

Н. И. Сорока, Г. А. Кривинченко

ТЕЛЕМЕХАНИКА

Конспект лекций

для студентов специальности

1-53 01 07 «Информационные технологии и управление
в технических системах»

Часть 5

Технологии передачи дискретной информации



Минск

ВВЕДЕНИЕ

В связи с бурным развитием микроэлектронной и вычислительной техники цифровые системы передачи информации находят широкое применение в решении самых различных задач связи (рис. В.1), телеметрии и телеуправления [15]. Особо важное значение такие системы приобрели в таких областях, как передача данных, космическая связь и управление, структурные схемы которых приведены в [15].

Специфика различных областей применения систем передачи цифровой информации существенно различного подхода к реализации таких систем. Система передачи данных по телефонным каналам совершенно не похожа на систему космической связи или на цифровую систему тропосферной связи ни по техническому исполнению, ни по параметрам. Однако в принципах построения и назначения отдельных устройств самых разных систем много общего. Это дает возможность рассматривать цифровые системы передачи информации с общих позиций, отвлекаясь от специфики их применения и конкретных условий. Такой подход позволяет выяснить ряд важных положений, характерных для любой системы передачи цифровой информации.

Любая современная система передачи дискретной информации состоит из источника информации, осуществляющего первичное кодирование и модуляцию, приемника информации, выполняющего соответственно обратные операции (демодуляцию и декодирование), и канала связи, по которому передаются сигналы от источника к приемнику информации.

Обобщенная структурная система передачи дискретной информации (СПДИ) однонаправленного действия (рис. В.2) состоит из оконечного оборудования, выполняющего функции отправителя (источник информации) и получателя (приемник информации) сообщений, устройства защиты от ошибок (УЗО), устройства преобразования сигналов (УПС) и канала связи (КС). В УЗО происходит кодирование (декодирование) информации. УПС осуществляет преобразование сигналов данных в форму, удобную для передачи по каналу связи, а устройство сопряжения (УС) обеспечивает обмен информационными и управляющими сигналами между СПДИ и устройством оконечной обработке данных (ООД). Координация взаимодействия составных частей СПДИ обеспечивается специальными импульсами, вырабатываемыми устройством управления (УУ). Совокупность непрерывного канала с включенными на его входе и выходе УПС называют дискретным каналом передачи данных, а объединение его с УЗО – защищенным от ошибок каналом (ЗКПД). Для обмена информацией между абонентами используются две однонаправленные СПДИ, которые конструктивно могут быть выполнены в виде одной дуплексной СПДИ.

В каждом канале передачи информации действуют внешние и внутренние помехи, искажающие кодовые комбинации (сообщение) (рис. В.3). Как видно из рисунка основные искажения сигнала вызываются затуханием, ограниченностью полосы пропускания, задержкой и шумами.

Так как мощность выходного сигнала среды передачи без промежуточных усилителей всегда меньше, чем мощность выходного сигнала, затухание среды передачи всегда является отрицательной величиной.

Под полосой пропускания понимается непрерывный диапазон частот, для которого отношение амплитуды выходного сигнала ко входному превышает некоторый заранее заданный предел, обычно 0,5; т.е. полоса пропускания определяет диапазон частот синусоидального сигнала, при которых этот сигнал передается по каналу связи без значительных искажений.

Так как любой дискретный сигнал состоит из компонент различной частоты, то на вход приемного устройства поступают только те компоненты, частота которых находится внутри полосы пропускания. Ограниченность полосы пропускания приводит к частотным искажениям сигнала. Известно, что амплитуда каждой из частотных гармоник снижается с ростом частоты. Поэтому, чем шире полоса пропускания среды передачи, тем большее число высокочастотных компонент проходит по линии связи, а следовательно, тем надежнее будет полученный сигнал воспроизводить переданный сигнал.

Искажение из-за задержки определяется тем, что скорость распространения синусоидального сигнала по линии изменяется с изменением частоты. Следовательно, при передаче цифрового сигнала различные компоненты, из которых образован сигнал, достигают приемника с различными задержками. Результатом этого является искажение сигнала, называемое искажением, вызванным задержкой. Степень искажения растет с увеличением скорости передачи битов, что вызвано следующей причиной: по мере роста скорости битов некоторые частотные компоненты, связанные с передачей данного бита, задерживаются и начинают влиять на частотные компоненты следующего бита. Поэтому искажения из-за задержки называют также межсимвольными взаимными помехами. В результате действия этого искажения в моменты измерения поступивший сигнал изменяется. Так как обычно поступивший сигнал измеряется в номинальном центре каждого битового интервала, то, следовательно, при увеличении скорости битов искажение из-за задержки может привести к некорректной интерпретации полученного сигнала.

Шумы постоянно присутствуют в реальном канале. В отсутствие передаваемого сигнала в идеальной линии связи должен быть нулевой уровень электрического сигнала. Однако на практике в линии имеют место случайные всплески даже тогда, когда никакой сигнал не передается. Эти всплески называют уровнем шумов в линии, и в пределе по мере затухания передаваемого сигнала его уровень становится сравнимым с уровнем шума.

Таким образом, основными функциями СПДИ являются: модуляция, линейное кодирование, сжатие информации, повышение достоверности передачи информации, обеспечения синхронности и синфазности передатчика и приемника. Этим и другим вопросам посвящен материал данного конспекта.

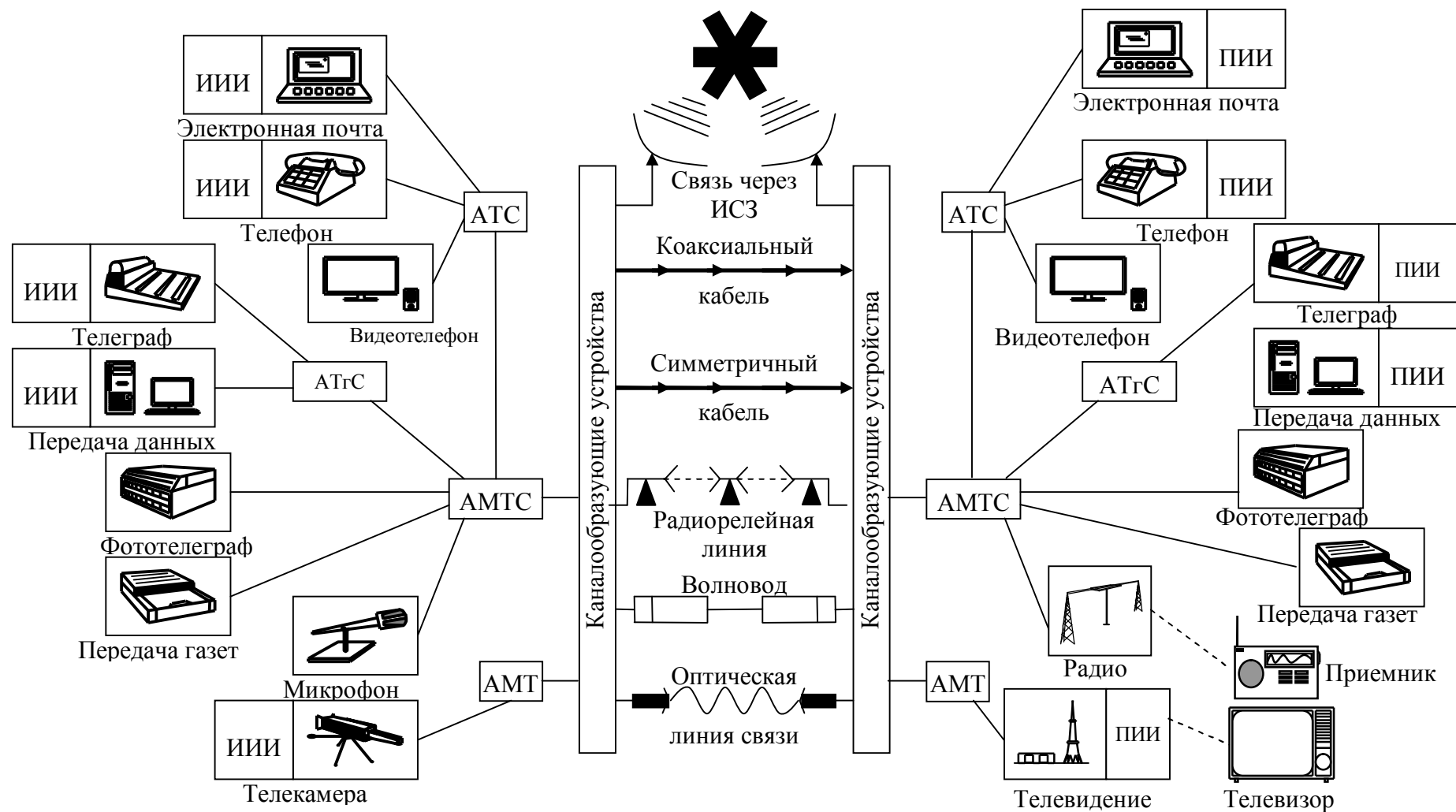


Рис. В.1. Основные виды электросвязи:

АТС – автоматическая телефонная станция; АТрС – автоматическая телеграфная станция;
 АМТС – аппаратная междугородней телефонной сети; АМТ - аппаратная междугороднего телевидения;
 ИИИ – источник измерительной информации; ПИИ – приемник измерительной информации.

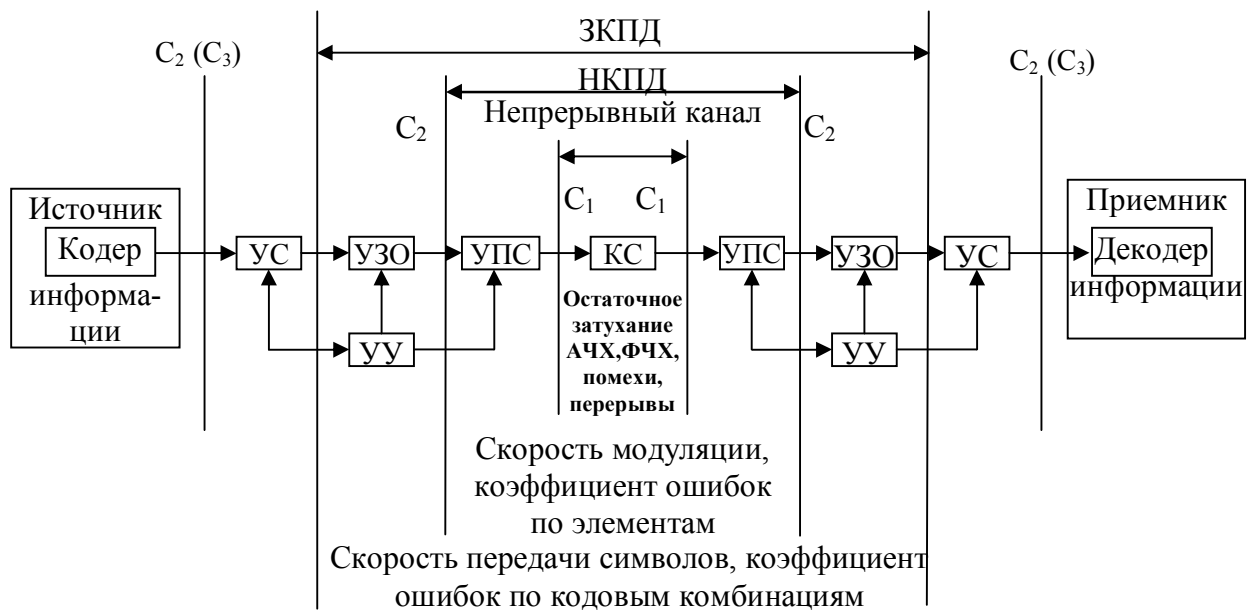


Рис. В.2. Структурная схема системы передачи данных (СПД); C_1, C_2, C_3 – стыки

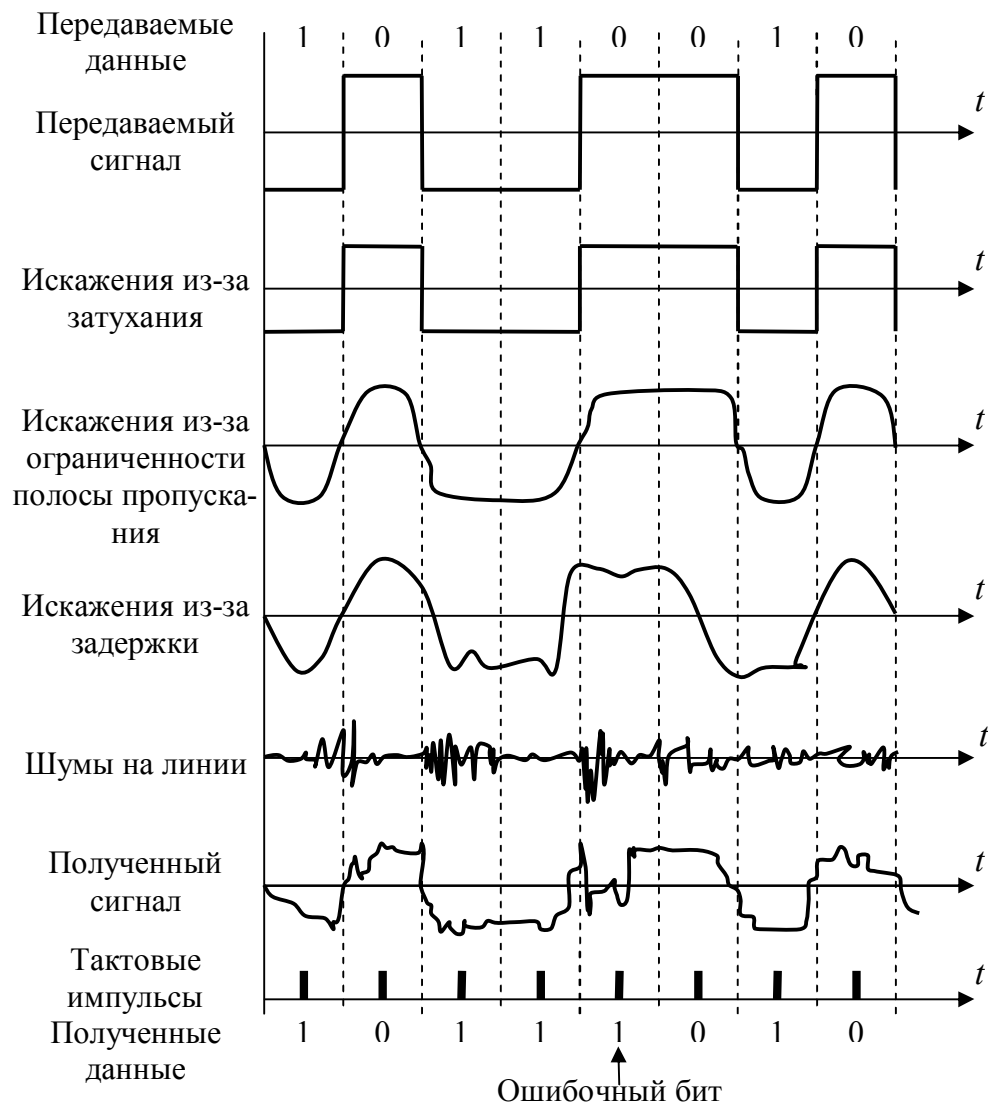


Рис. В.3. Виды искажения сигнала

1. ЦИФРОВЫЕ МЕТОДЫ МОДУЛЯЦИИ

1.1. Общие сведения

Центральной проблемой при построении любой системы связи является выбор и техническая реализация способов введения передаваемой информации в физический переносчик в точке передачи и выделения этой информации в точке приема. Эта наиболее тонкая задача известна как проблема модуляции и демодуляции.

Практически во всех современных системах связи используются методы цифровой модуляции и цифровая обработка сигналов при демодуляции. Такие системы принято называть **цифровыми системами передачи** в отличие от аналоговых систем, в которых реализованы аналоговая модуляция и аналоговая демодуляция. Современные достижения радиоэлектроники обеспечивают возможность реализовать в передатчике и приемнике системы связи достаточно сложные алгоритмы цифровой обработки электрических сигналов. В результате качество передачи практически любых сообщений в цифровых системах оказывается выше, чем качество передачи этих сообщений с помощью аналоговых систем связи. Например, оказалось **возможным** передавать **сообщения** в присутствии шума и помех с большей точностью или передавать больше сообщений при прочих равных условиях.

Цифровые системы передачи обладают двумя важнейшими особенностями:

- любые сообщения представляются в цифровой форме, т.е. в виде последовательностей битов;
- до формирования канальных символов подлежащие передаче биты обычно сначала преобразуются в последовательность положительных и отрицательных электрических импульсов длительностью T_c прямоугольной формы, для которой принимаем обозначение $V(t)$; последовательность полученных таким способом импульсов называют **модулирующим сигналом**. Преобразование последовательности битов в последовательность электрических импульсов осуществляется по следующему правилу: $0 \Rightarrow bV(t)$, $1 \Rightarrow -bV(t)$, где $b > 0$ – амплитуда импульса. При этом модулирующий сигнал

$$u(t) = \sum_i b_i v(t - iT_c). \quad (1.1)$$

В этом равенстве суммирование осуществляется по всем возможным значениям индекса i , а множитель b_i может принимать значения $+b$ или $-b$.

На рис. 1.1 представлены примеры реализаций сигналов при цифровой модуляции: последовательность переданных битов $\{a_i\}$, последовательность электрических импульсов модулятора прямоугольной формы и разной полярности $\{b_i\}$, восстановленная в приемнике последовательность электрических импульсов демодулятора $\{\hat{b}_i\}$ и, наконец, последовательность принятых битов

$\{\hat{a}_i\}$. Здесь не представлены каналные символы. Угловой скобкой над символами обозначены их оценки. Неверно принятый бит отмечен серым фоном.

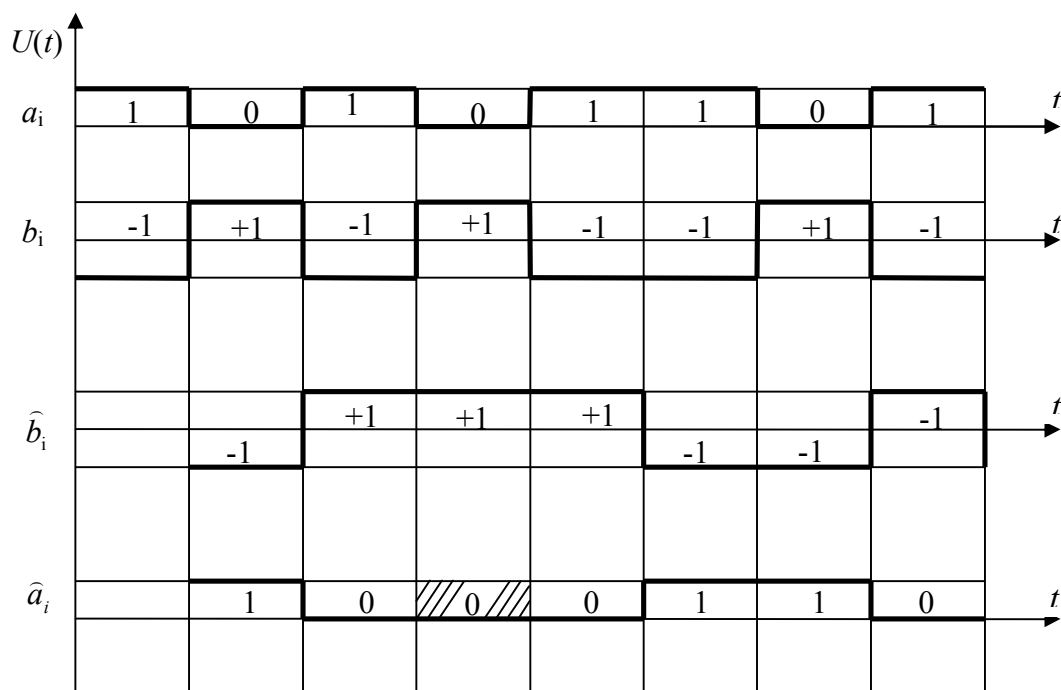


Рис. 1.1. Временные диаграммы сигналов в устройствах цифровой системы передачи

Модуляция – это процесс кодирования информации, получаемой от источника информации, в форму, наиболее удобную для передачи по каналу связи. В общем случае этот процесс предусматривает перенос основной полосы частот модулирующего сигнала ΔF в область высоких частот. Получающийся в результате модуляции **радиосигнал** $S[t; U(t)]$ занимает радиополосу шириной $2\Delta F$, центральная частота которой F_0 намного выше верхней граничной частоты F_B спектра модулирующего сигнала. Как правило, в качестве переносчика информации используется гармоническое колебание $s(t) = A \cos(2\pi f_0 t + \varphi)$, основными параметрами которого, доступными для модуляции, являются амплитуда A , частота F_0 и фаза φ .

На практике наибольшее применение нашли M -ичные системы модуляции, которым будет уделено большое внимание. Это связано с максимальной скоростью передачи информации. Известно, что чем выше частота несущего периодического сигнала, тем больше информации в единицу времени передается по линии связи и тем выше пропускная способность линии при фиксированном способе физического кодирования. Однако с увеличением частоты периодического несущего сигнала увеличивается и ширина спектра этого сигнала. Линия связи передает этот спектр синусоид с теми же искажениями, которые определяются её полосой пропускания. Чем больше несоответствие между полосой пропускания линии и шириной спектра передаваемых инфор-

мационных сигналов, тем больше сигналы искажаются и тем вероятнее ошибки в распознавании информации принимающей стороной, а значит, скорость передачи информации на самом деле оказывается меньше, чем можно было предположить.

Связь между полосой пропускания линии и её максимально возможной пропускной способностью, вне зависимости от принятого способа физического кодирования, установили Шеннон и Хартли. Эта формула называется законом Шеннона-Хартли:

$$C = \Delta F \log \left(\frac{P_x}{P_\varepsilon} + 1 \right), \quad (1.2)$$

где C – максимальная пропускная способность линии, бит/с; ΔF – ширина полосы пропускания линии, Гц; P_x и P_ε – соответственно мощность сигнала и помехи.

Из формулы (1.2) видно, что хотя теоретического предела пропускной способности линии с фиксированной полосой пропускания не существует, на практике такой предел имеется. Действительно, повысить пропускную способность линии можно за счет увеличения мощности передатчика или же уменьшения помех на линии связи. Обе эти составляющие поддаются изменению с большим трудом. К тому же влияние мощностей полезного сигнала и шума на пропускную способность ограничено логарифмической зависимостью, которая растет не так быстро, как прямо-пропорциональная. Так, при достаточно типичном исходном отношении мощности сигнала к мощности шума в 100 раз, повышение мощности передатчика в 2 раза даст только 15% увеличение пропускной способности линии.

Найквист вывел формулу, определяющую зависимость максимальной скорости передачи информации (данных) C [бит/с] от ширины полосы пропускания B без учета шума в канале:

$$C = 2B \log_2 M, \quad (1.3)$$

где M – число различных состояний информативного параметра сигнала. Если сигнал имеет два состояния, то пропускная способность равна удвоенному значению ширины полосы пропускания линии связи. Если же передатчик использует более чем два устойчивых состояния сигнала для кодирования данных, то пропускная способность линии повышается, так как за один такт работы передатчик передает несколько бит исходных данных.

Пример. Модем в телефонной сети общего пользования применяет метод квадратурной амплитудной модуляции с 8-ю уровнями (четыре значения фазы умножить на два значения амплитуды для каждой фазы) на каждый сигнальный элемент. Если полоса пропускания телефонной сети равна 3100 Гц, то согласно формуле Найквиста максимальная скорость передачи данных будет равна:

$$C = 2B \log_2 M = 2 \cdot 3100 \cdot \log_2 8 = 18600 \text{ бит/с.}$$

Хотя формула Найквиста явно не учитывает наличие шума, косвенно его влияние отражается в выборе числа состояний информативного параметра сигнала. Для повышения пропускной способности канала хотелось бы увеличить это число до значительной величины, но на практике этого сделать нельзя из-за шума на линии. Поэтому число возможных состояний сигнала фактически ограничивается соотношением мощности сигнала и шума, а формула Найквиста определяет предельную скорость передачи данных в том случае, когда количество состояний уже выбрано с учетом возможностей устойчивого распознавания приемником. Таким образом, наиболее эффективными являются *M*-ичные системы модуляции, которые обеспечивают повышенные скорости передачи информации.

1.2. М-ичная фазовая модуляция

1.2.1. Общие сведения. При фазовой модуляции мгновенное значение фазы радиосигнала отклоняется от фазы немодулированного несущего колебания на величину, зависящую от мгновенного значения модулирующего сигнала:

$$s[t, u(t)] = A \cos \{2\pi f_0 t + \phi[u(t)]\} = \operatorname{Re}[A \exp \{j\phi[u(t)]\} \exp \{j2\pi f_0 t\}]. \quad (1.4)$$

Из этого выражения следует, что передаваемая информация, содержащаяся в модулирующем сигнале $U(t)$, закодирована в **комплексной огибающей**

$$\dot{A}(t) = A \exp \{j\phi[u(t)]\} \quad (1.5)$$

передаваемого сигнала $S[t, U(t)]$. Понятие комплексной огибающей является очень важным как для теории, так и для техники цифровой связи и будет играть существенную роль в последующем изложении.

При цифровой фазовой модуляции фаза переносчика может отличаться от текущей фазы немодулированного несущего колебания на **конечное число** разных значений. В простейшем случае **двоичной фазовой манипуляции** (ФМ-2) в качестве таких значений обычно выбирают 0° и 180° [1]. В современных системах часто используют большие наборы фазовых углов, чтобы представлять в одном канальном символе сразу несколько битов передаваемых данных. Например, можно использовать четыре разных фазовых угла: 45° , 135° , -45° , -135° для представления возможных значений последовательности из двух битов (ФМ-4). Возможные значения трёхбитового слова можно представить группой из восьми разных фазовых углов (ФМ-8), четырёхбитового слова – группой из 16 фазовых углов (ФМ-16) и т.д.

1.2.2. Двоичная фазовая манипуляция. Простейшей формой цифровой фазовой модуляции является ФМ-2. Этот способ часто используется в системах с прямым расширением спектров, в которых модулирующий сигнал является псевдослучайной двоичной последовательностью. При ФМ-2 в зависи-

мости от значения модулирующего сигнала отклонение фазы сигнала от фазы модулированного несущего колебания равно либо 0° , либо 180° . Так, если для **фазомодулированного сигнала** (ФМ сигнала) принять общее описание (1.4), (1.5), то для ФМ-2 сигнала должны выполняться равенства:

$$\varphi[u(t)] \equiv 0 \text{ при } u(t) \equiv 1, \quad \varphi[u(t)] \equiv \pi \text{ при } u(t) \equiv -1; \quad 0 \leq t \leq T_c.$$

Комплексная огибающая этого сигнала не изменяется на указанном интервале времени и может принимать следующие два значения:

$$\dot{A}(t) \equiv A \text{ при } u(t) \equiv 1, \quad \dot{A}(t) \equiv -A \text{ при } u(t) \equiv -1; \quad 0 \leq t \leq T_c.$$

Полезным и наглядным оказывается графическое представление возможных значений этой огибающей на комплексной плоскости. Для рассматриваемого здесь сигнала комплексная огибающая принимает всего два значения, указанные на рис. 1.2. Такое изображение обычно называют **сигнальным созвездием**.



Рис. 1.2. Сигнальное созвездие ФМ-2 сигнала

На рис 1.3 представлены временные диаграммы модулирующего сигнала и радиосигнала. Основная особенность радиосигнала здесь состоит в том, что его текущая фаза имеет разрывы в моменты изменения полярности модулирующего сигнала.

Эти «скачки» фазы на 180° являются основной причиной того, что **спектральная плотность мощности** ФМ-2 сигнала в радиоканале оказывается существенно отличной от нуля в недопустимо широкой полосе частот. Поэтому в таком виде ФМ-2 сигналы практически не используются. Для уменьшения занимаемой ими полосы частот они подвергаются фильтрации.

Трудно осуществить фильтрацию этих сигналов после модулятора на высокой частоте, поскольку потребовались бы узкополосные высокочастотные

фильтры для каждой частоты несущего колебания. Число таких частот в современных цифровых системах связи с подвижными объектами может достигать нескольких десятков. Поэтому операция фильтрации практически всегда выполняется над модулирующим сигналом до модуляции. Соответствующий фильтр является низкочастотным и единственным, хотя обычно и достаточно сложным. Однако современные достижения радиоэлектроники обеспечивают его реализацию, а большое число частотных каналов в этом случае можно получить, если использовать несущее колебание с набором соответствующих частот. Такой фильтр будем называть **фильтром основной полосы**.

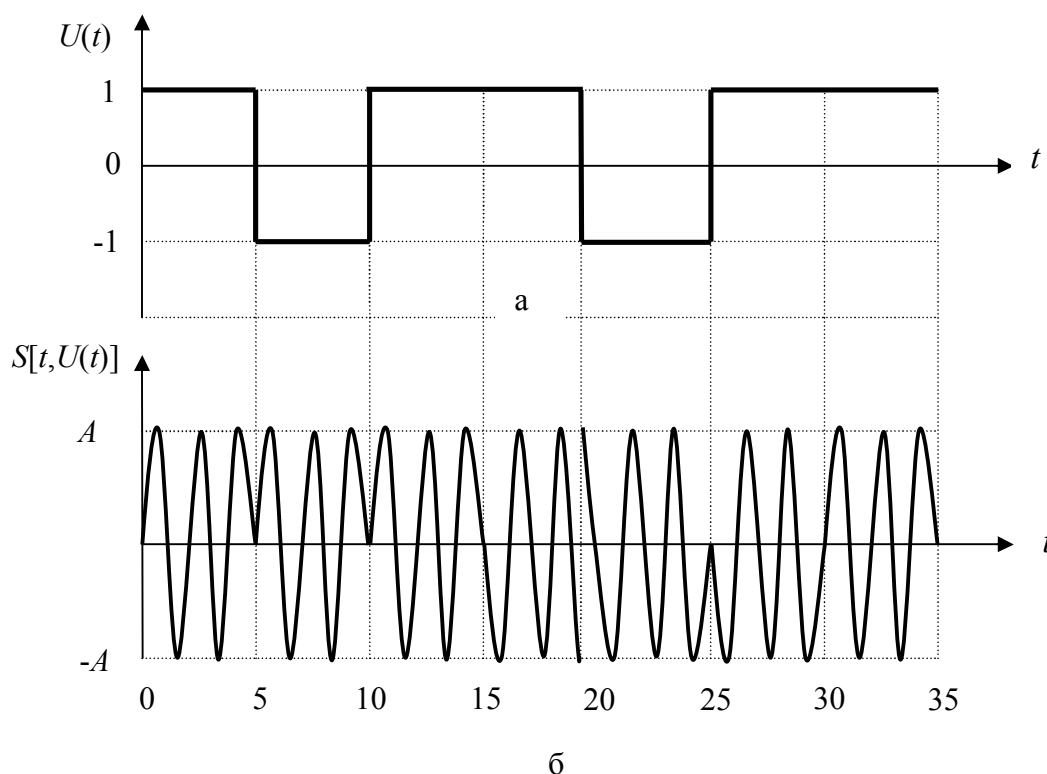


Рис. 1.3. Временные диаграммы модулирующего сигнала (а) и ФМ-2 радиосигнала (б)

Однако при уменьшении полосы частот, занимаемой радиосигналом, путем фильтрации приходится учитывать возникающую при этом проблему межсимвольной интерференции.

На рис. 1.4 представлена упрощенная функциональная схема передатчика, формирующего ФМ-2 радиосигнал

$$\{0\ 0\}, \{0\ 1\}, \{1\ 0\}, \{1\ 1\}.$$

Здесь после модулятора добавлены усилитель мощности радиосигнала и узкополосный высокочастотный фильтр. Основное назначение фильтра состоит в том, чтобы ослабить излучение передатчика на частотах, кратных основной частоте несущего колебания; опасность таких излучений обусловлена нелинейными эффектами в усилителе мощности, которые, как правило, име-

ют место и усиливаются при попытке увеличения эффективности этого усилителя. Часто данный фильтр используется одновременно и для приемника – он подавляет сильные сторонние сигналы вне полосы частот полезных радиосигналов до преобразования частоты «вниз».

