

Университет ИТМО
Факультет программной инженерии и компьютерной техники

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

Выполнил:
Студент группы Р3334
Галиуллин Р.Д.

Преподаватель:
Алиев Т. И.

Санкт-Петербург
2025 г.

ОГЛАВЛЕНИЕ

ФОРМИРОВАНИЕ СООБЩЕНИЯ	3
ФИЗИЧЕСКОЕ КОДИРОВАНИЕ ИСХОДНОГО СООБЩЕНИЯ	4
Манчестерский код	4
NRZ.....	4
NRZI	5
RZ.....	6
Сравнение методов кодирования	7
ЛОГИЧЕСКОЕ (ИЗБЫТОЧНОЕ) КОДИРОВАНИЕ ИСХОДНОГО СООБЩЕНИЯ.....	10
Избыточное кодирование (4В/5В)	10
Скремблирование	12
Сравнительный анализ методов логического кодирования	14
Вывод	14
ВЫВОД.....	16

ФОРМИРОВАНИЕ СООБЩЕНИЯ

Исходное сообщение: ГаРД

В шестнадцатеричном коде: C3 C0 D0 C4

В двоичном коде: 1100 0011 1100 0000 1101 0000 1100 0100

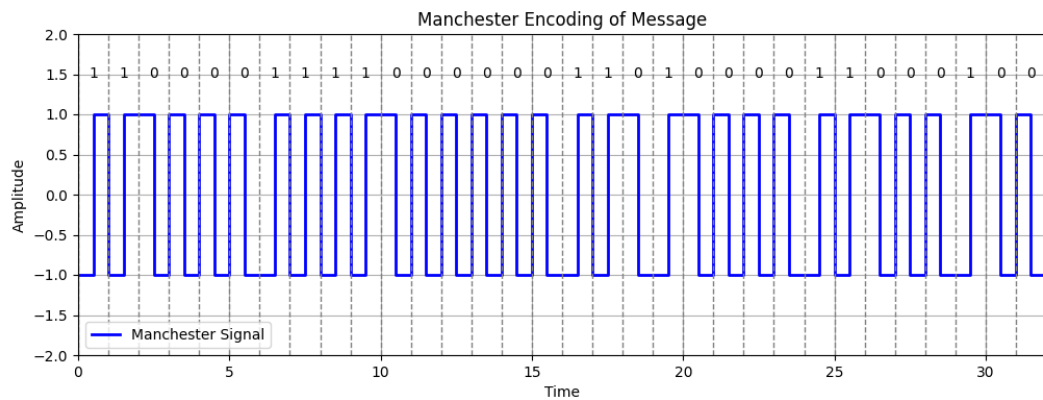
Длина сообщения: 4 байт (32 бит)

Пропускная способность канала: $C = 100 \text{ Мбит / с}$

ФИЗИЧЕСКОЕ КОДИРОВАНИЕ ИСХОДНОГО СООБЩЕНИЯ

Манчестерский код

Диаграмма для первых 4 байт:



Расчет верхней границы: $T_h = T_b \rightarrow f_h = C = 100 \text{ МГц}$

Расчет нижней границы: $T_l = 2 * T_b \rightarrow f_l = \frac{C}{2} = 50 \text{ МГц}$

Спектр сигнала: $S = f_h - f_l = 0.5C = 50 \text{ МГц}$

Средняя частота спектра: $f_{\text{ср}} = \frac{12*f_h + 20*f_l}{32} = 68,75 \text{ МГц}$

