

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

**Домашнее задание №1**  
**«Методы кодирования в компьютерных сетях»**  
по дисциплине “Компьютерные сети”

Выполнили:  
Студенты группы Р3334  
Баянов Р. Д.

Преподаватель:  
Алиев Т. И.

Санкт-Петербург  
2025 г.

# Оглавление

<b>Формирование сообщения .....</b>	<b>3</b>
<b>Физическое кодирование исходного сообщения .....</b>	<b>4</b>
Манчестерский код.....	4
Потенциальный код без возврата к нулю (NRZ – Non Return to Zero).....	6
Биполярный импульсный код (RZ – Return to Zero).....	8
Биполярное кодирование с альтернативной версией (AMI) .....	10
Потенциальный код с инверсией при единице (NRZI).....	12
Пятиуровневый код PAM-5 (2B1Q) .....	13
Сравнительный анализ методов физического кодирования .....	15
<b>Логическое кодирование исходного сообщения .....</b>	<b>17</b>
Избыточное кодирование (4B/5B) .....	17
Скремблирование .....	19
Сравнительный анализ методов логического кодирования.....	22
<b>Вывод .....</b>	<b>23</b>

# 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)

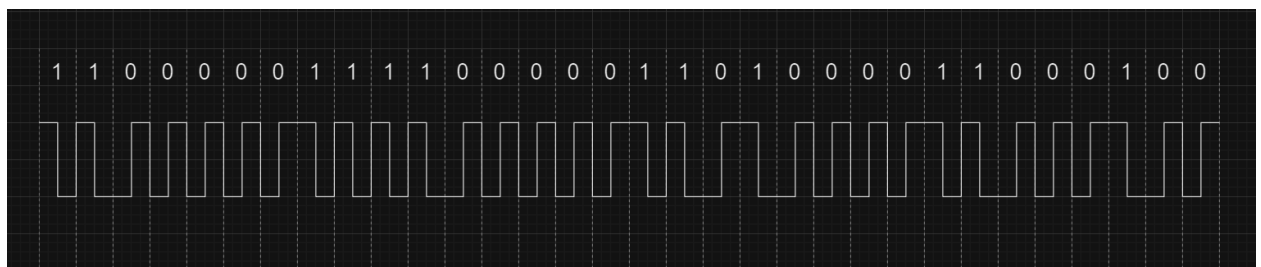
# Физическое кодирование исходного сообщения

Рассмотрим кодирование данного сообщения 6 разными способами, а именно:

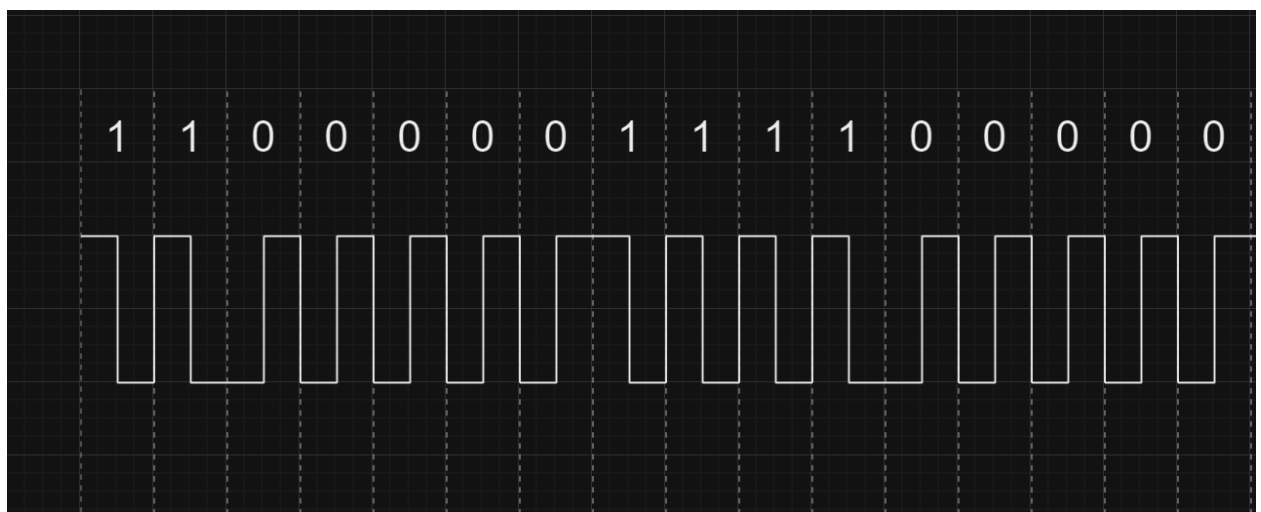
- Манчестерский код
- Потенциальный код без возврата к нулю
- Биполярный импульсный код
- Биполярное кодирование с альтернативной версией
- Потенциальный код с инверсией при единице
- Пятиуровневый код РАМ-5

## Манчестерский код

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт значений данного кодирования:

**Расчёт верхней границы:**  $C = 100 \frac{\text{Мбит}}{с}$  ;  $f_B = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = t \rightarrow f_B = C \text{ Гц} = 100 \text{ МГц}$  ;

**Расчёт нижней границы:**  $T = 2t \rightarrow f_H = \frac{C}{2} \text{ Гц} = 50 \text{ МГц}$  ;

**Следовательно спектр сообщения:**  $S = f_B - f_H = 0.5C \text{ Гц} = 50 \text{ МГц}$  ;

**Средняя частота спектра:**  $f_{\text{ср}} = \frac{(21 f_0 + \frac{11 f_0}{2})}{32} = 0.828125 f_0 = 82,8125 \text{ МГц}$  ;

