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 P3332

Антипин Г. В.

Преподаватель:

Алиев Т. И.

Санкт-Петербург

2025 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)

[**Дифференциальный манчестерский код** 8](#_Toc179759772)

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

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

[**Сравнительный анализ методов физического кодирования** 14](#_Toc179759776)

[**Логическое кодирование исходного сообщения** 15](#_Toc179759777)

[**Избыточное кодирование (4B/5B)** 15](#_Toc179759778)

[**Скремблирование** 17](#_Toc179759779)

[**Сравнительный анализ методов логического кодирования** 20](#_Toc179759780)

[**Вывод** 21](#_Toc179759781)

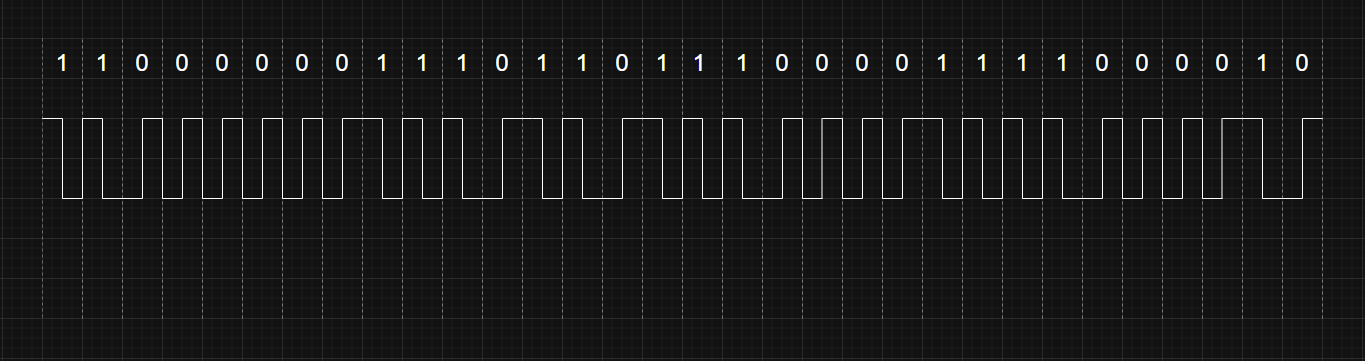
**Формирование сообщения**

* **Исходное сообщение:** АнГВ
* **В шестнадцатеричном коде:** С0 ED C3 C2
* **В двоичном коде:** 1100 0000 1110 1101 1100 0011 1100 0010
* **Длина сообщения:**  4 байт (32 бит)

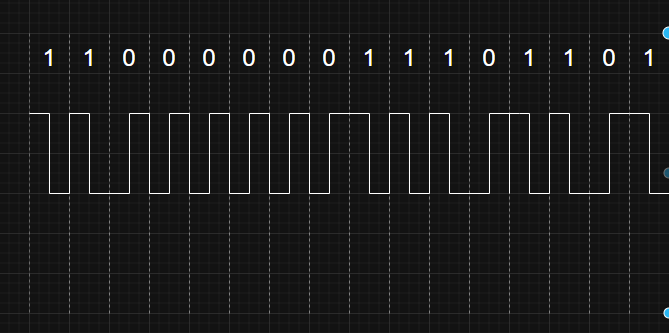
**Физическое кодирование исходного сообщения**

## **Манчестерский код**

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт значений данного кодирования:

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

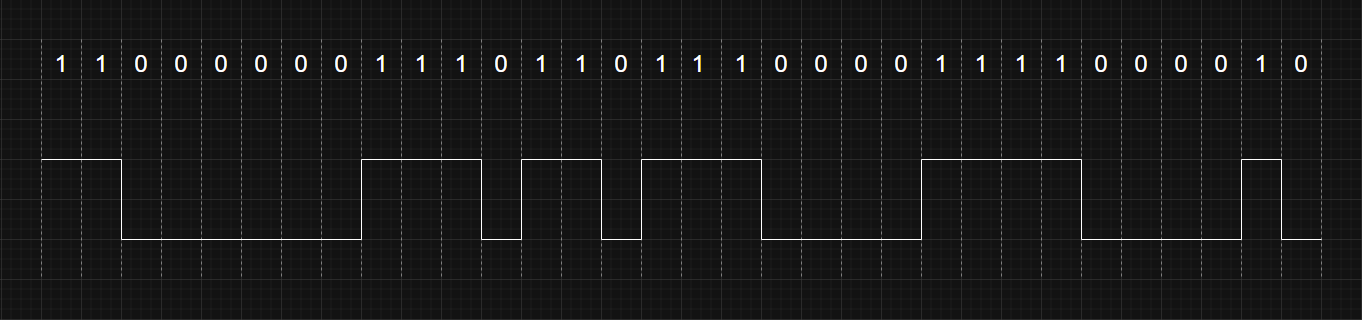
**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

