

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ЕЛЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ И.А. БУНИНА»

**И. В. Пешков**

**РАДИОПЕРЕДАЮЩИЕ УСТРОЙСТВА**  
**(НАПРАВЛЕНИЕ «РАДИОТЕХНИКА»)**

**Учебное пособие**

Елец – 2019

УДК 621.396.61

ББК 32.848

**П 23**

Печатается по решению редакционно-издательского совета  
Елецкого государственного университета имени И.А. Бунина  
от 31. 01. 2019 г., протокол № 1

Рецензенты:

*Рощупкин С.А.*, к. ф.-м.н., доцент кафедры математического  
моделирования и компьютерных технологий  
Елецкого государственного университета

*Нечаев Ю.Б.*, д-р ф.-м.н., профессор кафедры информационных систем  
Воронежского государственного университета

**И.В. Пешков**

**П 23** Радиопередающие устройства (направление «Радиотехника»): учебное  
пособие. – Елец: Елецкий государственный университет им.  
И.А. Бунина, 2019. – 58 с.

**ISBN 978-5-00151-049-9**

В методических указаниях представлены материалы лекций по дисциплине «Радиопередающие устройства». По каждой теме даются основные теоретические положения. Издание предназначено для студентов по направлению подготовки бакалавров «Радиотехника».

Предназначено для студентов вышеуказанных направлений подготовки и специальности очной и заочной форм обучения.

УДК 621.396.61

ББК 32.848

**ISBN 978-5-00151-049-9**

© Елецкий государственный  
университет им. И.А. Бунина, 2019

## **ВВЕДЕНИЕ**

В настоящее время современные беспроводные цифровые радиопередающие устройства становятся главенствующими системами для обслуживания всё более возрастающих требований к увеличивающимся трафику. Как установлено, за последнее время объем передаваемой информации по беспроводным радиоканалам превышает несколько мегабит в секунду. И этот объем продолжает свой рост. Учёными установлено, что годовой прирост передаваемой информации по беспроводному радиоканалу будет расти ежегодно на 10-20%. В связи с этим вводятся всё новые, постоянно усложняющиеся, всё более совершенные, но при этом требующие более пристального внимания устройства, а именно передатчики цифровой радиосвязи. Поэтому инженер-радиотехник в своей работе должен постоянно отслеживать новые направления развития современных технологических решений в области цифровой связи, в том числе способов реализации и проектирования, а также анализа работы передающих радиоустройств цифровой связи.

Как уже было сказано выше, в мире постоянно требуется прирост пропускной способности радиоканалов, вследствие чего плотность доступного гражданского радиочастотного спектра достигнет в скором времени предела. Это сказывается на том обстоятельстве, что каждый мобильный абонент становится активной помехой соседнему источнику радиосигнала. «Поэтому одной из наиболее существенных и актуальных проблем в системах беспроводной связи является снижение уровня активных помех. Наряду с проблемой снижения уровня помех стоит проблема увеличения количества пользователей базовой станции, работающих одновременно на одной частоте» [1].

В связи с этим вводятся новые стандарты передачи цифровой информации по радиоканалам на более высоких частотах. Можно выделить сети связи 4-го и 5-го поколений. Однако можно отметить, что общие способы построения цифровых радиопередатчиков остаются в принципе идентичными

друг к другу. Можно пояснить, что основополагающими блоками в таких устройствах служат: блок кодирования сигнала, т.е. его представление в цифровой форме, сжатие полученной информации, канальное кодирование, т.е. кодирования для обнаружения и/или исправления ошибочных бит и байт данных уже на приёмной стороне. Кроме того, зачастую вводятся блоки шифрования для защиты от злоумышленников. «В настоящее время для увеличения информационной емкости каналов связи применяются различные схемы уплотнения пользователей, основанные на разделении станций по таким параметрам, как положение в пространстве, время работы, частота или код. Задача уплотнения состоит в том, чтобы выделить каждому каналу связи положение в пространстве, время, частоту и/или код с минимумом взаимных помех и максимальным использованием характеристик среды передачи. В связи с этим зачастую вводится пространственная фильтрация на базе антенных решеток с возможностью адаптивного диаграммообразования, которое обеспечивает селекцию сигналов по направлениям прихода» [1]. Уже давно получило широчайшее распространение кодирование данных с целью пространственного мультиплексирования, т.е. разделение общего цифрового потока на несколько антенн в зависимости от качества того или иного канала и затем объединения этих потоков на приемной стороне. Таким образом, в данной работе приводятся основные положения, необходимые для понимания устройства и принципов действия современных цифровых радиопередатчиков.

## РАЗДЕЛ 1. ОБЩИЕ ВОПРОСЫ ПЕРЕДАЧИ

### Тема 1. Структура систем цифровой связи

В работе Н.М. Боева рассмотрены основные элементы цифровой системы связи на рис. 1 [2]. «Источник информации может выдавать данные для передачи по каналу связи как в цифровом виде» («современные носители цифровой информации, различные датчики с цифровым интерфейсом и т.д.»), так и в аналоговом виде (аналоговые датчики, передача звука и изображения и др.). «В независимости от типа источника информации данные должны быть представлены в как можно более сжатом цифровом виде». «Процесс эффективного преобразования данных в последовательность двоичных символов называется *кодированием источника* или *сжатием данных*». «Как правило, данные на цифровых носителях являются уже сжатыми» («например, формат цифрового кодирования звуковой информации с потерями MP3, алгоритмы сжатия видеoinформации MPEG, алгоритм сжатия изображений JPEG»), «тогда как данные с аналоговых источников информации зачастую слишком избыточны и требуют сжатия» [2].

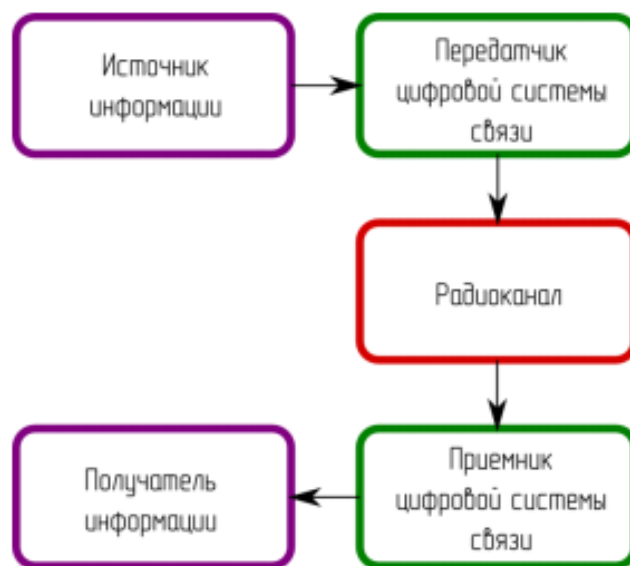


Рис. 1. Элементы цифровой системы связи

«С выхода кодера источника данные поступают на передатчик цифровой системы связи, в котором выполняется *канальное кодирование* данных и *цифровая модуляция*». «Процедура канального кодирования состоит в добавлении некоторой избыточности в цифровую последовательность данных источника информации». «На стороне приемника цифровой системы связи избыточность может быть использована для обнаружения и исправления ошибок в принимаемых данных». «Простейшим примером канального кодирования является передача одних и тех же данных несколько раз, в этом случае на приемной стороне решение принимается при помощи мажоритарной системы» (по принципу большинства). «В некоторых случаях передающая система содержит дополнительный блок кодирования, в котором выполняется *шифрование* передаваемых данных». «Под шифрованием понимаются операции преобразования информации из открытой в закрытую и обратно с целью ограничения доступа к передаваемой информации при передаче по открытым каналам связи» [2].

В качестве канала связи выступает физическая среда, в которой происходит передачи физического радиосигнала. При беспроводной связи свободное пространство выступает в качестве такой среды. Сигнал, передаваемый по каналу связи, подвержен наложению случайного шума, межсимвольным помехам, когда символы накладываются друг на друга, затуханию, умышленным помехам и другим факторам, которые вносят искажения в передаваемый сигнал.

В приемнике цифровой системы связи имеются системы синхронизации с принимаемым сигналом, цифровой демодулятор, блоки декодирования, дешифрования сигналов и интерфейс для выдачи полезных данных пользователю. При необходимости цифровой сигнал преобразуется в аналоговую форму на приемной стороне системы связи. Как правило, приемная часть системы связи является более сложной, чем передающая часть. Это в первую очередь связано с необходимостью синхронизации с принимаемым сигналом по частоте и фазе несущей волны, по частоте следования импульсов

(символьной частоте), по частоте слов и частоте кадров. Практическая реализация канального декодирования сигналов в приемнике затрачивает больше вычислительных ресурсов, таким образом являясь более дорогостоящей, чем кодирование сигнала в передатчике. Часто приемная часть системы связи вынуждена работать при очень низких соотношениях мощности сигнал к мощности шума. В этом случае должны применяться продвинутые методы цифровой обработки сигналов.

В современных системах связи выделяются симплексные, полудуплексные и дуплексные каналы. При организации симплексного канала связи данные передаются от передатчика к приемнику в одном направлении. В качестве примеров таких цифровых систем связи можно рассмотреть цифровое телевидение, цифровое радио и передачу данных с датчиков. Если имеется режим дуплексной связи, то в этом случае данные могут передаваться в двустороннем режиме одновременно за счет физического разделения сигналов (частотного, кодового, пространственного, поляризационного разделения). Последний, полудуплексный режим работы системы связи, реализован путем разделения потоков данных во времени, т.е. сигналы передаются по одному и тому же каналу связи в разное время.»

## **Тема 2. Пропускная способность систем связи**

Ширина полосы канала связи  $C$  с аддитивным белым гауссовским шумом (АБГШ) является функцией средней мощности принятого сигнала  $S$ , средней мощности шума  $N$  и ширины полосы  $W$ . Верхняя граница ширины полосы канала связи», т.е. пропускная способность определяется по «теореме Шеннона-Хартли [2]:

$$C = W \log_2 \left( 1 + \frac{S}{N} \right),$$

где  $C$  – пропускная способность канала связи, бит/сек;  
 $W$  – ширина полосы пропускания системы, Гц;  
 $S$  – средняя мощность принятого сигнала, Вт;  
 $N$  – средняя мощность шума, Вт.

Теоретически, информация по каналу связи может передаваться с произвольно малой ошибкой при любой скорости передачи данных  $R$ , удовлетворяющей условию  $R \leq C$ , которое достигается с использованием сложных методов кодирования. Если  $R > C$ , передача данных с произвольно малой вероятностью ошибки невозможна.

При сравнении различных типов модуляции они обычно работают не с отношением сигнал / шум, а с отношением энергии бита к плотности мощности шума, которое определяется следующим образом:

$$\frac{E_b}{N_0} = \frac{S}{N} \left( \frac{W}{R} \right),$$

где  $E_b$  – энергия бита, Вт/бит/сек;  
 $N_0$  – плотность мощности шума, Вт/Гц.

Таким образом, выражение верхней границы пропускной способности канала связи может быть модифицировано:

$$\frac{E_b}{N_0} = \frac{W}{C} 2^{\frac{C}{W}} - 1.$$

На рис. 2 представлен график отношения нормированной ширины полосы пропускания сигнала  $W / C$  к отношению энергии бита к плотности мощности шума  $E_b / N_0$ . Как видно из рисунка, существует нижнее предельное значение  $E_b / N_0$ , при котором безошибочная передача информации не может осуществляться при любой скорости передачи. Это значение  $E_b / N_0$  называется пределом Шеннона»:

$$\frac{E_b}{N_0} = \ln(2) = -1,59 \text{ дБ} . \quad W/C \text{ (Гц/дБм/Гц)}$$



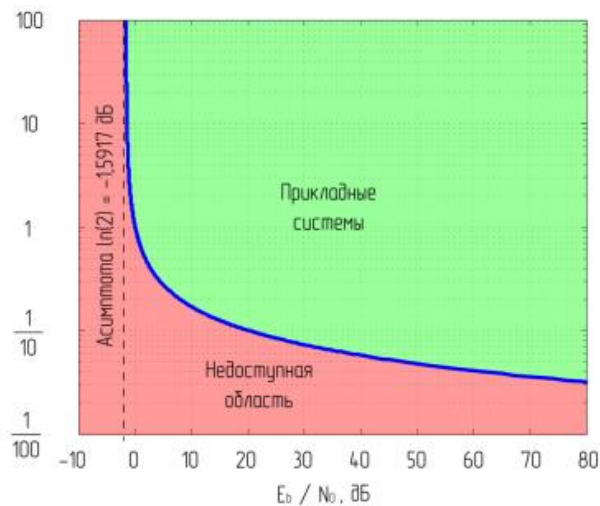


Рис. 2. Зависимость нормированной полосы пропускания канала от отношения энергии бита к плотности мощности шума

Шеннон теоретически доказал существование кодов, которые могут повысить вероятность битовой ошибки или уменьшить требуемое значение  $E_b / N_0$  от уровней некодированных систем двоичной модуляции до уровней, приближающихся к предельной кривой. Одной из основных задач разработчика современных цифровых систем связи является повышение спектральной эффективности модуляции, то есть снижение отношения  $W / R$  (рис. 2) до теоретического предела, что возможно при использовании современных методов канального кодирования. В настоящее время лучший подход к пределу позволяет использовать турбокодирование».

### Тема 3. Передача данных в канале с ограниченной полосой

Современные системы связи работают в условиях острого дефицита частотных ресурсов, организация доступа к услугам сети абонентов многоканальной системы связи требует ограничения спектра сигнала от каждого абонента передатчика с целью исключения взаимного влияния сигнала на сигналы других абонентов той же сети и сигналы других систем связи. Таким образом, возникает необходимость максимально ограничить спектр сигнала передатчика и повысить его спектральную эффективность при этом

постоянно требуется увеличить скорость передачи данных цифровыми системами связи.

Для ограничения спектра сигнала используется формирующий фильтр. Известно, что ограничение спектра сигнала приводит к искажениям во временной области. На рис. 3 показан пример фильтрации произвольного сигнала в виде прямоугольных импульсов с бесконечным спектром.

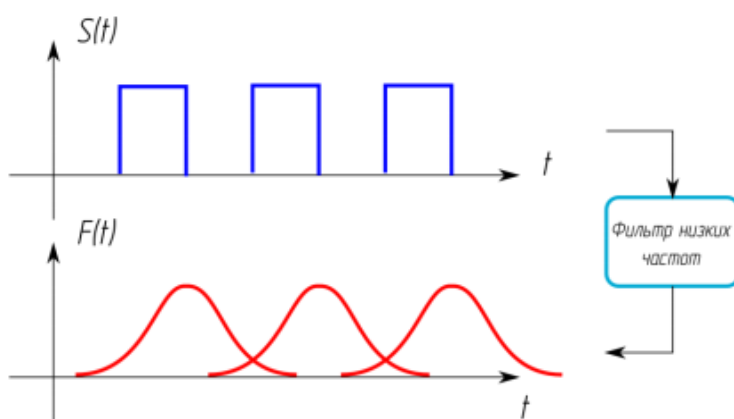


Рис. 3. Искажение формы полезного сигнала при ограничении его спектра

Сглаживание краев прямоугольных импульсов приводит к их расширению и взаимному перекрытию соседних импульсов, этот эффект называется межсимвольной интерференцией (MSI). Межсимвольные помехи или интерференции позволяют сузить спектр передаваемого сигнала, когда качество передачи информации ухудшается за счет увеличения вероятности битовой ошибки.