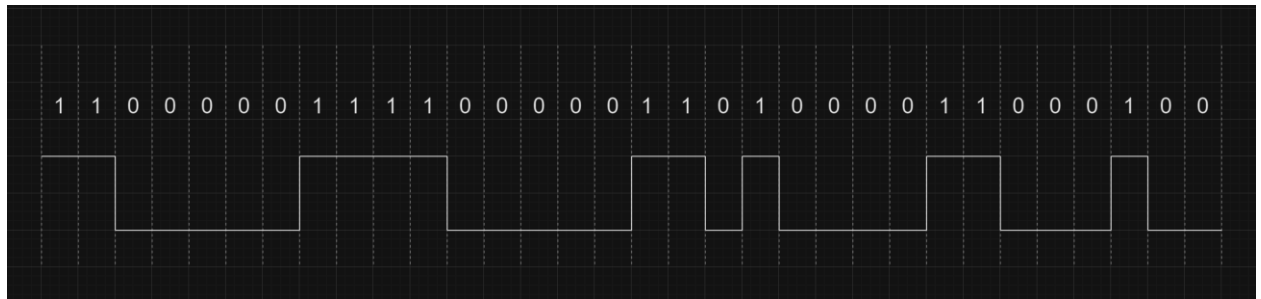
**Середина спектра:**  $f_{\frac{1}{2}} = \frac{(f_B + f_H)}{2} = 75 \text{ МГц}$  ;  $f_{\text{ср}} > f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала преобладают высокие частоты.

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  $F = 55 \text{ МГц}$  ;

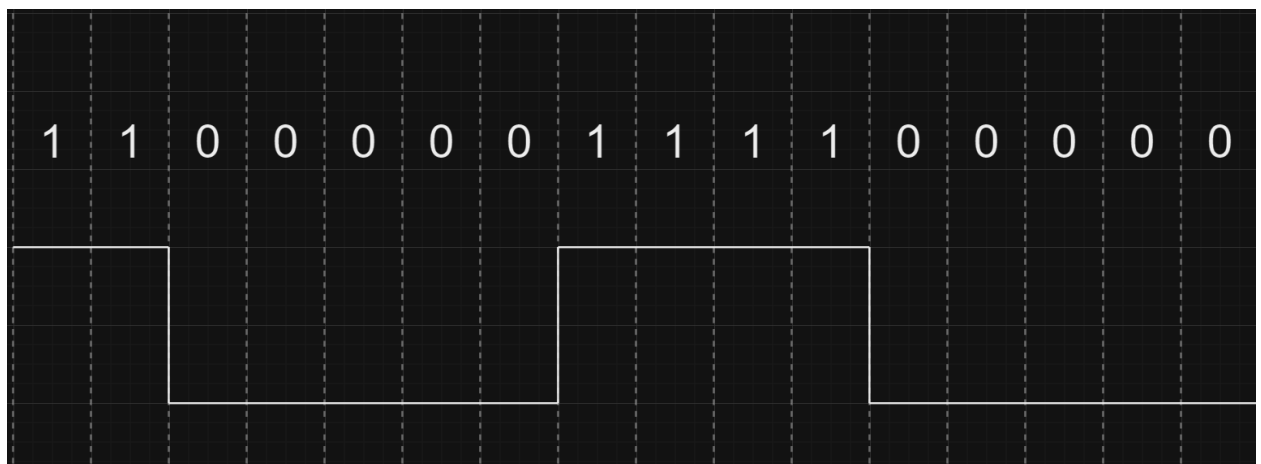
Отсюда можно сделать вывод, что манчестерский код очень даже неплох, так как не имеет постоянной составляющей и обладает способностью самосинхронизации. Но он уступает NRZ в спектре сигнала в общем случае, потому что больше, чем у потенциального. Также манчестерский имеет всего два уровня потенциала, 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:  $C = 100 \frac{\text{Мбит}}{с}$  ;  $f_{\text{в}} = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = 2t \rightarrow f_{\text{в}} = \frac{C}{2}$  Гц = 50 МГц ;

Расчёт нижней границы:  $T = 10t \rightarrow f_{\text{н}} = \frac{C}{10}$  Гц = 10 МГц ;

Следовательно спектр сообщения:  $S = f_{\text{в}} - f_{\text{н}} = 0.4C$  Гц = 40 МГц ;

Средняя частота спектра:  $f_{\text{ср}} = \frac{(3f_0 + \frac{8f_0}{2} + \frac{3f_0}{3} + \frac{8f_0}{4} + \frac{10f_0}{5})}{32} = 0.375f_0 = 18.75 \text{ МГц}$  ;

Середина спектра:  $f_{\frac{1}{2}} = \frac{(f_{\text{в}} + f_{\text{н}})}{2} = 30 \text{ МГц}$  ;  $f_{\text{ср}} < f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала преобладают низкие частоты.

