Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Учебно-исследовательская работа №1 (УИР 1)**

**«Кодирование данных в телекоммуникационных системах»** по дисциплине “Телекоммуникационные системы”

Выполнили:

Студенты группы P3334

Баянов Р. Д.

Преподаватель:

Алиев Т. И.

Санкт-Петербург

2024 г.

**Оглавление**

[**Формирование сообщения** 3](#_Toc179759768)

[**Физическое кодирование исходного сообщения** 4](#_Toc179759769)

[**Манчестерский код** 4](#_Toc179759770)

[**Потенциальный код без возврата к нулю (NRZ – Non Return to Zero)** 6](#_Toc179759771)

[**Биполярный импульсный код (RZ – Return to Zero)** 8](#_Toc179759772)

[**Биполярное кодирование с альтернативной версией (AMI)** 10](#_Toc179759773)

[**Потенциальный код с инверсией при единице (NRZI)** 12](#_Toc179759774)

[**Пятиуровневый код PAM-5 (2B1Q)** 13](#_Toc179759775)

[**Сравнительный анализ методов физического кодирования** 15](#_Toc179759776)

[**Логическое кодирование исходного сообщения** 17](#_Toc179759777)

[**Избыточное кодирование (4B/5B)** 17](#_Toc179759778)

[**Скремблирование** 19](#_Toc179759779)

[**Сравнительный анализ методов логического кодирования** 22](#_Toc179759780)

[**Вывод** 23](#_Toc179759781)

**Формирование сообщения**

* **Исходное сообщение:** БаРД
* **В шестнадцатеричном коде:** C1 E0 D0 C4
* **В двоичном коде:** 1100 0001 1110 0000 1101 0000 1100 0100
* **Длина сообщения:**  4 байт (32 бит)

**Физическое кодирование исходного сообщения**

Рассмотрим кодирование данного сообщения 6 разными способами, а именно:

* Манчестерский код
* Потенциальный код без возврата к нулю
* Биполярный импульсный код
* Биполярное кодирование с альтернативной версией
* Потенциальный код с инверсией при единице
* Пятиуровневый код PAM-5

## **Манчестерский код**

Временная диаграмма кодирования первых 4 байт:

Изображение выглядит как снимок экрана, линия, черный, Прямоугольник

Автоматически созданное описание

Временная диаграмма кодирования первых 2 байт:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Расчёт значений данного кодирования:

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  ;

Отсюда можно сделать вывод, что манчестерский код очень даже неплох, так как не имеет постоянной составляющей и обладает способностью самосинхронизации. Но он уступает NRZ в спектре сигнала в общем случае, потому что больше, чем у потенциального. Также манчестерский имеет всего два уровня потенциала, что делает реализацию данного кодирования не очень дорогой.

## **Потенциальный код без возврата к нулю (NRZ – Non Return to Zero)**

Временная диаграмма кодирования первых 4 байт:

Изображение выглядит как снимок экрана, черный, линия, текст

Автоматически созданное описание

Временная диаграмма кодирования первых 2 байт:

Изображение выглядит как снимок экрана, Шрифт, текст, линия

Автоматически созданное описание

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

**Середина спектра: следовательно в спектре сигнала преобладают низкие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**

Отсюда можно сделать вывод, что потенциальный код уязвим к шумам и не обладает способностью распознавания ошибок. Также NRZ обладает постоянной составляющей и не имеет синхронизации. Но NRZ неплох в том, что в общем случае имеет относительно небольшой спектр сигнала. Также потенциальный код легко реализовать, и он имеет неплохую скорость передачи. В классическом виде на практике не используется.

## **Биполярный импульсный код (RZ – Return to Zero)**

Временная диаграмма кодирования первых 4 байт:

Изображение выглядит как снимок экрана, линия, черный, текст

Автоматически созданное описание



Временная диаграмма кодирования первых 2 байт:

Изображение выглядит как снимок экрана, текст, Шрифт, дизайн

Автоматически созданное описание

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  ;

Отсюда делаем вывод, что код RZ, несмотря на самосинхронизацию и отсутствие постоянной составляющей, он имеет три уровня потенциала, что удорожает реализацию и имеет крайне большой спектр сигнала.