Межсимвольные интерференции были исследованы Гарри Найквистом. Он доказал, что минимальная теоретическая пропускная способность системы, необходимая для передачи данных со скоростью  $R_s$  символов в секунду без влияния межсимвольных помех, составляет  $R_s / 2$  (теорема Найквиста об ISI). Для передачи данных без MSI необходимо использовать идеальный фильтр Найквиста, амплитудно-частотная характеристика (АЧХ) которого имеет прямоугольную форму, как показано на рис. 4».

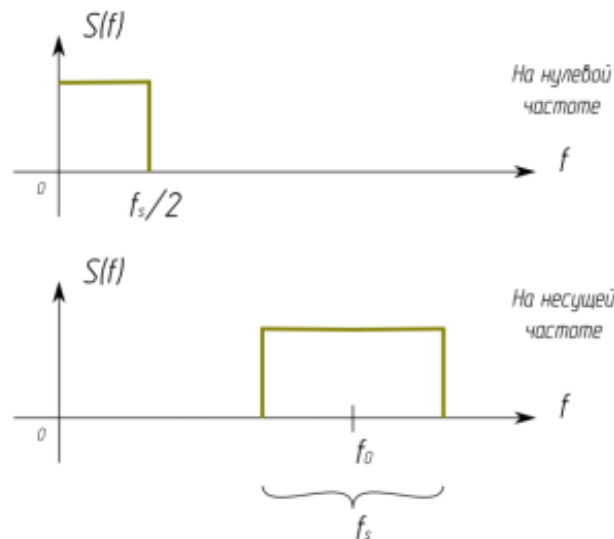


Рис. 4. АЧХ идеального фильтра Найквиста

Импульсный отклик или характеристика (ИХ) фильтра нижних частот Найквиста находится с помощью обратного преобразования Фурье и является функцией  $\text{sinc}$  (рис. 5).

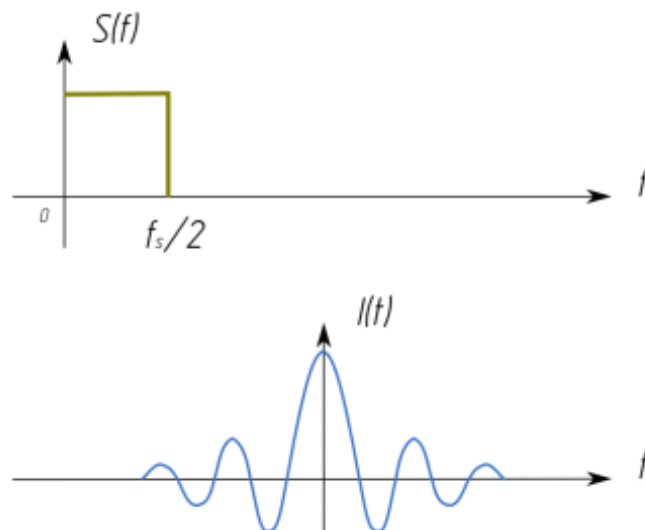


Рис. 5. АЧХ фильтра Найквиста и его ИХ

«Импульс, описываемый этой функцией  $\text{sinc}$ , называется идеальным импульсом Найквиста, он имеет бесконечную длительность во временной области, главном лепестке и бесконечном количестве боковых лепестков. На рис. 6 показывает принцип работы формирующего фильтра Найквиста» [2].

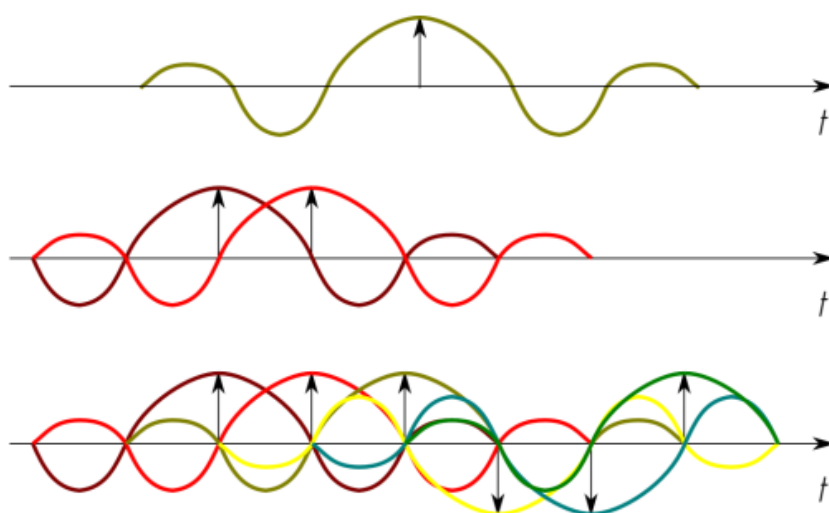


Рис. 6. Формирование сигнала в фильтре Найквиста

«На вход фильтра Найквиста принимаются дельта-импульсы, соответствующие информационному потоку битов, каждый из которых формирует отклик фильтра в виде их (верхний график на рисунке 6). Как видно на втором графике, отклик фильтра на второй входной дельта-импульс накладывается на первый таким образом, что отсутствует взаимное влияние на единицы выборки импульсов. На нижнем графике рис. На рисунке 6 показано, что отклики от всех соседних импульсов равны нулю в моменты отбора проб для каждого конкретного импульса» [2].

«Очевидно, что реализация фильтра Найквиста с бесконечным ИМ невозможна, поэтому, чтобы ограничить продолжительность ИМ, функция  $\text{sinc}$  умножается на оконную (весовую) функцию. Наиболее популярными реализациями фильтров Найквиста в цифровых системах связи являются фильтры с характеристикой приподнятого косинуса и корня приподнятого косинуса, которые будут рассмотрены позже» [1].

#### Тема 4. Передающая часть систем цифровой связи

На рис. 7 показаны основные этапы обработки сигнала в передающей части системы цифровой связи.

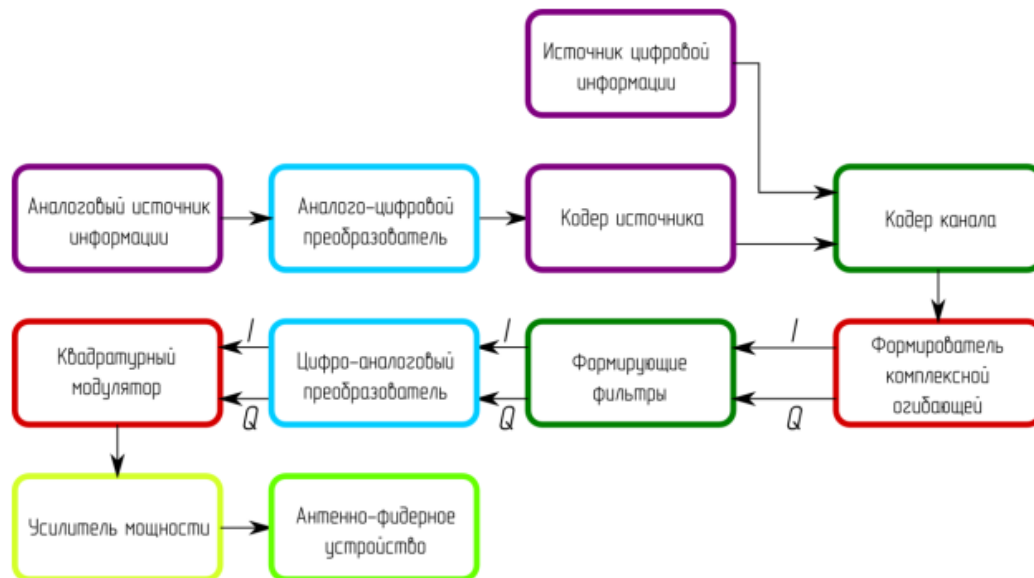


Рис. 7. Общая структура передающей части цифровой системы связи

«Как правило, в современных системах связи используется аналого-цифровой преобразователь (АЦП) для фильтрации и оцифровывания аналоговых сигналов источников информации. В условиях постоянно возрастающего трафика сигнал дополнительно обрабатывается и сжимается. Цифровые данные для передачи передаются в каналный кодер для повышения помехоустойчивости системы связи и обеспечения вероятности битовой ошибки при приеме сигнала на заданном уровне. Используя цифровой модулятор, генерируется сложная огибающая сигнала, которая поступает в формирующие фильтры, которые ограничивают спектр выходного сигнала. Двойной цифроаналоговый преобразователь (ЦАП) позволяет переключаться с системы с дискретным временем на систему обработки аналоговых сигналов. Аналоговый квадратурный модулятор передает спектр сигнала на заданную несущую или промежуточную частоту, после чего сигнал усиливается и подается на антенно-фидерное устройство (AFU)» [2].

Рассмотрим более подробно возможные варианты практической реализации передающей части цифровой системы связи. На рисунке 8 показана архитектура передатчика с аналоговой квадратурной обработкой сигнала.

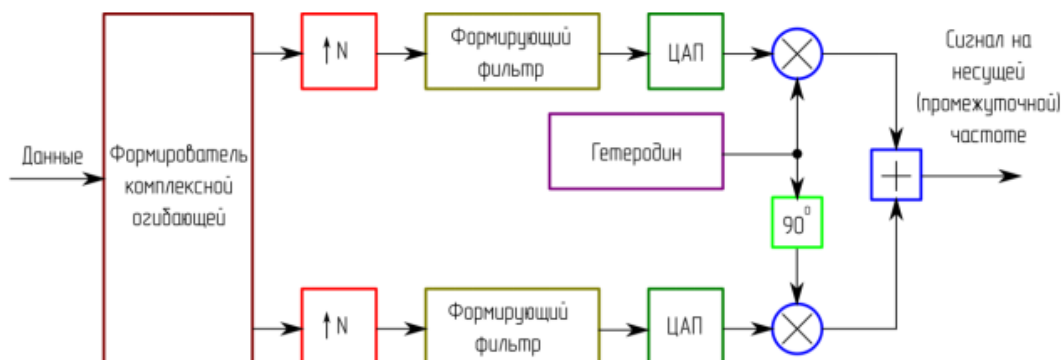


Рис. 8. Архитектура передающей части цифровой системы связи с аналоговой квадратурной обработкой сигнала

«Преимущество этого подхода заключается в относительно низких тактовых частотах квадратурных ЦАП, поскольку преобразование осуществляется на нулевой промежуточной частоте. Как правило, тактовая частота ЦАП выбирается в диапазоне от 2 до 8 отсчётов на символ. Недостатком данной схемы является аналоговая квадратурная обработка сигналов, которая принципиально не позволяет создавать смесители с одинаковыми характеристиками, строгую разность фаз на выходах квадратурного гетеродина, фильтры нижних частот с одинаковой частотной характеристикой и т. д. Эти недостатки приводят к искажению формы передаваемого созвездия, которое необходимо компенсировать передающей стороне путем введения обратной связи или на приемной стороне с использованием адаптивных эквалайзеров. Преимущество данной схемной реализации заключается в некоторой простоте реализации как аналоговой квадратурной передаче спектра сигнала, так и схемы фильтрации выходного сигнала передатчика и схему преобразователя частоты в целом» [2].

Другой подход к реализации передатчика предусматривает квадратурную обработку сигналов только в цифровом виде (рис. 9).

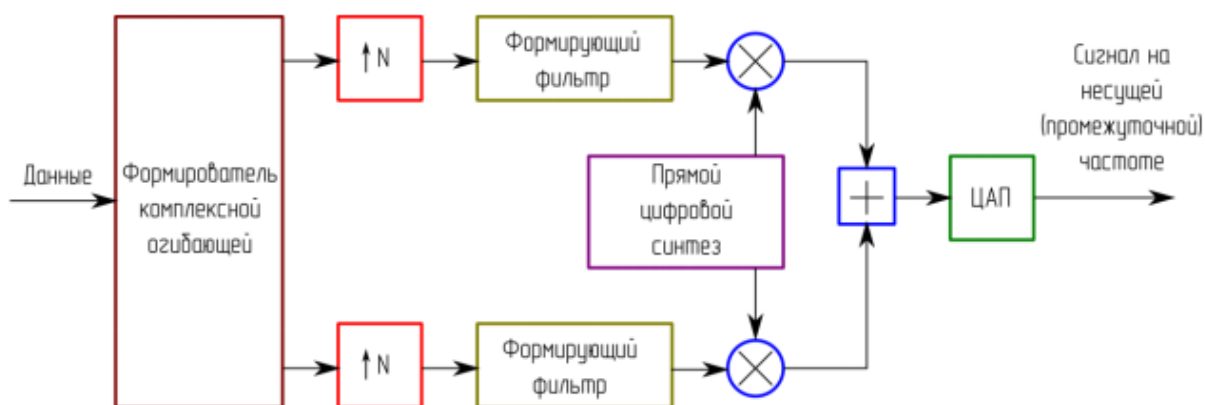


Рис. 9. Архитектура передающей части цифровой системы связи с цифровой квадратурной обработкой сигнала

«Полностью цифровая схема реализации квадратурной обработки сигналов заключается в нивелировании проблем аналоговых схем, таких как отсутствие синхронизации и т.д. Цифровой вариант обработка сигналов позволяет создать квадратурный локальный генератор с идеальной разностью фаз и одинаковой амплитудой выходных сигналов, а также операции умножения и фильтрация в цифровом виде выполняется абсолютно одинаково для обоих каналов  $I$  и  $Q$ . Платой за преимущества является требование к высокочастотному ЦАП. В некоторых случаях реализации сверхширокополосных систем связи может возникнуть ситуация, когда спектр сигнала не подходит даже для самых современных ЦАП в первой зоне Найквиста, и этот подход неприемлем» [2].

## РАЗДЕЛ 2. СТРУКТУРА ПЕРЕДАЮЩЕЙ ЧАСТИ

### Тема 5. Аналого-цифровое преобразование

Как уже выше показано, не всегда источник сигнала является цифровым. Зачастую информация представляется в аналоговом виде, тогда возникает необходимость его преобразования в цифровой вид при помощи АЦП. При этом преобразование аналого-цифра сводится к дискретизации сигнала во времени и квантование по уровню (рис. 10).