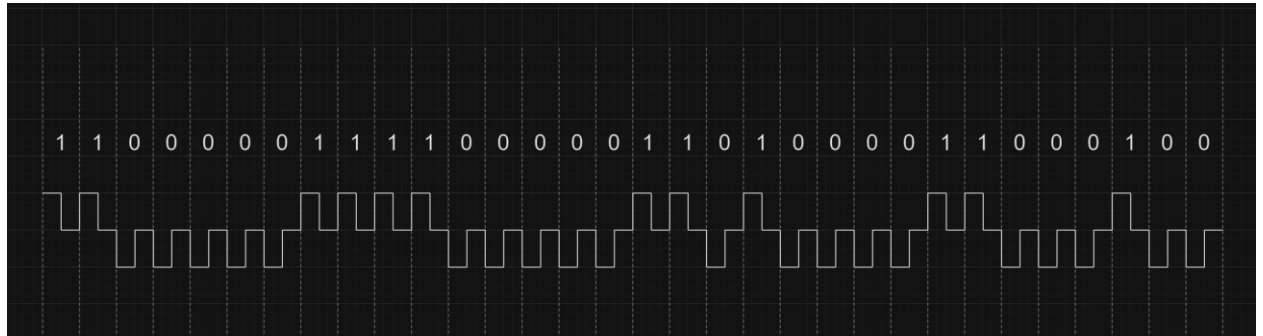
Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:  $F = 45 \text{ МГц}$

Отсюда можно сделать вывод, что потенциальный код уязвим к шумам и не обладает способностью распознавания ошибок. Также NRZ обладает

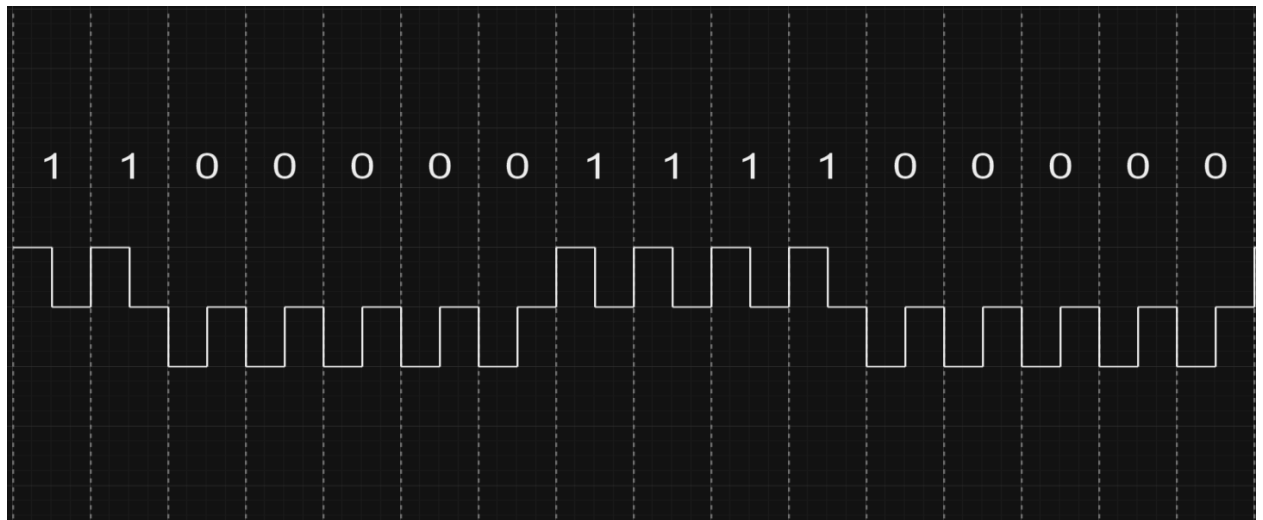
постоянной составляющей и не имеет синхронизации. Но NRZ неплох в том, что в общем случае имеет относительно небольшой спектр сигнала. Также потенциальный код легко реализовать, и он имеет неплохую скорость передачи. В классическом виде на практике не используется.

# Биполярный импульсный код (RZ – Return to Zero)

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт верхней границы:  $C = 100 \frac{\text{Мбит}}{с}$  ;  $f_B = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = t \rightarrow f_B = C \text{ Гц} = 100 \text{ МГц}$  ;

Расчёт нижней границы:  $T = 2.5t \rightarrow f_H = \frac{C}{2.5} \text{ Гц} = 40 \text{ МГц}$  ;

Следовательно спектр сообщения:  $S = f_B - f_H = 0.6C \text{ Гц} = 60 \text{ МГц}$  ;

Средняя частота спектра:  $f_{cp} = \frac{(28 f_0 + \frac{8f_0}{5})}{32} = 0.925 f_0 = 92.5 \text{ МГц}$  ;

Середина спектра:  $f_{\frac{1}{2}} = \frac{(f_B + f_H)}{2} = 70 \text{ МГц}$  ;  $f_{cp} > f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала преобладают высокие частоты.

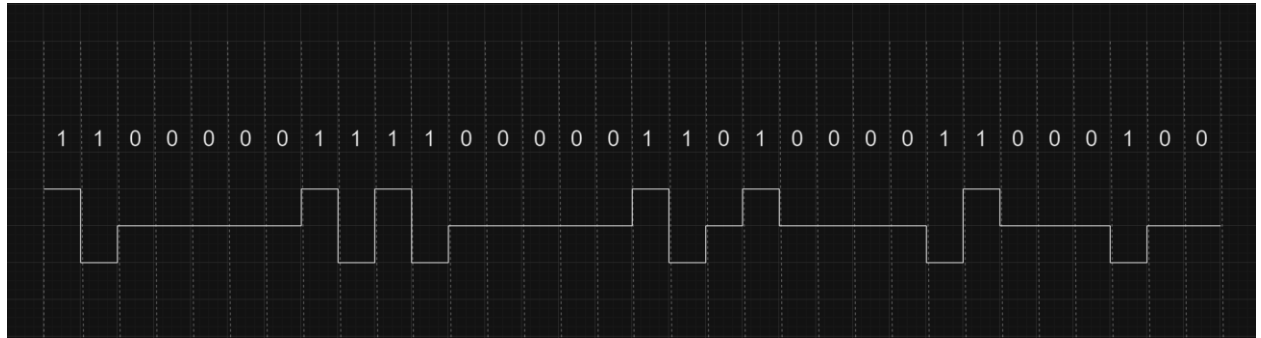
Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:  $F = 65 \text{ МГц}$  ;