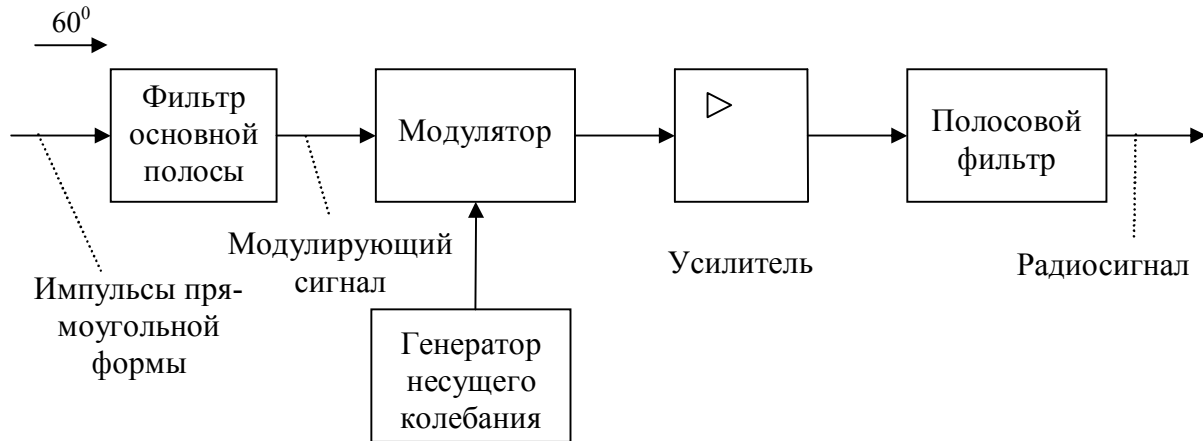


Рис. 1.4. Функциональная схема устройства формирования ФМ-2 радиосигнала

1.2.3. Квадратурная фазовая модуляция. При ФМ-2 один канальный символ переносит один передаваемый бит. Однако, как уже отмечалось выше, один канальный символ может переносить большее число информационных бит. Например, пара следующих друг за другом битов может принимать четыре значения:

Если для передачи каждой пары использовать один канальный символ, то потребуется четыре канальных символа, скажем $\{S_0(t), S_1(t), S_2(t), S_3(t)\}$, так что $M = 4$. При этом скорость передачи символов в канале связи оказывается в два раза ниже, чем скорость поступления информационных битов на вход модулятора и, следовательно, каждый канальный символ теперь может занимать временной интервал длительностью $T_{\text{КС}} = 2T_c$. В частности, при фазовой модуляции в качестве канальных символов можно выбрать следующие радиосигналы:

$$s_i(t) = s[t, \varphi_i(t)] = A \cos[2\pi f_0 t + \varphi_i(t)] = \text{Re}[A \exp\{j\varphi_i(t)\} \exp\{j2\pi f_0 t\}],$$

$$0 \leq t \leq 2T_c,$$

где $\varphi_i(t) \equiv \frac{\pi(2i+1)}{4}$ – отклонение фазы радиосигнала с номером i от фазы не-

модулированного несущего колебания; $\dot{A}_i(t) = A \exp\{j\varphi_i(t)\}$ – комплексная амплитуда этого сигнала на интервале времени $[0, 2T_c]$ для $i = 0, 1, 2, 3$.

В дальнейшем вместо четырех канальных символов или четырех радиосигналов будем говорить о единственном радиосигнале, комплексная ампли-

туда которого может принимать четыре указанных значения, представленных на рис. 1.5 в виде сигнального созвездия.

Каждая группа из двух битов представляется соответствующим фазовым углом, все фазовые углы отстоят друг от друга на 90° . Можно отметить, что каждая сигнальная точка отстоит от действительной или мнимой оси на 45° .

Данный способ модуляции может быть реализован следующим образом. Последовательность передаваемых битов 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, ... разбивается на две подпоследовательности нечетных 0, 1, 0, 0, 0, 1, 0, ... и четных 1, 0, 1, 1, 1, 0, ... битов.

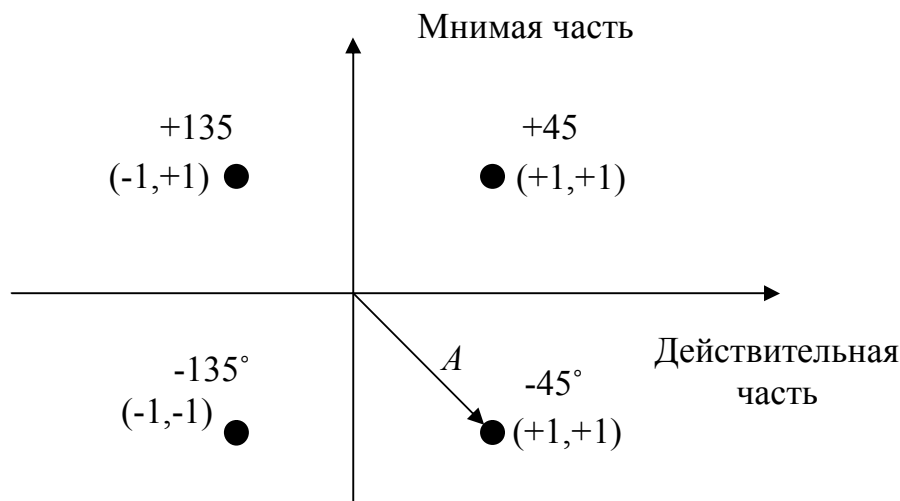


Рис. 1.5. Сигнальное созвездие ФМ-4 радиосигнала

Биты с одинаковыми номерами в этих подпоследовательностях образуют пары, которые удобно рассматривать как комплексные биты; действительная часть комплексного бита есть бит нечетной подпоследовательности, а мнимая часть – бит четной подпоследовательности. Полученные таким способом комплексные биты преобразуются в комплексную последовательность прямоугольных электрических импульсов длительностью $2T_C$ со значениями $+1$ или -1 их действительной и мнимой частей, которые используются для модуляции несущего колебания $\exp\{j2\pi f_0 t\}$. В результате получается ФМ-4 радиосигнал.

Рассмотрим один комплексный бит. Обозначим символом I значение электрического импульса, полученное из действительной части этого бита (это значение бита нечетной подпоследовательности), а символом Q – значение электрического импульса, полученное из мнимой части этого же комплексного бита (это значение соответствующего бита четной подпоследовательности). Отметим, что I и Q могут принимать значения $+1$ или -1 . Очевидно, что можно записать следующие равенства:

$$I \cos(2\pi f_0 t) = I \left[\frac{\exp\{j2\pi f_0 t\} + \exp\{-j2\pi f_0 t\}}{2} \right],$$

$$Q \sin(2\pi f_0 t) = Q \left[\frac{\exp\{j2\pi f_0 t\} - \exp\{-j2\pi f_0 t\}}{2j} \right].$$

Тогда можно сформировать сигнал

$$s(t) = I \cos(2\pi f_0 t) + Q \sin(2\pi f_0 t) = \frac{1}{2} [(I - jQ) \exp\{j2\pi f_0 t\} + (I + jQ) \exp\{-j2\pi f_0 t\}].$$

Если теперь ввести обозначения

$$I + jQ = \sqrt{I^2 + Q^2} e^{j\varphi}, \quad I - jQ = \sqrt{I^2 + Q^2} e^{-j\varphi}, \quad \varphi = \arctg(Q/I),$$

то

$$s(t) = \sqrt{(I^2 + Q^2)} \left[\frac{\exp(j2\pi f_0 t) + \exp(-j2\pi f_0 t + \varphi)}{2} \right] =$$

$$= \sqrt{(I^2 + Q^2)} \cos(2\pi f_0 t - \varphi). \quad (1.6)$$

Таким образом, меняя значения I и Q , можно получить амплитудную и фазовую модуляцию. В частности, если принять, что I и Q могут принимать значения $+1$ или -1 , то амплитуда этого сигнала постоянна и равна $\sqrt{2}$, а фаза φ принимает значения $+45^\circ$, -45° , $+135^\circ$, -135° . В результате для комплексной амплитуды высокочастотного сигнала с такой модуляцией можно записать

$$\dot{A}(t) = \sqrt{2} A \exp[j \frac{\pi}{4} (2i + 1)], \quad i = 0, 1, 2, 3 \text{ на интервале } 0 \leq t \leq 2T_c \quad (1.7)$$

Приведенные выше равенства позволяют формировать сигналы ФМ-4 с помощью устройства, функциональная схема которого приведена на рис. 1.6.

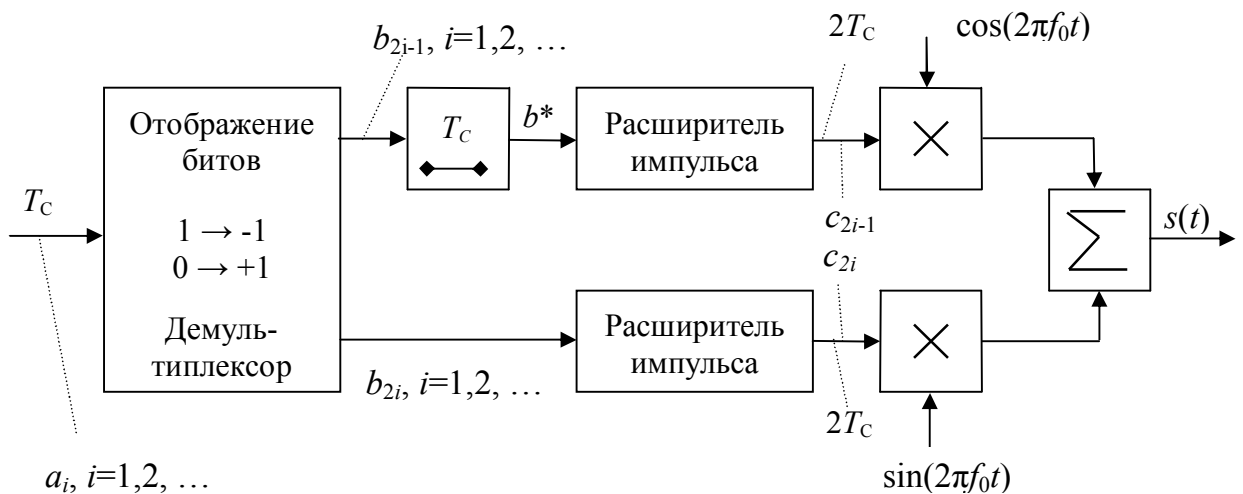


Рис. 1.6. Функциональная схема устройства формирования ФМ-4 радиосигнала

На вход первого блока поступают информационные биты, которые преобразуются в последовательность прямоугольных импульсов положительной и отрицательной полярности длительностью T_c . Эта последовательность в демультиплексоре разбивается на две подпоследовательности импульсов с нечетными и четными фазами, которые направляются в **синфазную и квадратурную** ветви соответственно (рис. 1.7).

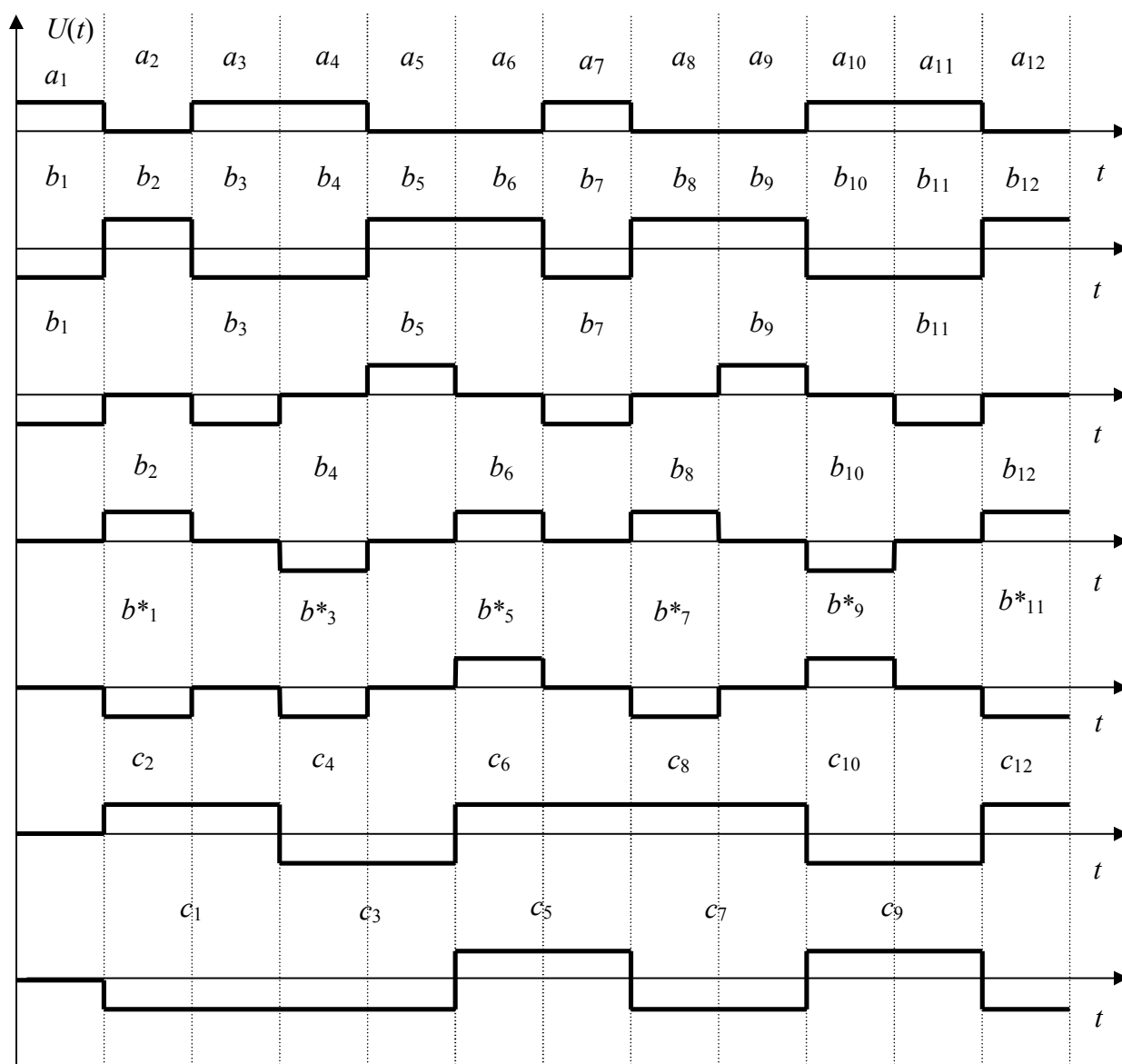


Рис. 1.7. Временные диаграммы при формировании ФМ-4 радиосигнала

Импульсы с нечетными номерами в синфазной ветви задерживаются на время T_c . Далее длительность импульсов каждой подпоследовательности увеличивается до значения $2T_c$, после чего осуществляется перенос на частоту F_0 в каждой ветви. Сложение результатов перемножений завершает процесс формирования ФМ-4 радиосигнала.

Для характеристики некоторых свойств модулированных сигналов часто используют так называемые **диаграммы фазовых переходов**, которые представляют собой графические изображения траекторий перемещений сигнальных точек в сигнальном созвездии при переходе от одного передаваемого канального символа к другому. Для ФМ-4 сигнала такая диаграмма представлена на рис. 1.8.

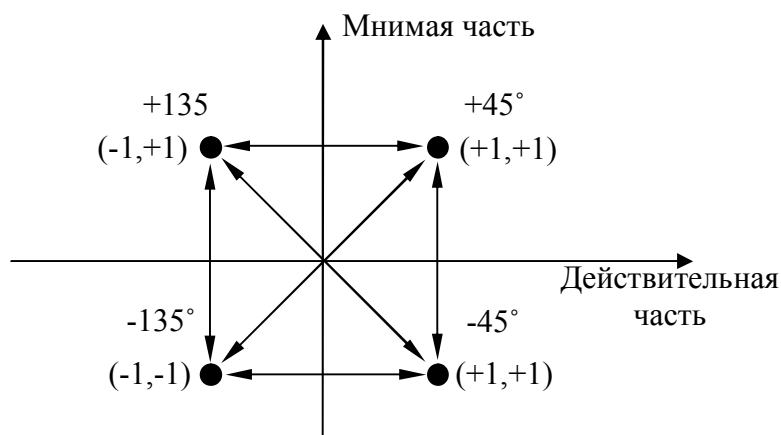


Рис. 1.8. Диаграмма фазовых переходов для ФМ-4 радиосигнала

На этой диаграмме сигнальная точка с координатами $(+1, +1)$ расположена на линии, образующей угол $+45^\circ$ с осями координат, и соответствует передаче символов $+1$ и $+1$ в квадратурных каналах модулятора. Если следующей парой символов будет $(-1, +1)$, которой соответствует угол $+135^\circ$, то из точки $(+1, +1)$ к точке $(-1, +1)$ можно провести стрелку, характеризующую переход фазы радиосигнала от значения $+45^\circ$ к значению $+135^\circ$.

Полезность этой диаграммы можно проиллюстрировать на следующем примере. Из рис. 1.8 следует, что четыре фазовые траектории проходят через начало координат. Например, переход из точки сигнального созвездия $(+1, +1)$ в точку $(-1, -1)$ означает изменение мгновенной фазы высокочастотного несущего колебания на 180° . Поскольку на выходе модулятора обычно устанавливают узкополосный высокочастотный фильтр, то такое изменение фазы сигнала сопровождается существенным изменением значений огибающей сигнала на выходе этого фильтра и, следовательно, во всей линии передачи. Непостоянство значений огибающей радиосигнала по многим причинам является нежелательным в цифровых системах передачи.

1.2.4. Квадратурная фазовая модуляция со смещением. Уже отмечалось, что при фазовой модуляции возможны изменения мгновенных значений фазы высокочастотного несущего колебания на $+180^\circ$, в результате чего могут возникнуть значительные изменения значений огибающей радиосигнала. Эти изменения оказываются не столь значительными для сигналов с квадратурной фазовой модуляцией со смещением.

Функциональная схема устройства формирования такого радиосигнала изображена на рис. 1.9.

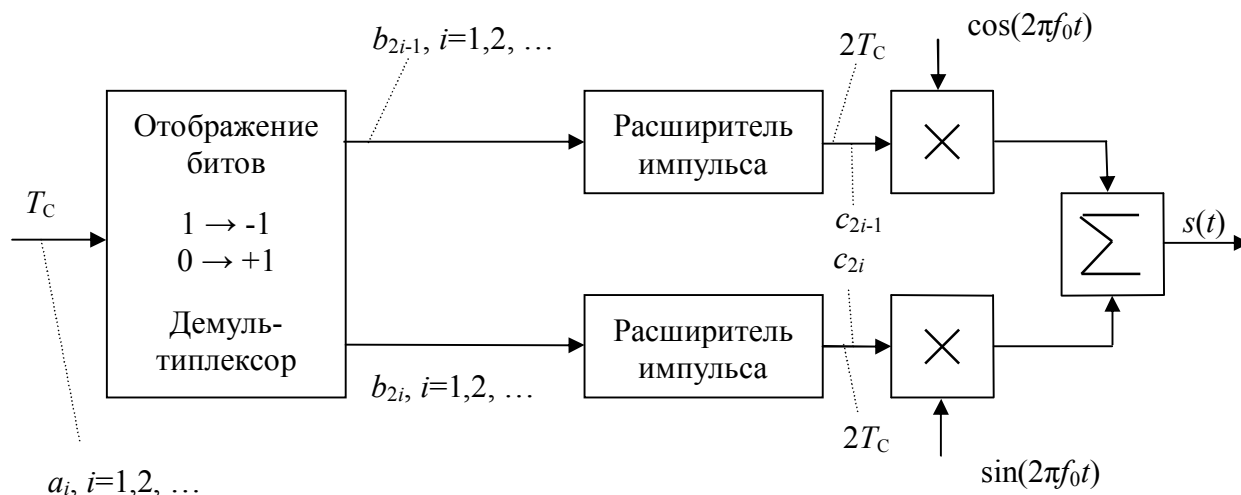


Рис. 1.9. Функциональная схема устройства формирования ФМ-4 радиосигнала со смещением

Этот способ формирования сигнала практически полностью аналогичен квадратурному способу формирования ФМ сигнала, однако с той лишь разницей, что подпоследовательность в квадратурной ветви сдвигается во времени (задерживается) на время T_C или, что эквивалентно, на половину длительности канального символа. Для реализации этого способа достаточно незначительно модифицировать функциональную схему, представленную на рис. 1.6, так, как это показано на рис. 1.9, необходимо удалить элемент задержки на время T_C в синфазной ветви. При таком изменении квадратурная подпоследовательность канальных символов окажется задержанной на время T_C относительно синфазной подпоследовательности.

На рис. 1.10 представлены временные диаграммы последовательностей информационных битов и соответствующих канальных символов для этой функциональной схемы. Основное отличие этих диаграмм от аналогичных диаграмм (рис. 1.7) в функциональной схеме рис. 1.6 состоит в том, что теперь изменения уровней сигнала в квадратурных каналах на выходе расширителей не могут происходить одновременно. В результате на диаграмме фазовых переходов для данного метода модуляции отсутствуют траектории, проходящие через начало координат. Это означает, что мгновенная фаза радиосигнала не имеет скачков на $+180^\circ$ и, следовательно, огибающая этого сигнала не имеет глубоких провалов, как это имело место при квадратурной ФМ-4.

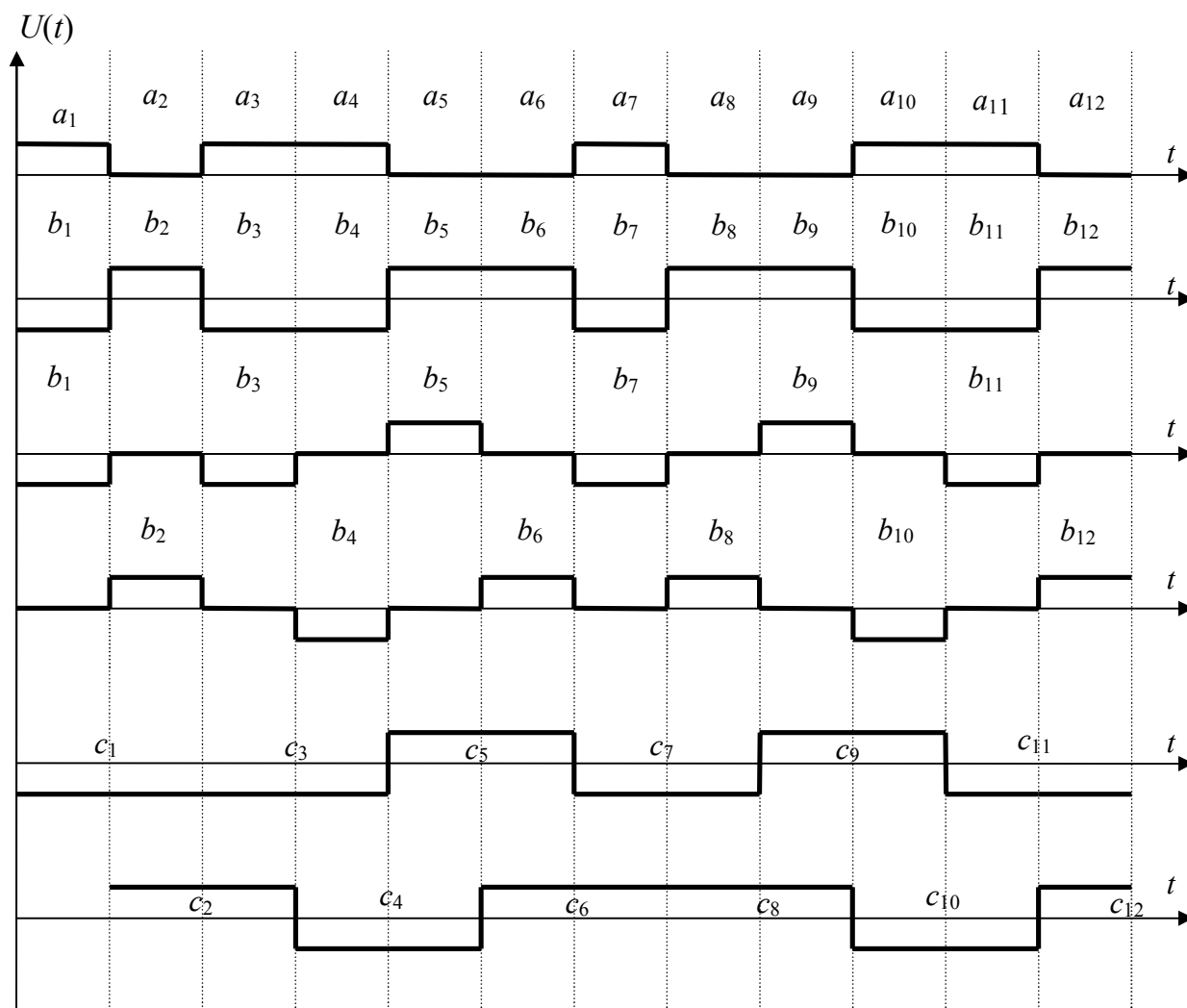


Рис. 1.10 Временные диаграммы при формировании ФМ-4 радиосигнала со смещением

Диаграмма фазовых переходов ФМ-4 радиосигнала со смещением представлена на рис. 1.11.

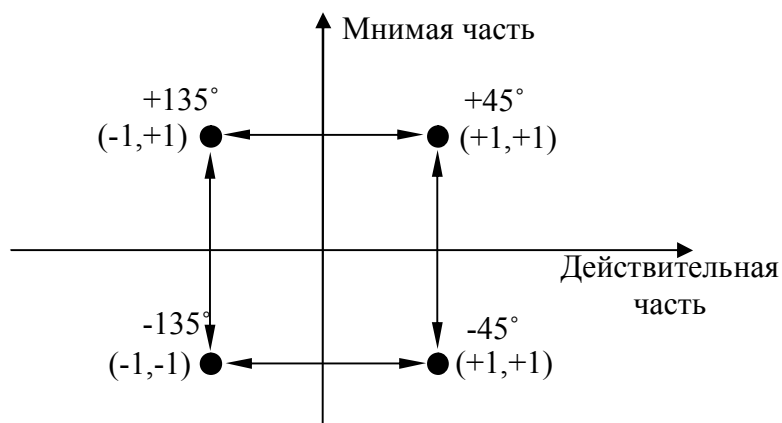


Рис. 1.11. Диаграмма фазовых переходов ФМ-4 радиосигнала со смещением

1.2.5. ФМ-8 сигналы. Поток информационных битов, поступающих на вход модулятора, можно разбивать на группы по 3, 4 бита и т.д., формируя затем ФМ-8, ФМ-16 сигналы и т.д. На рис. 1.12 изображено сигнальное созвездие для ФМ-8 радиосигнала.

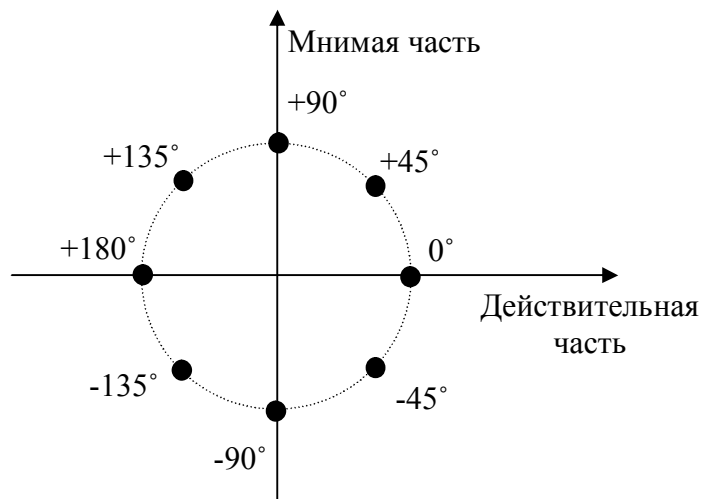


Рис. 1.12. Сигнальное созвездие для ФМ-8 радиосигнала

Для этого способа модуляции необходимо иметь восемь канальных символов, начальные фазы которых отличаются от мгновенной фазы немодулированного несущего колебания на угол, кратный 45° . Если амплитуды всех канальных символов одинаковы, то сигнальные точки располагаются на окружности. Возможные значения вещественных и мнимых частей комплексных амплитуд этих символов при этом пропорциональны коэффициентам I и Q , принимающим значения из множества

$$\left\{ -\frac{\sqrt{2}}{2}, -1, 0, +1, +\frac{\sqrt{2}}{2} \right\}. \quad (1.8)$$

Не совсем простым является вопрос об установлении соответствий между точками сигнального созвездия и тройками информационных битов. Этот процесс обычно называют **сигнальным кодированием**.

В табл. 1.1 приведён пример такого соответствия, который является возможным, но не наилучшим, поскольку для установления наилучшего соответствия необходимо сначала определить способ демодуляции такого сигнала в присутствии помехи, а затем вычислить вероятность ошибки при приеме либо одного канального символа, либо одного информационного бита. Наилучшим можно назвать тот способ сигнального кодирования, при котором вероятность ошибки оказывается наименьшей.

Таблица 1.1

Значения начальной фазы при ФМ-8	Значения коэффициентов		Группы из трех информационных символов (битов)
	I	Q	
0°	1	0	0 1 1
$+45^\circ$	$\sqrt{2}/2$	$\sqrt{2}/2$	0 0 1
$+90^\circ$	0	1	0 0 0
$+135^\circ$	$-\sqrt{2}/2$	$\sqrt{2}/2$	1 0 0
$+180^\circ$	-1	0	1 0 1
-135°	$-\sqrt{2}/2$	$-\sqrt{2}/2$	1 1 1
-90°	0	-1	1 1 0
-45°	$\sqrt{2}/2$	$-\sqrt{2}/2$	0 1 0

На рис. 1.13 приведена функциональная схема устройства формирования ФМ-8 радиосигнала.

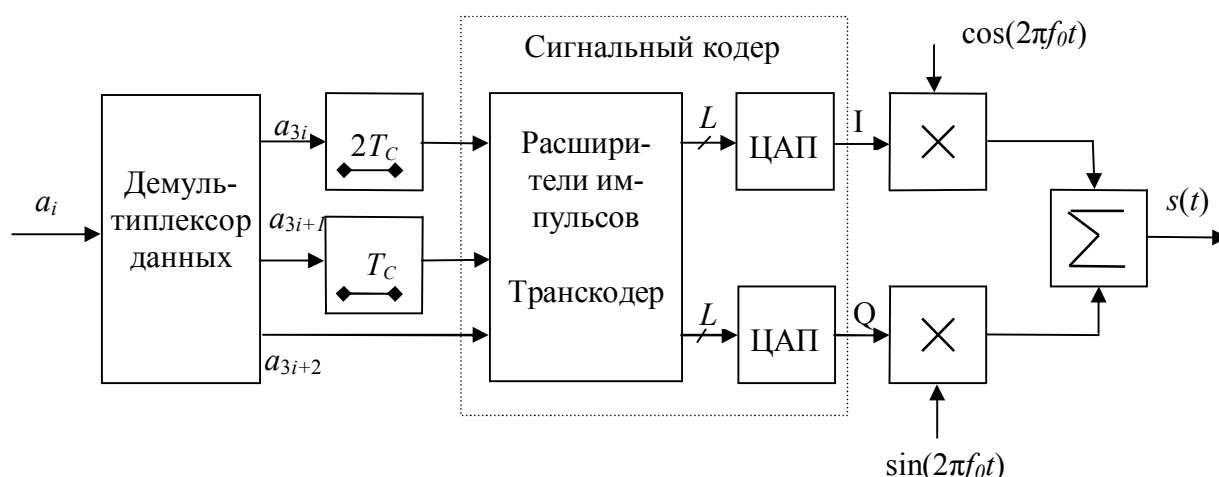


Рис. 1.13. Функциональная схема устройства формирования ФМ-8 радиосигнала

Основными здесь являются устройства, аналогичные соответствующим устройствам уже рассмотренных ранее модуляторов: демультиплексор распределяет входной поток информационных битов длительностью T_c на три подпоследовательности, элементы задержек выравнивают во времени эти подпоследовательности, расширители увеличивают длительность каждого символа до значения длительности канального символа $T_{KC} = 3T_c$. Сигнальное кодирование в этом случае сводится к вычислению значений синфазной и квадратурной компонент комплексной огибающей ФМ-8 радиосигнала. Эта операция выполняется сигнальным кодером, в состав которого входит транскодер, имеющий два цифровых выхода с L -битовыми словами, которые в цифро-аналоговых преобразователях (ЦАП) преобразуются в аналоговые величины с требуемыми значениями (1.8).

1.2.6. $\pi/4$ - квадратурная относительная фазовая модуляция. При квадратурной ФМ-4 и ФМ-4 со смещением максимальное изменение мгновенной фазы радиосигнала равно 180° и 90° соответственно. В настоящее время достаточно широко используется **$\pi/4$ -квадратурная относительная фазовая модуляция**, при которой максимальный скачок фазы равен 135° , а все возможные значения мгновенной фазы радиосигнала кратны значению $\pi/4$. Ни одна траектория фазовых переходов для этого способа модуляции не проходит через начало координат. В результате огибающая радиосигнала имеет меньшие провалы по сравнению с квадратурной фазовой модуляцией. Функциональная схема устройства формирования такого радиосигнала представлена на рис. 1.14.