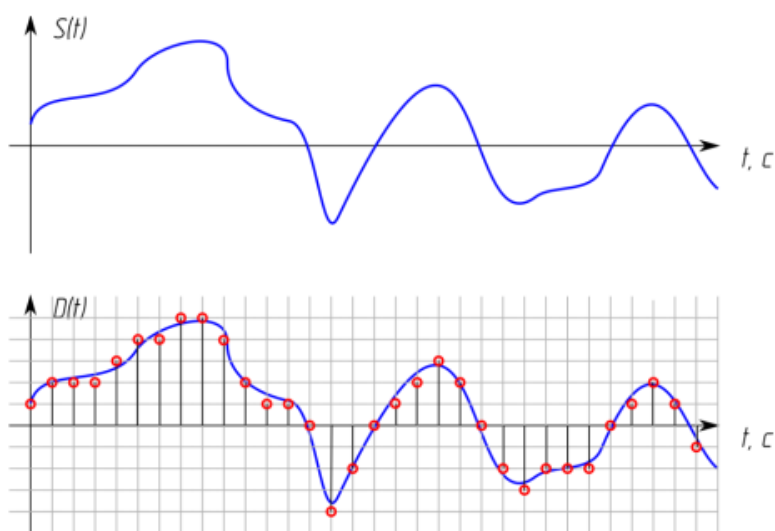


Рис. 10. Дискретизация сигнала по времени и квантование по уровню

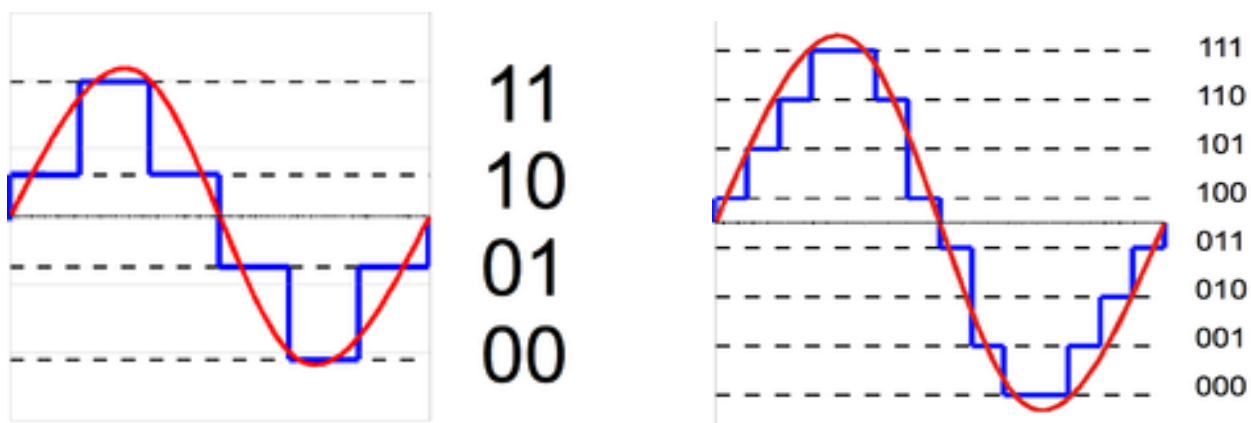
Для корректной дискретизации сигнала во времени необходима фильтрация сигнала, после чего дискретизация выполняется в соответствии с *теоремой Котельникова* (Найквиста), согласно которой частота дискретизации должна быть в два и более раз превышать максимальную частоту в спектре исходного сигнала и тогда этот сигнал может быть полностью восстановлен, т.е.:

$$f_d > 2f_{\max}.$$

Аналого-цифровой преобразователь (АЦП) может быть смоделирован как два процесса: выборка и квантование. Выборка преобразует изменяющийся во времени сигнал напряжения в дискретный сигнал, последовательность действительных чисел. Квантование заменяет каждое действительное число



приближением из конечного набора дискретных значений. Чаще всего эти дискретные значения представляются в виде слов с фиксированной запятой. Хотя возможно любое количество уровней квантования, общие длины слова составляют 8-битный (256 уровней), 16-битный (65 536 уровней) и 24-битный (16,8 млн. Уровней). Квантование последовательности чисел приводит к последовательности ошибок квантования, которая иногда моделируется как аддитивный случайный сигнал, называемый шумом квантования, из-за его стохастического поведения. Чем больше уровней использует квантователь, тем ниже его мощность шума квантования.



Шум квантования — это модель ошибки квантования, вносимая квантованием в аналого-цифровое преобразование (АЦП) в телекоммуникационных системах и обработке сигналов. Это ошибка округления между напряжением аналогового входа на АЦП и оцифрованным выходным значением. Это может быть смоделировано несколькими различными способами. Когда входной сигнал представляет собой синусоидальную волну с полной амплитудой, распределение сигнала больше не является равномерным, и коэффициент сигнал-шум квантования (SQNR) рассчитывается:

$$SQNR \approx 1.761 + 6.02 \cdot Q \text{ dB}$$

где  $Q$  – количество бит квантования. В этом случае 16-разрядный АЦП имеет максимальное отношение сигнал / шум 98,09 дБ.

Мощность шума квантования может быть получена из

$$N = \frac{(\delta v)^2}{12} W$$

где  $(\delta v)^2$  – напряжение, приходящееся на один уровень АЦП. Пусть диапазон входных значений от -1В до 1В, разрядность двоичного АЦП 8 бит:  $8^2 = 256$  уровней квантования. Тогда разрешение двоичного АЦП по напряжению:  $(1 - (-1)) / 256 = 0,0078125$  вольт = 7,81 мВ в одном разряде АЦП.

Однако приведенные расчеты предполагают наличие полностью заполненного входного канала. Если это не так – если входной сигнал мал – относительное искажение квантования может быть очень большим. Чтобы обойти эту проблему, можно использовать аналоговые компрессоры и расширители, но они также вносят большие искажения, особенно если компрессор не соответствует расширителю.

### *5.1. Канальное кодирование*

«Процедура канального кодирования заключается в добавлении избыточности к передаваемому информационному сообщению, которое можно использовать для обнаружения и исправления ошибок во время приема. Сегодня существует множество различных классов кодов с исправлением ошибок, которые могут различаться по алгоритмам кодирования и декодирования, эффективности использования энергии и многим другим параметрам. На рис. 11 показывает классификацию кодов, исправляющих ошибки» [2].

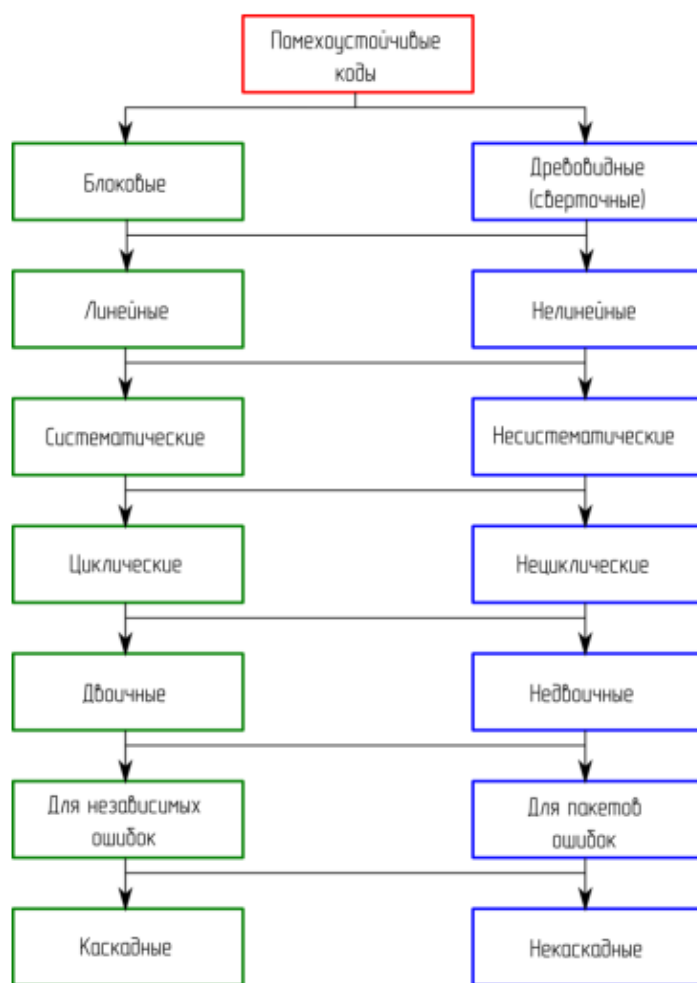


Рис. 11. Классификация помехоустойчивых кодов

«Блочное кодирование реализуется в одном блоке передаваемых данных. Кодер передатчика в данном случае не имеет памяти и преобразует входные  $k$  символов в выходные  $n$  символов (т.е.  $n > k$ ). Древовидное кодирование осуществляется с непрерывным потоком информации на входе кодера. Набор из  $k_0$  входных символов поступает в кодировщик древовидного кода, а набор из  $n_0$  символов появляется на выходе. Одной из характеристик блочного и древовидного кодирования является скорость кодирования  $r = k / n$ . В зависимости от возможного количества значений входных символов различаются двоичные и недвоичные коды. Линейные коды имеют следующее свойство: сумма двух кодовых слов дает на выходе третье кодовое слово. Сверточные коды обычно называют кодами линейного дерева. Без труда возможно выделить коды, которые исправляют случайные или независимые

ошибки, и коды, предназначенные для исправления пакетов ошибок, которые могут возникать в неблагоприятных условиях передачи данных» [2].

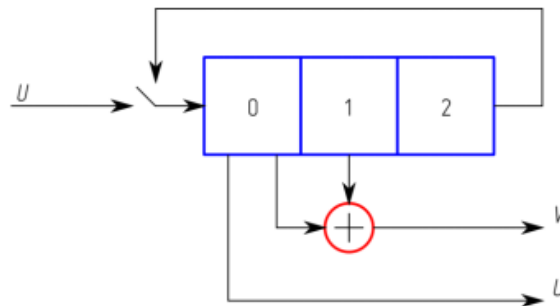


Рис. 12. Блоковый систематический кодер

На рис. 12 приведен пример кодера *систематического* (на выходе кодера можно выделить две части: данные и проверочную часть) блочного кода с параметрами: количество информационных символов  $k = 3$ , количество кодовых символов  $n = 6$ , кодовая скорость  $r = 1/2$ .

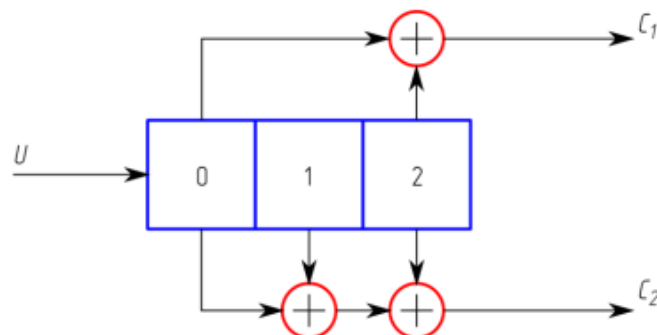


Рис. 13. Сверточный несистематический кодер

На рисунке 13 приведен пример кодера несистематического сверточного кода.

## Тема 6. Сжатие информации

«При обработке сигналов сжатие данных, исходное кодирование, или уменьшение скорости передачи битов включает в себя кодирование информации с использованием меньшего количества битов, чем в исходном представлении. Сжатие может быть с потерями или без потерь. Сжатие без потерь уменьшает количество битов, выявляя и устраняя статистическую

избыточность. При сжатии без потерь информация не теряется. Сжатие с потерями уменьшает биты, удаляя ненужную или менее важную информацию» [3].

«Процесс уменьшения размера файла данных часто называют сжатием данных. В контексте передачи данных это называется исходным кодированием; кодирование выполняется в источнике данных перед их сохранением или передачей. Исходное кодирование не следует путать с канальным кодированием, для обнаружения и исправления ошибок или линейным кодированием, средством отображения данных на сигнал» [3].

«Сжатие полезно, потому что оно уменьшает ресурсы, необходимые для хранения и передачи данных. Вычислительные ресурсы потребляются в процессе сжатия и, как правило, при обращении процесса (декомпрессия). Сжатие данных зависит от сложности пространства и времени. Например, схема сжатия для видео может потребовать, чтобы дорогостоящее аппаратное обеспечение распаковывало видео достаточно быстро, чтобы его можно было просматривать по мере его распаковки, и возможность полностью распаковывать видео перед просмотром может быть неудобной или требовать дополнительной памяти. Разработка схем сжатия данных включает компромиссы между различными факторами, включая степень сжатия, количество вносимых искажений (при использовании сжатия данных с потерями) и вычислительные ресурсы, необходимые для сжатия и распаковки данных» [3].

**Сжатие без потерь** – это класс алгоритмов сжатия данных, который позволяет полностью восстановить исходные данные из сжатых данных. С другой стороны, сжатие с потерями позволяет восстанавливать только аппроксимацию исходных данных, хотя обычно с улучшенными коэффициентами сжатия (и, следовательно, уменьшенными размерами мультимедиа).

Сжатие данных без потерь используется во многих приложениях. Например, он используется в формате файла ZIP и в инструменте GNU gzip. Он

также часто используется в качестве компонента в технологиях сжатия данных с потерями (например, предварительная обработка стереосигналов без потерь в средней / боковой части с помощью кодеров MP3 и других аудиокодеров с потерями).

Сжатие без потерь используется в тех случаях, когда важно, чтобы исходные и распакованные данные были идентичными или когда отклонения от исходных данных были бы неблагоприятными. Типичными примерами являются исполняемые программы, текстовые документы и исходный код. Некоторые форматы файлов изображений, такие как PNG или GIF, используют только сжатие без потерь, в то время как другие, такие как TIFF и MNG, могут использовать методы без потерь или с потерями. Аудиоформаты без потерь чаще всего используются для целей архивирования или производства, в то время как аудиофайлы с меньшими потерями обычно используются на портативных проигрывателях и в других случаях, когда пространство для хранения ограничено или точная репликация звука не требуется.

В информационной технологии **сжатие с потерями** или необратимое сжатие - это класс методов кодирования данных, которые используют неточные аппроксимации и частичное отбрасывание данных для представления контента. Эти методы используются для уменьшения размера данных для хранения, обработки и передачи контента. Различные версии фотографии кота справа показывают, как более высокие степени приближения создают более грубые изображения при удалении большего количества деталей. Это противоположно сжатию данных без потерь (обратимому сжатию данных), которое не ухудшает данные. Объем сокращения данных, возможный при использовании сжатия с потерями, намного выше, чем с помощью методов без потерь.

Хорошо продуманная технология сжатия с потерями часто значительно уменьшает размеры файлов, прежде чем конечный пользователь заметит ухудшение. Даже когда это заметно для пользователя, может потребоваться дальнейшее сокращение данных (например, для связи в реальном времени, для уменьшения времени передачи или для уменьшения потребностей в хранении).

Наиболее широко используемый алгоритм сжатия с потерями – это дискретное косинусное преобразование (DCT), впервые опубликованное Насиром Ахмедом, Т. Натараджаном и К. Р. Рао в 1974 году.

Сжатие с потерями чаще всего используется для сжатия мультимедийных данных (аудио, видео и изображений), особенно в таких приложениях, как потоковое мультимедиа и интернет-телефония. Напротив, сжатие без потерь обычно требуется для текстовых файлов и файлов данных, таких как банковские записи и текстовые статьи. Может быть выгодно создать основной файл без потерь, который затем можно будет использовать для создания дополнительных копий. Это позволяет избежать создания новых сжатых копий исходного файла с потерями, что может привести к появлению дополнительных артефактов и дальнейшей ненужной потере информации.

«При сжатии с потерями осуществляют преобразование Фурье для разложения сигнала на составляющие его спектра. Известно, что имеются некоторые недостатки слухового аппарата человека, которые позволяют кодировать сигнал таким образом, чтобы слушатель не чувствовал никакой разницы по сравнению с реальным сигналом, хотя эта разница будет очень заметна на осциллографе» [4]. «В частности, маскирование частоты – это способность громких звуков определенного частотного диапазона «скрывать» более тихие звуки других диапазонов (которые будут слышны при отсутствии громких звуков). Это связано с тем, что с появлением очень громкого звука усиление человеческого уха резко уменьшается, и после прекращения работы отбойных молотков требуется время, чтобы вернуться в нормальное состояние. Этот эффект называется временной маскировкой» [4].