Середина спектра: $f_{\text{ср}} = \frac{(f_h + f_l)}{2} = 75 \text{ МГц}$

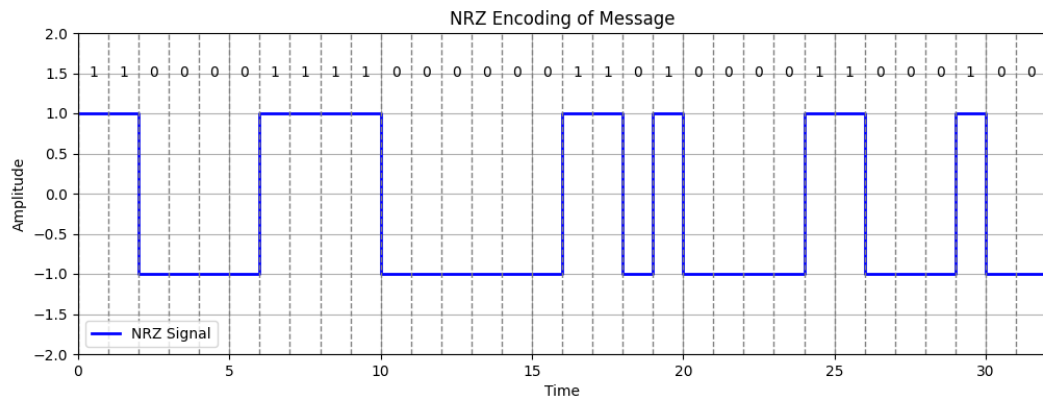
Необходимая полоса пропускания: $F \geq S \Rightarrow F \geq 50 \text{ МГц}$

$f_{\text{ср}} > f_{\text{ср}} \Rightarrow$ в спектре преобладают низкие частоты

Манчестерское кодирование обладает рядом преимуществ, таких как отсутствие постоянной составляющей в сигнале и встроенная возможность самосинхронизации, что делает его надёжным для передачи данных. Однако его спектральная эффективность в большинстве случаев ниже, чем у кода NRZ (Non Return to Zero), так как манчестерский сигнал занимает более широкую полосу частот. Кроме того, использование всего двух уровней потенциала (например, высокий и низкий) упрощает техническую реализацию данного метода, снижая стоимость оборудования. Таким образом, несмотря на ограничения в спектральной эффективности, манчестерский код остаётся практичным решением для систем, где критически важны синхронизация и простота реализации.

NRZ

Диаграмма для первых 4 байт:



Расчет верхней границы: $T_h = 2 * T_b \rightarrow f_h = \frac{c}{2} = 50 \text{ МГц}$

Расчет нижней границы: $T_l = 8 * T_b \rightarrow f_l = \frac{c}{8} = 12,5 \text{ МГц}$

Спектр сигнала: $S = f_h - f_l = 37,5 \text{ МГц}$

Средняя частота спектра: $f_{\text{ср}} = \frac{4*50+12*25+3*16,7+8*12,5}{15} = 24,08 \text{ МГц}$

Середина спектра: $f_{\text{ср}} = \frac{(f_h+f_l)}{2} = 31,25 \text{ МГц}$

Необходимая полоса пропускания: $F \geq S \Rightarrow F \geq 37,5 \text{ МГц}$

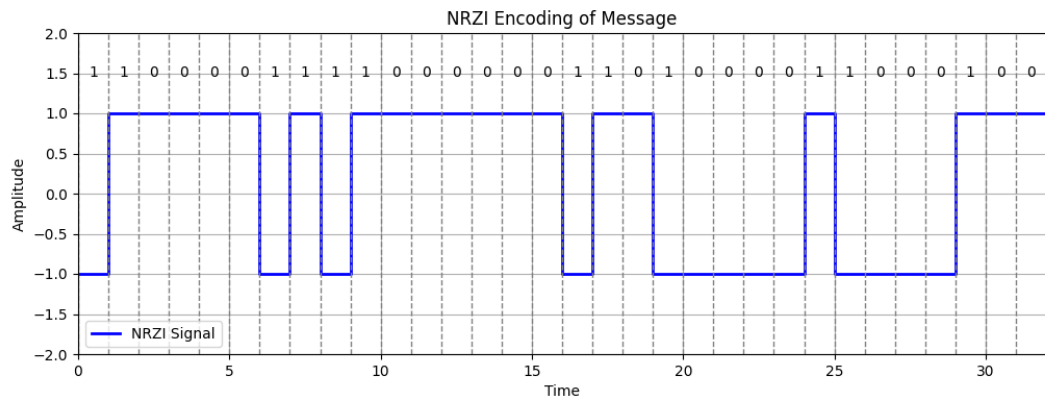
$f_{\text{ср}} > f_{\text{ср}} \Rightarrow$ в спектре преобладают низкие частоты

