

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«Омский государственный технический университет»

Кафедра «Информатика и вычислительная техника»

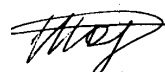
Лабораторная работа № 3 с дополнительным заданием

по дисциплине
«Сети и телекоммуникации»

Цифровое и логическое кодирование сигналов

Выполнила:

Студентка гр. ПИ-161 Шарипова М.С.



Проверил:

Старший преподаватель каф. ИВТ Звонов
А.О.



Омск, 2019

Задание

С помощью таблицы 2 (см. ниже) запишите своё ФИО и дату рождения в 16-ричном коде.

3.2 Переведите полученную запись в двоичный код (как известно, каждому символу 16-ричного кода соответствуют числа от 0 до 15 в двоичном коде).

3.3 Наглядно представьте первые 10 бит сообщения в трёх из перечисленных методов кодирования.

3.4 Рассчитайте характеристики сформированного двоичного сообщения:

- нижнюю частоту;
- максимальную частоту;
- среднюю частоту;
- ширину спектра;
- полосу пропускания.

3.5 Преобразуйте сообщение кодом 4B/5B, снова рассчитайте характеристики по п. 3.4.

3.6 Преобразуйте сообщение по п. 3.5 скремблером по п. 2.10. Снова рассчитайте характеристики по п. 3.4.

Сделайте выводы.

Дополнительное задание

Попробуйте разработать более эффективный скремблер для своего сообщения.

Результат выполнения

Исходные данные:

ШариповаМайнураСерикпаевна18011997

Исходные данные в 16-ном коде:

D8E0F0E8FEED2E0CCE0E9EDF3F0E0D1E5F0E8EAEF0E5E2EDE03138303131393937

Исходные данные в 2-ном коде:

0010110100100001001011100001100100101111000110010010111000100001001
01110001011110010111000101110001011010001101100101110000110010010110
0001011000010111000011001001011100010001000101110001011010010111100
0111000010111100011001001011100001100100101101000110100010111000011
110001011110001100100101110001000010010111000101010001011100010111
0001100100101110000111100010111000011011001011100010110100101110000
1100100011100000110100001110000100001000111000001100100011100000110
100001110000011010000111000010001000011100001000100001110000100000

(536 бит = 67 байт)

Первые 10 бит: 0010110100

1. Потенциальный код без возврата к нулю (NRZ)

Наиболее простым и очевидным методом кодирования двоичных сообщений является метод потенциального кодирования без возврата к нулю – NRZ (Non Return to Zero), в котором значению бита «1» соответствует высокий уровень потенциала, а значению «0» – низкий. Графически этот код представлен на рисунке 1.

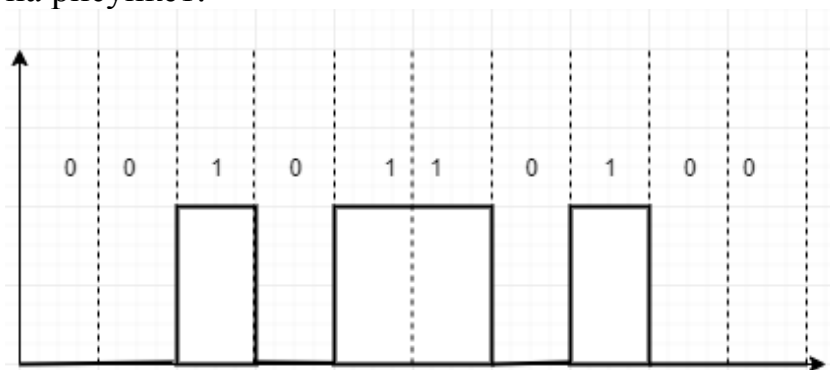


Рисунок 1. Потенциальный код без возврата к нулю

2. Биполярный импульсный код RZ

Одним из наиболее простых среди импульсных кодов является трехуровневый биполярный импульсный код с возвратом к нулю (Return to Zero, RZ), в котором единица представлена импульсом одной полярности, а ноль – импульсом другой полярности. Графически код представлен на рисунке 2.