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

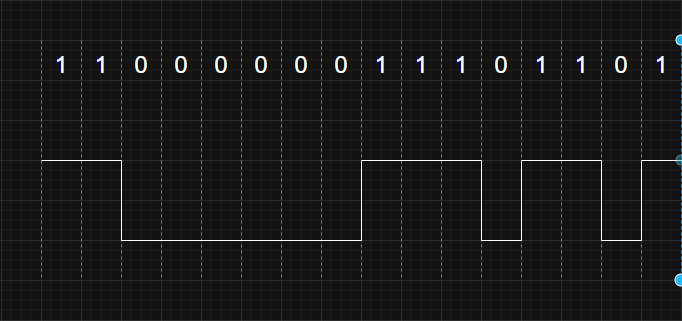
Отсюда можно сделать вывод, что манчестерский код очень даже неплох, так как не имеет постоянной составляющей и обладает способностью самосинхронизации. Но он уступает NRZ в спектре сигнала в общем случае, потому что больше, чем у потенциального. Также манчестерский имеет всего два уровня потенциала,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



Временная диаграмма кодирования первых 2 байт:



**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

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

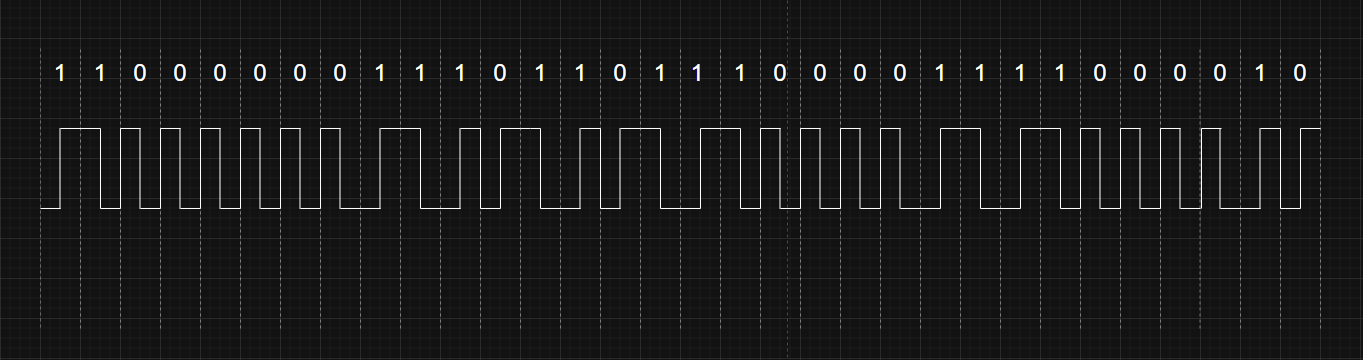
**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**

При NRZ-кодировании сигнал сохраняет постоянный уровень при передаче длинных серий одинаковых бит.  
Это вызывает появление постоянной составляющей и смещение спектра в область низких частот.  
Средняя частота спектра меньше середины диапазона, что подтверждает преобладание низкочастотных компонентов.  
Метод прост в реализации, но не обеспечивает самосинхронизации и плохо работает при длинных сериях нулей или единиц.

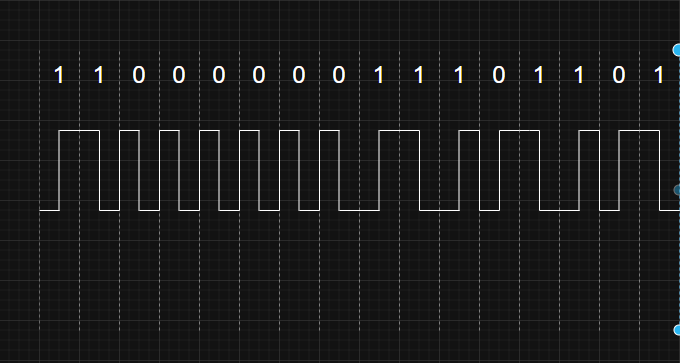
## 

## **Дифференциальный манчестерский код**

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт значений данного кодирования:

**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**  ;

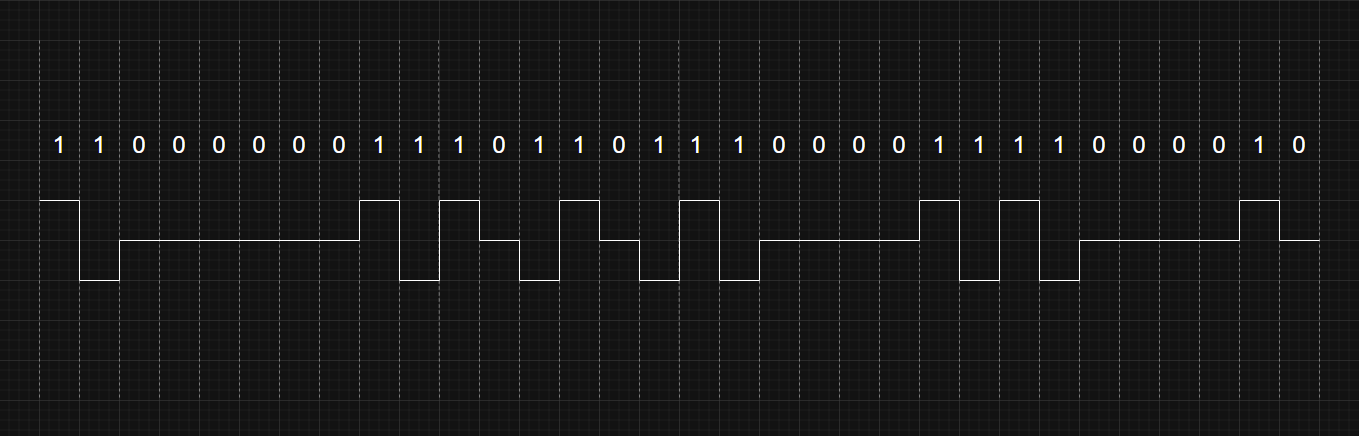
**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

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

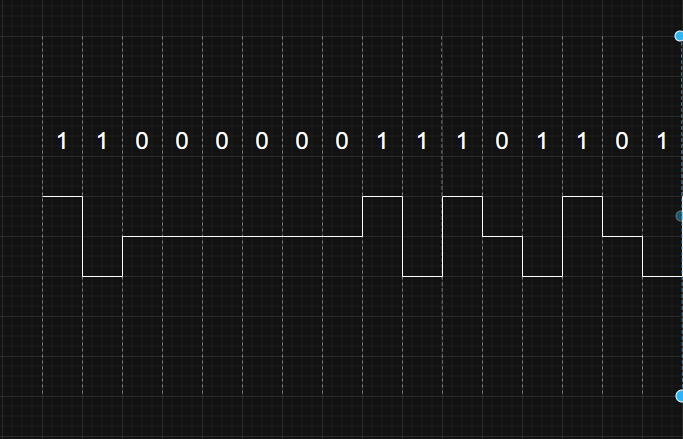
Обеспечивает полное отсутствие постоянной составляющей и стабильную синхронизацию при любой последовательности бит.