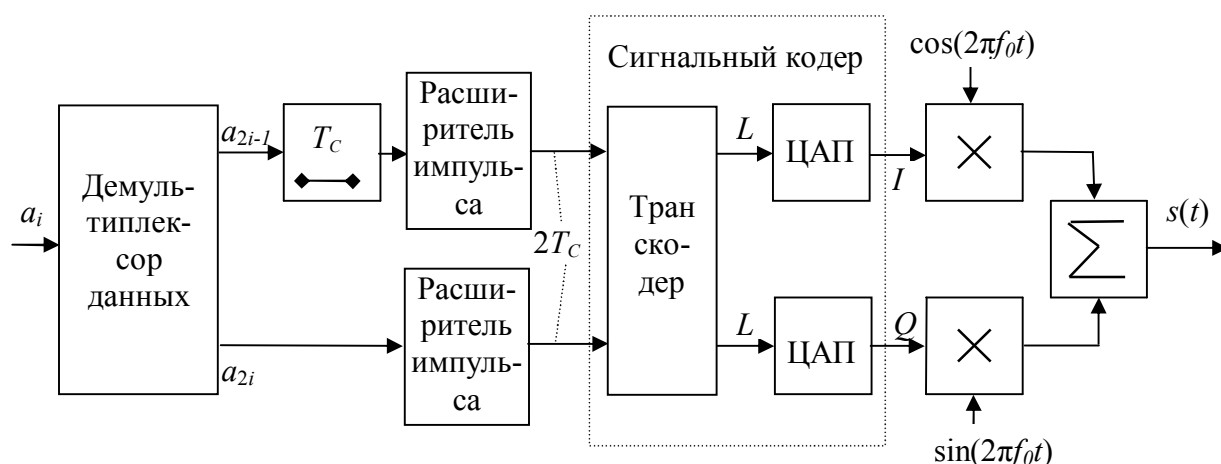


Рис. 1.14. Функциональная схема устройства формирования радиосигнала с $\pi/4$ -квадратурной относительной фазовой модуляцией

Последовательность информационных битов $\{a_i, i=1,2,\dots\}$ разбивается на две подпоследовательности: нечётных $\{a_{2i-1}, i=1,2,\dots\}$ и чётных $\{a_{2i}, i=1,2,\dots\}$ битов, из которых биты выбираются парами. Каждая новая пара таких битов определяет **приращение фазы** несущего колебания на величину $\Delta\varphi_i$ в соответствии с табл. 1.2.

Таблица 1.2

Значения информационных битов		Приращение фазы несущего колебания ($\Delta\varphi_i$)
a_{2i-1}	a_{2i}	
0	0	$\pi/4$
1	0	$3\pi/4$
1	1	$-3\pi/4$
0	1	$-\pi/4$

Если ввести обозначение φ_{i-1} для отклонения фазы радиосигнала от фазы немодулированного несущего колебания на предыдущем интервале, то новые значения отклонения фазы этого сигнала и комплексной амплитуды на текущем интервале определяются равенствами:

$$\varphi_i = \varphi_{i-1} + \Delta\varphi_i, \quad \dot{A}_i = A e^{j\varphi_i}.$$

В результате значения вещественной и мнимой частей комплексной огибающей этого сигнала на текущем интервале времени длительностью $2T_c$ оказываются равными:

$$\begin{aligned} I_i &= A \cos(\varphi_i) = A \cos(\varphi_{i-1} + \Delta\varphi_i) = \\ &= A \cos(\varphi_{i-1}) \cos(\Delta\varphi_i) - A \sin(\varphi_{i-1}) \sin(\Delta\varphi_i) \\ &= I_{i-1} \cos(\Delta\varphi_i) - Q_{i-1} \sin(\Delta\varphi_i); \end{aligned} \quad (1.9)$$

$$\begin{aligned} Q_i &= A \sin(\varphi_i) = A \sin(\varphi_{i-1} + \Delta\varphi_i) = \\ &= A \sin(\varphi_{i-1}) \cos(\Delta\varphi_i) + A \cos(\varphi_{i-1}) \sin(\Delta\varphi_i) = \\ &= Q_{i-1} \cos(\Delta\varphi_i) + I_{i-1} \sin(\Delta\varphi_i) \end{aligned} \quad (1.10)$$

Равенства (1.9), (1.10) можно записать в более наглядной форме для значений комплексных амплитуд канальных символов на двух соседних интервалах с номерами $(i-1)$ и i :

$$\begin{aligned} \dot{A}_i &= A \exp\{j\varphi_i\} = A \exp\{j[\varphi_{i-1} + \Delta\varphi_i]\} = \\ &= A \exp\{j\varphi_{i-1}\} \exp\{j\Delta\varphi_i\} = \dot{A}_{i-1} \exp\{j\Delta\varphi_i\}. \end{aligned} \quad (1.11)$$

Из последнего равенства следует, что возможные значения фазы на интервале с номером i зависят от значения фазы радиосигнала на интервале с номером $(i-1)$. В соответствии с табл. 1.2 новые значения кратны $\pi/2$.

На рис. 1.15, а изображено созвездие возможных сигнальных точек для интервала с номером i , если $\varphi_{i-1} = k\pi/2$; аналогичное созвездие для случая, когда $\varphi_{i-1} = k\pi/2 + \pi/4$, представлено на рис. 1.15, б. Общее созвездие сигнальных точек для данного способа модуляции изображено на рис. 1.15, в и получается путем наложения рис. 1.15, а, б друг на друга. На рис. 1.15, в не указаны стрелками направления переходов, поскольку для каждого перехода возможны направления в обе стороны.

Важно также подчеркнуть, что при данном способе модуляции каждая новая пара информационных битов определяет не полную фазу несущего колебания, а лишь приращение этой фазы для интервала с номером i относительно полной фазы комплексной огибающей на интервале с номером $(i-1)$. Такие методы модуляции называются **относительными**.

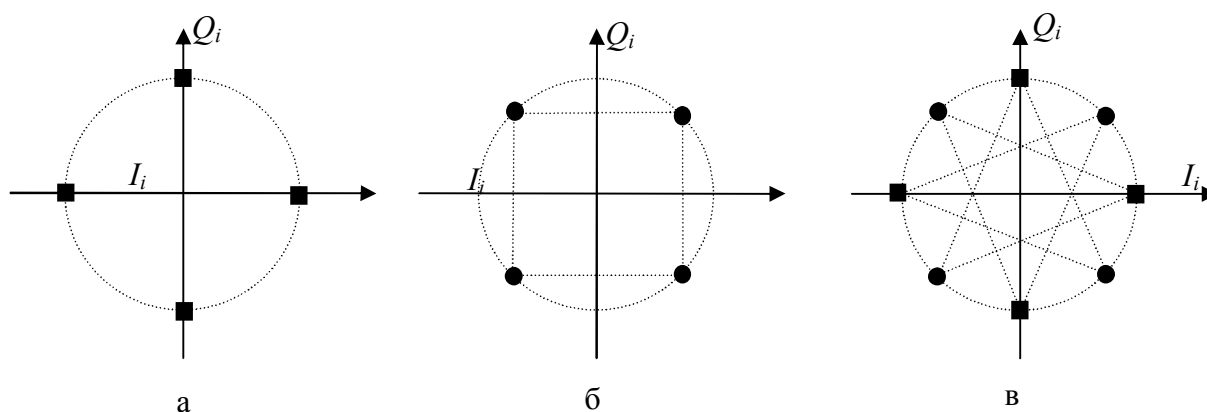


Рис. 1.15. Сигнальные созвездия радиосигнала с $\pi/4$ -квадратурной относительной модуляцией

1.3. М-ичная частотная модуляция

1.3.1. Сигналы с постоянной огибающей. Ранее рассматривались методы модуляции, в которых фаза несущего колебания изменялась скачком от одного возможного значения к другому в соответствии с изменением значения модулирующего сигнала. Отмечалось, что при таких изменениях фазы возможны значительные изменения амплитуды радиосигнала, которые приводят к заметному снижению как средней мощности радиосигнала, так и спектральной эффективности системы связи. Следует также отметить, что находят применение многопозиционные методы модуляции, в которых отдельные значения мгновенной фазы представляют группы информационных битов, что позволяет снизить скорость передачи канальных символов.

Вместо разбиения интервала возможных значений мгновенной фазы несущего колебания на небольшие интервалы и переходов между ними скачками можно переходить от одного значения фазы к другому плавно по какому-либо закону. Если это сделать таким образом, чтобы сигнальная точка оставалась на окружности радиусом единица, то можно получить радиосигнал с постоянным значением амплитуды.

Многие современные системы связи с подвижными объектами используют методы модуляции, которые обеспечивают формирование радиосигнала с постоянным значением амплитуды несущего колебания при меняющихся значениях модулирующего сигнала. Известно несколько таких методов модуляции, которые обеспечивают системам связи ряд следующих положительных свойств:

- возможность использования усилителей мощности класса C без риска расширения полосы занимаемых частот в радиоканале; известно, что усилители этого класса являются наиболее экономичными с точки зрения потребляемой энергии при прочих равных характеристиках;

– низкий уровень внеполосных излучений» достигающий значений от -60 до -70 дБ, что позволяет уменьшить защитный частотный интервал между соседними частотными каналами;

– возможность использования простых устройств демодуляции, содержащих устройства ограничения уровня принимаемого сигнала, что упрощает проектирование приемных устройств и обеспечивает устойчивый прием в условиях значительных замираний принимаемого сигнала.

Для методов модуляции с постоянным уровнем несущего колебания, при всех их достоинствах, все же необходима большая полоса частот в радиоканале по сравнению с линейными методами модуляции. Поэтому эти методы следует использовать в тех случаях, когда энергетическая эффективность системы связи более важна, чем спектральная.

Напомним, что мгновенная частота любого узкополосного колебания

$$s(t) = A(t) \cos[2\pi f_0 t + \varphi(t)]$$

может быть определена как производная по времени полной мгновенной фазы:

$$\frac{d}{dt}[2\pi f_0 t + \varphi(t)] = 2\pi f_0 + \frac{d}{dt}\varphi(t) = \varpi_0(t) = \varpi_0 + \Omega(t).$$

Поэтому фазовую модуляцию с непрерывным гладким изменением фазы можно рассматривать как частотную модуляцию. В этом случае частота несущего колебания является параметром, значение которого должно изменяться в зависимости от значения модулирующего сигнала.

1.3.2. Двоичная частотная манипуляция. При двоичной частотной манипуляции частота несущего колебания с постоянной амплитудой может иметь два возможных значения и изменяется скачками в соответствии со значениями модулирующего сигнала [2]. В зависимости от того, каким образом изменения частоты вводятся в передаваемое высокочастотное колебание, получающийся частотно-модулированный сигнал (ЧМ сигнал) будет иметь либо разрывную, либо непрерывно изменяющуюся мгновенную фазу между двумя соседними битами. В общем случае ЧМ сигнал можно представить следующим образом:

$$s(t) = A \cos[(2\pi f_0 + 2\pi \Delta f)t], \quad 0 \leq t \leq T_c \quad (\text{при передаче 1}),$$

$$s(t) = A \cos[(2\pi f_0 - 2\pi \Delta f)t], \quad 0 \leq t \leq T_c \quad (\text{при передаче 0}),$$

где $2\pi \Delta f$ определяет смещение частоты от её номинального значения.

Очевидный способ формирования ЧМ сигнала состоит в том, чтобы коммутировать выходные сигналы двух независимых генераторов двух гармонических колебаний в соответствии со значениями модулирующего сигнала (рис. 1.16). В этом случае формируемый радиосигнал будет иметь разрывную фазу в момент переключения; такие сигналы обычно называют ЧМ сигналами с разрывной фазой, которые можно представить следующими выражениями:

$$s(t) = A \cos[(2\pi f_e + \varphi_1)], \quad 0 \leq t \leq T_C \quad (\text{при передаче 1}), \quad (1.12)$$

$$s(t) = A \cos[(2\pi f_n + \varphi_2)], \quad 0 \leq t \leq T_C \quad (\text{при передаче 0}). \quad (1.13)$$

Разрывность фазы здесь является нежелательным свойством радиосигнала, приводящим к расширению спектра в радиоканале. Поэтому такой способ модуляции не используется в современных системах связи с подвижными объектами.

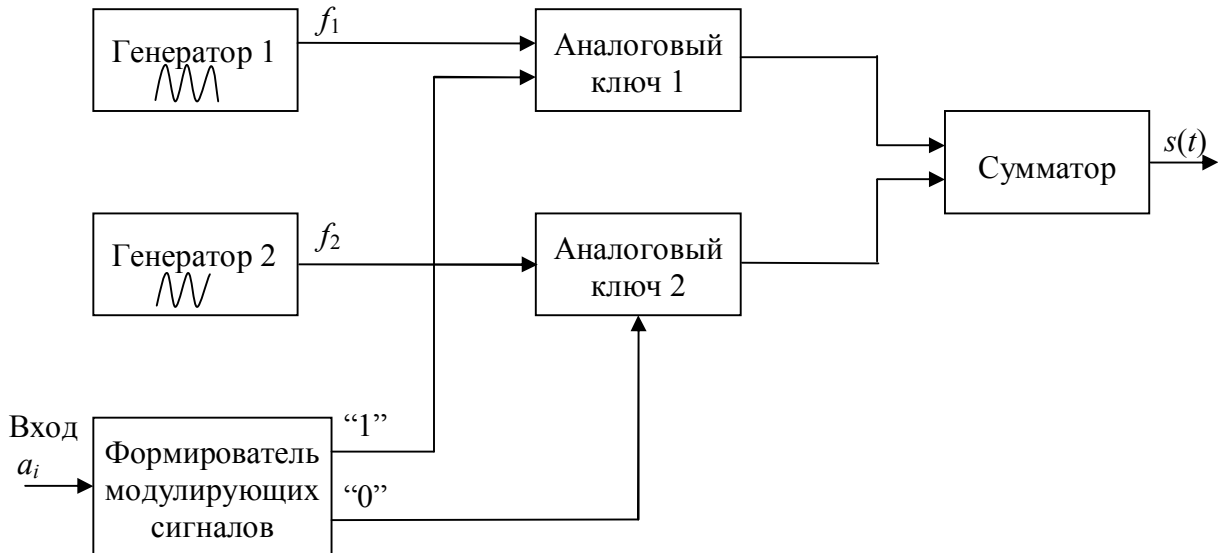


Рис. 1.16. Структурная схема частотного манипулятора с разрывом фазы

Более общий метод формирования ЧМ сигнала заключается в том, что используется один генератор несущего колебания, мгновенная частота которого изменяется в соответствии с изменениями модулирующего сигнала. Этот способ модуляции аналогичен методу формирования ЧМ сигнала при аналоговом модулирующем сигнале, однако в этом случае модулирующий сигнал является цифровым и принимает всего два возможных значения. Для такого радиосигнала можно записать

$$s[t; u(t)] = A \cos[2\pi f_0 t + \varphi(t)] = A \cos[2\pi f_0 t + k_f \int_{-\infty}^t u(\tau) d\tau]. \quad (1.14)$$

Важно отметить, что при разрывном модулирующем сигнале $u(t)$ отклонение фазы $\varphi(t)$ от фазы несущего колебания пропорционально интегралу от $u(t)$ и, следовательно, является непрерывной функцией.

На рис. 1.17 представлена функциональная схема устройства формирования ЧМ сигнала при двоичном модулирующем сигнале. Основным элементом этого модулятора является генератор гармонического несущего колебания, частота которого может управляться напряжением модулирующего сигнала (ГУН – генератор, управляемый напряжением).



Рис. 1.17. Функциональная схема устройства формирования ЧМ сигнала с непрерывной фазой

Поток информационных битов сначала преобразуется в модулирующий сигнал $U(t)$ – последовательность прямоугольных импульсов положительной и отрицательной полярности, амплитуды которых выбираются такими, чтобы обеспечить требуемое значение индекса частотной модуляции

$$m_{\text{ЧМ}} = \frac{f_{\text{В}} - f_{\text{Н}}}{F_{\text{С}}}, \quad (1.15)$$

где $F_{\text{С}} = 1/T_{\text{С}}$ обычно называют **частотой манипуляции**.

Начальная фаза несущего колебания в каждом канальном символе в данном случае не определена; поэтому данный модулятор формирует некогерентный ЧМ сигнал. Полосовой фильтр ослабляет возможные внеполосные гармонические колебания, которые могут появиться из-за нелинейности динамической характеристики усилителя.

На рис. 1.18 тонкими линиями изображена **фазовая решетка** ЧМ сигнала с непрерывной фазой. Жирной ломаной линией здесь представлена возможная фазовая **траектория** – отклонения мгновенной фазы сигнала от текущей фазы немодулированного несущего колебания. Эта траектория соответствует последовательности импульсов положительной и отрицательной полярности модулирующего сигнала, указанной на этом же рисунке вдоль оси времени.

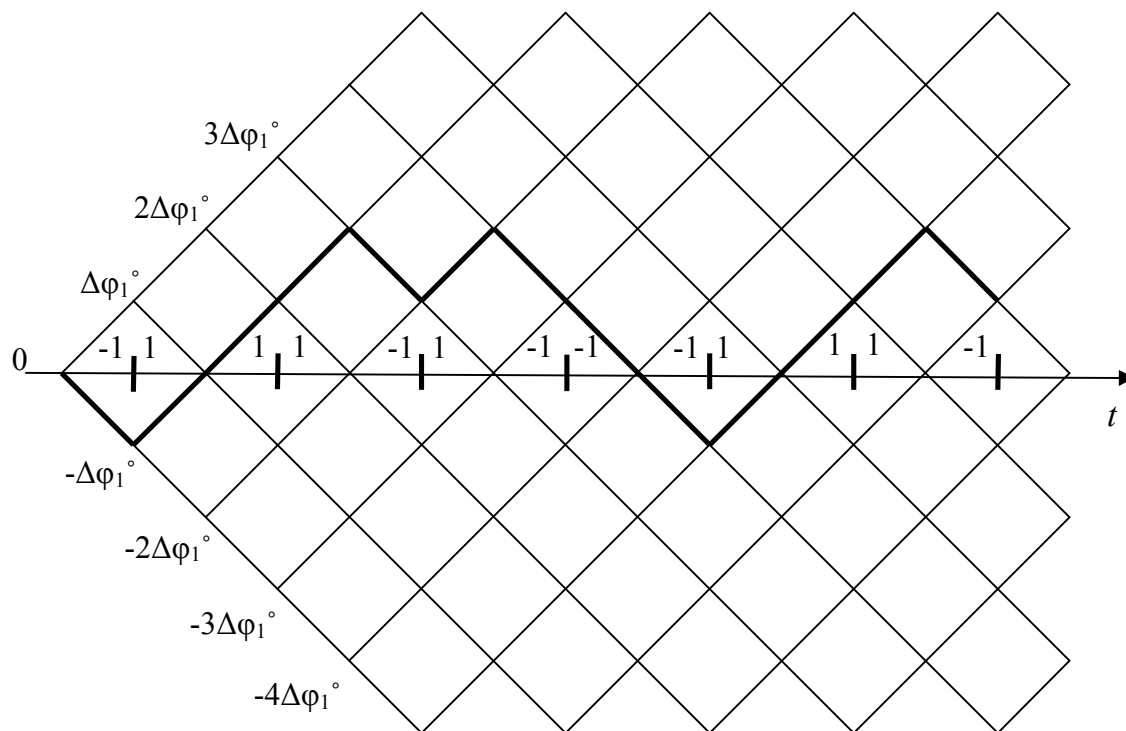


Рис. 1.18. Фазовые решётка и фазовая траектория ЧМ сигнала с непрерывной фазой

Отрезки траектории с положительным значением производной этой траектории по времени соответствуют более высокой частоте несущего колебания $f_B = f_0 + \Delta f$, а отрезки с отрицательным значением производной – более низкой частоте $f_H = f_0 - \Delta f$ по сравнению с частотой немодулированного несущего колебания. Можно найти и значение этого смещения, поскольку для такого сигнала можно записать следующее представление:

$$s[t; u(t)] = A \cos[2\pi f_0 t + 2\pi k_f \int_0^t u(\tau) d\tau] = A \cos[2\pi f_0 t + 2\pi k_f b_1 t], \quad 0 < t \leq T_C,$$

где b_1 является значением модулирующего сигнала на первом временном интервале длительностью T_C и может принимать значения +1 или -1. Вычисляя производную по времени от текущей фазы сигнала, получаем

$$\frac{d}{dt}[2\pi f_0 t + 2\pi k_f b_1 t] = 2\pi(f_0 + k_f b_1),$$

так что сдвиг частоты $\Delta f = k_f b_1$. При $b_1 = +1$ частота несущего колебания равна $f_B = f_0 + k_f |b_1|$; при $b_1 = -1$ имеем $f_B = f_0 - k_f |b_1|$. В результате $f_B - f_H = 2k_f b_1$ и индекс частотной модуляции

$$m_{\text{ЧМ}} = \frac{f_B - f_H}{F_C} = \frac{2k_f |b_1|}{F_C}. \quad (1.16)$$

В момент времени $t = T_C$ отклонение фазы радиосигнала от фазы немодулированного несущего колебания $\Delta\phi_1 = 2\pi k_f b_1 T_C$; в результате ЧМ сигнал с непрерывной фазой на втором интервале $T_C < t < 2T_C$ можно записать следующим образом:

$$s(t) = A \cos[2\pi f_0 t + 2\pi k_f b_1 T_C + 2\pi k_f b_2 (t - T_C)], \quad T_C < t \leq 2T_C.$$

Теперь очевидно, что на i -м интервале этот сигнал можно представить выражением

$$s(t) = A \cos\left\{2\pi f_0 t + 2\pi k_f T_C \left[\sum_{k=1}^{i-1} b_k + b_i \frac{[t - (i-1)T_C]}{T_C}\right]\right\}, \quad (i-1)T_C < t \leq iT_C, \quad (1.17)$$

для $i = 1, 2, 3, \dots$

1.3.3. Частотная манипуляция с минимальным сдвигом. Манипуляция с минимальным сдвигом (ММС) может рассматриваться как фазовая или как частотная модуляция с непрерывной фазой. Основная особенность этого способа модуляции состоит в том, что приращение фазы несущего колебания на интервале времени, равном длительности T_C одного символа, всегда равно

$+90^\circ$ или -90° в зависимости от знаков символов модулирующего сигнала. Например, фаза несущего колебания в начале очередного импульса модулирующего сигнала равна φ_0 ; далее, фаза несущего колебания, линейно нарастая к концу этого импульса, достигает значения $\varphi_0 + 90^\circ$, или, линейно убывая, к концу импульса достигает значения $\varphi_0 - 90^\circ$. Поскольку на интервале каждого очередного импульса модулирующего сигнала мгновенная фаза несущего колебания, отклоняясь от фазы немодулированного гармонического колебания, изменяется линейно, увеличиваясь или уменьшаясь, то мгновенная частота такого радиосигнала будет изменяться скачками. Таким образом, ММС сигнал является частным случаем ЧМ сигнала с непрерывной фазой.

Сигнал с ММС может быть сформирован с помощью квадратурной схемы, которая лишь незначительно отличается от приведенных в предыдущих разделах. Последовательность символов $\{b_i, i = 1, 2, \dots\}$, принимающих значения $+1$ или -1 , разбивается на две подпоследовательности нечётных $\{b_{2i-1}, i = 1, 2, \dots\}$ и чётных $\{b_{2i}, i = 1, 2, \dots\}$ символов. Из этих подпоследовательностей формируются квадратурные компоненты модулирующего сигнала

$$u_I(t) = \sum_i b_{2i-1} v[t - (2i-1)T_C], \quad u_Q(t) = \sum_i b_{2i} v[t - (2i)T_C], \quad (1.18)$$

где

$$v(t) = \begin{cases} 1, & 0 < t \leq 2T_C, \\ 0, & \text{вне этого интервала.} \end{cases} \quad (1.19)$$

Теперь сформируем сигнал в соответствии со следующим представлением:

$$s(t) = A[u_I(t) \cos\left(\frac{\pi t}{2T_C}\right) \cos(2\pi f_0 t) + u_Q(t) \sin\left(\frac{\pi t}{2T_C}\right) \sin(2\pi f_0 t)]. \quad (1.20)$$

Здесь F_0 – частота несущего колебания; функции $\cos\left(\frac{\pi t}{2T_C}\right)$ и $\sin\left(\frac{\pi t}{2T_C}\right)$

на интервале длительностью $2T_C$ имеют полуволну и фактически сглаживают прямоугольную форму импульсов квадратурных компонентов модулирующего сигнала.

Покажем, что сигнал (1.20) является ЧМ сигналом с постоянной огибающей и непрерывной фазой. Действительно, так как слагаемые в правой части равенства (1.20) являются квадратурными компонентами с несущими колебаниями $\cos(2\pi f_0 t)$ и $\sin(2\pi f_0 t)$, то значение огибающей этого сигнала в произвольный момент времени можно определить равенством

$$A_s(t) = \sqrt{A^2 u_I^2(t) \cos^2\left(\frac{\pi t}{2T_C}\right) + A^2 u_Q^2(t) \sin^2\left(\frac{\pi t}{2T_C}\right)} = A,$$

поскольку в соответствии с (1.18) и (1.19) для любого момента времени

$$u_I^2(t) = u_Q^2(t) = 1.$$

Отклонение фазы сигнала (1.20) от фазы несущего колебания на любом интервале времени длительностью T_C является линейным, приращение фазы на этом интервале равно 90° . Чтобы убедиться в этом, достаточно определить общую фазу этого сигнала на произвольном интервале времени $(i-1)T_C < t < iT_C$. Учитывая постоянство огибающей, узкополосный сигнал (1.20) можно записать следующим образом:

$$s(t) = A \cos[2\pi f_0 t + \Phi(t)]. \quad (1.21)$$

В этом случае для отклонения фазы $\Phi(t)$ на первом интервале на основании (1.20) имеем

$$\begin{aligned} \Phi(t) &= \arctg \left[\frac{A u_Q(t) \sin(\frac{\pi t}{2T_C})}{A u_I(t) \cos(\frac{\pi t}{2T_C})} \right] = \\ &= \arctg[u_I(t) u_Q(t) \tg(\frac{\pi t}{2T_C})], \quad 0 < t \leq T_C \end{aligned} \quad (1.22)$$

Из (1.18) следует, что произведение $u_I(t) u_Q(t)$ для любого момента времени на рассматриваемом интервале имеет постоянное значение, равное либо +1, либо -1. Поэтому можно записать для первого интервала

$$\Phi(t) = \begin{cases} \frac{\pi t}{2T_C} & \text{для } u_I(t) u_Q(t) = b_{-1} b_0 = +1, \\ -\frac{\pi t}{2T_C} & \text{для } u_I(t) u_Q(t) = b_{-1} b_0 = -1. \end{cases} \quad (1.23)$$

В зависимости от значения произведения символов $b_{-1} b_0$ при $t = T_C$ значение приращения фазы будет равно либо $+90^\circ$, либо -90° .

Поэтому для сигнала (1.20) на первом интервале можно использовать следующее представление:

$$s(t) = A \cos[2\pi f_0 t + b_{-1} b_0 \frac{\pi t}{2T_C}], \quad 0 < t \leq T_C$$

Эти рассуждения можно повторить для второго интервала времени, получив аналогичное представление

$$s(t) = A \cos[2\pi f_0 t + b_{-1} b_0 \frac{\pi}{2} + b_0 b_1 \frac{\pi(t - T_C)}{2T_C}], \quad T_C < t \leq 2T_C. \quad (1.24)$$

Из (1.24) становится очевидным рекуррентное соотношение между представлениями сигнала (1.20) на последовательности временных интервалов длительностью T_C , что позволяет сразу записать общее представление этого сигнала на произвольном интервале:

$$s(t) = A \cos \left\{ 2\pi f_0 t + \frac{\pi}{2} \sum_{k=1}^{i-2} b_{k-1} b_k + b_{i-2} b_{i-1} \frac{\pi[t - (i-1)T_C]}{2T_C} \right\}, \quad (1.25)$$

$$(i-1)T_C < t \leq iT_C.$$

Это представление аналогично представлению (1.17); следовательно, сигнал (1.20) действительно является ЧМ сигналом с непрерывной фазой.

Из (1.25) следует, что ММС сигнал, сформированный квадратурным модулятором, обладает следующей особенностью: знак приращения фазы на очередном временном интервале определяется не только очередным символом, передаваемым на этом интервале, но и значением предшествующего символа. Поэтому при демодуляции такого сигнала в приемнике необходимо учитывать эту взаимосвязь между соседними канальными символами.

На рис. 1.19 представлена функциональная схема устройства, обеспечивающего формирование ММС сигнала в соответствии с его представлением (1.25). Поток информационных битов поступает на вход демультиплексора, в котором разбивается на две подпоследовательности – нечетных и четных битов. Каждая подпоследовательность преобразуется в последовательность положительных или отрицательных прямоугольных импульсов. Далее импульсы обеих подпоследовательностей расширяются до длительностей $2T_C$, перемножаются на гармонические полуволны квадратурных каналов и переносятся на высокую частоту.

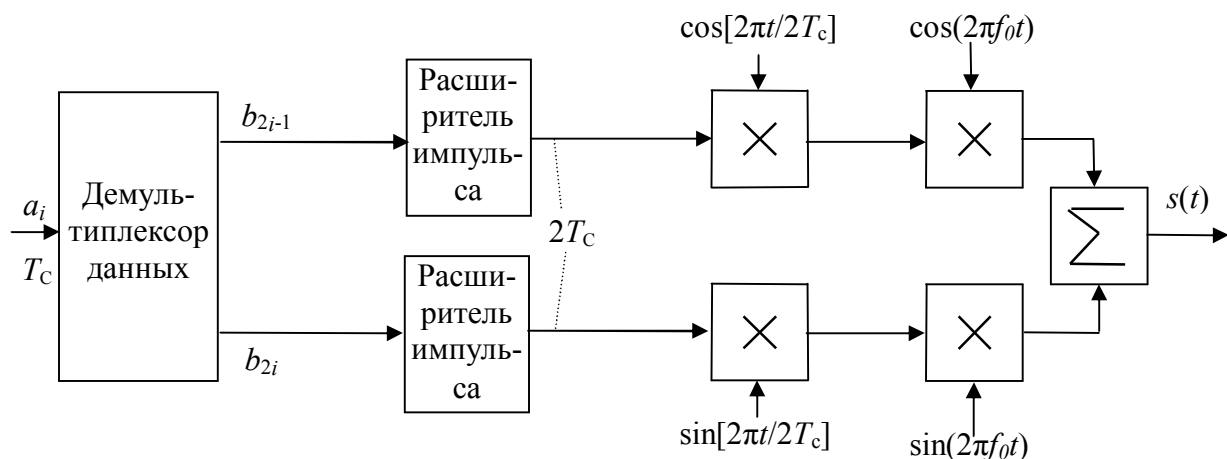


Рис. 1.19. Функциональная схема устройства формирования ММС сигнала (1.25)

На практике часто используется несколько иной способ формирования ММС сигнала; соответствующая функциональная схема представлена на рис. 1.20. Перемножение несущего гармонического колебания с $\cos[\pi t/(2T_C)]$

приводит к двум гармоническим сигналам с частотами $F_0+1/(4T_C)$ и $F_0-1/(4T_C)$ и связанными фазами. Эти сигналы разделяются узкополосными фильтрами и комбинируются так, чтобы сформировать соответственно синфазную и квадратурную компоненты несущего колебания, которые далее перемножаются с подпоследовательностями $u_I(t)$ нечётных и $u_Q(t)$ чётных символов.

Сигналы с ММС применяются в настоящее время в спутниковой связи, в военных системах связи и управления и в диапазоне очень низких частот для гидроакустической связи. Поскольку длительность импульсов в квадратурных каналах увеличена в два раза, то спектральная плотность мощности этих сигналов занимает значительно меньшую полосу частот, в результате чего эти сигналы обладают хорошей спектральной эффективностью.

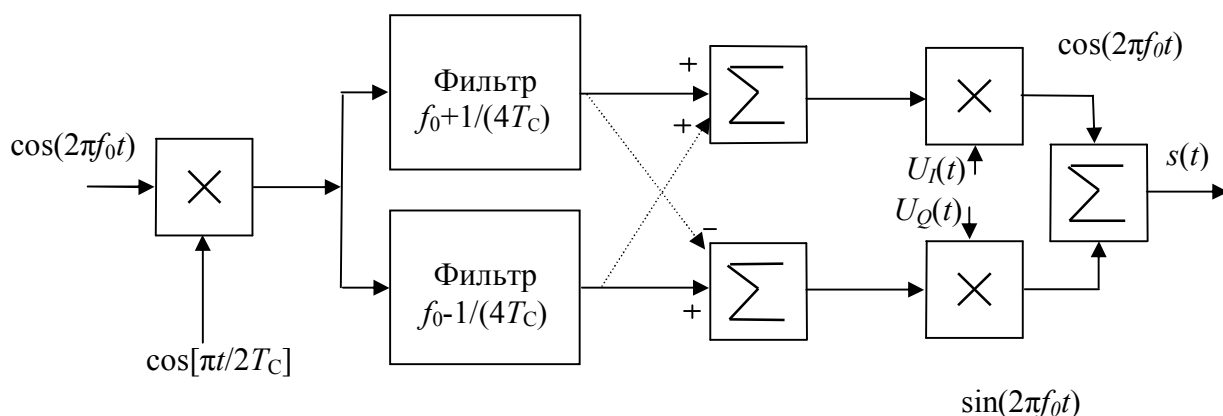


Рис. 1.20. Функциональная схема устройства формирования ММС сигнала

1.3.4. Гауссовская частотная модуляция с минимальным сдвигом.

ММС сигнал имеет постоянную огибающую и занимает меньшую полосу частот, чем сигнал с обычной частотной манипуляцией. Однако для многих приложений спектр ММС сигнала все еще остается достаточно широким. Основная причина этого состоит в том, что его фазовые траектории хотя и непрерывны, но являются ломаными линиями, так что их первая производная по времени оказывается разрывной. Сглаживание этих фазовых траекторий является одним из возможных путей дальнейшего уменьшения ширины спектра ММС сигнала. Достигается данный эффект использованием дополнительной фильтрации модулирующего сигнала до его переноса на высокую частоту.