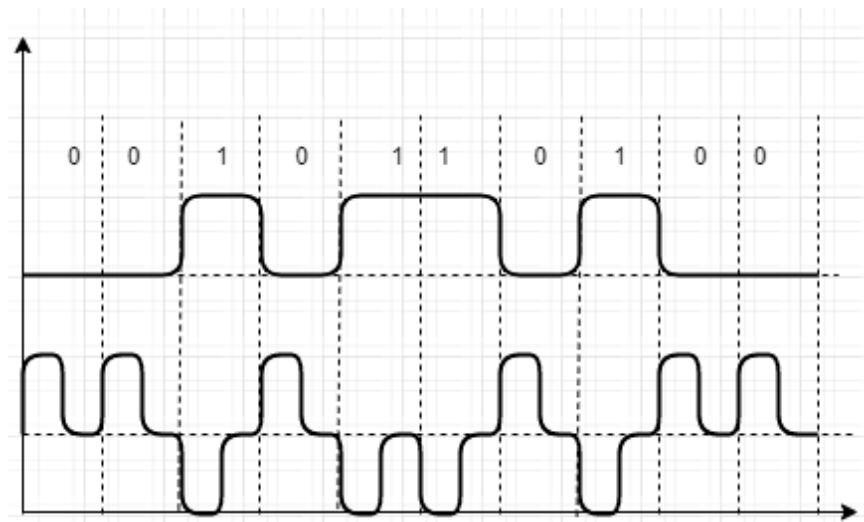


Рисунок 2. Биполярный импульсный код RZ

3.Пятиуровневый код РАМ-5

В пятиуровневом коде РАМ-5 используется 5 уровней сигнала, причем четыре уровня кодируют два бита передаваемых данных: 00, 01, 10, 11. Графически код представлен на рисунке 3.