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

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



**Расчёт верхней границы:** ; ; ; ;

**Расчёт нижней границы:** ;

**Следовательно спектр сообщения:** ;

**Средняя частота спектра:**

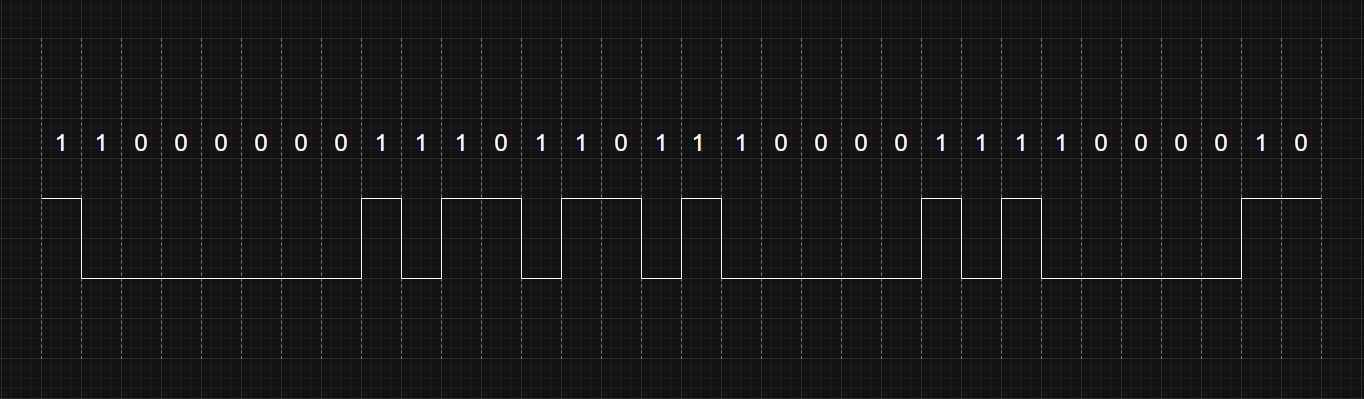
**Середина спектра: следовательно в спектре сигнала преобладают высокие частоты.**

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

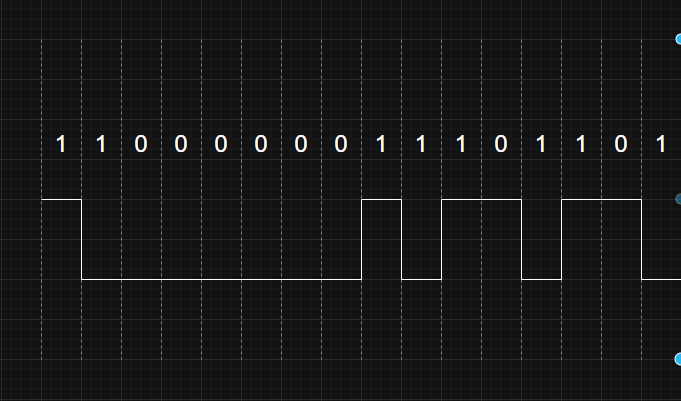
Отсюда сделаем вывод, что AMI крайне плох для нашего сообщения, он обладает достаточно широким спектром, страдает от постоянных составляющих, не имеет самосинхронизации, да и к тому же имеет 3 уровня потенциала. У AMI есть только один плюс в нашем случае – это обнаружение ошибок, благодаря чередованию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

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

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



**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:**

При передаче длинных серий нулей уровень сигнала не изменяется, что вызывает появление постоянной составляющей и потерю синхронизации.  
При чередующихся битах сигнал формирует частоты порядка C/2, как и NRZ-код.  
Преимущество метода — простая реализация и меньшая чувствительность к инверсии уровня сигнала.

## **Сравнительный анализ методов физического кодирования**

Временная диаграмма всех методов кодирования первых 4 байт для сравнения:

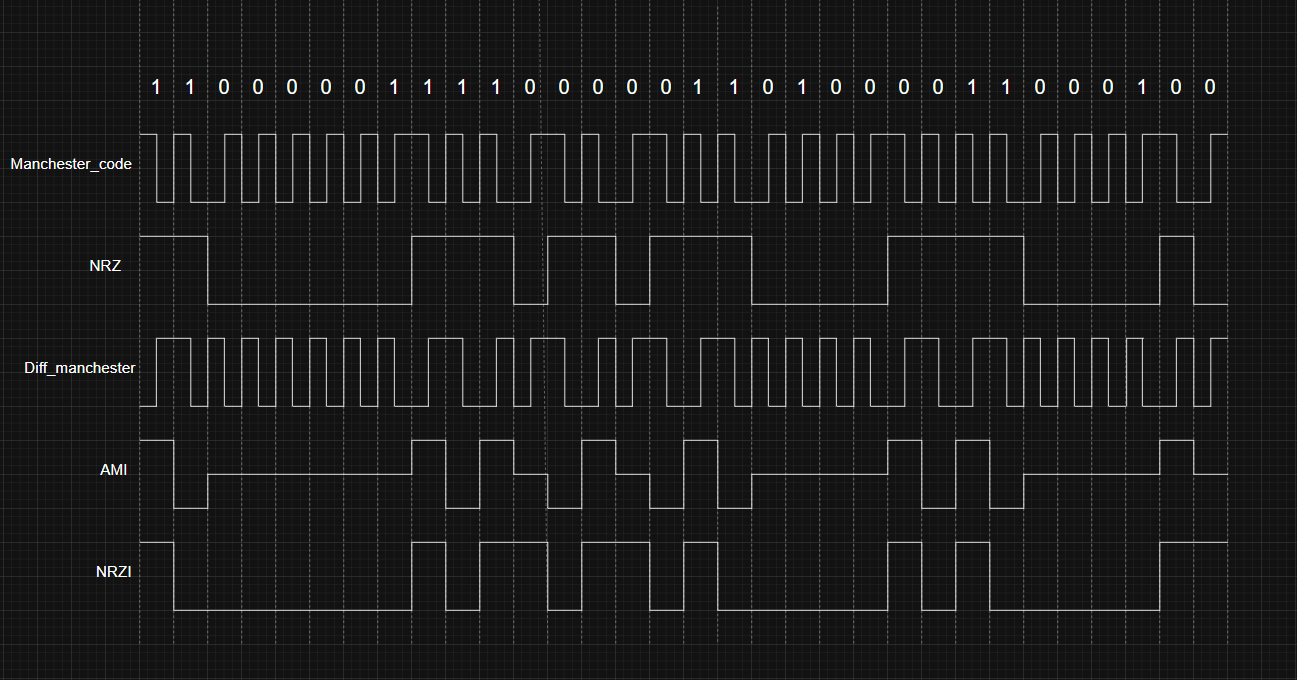


Таблица сравнения всех методов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| Manchester |  | Есть | Нет | Нет |
| NRZ |  | Нет | Есть | Нет |
| Diff manchester |  | Есть | Нет | нет |
| AMI |  | Нет | Есть | Есть |
| NRZI |  | Нет | Есть | Нет |

Параметры сравнения:

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

Теперь определим два наилучших физических метода кодирования для данного сообщения.

NRZ и NRZI – теряют переходы при длинных нулях -> не подходит  
AMI – дает провалы синхронизации при длинных нулях -> не подходит  
Манчестерский и дифф манчестерский – самосинхронизируются и устойчивы к длинным сериям нулей –> лучше всего подходит

**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 0000 1110 1101 1100 0011 1100 0010

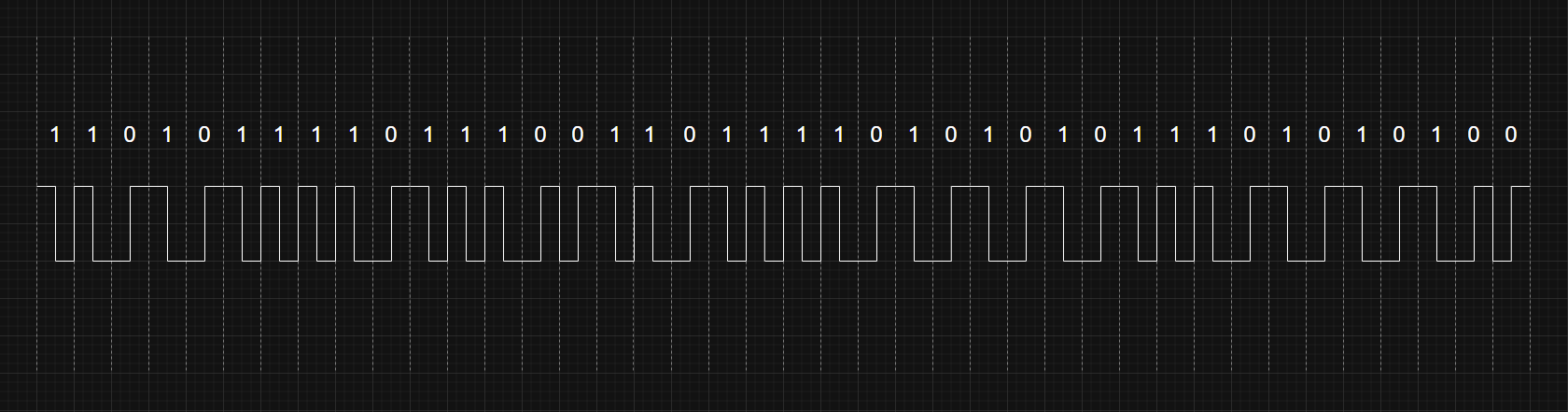
**Результат:** *B =* 11010 11110 11100 11011 11010 10101 11010 10100

**Шестнадцатеричный код:** D7 B9 BD 57

**Длина сообщения:** 40 бит (5 байт)

Избыточность 25%

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



**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:**  ;

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 0000 1110 1101 1100 0011 1100 0010