На рис. 1.21 представлена функциональная схема устройства формирования сигнала с гауссовской модуляцией с минимальным сдвигом (ГММС сигнал).

Новым элементом в данной схеме является гауссовский фильтр низкой частоты (ГФНЧ) с импульсным откликом и амплитудно-частотной характеристикой, определяемыми следующими равенствами:

$$g(t) = B \sqrt{\frac{2\pi}{\ln 2}} \exp \left\{ -\frac{2\pi^2 B^2}{\ln 2} t^2 \right\}, \quad (1.26)$$

$$G(f) = \exp \left\{ - \left(\frac{f}{B} \right)^2 \frac{\ln 2}{2} \right\}, \quad (1.27)$$

где B – ширина полосы пропускания фильтра на уровне -3 дБ.

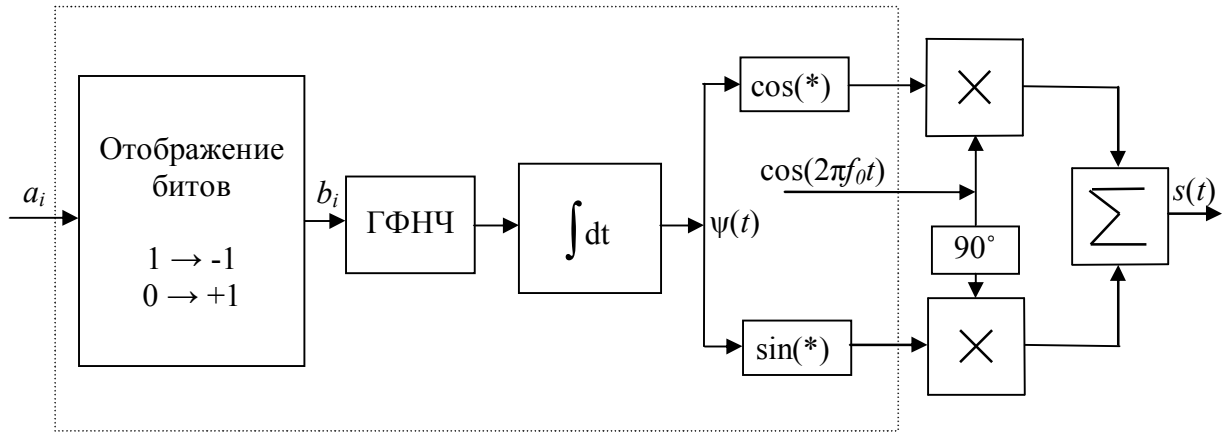


Рис. 1.21. Функциональная схема устройства формирования ГММС сигнала

На выходе ГФНЧ отклик на каждый импульс подпоследовательности определяется как свертка прямоугольного импульса на входе и импульсной характеристики фильтра:

$$r(t) = v(t) * g(t) = \int_t^{t+T_C} g\{\tau\} d\tau = \frac{1}{2} \left\{ \operatorname{erf} \left[-\frac{2}{\ln 2} \pi B t \right] + \operatorname{erf} \left[\frac{2}{\ln 2} \pi B (t + T_C) \right] \right\}. \quad (1.28)$$

Здесь

$$\operatorname{erf}(t) = \frac{2}{\pi} \int_0^t \exp \{-\tau^2\} d\tau. \quad (1.29)$$

При $BT_C = 1$ длительность импульсного отклика фильтра примерно равна длительности одного импульса на входе фильтра и сглаживание формы импульса оказывается незначительным. При уменьшении полосы пропускания B отклик на одиночный импульс приближается к гауссовской кривой и имеет более компактный спектр; в результате фазовые траектории не имеют изломов. Если полоса B неограниченно увеличивается, то форма отклика приближается к форме одиночного импульса. В пределе при $BT_C = \infty$ можно считать, что фильтр отсутствует, а схема формирует ММС сигнал.

При практическом построении устройства формирования ГММС сигнала основные операции формирования сигнала могут быть выполнены с помощью цифровых элементов, которые на рис. 1.21 обведены пунктирной лини-

ей; только преобразование частоты выполняется аналоговыми устройствами. Наиболее сложным элементом этой схемы является гауссовский фильтр низкой частоты. Обычно он реализуется как цифровой фильтр с конечной импульсной характеристикой. Значительная доля операций при этом может быть выполнена специализированным процессором.

Гауссовская ММС обеспечивает достаточно высокую спектральную эффективность системы связи и широко используется в современных системах связи с подвижными объектами. В частности, в системах сотовой связи стандарта *GSM* при $BT_C = 0,3$ и в системах цифровой бесшнуровой связи европейского стандарта *DECT* при $BT_C = 0,5$. Эта функциональная схема устройства формирования ГММС сигнала используется также в системах связи стандарта *PCS-1900*.

На рис. 1.22 более детально представлены некоторые элементы данного устройства. Пунктирной линией обведены аналоговые устройства.

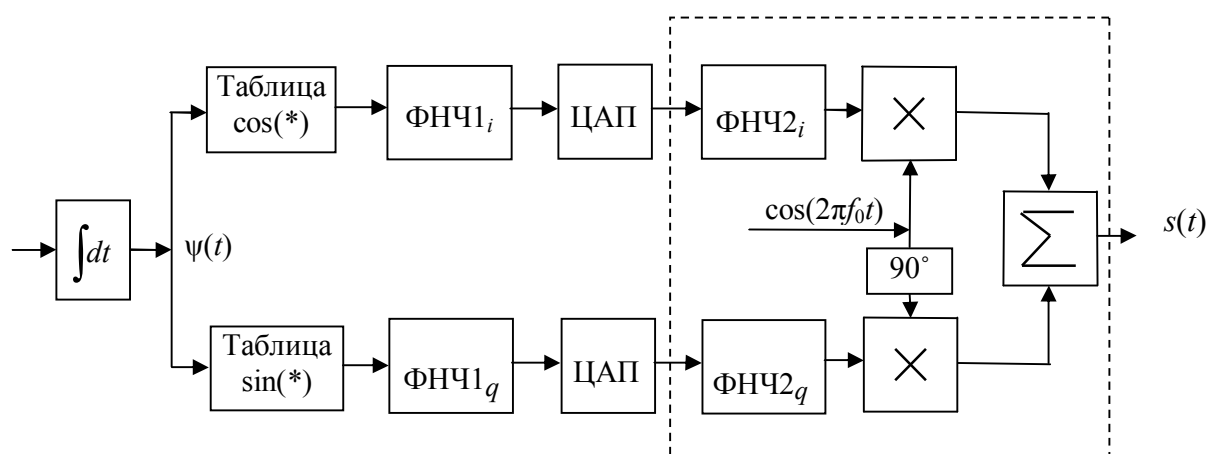


Рис. 1.22. Элементы устройства формирования ГММС сигнала

В современных системах связи предусматриваются серьезные меры для обеспечения устойчивой связи в сложных условиях распространения радиоволн (движение передатчика и приемника, препятствия на пути распространения радиоволн). К таким мерам в первую очередь следует отнести медленные скачки по частоте в течение сеанса связи и управление мощностью, излучаемой передатчиками. Способы реализации этих мер можно проиллюстрировать, если ввести незначительные изменения в схему, представленную на рис. 1.21. Эти изменения показаны на рис. 1.23.

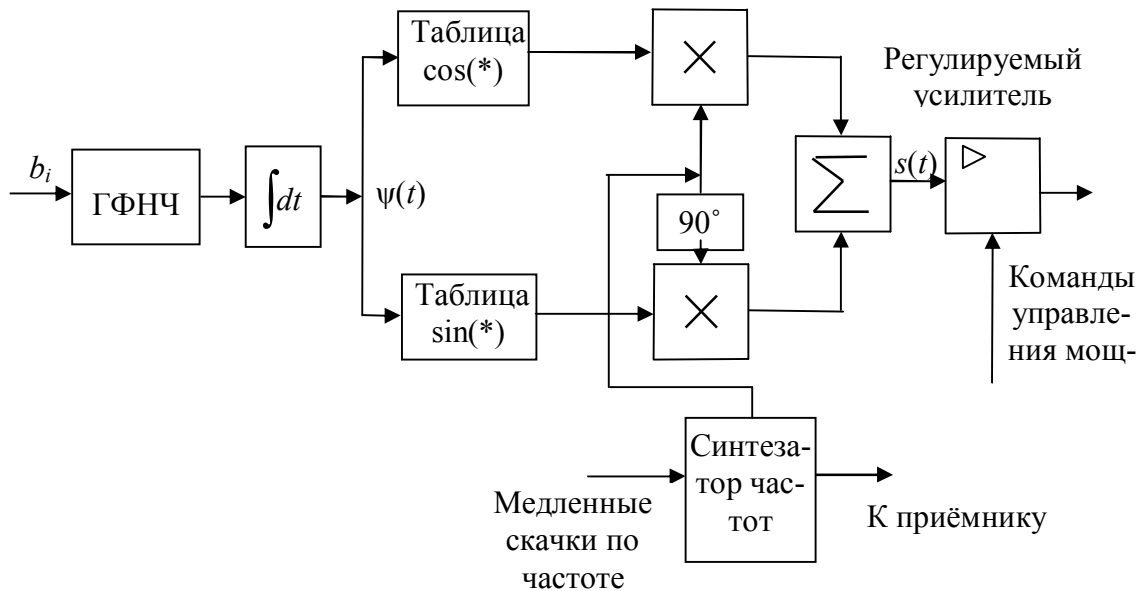


Рис. 1.23. Функциональная схема устройства формирования ГММС сигнала, включающая элементы управления излучаемой мощностью и частотой несущего колебания

Команды управления мощностью формируются специальной системой, которая контролирует уровень сигнала на входе приемника. Медленные скачки по частоте осуществляются с частотой в несколько сотен герц. Для стандарта GSM эта частота равна 217 Гц.

1.3.5. Многопозиционная частотная модуляция. Для M -ичной частотной модуляции передаваемые каналные символы определяются выражениями

$$s_m(t) = A \cos \left[\frac{\pi}{T_C} (n_0 + m)t \right], \quad (i-1)T_C < t \leq iT_C, \quad m = 1, 2, \dots, M, \quad (1.30)$$

где $f_0 = n_0 / (2T_C)$ для некоторого целого числа n_0 . m передаваемых каналных символов имеют одинаковую энергию и одинаковую длительность; частоты символов отстоят друг от друга на $1/(2T_C)$ Гц, что обеспечивает попарную ортогональность всех символов.

Один символ M -ичного ЧМ сигнала может переносить $m = \log_2 M$ информационных битов. Структурная схема соответствующего устройства представлена на рис. 1.24.

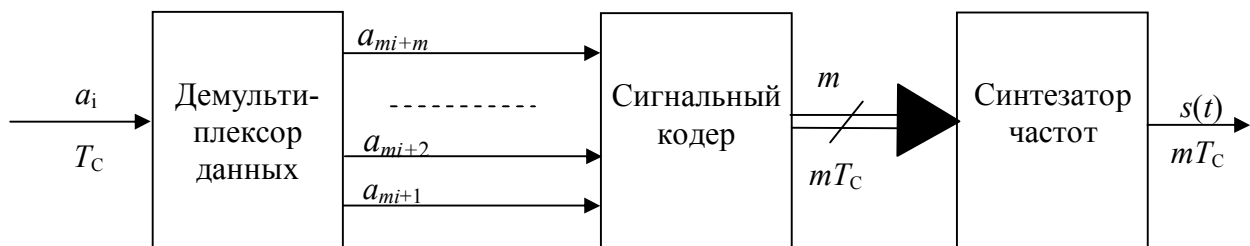


Рис. 1.24. Структурная схема устройства формирования M -ичного ЧМ сигнала

1.4. М-ичная амплитудная модуляция

При амплитудной модуляции в соответствии с символами передаваемого сообщения изменяется амплитуда передаваемого сигнала, поэтому АМ сигнал можно записать в следующем виде:

$$s(t) = A(t) \cos(2\pi f_0 t), \quad (1.31)$$

где

$$A(t) = \sum_{k=0}^{k-1} d_k U_{T_S}(t - kT_S), \quad (1.32)$$

d_k – случайные величины, принимающие значения из множества $\{a_i, i = \overline{1, M}\}$, элементы которого

$$a_i = (2i - 1 - M) \frac{d}{2}, \quad i = \overline{1, M}; \quad (1.33)$$

$M=2^m$ – количество возможных двоичных последовательностей, создаваемых источником дискретных сообщений, $U_{T_S}(t)$ – импульс прямоугольной формы длительностью T_S и единичной энергией.

На рис. 1.25 дана геометрическая иллюстрация формируемого ансамбля АМ сигналов для $M = 4$, $M = 8$, $D = 2$.

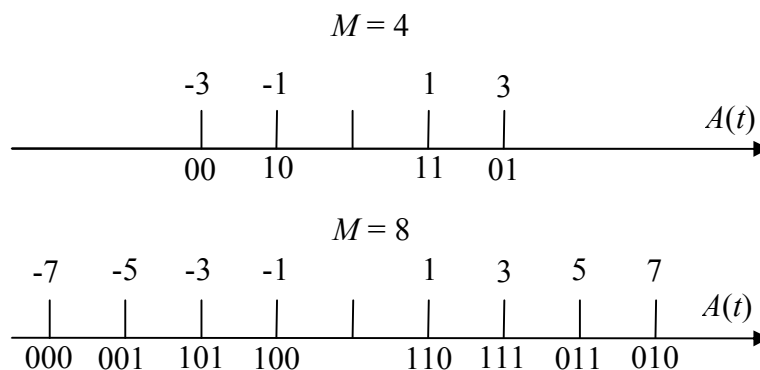


Рис. 1.25. Геометрическое представление ансамбля АМ сигналов

1.5. Комбинированные методы модуляции

1.5.1. Квадратурная амплитудная модуляция. При M -ичной фазовой модуляции, рассмотренной в подразделе 1.2, амплитуда и частота несущего колебания в течение сеанса связи остаются постоянными. Изменяется только начальная фаза каждого канального символа.

При квадратурной амплитудной модуляции (КАМ) изменяются значения амплитуды и начальной фазы каждого канального символа. Если число возможных значений этих параметров дискретно и конечно, то этот тип модуляции также является цифровым. Один канальный символ сигнала при таком способе модуляции можно представить следующим равенством:

$$s_m(t) = A_m \cos(2\pi f_0 t + \Phi_m) = \operatorname{Re}[A_m \exp\{j\Phi_m\} \exp\{2\pi f_0 t\}], \quad (i-1)T_C < t \leq iT_C, \quad (1.34)$$

в котором $\dot{A}_m = A_m \exp\{j\Phi_m\}$ является комплексной амплитудой этого канального символа, $m = 1, 2, \dots, M$. При построении сигнального созвездия этого сигнала удобнее использовать вещественную и мнимую части комплексной амплитуды:

$$s_m(t) = A_m \cos(2\pi f_0 t + \Phi_m) = A_m \cos(\Phi_m) \cos(2\pi f_0 t) + A_m \sin(\Phi_m) \times \sin(2\pi f_0 t) = I \cos(2\pi f_0 t) + Q \sin(2\pi f_0 t), \quad (i-1)T_C < t \leq iT_C, \quad (1.35)$$

где I и Q – координаты m -й точки сигнального созвездия КАМ сигнала.

На рис. 1.26 показана структурная схема модулятора и на рис. 1.27 сигнальное созвездие для случая, когда $a(t)$ и $b(t)$ принимают значения $\pm 1, \pm 3$ (4-уровневая КАМ).

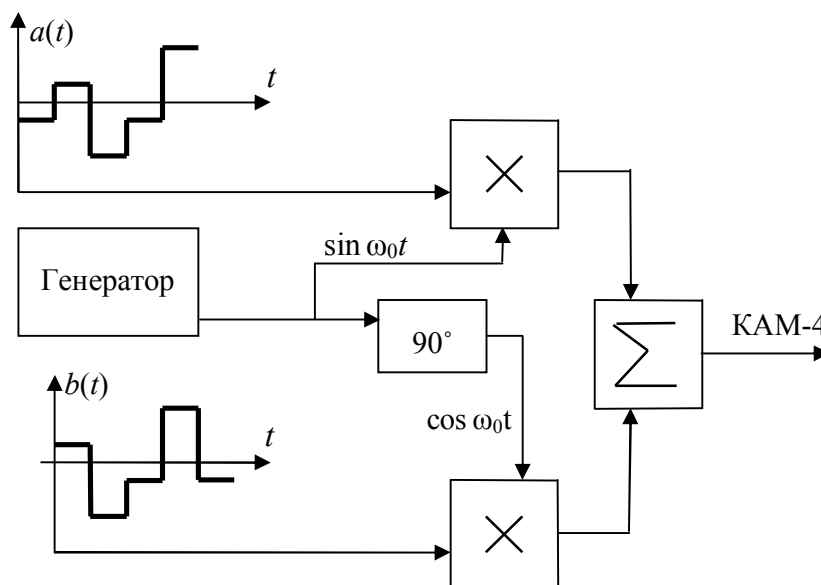


Рис. 1.26. Структурная схема модулятора КАМ-16

Величины $\pm 1, \pm 3$ определяют уровни модуляции и имеют относительный характер. Созвездие содержит 16 сигнальных точек, каждая из которых соответствует четырём передаваемым информационным битам.

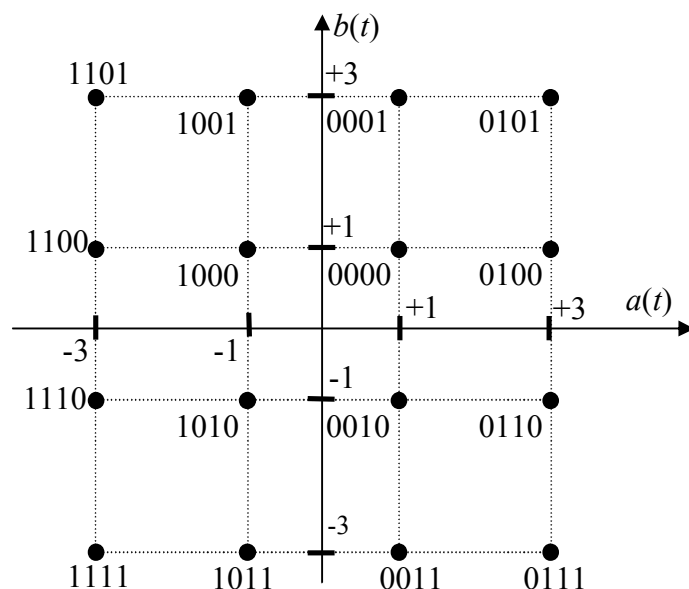


Рис. 1.27. Сигнальная диаграмма КАМ-16

Существует несколько способов практической реализации 4-уровневой КАМ, наиболее распространённым из которых является так называемый **способ модуляции наложением**. В схеме, реализующей данный способ, используются два одинаковых 4-фазных модулятора. Структурная схема такого модулятора приведена на рис. 1.28, а диаграммы, поясняющие его работу, приведены на рис. 1.29.

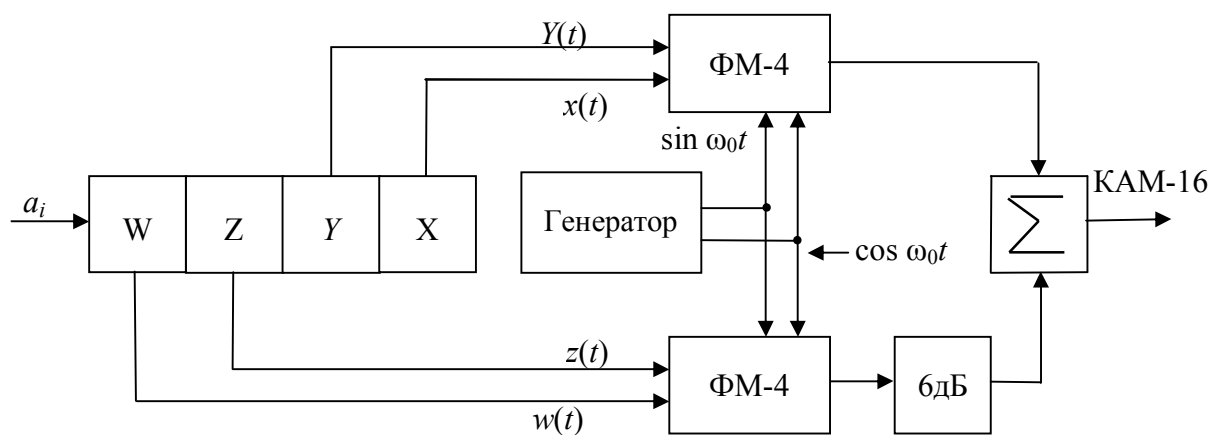


Рис. 1.28. Структурная схема модулятора КАМ-16, использующего способ модуляции наложением

Необходимо отметить, что разные каналные символы этого сигнала имеют разную энергию; расстояние между разными сигнальными точками также оказывается различным. В результате вероятность перепутывания символов в приемнике для разных символов оказывается разной.

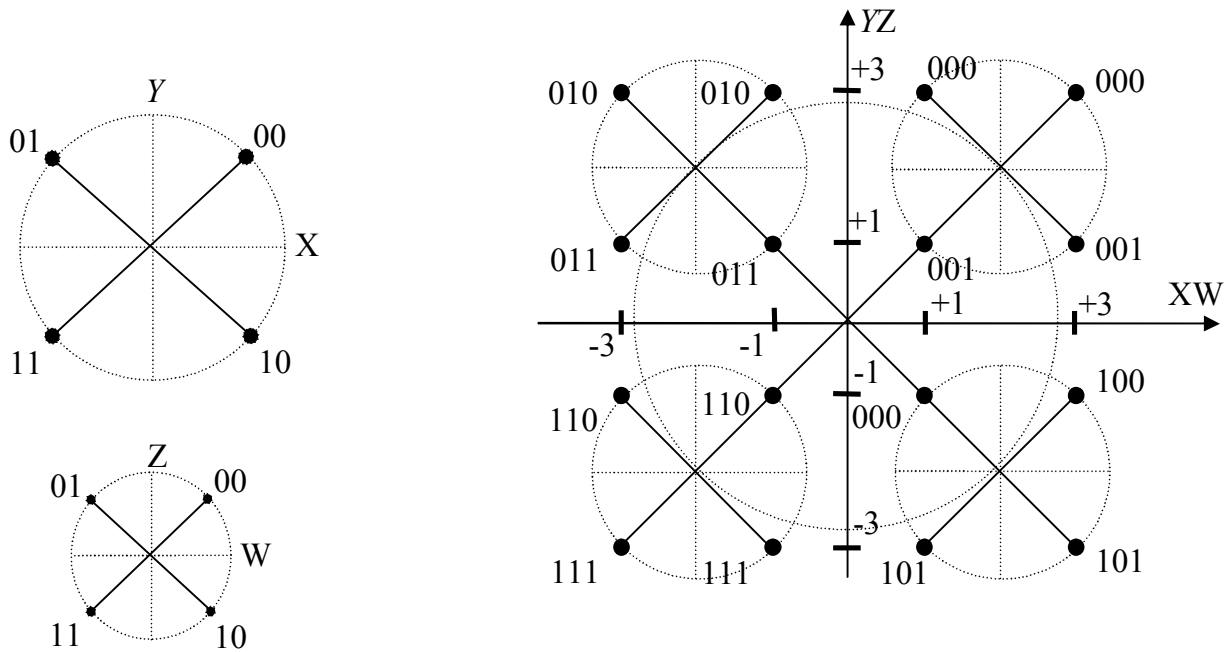


Рис. 1.29. Диаграммы модулятора КАМ-16, использующего способ модуляции наложением

Один канальный символ такого сигнала может переносить $m = L_0 G_2 M$ информационных битов. В частности, при $M = 16$ имеем $m = 4$. Поэтому если по-прежнему считать, что длительность одного бита равна T_c то длительность одного канального символа КАМ сигнала равна $T_{kc} = mT_c$. Следовательно, при формировании этого сигнала поток информационных битов должен группироваться в блоки по m битов. Каждому блоку должен быть поставлен в соответствие один канальный символ. Установление такого соответствия называется сигнальным кодированием.

На рис. 1.27 сигнальное созвездие имеет форму квадрата или квадратной решетки, в узлах которой располагаются сигнальные точки. Это не единственно возможная форма сигнального созвездия и не всегда лучшая. Сигнальные созвездия могут иметь форму, например, креста, круга, что часто оказывается необходимым при больших значениях M . В современных системах связи значения этого параметра могут превышать 1024.

При больших значениях M задавать множества возможных координат сигнальных точек проще с помощью целых чисел, нумеруя сигнальные точки от начала координат. Например, для квадратной сигнальной решетки, изображенной на рис. 1.27, можно ввести обозначения $a_{\min}$ и $b_{\min}$ для координат точек ближайших к началу координат. Тогда если все соседние точки имеют одинаковые расстояния между собой вдоль каждой оси, то координаты остальных точек можно выразить через значения координат ближайших точек с помощью соотношений:

$$a_k = \pm k a_{\min}, \quad b_l = \pm l b_{\min}, \quad (1.36)$$

где индексы k и l принимают целочисленные значения.

Например, для созвездия рис. 1.27 значения индексов принадлежат множеству $\{-3, -1, +1, +3\}$. Совокупность всех точек этого сигнального созвездия может быть задана с помощью матрицы

$$\{k, l\} = \begin{bmatrix} (-3,+3) & (-1,+3) & (+1,+3) & (3,+3) \\ (-3,+1) & (-1,+1) & (+1,+1) & (3,+1) \\ (-3,-1) & (-1,-1) & (+1,-1) & (3,-1) \\ (-3,-3) & (-1,-3) & (+1,-3) & (3,-3) \end{bmatrix}. \quad (1.37)$$

Из теории связи известно, что при равном числе точек в сигнальном созвездии спектр сигналов КАМ идентичен спектру сигналов ФМ. Однако помехоустойчивость систем ФМ и КАМ различна. При большом числе точек сигналы системы КАМ имеют лучшие характеристики, чем системы ФМ. Основная причина этого состоит в том, что расстояние между сигнальными точками в системе ФМ меньше расстояния между сигнальными точками в системе КАМ. Расстояние D между соседними точками сигнального созвездия в системе КАМ (рис. 1.30) с L уровнями модуляции определяется выражением: $d = \sqrt{2}/(L-1)$.

Аналогично для ФМ (рис. 1.31): $d = 2 \sin(\pi/M)$, где M – число фаз.

Из приведённых выражений следует, что при увеличении значения M и одном и том же уровне мощности системы КАМ предпочтительнее систем ФМ. Например, при $M=16$ ($L=4$) $D_{\text{КАМ}}=0,471$ и $D_{\text{ФМ}}=0,390$, а при $M=32$ ($L=6$) $D_{\text{КАМ}}=0,283$, $D_{\text{ФМ}}=0,196$.

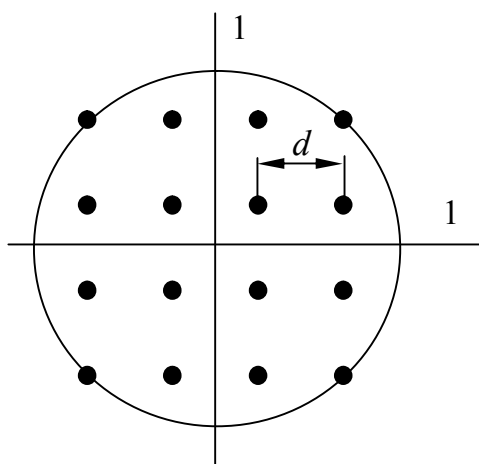


Рис. 1.30. Сигнальное созвездие КАМ-16

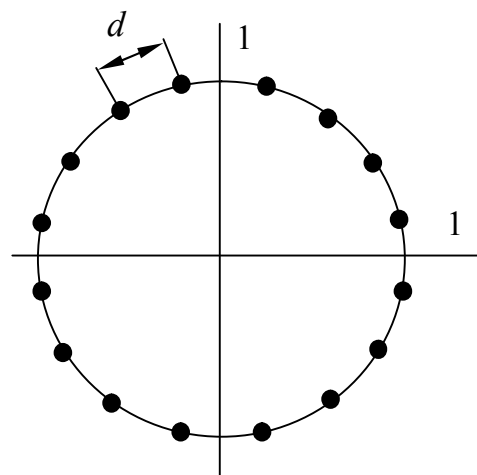


Рис. 1.31. Сигнальное созвездие ФМ-16

В качестве примера применения квадратурной амплитудной модуляции рассмотрим протокол *V.29*, который предусматривает возможность работы со скоростями 9600, 7200 и 4800 бит/с по четырехпроводным арендованным телефонным каналам. Частота несущего сигнала равна 1700 Гц, а скорость модуляции – 2400 Бод. Применена квадратурная амплитудная модуляция. Сиг-

нальное созвездие протокола изображено на рис. 1.32. При скорости 9600 бит/с поток двоичных символов разделяется на блоки по 4 бита ($Q1$, $Q2$, $Q3$, $Q4$). Второй ($Q2$), третий ($Q3$), четвертый ($Q4$) биты в блоке определяют изменение фазы сигнала по отношению к фазе предшествующего элемента в соответствии с табл. 1.3, реализуя таким образом закон относительного кодирования.

Таблица 1.3

Закон изменения фазы сигнала для протокола V.29

$Q2$	$Q3$	$Q4$	Изменение фазы
0	0	1	0^0
0	0	0	45^0
0	1	0	90^0
0	1	1	135^0
1	1	1	180^0
1	1	0	225^0
1	0	0	270^0
1	0	1	315^0

Амплитуда передаваемого сигнального элемента определяется первым битом ($Q1$) и величиной абсолютной фазы сигнального элемента в соответствии с табл. 1.4.

Таблица 1.4

Закон изменения амплитуды сигнала для протокола V.29

Абсолютная фаза	$Q1$	Амплитуда
$0^0, 90^0, 180^0, 270^0$	0	3
	1	5
$45^0, 135^0, 225^0, 315^0$	0	$3\sqrt{2}$
	1	$3\sqrt{2}$

Принцип формирования сигнала рассмотрим на примере передачи последовательности 1011 0000 1101 0110 0011 0101. Значения амплитуд и фаз сигналов в соответствии с диаграммой на рис. 1.32, представлены в табл. 1.5 (за исходную фазу предыдущей посылки по диаграмме выбрана фаза 135^0). Значения фазы 270^0 и амплитуды 5 первой строки табл. 1.5 получились следующим образом. Для блока $Q2Q3Q4 = 011$ по табл. 1.3 находится фаза, равная 135^0 . Это значение определяет скачок фазы по отношению к фазе предыдущей посылки, равной также 135^0 . В результате абсолютному значению фазы 270^0 и значению бита $Q1 = 1$ по табл. 1.4 соответствует амплитуда, равная 5.

Пример кодирования фазы и амплитуды при передаче последовательности данных по протоколу V.29

Блок данных	Абсолютная фаза	Амплитуда
1011	270°	5
0000	315°	$3\sqrt{2}$
1101	270°	5
0110	135°	$3\sqrt{2}$
0011	270°	3
1101	225°	$3\sqrt{2}$

При скорости 9600 бит/с в соответствии с диаграммой на рис. 1.32 применяется сигнал КАМ-16 с четырьмя градациями амплитуды и восемью градациями фазы. На скорости 7200 бит/с при объединении в блок трех бит достаточно 8-и позиционного сигнала. Согласно V.29 в этом случае используются 2 значения амплитуды (3 и $3\sqrt{2}$) и все 8 градаций фаз. Трехбитный блок образуют 3 последние цифры комбинаций, указанных на рис. 1.32.

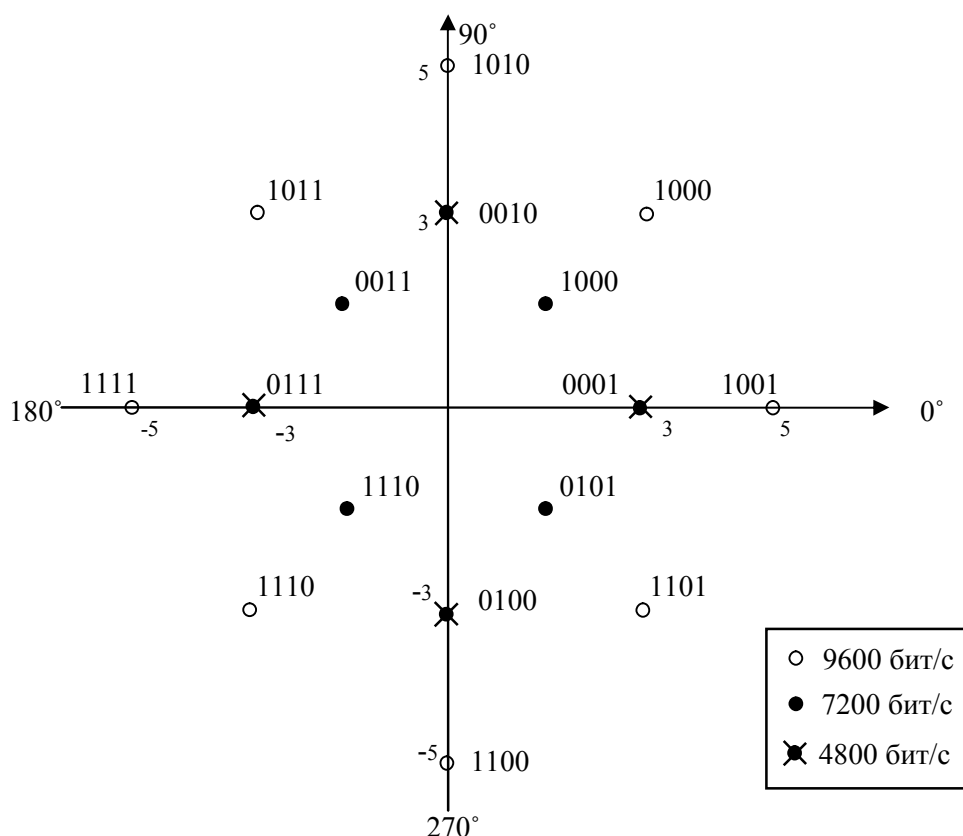


Рис. 1.32. Сигнальная диаграмма протокола V.29

На скорости 4800 бит/с при объединении двух бит для передачи требуется четыре элементарных сигнала. В данном случае эти сигналы имеют вид обычной ФМ-4. Двухбитный блок образуют два средних двоичных знака комбинаций, указанных на рис. 1.32.