Код обладает существенными недостатками: он чувствителен к помехам и не предусматривает механизмов обнаружения ошибок. Кроме того, метод NRZ (Non-Return-to-Zero), несмотря на отсутствие встроенной синхронизации и наличие постоянной составляющей (DC-уровня), демонстрирует преимущество в виде компактного спектра сигнала, что повышает его эффективность в условиях ограниченной полосы пропускания.

С другой стороны, потенциальный код привлекателен простотой реализации и высокой скоростью передачи данных. Однако в классическом виде он редко применяется на практике из-за указанных ограничений. Таким образом, выбор между этими методами зависит от требований конкретной системы: NRZ подходит для задач с фокусом на спектральную эффективность, а потенциальный код — для сценариев, где приоритетом являются простота и скорость.

NRZI

Диаграмма для первых 4 байт:



Расчет верхней границы: $T_h = 2 * T_b \rightarrow f_h = \frac{c}{2} = 50 \text{ МГц}$

Расчет нижней границы: $T_l = 10 * T_b \rightarrow f_l = \frac{c}{10} = 10 \text{ МГц}$

Спектр сигнала: $S = f_h - f_l = 40 \text{ МГц}$

Средняя частота спектра: $f_{\text{ср}} = \frac{6*50*1+1*25*2+1*16,7*3+2*12,5*4+1*10*5}{24} = 22.92 \text{ МГц}$

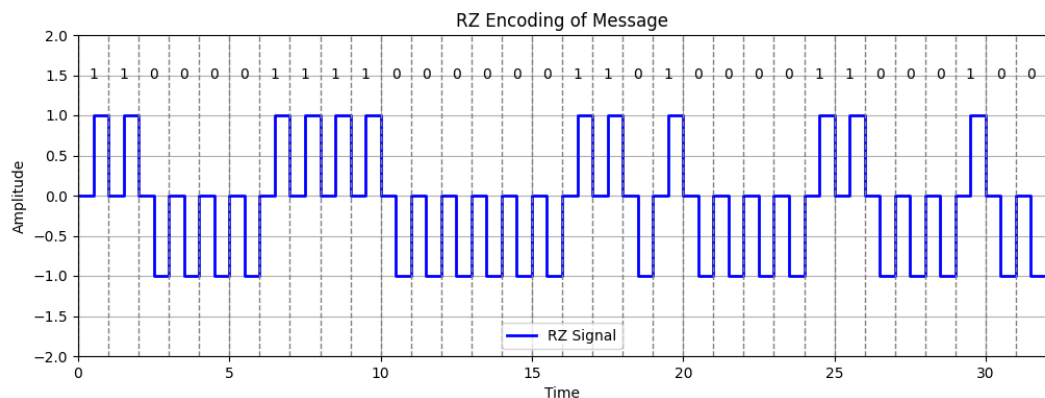
Середина спектра: $f_{\text{ср}} = \frac{(f_h + f_l)}{2} = 30 \text{ МГц}$

Необходимая полоса пропускания: $F \geq S \Rightarrow F \geq 40 \text{ МГц}$

$f_{\text{ср}} > f_{\text{ср}} \Rightarrow$ в спектре преобладают низкие частоты

Метод NRZI (Non-Return-to-Zero Inverted) демонстрирует определённую эффективность в передаче данных, однако его применение ограничено рядом недостатков. Наличие постоянной составляющей (DC-уровня), отсутствие встроенной самосинхронизации и более широкая полоса спектра по сравнению с классическим NRZ снижают его практическую ценность для задач, где критически важны устойчивость к помехам, компактность спектра и автономная синхронизация. Таким образом, несмотря на отдельные достоинства, NRZI не является оптимальным выбором для реализации в наших условиях.

