Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Компьютерные сети

**Домашнее задание**

## Выполнил:

Гуменник Петр Олегович

**Группа:** P3333

## Преподаватель:

Авксентьева Елена Юрьевна

2025 г.

Санкт-Петербург

# Цель работы

* Изучение методов физического кодирования.
* Изучение методов логического кодирования.
* Проведение сравнительного анализа используемых способов кодирования для выявления их достоинств и недостатков.
* Определение наилучшего способа кодирования для передачи исходного сообщения.

# Этап 1. Формирование сообщения

**Исходное сообщение:** ГПО

**В шестнадцатеричном коде:** C3 CF CE

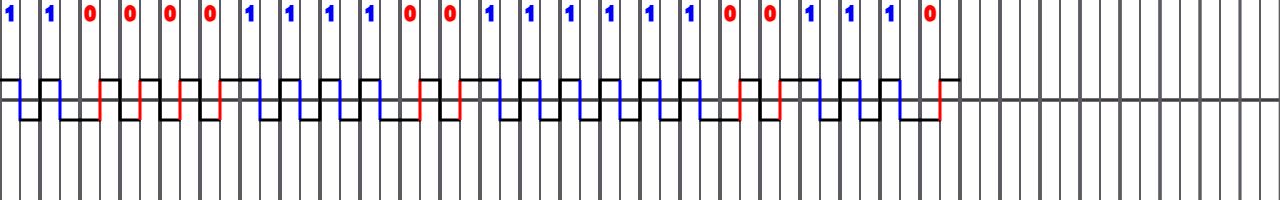
#### В двоичном коде: 1100 0011 1100 1111 1100 1110

**Длина сообщения:** 3 байт (24 бит)

# Этап 2. Физическое кодирование исходного сообщения

### M2 - манчестерский код

### Верхняя граница частот: 100 МГц Нижняя граница частот: 50 МГц Середина спектра: 75 МГц Средняя частота: (34\*f0 + 14 \*f0/2)/ 48 = 85.417 МГц Спектр сигнала: 650 МГц Полоса пропускания: 650 МГц



### NRZ - потенциальный код без возврата к нулю

### Верхняя граница частот: 50 МГц

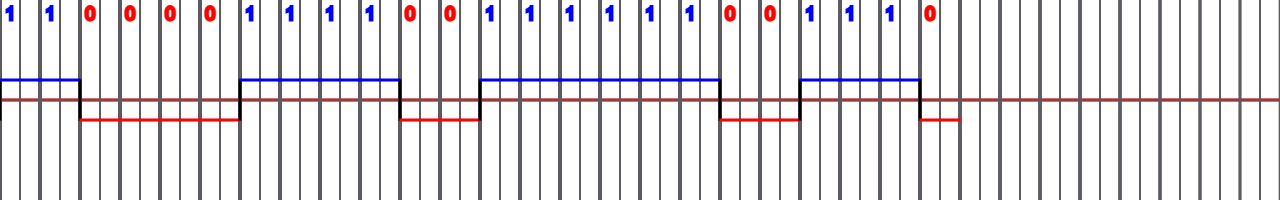
### Нижняя граница частот: 8.334 МГц

### Середина спектра: 29.167 МГц

### Средняя частота: (1\*f0/1 + 6\*f0/2 + 3\*f0/3 + 8\*f0/4 + 6\*f0/6)/24 = 16.667 МГц

### Спектр сигнала: 341.667 МГц

### Полоса пропускания: 342 МГц



### RZ - биполярный импульсный код

### Верхняя граница частот: 100 МГц Нижняя граница частот: 50 МГц Середина спектра: 75 МГц Средняя частота: (48\*f0 + 0\*f0/2.5)/48 = 100 МГц Спектр сигнала: 650 МГц Полоса пропускания: 650 МГц

### 

### AMI - биполярное кодирование с чередующейся инверсией

### Верхняя граница частот: 50 МГц Нижняя граница частот: 12.5 МГц Середина спектра: 31.25 МГц Средняя частота: (16\*f0/1 + 4\*f0/2 + 4\*f0/4)/24 = 39.583 МГц Спектр сигнала: 337.5 МГц Полоса пропускания: 340 МГц

### 

### Сравнительный анализ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод кодирования | Спектр сигнала (МГц) | Само  синхронизация | Постоянная составляющая | Обнаружение ошибок | Стоимость реализации |
| **M2** | 650 | есть | нет | есть | 2 |
| **NRZ** | 341.667 | нет | есть | нет | 1 |
| **RZ** | 650 | есть | нет | есть | 3 |
| **AMI** | 337.5 | нет | есть | есть | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод кодирования | Верхняя граница (МГц) | Нижняя граница (МГц) | Средняя частота | Полоса пропускания |
| **M2** | 100 | 50 | 85.417 | 650 |
| **NRZ** | 50 | 8.334 | 16.667 | 342 |
| **RZ** | 100 | 50 | 100 | 650 |
| **AMI** | 50 | 12.5 | 39.583 | 340 |

**Выбор: M2 и AMI.**

**M2 (Манчестерский код)** — обеспечивает самосинхронизацию, не имеет постоянной составляющей, позволяет обнаруживать ошибки. Подходит для систем, где важно стабильное восстановление тактового сигнала.