1.5.2. CAP-модуляция. Амплитудно-фазовая модуляция с подавлением несущей (CAP – *CaRRieR LeSS AmpLITUDe moDULation/PhaSe moDULation*) является одним из широко используемых в настоящее время на DSL линиях способов модуляции. CAP-модуляция представляет собой одну из разновидностей КАМ, её особенность заключается в специальной обработке модулированного информационного сигнала перед отправкой его в линию. В процессе этой обработки из спектра модулированного сигнала исключается составляющая, которая соответствует частоте несущего колебания КАМ. После того, как приёмник принимает переданный информационный сигнал, он сначала восстанавливает частоту несущего колебания, а уже после этого восстанавливает информационный сигнал. Такие манипуляции со спектром выполняются для того, чтобы уменьшить долю неинформативной составляющей в спектре передаваемого информационного сигнала. Это в свою очередь делается для обеспечения большей энергетики сигнала и уменьшения уровня перекрёстных помех у сигналов, которые передаются одновременно.

Основные принципы формирования выходного CAP-модулированного сигнала соответствует принципам формирования КАМ-сигнала. Отличия указанных методов заключаются в наличии дополнительных процедур, необходимых для формирования и восстановления спектра CAP-модулированного сигнала.

Одна из возможных функциональных схем формирования сигнала, модулированного в соответствии с принципами метода CAP, представлена на рис. 1.33.

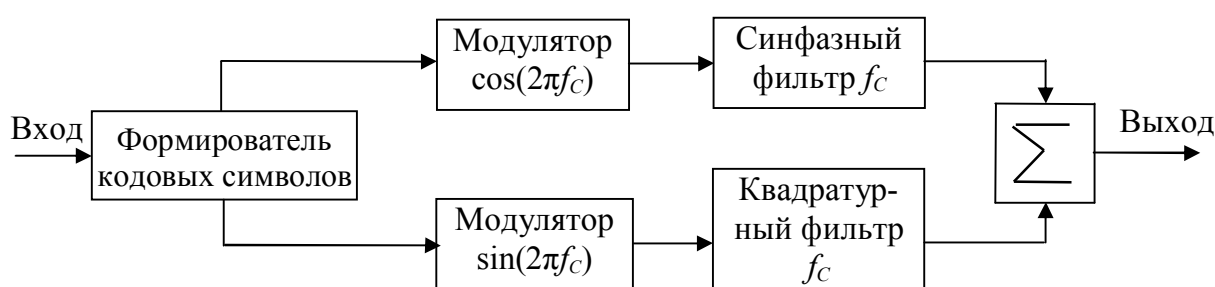


Рис. 1.33. Схема формирования CAP-модулированного сигнала

Для подавления гармоник несущего колебания используется синфазный и квадратурный фильтры. Для адекватного восстановления сформированного таким образом сигнала на приёмной стороне должны быть выполнены операции по восстановлению несущего колебания. После восстановления несущей приемник выполняет те же операции, что и приемник КАМ. Поэтому, по

крайней мере, теоретически приемник CAP может взаимодействовать с передатчиком КАМ. CAP-модуляция может быть использована для формирования сигналов в различных технологиях *xDSL*.

Требуемое соотношение сигнал/шум находится в следующей зависимости от конкретного используемого способа CAP:

Способ CAP	Требуемое отношение сигнал/шум, дБ
CAP-4	14,5
CAP-8	18,0
CAP-16	21,5
CAP-32	24,5
CAP-64	27,7
CAP-128	30,6
CAP-256	33,8

В соответствии с определением значение относительного соотношения сигнал/шум соответствует уровню помехи, при которой вероятность искажения бита на приемной стороне не будет превышать значения 10^{-7} . Как и в случае КАМ, помехоустойчивость метода модуляции уменьшается при повышении его спектральной эффективности.

Поскольку амплитудно-фазовая модуляция с подавлением несущей являются, по сути, квадратурно-амплитудной модуляцией, ей свойственны все положительные качества, которые присущи этому классу способов – относительная простота реализации и высокая спектральная эффективность. Несомненным достоинством собственно CAP-модуляции является высокая энергетическая эффективность формируемого сигнала. Именно этот способ модуляции теоретически способен обеспечить максимальные значения соотношения сигнал/шум. Все эти полезные качества CAP-модуляции позволяют применять ее для построения эффективных и экономичных приёмо-передающих устройств широкого спектра технологий *DSL*.

1.5.3. DMT-модуляция. Многочастотный способ модуляции (*DMT – DiSCReTe MULti Tone*) не является принципиально новым. Формирование сигналов согласно *DMT* принципиально иное, чем у представленных выше. В отличие от КАМ-модуляции, в данном случае используется не одна, группа частот несущих колебаний. Применительно к конкретике *xDSL*-технологий весь расчётный частотный диапазон делится на несколько участков шириной по 4,3125 кГц. Каждый из этих участков используется для организации независимого канала передачи данных. На этапе вхождения в связь, а именно на этапе проверки качества линии, передатчик, исходя из уровня помех в частотном диапазоне участка, для каждого из отдельных каналов выбирает подходящую схему модуляции. На «чистых» частотных участках с малым уровнем шумов могут быть использованы методы с большими значениями спектральной плотности, например, КАМ-64. Очевидно, что использование такого

принципа регулирования скорости передачи данных позволяет наиболее точно согласовывать параметры модулированного сигнала с параметрами линии, по которой он будет передаваться. При передаче данных информация распределяется между независимыми каналами пропорционально их пропускной способности, приёмник выполняет операцию демультимплексирования и восстанавливает исходный информационный поток. Рис. 1.34 и 1.35 иллюстрируют описанный процесс адаптации.

На рис. 1.34 сплошной линией обозначена неадаптированная частотная характеристика передаваемого DMT-сигнала. Пунктиром выделена кривая зависимости затухания в линии от частоты передаваемого сигнала. Штрихпунктирной линией обозначена частотная помеха, которая постоянно действует на сравнительно небольшом участке в пределах рабочего диапазона частот передатчика. После выполнения операций согласования пропускной способности элементарных каналов с приведенными частотными характеристиками линии, зависимость скоростей передачи данных от частотного номера элементарного канала приведена на рис. 1.35.

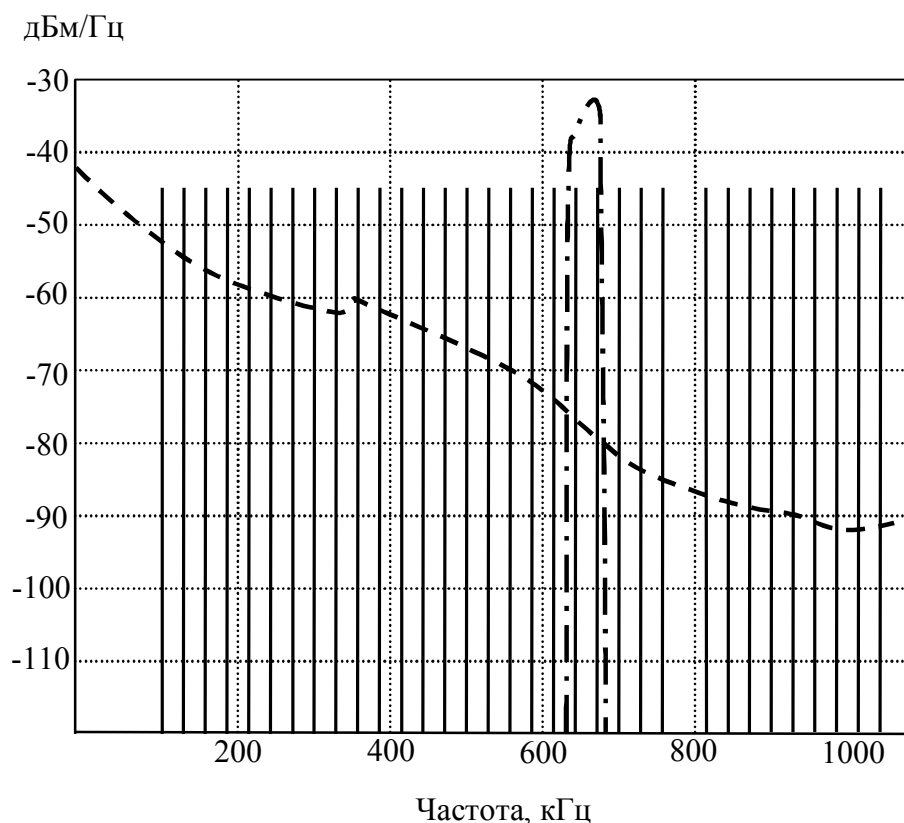


Рис. 1.34. Пример условий адаптации DMT-передатчика

Способ DMT-модуляции это дальнейшее развитие идеи, составляющей основу КАМ. Этот способ может обеспечить высокую скорость и надежность передачи данных. К дополнительным достоинствам также относятся возможность оперативной и точной адаптации приемопередающих устройств к характеристикам конкретной линии. Не последним положительным фактором

выступает практически повсеместное признание этого алгоритма стандартизирующими организациями.

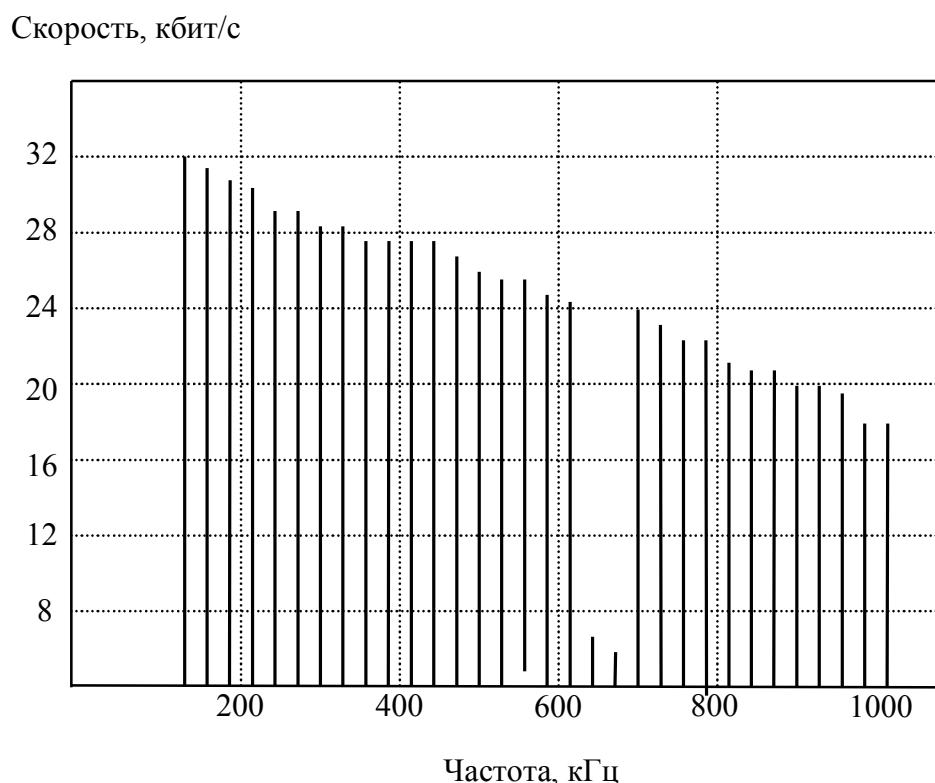


Рис. 1.35. Результат адаптации при использовании *DMT*-модуляции

Недостатками *DMT*-модуляции можно считать его громоздкость и недостаточную технологичность.

1.5.4. OFDM-модуляция. Способ *OFDM* (*ORthoGonaL FReQUencY DiVision MULtiLexinG*) является упрощенным вариантом *DMT*. В отличие от *DMT*, *OFDM*-модуляция использует единое значение спектральной эффективности для всех частотных каналов. Основные принципы формирования *OFDM*-модулированного сигнала соответствуют принципам формирования *DMT*-сигнала. Единственное, но существенное, отличие этих двух методов заключается в способе управления пропускной способностью отдельных элементарных каналов. Значения пропускной способности элементарных частотных каналов, которые формируются согласно *DMT*, могут отличаться в различных частотных диапазонах. Способ *OFDM* использует одно значение пропускной способности и скорости передачи данных для элементарных каналов всего частотного диапазона.

К достоинствам многочастотных способов модуляции относятся обеспечение высоких скоростей передачи данных и способность нивелировать воздействие на сигнал помех, которые возникают в линии. Несомненным достоинством также является наличие для них стандартов *ITU*. По сложности реализации способ *OFDM* занимает промежуточную позицию между *CAP* и *DMT*, что делает возможным его применение в малосерийных приложениях.

Недостатком способа *OFDM* можно считать невозможность избирательной адаптации пропускной способности элементарных каналов к частотным характеристикам всего частотного канала. Элементарные частотные каналы *OFDM* должны быть разделены технологическими заградительными интервалами. Чем больше будет число элементарных частотных каналов, тем шире будет совокупная длина частотного интервала, который не может быть использован непосредственно для передачи данных. Следствием этого является меньшая эффективность использования полосы пропускания линии.

1.6. Треллис-модуляция

Применение многопозиционной *КАМ* в чистом виде сопряжено с проблемой недостаточной помехоустойчивостью. Поэтому во всех современных высокоскоростных протоколах *КАМ* используется совместно с решетчатым кодированием – специальным видом сверточного кодирования. В результате появился новый способ модуляции, называемый **треллис-модуляцией** (*TCM – TReLLiS CoDeD MoDULation*). Выбранная определённым образом комбинация конкретной *КАМ* и помехоустойчивого кода в отечественной технической литературе носит название **сигнально-кодовой конструкции** (*СКК*), которая позволяет повысить помехозащищённость передачи информации наряду со снижением требований к отношению сигнал/шум в канале на 3...6 дБ. При этом число сигнальных точек увеличивается вдвое за счёт добавления к информационным битам одного избыточного, образованного путём свёрточного кодирования. Расширенный таким образом блок битов подвергается всё той же *КАМ*. В процессе демодуляции принятый сигнал декодируется по алгоритму Витерби. Именно этот алгоритм за счёт использования введённой избыточности и знания предыстории процесса приёма позволяет по критерию максимального правдоподобия выбрать из сигнального пространства наиболее достоверную эталонную точку.

Выбор способов модуляции и кодирования сводится к поиску такого заполнения сигнального пространства, при котором обеспечивается высокая скорость и высокая помехоустойчивость. Комбинирование различных ансамблей многопозиционных сигналов и помехоустойчивых кодов порождает множество вариантов сигнальных конструкций. Согласованные определённым образом варианты, обеспечивающие улучшение энергетической и частотной эффективности, и являются сигнально-кодовыми конструкциями. Все применяемые сегодня СКК используют свёрточное кодирование.

Типичный кодер, применяемый совместно с модулятором ФМ-8, представлен на рис. 1.36. Он является сверточным кодером с относительной скоростью кода, равной $2/3$. Каждым двум информационным битам на входе кодера сопоставляет трехсимвольные двоичные блоки на своем выходе, которые и поступают на модулятор ФМ-8.

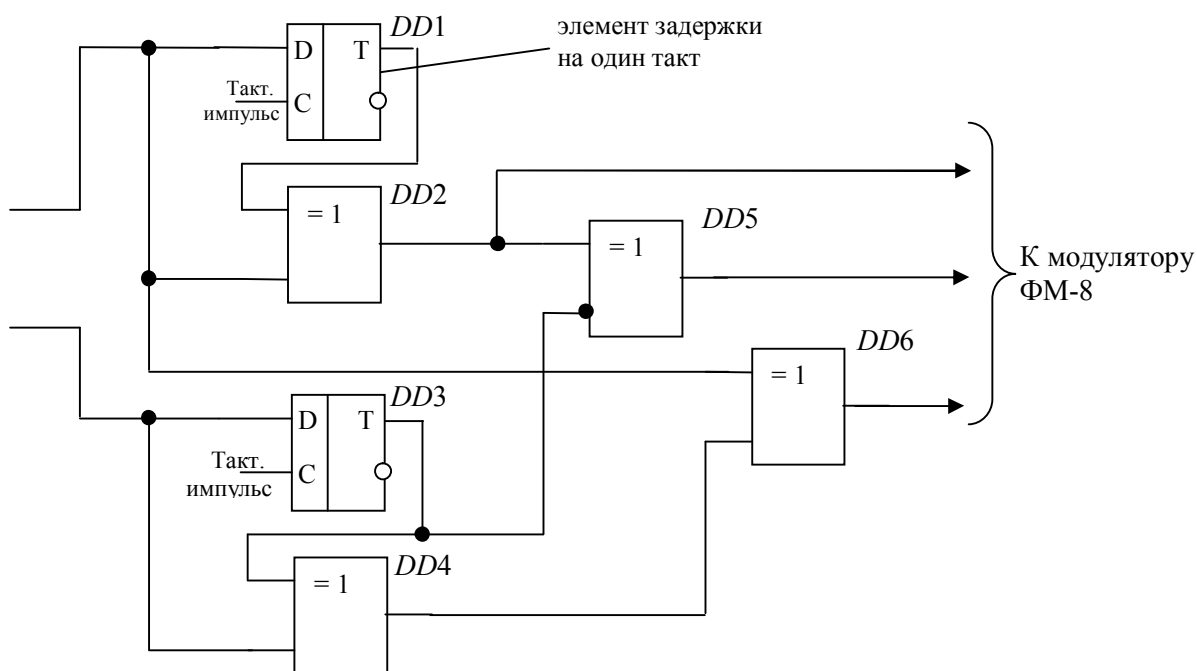


Рис. 1.36. Схема сверточного 2/3 кодера

При осуществлении треллис-кодирования в модемах серии *V.32* поток данных вначале разбивается на группы, состоящие из четырех последовательно идущих битов. Затем первые два бита каждой группы, Q_{1n} и Q_{2n} , подвергаются дифференциальному кодированию и преобразуются в биты Y_{Ln} и Y_{2n} (рис. 1.37).

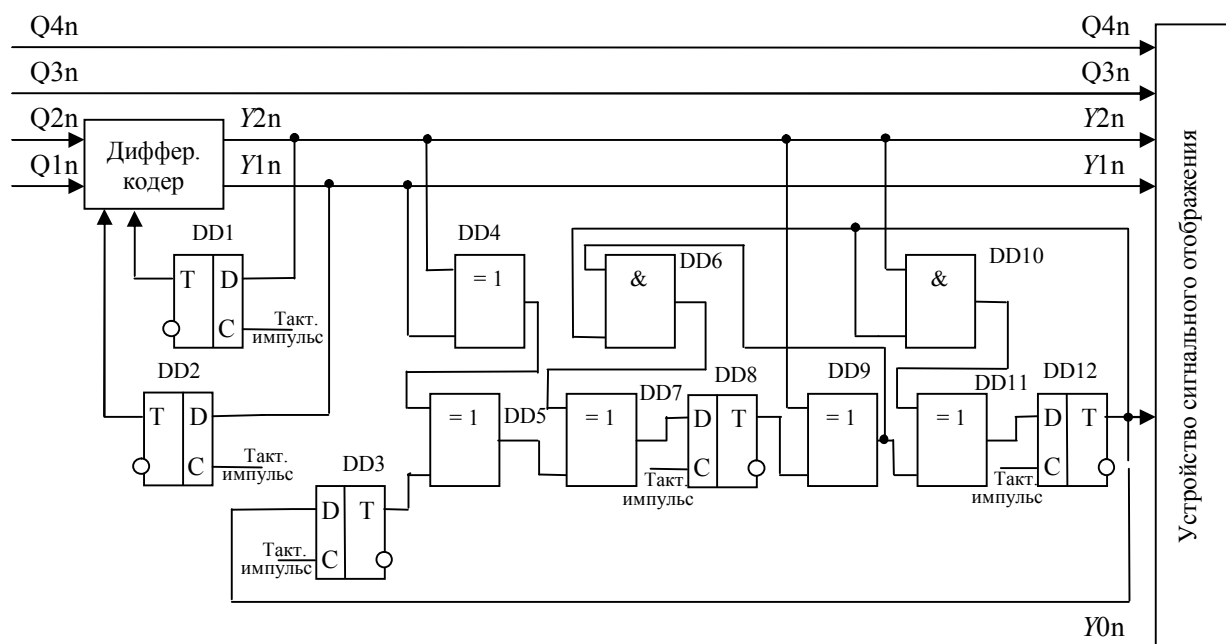


Рис. 1.37. Схема кодирования в модемах *V.32*

В табл. 1.6 приведены наборы битов, которые применяются при дифференциальном кодировании, используемом вместе с альтернативным треллис-кодированием в модемах серии V.32, которые поддерживают скорость 9600 бит/с.

Таблица 1.6

Наборы битов, применяемые при дифференциальном кодировании
и альтернативном треллис-кодировании

Биты на входе		Биты предыдущей группы на выходе		Биты на выходе	
Q_{1n}	Q_{2n}	Y_{1n-1}	Y_{2n-1}	Y_{1n}	Y_{2n}
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

Далее биты Y_{Ln} и Y_{2n} подаются на вход блока, выполняющего свёрточное кодирование и генерирующего избыточный бит Y_{0n} . Затем этот бит и четыре информационных бита – Y_{Ln} , Y_{2n} , Q_{3n} и Q_{4n} – преобразовываются в передаваемый сигнал, состояние которого (фаза-амплитуда) соответствует одному из двух наборов координат, приведенных в табл. 1.7.

Два набора состояний сигналов, которые используются в модемах серии V.32

Такие наборы координат применяются для представления состояния сигналов в модемах серии V.32, работающих со скоростью 9600 бит/с, причем один из наборов применяется при безызбыточном кодировании, а второй –

при треллис-кодировании. Если каждое из возможных состояний сигналов (фаза-амплитуда), используемых при треллис-кодировании представить в виде точки в двухмерной системе координат, получится 32-точечное созвездие, показанное на рис. 1.38.

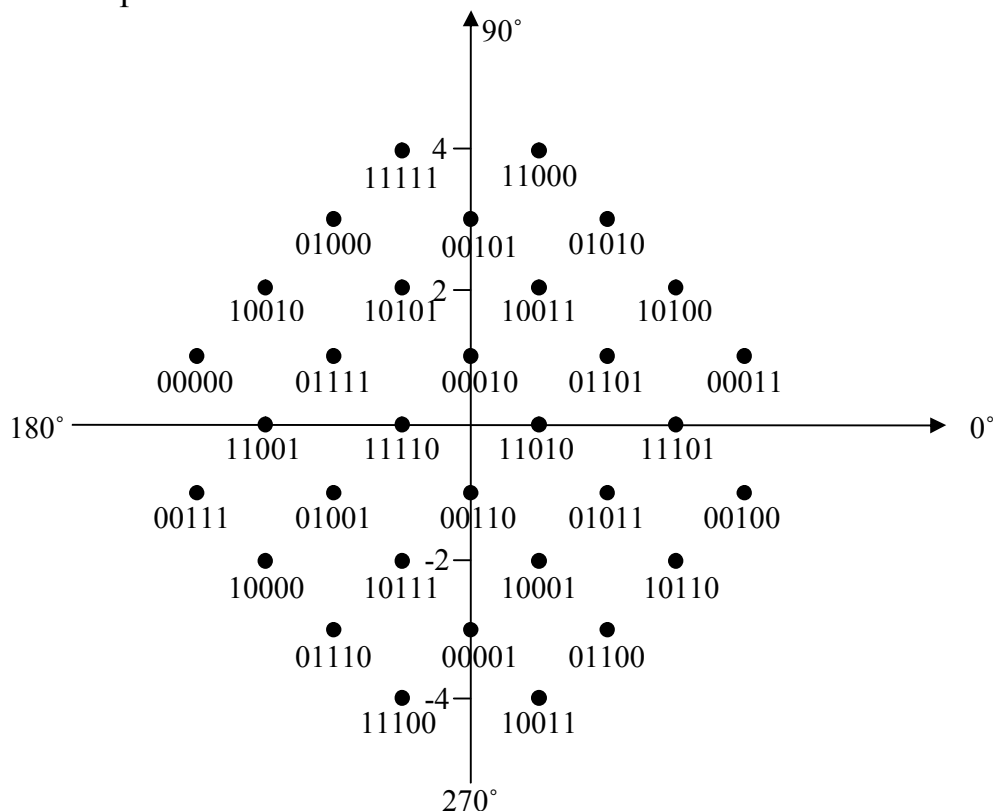


Рис. 1.38. Представленные в виде 32-точечного сигнального созвездия состояния сигналов при использовании треллис-модуляции

С помощью дополнительного бита, генерируемого в процессе треллис-кодирования, принимающий модем может обнаружить сигналы, состояния которых не соответствуют сигнальному созвездию, и исправить большинство возникающих при передаче ошибок. А это означает, что в случае выполнения треллис-кодирования количество неисправленных ошибок на порядок меньше, чем при выборе другого метода кодирования. Поэтому почти во всех модемах серии V.32, поддерживающих как треллис-кодирование, так и безызбыточное кодирование, в основном используется первый метод.

1.7. Принцип работы треллис-кодера

Рассмотрим принципы треллис-кодирования на основе простейшего кодера, состоящего из двух запоминающих ячеек и сумматоров по модулю два (рис. 1.39).

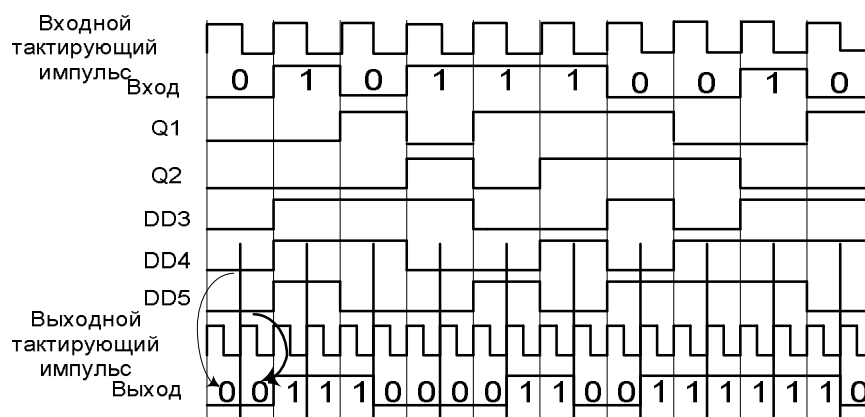
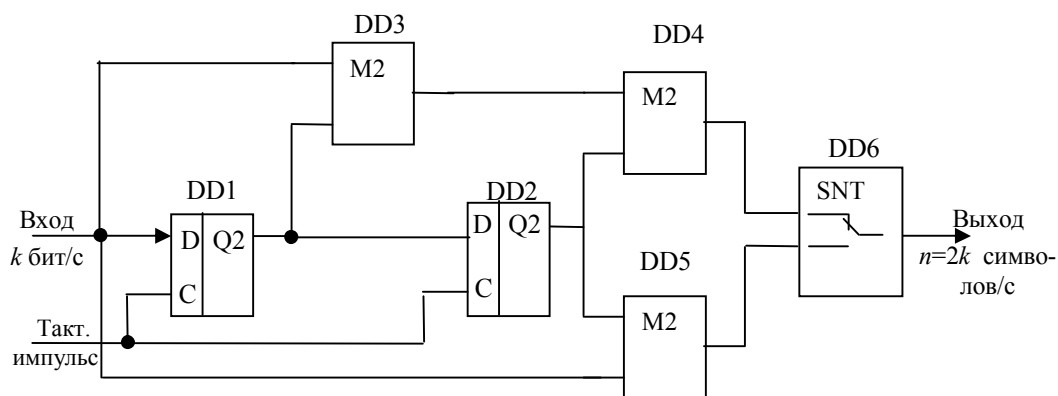


Рис. 1.39. Схема и принцип работы простейшего кодера Треллиса

Пусть на вход такого кодера поступает со скоростью k бит/с последовательность бит 0101110010. Если на выходе кодера установить считывающую головку, работающую с вдвое большей частотой, чем скорость поступления бит на вход кодера, то скорость выходного потока будет в два раза выше скорости входного потока. При этом считывающая ячейка за первую половину такта работы кодера считывает данные сначала с логического элемента $DD4$, а вторую половину такта – с логического элемента $DD5$. В результате каждому входному биту ставится в соответствие два выходных бита, то есть дибит, первый бит которого формируется элементом $DD4$, а второй – элементом $DD5$. По временной диаграмме состояния кодера нетрудно проследить, что при входной последовательности бит 0101110010 выходная последовательность будет 00 11 10 00 01 10 01 11 11 10. Отметим одну важную особенность принципа формирования дибитов. Значение каждого формируемого дибита зависит не только от входящего информационного бита, но и от двух предыдущих бит, значения которых хранятся в двух запоминающих ячейках. Действительно, если принято, что A_i – входящий бит, то значение элемента $DD5$ определится выражением $A_i \oplus A_{i-1} \oplus A_{i-2}$, а значение элемента $DD4$ – выражением $A_i \oplus A_{i-2}$. Таким образом, дибит формируется из пары битов, значение первого из которых равно $A_i \oplus A_{i-1} \oplus A_{i-2}$, а второго – $A_i \oplus A_{i-2}$. Следова-

тельно, значение дибита зависит от трех состояний: значения входного бита, значения первой запоминающей ячейки и значения второй запоминающей ячейки. Такие кодеры получили название сверточных кодеров на три состояния ($K = 3$) с выходной скоростью $1/2$.

Работу кодера удобно рассматривать на основе не временных диаграмм, а так называемой диаграммы состояния. Состояние кодера будем указывать с помощью двух значений – значения первой и второй запоминающих ячеек. К примеру, если первая ячейка хранит значение 1 ($Q1 = 1$), а вторая – 0 ($Q2 = 0$), то состояние кодера описывается значением 10. Всего возможно четыре различных состояния кодера: 00, 01, 10 и 11.

Пусть в некоторый момент времени состояние кодера равно 00. Нас интересует, каким станет состояние кодера в следующий момент времени и какой дибит будет при этом сформирован. Возможны два исхода в зависимости от того, какой бит поступит на вход кодера. Если на вход кодера поступит 0, то следующее состояние кодера также будет 00, если же поступит 1, то следующее состояние (т. е. после сдвига) будет 10. Значение формируемых при этом дибитов рассчитывается по формулам $A_i \oplus A_{i-1} \oplus A_{i-2}$ и $A_i \oplus A_{i-2}$. Если на вход кодера поступает 0, то будет сформирован дибит 00 ($0 \oplus 0 \oplus 0 = 0, 0 \oplus 0 = 0$), если же на вход поступает 1, то формируется дибит 11 ($1 \oplus 0 \oplus 0 = 1, 1 \oplus 0 = 1$). Приведенные рассуждения удобно представить наглядно с помощью диаграммы состояний (рис. 1.40), где в кружках обозначаются состояния кодера, а входящий бит и формируемый дибит пишутся через косую черту. Например, если входящий бит 1, а формируемый дибит 11, то записываем: 1/11.

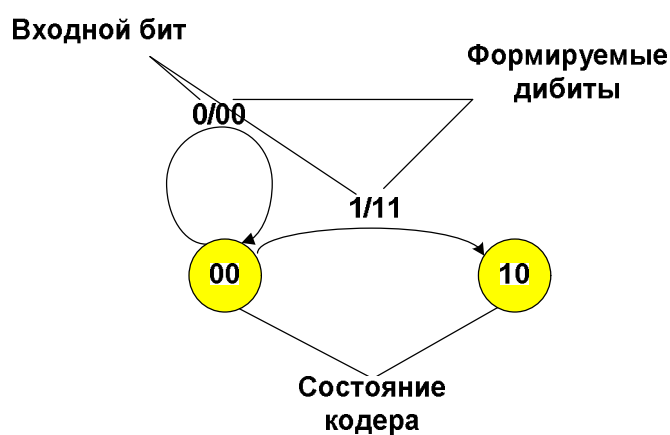


Рис. 1.40. Диаграмма возможных переходов кодера из начального состояния 00

Продолжая аналогичные рассуждения для всех остальных возможных состояний кодера, легко построить полную диаграмму состояний, на основе которой легко вычисляется значение формируемого кодером дибита.