«На рис. 14а показан график, показывающий среднюю зависимость слухового порога от частоты звука. Наиболее очевидный вывод, который можно сделать, взглянув на эту кривую, состоит в том, что нет необходимости кодировать частоты, амплитуды которых ниже порога слышимости» [4]. «Это отражено на графике на рис. 14, б. Из последнего наблюдения мы можем сделать следующий вывод: зная, какие сигналы маскируются более мощными

сигналами на соседних частотах, мы можем пренебречь соответствующими частотами и не кодировать их, тем самым сохраняя биты. Если посмотреть на график из рис. 14, б очевидно, что сигналом с частотой 125 Гц можно полностью пренебречь, и никто не заметит разницы. Даже после того, как звук громкого сигнала прекращается в любом частотном диапазоне, знание свойств временной маскировки позволяет вам некоторое время продолжать игнорировать кодирование этой частоты (пока слух настроен на более низкую мощность звука). Таким образом, суть алгоритмов MP3 и AAC заключается в расширении сигнала в серии Фурье для получения силы звука на каждой частоте с последующей передачей исключительно немаскированных частот, закодированных с минимально возможным количеством битов» [4].

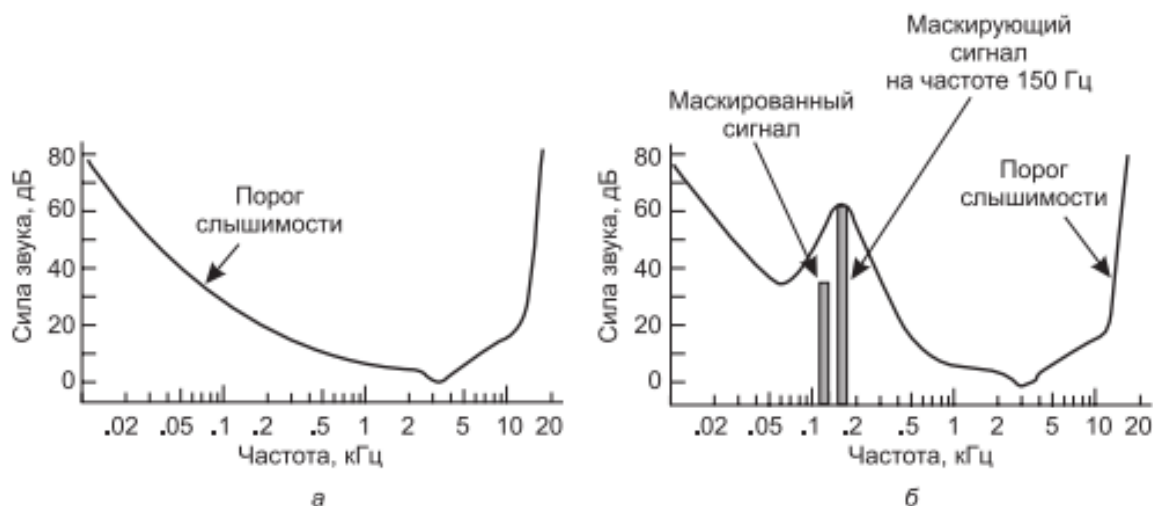


Рис. 14. Порог слышимости как функция от частоты (а);  
эффект маскирования (б)

«Теперь, ознакомившись с основными типами и подходами к сжатию информации, рассмотрим ход выполнения самого кодирования. Вначале осуществляется разложения на спектральные составляющие с помощью преобразования Фурье. Затем информация о частоте закладывается в психоакустическую модель для определения маскирующих частот. Следующим шагом является распределение всего имеющегося доступного объема бит между частотными составляющими. В этом случае большее количество битов



назначаются частоте или диапазону частот с наибольшей немаскированной спектральной мощностью, меньшее количество выделяется немаскированным диапазонам с меньшей спектральной мощностью, и маскированным диапазонам биты не присваиваются вовсе. Наконец, битовые последовательности шифруются с использованием алгоритма кодирования без потерь Хаффмана, который назначает короткие коды символам, которые появляются чаще всего, и длинные – редко появляющимся» [4].

### *6.1. Кодирование Хаффмана*

Дэвид Хаффман, аспирант Массачусетского технологического института, был одним из первых, кто предложил в 1952 году алгоритм эффективного кодирования информации. Идея алгоритма заключается в том, что если известны вероятности появления символов в каком-либо потоке информации, тогда достаточно просто описать процедуру построения кодов переменной длины, состоящую из целого числа битов. Символам, скорее всего, будут назначены более короткие коды. Такой способ кодирования информации получил название - Коды Хаффмана – и они обладают т.н. свойством префикса (другими словами ни одно кодовое слово не является продолжением другого), что позволяет их уникально декодировать.

Классический алгоритм Хаффмана на входе получает таблицу, в которой указывается, как часто встречаются символы в сообщении. Далее на основании этой таблицы строится древо кодирования Хаффмана.

1. Подсчитывают символы входного алфавита и строят список свободных узлов.
2. Затем каждому узлу присваивается вес, который может быть равен либо вероятности, либо количеству вхождений символа в сжимаемое сообщение. Выстраивают узлы в порядке убывания вероятностей.
3. Выбираются два свободных узла дерева с наименьшими весами.
4. Создается новый узел с весом, равным их общему весу.

5. Заново созданный узел добавляется в список свободных узлов в порядке убывания вероятностей, а два его потомка удаляются из списка.

6. Одной ветви, которая исходит из объединенного узла, присваивается бит 1, другой - бит 0. Значения битов ветвей, происходящих из корня, не зависят от весов потомков.

7. Шаги, начиная со второго, повторяются до тех пор, пока в списке не останется только один свободный узел, который в свою очередь будет считаться корнем дерева.

Допустим, у нас есть следующая таблица соответствия символ-частота:

Символ	А	Б	В	Г	Д
Частота	15	7	6	6	5

Этот процесс обычно представляется как построение дерева, корнем которого является символ с суммой вероятностей объединенных символов, полученных путем объединения символов из последнего шага, его  $n$ -потомками являются символы из предыдущего шага и т.д.

Чтобы определить код для каждого из символов, включенных в сообщение, мы должны перейти от листа дерева, соответствующего текущему символу, к его корню, накапливая биты при перемещении по ветвям дерева (первая ветвь в пути соответствует до наименее значимого бита). Полученная таким образом последовательность битов представляет собой код этого символа, записанный в обратном порядке.

Для этой таблицы символов коды Хаффмана будут выглядеть следующим образом.

Символ	А	Б	В	Г	Д
Код	0	100	101	110	111

Как известно, что ни один из полученных по вышеописанной схеме кодов не является началом другого, они могут быть однозначно декодированы при чтении их из входного потока. Видно также, что чем больше вероятность появления символа «А», то он имеет наименьшее количество бит, а наиболее редкий символ Д – наибольшим.

Несложно подсчитать, что длина исходного сообщения составляет 117 87 бит. После использования новых кодов общая длина сообщения стала = 87 бит. Заметим, что энтропия источника, который произвел символы с равными обозначенными изначально частотами приблизительно 2.1858 бит на символ, т.е., то есть избыточность кода Д. Хаффмана, построенного для такого источника, понимаемого как разница между средним числом битов на символ и энтропией меньше 0,05 бит на символ.

Классический алгоритм Хаффмана имеет ряд очень важных недостатков.

1) декодер должен знать таблицу частот, которая на стороне передатчика. Поэтому таблица частот должна быть отправлена перед данными, что может обратно увеличить общую длину передачи. Снижается быстродействие, т.е. необходимы два прохода по сообщению до начала фактического кодирования: один для построения модели сообщения (таблица частот и Н-дерево), а другой для фактического кодирования.

2) избыточность кодирования исчезает только в тех случаях, когда вероятности кодируемых символов имеют обратные степени 2.

3) для источника с энтропией не более 1 (например, для двоичного источника) прямое применение кода Хаффмана бессмысленно.

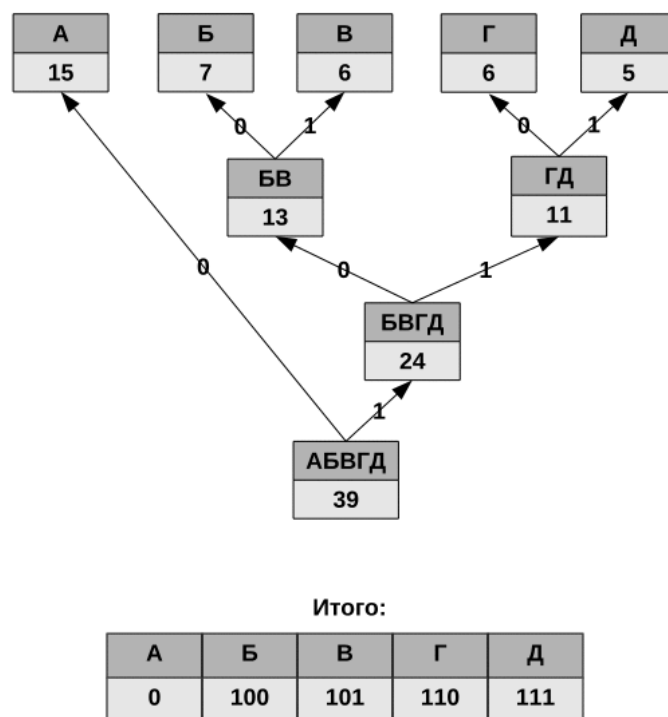


Рис. 15. Дерево Хаффмана

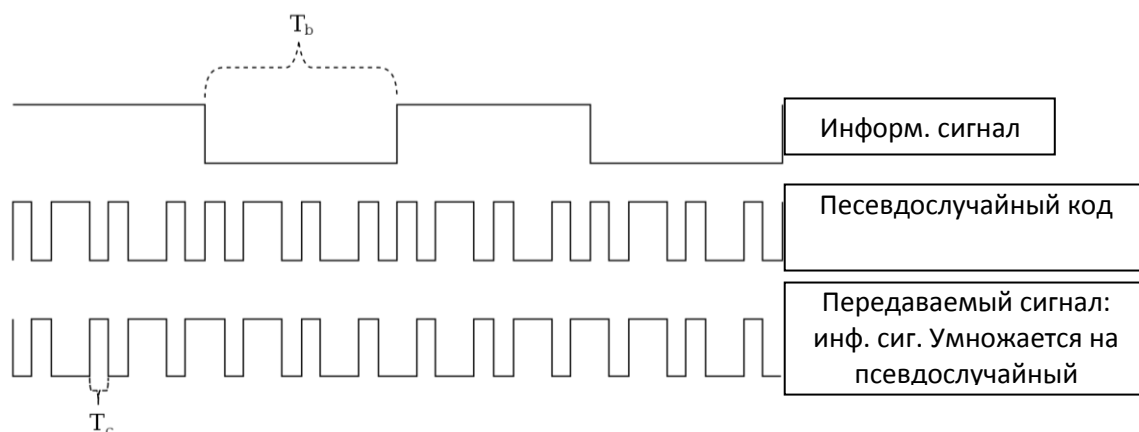
## Тема 7. Канальное кодирование. Множественный доступ

«Множественный доступ с кодовым разделением (CDMA) – это метод доступа к каналу, используемый различными технологиями радиосвязи. CDMA является примером множественного доступа, где несколько передатчиков могут отправлять информацию одновременно по одному каналу связи. Это позволяет нескольким пользователям совместно использовать полосу частот. Чтобы разрешить это без чрезмерных помех между пользователями, CDMA использует технологию расширенного спектра и специальную схему кодирования (где каждому передатчику назначается код). CDMA используется в качестве метода доступа во многих стандартах: IS-95» [5], 3GMA CDMA2000, стандарт 3G.

«В CDMA каждый битовый интервал разделен на  $m$  коротких периодов, называемых чипами или чипами. Как правило, 64 или 128 чипов расположены в битовом интервале. В нашем примере мы предполагаем, что для простоты битовый интервал содержит только 8 чипов на бит. Каждая станция имеет уникальный  $m$ -разрядный код, называемый элементарной

последовательностью. Удобнее использовать биполярную запись в виде последовательности -1 и +1» [4].

«CDMA равномерно расширяет полосу пропускания данных для одной и той же передаваемой мощности. Расширяющий код представляет собой псевдослучайный код, который имеет узкую функцию неопределенности, в отличие от других узких импульсных кодов. В CDMA локально сгенерированный код работает с гораздо большей скоростью, чем передаваемые данные. Данные для передачи объединяются побитовым XOR (исключающее ИЛИ) с более быстрым кодом. На рисунке показано, как генерируется сигнал с расширенным спектром. Сигнал данных с длительностью импульса  $T_b$  (период символа) кодируется с помощью операции XOR сигналом кода с длительностью импульса  $T_c$  (период чипа). (Примечание: пропускная способность пропорциональна  $1 / T$ , где  $T$  = битовое время.) Следовательно, пропускная способность сигнала данных равна  $1 / T_b$ , а ширина полосы сигнала расширенного спектра равна  $1 / T_c$ . Поскольку  $T_c$  намного меньше, чем  $T_b$ , ширина полосы сигнала с расширенным спектром намного больше, чем ширина полосы исходного сигнала. Соотношение  $T_b / T_c$  называется коэффициентом расширения или коэффициентом усиления при обработке и в определенной степени определяет верхний предел общего числа пользователи поддерживаются одновременно базовой станцией» [5].



«Каждый передатчик имеет свой уникальный вектор, но метод построения переданного вектора идентичен. Обозначим символом  $S$  вектор длины  $m$  для станции  $S$ , а символом  $\bar{S}$  - дополнение  $S$ . Все элементарные

последовательности попарно **ортогональны**. Мы имеем в виду, что нормированное скалярное произведение двух различных элементарных последовательностей  $S$  и  $T$  (пишется  $S \bullet T$ ) равно 0. Известно, как генерировать такие последовательности с помощью метода, известного как **коды Уолша**. Используя математическую запись, можно выразить вышесказанное таким образом» [4]:

$$S \bullet T \equiv \frac{1}{m} \sum_{i=1}^m S_i T_i = 0.$$

Попросту говоря, сколько всего пар, столько и разных пар. Это свойство ортогональности мы строго докажем чуть позже. Обратите внимание: если  $S \bullet T = 0$ , то и  $S \bullet \bar{T}$  также равно 0. Нормированное скалярное произведение любой элементарной последовательности на саму себя равно 1:

$$S \bullet S = \frac{1}{m} \sum_{i=1}^m S_i S_i = \frac{1}{m} \sum_{i=1}^m S_i^2 = \frac{1}{m} \sum_{i=1}^m (\pm 1)^2 = 1.$$

Это действительно так, поскольку каждое из  $m$  слагаемых суммы равно 1, поэтому вся сумма равна  $m$ . Обратите также внимание на то, что  $S \bullet \bar{S} = -1$ .

«Во время каждого битового интервала станция может либо передавать 1, посылая свою элементарную последовательность, либо передавать 0, посылая дополнение к последовательности, либо она может молчать и ничего не передавать. Предположим, что все станции синхронизированы во времени, то есть все последовательности начали передаваться в один и тот же момент» [4].

Когда две или более станций пытаются передавать одновременно, их биполярные сигналы линейно суммируются. Например, если во время передачи одного элементарного сигнала три станции отправили +1, а одна отправила -1, то результат будет +2. Мы можем рассматривать это как сумму напряжений: три станции имеют выход +1 В, а одна имеет выход -1 В. В результате сложения мы получаем +2 В.