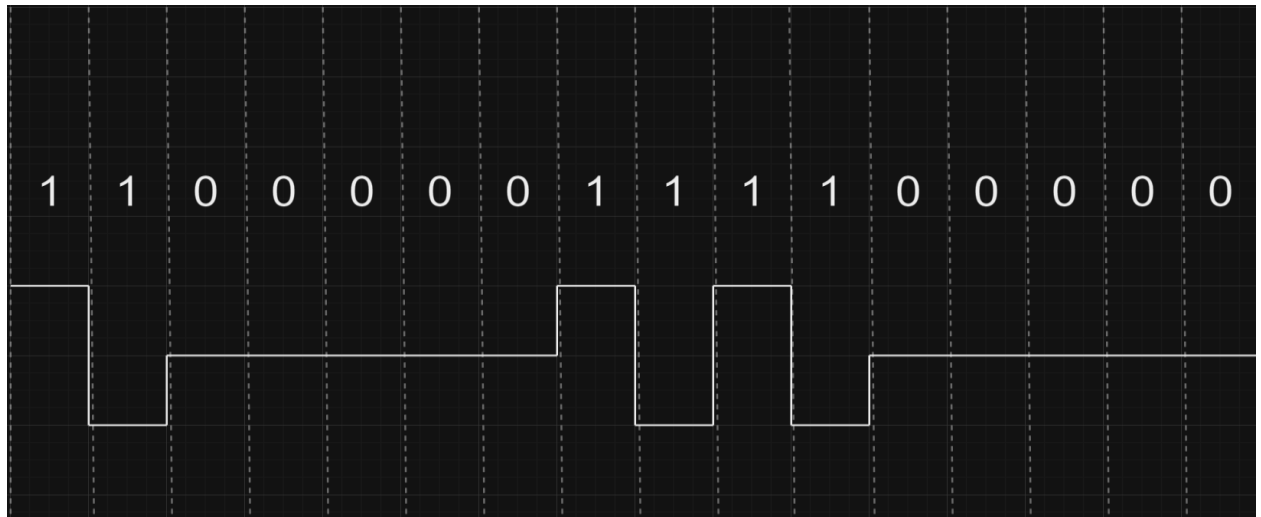
Отсюда делаем вывод, что код RZ, несмотря на самосинхронизацию и отсутствие постоянной составляющей, он имеет три уровня потенциала, что удорожает реализацию и имеет крайне большой спектр сигнала.

## Биполярное кодирование с альтернативной версией (AMI)

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт верхней границы:  $C = 100 \frac{\text{Мбит}}{c}$  ;  $f_B = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = 2t \rightarrow f_B = \frac{C}{2}$  Гц = 50 МГц ;

Расчёт нижней границы:  $T = 14t \rightarrow f_H = \frac{C}{14}$  Гц = 7.143 МГц ;

Следовательно спектр сообщения:  $S = f_B - f_H = 0.42857C$  Гц = 42.857 МГц ;

Средняя частота спектра:  $f_{cp} = \frac{(12f_0 + \frac{10f_0}{7} + \frac{4f_0}{6} + \frac{6f_0}{5})}{32} = 0.47798f_0 = 47.798 \text{ МГц}$

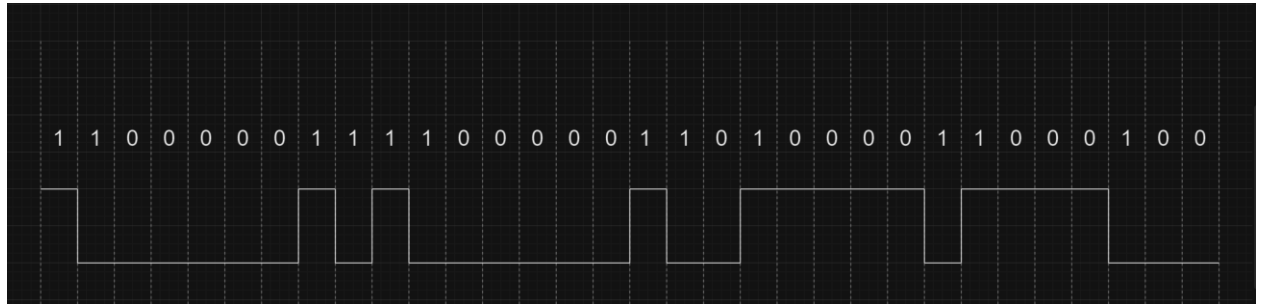
Середина спектра:  $f_{\frac{1}{2}} = \frac{(f_B + f_H)}{2} = 28.5715 \text{ МГц}$  ;  $f_{cp} > f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала преобладают высокие частоты.

Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:  $F = 45 \text{ МГц}$

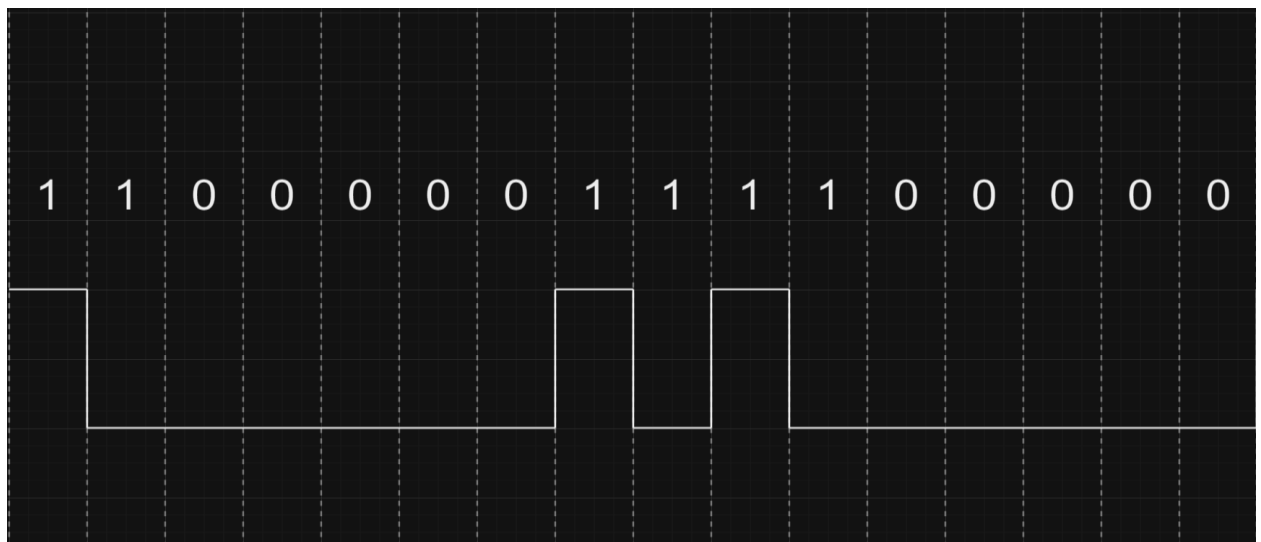
Отсюда сделаем вывод, что АМІ крайне плох для нашего сообщения, он обладает достаточно широким спектром, страдает от постоянных составляющих, не имеет самосинхронизации, да и к тому же имеет 3 уровня потенциала. У АМІ есть только один плюс в нашем случае – это обнаружение ошибок, благодаря чередованию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

# Потенциальный код с инверсией при единице (NRZI)

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт верхней границы:  $C = 100 \frac{\text{Мбит}}{\text{с}}$  ;  $f_{\text{в}} = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = 2t \rightarrow f_{\text{в}} = \frac{C}{2}$  Гц = 50 МГц ;

Расчёт нижней границы:  $T = 12t \rightarrow f_{\text{н}} = \frac{C}{12}$  Гц = 8.333 МГц ;

Следовательно спектр сообщения:  $S = f_{\text{в}} - f_{\text{н}} = 0.417C$  Гц = 41.667 МГц ;

Средняя частота спектра:  $f_{\text{ср}} = \frac{(6f_0 + \frac{2f_0}{2} + \frac{3f_0}{3} + \frac{4f_0}{4} + \frac{5f_0}{5} + \frac{12f_0}{6})}{32} = 0.375f_0 = 18.75 \text{ МГц}$

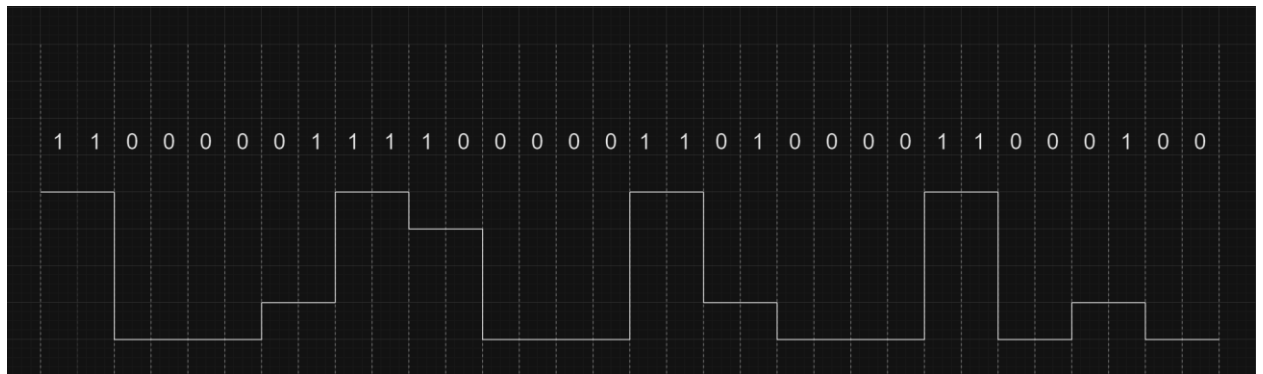
Середина спектра:  $f_{\frac{1}{2}} = \frac{(f_{\text{в}} + f_{\text{н}})}{2} = 29.1665 \text{ МГц}$  ;  $f_{\text{ср}} < f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала преобладают низкие частоты.

Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:  $F = 45 \text{ МГц}$

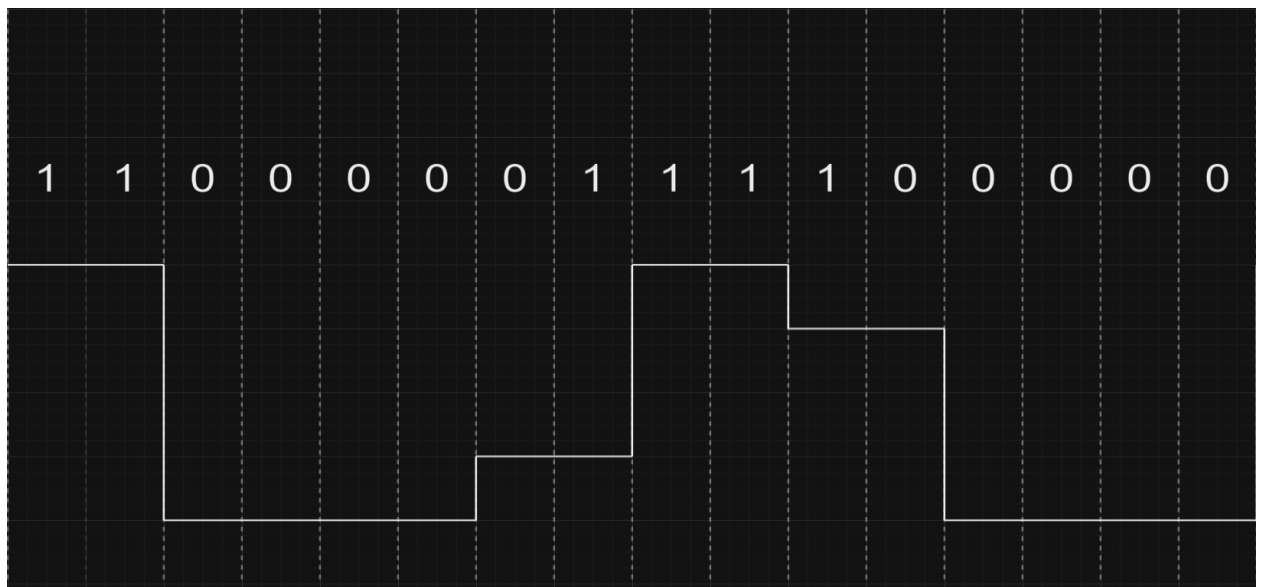
Неплохой способ передачи сообщения, но постоянные составляющие, отсутствие самосинхронизации и спектр шире, чем у NRZ, не делает этот метод полезным для нас.

## Пятиуровневый код РМ-5 (2В1Q)

Временная диаграмма кодирования первых 4 байт:



Временная диаграмма кодирования первых 2 байт:



Расчёт верхней границы:  $C = 100 \frac{\text{Мбит}}{\text{с}}$  ;  $f_{\text{в}} = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = 4t \rightarrow f_{\text{в}} = \frac{C}{4}$  Гц = 25 МГц ;

Расчёт нижней границы:  $T = 8t \rightarrow f_{\text{н}} = \frac{C}{8}$  Гц = 12.5 МГц ;

Следовательно спектр сообщения:  $S = f_{\text{в}} - f_{\text{н}} = 0.125C$  Гц = 12.5 МГц ;

Средняя частота спектра:  $f_{\text{ср}} = \frac{(20 f_0 + \frac{12 f_0}{2})}{32} = 0.8125 f_0 = 20.3125 \text{ МГц}$

**Середина спектра:**  $f_{\frac{1}{2}} = \frac{(f_{\text{в}} + f_{\text{н}})}{2} = 18.75 \text{ МГц}$ ;  $f_{\text{ср}} > f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала преобладают высокие частоты.

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  $F = 15 \text{ МГц}$

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 возможностью 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:

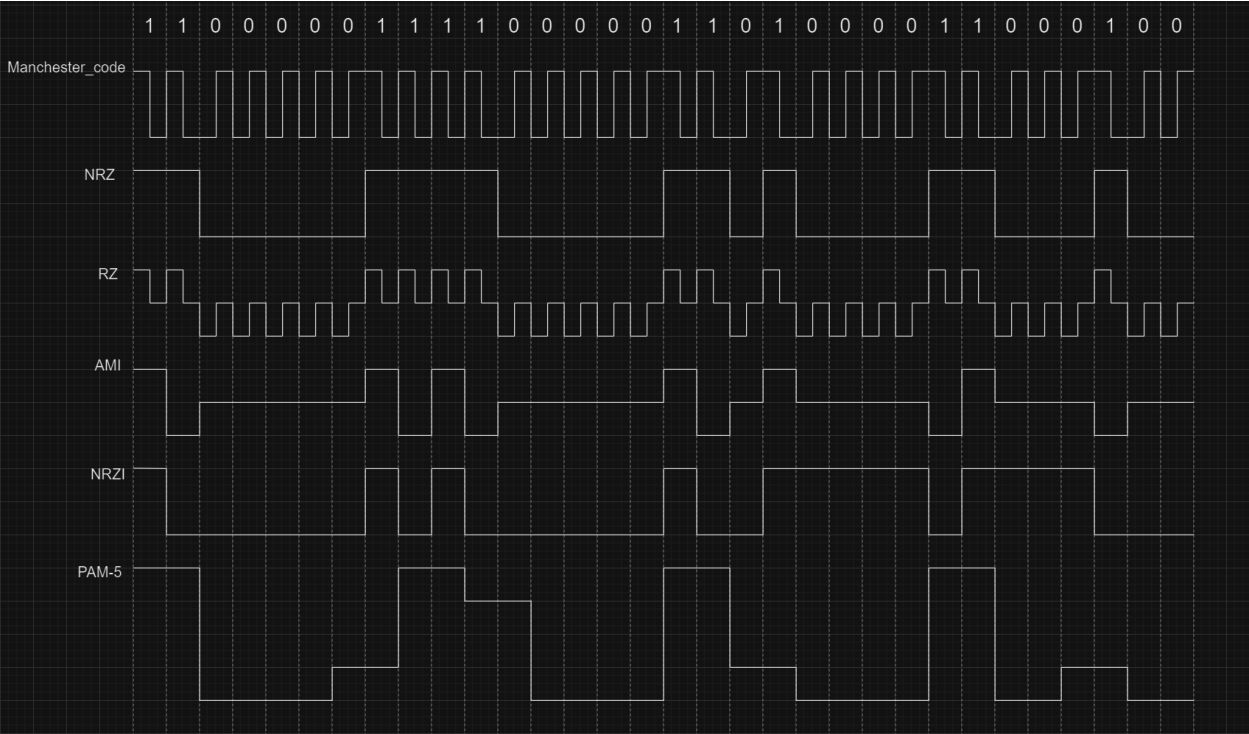


Таблица сравнения всех методов:

	1	2	3	4	5
Manchester	$f_0 = C$	Есть	Нет	Нет	2
NRZ	$f_0 = \frac{C}{2}$	Нет	Есть	Нет	2
RZ	$f_0 = C$	Есть	Нет	Есть	3
AMI	$f_0 = \frac{C}{2}$	Нет	Есть	Есть	3
NRZI	$f_0 = \frac{C}{2}$	Нет	Есть	Нет	2
PAM-5	$f_0 = \frac{C}{4}$	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 АМІ, 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 АМІ нет способности обнаружения ошибок. Наше сообщение обладает большим количеством участков с идущими нулями подряд, поэтому АМІ 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 РАМ-5 является неплохим вариантом, так как он обладает очень быстрой скоростью передачи за счёт 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 манчестерский 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 (4В/5В)

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:**  $A = 1100\ 0001\ 1110\ 0000\ 1101\ 0000\ 1100\ 0100$

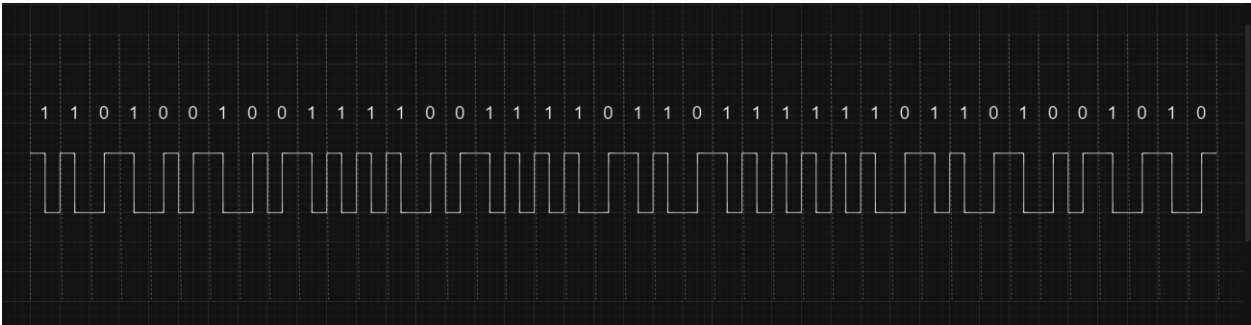
**Результат:**  $B = 11010\ 01001\ 11100\ 11110\ 11011\ 11110\ 11010\ 01010$

**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:



**Расчёт верхней границы:**  $C = 100 \frac{\text{Мбит}}{c}$  ;  $f_B = \frac{1}{T}$  ;  $C = \frac{1}{t}$  ;  $T = t \rightarrow f_B = C \text{ Гц} = 100 \text{ МГц}$  ;

**Расчёт нижней границы:**  $T = 2t \rightarrow f_H = \frac{C}{2} \text{ Гц} = 50 \text{ МГц}$  ;

**Следовательно спектр сообщения:**  $S = f_{\text{в}} - f_{\text{н}} = 0.5\text{С Гц} = 50 \text{ МГц}$  ;

**Средняя частота спектра:**  $f_{\text{ср}} = \frac{(20 f_0 + \frac{20 f_0}{2})}{40} = 0.75 f_0 = 75 \text{ МГц}$  ;

**Середина спектра:**  $f_{\frac{1}{2}} = \frac{(f_{\text{в}} + f_{\text{н}})}{2} = 75 \text{ МГц}$  ;  $f_{\text{ср}} = f_{\frac{1}{2}} \Rightarrow$  следовательно в спектре сигнала не преобладают ни высокие, ни низкие частоты.

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  $F = 55 \text{ МГц}$  ;

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:  $B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$ ;