Используя диаграмму состояний кодера, несложно построить временную диаграмму переходов для уже рассмотренной нами входной последовательности бит 0101110010. Для этого строится таблица, в столбцах которой отмечаются возможные состояния кодера, а в строках — моменты времени. Возможные переходы между различными состояниями кодера отображаются стрелками (на основе полной диаграммы состояний кодера — рис. 1.41), над которыми обозначаются входной бит, соответствующий данному переходу, и соответствующий дибит. Например, для двух первых моментов времени диаграмма состояния кодера выглядит так, как показано на рис. 1.42.

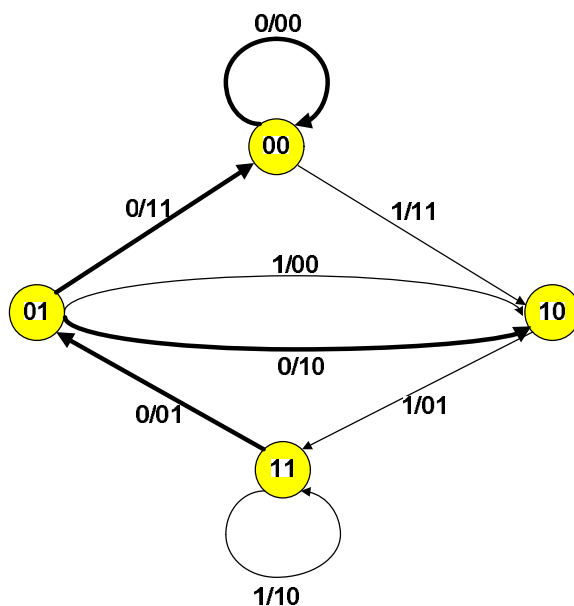


Рис. 1.41. Полная диаграмма состояния кодера

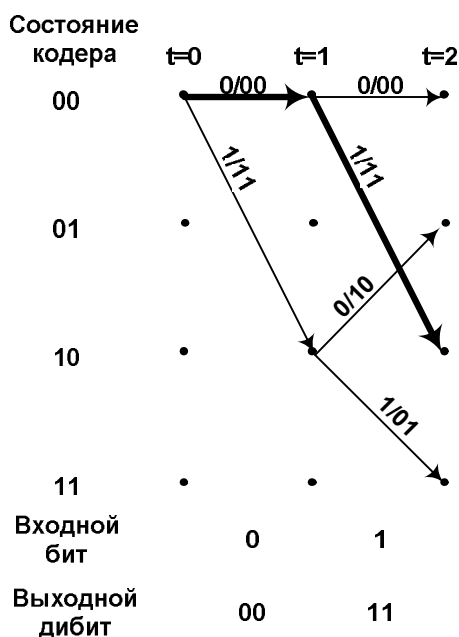


Рис. 1.42. Временная диаграмма состояния кодера для двух первых моментов времени

Продолжая отображать возможные и реальные переходы между различными состояниями кодера, соответствующие различным моментам времени, получим полную временную диаграмму состояний кодера (рис. 1.43).

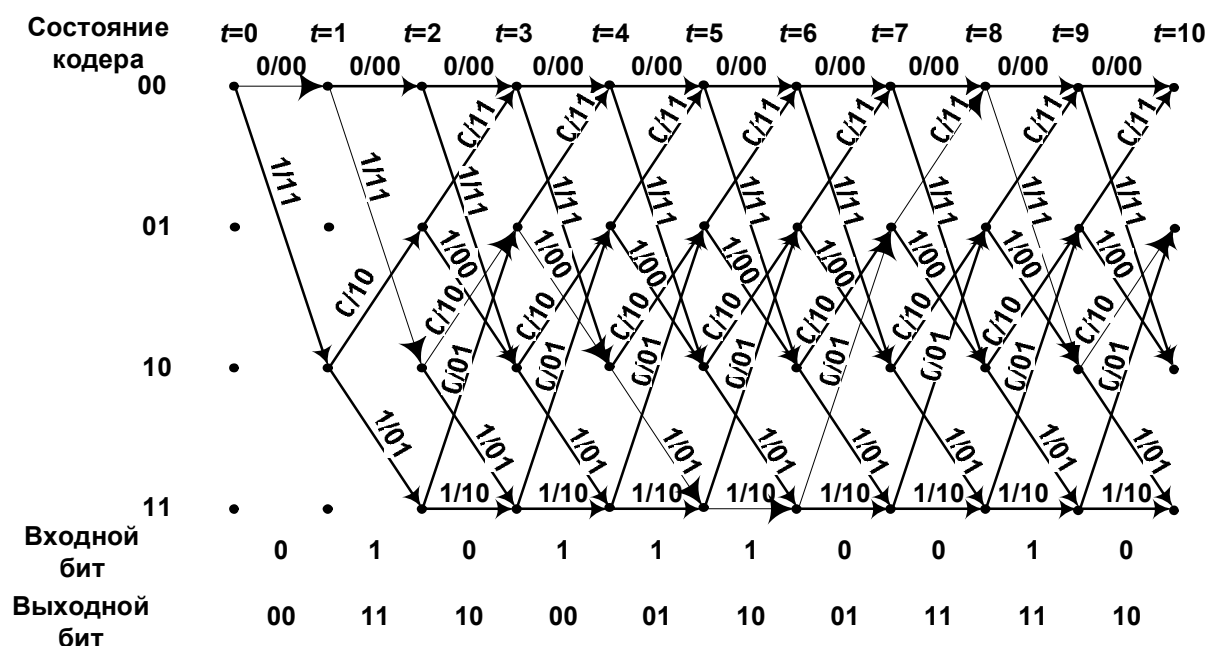


Рис. 1.43. Временная диаграмма состояния кодера для рассматриваемой входной последовательности бит

Основным достоинством изложенного выше метода треллис-кодирования является его помехоустойчивость. Как будет показано в дальнейшем, благодаря избыточности кодирования (вспомним, что каждому информационному биту ставится в соответствие дибит, то есть избыточность кода равна 1/2) даже в случае возникновения ошибок приема (к примеру, вместо дибита 11 ошибочно принят дибит 10) исходная последовательность бит может быть безошибочно восстановлена.

Для восстановления исходной последовательности бит на стороне приемника используется декодер Витерби.

1.8. Декодер Витерби

Декодер Витерби в случае безошибочного приема всей последовательности дибитов 00 11 10 00 01 10 01 11 11 10 будет обладать информацией об этой последовательности, а также о строении кодера (то есть о его диаграмме состояний) и о его начальном состоянии (00). Исходя из этой информации он должен восстановить исходную последовательность бит. Рассмотрим, каким образом происходит восстановление исходной информации.

Зная начальное состояние кодера (00), а также возможные изменения этого состояния (00 и 10), построим временную диаграмму для первого мо-

мента времени (рис. 1.44). На этой диаграмме из состояния 00 существует только два возможных пути, соответствующих различным входным дибитам. Поскольку входным дибитом декодера является 00, то, пользуясь диаграммой состояний кодера Треллиса, устанавливаем, что следующим состоянием кодера будет 00, что соответствует исходному биту 0.

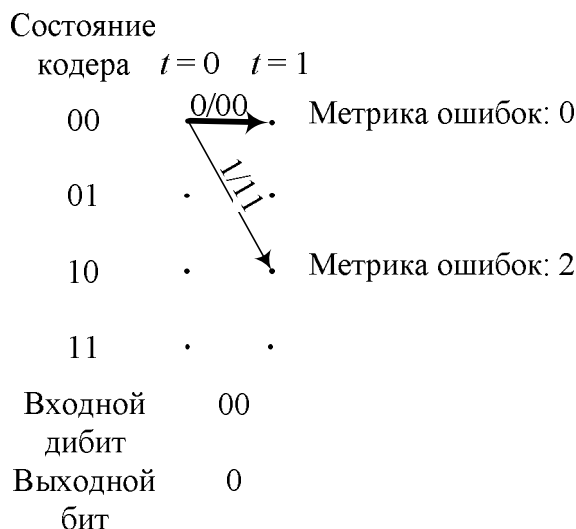


Рис. 1.44. Временная диаграмма возможных состояний кодера для первых двух моментов времени

Однако у нас нет 100% гарантии того, что принятый дибит 00 является правильным, поэтому не стоит пока отмечать и второй возможный путь из состояния 00 в состояние 10, соответствующий дибиту 11 и исходному биту 1. Два пути, показанные на диаграмме, отличаются друг от друга так называемой метрикой ошибок, которая для каждого пути рассчитывается следующим образом. Для перехода, соответствующего принятому дибиту (то есть для перехода, который считается верным), метрика ошибок принимается равной нулю, а для остальных переходов она рассчитывается по количеству отличающихся битов в принятом дибите и дибите, отвечающем рассматриваемому переходу. Например, если принятый дибит 00, а дибит, отвечающий рассматриваемому переходу, равен 11, то метрика ошибок для этого перехода равна 2.

Для следующего момента времени, соответствующего принятому дибиту 11, возможными будут два начальных состояния кодера: 00 и 10, а конечных состояния будет четыре: 00, 01, 10 и 11 (рис. 1.45). Соответственно для этих конечных состояний существует несколько возможных путей, отличающихся друг от друга метрикой ошибок. При расчете метрики ошибок необходимо учитывать метрику предыдущего состояния, то есть если для предыдущего момента времени метрика для состояния 10 была равной 2, то при переходе из этого состояния в состояние 01 метрика ошибок нового состояния (метрика всего пути) станет равной $2 + 1 = 3$.

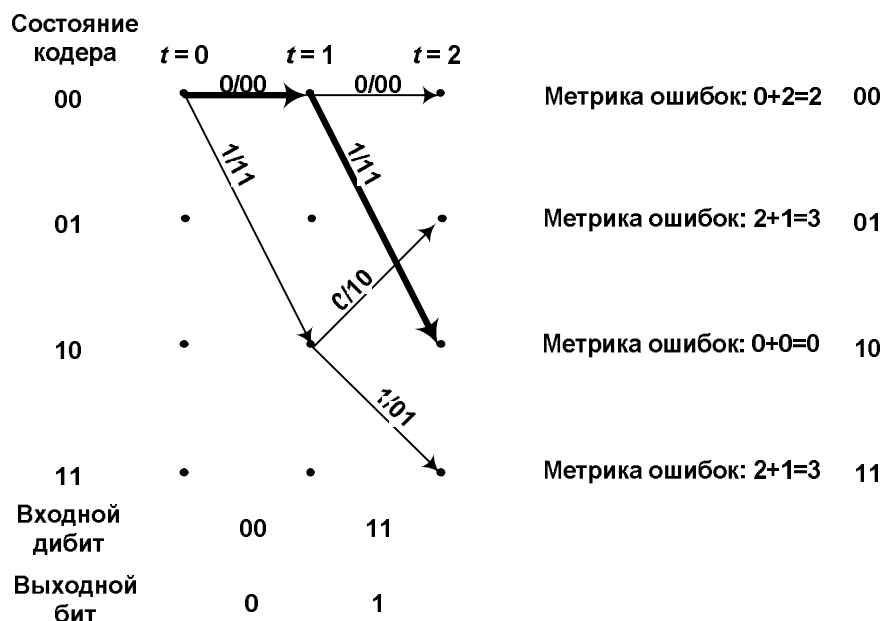


Рис. 1.45. Временная диаграмма возможных состояний кодера для первых двух моментов времени

Для следующего момента времени, соответствующего принятому дибиту 10, отметим, что в состояния 00, 01 и 11 ведут по два пути (рис. 1.46). В этом случае необходимо оставить только те переходы, которым отвечает меньшая метрика ошибок. Кроме того, поскольку переходы из состояния 11 в состояние 11 и в состояние 01 отбрасываются, переход из состояния 10 в состояние 11, отвечающий предыдущему моменту времени, не имеет продолжения, поэтому тоже может быть отброшен. Аналогично отбрасывается переход, отвечающий предыдущему моменту времени из состояния 00 в 00.

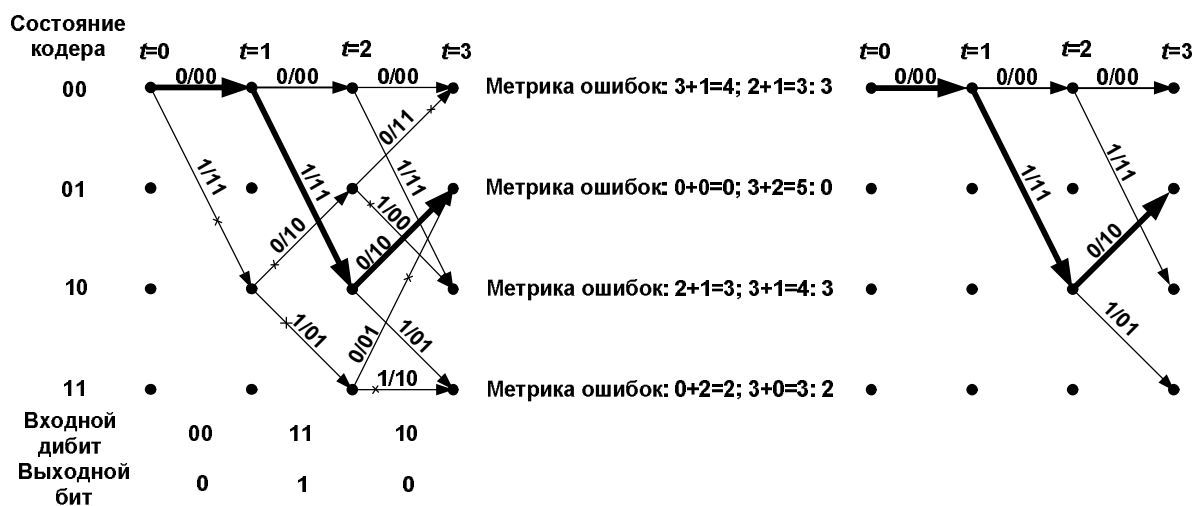


Рис. 1.46. Временная диаграмма возможных состояний кодера для первых трех моментов времени

Продолжая подобные рассуждения, можно вычислить метрику всех возможных путей и изобразить все возможные пути.

При этом количество самих возможных путей оказывается не так велико, как может показаться, поскольку большинство из них отбрасываются в процессе построения, как не имеющие продолжения (рис. 1.47). К примеру, на пятом такте работы декодера по описанному алгоритму остается всего четыре возможных пути.

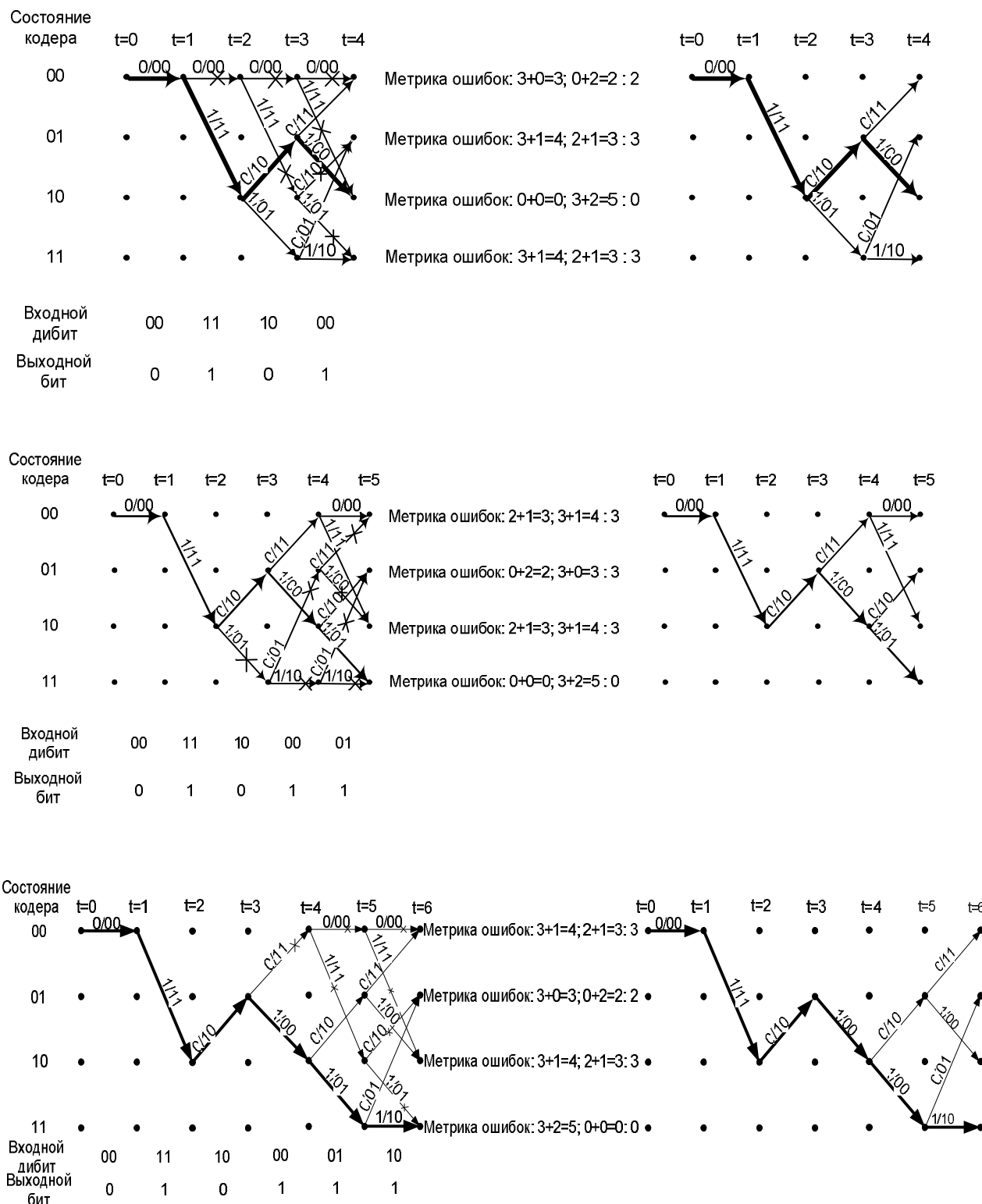


Рис. 1.47. Временная диаграмма возможных состояний кодера для четвертого, пятого и шестого тактов

Аналогично и на последнем такте работы декодера имеется всего четыре возможных пути (рис. 1.48), причем истинный путь, однозначно восстанавливающий исходную последовательность битов 0101110010, соответствует метрике ошибок, равной 0.

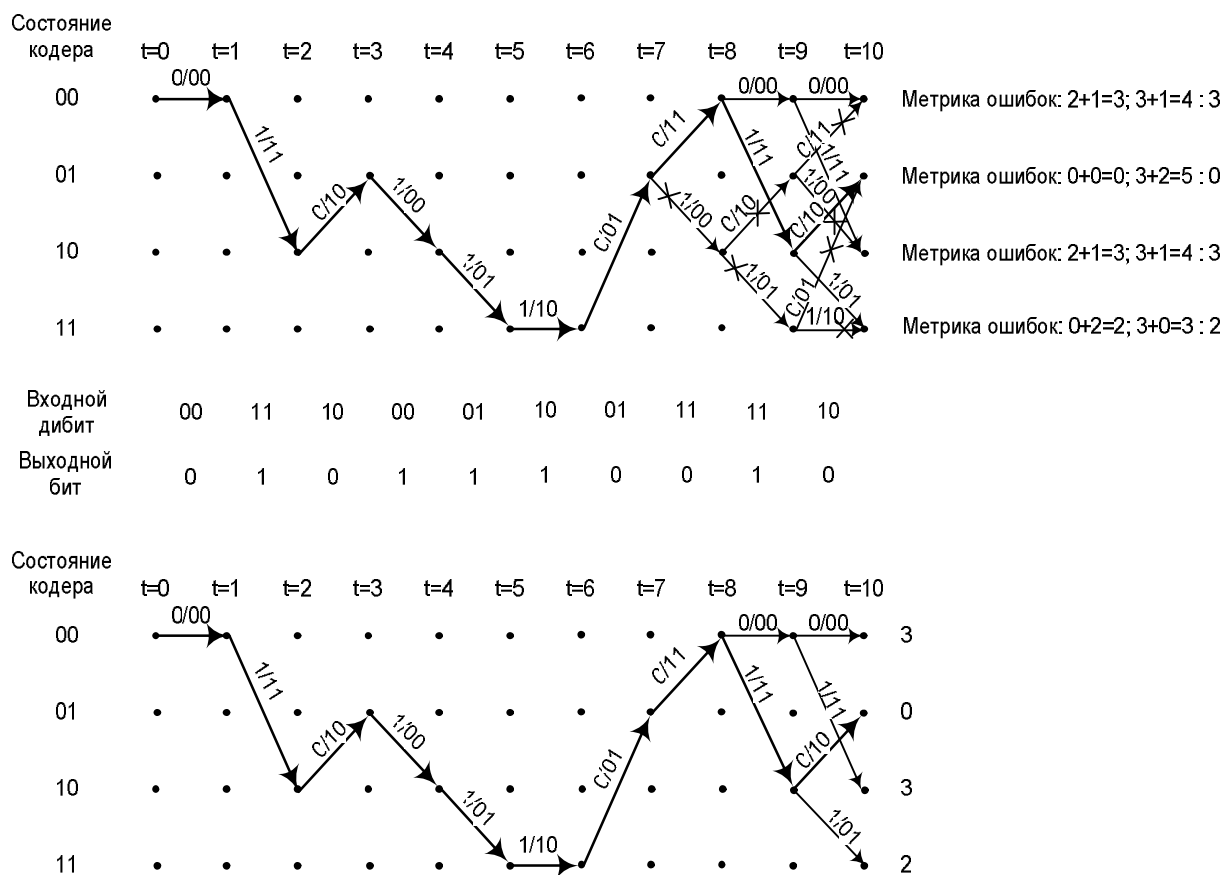


Рис. 1.48. Временная диаграмма возможных состояний кодера для последнего момента времени

При построении рассмотренных временных диаграмм удобно отображать метрику накопленных ошибок для различных состояний кодера в виде таблицы. Именно эта таблица и является источником той информации, на основе которой возможно восстановить исходную последовательность бит (табл. 1.8).

В описанном выше случае мы предполагали, что все принятые декодером дибиты не содержат ошибок. Рассмотрим далее ситуацию, когда в принятой последовательности дибитов содержатся две ошибки. Пусть вместо правильной последовательности 00 11 10 00 01 10 01 11 11 10 декодер принимает последовательность 00 11 11 00 11 10 01 11 11 10, в которой третий и пятый дибит являются сбойными. Попробуем применить рассмотренный выше алгоритм Витерби, основанный на выборе пути с наименьшей метрикой ошибок, к данной последовательности и выясним, сможем ли мы восстановить в правильном виде исходную последовательность битов, то есть исправить сбойные ошибки.

Таблица 1.8

Метрика ошибок для различных состояний кодера

Состояния кодера	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
00	-	0	2	3	2	3	3	3	0	2	3
01	-	-	3	0	3	2	2	0	3	3	0
10	-	2	0	3	0	3	3	3	2	0	3
11	-	-	3	2	3	0	0	2	3	3	2
Входной дибит		00	11	10	00	01	10	01	11	11	10
Выходной бит		0	1	0	1	1	1	0	0	1	0

Вплоть до получения третьего (сбойного) дибита алгоритм вычисления метрики ошибок для всех возможных переходов не отличается от рассмотренного ранее случая. До этого момента наименьшей метрикой накопленных ошибок обладал путь, отмеченный на рис. 1.49 полужирной линией. После получения такого дибита уже не существует пути с метрикой накопленных ошибок, равной 0. Однако при этом возникнут два альтернативных пути с метрикой, равной 1. Поэтому выяснить на данном этапе, какой бит исходной последовательности соответствует полученному дибиту, невозможно.

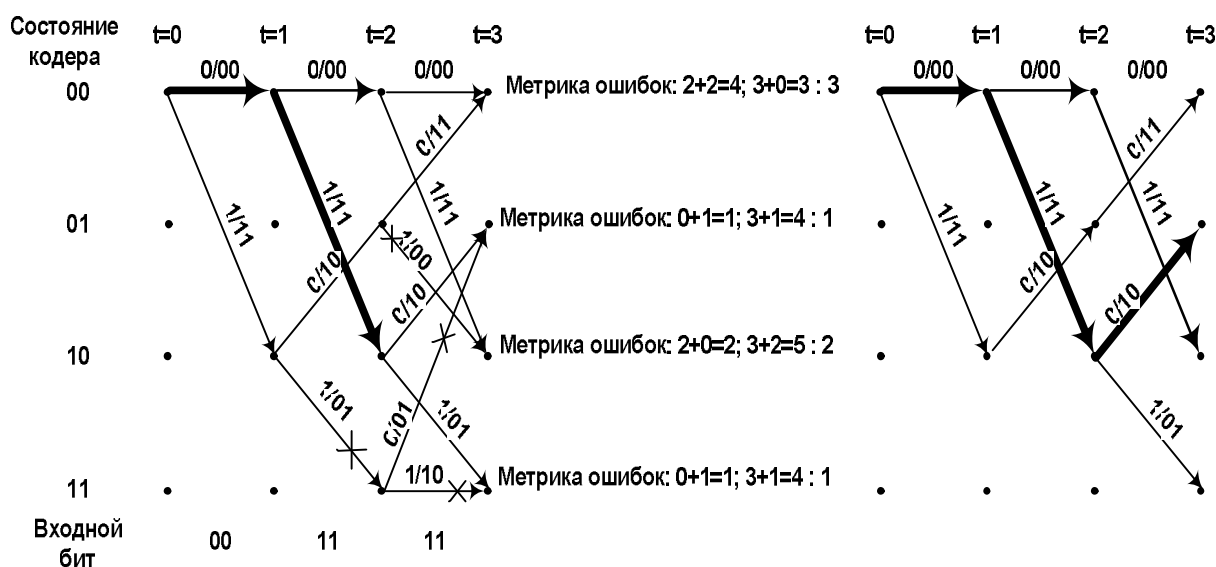
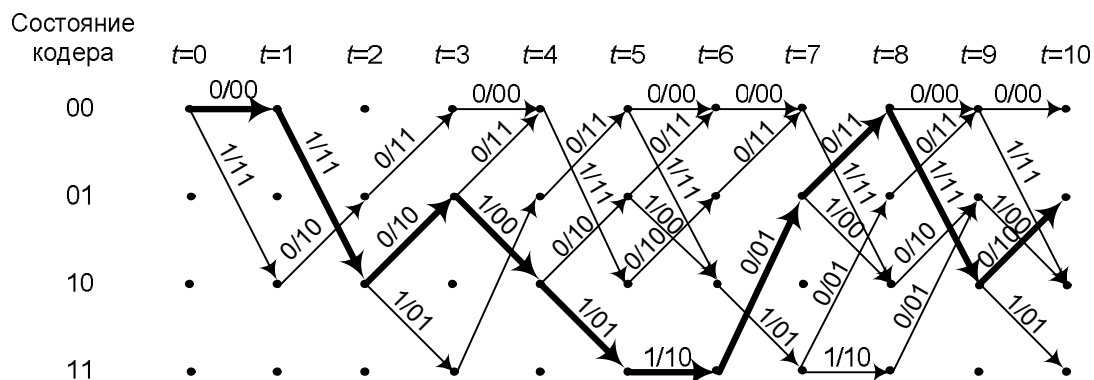


Рис. 1.49. Вычисление метрики накопленных ошибок при получении первого ошибочного дибита

Figure 1 consists of two state transition diagrams for a 2-bit Viterbi decoder. The left diagram shows the full state space with 4 states (00, 01, 10, 11) and 6 time steps (t=0 to t=5). The right diagram shows a reduced state space with 3 states (00, 01, 10) and 6 time steps, where state 11 is pruned. Both diagrams include branch metrics (e.g., 0/00, 1/11) and path metrics (e.g., 3+2=5).

После получения десятого дибита количество возможных путей с различной метрикой накопленных ошибок станет достаточно большим (рис. 1.51), однако на приведенной диаграмме (с использованием табл. 1.9, где представлена метрика накопленных ошибок для различных путей) не трудно выбрать единственный путь с наименьшей метрикой (на рис. 1.51 этот путь отмечен полужирной линией). По данному пути, пользуясь диаграммой состояния треллис-кодера (см. рис. 1.41), можно однозначно восстановить исходную последовательность бит 0101110010, невзирая на допущенные ошибки при получении дибитов.



58

Метрика накопленных ошибок для различных путей

Состояния кодера	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
00	-	0	2	3	3	2	3	4	2	4	5
01	-	-	3	1	2	2	3	2	4	5	2
10	-	2	0	2	1	3	3	4	4	2	5
11	-	-	3	1	2	2	2	3	4	5	4
Полученный дибит		00	11	11	00	11	10	01	11	11	10
Выходной бит		0	1	0 1*	1	0* 1	1	0	0	1	0

Рассмотренный сверточный кодер Треллиса на три состояния и алгоритм Витерби являются простейшими примерами, иллюстрирующими основной принцип работы. В реальности используемые кодеры Треллиса (и в гигабитных адаптерах, и в модемах) гораздо более сложные, но именно благодаря их избыточности удастся значительно повысить помехоустойчивость протокола передачи данных.

1.9. Треллис-кодирование 4D/PAM-5

Рассмотренный пример сверточного кодера имел всего четыре различных состояния: 00, 01, 10 и 11. В технологии 1000*BaSe-T* используется сверточный кодер уже на восемь различных состояний (с тремя элементами задержки), поэтому он называется восьмипозиционным. Кроме того, поскольку символы передаются по всем четырем витым парам кабеля одновременно с использованием пятиуровневого кодирования *PAM-5* (см. пункт 3.3), такое кодирование получило название четырехмерного 4D/*PAM-5*.

Другим существенным отличием кодера Треллиса, используемого в технологии 1000*BaSe-T*, является алгоритм перехода между различными состояниями кодера. В рассмотренном нами простейшем примере состояние кодера в следующий момент времени определялось исключительно текущим состоянием и входным битом. Так, если текущее состояние 00, а входной бит 1, то следующее состояние, то есть поле сдвига битов по запоминающим ячейкам, будет соответствовать 10. В реальном восьмипозиционном кодере Треллиса, управляющих (входных) битов два, а переходы между различными состояниями определяются по алгоритму наибольшего расстояния между точками сигнального созвездия.

Вспомним, что в коде *PAM-5* используется пять уровней для передачи сигналов: -2 , -1 , 0 , $+1$, $+2$. При этом уровням $+2/-2$ соответствует напряжение $+1/-1$ В, а уровням $+1/-1$ — напряжение $+0,5/-0,5$ В. Учитывая, что по четырем витым парам одновременно передается четыре уровня сигнала и каждый

из этих уровней может принимать одно из пяти значений, всего получаем 625 ($5 \times 5 \times 5 \times 5$) разных комбинаций сигналов. Различные возможные состояния сигнала удобно изображать на так называемой сигнальной плоскости. На этой плоскости каждое возможное состояние сигнала изображается сигнальной точкой, а совокупность всех сигнальных точек называют сигнальным созвездием. Естественно, что изобразить четырехмерное пространство не представляется возможным, поэтому рассмотрим для наглядности двухмерное сигнальное созвездие 5×5 . Такое созвездие формально может соответствовать двум витым парам. Изобразим вдоль оси X точки, отвечающие одной витой паре, а вдоль оси Y – другой. Тогда наше $2D$ -созвездие будет выглядеть так, как показано на рис. 1.52

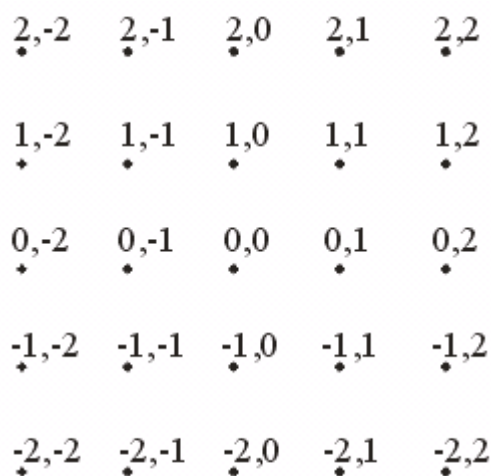


Рис. 1.52. Двухмерное сигнальное созвездие $2D/PAМ-5$ для двух витых пар

Обратим внимание, что минимальное расстояние между двумя точками такого созвездия равно 1.

Под воздействием шума и затухания сигнала сигнальное созвездие претерпевает искажения (рис. 1.53), в результате которых положение каждой сигнальной точки расплывается, а расстояние между ними сокращается. Вследствие этого точки в сигнальном созвездии становятся трудноразличимыми и велика вероятность их перепутывания.

Поэтому одной из задач кодера Треллиса является такое формирование сигнального созвездия, которое обеспечивало бы максимальное расстояние между различными сигнальными точками. Для того чтобы понять, как это делается, обозначим уровни сигналов -1 и $+1$ через X , а уровни -2 , 0 , $+2$ через Y . Тогда исходное созвездие можно изобразить в виде, показанном на рис. 1.54.

