдром позволяет определить в какой позиции произошла ошибка.

Коды Хэмминга являются самокорректирующимися кодами, то есть кодами, позволяющими автоматически обнаруживать ошибки при передаче данных и исправлять их.

Для построения самокорректирующегося кода, рассчитанного на исправление одиночных ошибок, одного контрольного разряда недостаточно. Как видно из дальнейшего, количество контрольных разрядов  $k$  должно быть выбрано так, чтобы удовлетворялось неравенство

$$2^k \geq k + m + 1 \quad (3)$$

или

$$k \geq \log_2(k + m + 1) \quad (4)$$

где  $m$  — количество основных двоичных разрядов кодового слова.

Построение кодов Хэмминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным.

$$r_1 = i_1 \oplus i_2 \oplus \dots \oplus i_k \quad (5)$$

$$S = i_1 \oplus i_2 \oplus \dots \oplus i_n \oplus r_1 \quad (6)$$

Тогда если  $S = 0$  - ошибки нет, иначе есть однократная ошибка.

Такой код называется  $(k + 1, k)$ . Первое число — количество элементов последовательности, второе — количество информационных символов.

Получение кодового слова выглядит следующим образом:

$$(i_1 \ i_2 \ i_3 \ i_4) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (i_1 \ i_2 \ i_3 \ i_4 \ r_1 \ r_2 \ r_3) \quad (7)$$

Получение синдрома выглядит следующим образом:

$$(i_1 \ i_2 \ i_3 \ i_4 \ r_1 \ r_2 \ r_3) \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (S_1 \ S_2 \ S_3) \quad (8)$$

### 2.2.2. Циклические коды

Циклический код — линейный код, обладающий свойством цикличности, то есть каждая циклическая перестановка кодового слова также является кодовым словом. Используется для преобразования информации для защиты её от ошибок.

### 2.2.3. Коды БЧХ

Коды Боуза — Чоудхури — Хоквингема (БЧХ-коды) — в теории кодирования это широкий класс циклических кодов, применяемых для защиты информации от ошибок. Отличается возможностью построения кода с заранее определёнными корректирующими свойствами, а именно, минимальным кодовым расстоянием. Частным случаем БЧХ-кодов является код Рида — Соломона.

### 2.2.4. Коды Рида-Соломона

Коды Рида—Соломона (англ. Reed–Solomon codes) — не двоичные циклические коды, позволяющие исправлять ошибки в блоках данных. Элементами кодового вектора являются не биты, а группы битов (блоки). Код Рида—Соломона является частным случаем БЧХ-кода.

## 3. Ход работы

При исследовании использовался код, приведённый в листинге 1.

### 3.1. Коды Хэмминга

Результат кодирования и декодирования сигнала кодом Хэмминга (7, 4) представлены ниже 1. Декодированное сообщение совпадает с исходным.

---

#### Текст 1 Полученный результат

---

Исходное сообщение

1      1      0      0

Кодированное сообщение

1      0      1      1      1      0      0

Декодированное сообщение

1      1      0      0

---

Для следующего исследования были получены порождающая и проверочная матрицы (текст 2).

---

#### Текст 2 Порождающая и проверочная матрицы

---

Порождающая матрица

1	1	0	1	0	0	0
0	1	1	0	1	0	0
1	1	1	0	0	1	0
1	0	1	0	0	0	1

Проверочная матрица

1	0	0	1	0	1	1
0	1	0	1	1	1	0
0	0	1	0	1	1	1

---

Результаты кодирования, определения синдрома и декодирования приведены в тексте 3.

При кодировании сообщений с кодовым расстоянием 1, получаются закодированные сообщения с кодовым расстоянием 3.

### 3.2. Циклические коды

Результат кодирования циклическим кодом приведён в тексте 4.

---

**Текст 3** Полученный результат

---

Исходное сообщение

0      1      0      1

Кодированное сообщение

1      1      0      0      1      0      1

Синдром

0      0      0

Скорректированное сообщение

1      1      0      0      1      0      1

Декодированное сообщение

0      1      0      1

---

**Текст 4** Полученный результат

---

Исходное сообщение

1      1      0      0

Кодированное сообщение

0      1      0      1      1      0      0

Декодированное сообщение

1      1      0      0

При кодировании сообщений с кодовым расстоянием 1, получаются закодированные сообщения с кодовым расстоянием 3.

### 3.3. Коды БЧХ

Результат кодирования кодом БЧХ приведён в текстах 5 - 8. Исходное сообщение показано в тексте 5. Порождающий полином приведен в тексте 6. Результат кодирования показан в тексте 7, а декодированное сообщение в тексте 8. При кодировании сообщений с кодовым расстоянием 1, получаются закодированные сообщения с кодовым расстоянием 3 или 4.