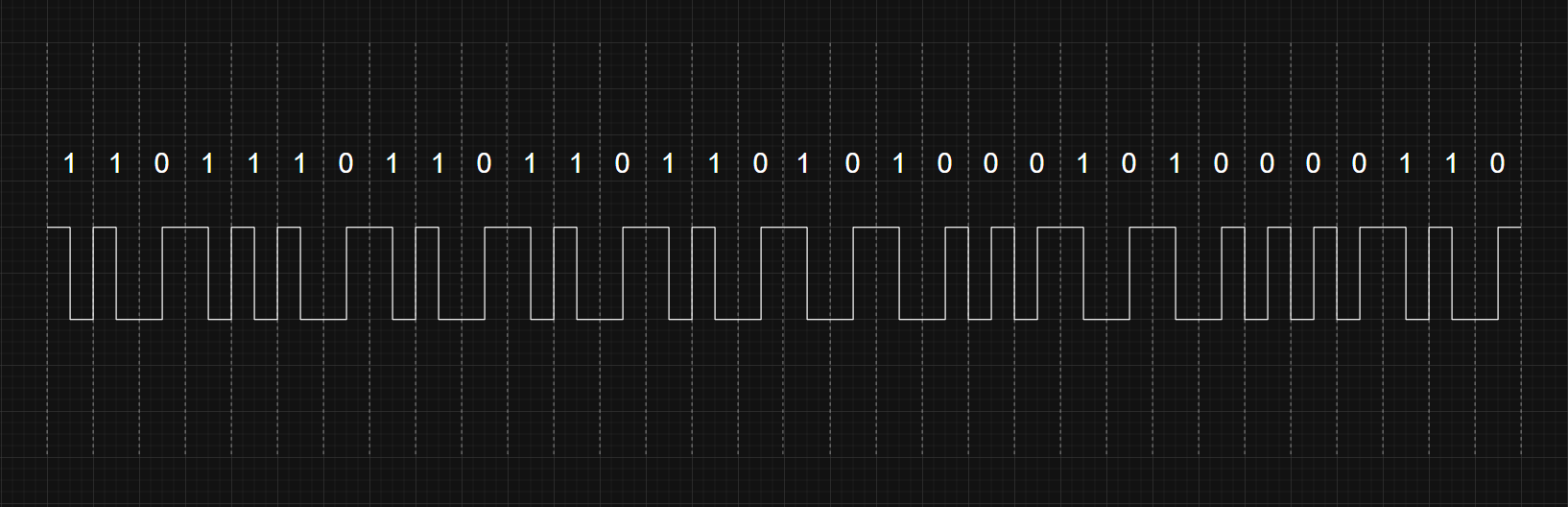
Полином для скремблирования:

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

**Результат:** *B =* 1101 1101 1011 0110 1010 0010 1000 0110

**Шестнадцатеричный код:** DD B6 A2 86

Временная диаграмма для манчестерского кода, модифицированного с помощью скремблирования:

****

**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, можно заключить, что избыточное кодирование оказывается значительно более эффективным для модификации манчестерского кода, чем скремблирование.  
Скремблирование не приносит практической пользы для манчестерского кодирования и конкретно для данного сообщения, а также требует заметных затрат на реализацию как скремблера, так и дескремблера.  
В то же время избыточное кодирование отличается простотой реализации и позволяет обнаруживать ошибки благодаря использованию запрещённых комбинаций.

**Вывод**

В ходе лабораторной работы мы изучили основные методы кодирования: NRZ, NRZI, AMI, Манчестерское и Дифференциальное манчестерское.  
Анализ показал, что для нашего сообщения лучше всего подходят манчестерское и дифференциальное манчестерское кодирования — они обеспечивают надёжную синхронизацию, не имеют постоянной составляющей и просты в реализации.

Методы NRZ и NRZI теряют синхронизацию при длинных сериях одинаковых бит, а AMI хотя и убирает постоянную составляющую, уступает по устойчивости к ошибкам.  
Скремблирование в манчестерском коде не даёт пользы, так как он уже самосинхронизируется, зато избыточное кодирование позволяет обнаруживать ошибки.

Однако избыточность снижает скорость передачи, поэтому оптимально передавать сообщение обычным манчестерским или дифференциальным манчестерским кодом.  
Для NRZ же скремблирование было бы полезным — оно устранило бы постоянную составляющую и повысило надёжность передачи.