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

$$B_1 = A_1 = 1;$$

$$B_2 = A_2 = 1;$$

$$B_3 = A_3 = 0;$$

$$B_4 = A_4 \oplus B_1 = 0 \oplus 1 = 1;$$

$$B_5 = A_5 \oplus B_2 = 0 \oplus 1 = 1;$$

$$B_6 = A_6 \oplus B_3 \oplus B_1 = 0 \oplus 0 \oplus 1 = 1;$$

$$B_7 = A_7 \oplus B_4 \oplus B_2 = 0 \oplus 1 \oplus 1 = 0;$$

$$B_8 = A_8 \oplus B_5 \oplus B_3 = 1 \oplus 1 \oplus 0 = 0;$$

$$B_9 = A_9 \oplus B_6 \oplus B_4 = 1 \oplus 1 \oplus 1 = 1;$$

$$B_{10} = A_{10} \oplus B_7 \oplus B_5 = 1 \oplus 0 \oplus 1 = 0;$$

$$B_{11} = A_{11} \oplus B_8 \oplus B_6 = 1 \oplus 0 \oplus 1 = 0;$$

$$B_{12} = A_{12} \oplus B_9 \oplus B_7 = 0 \oplus 1 \oplus 0 = 1;$$

$$B_{13} = A_{13} \oplus B_{10} \oplus B_8 = 0 \oplus 0 \oplus 0 = 0;$$

$$B_{14} = A_{14} \oplus B_{11} \oplus B_9 = 0 \oplus 0 \oplus 1 = 1;$$

$$B_{15} = A_{15} \oplus B_{12} \oplus B_{10} = 0 \oplus 1 \oplus 0 = 1;$$

$$B_{16} = A_{16} \oplus B_{13} \oplus B_{11} = 0 \oplus 0 \oplus 0 = 0;$$

$$B_{17} = A_{17} \oplus B_{14} \oplus B_{12} = 1 \oplus 1 \oplus 1 = 1;$$

$$B_{18} = A_{18} \oplus B_{15} \oplus B_{13} = 1 \oplus 1 \oplus 0 = 0;$$

$$B_{19} = A_{19} \oplus B_{16} \oplus B_{14} = 0 \oplus 0 \oplus 1 = 1;$$

$$B_{20} = A_{20} \oplus B_{17} \oplus B_{15} = 1 \oplus 1 \oplus 1 = 1;$$

$$B_{21} = A_{21} \oplus B_{18} \oplus B_{16} = 0 \oplus 0 \oplus 0 = 0;$$

$$B_{22} = A_{22} \oplus B_{19} \oplus B_{17} = 0 \oplus 1 \oplus 1 = 0;$$

$$B_{23} = A_{23} \oplus B_{20} \oplus B_{18} = 0 \oplus 1 \oplus 0 = 1;$$

$$B_{24} = A_{24} \oplus B_{21} \oplus B_{19} = 0 \oplus 0 \oplus 1 = 1;$$

$$B_{25} = A_{25} \oplus B_{22} \oplus B_{20} = 1 \oplus 0 \oplus 1 = 0;$$

$$B_{26} = A_{26} \oplus B_{23} \oplus B_{21} = 1 \oplus 1 \oplus 0 = 0;$$

$$B_{27} = A_{27} \oplus B_{24} \oplus B_{22} = 0 \oplus 1 \oplus 0 = 1;$$

$$B_{28} = A_{28} \oplus B_{25} \oplus B_{23} = 0 \oplus 0 \oplus 1 = 1;$$

$$B_{29} = A_{29} \oplus B_{26} \oplus B_{24} = 0 \oplus 0 \oplus 1 = 1;$$

$$B_{30} = A_{30} \oplus B_{27} \oplus B_{25} = 1 \oplus 1 \oplus 0 = 0;$$

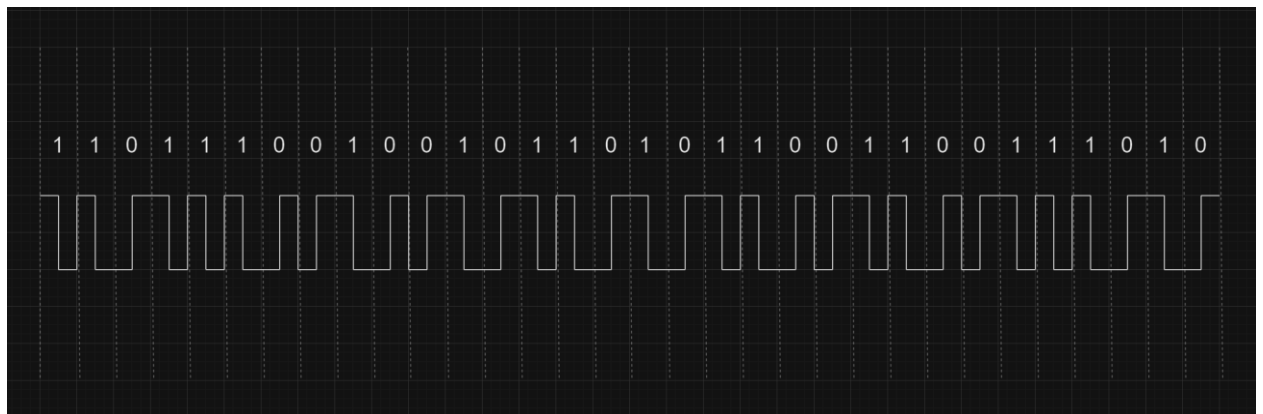
$$B_{31} = A_{31} \oplus B_{28} \oplus B_{26} = 0 \oplus 1 \oplus 0 = 1;$$

$$B_{32} = A_{32} \oplus B_{29} \oplus B_{27} = 0 \oplus 1 \oplus 1 = 0;$$

**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:



**Расчёт верхней границы:**  $C = 100 \frac{\text{Мбит}}{с} ; f_{\text{В}} = \frac{1}{T} ; C = \frac{1}{t} ; T = t \rightarrow f_{\text{В}} = C \text{ Гц} = 100 \text{ МГц} ;$

**Расчёт нижней границы:**  $T = 2t \rightarrow f_{\text{Н}} = \frac{C}{2} \text{ Гц} = 50 \text{ МГц} ;$

**Следовательно спектр сообщения:**  $S = f_{\text{В}} - f_{\text{Н}} = 0.5C \text{ Гц} = 50 \text{ МГц} ;$

**Средняя частота спектра:**  $f_{\text{ср}} = \frac{(21 f_0 + \frac{19 f_0}{2})}{40} = 0.7625 f_0 = 76.25 \text{ МГц} ;$

**Середина спектра:**  $f_{\frac{1}{2}} = \frac{(f_{\text{В}} + f_{\text{Н}})}{2} = 75 \text{ МГц} ; f_{\text{ср}} > f_{\frac{1}{2}} \Rightarrow \text{следовательно в спектре сигнала преобладают высокие частоты.}$

**Полоса пропускания, необходимая для точной передачи данного сообщения данным кодом, например:**  $F = 55 \text{ МГц} ;$

Отсюда мы можем 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	$f_0 = C$	Есть	Нет	Есть	2
Скремблирование	$f_0 = C$	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 дескремблирования. Избыточное же кодирование реализуется крайне просто и добавляет возможность обнаруживать ошибки за счёт запрещённых кодовых.

# Вывод

Выполнив данную лабораторную работу, мы изучили большинство методов кодирования сообщений. Отталкиваясь от данного варианта 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 за счёт 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.