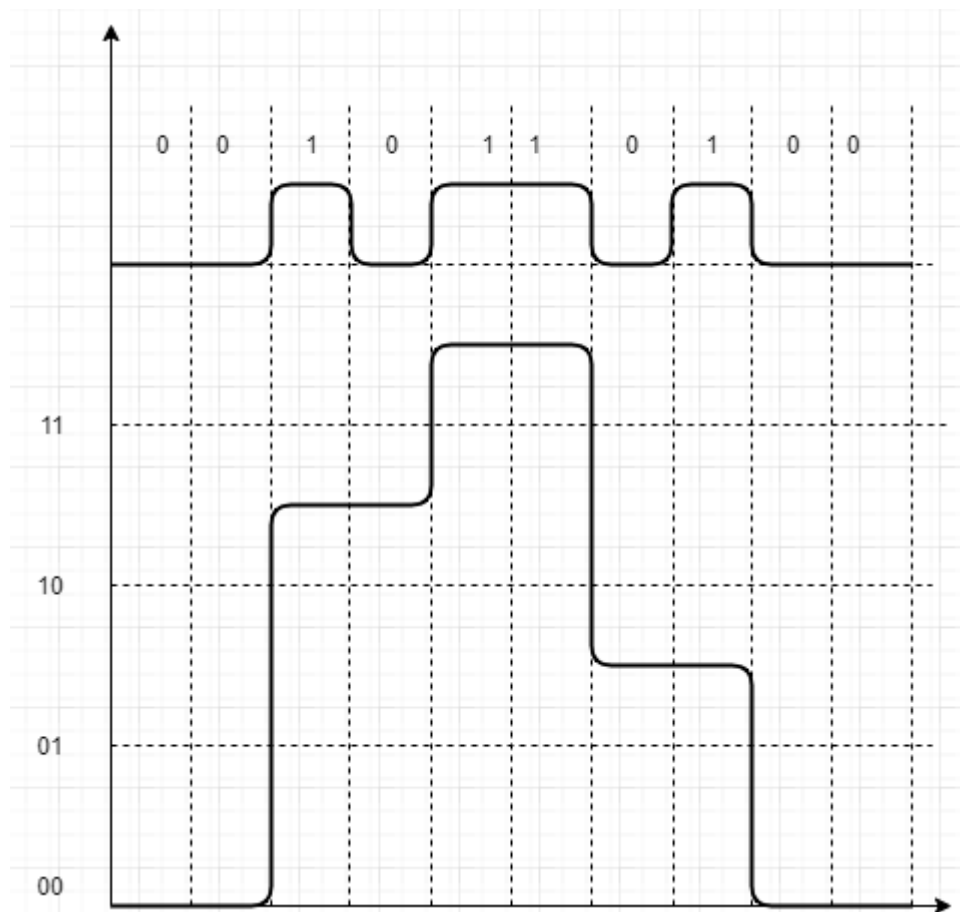


Рисунок 3. Пятиуровневый код РАМ-5

Расчет характеристик сформированного двоичного сообщения

Последовательность «0» -5

Последовательность «1» - 4

$C = 2,5$ Мбит/с

Нижняя и верхняя границы частот в передаваемом сообщении:

$$f_b = C / 2 = 2,5 / 2 = 125 \text{ кГц}$$

$$f_h = C / 9 = 2,5 / 6 = 28 \text{ кГц}$$

Полоса пропускания и ширина спектра:

$$S = f_b - f_h = 125 - 28 = 97 \text{ кГц}$$

$$F = 98 \text{ кГц}$$

Среднее значение частоты передаваемого сообщения:

Соответствующая частота сигнала для каждого битового интервала:

$$F_0 = 130$$

$$F_0 / 2 = 54$$

$$F_0 / 3 = 56$$

$$F_0 / 4 = 29$$

$$F_0 / 5 = 5$$

$$f_{cp} = (130 f_0 + 54 f_0 / 2 + 56 f_0 / 3 + 29 f_0 / 4 + 5 f_0 / 5) / 536 =$$

$$(16250 + 3375 + 2333 + 906 + 125) / 536 = 43 \text{ кГц}$$

5. Преобразование сообщения кодом 4B5B

До :

```
0010110100100001001011100001100100101111000110010010111000100001001
01110001011110010111000101110001011010001101100101110000110010010110
0001011000010111000011001001011100010001000101110001011010010111100
0111000010111100011001001011100001100100101101000110100010111000011
110001011110001100100101110001000010010111000101010001011100010111
0001100100101110000111100010111000011011001011100010110100101110000
1100100011100000110100001110000100001000111000001100100011100000110
100001110000011010000111000010001000011100001000100001110000100000
```

(536)

Длина сообщения: 67 байт (536 бит)

После:

```
1010011011101000100110100010011010011101010011001110100111001010010
00110100111001000111001110011100101001101101001101111010011100010011
0011101011101010100110101010011100010011001110100111001010010100101
00111001010011011101001101110100111010100111010101001110101001100111
01001110001001100111010011011010011011010100111000100111100101001110
0010011110010100111010100110011101001010011100101000100110100111001
0100101001010010100111001010011101010011001110100111000100111100101
```

00111000100110111101001111001010011011101001110001001100111001111010
 01001101011100001001101111001101001111000100110110010011101001001101
 10010011101101001010010100111100101001010001001110101010011110 (672)

Длина сообщения: 84 байт (672бит)

Избыточность: $(84-67)/67*100\%=25\%$

6.Расчет характеристик сформированного преобразованного двоичного сообщения кодом 4B/5B

Первые 10 бит: 1010011011

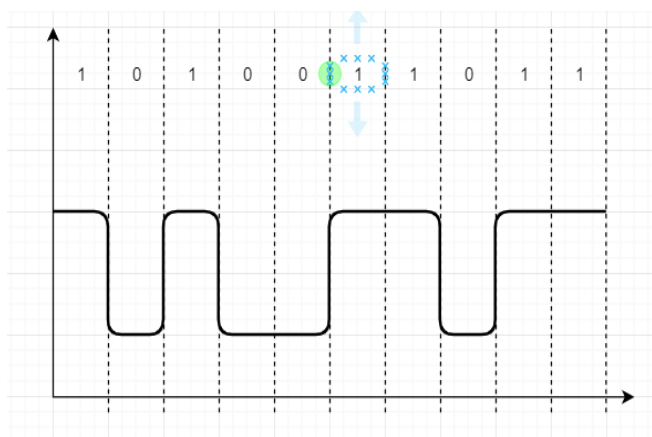


Рисунок 4.Графическое изображение 10 бит после кодирования 4B/5B

7.Расчет характеристик сформированного двоичного сообщения после преобразования 4B/5B