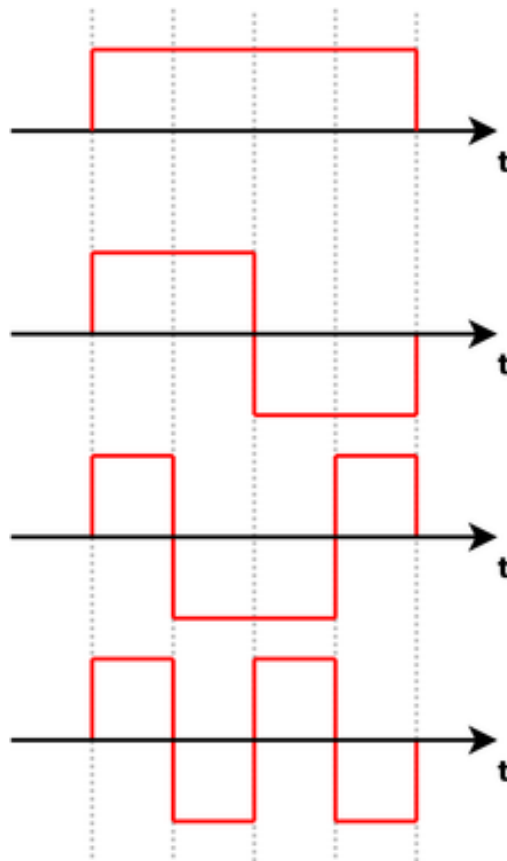


Рис. 16. Последовательность: а – двоичные элементарные последовательности;  
 б – биполярные двоичные элементарные последовательности;  
 в – примеры передачи; г – восстановление сигнала станции С

Если *Передатчик0* имеет код (1, -1) и данные (1, 0, 1, 1), а *Передатчик1* имеет код (1, 1) и данные (0, 0, 1, 1), и оба отправителя передают одновременно, то эта таблица описывает шаги кодирования:

Шаг	Кодирование Передатчик0	Кодирование Передатчик1
0	код0 = (1, -1), данные0 = (1, 0, 1, 1)	код1 = (1, 1), данные1 = (0, 0, 1, 1)
1	закодир0 = $2(1, 0, 1, 1) - (1, 1, 1, 1) = (1, -1, 1, 1)$	закодир1 = $2(0, 0, 1, 1) - (1, 1, 1, 1) = (-1, -1, 1, 1)$
2	сигнал0 = закодир0 $\otimes$ код0 $= (1, -1, 1, 1) \otimes (1, -1)$ $= (1, -1, -1, 1, 1, -1, 1, -1)$	сигнал1 = закодир1 $\otimes$ код1 $= (-1, -1, 1, 1) \otimes (1, 1)$ $= (-1, -1, -1, -1, 1, 1, 1, 1)$

Поскольку *сигнал 0* и *сигнал 1* передаются одновременно в эфир, они складываются для получения необработанного сигнала

$$(1, -1, -1, 1, 1, -1, 1, -1) + (-1, -1, -1, -1, 1, 1, 1, 1) = (0, -2, -2, 0, 2, 0, 2, 0).$$

Этот необработанный сигнал называется интерференционной картиной. Затем приемник извлекает принимаемый сигнал для любого известного отправителя, складывая код передатчика с шаблоном помехи. Следующая таблица объясняет, как это работает, и показывает, что сигналы не мешают друг другу:

Шаг	Декодирование передатчик0	Декодирование передатчик1
0	код0 = (1, -1), сигнал = (0, -2, -2, 0, 2, 0, 2, 0)	код1 = (1, 1), сигнал = (0, -2, -2, 0, 2, 0, 2, 0)
1	декод0 = шаблон.код0	декод1 = шаблон.код1
2	декод0 = ((0, -2), (-2, 0), (2, 0), (2, 0)) · (1, -1)	декод1 = ((0, -2), (-2, 0), (2, 0), (2, 0)) · (1, 1)
3	декод0 = ((0 + 2), (-2 + 0), (2 + 0), (2 + 0))	декод1 = ((0 - 2), (-2 + 0), (2 + 0), (2 + 0))
4	данные0=(2, -2, 2, 2), значение (1, 0, 1, 1)	данные1=(-2, -2, 2, 2), значение (0, 0, 1, 1)

«Кроме того, после декодирования все значения, превышающие 0, интерпретируются как 1, тогда как все значения, меньшие нуля, интерпретируются как 0. Например, после декодирования данные0 равно (2, -2, 2, 2), но приёмник интерпретирует это как (1, 0, 1, 1). Значения ровно 0 означают, что передатчик не передал никаких данных, как в следующем примере: Предположим, что сигнал0 = (1, -1, -1, 1, 1, -1, 1, -1) передается один. Следующая таблица показывает декодирование в приемнике» [5]:

Шаг	Декодирование передатчик0	Декодирование передатчик1
0	код0 = (1, -1), сигнал = (1, -1, -1, 1, 1, -1, 1, -1)	код1 = (1, 1), сигнал = (1, -1, -1, 1, 1, -1, 1, -1)
1	декод0 = шаблон.код0	декод1 = шаблон.код1
2	декод0 = ((1, -1), (-1, 1), (1, -1), (1, -1)) · (1, -1)	декод1 = ((1, -1), (-1, 1), (1, -1), (1, -1)) · (1, 1)
3	декод0 = ((1 + 1), (-1 - 1), (1 + 1), (1 + 1))	декод1 = ((1 - 1), (-1 + 1), (1 - 1), (1 - 1))
4	данные0= (2, -2, 2, 2), значение (1, 0, 1, 1)	данные1= (0, 0, 0, 0), значение нет данных



«Когда передатчик пытается декодировать сигнал, используя код передатчика<sup>1</sup>, все данные равны нулю, поэтому взаимная корреляция равна нулю, и ясно, что передатчик<sup>1</sup> не передал никаких данных» [5].

### **Тема 8. Типы защиты от ошибок**

Перед началом обсуждения структурированной избыточности рассмотрим два основных метода использования избыточности для защиты от ошибок. В первом методе, обнаружении и повторной передаче ошибок, для проверки ошибок используется бит контроля четности (дополнительный бит, прикрепленный к данным). В этом случае приемный терминал не пытается исправить ошибку, он просто отправляет передатчику запрос на повторную передачу данных. Следует отметить, что для такого диалога между передатчиком и приемником необходима двусторонняя связь. Второй метод, прямое исправление ошибок (FEC), требует только односторонней линии связи, поскольку в этом случае бит проверки четности служит как для обнаружения, так и для исправления ошибок. Позже мы увидим, что не все комбинации ошибок могут быть исправлены, поэтому коды исправлений классифицируются в соответствии с их возможностями исправления ошибок.

Терминальные устройства систем связи часто классифицируются в соответствии с их связностью с другими оконечными устройствами. Возможные типы подключения показаны на рис. 17 и называются симплексными (не путать с симплексными или трансортгональными кодами), полудуплексными (полудуплексными) и полнодуплексными (полнодуплексными). Симплексное соединение на рис. 17 а является односторонней линией связи.

Сигналы передаются только от оконечного устройства А к оконечному устройству В. Полудуплексное соединение на рис. 17б является линией связи, по которой можно передавать сигналы в обоих направлениях, но не одновременно. И, наконец, полнодуплексное соединение (рис. 17, в) - это

двусторонняя связь, в которой сигналы передаются одновременно в обоих направлениях.

Передача только в одном направлении а)

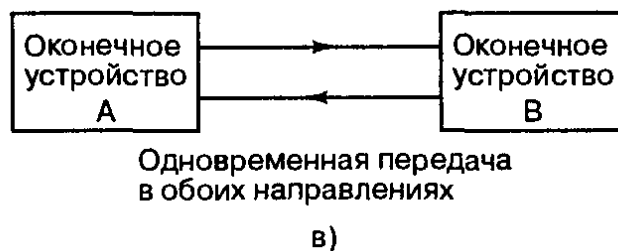
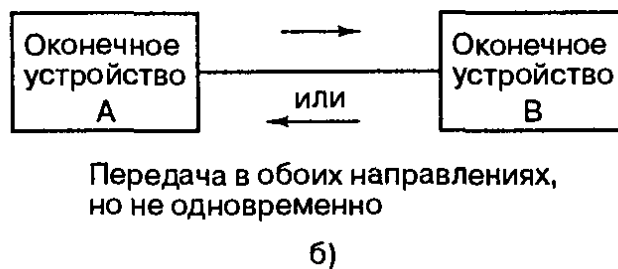
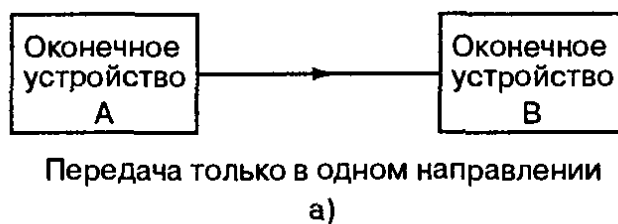


Рис. 17. Классификация типов соединения оконечных устройств:

а) симплексное; б) полудуплексное; в) полнодуплексное

### Автоматический запрос повторной передачи

«Автоматический запрос на повтор (ARQ) является методом контроля ошибок для передачи данных, который использует подтверждения (сообщения, отправленные получателем, указывающие, что он правильно принял пакет) и тайм-ауты (периоды времени, разрешенные для получения подтверждения) для обеспечения надежной передачи данных. Если отправитель не получает подтверждение до истечения времени ожидания, он обычно повторно передает пакет до тех пор, пока отправитель не получит подтверждение или не превысит заранее заданное количество повторных передач. На рис. 18 показаны три наиболее распространенных процедуры ARQ. Первая процедура ARQ, запрос ARQ с остановками (остановка и ожидание ARQ), показана на рис.18а. Для его

реализации требуется только полудуплексное соединение, поскольку перед началом следующей передачи передатчик ожидает подтверждения успешного приема (подтверждение – ACK) предыдущего. В примере, показанном на рисунке, третий блок переданных данных был получен с ошибкой. Следовательно, получатель передает отрицательное подтверждение (NAK); передатчик повторяет передачу третьего блока сообщений и только после этого передает следующий блок последовательно. Вторая процедура ARQ, непрерывный ARQ с возвратом, показана на рис. 18б. Для этого требуется дуплексное соединение. Оба терминала начинают передачу одновременно: передатчик отправляет информацию, а приемник передает подтверждение получения данных. Следует отметить, что каждому блоку передаваемых данных присваивается порядковый номер. Кроме того, номера кадров ACK и NAK должны быть согласованы; другими словами, задержка распространения сигнала должна быть известна априори, чтобы передатчик знал, какому блоку сообщения принадлежит этот кадр подтверждения. В примере на рис. 18б, время выбрано так, что между отправленным блоком сообщения и полученным подтверждением получения существует постоянный интервал из четырех блоков. Например, после отправки сообщения 8 он возвращает сигнал NAK, сообщаящий об ошибке в блоке 4. При использовании процедуры ARQ передатчик «возвращается» к сообщению об ошибке и снова передает всю информацию, начиная с поврежденного сообщения. И наконец, третья процедура, называемая непрерывным ARQ с выборочным повторением (непрерывный ARQ с селективным повторением), показана на рис. 18в. Здесь, как и во второй процедуре, требуется полнодуплексное соединение. Однако в этой процедуре только искаженное сообщение передается повторно; затем передатчик продолжает передачу с того места, где он был прерван, без необходимости повторно передавать правильно принятые сообщения» [6].

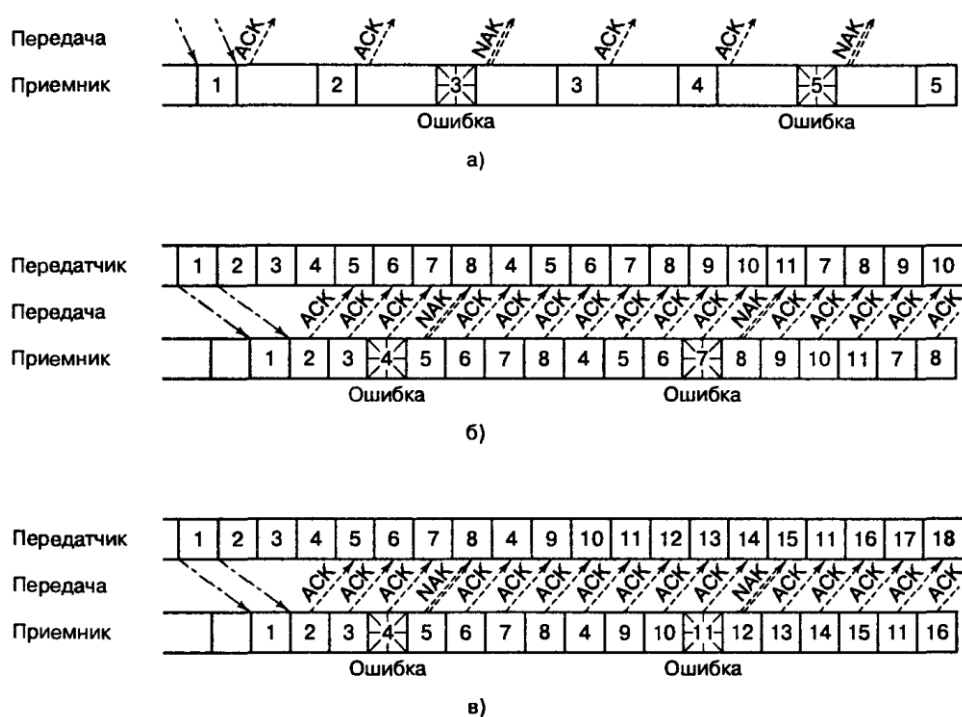


Рис. 18. Автоматический запрос повторной передачи (ARQ): а) запрос ARQ с остановками (полудуплексная связь); б) непрерывный запрос ARQ с возвратом (полнодуплексная связь); в) непрерывный запрос ARQ с выборочным повторением (полнодуплексная связь).

«Выбор конкретной процедуры ARQ является компромиссом между требованиями эффективного использования коммуникационных ресурсов и необходимостью полнодуплексной связи. Полудуплексная связь (рис. 18, а) является менее дорогостоящим, чем полный дуплекс; в то же время он менее эффективен, что можно определить по количеству пустых временных интервалов. Более эффективная работа показана на рис. 18б, требует более дорогой полнодуплексной связи» [6]. «Основное преимущество схем ARQ перед схемами прямого исправления ошибок (FEC) заключается в том, что «обнаружение ошибок требует более простого декодирующего оборудования и меньшей избыточности, чем исправление ошибок. Кроме того, он является более гибким; информация повторно передается только при обнаружении ошибки с другой стороны, метод FEC может быть более приемлемым (или дополняющим) по любой из следующих причин» [6].

1. Обратный канал недоступен или задержка при использовании ARQ слишком велика.
2. Алгоритм повторной передачи не может быть реализован удобным образом.
3. При ожидаемом количестве ошибок потребуется слишком много повторных передач.