## **Биполярное кодирование с альтернативной версией (AMI)**

Временная диаграмма кодирования первых 4 байт:

Изображение выглядит как снимок экрана, текст, черный, линия

Автоматически созданное описание



Временная диаграмма кодирования первых 2 байт:

Изображение выглядит как снимок экрана, текст, Шрифт, дизайн

Автоматически созданное описание

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**

**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**

Отсюда сделаем вывод, что AMI крайне плох для нашего сообщения, он обладает достаточно широким спектром, страдает от постоянных составляющих, не имеет самосинхронизации, да и к тому же имеет 3 уровня потенциала. У AMI есть только один плюс в нашем случае – это обнаружение ошибок, благодаря чередованию единиц.

## **Потенциальный код с инверсией при единице (NRZI)**

Временная диаграмма кодирования первых 4 байт:

Изображение выглядит как снимок экрана, черный, текст, линия

Автоматически созданное описание

Временная диаграмма кодирования первых 2 байт:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**

**Середина спектра: следовательно в спектре сигнала преобладают низкие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**

Неплохой способ передачи сообщения, но постоянные составляющие, отсутствие самосинхронизации и спектр шире, чем у NRZ, не делает этот метод полезным для нас.

## **Пятиуровневый код PAM-5 (2B1Q)**

Временная диаграмма кодирования первых 4 байт:

Изображение выглядит как снимок экрана, Прямоугольник, линия, черный

Автоматически созданное описание



Временная диаграмма кодирования первых 2 байт:

Изображение выглядит как снимок экрана, текст, Шрифт, Прямоугольник

Автоматически созданное описание

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**

**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**

Данный метод отличился скоростью передачи сообщения, а именно потому что он обладает невероятно узким спектром. Но 5 уровней потенциала делает реализацию этого метода затратной по ресурсам. Этот метод обладает возможность обнаружения ошибок и в нашем сообщении не имеет частых и длинных постоянных составляющих, но он всё ещё проигрывает манчестерскому и RZ.

## **Сравнительный анализ методов физического кодирования**

Временная диаграмма всех методов кодирования первых 4 байт для сравнения:

Изображение выглядит как снимок экрана, прямоугольный, астрономия

Автоматически созданное описание

Таблица сравнения всех методов:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Manchester |  | Есть | Нет | Нет | 2 |
| NRZ |  | Нет | Есть | Нет | 2 |
| RZ |  | Есть | Нет | Есть | 3 |
| AMI |  | Нет | Есть | Есть | 3 |
| NRZI |  | Нет | Есть | Нет | 2 |
| PAM-5 |  | Нет | Есть | Есть | 5 |

Параметры сравнения:

1. Спектр сигнала
2. Самосинхронизация
3. Постоянная составляющая
4. Обнаружение ошибок
5. Стоимость реализации

Теперь определим два наилучших физических метода кодирования для данного сообщения.

* Сразу же отбросим код AMI, так как мало того, что он является дорогим из-за 3 уровней потенциала, не является самосинхронизирующимся, он ещё и обладает самый широким спектром сигнала среди кодов NRZI и NRZ. К тому же у AMI нет способности обнаружения ошибок. Наше сообщение обладает большим количество участков с идущими нулями подряд, поэтому AMI наихудший выбор для нашего сообщения.
* Также отбросим коды NRZ и NRZI, так как они хоть и обладают не очень широким спектром сигнала, они сильно проигрывают другим кодам из-за наличия постоянной составляющей и отсутствия самосинхронизации.
* Метод PAM-5 является неплохим вариантом, так как он обладает очень быстрой скоростью передачи засчёт самого узкого спектра сигнала из всех представленных методов. Также он обладает способностью к обнаружению ошибок, но он очень дорог в реализации, и страдает из-за постоянных составляющих в нашем сообщении. Также не имеет самосинхронизации.
* Остаются два хороших варианта, а именно RZ и манчестеркий код. Они имеют самосинхронизацию, не имеют постоянных составляющих и не дороги в реализации. Также RZ имеет способность к обнаружению ошибок.
* Но нетрудно заметить, что манчестерский код по сравнению с RZ имеет более узкий спектр сигнала. Именно поэтому манчестерский код лучший вариант для нашего сообщения.

Таким образом два лучших физических метода кодирования для данного сообщения — это RZ и манчестерский код.