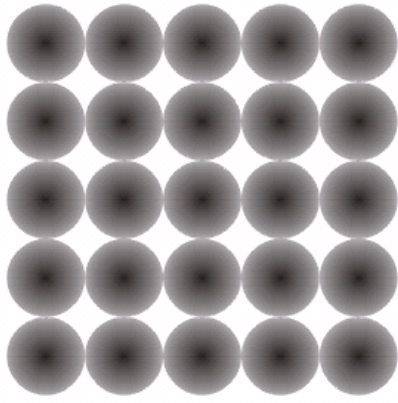


Рис. 1.53. Искажение сигнального созвездия в результате действия шума и затухания сигнала

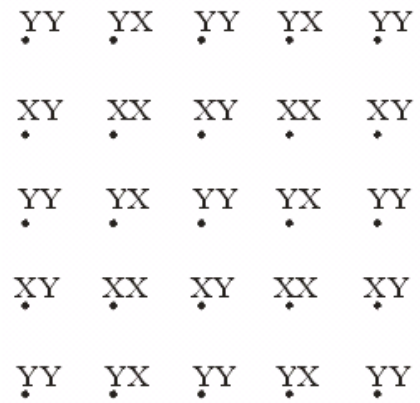


Рис. 1.54. Представление исходного созвездия

Разделив это созвездие на два подсозвездия, одно из которых сформировано из точек XX и YY , а другое – из точек XY и YX , можно увеличить расстояние между сигнальными точками до $\sqrt{2}$ (рис. 1.55).

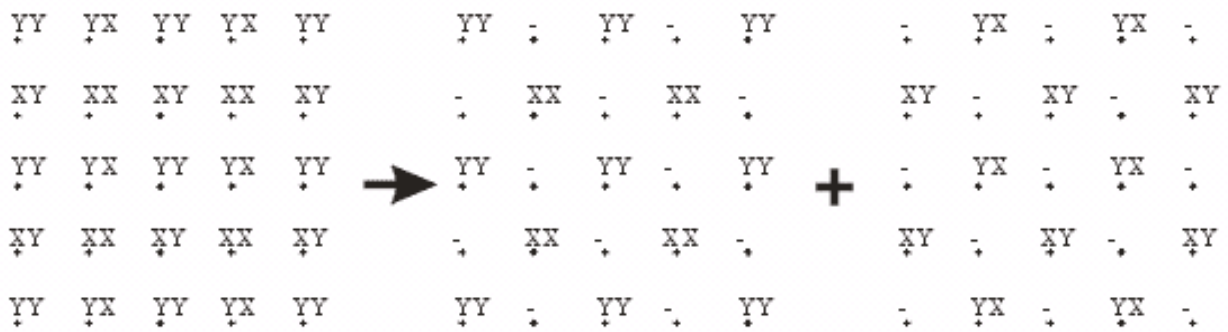


Рис. 1.55. Разделение исходного созвездия на два подсозвездия приводит к увеличению минимального расстояния между сигнальными точками

При использовании двух витых пар задача кодера Треллиса заключается в том, чтобы по одной витой паре посылать только символы, принадлежащие какому-либо одному из сигнальных созвездий, например $D0=XX+YY$, а по второй витой паре — символы, принадлежащие другому созвездию, например $D1=XY+YX$. Тогда расстояние между посылаемыми символами станет в два раза больше, чем было в исходном созвездии. В результате улучшается распознаваемость точек в сигнальном созвездии, то есть возрастает помехозащищенность.

Приблизительно по такой же схеме работает и реальный треллис-кодер, формирующий символы, посылаемые по четырем витым парам, однако, поскольку каждой точке созвездия отвечают четыре координаты (по одной на каждую пару) и каждая точка может принимать значение X или Y , то всего существует 16 различных комбинаций, из которых можно сформировать восемь подсозвездий:

$$\begin{aligned}
D0 &= XXXX + YYY \\
D1 &= XXXY + YYYX \\
D2 &= XXYY + YYXX \\
D3 &= XXYX + YXYX \\
D4 &= XYXX + YXYX \\
D5 &= XYXY + YXYX \\
D6 &= XYYX + YXXY \\
D7 &= XYYY + YYYX
\end{aligned}$$

В полученных подсозвездиях минимальное расстояние между точками в два раза больше, чем в исходном созвездии. Кроме того, минимальное расстояние между точками двух разных подсозвездий также равно 2. Именно эти восемь сигнальных созвездий формируют диаграмму состояний треллис-кодера. К примеру, состоянию кодера 000 соответствует комбинация точек из созвездий $D0D2D4D6$ в том смысле, что по первой паре передаются точки из созвездия $D0$, по второй паре — из созвездия $D2$ и т.д. Следующему возможному состоянию кодера будет соответствовать такая комбинация, при которой минимальное расстояние между посылаемыми символами по каждой паре равно 2.

Использование треллис-кодирования по описанной схеме позволяет снизить соотношение «сигнал/шум» (SNR) на 6 дБ, то есть значительно увеличить помехоустойчивость при передаче данных.

2. ШИРОКОПОЛОСНЫЕ СИГНАЛЫ

2.1. Основные термины и определения

Широкополосные сигналы первоначально нашли применение в военных и специальных системах связи, для которых характерны низкая вероятность перехвата и повышенная помехозащищенность [3]. Эти методы лежат в основе так называемой технологии расширения спектра сигналов, которая в последнее время начинает широко использоваться в коммерческих системах связи [1].

Технология расширения спектра сигналов заключается в преднамеренном увеличении в передатчике полосы частот, занимаемой сигналом-переносчиком сообщений и существенно превышающей ту, которая необходима при передаче с требуемыми скоростью и уровнем искажений, и обратном ее уменьшении до информационной полосы частот в приемнике системы связи.

Тогда полоса частот, занимаемая полезным сообщением будет $\Delta F_b = 1/\tau_0$, а полоса частот широкополосного сигнала $\Delta F_s = 1/\Delta\tau_0$. Так как $\Delta\tau_0 < \tau_0$, то и $\Delta F_s > \Delta F_b$.

Поясним сказанное с помощью рис. 2.1. Процесс расширения спектра полезного сообщения заключается в том, что элемент передаваемого сообщения $U_b(t)$ длительностью τ_0 заменяется случайной последовательностью $U_s(t)$, где каждый элемент имеет длительность $\Delta\tau_0$.

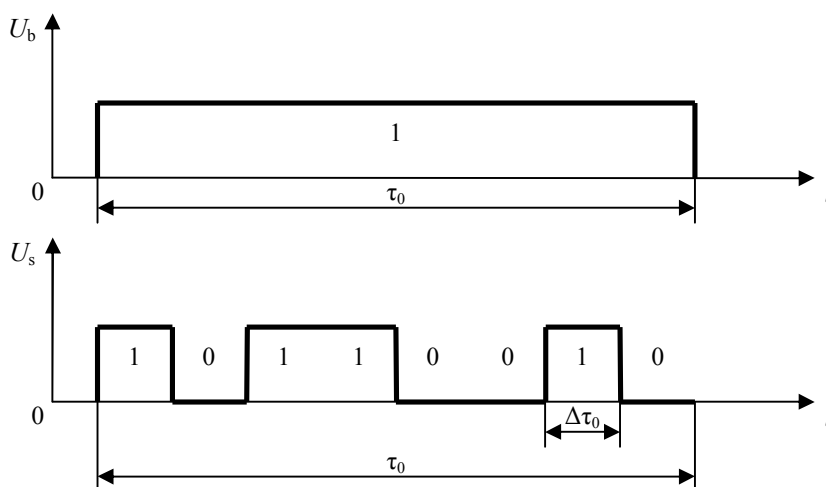


Рис. 2.1. Процесс замены одного бита сложным сигналом

В результате выполнения операции расширения спектра происходит уменьшение уровня спектральной плотности передаваемого сигнала, что затрудняет обнаружение факта работы системы связи и перехват передаваемых сообщений. Благодаря низкому уровню спектральной плотности излучаемых сигналов с расширенным спектром системы связи могут использовать занятые участки радиочастотных диапазонов, не создавая существенных помех другим системам связи и службам.

При обратной операции сжатия спектра происходит восстановление первоначального спектра сигнала-переносчика сообщений и расширение, «размазывание» спектра сигналов преднамеренных и непреднамеренных помех. А поскольку в информационную полосу попадает лишь часть энергии сигналов, «размытой» в широкой полосе частот, то в результате обеспечивается увеличение отношения мощностей сигнала и помехи (отношение сигнал/помеха), определяемое соотношением полос до и после сжатия спектра сигнала. Процесс ослабления помехи при обработке сигналов с расширенным спектром условно показан на рис. 2.2, б.

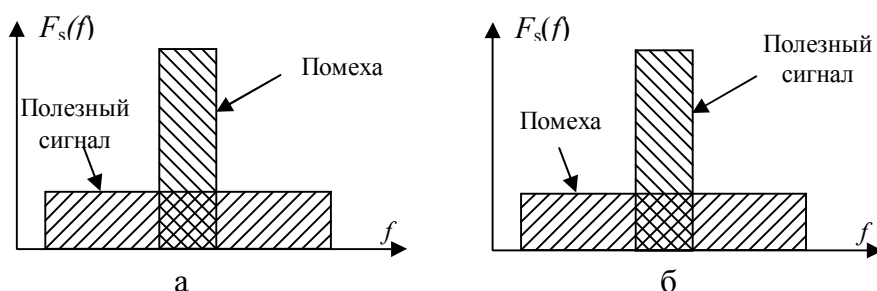


Рис. 2.2. Спектры сигнала и помехи при обработке сигналов с расширенным спектром: а – в канале связи; б – на выходе приёмника

Широкополосными сигналами (ШПС) называют такие сигналы, у которых произведение ширины спектра ΔF_s на длительность τ_0 много больше единицы. Это произведение называется базой сигнала и обозначается B , т.е.

$$B = \Delta F_s \tau_0. \quad (2.1)$$

У ШПС $B \gg 1$. Широкополосные сигналы иногда называют сложными, псевдослучайными, шумоподобными, составными, многомерными.

В цифровых системах, передающих информацию в виде двоичных символов, длительность ШПС и скорость передачи информации R_b связаны соотношением $\tau_0 = 1/R_b$. Поэтому база ШПС

$$B = \Delta F_s / R_b \quad (2.2)$$

характеризует расширение спектра ШПС относительно спектра сообщения. В аналоговых системах связи, у которых частота сообщения равна F_m и частота дискретизации равна $2F_m$,

$$B = \Delta F_s / 2F_m. \quad (2.3)$$

И если $B \gg 1$, то $\Delta F_s \gg R_b$ и $\Delta F_s \gg 2F_m$. Именно поэтому системы с ШПС в зарубежной литературе получили название систем с расширенным (или распределенным) спектром, а в отечественной литературе – широкополосные системы.

Известны несколько методов модуляции с расширением спектра сигнала, которые можно классифицировать в соответствии с тем, по какому из четырех основных параметров радиосигнала: амплитуде, фазе, частоте или временному положению (задержке) осуществляется модуляция сигналом, расширяющим спектр.

Поскольку для обеспечения высокого КПД выходных каскадов передатчика амплитуду желательно сохранять постоянной, то к настоящему времени наибольшее распространение получили методы расширения спектра сигналов: основанные на изменении их фазы, частоты и временного положения (задержки) в соответствии с некоторым законом. Среди этих методов можно выделить следующие базовые:

- непосредственная модуляция несущей частоты псевдослучайной последовательностью (ПСП), иначе псевдослучайная частотная или фазовая манипуляция; в результате формируется так называемый сигнал с прямым расширением спектра;
- программная перестройка рабочей частоты (ППРЧ), приводящая к формированию сигнала со скачкообразным изменением несущей частоты;
- программная перестройка временного положения импульсных сигналов или псевдослучайная время-импульсная модуляция (ПВИМ). В итоге получаются импульсные сигналы со скачкообразным изменением временного положения;

– частотная модуляция по линейному закону (ЛЧМ); в этом случае мгновенная частота радиосигнала в течение интервала времени, равного длительности сигнала T_s , либо нарастает, либо убывает по линейному закону и в результате формируются ЛЧМ сигналы с различными законами изменения частоты.

Рассмотрим систему связи, в которой осуществляется передача дискретных сообщений со скоростью R_b , бит/с. После модуляции с расширением спектра полоса передаваемого сигнала увеличивается до ΔF_s Гц, причем $\Delta F_s \gg R_b$. Величину ΔF_s будем называть шириной полосы сигнала с расширенным спектром. Предположим, что в канале связи действует помеха мощностью P_{Π} , которая распределена в пределах полосы ΔF_{Π} . После сжатия спектра ширина полосы полезного сигнала снова становится равной R_b , а спектральная плотность помехи – равной $P_0 = P_{\Pi} / \Delta F_s$. Отношение энергии сигнала, приходящейся на бит сообщения, к спектральной плотности эквивалентного шума определяется выражением

$$\frac{E_b}{P_0} = \frac{P_s / R_b}{P_{\Pi} / \Delta F_s} = \frac{P_s \cdot \Delta F_s}{P_{\Pi} \cdot R_b} = \frac{P_s}{P_{\Pi}} B, \quad (2.3)$$

где $\frac{P_s}{P_{\Pi}}$ – отношение мощностей полезного сигнала и помехи, а $\frac{\Delta F_s}{R_b} = B$ – база сигнала или выигрыш при обработке сигналов системы.

Так как при оптимальной демодуляции сигналов вероятность ошибки зависит только от значения отношения E_b/P_0 , то максимально допустимое превышение мощности помехи над мощностью сигнала полностью определяется значением выигрыша при обработке B . Если учесть возможные энергетические потери L_s при реализации системы связи, то максимально допустимое превышение мощности помехи над мощностью сигнала, выраженное в децибелах будет определяться так называемым коэффициентом помехозащищенности

$$Q = B - L_s - E_b/P_0. \quad (2.4)$$

Например, при $R_b = 5$ кбит/с, $\Delta F_s = 20$ МГц, $E_b/P_0 = 20$ и $L_s = 2$ коэффициент помехозащищенности $Q = 36$ дБ - 13 дБ - 3 дБ = 20 дБ, т.е. мощность помехи может превышать мощность сигнала в 100 раз.

2.2. Методы расширения спектра частот передаваемых сообщений

2.2.1. Прямое расширение спектра. Обобщенная структурная схема системы связи, использующей сигналы с прямым расширением спектра, изображена на рис. 2.3. Она содержит источник дискретных сообщений, модулятор первой ступени, модулятор второй ступени, синтезатор частот, преобразователь частоты вверх, усилитель мощности, передающую и приемную антенны, преселектор, маломощный усилитель, преобразователь частоты вниз, синтезатор частот, демодулятор второй ступени, демодулятор первой ступени, блоки поиска и синхронизации.

В модуляторе первой ступени осуществляется модуляция сигнала, как правило, промежуточной частоты сигналом передаваемого дискретного сообщения $D(t)$. На первой ступени модуляции могут быть использованы все рассмотренные ранее традиционные и нетрадиционные виды модуляции: фазовая (ФМ), частотная (ЧМ) и спектрально-эффективные: квадратурная, фазовая, квадратурная фазовая со сдвигом, квадратурная фазовая с фазовым сдвигом $\pi/4$, частотная с минимальным частотным сдвигом и частотная с минимальным частотным сдвигом и гауссовской предмодуляционной фильтрацией. В результате формируется сигнал

$$s_1(t) = \text{Re}\{A_1 \exp[j(2\pi f_1 t + \theta(t) + \theta_0(t))]\}, \quad (2.5)$$

где A_1 – амплитуда сигнала; F_1 – частота модулируемого гармонического колебания; $Q(t)$ – изменения фазы, обусловленные передаваемым сообщением и зависящие от вида используемой модуляции; $\theta_0(t)$ – случайная начальная фаза.

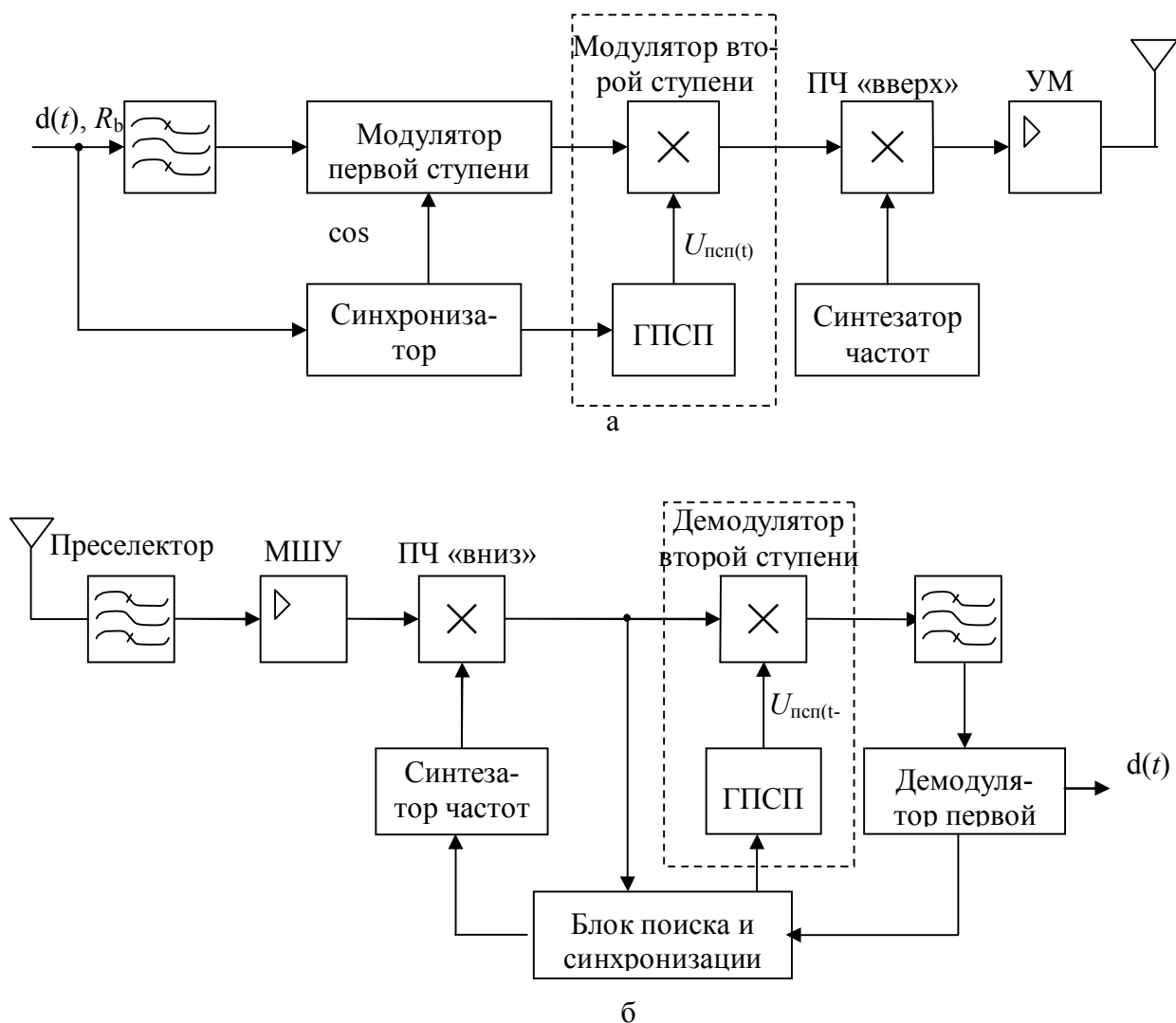


Рис. 2.3. Структурная схема системы связи, использующая сигналы с прямым расширением спектра: а – передающая часть; б – приёмная часть

В модуляторе второй ступени осуществляется расширение спектра сигнала путем балансной модуляции сигналом расширения спектра в виде двоичной псевдослучайной последовательности (ПСП), формируемой генератором (ГПСП). На выходе модулятора второй ступени формируется сигнал с расширенным спектром

$$s_2(t) = \text{Re}\{A_2 U_{\text{ПСП}} \exp[j(2\pi f_1 t + \theta(t) + \theta_0(t))]\}, \quad (2.6)$$

где

$$U_{\text{ПСП}}(t) = \sum_{n=-\infty}^{\infty} C_n U_{T_0}(t - nT_0);$$

C_n – элементы ПСП, принимающие значения ± 1 ; $U_{T_0}(t)$ – функция, описывающая форму одиночного импульса сигнала ПСП. После переноса на частоту излучения F_0 и усиления передается сигнал

$$s(t) = \text{Re}\{AU_{\text{ПСП}} \exp[j(2\pi f_1 t + \theta(t) + \theta_0)]\}. \quad (2.7)$$

В приемнике после предварительной селекции и усиления принимаемый сигнал переносится на промежуточную частоту для последующей обработки. В демодуляторе второй ступени осуществляется сжатие спектра входного сигнала путем его перемножения с опорной ПСП и последующей полосовой фильтрации. В результате на выходе полосового фильтра формируется сигнал

$$\hat{s}_2(t) = \text{Re}\{KA_2 \overline{U_{\text{ПСП}}(t - \tau)} U_{\text{ПСП}}(t - \hat{\tau}) \exp[j(2\pi f_1 t + \theta(t) + \theta_0(t))]\}, \quad (2.8)$$

где K – коэффициент передачи приемопередающего тракта и среды распространения, черта сверху – символ временного усреднения; τ и $\hat{\tau}$ – соответственно задержка модулирующей ПСП в точке приема и ее оценка, формируемая в блоке поиска и синхронизации.

При $\tau = \hat{\tau}$ модуляция, расширяющая спектр сигнала, полностью устраняется и получается сигнал

$$s_1(t) = \text{Re}\{KA_2 \exp[j(2\pi f_1 t + \theta(t) + \theta_0(t))]\}, \quad (2.9)$$

который поступает на вход демодулятора первой ступени, где осуществляется выделение передаваемого сообщения $\hat{d}(t)$.

Таким образом, необходимым условием сжатия спектра входного сигнала является наличие «жесткой» временной синхронизации модулирующей и опорной ПСП. Она обеспечивается блоком поиска и синхронизации и осуществляется в два этапа: на первом – устраняется начальная неопределенность в результате выполнения процедуры поиска с точностью до длительности $\Delta\tau_0$ элемента ПСП, на втором – точная синхронизация. За счет работы следящей системы (системы слежения за задержкой) ошибки синхронизации уменьшаются до значений, не превышающих $(0,05 - 0,1)\Delta\tau_0$ [4].

Поиск осуществляется, например, путем последовательного временного сдвига на величину $0,5\Delta\tau_0$ опорной ПСП через интервалы времени T_a и проверки наличия синхронизации модулирующей и опорной ПСП с помощью коррелятора и решающего устройства. Если сигнал на выходе коррелятора превышает пороговое значение, устанавливаемое в соответствии с выбранным критерием обнаружения, то поиск прекращается и включается система слежения за задержкой.

Система слежения за задержкой содержит временной дискриминатор, сглаживающий фильтр, перестраиваемый генератор тактовых импульсов и генератор опорных ПСП. Временной дискриминатор формирует управляющее напряжение, под воздействием которого изменяется задержка тактовых импульсов и опорных ПСП до тех пор, пока управляющее напряжение не станет равным или близким нулю.

Трудности синхронизации генератора опорных ПСП с точностью до нескольких наносекунд ограничивают частоту следования символов ПСП значениями порядка десятков мегасимволов на секунду. Это означает, что ширина спектра ΔF_s сигнала при таком методе расширения спектра не может превышать нескольких десятков мегагерц.

Определим значение выигрыша при обработке, обеспечиваемого сигналом с прямым расширением спектра, полагая, что для передачи символов сообщения используется двоичная фазовая манипуляция, а для расширения спектра – случайная двоичная последовательность импульсов прямоугольной формы длительностью $\Delta\tau_0$ и равновероятным появлением значений амплитуды $+A$ и $-A$.

Спектральная плотность такой последовательности определяется следующим выражением:

$$F_s(f) = A^2 \Delta\tau_0 \frac{\sin^2(\pi f \Delta\tau_0)}{(\pi f \Delta\tau_0)^2} = \frac{A^2 \tau_0}{B} \frac{\sin^2(\pi f \tau_0 / B)}{(\pi f \tau_0 / B)}, \quad (2.10)$$

где τ_0 – длительность двоичного символа сообщения (бита); $B = \tau_0 / \Delta\tau_0$ – отношение длительности двоичных символов передаваемого сообщения и последовательности, используемой для расширения спектра.

График спектральной плотности сигналов с прямым расширением спектра, построенный для положительных частот, изображен на рис. 2.4. Там же показан график спектральной плотности обычного сигнала с двоичной фазовой манипуляцией. При его построении учтено соотношение длительностей импульсов модулирующих сигналов.

Если ширину полосы частот сигналов определить по первым нулям спектральной плотности, то выигрыш при обработке, обеспечиваемый сигналом с прямым расширением спектра, будет определяться следующим выражением:

$$B = \frac{\Delta F_s}{\Delta F_d} = \frac{2\tau_0}{\Delta\tau_0} = \frac{2R_c}{R_b}, \quad (2.11)$$

где R_C – частота следования символов последовательности, используемой для расширения спектра; ΔF_D – ширина полосы сигнала передаваемого сообщения.

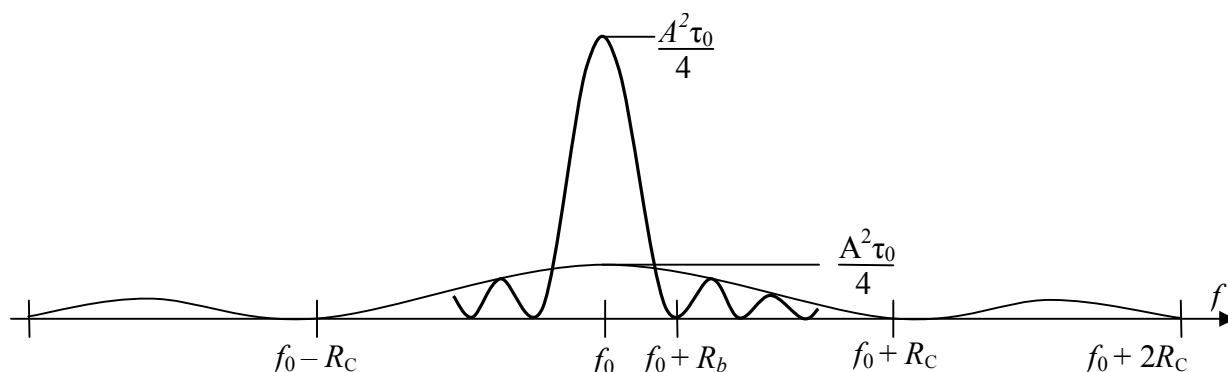


Рис. 2.4. Спектральные плотности сигнала с прямым расширением спектра и обычного сигнала с двоичной фазовой манипуляцией

Например, при $R_C = 48$ Мсимв/с, $R_b = 4,8$ Кбит/с имеем $B = 210^4$ или 43 дБ.

Перечислим некоторые свойства сигналов с прямым расширением спектра, наиболее важные с точки зрения организации множественного доступа в системах связи с подвижными объектами.

– **Множественный доступ.** Если одновременно несколько абонентов используют канал передачи, то в канале одновременно присутствуют несколько сигналов с прямым расширением спектра. Каждый из этих сигналов занимает всю полосу канала. В приемнике сигнала конкретного абонента осуществляется обратная операция – свертывание сигнала этого абонента путем использования того же псевдослучайного сигнала, который был использован в передатчике этого абонента. Эта операция концентрирует мощность принимаемого широкополосного сигнала снова в узкой полосе частот, равной ширине спектра информационных символов. Если взаимная корреляционная функция между псевдослучайными сигналами данного абонента и других абонентов достаточно мала, то при когерентном приеме в информационную полосу приемника абонента попадет лишь незначительная доля мощности сигналов остальных абонентов. Сигнал конкретного абонента будет принят верно.

– **Многолучевая интерференция.** Если псевдослучайный сигнал, используемый для расширения спектра, имеет идеальную автокорреляционную функцию, значения которой вне интервала $[-\tau_0, \tau_0]$ равны нулю, и если принимаемый сигнал и копия этого сигнала в другом луче сдвинуты во времени на величину, большую $2\tau_0$, то при сворачивании сигнала его копия может рассматриваться как мешающая интерференция, вносящая лишь малую долю мощности в информационную полосу.

– **Узкополосная помеха.** При когерентном приеме в приемнике осуществляется умножение принятого сигнала на копию псевдослучайного сигнала, используемого для расширения спектра в передатчике. Следовательно, в при-

емнике будет осуществляться операция расширения спектра узкополосной помехи, аналогичная той, которая выполнялась с информационным сигналом в передатчике. Следовательно, спектр узкополосной помехи в приемнике будет расширен в B раз, где B – коэффициент расширения, так что в информационную полосу частот попадет лишь малая доля мощности помехи, в B раз меньше исходной мощности помехи.

– **Вероятность перехвата.** Так как сигнал с прямым расширением спектра занимает всю полосу частот системы в течение всего времени передачи, то его излучаемая мощность, приходящаяся на 1 Гц полосы, будет иметь очень малые значения. Следовательно, обнаружение такого сигнала является очень трудной задачей.

Кроме перечисленных свойств, эти сигналы имеют и другие специфические свойства, одни из которых можно считать достоинствами (+), а другие – недостатками (–).

(+) Генерирование необходимых псевдослучайных сигналов может быть обеспечено простыми устройствами (регистрами сдвига).

(+) Операция расширения спектра может быть реализована простым умножением или сложением цифровых сигналов по модулю два.

(+) Генератор несущего колебания является простым, так как необходимо генерировать гармоническое несущее колебание только с одной частотой.

(+) Может быть реализован когерентный прием сигнала с прямым расширением спектра.

(+) Нет необходимости обеспечивать синхронизацию между абонентами системы.

(–) Выравнивание и поддержание синхронизации между генерируемым в приемнике и содержащимся в принимаемом сигнале псевдослучайными кодами является трудной задачей. Синхронизация должна поддерживаться с точностью

до малой доли длительности элементарного символа,

(–) Правильный прием информации обеспечивается только при высокой точности временной синхронизации, когда ошибка составляет малую долю длительности элементарного символа, что ограничивает возможность уменьшения длительности этого символа и, следовательно, возможность расширения полосы лишь до 10 – 20 МГц. Таким образом, существует ограничение на увеличение коэффициента расширения спектра.

(–) Мощность сигнала, принимаемого от близких к БС абонентов, намного превышает мощность сигнала далеких абонентов. Следовательно «близкий» абонент постоянно создает очень мощную помеху «далекому» абоненту, часто делая прием его сигнала невозможным. Эта проблема «близкий – далекий» может быть решена применением системы управления мощностью, излучаемой АС; цель управления – обеспечить одинаковую среднюю мощность сигналов разных пользователей на входе приемника БС. Однако обеспечение такого управления оказывается очень трудной задачей из-за задержек во времени в петле обратной связи, неточности оценивания уровней принимаемых

сигналов, ошибок при передаче команд управления, снижения скорости передачи информации.

2.2.2. Программная перестройка рабочей частоты. Обобщённая схема системы связи, использующей сигналы с программной перестройкой рабочей частоты (ППРЧ сигналы), изображена на рис. 2.5. В основном она содержит те же элементы, что и система, показанная на рис. 2.3. Отличие состоит в том, что цифровые синтезаторы частот в преобразователях частот «вверх» и «вниз» являются программно-перестраиваемыми по псевдослучайному закону. Здесь в явном виде отсутствуют модулятор и демодулятор второй ступени, а их функции выполняют преобразователи частот «вверх» и «вниз».

На выходе преобразователя частоты «вверх» формируется сигнал

$$s_2(t) = \operatorname{Re}\{A_2 c(t) \exp j[2\pi f_0 t + \theta(t)]\}, \quad (2.12)$$

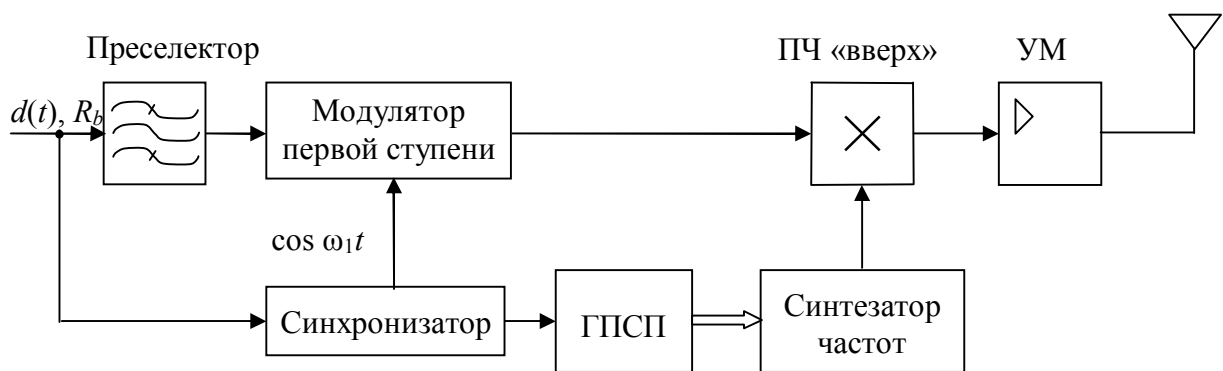
где

$$c(t) = \sum_{n=-\infty}^{\infty} U_{\Delta\tau_0}(t - n\Delta\tau_0) \exp\{j(2\pi f_n + \theta_n)\}; \quad (2.13)$$

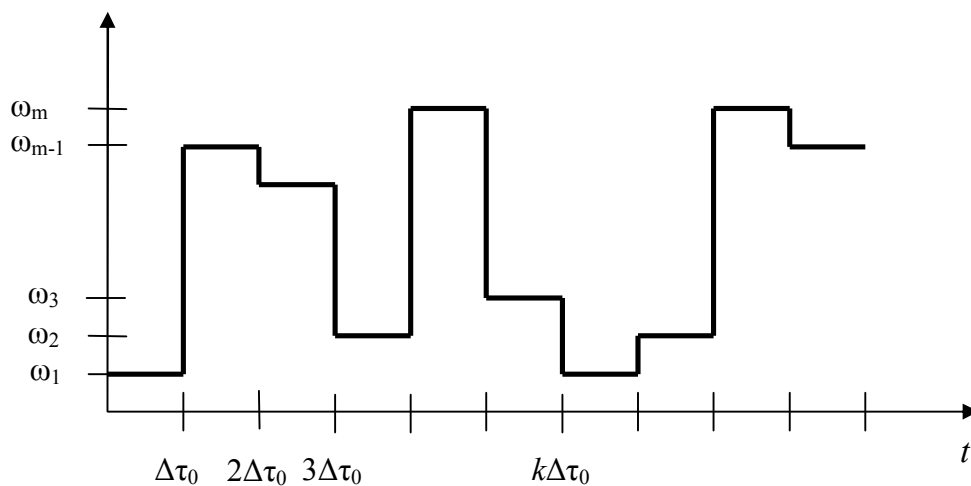
$U_{\Delta\tau_0}$ – функция, описывающая форму одиночного импульса длительностью $\Delta\tau_0$; $\{\theta_n\}$ – последовательность случайных фаз, возникающих при перестройке частоты на n -м интервале времени длительностью $\Delta\tau_0$; $\{F_n\}$ – псевдослучайная последовательность частотных сдвигов; $\theta(t)$ – изменения фазы, обусловленные модуляцией сигналом передаваемого сообщения.