Последовательность «0» -4

Последовательность «1» - 4

$C = 2,5$ Мбит/с

Нижняя и верхняя границы частот в передаваемом сообщении:

$$f_b = C / 2 = 2,5 / 2 = 125 \text{ кГц}$$

$$f_n = C / 8 = 2,5 / 8 = 31 \text{ кГц}$$

Полоса пропускания и ширина спектра:

$$S = f_b - f_n = 125 - 31 = 94 \text{ кГц.}$$

$$F = 95 \text{ кГц}$$

Среднее значение частоты передаваемого сообщения:

Соответствующая частота сигнала для каждого битового интервала:

$$F_0 = 172$$

$$F_0/2 = 101$$

$$F_0/3 = 55$$

$$F_0/4 = 10$$

$$F_0/5 = 0$$

$$f_{cp} = (172f_0 + 101f_0/2 + 55f_0/3 + 10f_0/4) / 672 = (21500 + 6312 + 2255 + 312) / 672 = 45.2 \text{ кГц}$$

8. Преобразование сообщения скремблером

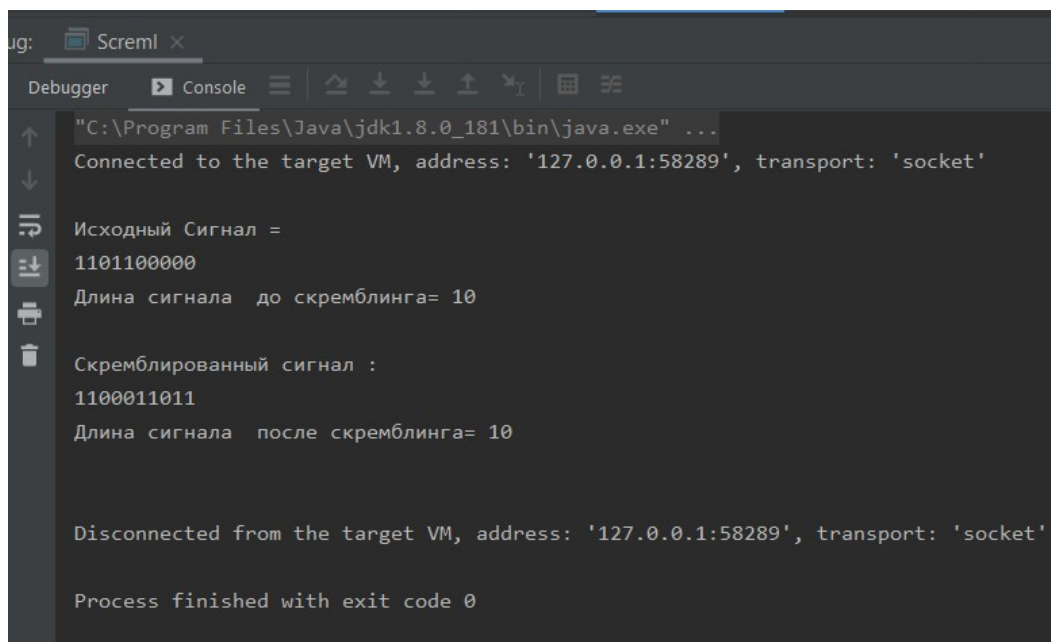
Скремблирование считалось программой.

Правильность работы программы проверялась на примере из пособия.

Пусть исходное сообщение включает последовательность из 5 нулей: 1101100000

Результирующее сообщение не будет содержать длинных последовательностей: 1100011011.

Результат выполнения на рисунке 5.



```

Debugger Console
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Connected to the target VM, address: '127.0.0.1:58289', transport: 'socket'

Исходный Сигнал =
1101100000
Длина сигнала до скремблинга= 10

Скремблированный сигнал :
1100011011
Длина сигнала после скремблинга= 10

Disconnected from the target VM, address: '127.0.0.1:58289', transport: 'socket'

Process finished with exit code 0

```

Рисунок 5. Результат проверки правильности работы программы скремблирования

На Рисунке 6 изображен результат скремблирования своего сообщения (4В/5В) длиной 672 символа. Полное сообщение после скремблинга представлено выше.