RZ



Расчет верхней границы: $T_h = T_b \rightarrow f_h = C = 100 \text{ МГц}$

Расчет нижней границы: $T_l = 5 * T_b \rightarrow f_l = \frac{C}{4} = 25 \text{ МГц}$

Спектр сигнала: $S = f_h - f_l = 75 \text{ МГц}$

Средняя частота спектра: $f_{cp} = \frac{12*100+20*25}{32} = 53,13 \text{ МГц}$

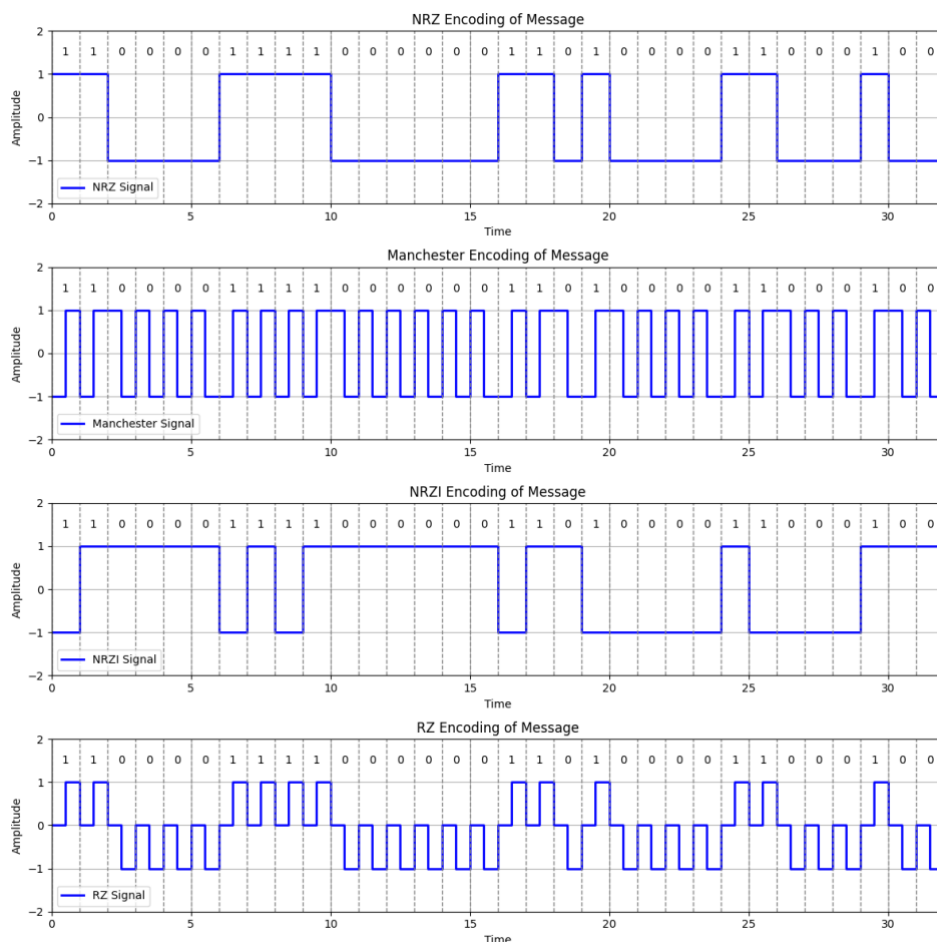
Середина спектра: $f_{сер} = \frac{(f_h+f_l)}{2} = 62,5 \text{ МГц}$