**AMI (биполярное с чередующейся инверсией)** — не требует широкой полосы пропускания, позволяет обнаруживать ошибки. Эффективен в каналах с ограниченным спектром.

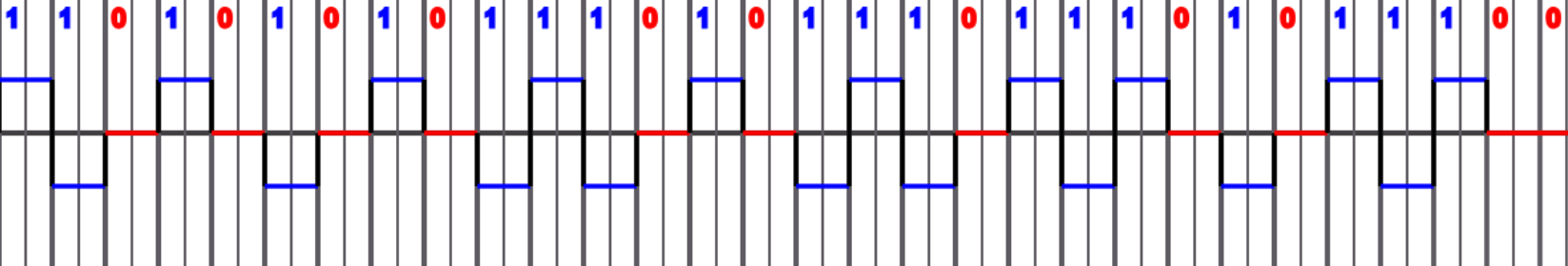
# Этап 3. Логическое (избыточное) кодирование исходного сообщения - (4B/5B)

**В двоичном коде:** 11 0101 0101 1101 0111 0111 0101 1100

**В шестнадцатеричном коде:** 35 5D 77 5C

**Длина сообщения:** 3.75 байт (30 бит)

#### Избыточность: 25%



Верхняя граница частот: 50 МГц  
Нижняя граница частот: 25 МГц  
Середина спектра: 37.5 МГц  
Средняя частота: (28\*f0/1 + 2\*f0/2)/30 = 48.333333333333336 МГц  
Спектр сигнала: 325 МГц  
Полоса пропускания: 330 МГц

# Этап 4. Скремблирование исходного сообщения

𝐵𝑖 = 𝐴𝑖 ⊕ 𝐵𝑖−3 ⊕ 𝐵𝑖−5

Данный полином выбран так как за счет того, что мы учитываем разряды «через один» (имеются в виду i-3 и i-5) мы сможем разбавить постоянную составляющую. Длина постоянной составляющей в моем случае не превышает 4 -> если мы будем учитывать биты со сдвигом на 3 и на 5 (которые, допустим, будут входить в постоянную

составляющую), то текущее значение Ai почти всегда не будет совпадать с ними, так как оно не будет входить в постоянную составляющую -> команда XOR будет выдавать попеременно (в основном) разные значения.