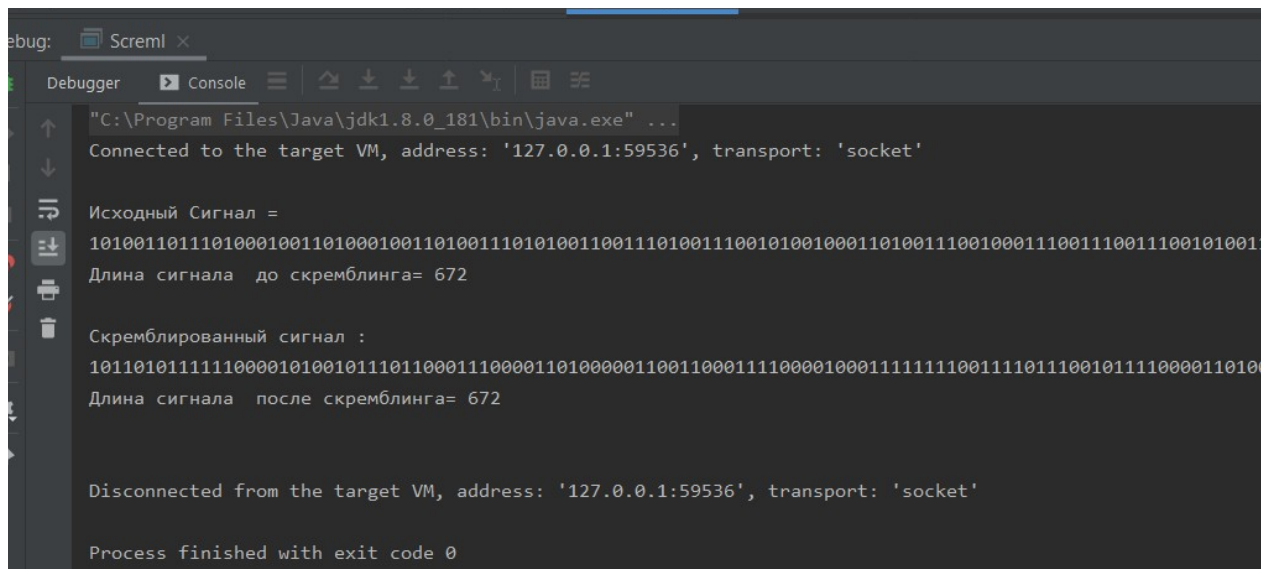


Рисунок 6. Результат скремблирования сообщения после 4В/5В.

Код программы

```
import java.util.ArrayList;

public class Screml {

    public static void main(String[] args) {

        // int intArr[]={1,1,0,1,1,0,0,0,0,0};
        int count=0;
        String str= "1010011011101000100110100010" +
            "0110100111010100110011101001110010100100011010" +
            "011100100011100111001110010100110110100110111101001110001001" +
            "1001110101110101010011010101001110001001100111010011100101001010" +
            "0101001110010100110111010011011101001110101001110101010011101010011100" +
            "11101001110001001100111010011011010011011010100111000100111100101001110001001"
            +
            "111001010011101010011001110100101001110010100010011010011100101001010010100101
            001110010100111" +
            "010100110011101001110001001111001010011100010011011110100111100101001101110100
            11100010011001110011110100100110101" +

            "110000100110111100110100111100010011011001001110100100110110010011101101001010
            01010011110010100101000100111010101000111110";

        String items[] = str.split("");
        int intArr[] = new int[items.length];
        for(int i=0;i<items.length;i++){

            intArr[i] = Integer.parseInt(items[i]);

        }

        System.out.println(" \nИсходный Сигнал = " );
        for(int i=0;i<intArr.length;i++) {
            System.out.print( intArr[i]);
```



```

    }

    System.out.println("\nДлина сигнала до скремблинга= "+intArr.length );

    System.out.println("\nСкремблированный сигнал : ");
    for ( int i=0;i <intArr.length; i++) {

        if(i<3 ){
            intArr[i] ^=0;
        }
        else if (i<5){
            intArr[i]^=intArr[i-3]^0;
        }
        else {

            intArr[i]^= intArr[i - 3] ^ intArr[i - 5];
        }
        count++;
        System.out.print(intArr[i] );
    }
    System.out.println("\nДлина сигнала после скремблинга= "+ count );

    System.out.println("\n" );
}
}