Необходимая полоса пропускания: $F \geq S \Rightarrow F \geq 25 \text{ МГц}$

$f_{сер} > f_{cp} \Rightarrow$ в спектре преобладают низкие частоты

Код RZ (Return-to-Zero), несмотря на такие преимущества, как встроенная самосинхронизация и отсутствие постоянной составляющей, имеет ряд существенных ограничений. Использование трёх уровней сигнала усложняет аппаратную реализацию, увеличивая стоимость системы, а также приводит к значительному расширению спектра сигнала по сравнению с другими методами кодирования. Это делает RZ менее практичным в сценариях, где критически важны экономическая эффективность и оптимизация полосы пропускания.

Сравнение методов кодирования



Для сравнения выберем следующие параметры:

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

	1	2	3	4	5
Manchester	$f_0 = C$	Есть	Нет	Нет	2
NRZ	$f_0 = \frac{C}{2}$	Нет	Есть	Нет	2
NRZI	$f_0 = \frac{C}{2}$	Нет	Есть	Нет	2
RZ	$f_0 = C$	Есть	Нет	Есть	3

NRZ и NRZI демонстрируют наименьшую ширину спектра из рассматриваемых методов (в частности, у NRZ — 37.5 МГц), при этом оставаясь самыми простыми и дешёвыми в реализации за счёт использования всего двух уровней сигнала. Однако их главные недостатки — отсутствие самосинхронизации, невозможность обнаружения ошибок и наличие постоянной составляющей. Это делает их мало надёжными для передачи данных в нестабильной среде.

Манчестерское кодирование показывает отличный баланс характеристик. Оно имеет встроенную самосинхронизацию, отсутствие постоянной составляющей и использует всего два уровня сигнала, как и NRZ. Частотный спектр у него шире (50 МГц), но это умеренная плата за устойчивость и надёжность. Также возможна частичная диагностика ошибок: если не происходит перехода в середине бита — это может указывать на проблему. С учётом всех плюсов, Манчестерский код — один из двух лучших вариантов.

RZ схож с Манчестерским кодом по преимуществам: есть самосинхронизация, нет постоянной составляющей, возможна частичная проверка на ошибки. Однако из-за необходимости использования трёх уровней сигнала и самой широкой полосы (75 МГц), он становится наименее выгодным с практической точки зрения.

Итог:

- Победитель: Манчестерский код — оптимальное сочетание надёжности, простоты и умеренного спектра.
- Аутсайдер: RZ — слишком высокие требования к полосе и реализации.
- NRZ и NRZI — подходят, если приоритет в минимальной стоимости и ограниченной полосе, но с компромиссом по надёжности.

ЛОГИЧЕСКОЕ (ИЗБЫТОЧНОЕ) КОДИРОВАНИЕ ИСХОДНОГО СООБЩЕНИЯ

Избыточное кодирование (4В/5В)

4В	5В	4В	5В
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:

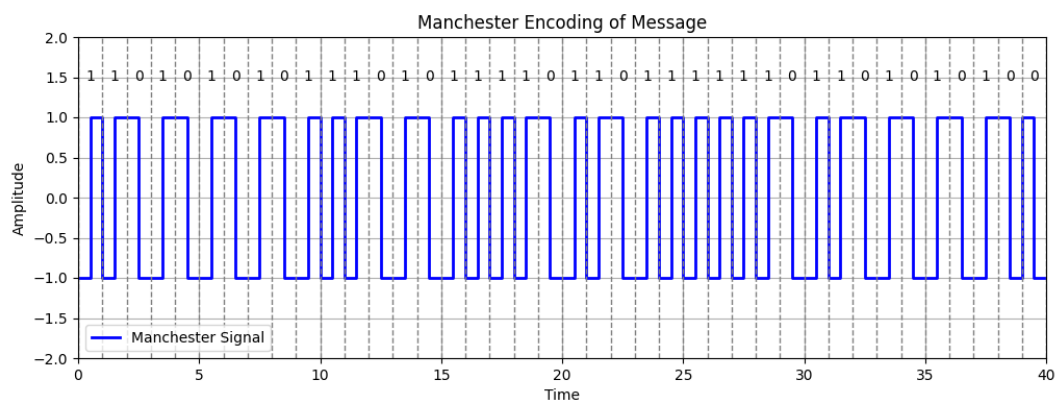
- В шестнадцатеричном коде: C3 C0 D0 C4
- В двоичном коде: 1100 0011 1100 0000 1101 0000 1100 0100
- Длина сообщения: 4 байт (32 бит)