**Исходное сообщение:** 1100 0011 1100 1111 1100 1110

𝐵1 = 𝐴1 = 1

𝐵2 = 𝐴2 = 1

𝐵3 = 𝐴3 = 0

𝐵4 = 𝐴4 ⊕ 𝐵1 = 0 ⊕ 1 = 1

𝐵5 = 𝐴5 ⊕ 𝐵2 = 0 ⊕ 1 = 1

𝐵6 = 𝐴6 ⊕ 𝐵3 ⊕ 𝐵1 = 0 ⊕ 0 ⊕ 1 = 1

𝐵7 = 𝐴7 ⊕ 𝐵4 ⊕ 𝐵2 = 1 ⊕ 1 ⊕ 1 = 1

𝐵8 = 𝐴8 ⊕ 𝐵5 ⊕ 𝐵3 = 1 ⊕ 1 ⊕ 0 = 0

𝐵9 = 𝐴9 ⊕ 𝐵6 ⊕ 𝐵4 = 1 ⊕ 1 ⊕ 1 = 1

𝐵10 = 𝐴10 ⊕ 𝐵7 ⊕ 𝐵5 = 1 ⊕ 1 ⊕ 1 = 1

𝐵11 = 𝐴11 ⊕ 𝐵8 ⊕ 𝐵6 = 0 ⊕ 0 ⊕ 1 = 1

𝐵12 = 𝐴12 ⊕ 𝐵9 ⊕ 𝐵7 = 0 ⊕ 1 ⊕ 1 = 0

𝐵13 = 𝐴13 ⊕ 𝐵10 ⊕ 𝐵8 = 1 ⊕ 1 ⊕ 0 = 0

𝐵14 = 𝐴14 ⊕ 𝐵11 ⊕ 𝐵9 = 1 ⊕ 1 ⊕ 1 = 1

𝐵15 = 𝐴15 ⊕ 𝐵12 ⊕ 𝐵10 = 1 ⊕ 0 ⊕ 1 = 0

𝐵16 = 𝐴16 ⊕ 𝐵13 ⊕ 𝐵11 = 1 ⊕ 0 ⊕ 1 = 0

𝐵17 = 𝐴17 ⊕ 𝐵14 ⊕ 𝐵12 = 1 ⊕ 1 ⊕ 0 = 0

𝐵18 = 𝐴18 ⊕ 𝐵15 ⊕ 𝐵13 = 1 ⊕ 0 ⊕ 0 = 1

𝐵19 = 𝐴19 ⊕ 𝐵16 ⊕ 𝐵14 = 0 ⊕ 0 ⊕ 1 = 1

𝐵20 = 𝐴20 ⊕ 𝐵17 ⊕ 𝐵15 = 0 ⊕ 0 ⊕ 0 = 0

𝐵21 = 𝐴21 ⊕ 𝐵18 ⊕ 𝐵16 = 1 ⊕ 1 ⊕ 0 = 0

𝐵22 = 𝐴22 ⊕ 𝐵19 ⊕ 𝐵17 = 1 ⊕ 1 ⊕ 0 = 0

𝐵23 = 𝐴23 ⊕ 𝐵20 ⊕ 𝐵18 = 1 ⊕ 0 ⊕ 1 = 0

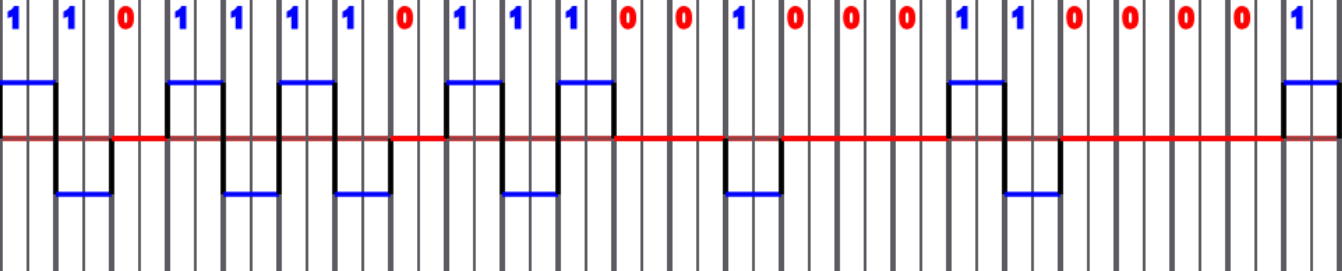
𝐵24 = 𝐴24 ⊕ 𝐵21 ⊕ 𝐵19 = 0 ⊕ 0 ⊕ 1 = 1

**Получившееся сообщение:** 1101 1110 1110 0100 0110 0001

**В шестнадцатеричном коде:** DE E4 61

**Длина сообщения:** 3 байт (24 бита)

#### Максимальное количество повторяющихся символов: 4



Верхняя граница частот: 50 МГц  
Нижняя граница частот: 12.5 МГц  
Середина спектра: 31.25 МГц  
Средняя частота: (15\*f0/1 + 2\*f0/2 + 3\*f0/3 + 4\*f0/4)/24 = 37.5 МГц  
Спектр сигнала: 337.5 МГц  
Полоса пропускания: 340 МГц

# Сравнительный анализ (логическое кодирование)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метод кодирования | Полезная пропускная способность | Спектр | Синхрон изация | Обнаружение ошибок | Реализация | Доп временные затраты |
| **Избыточное 4B/5B** | Уменьшается | Сужается | Есть | Есть | Простая | Есть |
| **Скремблир ование** | Сохраняется | Как повезет | Нет | Нет | Средней сложности | Есть |

В результате анализа двух способов логического кодирования можно сделать вывод, что избыточное кодирование оказывается наиболее предпочтительным. Да, приходится жертвовать частью пропускной способности, однако это почти всегда (в отличие от скремблирования) устраняет постоянную составляющую, позволяя сузить спектр и упростить синхронизацию. Дополнительное преимущество — возможность обнаружения ошибок за счёт так называемых «запрещённых символов» (16 вариантов). Метод также выделяется универсальностью и простой реализацией.

В случае скремблирования критически важно правильно подобрать полином, что нередко оказывается сложным, так как универсального метода выбора не существует. При этом и избыточное кодирование, и скремблирование потребуют времени на промежуточные преобразования исходных данных.

# Вывод

В ходе работы были рассмотрены разные методы физического и логического кодирования, выявлены их достоинства и недостатки. Идеальной универсальной методики не существует — всё зависит от конкретных условий и требований. Для моих задач лучшими оказались **M2** и AMI. Что до логического кодирования, эксперимент показал, что избыточное кодирование существенно поднимает нижнюю границу частоты, а скремблирование почти не повлияло на итоговые показатели (возможно, из-за неподходящего полинома). Сложность выбора полинома и отсутствие унифицированного подхода — ключевой недостаток скремблирования.