### **Степень кодирования и избыточность**

«При использовании блочных кодов исходные данные делятся на блоки по  $k$  битов, которые иногда называют информационными битами или битами сообщения; каждый блок может представлять любое из  $2^k$  отдельных сообщений. В процессе кодирования каждый  $k$ -битный блок данных преобразуется в больший блок из  $n$  битов, который называется кодовым битом или символом канала. К каждому блоку данных кодер добавляет  $(n - k)$  битов, которые называются избыточными битами, битами четности или управляющими битами; они не несут новую информацию. Для обозначения описанного кода используется обозначение  $(n, k)$ . Отношение количества избыточных битов к количеству информационных битов  $(n-k) / k$  называется избыточностью кода; Отношение количества бит данных к общему количеству бит,  $k / n$ , называется скоростью кодирования. Под степенью кодирования понимается фрагмент кода, который учитывает полезную информацию. Например, в коде со степенью  $1/2$  каждый бит кода несет  $1/2$  бит информации» [6, 7].

### **Коды с контролем четности**

#### **Код с одним контрольным битом**

Бит четности или контрольный бит добавляется в строку двоичного кода, чтобы гарантировать, что общее число 1-битов в строке является четным или нечетным. Биты четности используются в качестве простейшей формы кода обнаружения ошибок.

«Существует два варианта битов четности: четный бит и нечетный бит четности. В случае четной четности для заданного набора бит подсчитываются

вхождения битов, значение которых равно 1. Если этот счет нечетный, значение бита четности устанавливается равным 1, что делает общее число вхождений 1 во всем наборе (включая бит четности) четным числом. Если счетчик 1 в данном наборе битов уже четный, значение бита четности равно 0. Операция суммирования использует арифметику по модулю 2 (исключающая операция ИЛИ). В случае нечетной четности кодирование меняется на противоположное. Для заданного набора битов, если число битов со значением 1 является четным, значение бита четности устанавливается равным 1, что делает общее число 1 во всем наборе (включая бит четности) нечетным числом. Если число битов со значением 1 является нечетным, счет уже нечетный, поэтому значение бита четности равно 0. На рис. 19 показана последовательная передача данных (первый - самый правый бит). Один бит четности добавляется к каждому блоку (самый левый бит в каждом блоке), давая положительную четность» [6, 7].

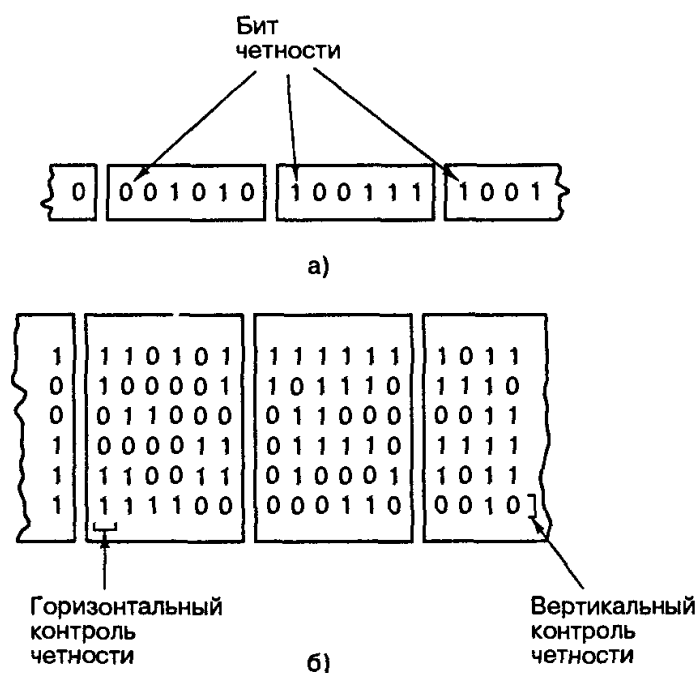


Рис. 19. Проверка четности для последователь и параллельной структуры кода: а) последовательная структура; б) параллельная структура.

На приемном терминале выполняется декодирование, чтобы проверить, равна ли ноль сумма принятых битов кодового слова по модулю 2 (положительная четность). Если результат равен 1, то кодовое слово, очевидно,

содержит ошибки. Степень кодирования такого кода может быть записана как  $k / (k + 1)$ . Считаете ли вы, что декодер может автоматически исправить полученную цифру с ошибкой? Нет это не возможно. Вы можете только обнаружить, что в символе кода присутствует нечетное количество ошибок. (Если была сделана ошибка в четном количестве битов, то проверка четности не покажет ошибок; этот случай является примером необнаруженной ошибки.)

### Линейные блочные коды

Линейные блочные коды— это класс кодов с контролем четности, которые можно описать парой чисел  $(n, k)$  (объяснение этой формы записи приводилось выше). В процессе кодирования блок из  $k$  символов сообщения (вектор сообщения) преобразуется в больший блок из  $n$  символов кодового слова (кодировый вектор), образованного с использованием элементов данного алфавита. Если алфавит состоит только из двух элементов (0 и 1), код является двоичным и включает двоичные разряды (биты). Если не будет оговорено противное, наше последующее обсуждение линейных блочных кодов будет подразумевать именно двоичные коды [8].

Одним из важнейших параметров кода является кодовое расстояние. *Кодовым расстоянием* или просто *расстоянием* кода  $V$  называется минимальное расстояние между двумя различными кодовыми словами, т.е.

$$d_V = \min d(x, y), x \neq y \text{ и } x, y \in V.$$

Для двоичного кода  $V$  под расстоянием понимается расстояние Хэмминга.

**Теорема 1.1.** «Код  $V$  исправляет все комбинации из  $t$  или менее ошибок, если и только если кодовое расстояние не меньше, чем  $2t + 1$ , т.е.  $d_V \geq 2t + 1$ » [8].

Доказательство теоремы основано на построении сфер с радиусом  $t$  вокруг каждого кодового слова. Для того чтобы каждая ошибка кратности  $t$  была исправлена, слово с такой ошибкой должно содержаться внутри сферы, описанной только вокруг одного кодового слова. Таким образом, расстояние между центрами двух сфер, т. е. различными кодовыми словами должно быть не меньше  $2t + 1$ . Геометрическая иллюстрация приведена на рис. 20 [8].

**Теорема 1.2.** «Код  $V$  обнаруживает все комбинации из  $t$  или менее ошибок, если и только если кодовое расстояние не меньше, чем  $t + 1$ , т.е.  $d_V \geq t + 1$ » [7].

Для того, чтобы каждая ошибка кратности  $t$  была обнаружена, слово с такой ошибкой не должно быть кодовым, т. е. расстояние между любыми различными кодовыми словами должно быть больше  $t$ .

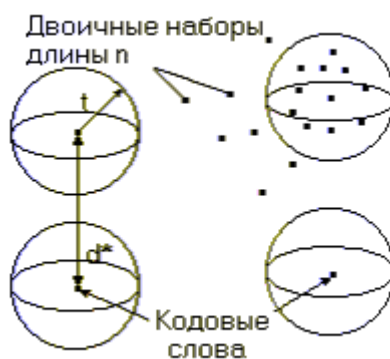


Рис 20. Сферы, центрами которых являются кодовые слова.

Например, код с расстоянием 3 исправляет все одиночные ошибки и обнаруживает все двукратные ошибки. Код с расстоянием 4 обнаруживает все ошибки кратности 3 и менее; однако такой код по-прежнему исправляет только одну ошибку.

Задача построения реального кода, т. е. кода, длина слов в котором достигает сотен символов, с заданным кодовым расстоянием не решается полным перебором даже с использованием современной вычислительной техники. Теория кодирования предлагает методы для построения таких кодов.

**Пример.** «Пусть необходимо построить код, у которого кодовые слова имеют длину равную 5, 3 (три) из которых информационных символа и исправляет одну ошибку. Так как информационных символов три, то различных кодовых слов должно быть  $2^3 = 8$ . Согласно теореме 1.1 расстояние между двумя различными кодовыми словами должно быть не меньше  $2 \times 1 + 1 = 3$ . Включим в код  $V$  слово 00000. Тогда все остальные кодовые слова имеют вес не менее 3. После включения в код  $V$  любого слова веса 3, ни одно слова веса 3



в код добавлено быть не может, так как расстояние между любыми двумя словами веса 3 равно 2. По той же причине слово 11111 веса 5 также не принадлежит коду  $V$ . Остаются 5 слов веса 4, которых не достаточно, чтобы иметь 8 кодовых слов» [8].

## Тема 9. Алгоритм Хэмминга

«В 1950 году Хэмминг ввел код [7,4] Хэмминга. Он кодирует четыре бита данных в семь битов, добавляя три бита четности. Он может обнаруживать и исправлять однобитовые ошибки. С добавлением общего бита четности он также может обнаруживать (но не исправлять) двухбитовые ошибки» [9].

Матрица генератора кода  $\mathbf{G}$  и матрица проверки на четность  $\mathbf{H}$  [9]

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{H} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Наконец, эти матрицы могут быть преобразованы в эквивалентные несистематические коды с помощью следующих операций:

- *Перестановка столбцов (замена столбцов)*
- *Элементарные операции со строками (замена строки линейной комбинацией строк)*

Как упомянуто выше, строки 1, 2 и 4 матрицы  $\mathbf{G}$  должны выглядеть знакомо, поскольку они отображают биты данных на их биты четности [8]:

- $p_1$  покрывает  $d_1, d_2, d_4$
- $p_2$  покрывает  $d_1, d_3, d_4$
- $p_3$  покрывает  $d_2, d_3, d_4$

«Остальные строки (3, 5, 6, 7) отображают данные в их позиции в закодированном виде, и только 1 в этой строке является идентичной копией. На самом деле эти четыре строки линейно независимы и образуют единичную матрицу (по замыслу, а не по совпадению)» [9].

«Также, как упоминалось выше, три строки  $H$  должны быть знакомы. Эти строки используются для вычисления вектора синдрома на принимающей стороне, и если вектор синдрома является нулевым вектором (все нули), то результирующее слово не содержит ошибок; если не ноль, то значение указывает, какой бит был перевернут» [9].

«Четыре бита данных, собранные в форме вектора  $p$ , предварительно умножают на  $G$  (то есть  $Gp$ ) и берут по модулю 2, чтобы получить кодированное значение, которое передается. Первоначальные 4 бита данных преобразуются в семь битов (отсюда и название Хэмминга (7.4)) с добавлением трех битов четности для обеспечения четности с использованием вышеуказанных покрытий битов данных. В первой таблице выше показано отображение между данными и битом четности в его конечную битовую позицию (от 1 до 7), но это также может быть представлено на диаграмме Венна. Первая диаграмма в этой статье показывает три круга (по одному на каждый бит четности) и включает в себя биты данных, которые покрывает каждый бит четности. Вторая диаграмма (показанная справа) идентична, но вместо этого отмечены позиции битов» [9].

В оставшейся части этого раздела в качестве рабочего примера будут использованы следующие 4 бита (показаны как вектор столбцов) [9]:

$$\mathbf{p} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

### Кодирование

«Предположим, мы хотим передать эти данные (1011) по шумному каналу связи. В частности, двоичный симметричный канал, означающий, что искажение ошибки не способствует ни нулю, ни единице (оно симметрично в возникновении ошибок). Кроме того, все исходные векторы предполагаются равновероятными. Мы берем произведение  $G$  и  $p$  с записями по модулю 2, чтобы определить переданное кодовое слово  $x$ » [9]:

$$\mathbf{x} = \mathbf{G}\mathbf{p} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Это означает, что 0110011 будет передаваться вместо передачи 1011.

«Программисты, обеспокоенные умножением, должны заметить, что каждая строка результата является младшим значащим битом счетчика чисел установленных битов, получаемых в результате того, что строка и столбец поразрядно объединяются, а не умножаются» [9].

### Проверка четности

Если при передаче не возникает ошибок, то полученное кодовое слово  $\mathbf{r}$  идентично переданному кодовому слову  $\mathbf{x}$  [9]:

$$\mathbf{r} = \mathbf{x}$$

Приемник умножает  $\mathbf{H}$  и  $\mathbf{r}$  для получения вектора синдрома  $\mathbf{z}$ , который указывает, произошла ли ошибка и, если да, то для какого бита кодового слова. Выполняя это умножение (опять же, записи по модулю 2) [9]:

$$\mathbf{z} = \mathbf{H}\mathbf{r} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Поскольку синдром  $\mathbf{z}$  является нулевым вектором, получатель может сделать вывод, что ошибки не произошло. Этот вывод основан на наблюдении, что когда вектор данных умножается на  $\mathbf{G}$ , изменение базиса происходит в векторном подпространстве, которое является ядром  $\mathbf{H}$ . Пока ничего не происходит во время передачи,  $\mathbf{r}$  останется в ядре  $\mathbf{H}$  и умножение даст нулевой вектор [8].

## Коррекция ошибок

В противном случае предположим, что произошла ошибка в один бит.

Математически мы можем написать

$$\mathbf{r} = \mathbf{x} + \mathbf{e}_i$$

по модулю 2, где  $\mathbf{e}_i$  это  $i$ -й единичный вектор, т.е., нулевой вектор с 1 в  $i$ -й месте, считая от 1.

$$\mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Таким образом, вышеприведенное выражение означает ошибку с одним битом в  $i$ -м месте.

Теперь, если мы умножим этот вектор на  $\mathbf{H}$ :

$$\mathbf{H}\mathbf{r} = \mathbf{H}(\mathbf{x} + \mathbf{e}_i) = \mathbf{H}\mathbf{x} + \mathbf{H}\mathbf{e}_i$$

Поскольку  $\mathbf{x}$  переданные данные, они без ошибок, и, как результат, произведение  $\mathbf{H}$  и  $\mathbf{x}$  ноль. Таким образом

$$\mathbf{H}\mathbf{x} + \mathbf{H}\mathbf{e}_i = \mathbf{0} + \mathbf{H}\mathbf{e}_i = \mathbf{H}\mathbf{e}_i$$

Теперь произведение  $\mathbf{H}$  на  $i$ -й стандартный базисный вектор выделяет этот столбец  $\mathbf{H}$ , мы знаем, что ошибка происходит в том месте, где встречается этот столбец  $\mathbf{H}$ .

Например, предположим, что мы ввели битную ошибку в бит № 5

$$\mathbf{r} = \mathbf{x} + \mathbf{e}_5 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

И тогда,

$$\mathbf{z} = \mathbf{H}\mathbf{r} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

который соответствует пятому столбцу  $\mathbf{H}$ . Кроме того, используемый общий алгоритм был преднамеренным в своей конструкции, так что синдром 101 соответствует двоичному значению 5, которое указывает, что пятый бит был поврежден. Таким образом, ошибка была обнаружена в бите 5 и может быть исправлена (просто перевернуть или отменить ее значение) [9]:

$$\mathbf{r}_{\text{corrected}} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \bar{1} \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Это исправленное полученное значение действительно теперь соответствует переданному значению  $\mathbf{x}$  сверху [9].

### Дешифрование

После того, как принятый вектор был определен как свободный от ошибок или исправленный, если произошла ошибка (при условии, что возможны только нулевые или однокбитовые ошибки), полученные данные должны быть декодированы обратно в исходные четыре бита [9].