После кодировки:

- В шестнадцатеричном коде: D5 75 ED FB 54
- 11010 10101 11010 11110 11011 11110 11010 10100
- Длина сообщения: 5 байт (40 бит)

Избыточность: 25%

Так как победителем на прошлом этапе был Манчестерский код, воспользуемся им для получения временной диаграммы



Расчет верхней границы: $T_h = T_b \rightarrow f_h = C = 100 \text{ МГц}$

Расчет нижней границы: $T_l = 2 * T_b \rightarrow f_l = \frac{C}{2} = 50 \text{ МГц}$

Спектр сигнала: $S = f_h - f_l = 0.5C = 50 \text{ МГц}$

Средняя частота спектра: $f_{\text{ср}} = \frac{26*f_h + 14*f_l}{40} = 82,5 \text{ МГц}$

Середина спектра: $f_{\text{ср}} = \frac{(f_h + f_l)}{2} = 75 \text{ МГц}$

Необходимая полоса пропускания: $F \geq S \Rightarrow F \geq 50 \text{ МГц}$

$f_{\text{ср}} > f_{\text{ср}} \Rightarrow$ в спектре преобладают высокие частоты

Применение избыточного кодирования к манчестерскому физическому кодированию позволило улучшить его функциональность. Хотя ширина спектра сигнала осталась неизменной, добавление избыточных битов ввело механизм обнаружения ошибок за счёт использования запрещённых комбинаций сигналов. Например, определённые переходы или уровни, не соответствующие правилам кодирования, теперь явно указывают на искажение данных.

Однако ключевое изменение произошло в распределении частот внутри спектра. Если ранее в спектре преобладали низкие частоты, то после внедрения избыточности основная энергия сигнала сместилась в область высоких частот. Это связано с увеличением количества переходов между уровнями, вызванным добавлением избыточных битов, что усилило высокочастотные компоненты.

Избыточное кодирование незначительно повлияло на общее качество передачи, но добавило возможность контроля ошибок.

Спектральные характеристики изменились: высокие частоты стали доминировать, что может потребовать корректировки полосы пропускания канала.

Таким образом, манчестерский код с избыточностью сохранил свои базовые преимущества (самосинхронизацию, отсутствие DC-уровня), но приобрёл новые свойства, актуальные для систем с повышенными требованиями к надёжности.