```

Начальные данные после кодирования 4В/5В:

```

10100110111010001001101000100110100111010100110011101001110010100
1000110100111001000111001110011100101001101101001101110100111000100
1100111010111010101001101010100111000100110011101001110010100101001
0100111001010011011101001101110100111010100111010101001110101001100
11101001110001001100111010011011010011011010100111000100111100101001
1100010011110010100111010100110011101001010011100101000100110100111
0010100101001010010100111001010011101010011001110100111000100111100
10100111000100110111101001111001010011011101001110001001100111001111
0100100110101110000100110111100110100111100010011011001001110100100
1101100100111011010010100101001111001010010100010011101010100011111
0

```

Длина=672 символа

После скремблинга:

1011010111110000101001011101100011100001101000001100110001111000010
00111111110011110111001011110000110100010000111101100011100011111110
0100111001101011000011001011110001001101101011000000010110110010100
01011011001001100101111111101011111101011111101000001100101111101100
01110001111111001001111010010010000010111011011100010101001001000110
10101111101000001100101111101100011110000010000000010001111111100110
11000111100001101001010010101101100100100011111111000010101100010011
00111010010001000011100010000111101111101100110110111101011110011101
0010011001111000000111010010010001001001000001000011101110001011111
01011111110101101111110010000110011010100101011011000110000010

Длина=672 символа

8.Расчет характеристик сформированного двоичного сообщения после скремблирования

Последовательность «0» -8

Последовательность «1» - 8

$C = 2,5$ Мбит/с

Нижняя и верхняя границы частот в передаваемом сообщении:

$$f_b = C / 2 = 2,5 / 2 = 125 \text{ кГц}$$

$$f_n = C / 16 = 2,5 / 16 = 16 \text{ кГц}$$

Полоса пропускания и ширина спектра:

$$S = f_b - f_n = 125 - 16 = 109 \text{ кГц.}$$

$$F = 110 \text{ кГц}$$

Среднее значение частоты передаваемого сообщения:

$$F_0 = 145$$

$$F_0/2 = 86$$

$$F_0/3 = 38$$

$$F_0/4 = 23$$

$$F_0/5 = 10$$

$$F_0/6 = 6$$

$$F_0/7 = 4$$

$$F_0/8 = 4$$

$$f_{cp} = (145f_0 + 86f_0/2 + 38f_0/3 + 23f_0/4 + 10f_0/5 + 6f_0/6 + 4f_0/7 + 4f_0/8) / 672 = \\ (18125 + 5375 + 1583 + 719 + 250 + 125 + 71,4 + 62,5) / 672 = 39 \text{ кГц}$$

Вывод

В ходе выполнения лабораторной работы были выполнены все пункты задания, а именно, ФИО и дата рождения записаны в 16-чном коде с помощью таблицы 2 в пособии, затем из 16-чного кода запись преобразована в двоичный код, 10 первых бит данного сообщения наглядно представлены с помощью трёх методов кодирования, таких как, потенциальный код без возврата к нулю (NRZ), биполярного импульсного кода (RZ), пятиуровневого кода PAM-5. Также после этого, были рассчитаны характеристики сформированного двоичного сообщения. Затем, исходное сообщение было преобразовано кодом 4В/5В и для уже преобразованного сообщения также были рассчитаны характеристики двоичного сообщения.

В заключении, можно сделать вывод, что при каждом кодировании изменяется числовые характеристики нижней границы частот в передаваемом сообщении (28,31,16 кГц). Наименьшая частота получается при скремблировании. Что касается средней частоты передаваемого сообщения, то при различных методах кодирования, она получилась разной, но отличия небольшие, а именно 43кГц(простое кодирование), 45кГц(4В/5В) и 39кГц (Скремблирование). Исходя из результата скремблирования, можно увидеть что, скремблирование гарантирует полную синхронизацию в линии передач. Скремблирование выполнено как написано в дополнительном задании.