Сначала определим матрицу  $\mathbf{R}$ :

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

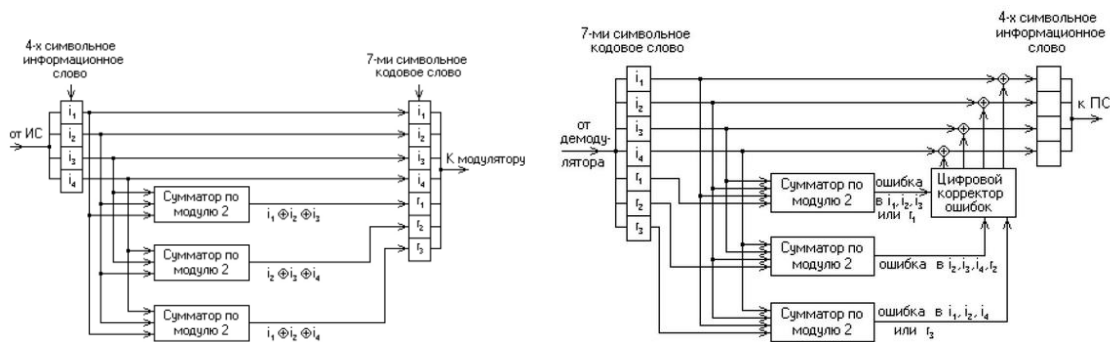
Тогда полученное значение  $\mathbf{p}_r$  равно  $\mathbf{R}\mathbf{r}$ . Используя запущенный пример сверху [8]

$$P_r = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

### Несколько ошибок

Однако коды Хэмминга (7,4) и аналогичные коды Хэмминга не могут различить однобитовые ошибки и двухбитовые ошибки. То есть двухбитовые ошибки выглядят так же, как и однобитовые ошибки. Если коррекция ошибок выполняется для двухбитовой ошибки, результат будет неверным.

Точно так же коды Хэмминга не могут обнаружить или восстановить произвольную трехбитовую ошибку; Рассмотрим схему: если бит в зеленом круге (красного цвета) равен 1, проверка на четность вернет нулевой вектор, указывая, что в кодовом слове нет ошибки.



## Тема 10. Цифровые виды модуляции

Хорошо известно, что такое модуляция – это процесс изменения амплитуда, частота и фаза гармонических колебаний в зависимости от сообщения. В цифровых видах модуляции переменными параметрами могут быть такие же, как и в аналоговых. Основное отличие цифровых от аналоговых методов модуляции заключается в дискретности, с которой происходят изменения этих параметров в соответствии с передаваемым дискретным информационным сообщением. Из-за дискретного характера изменения параметров сигнала, цифровую модуляцию часто называют манипуляцией.

«Рассмотрим первый, и он же является самым простым видом модуляции (манипуляции) - амплитудная манипуляция (АМн), на рис. 21 показан пример четырехпозиционной амплитудной манипуляции. Исторический факт: в своем первом в мире приемнике А.С. Попов использовал именно амплитудное изменение сигнала, т.е. АМн» [2].

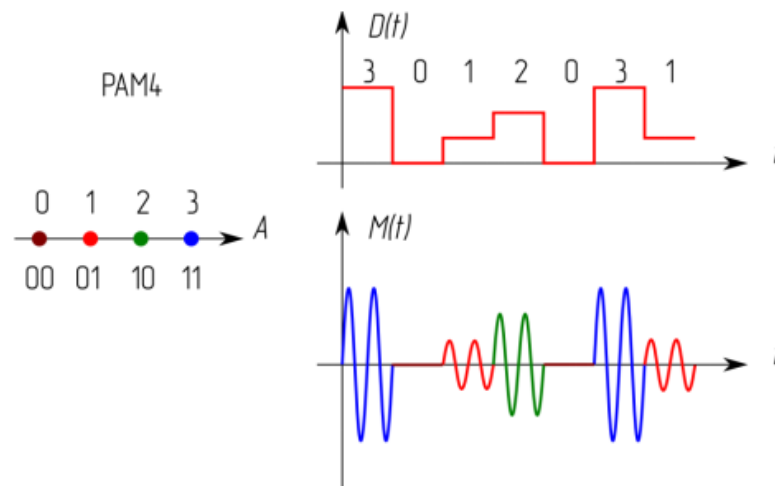


Рис. 21. 4х-позиционная амплитудная модуляция

При АМн модулированный сигнал имеет вид:

$$M(t) = D(t) \cdot A_0 \sin 2\pi f_0 t + \varphi_0 ,$$

где  $M(t)$  – модулированный сигнал;

$D(t)$  – данные для передачи;

$A_0$  – амплитуда несущего колебания;

$f_0$  – частота несущего колебания;

$\varphi_0$  – начальная фаза несущего колебания.

«Следующим видом или типом цифровой манипуляции является *частотная* манипуляция, кратко обозначается как ЧМн. В этом случае частота гармонического колебания претерпевает дискретные изменения в зависимости от сообщения» (рис. 22).

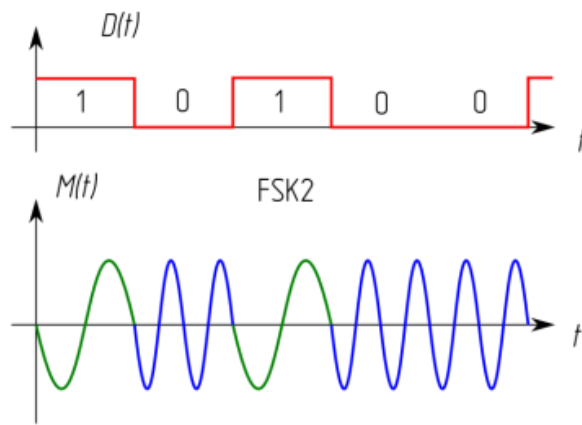


Рис. 21. 2-ичная частотная модуляция

В этом случае ЧМн модулированный сигнал имеет следующий вид:

$$M(t) = A_0 \sin(2\pi f_0 t + 2\pi f_d D(t) + \varphi_0),$$

где  $M(t)$  – модулированный сигнал;

$A_0$  – амплитуда несущего колебания;

$D(t)$  – данные для передачи;

$f_0$  – частота несущего колебания;

$\varphi_0$  – начальная фаза несущего колебания;

$f_d$  – девиация частоты.

«Рассмотрим последний вид манипуляции, который является родственным предыдущему, в котором изменяемым параметром является фаза гармонического колебания, т.е. *фазовая манипуляция* (ФМн) (рис. 22). При 2-ичной фазовой манипуляции модулированный сигнал имеет вид» [2]:

$$M(t) = \begin{cases} A_0 \sin(2\pi f_0 t + \varphi_0), & \text{if } D(t) = 0; \\ A_0 \sin(2\pi f_0 t + \varphi_0 + \pi), & \text{if } D(t) = 1, \end{cases}$$

где  $M(t)$  – модулированный сигнал;

$A_0$  – амплитуда несущего колебания;

$D(t)$  – данные для передачи;

$f_0$  – частота несущего колебания;

$\varphi_0$  – начальная фаза несущего колебания.



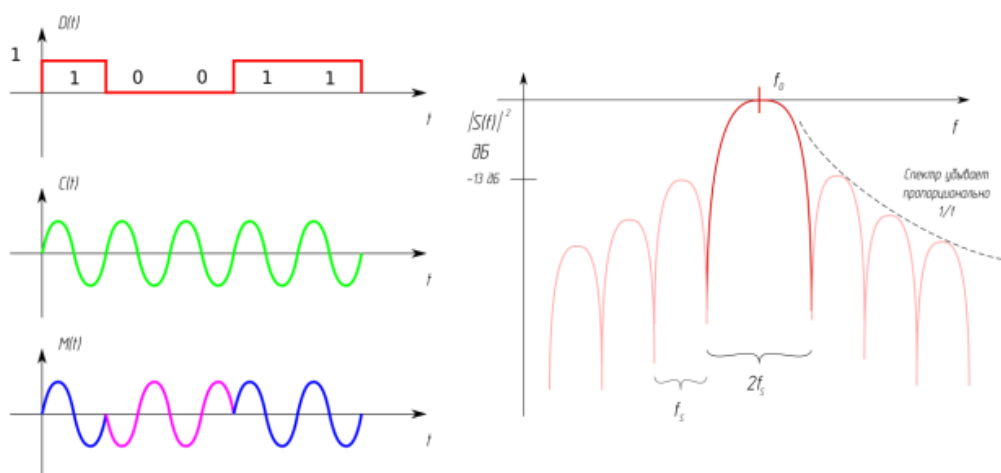


Рис. 21. Двоичная фазовая модуляция и её спектр

Фазовая и частотная манипуляции различных порядков обладает тем преимуществом, что амплитуда синусоидального колебания остается всегда неизменной. Следовательно, выходные каскады усилителей могут быть изготовлены с немного ослабленными характеристиками по линейности и динамическому диапазону.

«Наиболее популярным видом цифровой модуляции, особенно в сотовых сетях последних поколений – различные виды *квадратурной амплитудной манипуляции* (КАМ). Связано это с тем, что КАМ-манипуляция является наиболее эффективным видом модуляции. В современных системах подвижной связи, начиная с поколения EDGE, произошел переход к фазовым видам манипуляции и квадратурным видам амплитудной манипуляции высоких порядков. На рис. 22 показаны сигнальные созвездия фазовых видов манипуляции двоичная фазовая – BPSK, четырехпозиционная фазовая – QPSK, восьмипозиционная фазовая – 8PSK и квадратурной амплитудной манипуляции КАМ16 или QAM16 на комплексной плоскости» [2].

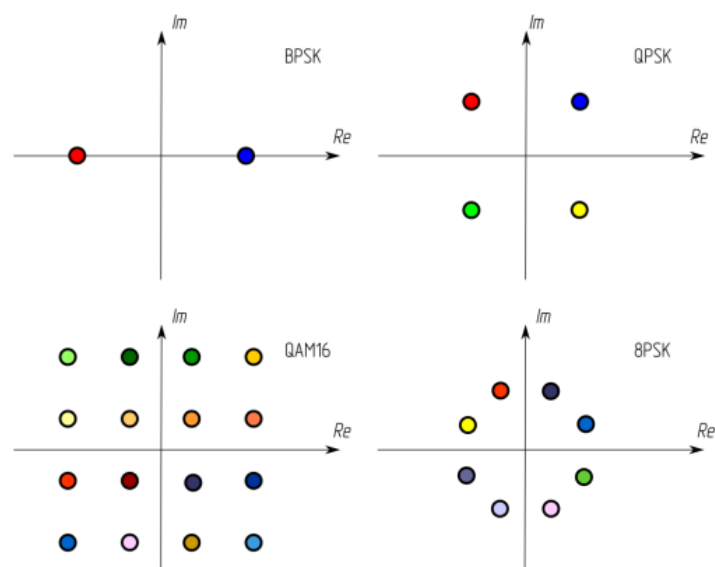


Рис. 22. Сигнальные созвездия фазовых видов модуляции BPSK, QPSK, 8PSK и квадратурной амплитудной манипуляции QAM16

КАМ модуляция реализуется на практике с использованием таблицы соответствия или таблицы истинности, в которой каждый входной символ ставится в соответствие одной точке на комплексной плоскости. Модулятор в таких системах называется формирователем комплексной огибающей сигнала. Выходной сигнал формирователя комплексной огибающей фильтруется фильтром НЧ и подается на квадратурный модулятор, который передает спектр сигнала на высокую несущую или промежуточную частоту.

«Для сравнения различных видов модуляции, как правило, используются два критерия: критерий спектральной эффективности модуляции и критерий энергетической эффективности модуляции. Эффективность спектральной модуляции характеризует полосу частот, необходимую для передачи полезной информации с заданной скоростью. Эффективность модуляции энергии определяет мощность, необходимую для передачи информации с заданной надежностью (с заданной вероятностью ошибки в битах или символах)» [2].

«На рис. 23 показаны энергетические характеристики некоторых видов модуляции. Как видно, двоичная фазовая манипуляция BPSK и квадратурная фазовая манипуляция QPSK имеют одинаковую зависимость вероятности

битовой ошибки от отношения  $E_b / N_0$ . Это связано с тем, что для передачи данных с одинаковой скоростью передачи при квадратурной фазовой манипуляции требуется полуширина частот. На рисунке четко видно, что фазовая манипуляция с числом точек в созвездии равным 16 имеет примерно равные характеристики с квадратурной амплитудной манипуляцией с числом точек в созвездии равным 64, т. е. при тех же условиях переход от ПСК16 к QAM64 увеличит скорость передачи данных в полтора раза» [2].

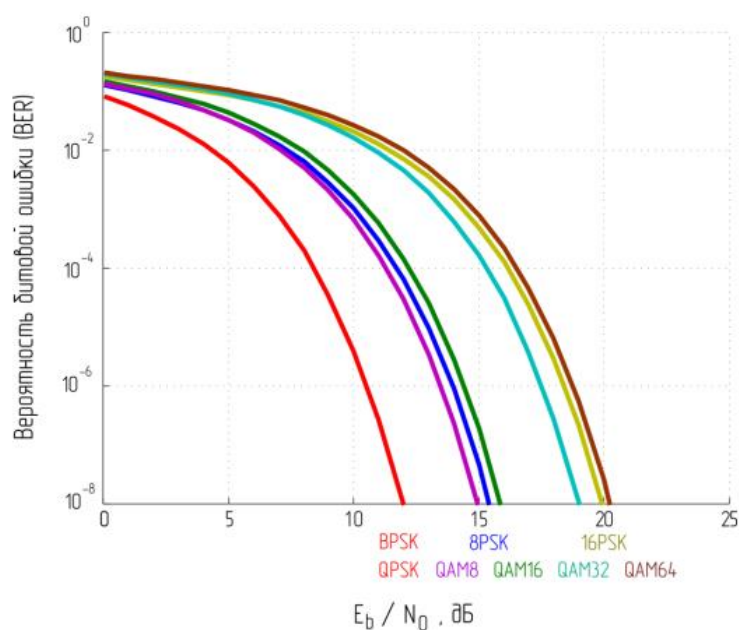


Рис. 23. Энергетическая эффективность различных видов цифровой модуляции

При создании эффективных созвездий КАМ манипуляции используются следующие два правила:

- 1) необходимо, чтобы СКО значение модулей всех векторов созвездия было нормировано на единицу;
- 2) точки созвездия расположены таким образом, чтобы максимизировать наименьшее евклидово расстояние между двумя точками.

Прямоугольные созвездия, как показано выше, квадратурных амплитудных манипуляций не являются оптимальными с точки зрения вышеуказанных критериев, но наиболее часто используются за счёт простоты реализации в сигнальных цифровых схемах.

Спектральная эффективность различных типов систем модуляции и связи определяется отношением скорости передачи данных к занятой полосе частот  $R/W$  [бит / с / Гц] (рис. 24).