Скремблирование

Исходное сообщение: 1100 0011 1100 0000 1101 0000 1100 0100

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

$$B_1 = A_1 = 1$$

$$B_2 = A_2 = 1$$

$$B_3 = A_3 = 0$$

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

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

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

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

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

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

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

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

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

$$B_{13} = A_{13} \oplus B_{10} \oplus B_8 = 0 \oplus 1 \oplus 0 = 1$$

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

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

$$B_{16} = A_{16} \oplus B_{13} \oplus B_{11} = 0 \oplus 1 \oplus 1 = 0$$

$$B_{17} = A_{17} \oplus B_{14} \oplus B_{12} = 1 \oplus 0 \oplus 0 = 1$$

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

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

$$B_{20} = A_{20} \oplus B_{17} \oplus B_{15} = 0 \oplus 1 \oplus 1 = 0$$

$$B_{21} = A_{21} \oplus B_{18} \oplus B_{16} = 1 \oplus 1 \oplus 0 = 0$$

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

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

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

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

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

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

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

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

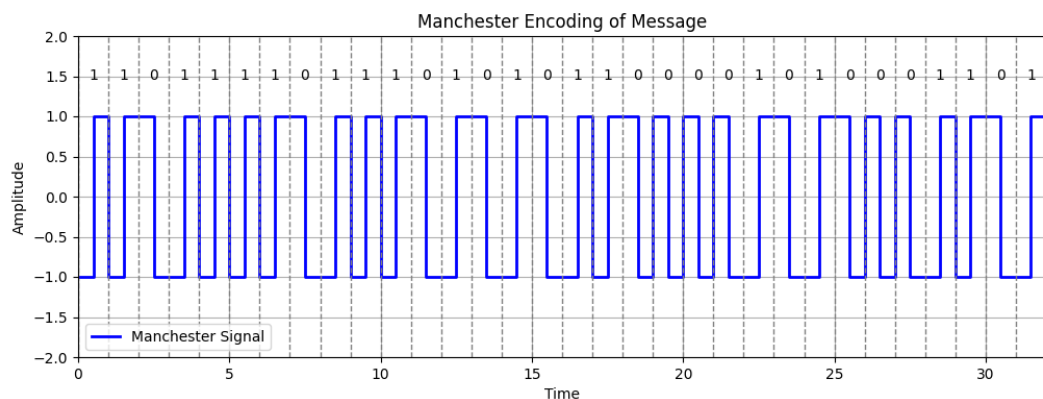
$$B_{30} = A_{30} \oplus B_{27} \oplus B_{25} = 0 \oplus 0 \oplus 1 = 1$$

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

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

Полученный код: 1101 1110 1110 1010 1100 0010 1000 1101

Шестнадцатеричный код: DE EA C2 8D



Расчет верхней границы: $T_h = T_b \rightarrow f_h = C = 100 \text{ МГц}$

Расчет нижней границы: $T_l = 2 * T_b \rightarrow f_l = \frac{C}{2} = 50 \text{ МГц}$

Спектр сигнала: $S = f_h - f_l = 0.5C = 50 \text{ МГц}$

Средняя частота спектра: $f_{\text{ср}} = \frac{18*f_h + 14*f_l}{32} = 78,13 \text{ МГц}$

Середина спектра: $f_{\text{ср}} = \frac{(f_h + f_l)}{2} = 75 \text{ МГц}$

Необходимая полоса пропускания: $F \geq S \Rightarrow F \geq 50 \text{ МГц}$

$f_{\text{ср}} > f_{\text{ср}} \Rightarrow$ в спектре преобладают высокие частоты

Скремблирование доказало свою эффективность в устранении длинных последовательностей нулей и единиц, что особенно критично для методов кодирования, склонных к возникновению постоянной составляющей (например, NRZ или NRZI). В таких случаях рандомизация данных через скремблирование минимизирует риск искажений, связанных с DC-уровнем.

Однако применение скремблирования к манчестерскому коду оказалось избыточным. Этот метод уже обладает встроенной защитой от постоянной составляющей и обеспечивает

самосинхронизацию за счёт обязательных переходов между уровнями в каждом битовом интервале. Таким образом, дополнительная обработка скремблированием не принесла значимых преимуществ, так как базовые недостатки, которые оно устраняет, в манчестерском коде изначально отсутствуют.

Скремблирование полезно для кодов с уязвимостью к постоянной составляющей. Для манчестерского кода его использование не оправдано из-за inherent-устойчивости к таким проблемам.