---

**Текст 5** Исходное сообщение

---

1      1      1      0      1

0      1      1      1      1

0      1      1      0      0

1      1      0      0      1

1      1      1      1      1

---

**Текст 6** Порождающий полином

---

Columns 1 through 6

1	0	1	0	0	1
---	---	---	---	---	---

Columns 7 through 11

1	0	1	1	1
---	---	---	---	---

---

---

**Текст 7** Кодированное сообщение

---

Columns 1 through 6

1	1	1	0	1	0
0	1	1	1	1	0
0	1	1	0	0	1
1	1	0	0	1	0
1	1	1	1	1	1

Columns 7 through 12

1	1	0	0	1	0
1	0	1	1	0	0
0	0	0	1	1	1
0	0	1	1	1	1
1	1	1	1	1	1

Columns 13 through 15

0	0	1
1	0	0
1	0	1
0	1	0
1	1	1

---

---

**Текст 8** Декодированное сообщение

---

1	1	1	0	1
0	1	1	1	1
0	1	1	0	0
1	1	0	0	1
1	1	1	1	1

---

### 3.4. Коды Рида-Соломона

Код Рида-Соломона позволяет кодировать не двоичные сообщения. Результаты приведены в тексте 9

При кодировании сообщений с кодовым расстоянием 1, получаются закодированные сообщения с кодовым расстоянием 3 или 4.

---

**Текст 9** Результат кодирования

---

Исходное сообщение

2	7	3
4	0	6
5	1	1

Кодированное сообщение

Columns 1 through 6

2	7	3	3	6	7
4	0	6	4	2	2
5	1	1	4	5	4

Column 7

6
0
0

Декодированное сообщение

2	7	3
4	0	6
0	7	6

Число исправленных ошибок

1
2
-1

---

## 4. Выводы

Кодирование позволяет защитить данные при передаче по каналам связи от ошибок, вызываемых как правило помехами в канале связи. Некоторые коды позволяют только обнаружить ошибку, а некоторые и обнаружить и устранить. Все коды рассмотренные в работе (Хэмминга, циклические коды, коды БЧХ, коды Рида-Соломона) позволяют устранять ошибки (являются самокорректирующимися). Рассмотренные коды Хэмминга и Циклические позволяют устранить только одну ошибки, а рассмотренные коды БЧХ и Рида-Соломона позволяют устранить до двух ошибок.



## 5. Листинг

Листинг 1: Используемый при исследовании код

```
1 function lab7()
2 close all
3 format loose
4 clc
5 out = randerr(1,4);
6 disp('Hamming')
7 fprintf('Исходное_сообщение\n');
8 disp(out);
9 code = encode(out, 7, 4, 'hamming/binary');
10 fprintf('Кодированное_сообщение\n');
11 disp(code);
12 dcode = decode(code, 7, 4, 'hamming/binary');
13 fprintf('Декодированное_сообщение\n');
14 disp(dcode)
15
16 msg = out;
17 [h, gen] = hammgen(3);
18 % gen = gen2par(h);
19 code = rem(msg * gen, 2);
20 trt = syndtable(h);
21 syndrome = rem(code * h', 2);
22
23 % error location:
24 err = bi2de(flipr(syndrome));
25 err_loc = trt(err + 1, :);
26
27 % corrected code
28 ccode = rem(err_loc + code, 2);
29
30 dcode = ccode(7 - 4 + 1:7);
31
32
33 fprintf('Порождающая_матрица\n');
34 disp(gen)
35 fprintf('Проверочная_матрица\n');
36 disp(h)
37
38 fprintf('Исходное_сообщение\n');
39 disp(msg)
40 fprintf('Кодированное_сообщение\n');
41 disp(code)
42 fprintf('Синдром\n');
43 disp(syndrome)
44 fprintf('Скорректированное_сообщение\n');
45 disp(ccode)
46 fprintf('Декодированное_сообщение\n');
47 disp(dcode)
48
49 disp('Cyclic')
50 out = randerr(1,4);
51 fprintf('Исходное_сообщение\n');
52 disp(out);
53 code = encode(out, 7, 4, 'cyclic/binary');
54 fprintf('Кодированное_сообщение\n');
55 disp(code);
56 dcode = decode(code, 7, 4, 'cyclic/binary');
```

```

57 fprintf( 'Декодированное_сообщение\n' );
58 disp(dcode)
59
60
61 m = 4;
62 n = 2^m-1;
63 k = 5;
64 nwords = 5;
65 msg = randi([0 1],nwords,k);
66 code = gf(msg);
67 [poly,t] = bchgenpoly(n,k);
68 enc = bchenc(code,n,k);
69 noisycode = enc + randerr(nwords,n,1:t);
70 dcode = bchdec(noisycode,n,k);
71
72 fprintf( 'Исходное_сообщение\n' );
73 disp(msg)
74 fprintf( 'Полином\n' );
75 disp(poly.x)
76 fprintf( 'Кодированное_сообщение\n' );
77 disp(enc.x)
78 fprintf( 'Декодированное_сообщение\n' );
79 disp(dcode.x)
80 disp( 'Rid_-_Salomon' )
81
82 m = 3;
83 n = 2^m - 1;
84 k = 3;
85
86 msg = gf([2 7 3; 4 0 6; 5 1 1],m);
87 code = rsenc(msg,n,k);
88
89 errors = gf([2 0 0 0 0 0 0; 3 4 0 0 0 0 0; 5 6 7 0 0 0 0],m);
90 noisycode = code + errors;
91
92 [dcode,cnumerr] = rsdec(noisycode,n,k);
93
94 fprintf( 'Исходное_сообщение\n' );
95 disp(msg.x)
96 fprintf( 'Кодированное_сообщение\n' );
97 disp(code.x)
98 fprintf( 'Декодированное_сообщение\n' );
99 disp(dcode.x)
100 fprintf( 'Число_исправленных_ошибок\n' );
101 disp(cnumerr)
102 end

```