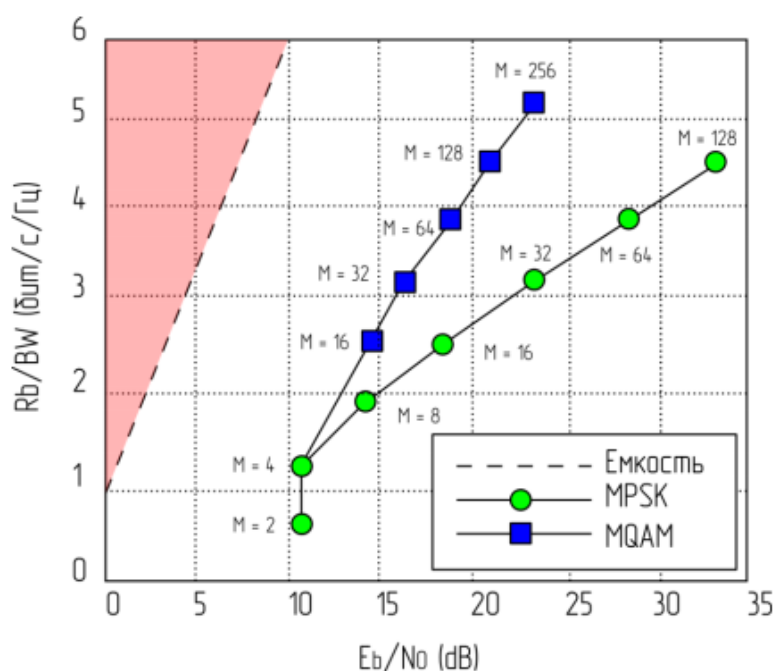


Рис. 24. Спектральная эффективность различных видов цифровой модуляции

Как видно из рис. 24, ФМн, т.е. фазовая манипуляция теряет в спектральной эффективности квадратурной амплитудной манипуляции. В условиях, когда постоянно растут требования к цифровым системам связи для эффективного использования спектра возникает необходимость использования видов манипулирования возрастающими заказами. Так, например, сегодня известны радиорелейные линии связи с модуляцией КАМ 1024-го порядка. Следует отметить, что использование таких высоких порядков манипуляции ограничено в первую очередь нелинейностью аналоговых трактов современных систем связи, искажениями, вносимыми средой распространения сигнала, и параметрами приемных систем синхронизации.

## Тема 11. Формирователь комплексной огибающей сигнала

В современных цифровых системах связи модулятор реализуется с помощью таблиц соответствия, в которых для каждого символа данных имеется однозначная одна точка на плоскости комплексных чисел (рис. 25).

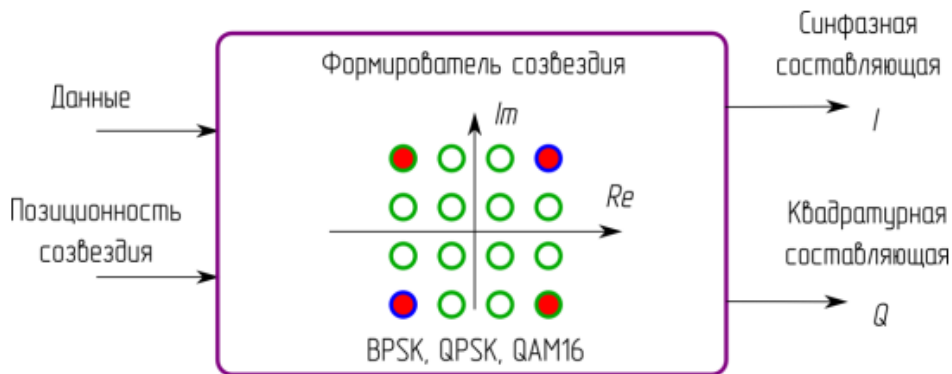


Рис. 25. Формирователь комплексной огибающей сигнала

В таблице 2 приведены примеры созвездий для двухпозиционных BPSK, четырехпозиционных QPSK манипуляций, а также 16-поз. КАМ.

Таблица 1.2

Таблицы соответствий для различных видов манипуляции

Название манипуляции	Позиционность созвездия	Данные для передачи				Выход формирователя комплексной огибающей			
BPSK	2	0	1			-1-1j	+1+1j		
QPSK	4	0	1			-1+1j	+1+1j		
		2	3			+1-1j	-1-1j		
QAM16	16	0	1	2	3	-3+3j	-1+3j	+1+3j	+3+3j
		4	5	6	7	-3+1j	-1+1j	+1+1j	+3+1j
		8	9	10	11	-3-1j	-1-1j	+1-1j	+3-1j
		12	13	14	15	-3-3j	-1-3j	+1-3j	+3-3j

Простота реализации цифрового модулятора, т.е. формирователя огибающей, позволяет в режиме реального времени создавать цифровые беспроводные системы связи с адаптивно изменяемыми в процессе работы созвездиями. Данное преимущество дает возможность подстраиваться под изменение условий распространения радиосигналов в среде и использовать спектральный и энергетический ресурс наиболее эффективно.

## Формирующий фильтр

«Известно, что инструментом устранения межсимвольной интерференции является идеальный фильтр Найквиста, который в свою очередь нереализуем в практических схемах. В системах цифровой беспроводной связи принято использовать фильтры с конечной импульсной характеристикой (FIR) для формирования импульсов подходящей формы. В данных КИХ-фильтрах импульсная характеристика фильтра Найквиста усекается оконной (весовой) функцией, что в этом случае приводит к увеличению уровня боковых лепестков, низкой скорости затухания частотной характеристики и пульсаций в зоне пропускания фильтра. Для борьбы с этими негативными последствиями Найквистом предложено сглаживать края частотной характеристики (ЧХ) фильтра с помощью аппроксимации увеличенной косинусной функцией» (рис. 26).

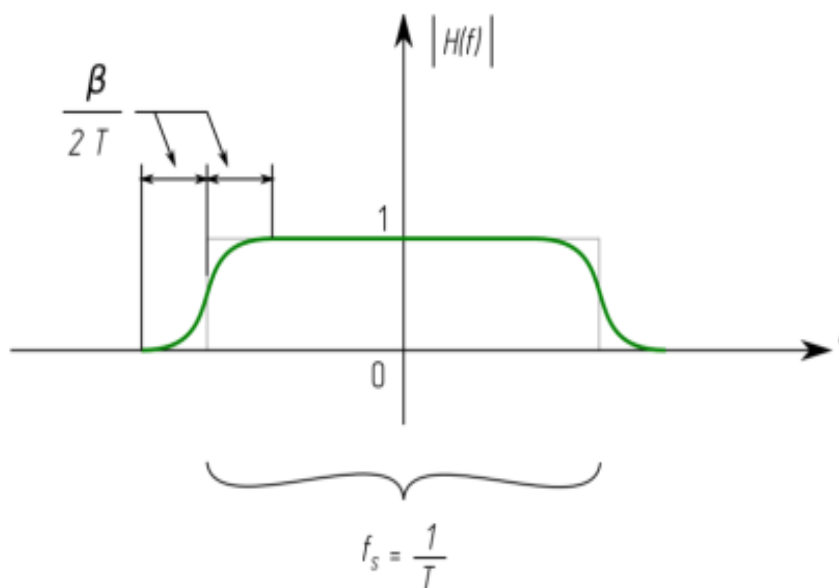


Рис. 26. АЧХ фильтра Найквиста при аппроксимации фронтов приподнятым косинусом

АЧХ фильтра описывается следующим мат. выражением:

$$|H(f)| = \begin{cases} 1, & \text{if } |f| < \frac{1-\beta}{2T}; \\ \frac{1}{2} \left( 1 + \cos \left( \frac{\pi T}{\beta} \left[ |f| - \frac{1-\beta}{2T} \right] \right) \right), & \text{if } \frac{1-\beta}{2T} < |f| < \frac{1+\beta}{2T}; \\ 0, & \text{if } |f| > \frac{1+\beta}{2T}. \end{cases}$$

где  $T$  – период следования символов;

$\beta$  – коэффициент сглаживания фронтов (в диапазоне от 0 до 1).

При коэффициенте  $\beta = 0$ , АЧХ фильтра приобретает форму прямоугольника, а при  $\beta = 1$ , АЧХ становится похожим на приподнятый косинус. При реализации приемного устройства системы связи на практике используют согласованную фильтрацию сигнала (рис. 27).

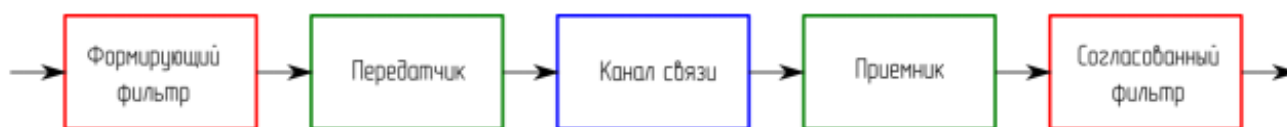


Рис. 27. Формирующий фильтр и согласования фильтрация  
в системе цифровой связи

Сигнал с выхода формирующего фильтра и после канала связи поступает на вход, т.н. согласованного фильтра. Здесь для эффективного и качественного подавления шума частотная характеристика последнего связана с формирующим фильтром таким образом, общая частотная характеристика системы связи передатчик-приемник определяется как произведение ЧХ формирующего фильтра и ЧХ согласованного фильтра. Именно эта общая частотная характеристика должна соответствовать требованиям Найквиста. Следовательно, при проектировании и реализации цифровой связи на передающей и приемной сторонах должны быть фильтры с ЧХ в форме квадратного корня от характеристики типа приподнятого косинуса. В таком

случае общая ЧХ повторяет частотную характеристику фильтра Найквиста, что позволяет исключить межсимвольные искажения при приеме информации.

### Квадратурный модулятор

Особенностью модулятора является то, что спектр сигнала с низкой переносится на более высокую промежуточной частоту путем умножения на  $\cos$  и  $-\sin$ . На практике такое умножение реализуется с помощью квадратурного гетеродина и двух умножителей (рис. 28). Перенос спектра полезного сигнала предпочтительно реализовывать в цифровом виде, но можно и в аналоговом.

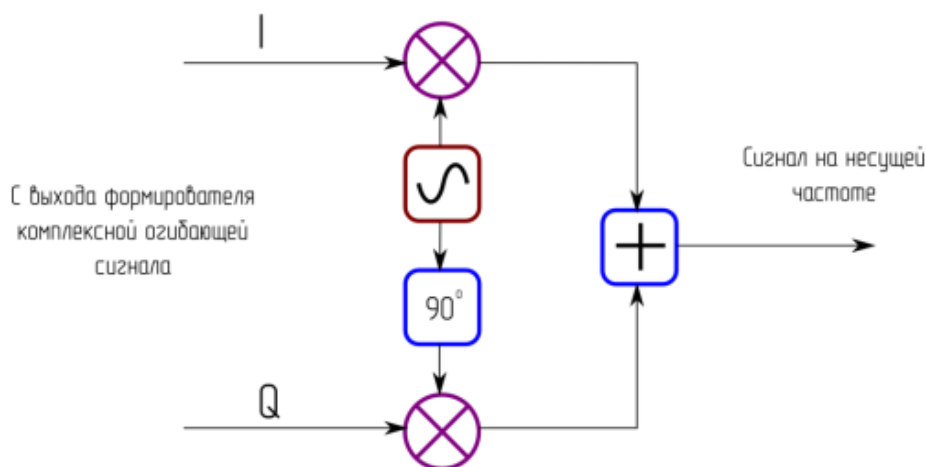


Рис. 28. Квадратурный модулятор

Квадратурная аналоговая обработка сигналов позволяет использовать недорогие низкочастотные цифро-аналоговые преобразователи для переноса спектра сигнала с низкой частоты на высокую несущую частоту, что удешевляет производство. Благодаря этому появляется возможным создание недорогих, небольших по размеру и многоцелевых систем цифровой радиосвязи, т.к. генерация сигнала происходит в цифровом виде и может быть изменена путем небольших изменений в программном коде.

В то же самое время аналоговые квадратурные модуляторы имеют ряд недостатков по той причине, что невозможно достичь полной идентичности I и Q каналов. Требования к аналоговым трактам возрастают вместе с увеличением местоположения используемых манипуляций.



## СПИСОК ЛИТЕРАТУРЫ

1. Пешков, И.В. Адаптивные алгоритмы пространственной обработки сигналов, эффективные при случайных дестабилизирующих воздействиях: дис. ... канд. физ.-мат. наук : 01.04.03 / И.В. Пешков. – Воронеж, 2012. – 182 с.
2. Боев, Н.М. Системы связи. Подвижные системы связи: учебно-методическое пособие [Электронный ресурс]; сост. Н.М. Боев. – Красноярск: Сиб. федер. ун-т, 2012. – Режим доступа: URL: <http://diss.seluk.ru/m-radiotekhnika/1087216-1-sistemi-svyazi-podvizhnie-sistemi-svyazi-lekcii-uchebno-metodicheskoe-posobie-elektronnoe-izdanie-krasnoyarsk-sfu-2013-udk-621396930.php>
3. Сжатие данных. URL: [https://en.wikipedia.org/wiki/Data\\_compression](https://en.wikipedia.org/wiki/Data_compression).
4. Таненбаум, Эндрю (1944-). Компьютерные сети / Э. Таненбаум, Д. Уэзеролл; [пер. с англ. А. Гребеньков]. – 5-е изд. – М. [и др.] : Питер, 2014. – 955 с.
5. Кодовое разделение пользователей [Электронный ресурс]. – Режим доступа: URL: [https://en.wikipedia.org/wiki/Code\\_division\\_multiple\\_access](https://en.wikipedia.org/wiki/Code_division_multiple_access).
6. Ташатов, Н.Н. Использование избыточности для защиты от ошибок / Н.Н. Ташатов // Известия НАН РК. Серия физ.-мат. наук. – 2007. – № 5. – С. 63-68.
7. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение. – Изд. 2-е, испр.; пер. с англ. / Б. Скляр. – М.: Издательский дом «Вильямс», 2003. – 1104 с.
8. Коды, исправляющие ошибки [Электронный ресурс]. – Режим доступа: URL: [https://ido.tsu.ru/iop\\_res1/kodi/index.php-mod=article&id=169.htm](https://ido.tsu.ru/iop_res1/kodi/index.php-mod=article&id=169.htm).
9. Коды Хэмминга [Электронный ресурс]. – Режим доступа: URL: [https://en.wikipedia.org/wiki/Hamming\(7,4\)](https://en.wikipedia.org/wiki/Hamming(7,4)).

## СОДЕРЖАНИЕ

Введение .....	3
<b>Раздел 1. Общие вопросы передачи .....</b>	<b>5</b>
Тема 1. Структура систем цифровой связи .....	5
Тема 2. Пропускная способность систем связи .....	7
Тема 3. Передача данных в канале с ограниченной полосой .....	9
Тема 4. Передающая часть систем цифровой связи .....	13
<b>Раздел 2. Структура передающей части .....</b>	<b>16</b>
Тема 5. Аналого-цифровое преобразование .....	16
5.1. Канальное кодирование .....	18
Тема 6. Сжатие информации .....	20
6.1. Кодирование Хаффмана .....	25
Тема 7. Канальное кодирование. Множественный доступ .....	28
Тема 8. Типы защиты от ошибок .....	33
Тема 9. Алгоритм Хэмминга .....	41
Тема 10. Цифровые виды модуляции .....	46
Тема 11. Формирователь комплексной огибающей сигнала .....	53
Список литературы .....	57

Учебное издание

**Илья Владимирович Пешков**

# **РАДИОПЕРЕДАЮЩИЕ УСТРОЙСТВА**

## **(НАПРАВЛЕНИЕ «РАДИОТЕХНИКА»)**

**Учебное пособие**

*Технический редактор – О. А. Ядыкина*

*Техническое исполнение – В. М. Гришин*

Формат 60 x 84 /16. Гарнитура Times. Печать трафаретная.

Печ.л. 3,5 Уч.-изд.л. 3,4

Тираж 300 экз. Заказ 93

Отпечатано с готового оригинал-макета на участке оперативной полиграфии  
Елецкого государственного университета им. И. А. Бунина

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Елецкий государственный университет им. И. А. Бунина»  
399770, г. Елец, ул. Коммунаров, 28,1