Сравнительный анализ методов логического кодирования

Для сравнения выберем следующие параметры:

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

	1	2	3	4	5
4B/5B	$f_0 = C$	Есть	Нет	Есть	2
Скремблирование	$f_0 = C$	Есть	Нет	Нет	2

Вывод

Избыточное кодирование демонстрирует более высокую эффективность в модификации манчестерского кода по сравнению со скремблированием. Причины:

- Скремблирование для манчестерского кода:
 - Не приносит практической пользы: манчестерский код уже исключает длинные последовательности нулей/единиц и постоянную составляющую благодаря обязательным переходам в каждом бите.
 - Высокие затраты: требует сложных алгоритмов скремблирования/дескремблирования, что увеличивает вычислительную нагрузку.
- Избыточное кодирование:
 - Простота реализации: добавление избыточных битов не требует ресурсоёмких вычислений.

- Обнаружение ошибок: использование запрещённых комбинаций сигналов (например, нарушение паттернов переходов) позволяет выявлять искажения данных.

Для данного сообщения скремблирование оказалось избыточным, так как базовые недостатки, которые оно устраняет, в манчестерском коде отсутствуют. В то же время избыточное кодирование, не меняя структуру сигнала, добавило критически важную функцию контроля целостности данных.

Избыточное кодирование — оптимальный выбор для улучшения манчестерского кода, обеспечивающий баланс между простотой, надёжностью и функциональностью.

Скремблирование целесообразно применять только для методов, подверженных DC-уровню и длинным последовательностям (NRZ, NRZI).

ВЫВОД

В ходе лабораторной работы был проведён анализ методов физического и логического кодирования для заданного сообщения. Основные результаты и выводы:

1. Выбор оптимального физического кодирования

- Манчестерский код и RZ были выделены как наиболее подходящие методы:
 - Преимущества:
 - Отсутствие постоянной составляющей (DC-уровня).
 - Самосинхронизация (переходы в каждом битовом интервале).
 - Простота реализации (2 уровня сигнала для Manchester, 3 уровня для RZ).
 - Возможность обнаружения ошибок (для RZ — возврат к нулю, для Manchester — нарушение паттернов переходов).
 - Недостатки:
 - RZ имеет более широкий спектр (80 МГц) по сравнению с Manchester (50 МГц).
 - RZ дороже из-за трёх уровней сигнала.

2. Влияние логического кодирования

- Скремблирование:
 - Эффективно для методов, склонных к постоянной составляющей (NRZ, NRZI), но бесполезно для Manchester, так как он уже исключает длинные последовательности нулей/единиц.
 - Увеличивает вычислительную нагрузку из-за сложности алгоритмов.
- Избыточное кодирование (4B/5B):
 - Добавляет возможность обнаружения ошибок через запрещённые комбинации.
 - Недостаток: снижает скорость передачи на 25% из-за избыточных битов.

3. Оптимальный метод для данного сообщения

- Манчестерский код с логическим кодированием 4B/5B был выбран как лучший вариант:
 - Преимущества:
 - Надёжность: самосинхронизация + обнаружение ошибок.
 - Отсутствие DC-уровня.

- Низкая стоимость реализации (2 уровня сигнала).
- Компромиссы:
 - Избыточность 25% замедляет передачу, но повышает надёжность.
 - Спектр шире, чем у NRZ (50 МГц против 37.5 МГц), но приемлем для большинства каналов.

4. Общие итоги

- Изучены принципы физического и логического кодирования, их сильные и слабые стороны.
- Практическое применение методов показало, что выбор зависит от приоритетов:
 - Скорость передачи: Manchester без избыточности.
 - Надёжность: Manchester с 4В/5В.
- Манчестерский код с 4В/5В — баланс между скоростью, стоимостью и устойчивостью к ошибкам, что делает его универсальным решением для большинства сценариев.

Заключение:

Для заданного сообщения оптимальным решением является манчестерский код с логическим кодированием 4В/5В, обеспечивающий надёжную передачу данных без критического ущерба скорости. Однако в условиях строгих требований к полосе пропускания стоит рассмотреть NRZ со скремблированием, несмотря на его ограничения.