**Логическое кодирование исходного сообщения**

## **Избыточное кодирование (4B/5B)**

Воспользуемся таблицей перекодировки:

|  |  |  |  |
| --- | --- | --- | --- |
| Исходные символы | Результирующие символы | Исходные символы | Результирующие символы |
| 0000 | 11110 | 1000 | 10010 |
| 0001 | 01001 | 1001 | 10011 |
| 0010 | 10100 | 1010 | 10110 |
| 0011 | 10101 | 1011 | 10111 |
| 0100 | 01010 | 1100 | 11010 |
| 0101 | 01011 | 1101 | 11011 |
| 0110 | 01110 | 1110 | 11100 |
| 0111 | 01111 | 1111 | 11101 |

**Исходное сообщение:** *A =* 1100 0001 1110 0000 1101 0000 1100 0100

**Результат:** *B =* 11010 01001 11100 11110 11011 11110 11010 01010

**Шестнадцатеричный код:** D2 79 ED FB 4A

**Длина сообщения:** 40 бит (5 байт)

Избыточность 25%

Временная диаграмма для манчестерского кода, модифицированного с помощью избыточного кодирования:

Изображение выглядит как пространство, черный, линия, снимок экрана

Автоматически созданное описание

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

**Середина спектра: следовательно в спектре сигнала не преобладают ни высокие, ни низкие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  ;

Так как избыточное кодирование применяется для модификации физического кодирования рассмотрим, как манчестерский код улучшился. Так как мы использовали манчестерский код спектр сигнала никак не изменился, но мы получили возможность обнаруживать ошибки в передаче сообщения, благодаря запрещённым кодировкам. Но в целом избыточное кодирование никак сильно не повлияло на качество кодирования.

## **Скремблирование**

**Исходное сообщение:** *A =* 1100 0001 1110 0000 1101 0000 1100 0100

Полином для скремблирования:

Выбор обосновывается тем, что первые три символа это 110, поэтому не будет постоянной составляющей и такой выбор выгоден для нас.

**Результат:** *B =* 1101 1100 1001 0110 1011 0011 0011 1010

**Шестнадцатеричный код:** DC 96 B3 3A

Временная диаграмма для манчестерского кода, модифицированного с помощью скремблирования:

**Изображение выглядит как снимок экрана, линия, дизайн

Автоматически созданное описание**

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  ;

Отсюда мы модем сделать вывод, что скремблирование и вправду помогло избавиться от длинных последовательностей нулей и единиц, и если бы мы использовали какой-либо код, находящийся в зоне риска постоянных составляющих, то скремблирование нам бы очень помогло. Но так как мы выбрали скремблировать манчестерский код, то скремблирование не дало нам никакой выгоды.

## **Сравнительный анализ методов логического кодирования**

Таблица сравнения методов кодирования:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Избыточное |  | Есть | Нет | Есть | 2 |
| Скремблирование |  | Есть | Нет | Нет | 2 |

Параметры сравнения:

1. Спектр сигнала
2. Самосинхронизация
3. Постоянная составляющая
4. Обнаружение ошибок
5. Стоимость реализации

Таким образом, можем сделать вывод, что избыточное кодирование для модификации манчестерского кода намного эффективнее, чем скремблирование, так как скремблирование не несёт в себе никакой практической пользы для манчестерского кодирования и конкретно для нашего сообщения. Скремблирование к тому же затрачивает много усилий на реализацию алгоритмов скремблирования и десклемблирования. Избыточное же кодирование реализуется крайне просто и добавляет возможность обнаруживать ошибки засчёт запрещённых кодировок.

**Вывод**

Выполнив данную лабораторную работу, мы изучили большинство методов кодирования сообщений. Отталкиваясь от данного вариантом сообщения, мы выяснили, что в нашем случае нам подходят манчестерский и RZ физические кодирования. Так как они обладают неплохой скоростью передачи, у них отсутствуют постоянные составляющие, также они крайне просты в реализации, и к тому же у RZ есть возможность обнаруживать ошибки засчёт возвращения сигнала к нулевому потенциалу. Просмотрев на логическое кодирование, я заметил, что оно сильно помогает в улучшении передачи сообщения. Но модифицируя манчестерский код нашего сообщения, от скремблирования мы не получаем полезных итогов. Но избыточное кодирование даёт манчестерскому коду возможность обнаружения ошибок. Но логическое кодирование сильно снижает скорость передачи из-за избыточности в 25%. Именно поэтому я считаю логическое кодирование будет лишним для нашего сообщения и его стоит передавать обычным манчестерским кодом. Но если бы мы модифицировали, например, код NRZ с помощью скремблирования мы вполне могли бы получить замечательный вариант кодирования сообщения, так как оно спасло бы нас от постоянных составляющих, что так мешают NRZ быть неплохим способом кодирования.