Традиционно программная перестройка рабочей частоты используется в сочетании с двоичной многопозиционной частотной манипуляцией сигналом передаваемого сообщения. Перестройка частоты осуществляется путем непосредственного управления синтезатора сегментами ПСП, содержащими k символов. В результате диапазон изменения частоты несущей содержит 2^k различных частот. При достаточно широком диапазоне и высокой скорости перестройки трудно сохранить непрерывность фазы несущей (фазовую когерентность). Этим объясняется включение последовательности $\{\theta_n\}$ в модель сигнала (2.13).

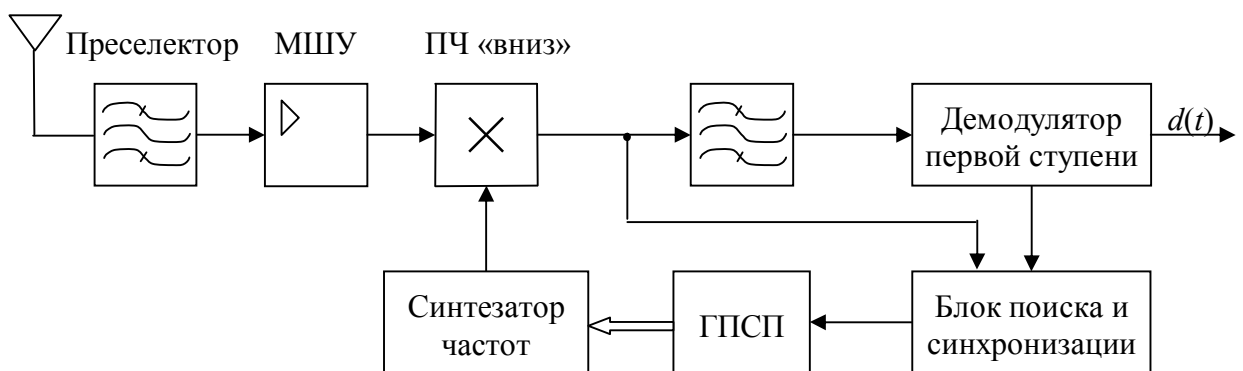
Если рассматривать кратковременный спектр сигнала с ППРЧ, например, при усреднении за длительность интервала перестройки $\Delta\tau_0$, то ширина полосы будет почти такой же, как и обычного ЧМ сигнала. При длительном усреднении спектр и соответственно ширина полосы частот сигнала с ППРЧ будет определяться диапазоном перестройки частот синтезатора, которая может достигать нескольких гигагерц. Это значительно превышает значения, получаемые при прямом расширении спектра.



а



б



в

Рис. 2.5. Структурная схема системы связи, использующей сигналы с программной перестройкой рабочей частоты:

а – передающая часть; б – закон изменения частоты несущей; в – приёмная часть

В преобразователе частоты «вниз» приемника частота опорного сигнала отличается от частоты принимаемого сигнала на значение промежуточной частоты F_1 . Поэтому на выходе преобразователя частоты «вниз»

$$\hat{s}_2(t) = \text{Re}\{KA_2 \exp[j2\pi f_1 t + f_0(t - \tau) + \theta(t)]\}. \quad (2.14)$$

При $\hat{\tau} = \tau$ изменение частоты по закону ПСП полностью устраняется, спектр входного сигнала сжимается и получается сигнал

$$\hat{s}_2(t) = \text{Re}\{KA_2 \exp[j((2\pi f_1 t - \theta(t) - \theta_0))]\}, \quad (2.15)$$

который поступает на вход демодулятора, осуществляющего выделение сигнала передаваемого сообщения $\hat{d}(t)$.

Таким образом, необходимым условием сжатия спектра является совпадение по форме управляющих сигналов $c(t)$ и $\hat{c}(t)$, определяющих изменение частоты соответственно входного и опорного сигналов. Выполнение этого условия обеспечивается блоком поиска и синхронизации в результате реализации процедур поиска и слежения, аналогичных тем, которые были рассмотрены выше.

В зависимости от соотношения между длительностями модулирующего τ_0 и управляющего перестройкой частоты $\Delta\tau_0$ сигналов различают медленную (МППРЧ) и быструю (БППРЧ) программную перестройку рабочей частоты. В первом случае в течение интервала перестройки $\Delta\tau_0$ может передаваться несколько символов сообщения, а во втором – один и тот же символ сообщения многократно передается на нескольких несущих частотах. Поэтому выполняются следующие соотношения:

$$\left. \begin{aligned} \Delta\tau_0 &= N\tau_0 \\ R_C &= R_b / N \end{aligned} \right\} \text{ для медленной ППРЧ;} \quad (2.16)$$

$$\left. \begin{aligned} \Delta\tau_0 &= \tau_0 / N \\ R_C &= NR_b \end{aligned} \right\} \text{ для быстрой ППРЧ.} \quad (2.17)$$

Таким образом, скорость передачи элементов сигнала (скорость манипуляции) здесь определяется максимальной из двух величин $R_b = 1/\tau_0$ и $R_C = 1/\Delta\tau_0$ и равна тактовой частоте.

Шаг перестройки, или расстояние между соседними частотами, определяется скоростью манипуляции, т.е. видом ППРЧ. В частности, при $R_b = R_C$ и использовании ППРЧ в сочетании с многопозиционной частотной манипуляцией шаг перестройки равен расстоянию между соседними частотами сигнала с многопозиционной частотной манипуляцией. В случае некогерентной демодуляции, а именно этот алгоритм чаще всего используется в системах ППРЧ/МЧМ для соблюдения условия ортогональности сигналов в усиленном смысле, расстояние между соседними частотами выбирается кратным R_b . При

минимальном частотном разnose R_b в пределах всей полосы частот ΔF_s размещаются $M = \Delta F_s / R_b$ рабочих частот или частотных каналов с равномерным шагом. Эти рабочие частоты в одном из вариантов можно разбить на M_C неперекрывающихся частотных групп, каждая из которых занимает полосу шириной $mR_b = mR_C$ (m – кратность частотной манипуляции). Пусть частоты, отображающие символы передаваемого сообщения, располагаются симметрично относительно текущей частоты несущей. Тогда центральные частоты $M_C = 2^k$ частотных групп будут определять множество программно перестраиваемых рабочих частот. Перестройка осуществляется с помощью цифрового синтезатора частот, управляемого k –разрядными сегментами ПСП, формируемой генератором ПСП. Частота передаваемого сигнала в интервале времени τ_0 определяется комбинацией текущей частоты несущей и частоты, отображающей символ передаваемого сообщения. В другом варианте допускается перекрытие соседних частотных групп, содержащих m частот, при минимальном расстоянии между ними равном R_C .

Пусть, как и в предыдущем варианте, центральные частоты частотных групп определяют множество программно перестраиваемых рабочих частот. Количество частот в множестве в этом случае увеличивается с $M_C = M/m$ до $M - (m-1)$, т. е. $M \gg m$ (приблизительно в m раз). Определим значение выигрыша при обработке, обеспечиваемого сигналом с программной перестройкой рабочей частоты.

Если в синтезаторе формируется M частот с шагом ΔF , то занимаемая сигналом с программной перестройкой рабочей частоты ширина полосы частот

$$\begin{aligned} \Delta F_{\overline{F}} &= M\Delta f + 2R_b \text{ (медленная ППРЧ),} \\ \Delta F_{\overline{F}} &= M\Delta f + 2R_C \text{ (быстрая ППРЧ).} \end{aligned} \quad (2.18)$$

В случае медленной перестройки $\Delta F = R_b$ и $\Delta F_C = R_b$, поэтому

$$G_{\text{МППРЧ}} = \Delta F_s / \Delta F_C = (MR_b + 2R_{\overline{F}}) / R_b \quad (M + 2) \approx M. \quad (2.19)$$

В случае быстрой перестройки один и тот же символ сообщения передается на N различных частотах (кратность повторения равна N), поэтому $\Delta F = R_C = R_b N$ и выигрыш

$$G_{\text{БППРЧ}} = \Delta F_s / \Delta F_C = (MR_b + 2R_b N) / R_b \quad (MN + 2N) \approx MN. \quad (2.20)$$

Таким образом, для сигналов с быстрой ППРЧ выигрыш при обработке определяется не только количеством используемых частот M , но и кратностью повторения.

Следует подчеркнуть, что в случае быстрой ППРЧ при определении коэффициента помехозащищенности необходимо учитывать дополнительные энергетические потери, связанные с некогерентным сложением $N = R_C / R_b$ элементов сигнала, передаваемых на различных частотах. Это сложение осу-

ществляется при формировании статистики, по которой производится оценка символа передаваемого сообщения в демодуляторе.

В заключение отметим, что линейную частотную модуляцию сигнала несущей в пределах длительности τ_0 элемента сигнала можно рассматривать как частный случай ППРЧ. Поэтому структурная схема системы связи с использованием ЛЧМ сигналов имеет структурную схему, подобную изображенной на рис. 2.5.

Охарактеризуем теперь некоторые свойства сигналов с расширенным спектром скачками по частоте с точки зрения их использования в системах с множественным доступом.

– **Множественный доступ.** При быстрых скачках частоты один информационный символ передается на нескольких частотах. Если большая часть частот из множества I_F используется для передачи информации единственному абоненту, то принимаемая мощность полезного сигнала будет намного больше помех на этих частотах и соответствующих им временных интервалах и сигнал будет принят верно. При медленных скачках несколько информационных символов передаются на одной частоте. Если вероятность передачи в той же полосе сигналов другим абонентам достаточно мала, сигнал рассматриваемого абонента будет принят верно на большей части временных интервалов скачков частоты. На интервалах, в которых передаются сигналы другим абонентам, можно использовать корректирующие коды, чтобы восстановить данные на этих интервалах времени.

– **Многолучевая интерференция.** При быстрых скачках частоты частота несущего колебания изменяется несколько раз за время передачи одного информационного символа. Эффекты многолучевости на разных частотах часто можно считать независимыми. Поэтому можно ожидать, что на одних частотах уровень принимаемого сигнала может возрасть, в то время как на других частотах могут оказаться глубокие замирания, и наоборот. В приемнике отклики на разных частотах усредняются, что обеспечивает уменьшение влияния многолучевых замираний. При этом обычно используется некогерентное сложение сигналов с разных частот, что дает некоторое улучшение качества приема.

– **Узкополосная помеха.** Предположим, что узкополосный мешающий сигнал попал в одну из мгновенных полос сигнала с расширенным спектром. Если множество I_F содержит N_F частот, то рассматриваемый абонент будет использовать занятую помехой полосу в среднем $1/N_F$ процентов времени. Следовательно, такая помеха будет подавлена примерно в N_F раз.

– **Низкая вероятность перехвата.** При передаче сигнала с быстрыми скачками частоты используется та же плотность мощности на 1 Гц занимаемой полосы, что и при узкополосной передаче без скачков частоты. Но частота, на которой будет передана очередная доля энергии сигнала, неизвестна, а длительность передачи на одной частоте очень мала. Следовательно, хотя такой сигнал легче обнаруживается, чем сигнал с прямым расширением, все же его прием по-прежнему остается трудной задачей.

Кроме перечисленных выше свойств, сигнал с расширением спектра скачками частоты имеет ряд других специфических свойств, которые можно рассматривать либо как преимущества (+), либо как недостатки (–).

(+) Синхронизация для таких сигналов заметно проще, чем для сигналов с прямым расширением спектра. Обычно расширение спектра скачками частоты достигается не за счет высокой частоты скачков, а за счет использования достаточно большого их числа (большого множества I_f) при относительно больших значениях периода скачков T_f , который обычно намного больше длительности τ_0 элементарного символа сигнала с прямым расширением. Следовательно, система со скачками частоты допускает большие значения ошибок синхронизации.

(+) Различные мгновенные полосы частот, которые могут быть заняты элементами сигнала с прыгающей частотой, не обязательно должны быть соприкасающимися, поскольку не возникает никаких проблем при построении синтезатора частот, обеспечивающего быстрый переход от одной частоты к другой, не обязательно принадлежащей соседней полосе. С учетом более устойчивой синхронизации это свойство обеспечивает возможность использования сигналов со значительно большим расширением спектра.

(+) Вероятность передачи многими пользователями одновременно в одной и той же полосе мала. Сигнал, излучаемый АС, находящейся далеко от БС, тем не менее будет принят одновременно с сигналом от другой близкой к БС абонентской станции, так как их сигналы с большой вероятностью будут передаваться на разных частотах. Следовательно, проблема «близкий-далёкий» для данных систем решается намного лучше, чем для систем с прямым расширением.

(+) Поскольку системы с быстрыми скачками частоты допускают больший коэффициент расширения спектра, то подавление узкополосных помех в таких системах оказывается более эффективным, чем в системах с прямым расширением спектра.

(–) Необходимо синтезировать довольно сложный синтезатор частот.

(–) Быстрые изменения уровня сигнала при переключении частоты приводят к увеличению мгновенной полосы частот. Чтобы избежать этого, приходится уровень сигнала уменьшать перед и увеличивать после переключения.

(–) Требуется усиление когерентная демодуляция сигнала, поскольку трудно сохранить необходимые фазовые соотношения при переключениях частот.

2.2.3. Программная перестройка временного положения импульсных сигналов. Характерной особенностью системы связи с программной перестройкой временного положения сигналов, изображенной на рис. 2.6, является импульсный режим передачи. Поэтому для непрерывных источников дискретных сообщений передача осуществляется после предварительного преобразования в пакеты длительностью T_P в пределах цикла длительностью T_F . Пакет занимает определенный временной интервал (слот) в пределах цикла, а номер временного интервала определяется сегментом псевдослучайной по-

следовательности, формируемой ГПСП. Устройство формирования пакетов осуществляет временное сжатие последовательностей символов передаваемого сообщения (трансформацию скорости). При этом скорость передачи по отношению к исходной увеличивается пропорционально коэффициенту сжатия. Отношение T_F / T_P характеризует скажность передачи. Для сохранения энергетического потенциала радиолинии неизменным приходится увеличивать излучаемую мощность пропорционально скажности.

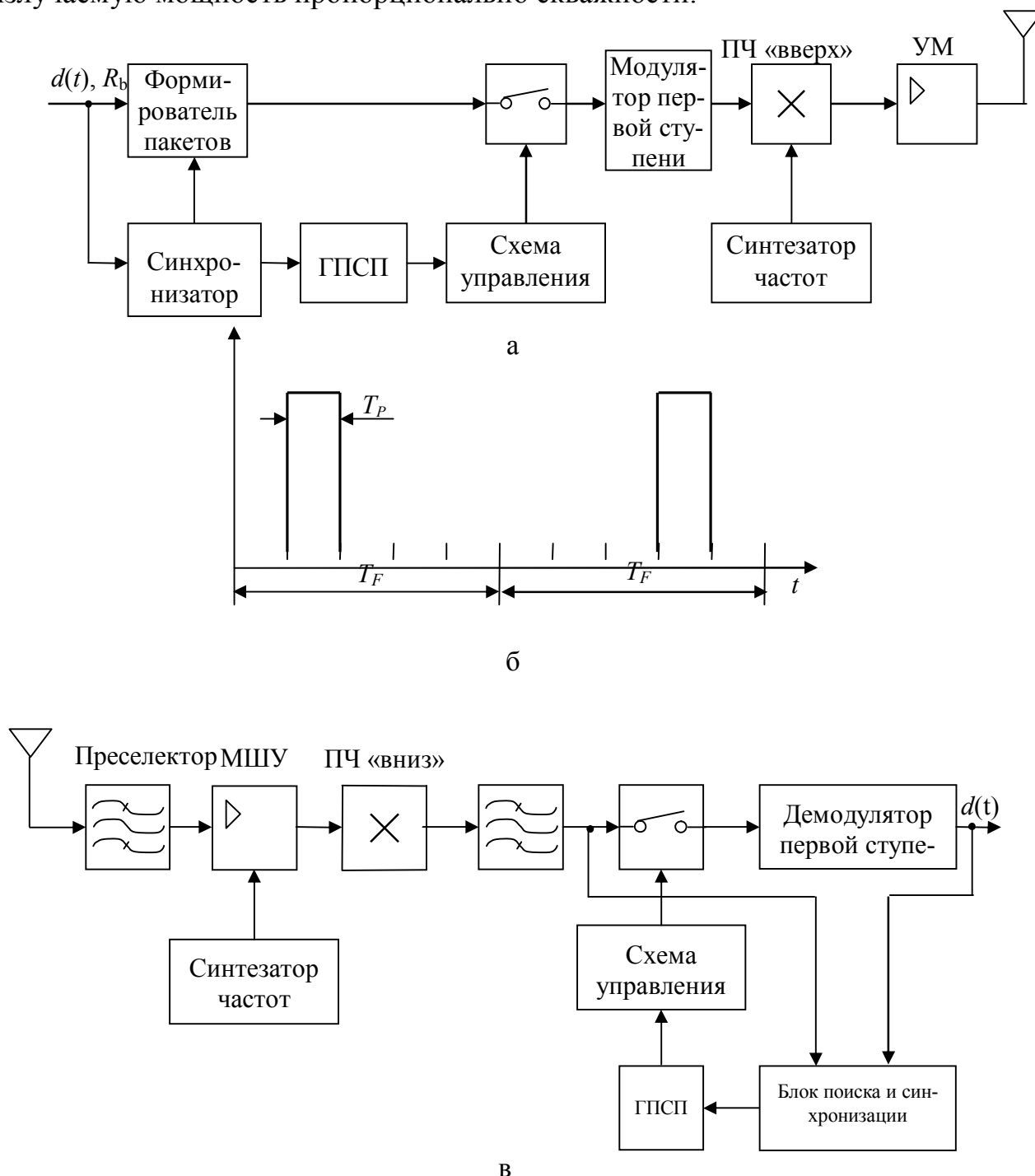


Рис. 2.6. Структурная схема системы связи, использующей сигналы с программной перестройкой временного положения (псевдослучайной время-импульсной модуляцией): а – передающая часть; б – временная диаграмма работы радиолинии; в – приёмная часть

Последовательность импульсных сигналов с программной перестройкой временного положения может быть представлена следующим образом:

$$s(t) = \text{Re} \{ A c(t) \exp[j(2\pi f_0 t + \theta(t) + \theta_0)] \},$$

где

$$c(t) = \sum_{n=-\infty}^{\infty} U_{T_P} \left[t - \left(n + \frac{c_n}{M} \right) T_F \right],$$

$U_{T_P}(t)$ – единичный импульс длительности T_P ; M – количество временных слотов; c_n – псевдослучайное число с равномерным законом распределения в интервале $[1, M]$; $Q(t)$ – изменения фазы, вызванные модуляцией сигналом передаваемого сообщения.

Для обеспечения приема переданных пакетов необходимо, чтобы в приемнике ключ замыкался точно в ожидаемые моменты прихода пакетов. Это достигается путем строгого временного согласования генераторов ПСП передатчика и приемника. При выполнении этого условия ослабление помех происходит в результате временной селекции полезных сигналов. Определим значение выигрыша при обработке, обеспечиваемого при программной перестройке временного положения сигналов. Допустим, что в пределах цикла T_F при наличии M временных слотов в каждом временном слоте необходимо передать k двоичных символов сообщения. Поэтому длительность элемента сигнала

$$T_s = \frac{T_F \log_2 m}{kM}, \quad (2.21)$$

где m – кратность манипуляции.

При фазовой манипуляции несущей и прямоугольной форме импульсов модулирующего сигнала необходимая ширина полосы частот, определяемая по первым нулям спектра, равна

$$\Delta F_s = \frac{2}{\Delta \tau_0} = \frac{2kM}{T_F \log_2 m}. \quad (2.22)$$

Так как ширина полосы частот сигнала передаваемого сообщения равна $\Delta F_d = 1/\tau_0 = R_b$, то при обработке сигналов с программной перестройкой временного положения выигрыш

$$G_{\text{ппвп}} = \frac{\Delta F_s}{\Delta F_d} = \frac{2kM\tau_0}{k\tau_0 \log_2 m} = \frac{2kM}{T_F \log_2 m}. \quad (2.23)$$

Для систем связи с подвижными объектами важными являются свойства рассматриваемых здесь сигналов при их применении для организации множественного доступа.

– **Множественный доступ.** Все абоненты используют всю полосу частот системы. Данный способ расширения спектра уменьшает вероятность использования двумя абонентами одной и той же полосы частот одновременно, так как разным абонентам назначаются разные псевдослучайные коды. Если же иногда два абонента используют для передачи одно и то же окно, то корректирующие коды обеспечат требуемую достоверность передачи информации.

Если между абонентами имеет место синхронизация и назначаемые коды таковы, что в каждом временном окне передачу осуществляет только один абонент, то данная схема расширения спектра приводит к традиционной системе с временным разделением каналов, в которой, однако, используемое абонентом в каждом кадре временное окно не фиксировано, а изменяется от кадра к кадру.

– **Многолучевая интерференция.** При данном способе расширения сигнал передается за более короткое время, т. е. скорость передачи более высокая и, следовательно, при наличии двух лучей соседние символы будут перекрываться так же, как в обычной системе с временным разделением каналов. Поэтому данный способ расширения спектра не обеспечивает каких-либо преимуществ в борьбе с многолучевостью.

– **Узкополосная помеха.** Уменьшение времени приема при данном способе расширения спектра обеспечивается в M раз. Следовательно, во столько же раз обеспечивается уменьшение времени воздействия помехи на приемник и уменьшается ее средняя мощность.

– **Вероятность перехвата.** Частота, на которой абонент передает информацию, постоянна. Однако интервалы времени передачи неизвестны и малы. Поэтому при наличии многих абонентов, одновременно передающих информацию, приемнику перехвата трудно выделить начало и конец передачи одного абонента и решить, какая информация предназначена каждому абоненту.

Кроме указанных свойств, сигналы с расширением спектра скачками по времени обладают рядом других специфических свойств, некоторые из которых можно считать полезными для систем связи с подвижными абонентами (+), а некоторые – нежелательными (–).

(+) Техническая реализация передатчика и приемника при данном способе модуляции проще, чем при расширении спектра скачками по частоте.

(+) Средняя мощность передатчика может быть незначительной, что очень полезно при наличии ограничений на эту мощность и при отсутствии жестких ограничений на пиковую мощность. Передача при данном способе осуществляется очень короткими временными интервалами при значительной мощности излучения.

(+) Проблема «близкий – далекий» для данного способа не является существенной, так как «далекая» АС большую часть времени осуществляет передачу информации в тех временных окнах, в которых «близкая» АС не излучает.

(–) Значительное время вхождения в синхронизм с псевдослучайным кодом БС и малые интервалы времени, в течение которых АС может корректировать синхронизацию.

(–) Если имеют место временные окна, в которых передачу осуществляют несколько абонентов, то теряется большое число бит передаваемой информации, так что приходится использовать коды с хорошей корректирующей способностью и глубокое перемежение.

2.3. Псевдослучайные бинарные последовательности (ПСП)

2.3.1. Требования, предъявляемые к ПСП. Псевдослучайной бинарной последовательностью длительностью τ_0 называют последовательность, сформированную по определенным правилам из дискретных элементов 0 и 1 так, чтобы ее корреляционные свойства были близки к соответствующим свойствам шумовой реализации такой же длительности. Если обозначить длительность элемента последовательности через $\Delta\tau_0$, то число таких элементов в последовательности с длительностью τ_0 равно

$$M = \tau_0 / \Delta\tau_0 = B \gg 1. \quad (2.24)$$

Такие последовательности в литературе часто называют M -последовательностями, а величину M – длиной последовательности.

В общем случае к ПСП, используемым для расширения спектра сигналов, предъявляются следующие требования:

- большой объем ансамбля последовательностей, формируемых с помощью единого алгоритма;
- «хорошие» авто- и взаимно-корреляционные свойства последовательностей, входящих в состав ансамбля;
- сбалансированность структуры, то есть число единиц и нулей в ней должно отличаться не более чем на один символ;
- максимальный период для заданной длины регистра сдвига, формирующего последовательность;
- непредсказуемость структуры последовательности по ее неискаженному сегменту ограниченной длины.

В соответствии с алгоритмами формирования различные ПСП можно классифицировать на линейные, нелинейные, комбинированные и каскадные.

2.3.2. Получение псевдослучайных последовательностей. Наиболее известны и хорошо исследованы M -последовательности, формируемые (генерируемые) двоичным регистром сдвига с логической обратной связью. Такой регистр представляет собой m последовательно соединенных триггерных ячеек, управляемых периодической последовательностью тактовых импульсов, вырабатываемых тактовым генератором, и импульсами, формируемыми в устройстве логической обратной связи.

Упрощенная структурная схема регистра показана на рис. 2.7. Рассмотрим работу этой схемы. Каждое плечо триггера (T) может принимать два со-

стояния: закрытое и открытое (которые условно можно обозначать символами 0 и 1 или наоборот). Состоянием триггера будем называть состояние одного из его плеч (например, левого). Обозначим начальные состояния триггеров регистра через $\gamma_{01}, \gamma_{02}, \dots, \gamma_{0j}, \dots, \gamma_{0m}$. Ключи $K1$ и $K2$ при этом считаем разомкнутыми. Пусть в момент t_0 ключ $K1$ замыкается и на все триггерные ячейки начинают поступать импульсы от генератора тактовых импульсов (ГТИ). Под действием этих импульсов триггеры изменяют свое состояние на противоположное или сохраняют прежнее состояние в зависимости от того, какое начальное состояние имел данный триггер и триггер, находящийся перед ним. В результате каждый триггер последовательно принимает все состояния предшествующих ему триггеров. Если снимать напряжение с последнего триггера, то оно последовательно принимает состояния $\gamma_{0m}, \gamma_{0(m-1)}, \dots, \gamma_{0j}, \dots, \gamma_{01}$, т. е. представляет последовательность, состоящую из m элементов, соответствующих символам 0 и 1. После m тактовых импульсов напряжение на выходе последней триггерной ячейки не меняется и она принимает состояние γ_{01} .

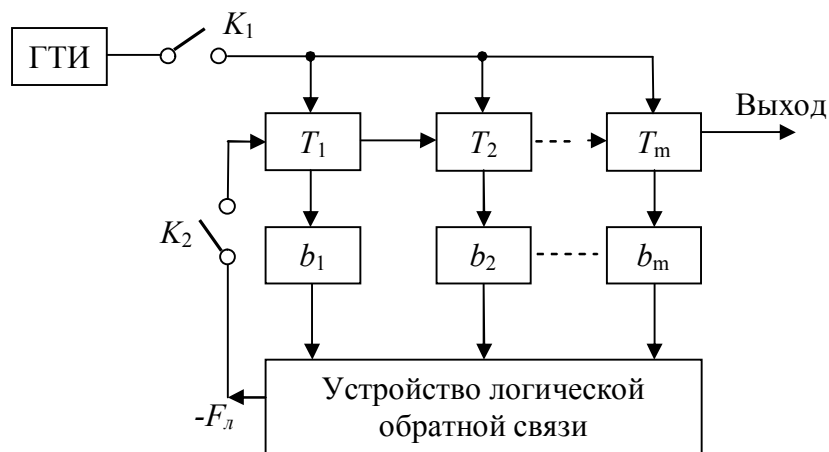


Рис. 2.7. Регистр сдвига с логической обратной связью

Таким образом, если регистр не имеет обратной связи (ключ $K2$ разомкнут), то под действием тактовых импульсов на его выходе образуется только одна кодовая комбинация из m элементов, соответствующих начальным состояниям триггерных ячеек. В данном случае регистр представляет собой устройство памяти, в котором записана некоторая кодовая комбинация, которая может быть выведена (считана) в нужное время.

Чтобы регистр формировал дискретную последовательность непрерывно, т. е. работал как генератор такой последовательности, необходимо ввести обратную связь – замкнуть ключ $K2$. В этом случае в устройстве логической обратной связи формируется функция

$$F_L(\gamma_1, \gamma_2, \dots, \gamma_j, \dots, \gamma_m) = \sum_{i=1}^m b_i \gamma_i \pmod{2}, \quad (2.25)$$

которая называется логической функцией обратной связи. В (2.25) коэффициенты b_J могут принимать значения 1 или 0 в зависимости от того, какие ячейки триггеров участвуют в формировании функции F_L , определяющей обратную связь. Значения γ_J определяются фактическими состояниями триггерных ячеек в процессе работы.

Получение логической функции сводится к выбору совокупности коэффициентов b_J определяющих, какие ячейки триггеров должны быть связаны с устройством логической обратной связи. Коэффициенты b_J выбираются в соответствии с определенным правилом, которое позволяет получить последовательность максимальной длины для регистра с заданным числом ячеек. Это правило может быть установлено на основе хорошо разработанной теории рекуррентных последовательностей, которое задаётся характеристическим многочленом

$$f(x) = b_0 x^m + b_1 x^{m-1} + b_2 x^{m-2} + \dots + b_{m-1} x^1 + b_m \quad (2.26)$$

с $b_0 = 1$ и $b_m = 1$. Значения вектора $b = [b_0 \ b_1 \ b_2 \ \dots \ b_m]$ полностью определяет структуру автомата формирования ПСП: если коэффициент $b_J = 0$, то это означает, что выход ячейки с номером J к цепи обратной связи не подключён; при $b_J = 1$ J -й выход подключён. Например, пусть $m = 7$ и $b_1 = [10100111]$, $b_2 = [11000001]$, т.е. Характеристические многочлены соответствующих двух последовательностей имеют вид

$$f_1(x) = 1 \cdot x^7 + 0 \cdot x^6 + 1 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 1 = x^7 + x^5 + x^2 + x^1 + 1;$$

$$f_2(x) = 1 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + \cdot x^1 + 1 = x^7 + x^6 + 1.$$

Цифровой автомат, формирующий M -последовательность с характеристическим многочленом $F_1(x)$, имеет отводы с выходов ячеек 1, 2 и 5, а с характеристическим многочленом $F_2(x)$ – ячейки 6. Обе последовательности имеют максимальную длину $N = 2^7 - 1 = 127$.

В табл. 2.1 приведены некоторые данные, касающиеся количества и номеров отводов генераторов M -последовательностей, для различного числа разрядов регистра сдвига. Как следует из данных этой таблицы, с увеличением количества разрядов m количество M -последовательностей возрастает. Сформированная в соответствии с логической функцией дискретная последовательность символов 0 и 1 подается на вход первой триггерной ячейки регистра.

В результате действия тактовых импульсов и импульсов обратной связи регистр окажется в режиме непрерывной смены состояний, т.е. будет генерировать дискретную последовательность элементов. Общее число различных состояний, которые последовательно примет регистр сдвига, определяется числом триггерных ячеек m и тем, что каждая из них может находиться в одном из двух возможных состояний. Очевидно, что число различных состояний регистра (исключая нулевое состояние) равно

$$M = 2^m - 1. \quad (2.27)$$

Таблица 2.1

Количество разрядов (m)	Период (длина) M -последовательности	Количество M -последовательностей	Номера отводов регистра для цепи обратной связи
2	3	1	[2, 1]
3	7	2	[3, 2]; [3, 1]
4	15	2	[4, 3]; [4, 1]
5	31	6	[5, 3]; [5, 2]
6	63	6	[6, 5]; [6, 1]
7	127	18	[7, 6]; [7, 3]; [7, 1]
8	255	16	[8, 6, 5, 4]; [8, 6, 5, 3]
9	511	48	[9, 5]; [9, 6, 4, 3]
10	1023	60	[10, 7]; [10, 3]
11	2047	176	[11, 9]; [11, 8, 5, 2]
12	4095	144	[12, 6, 4, 1]
13	8191	630	[13, 4, 3, 1]
14	16383	756	[14, 5, 3, 1]
15	32767	1800	[15, 14]; [15, 14]
16	65535	2048	[16, 15, 13, 4]
17	131071	7710	[17, 14]; [17, 13]
18	262143	7776	[18, 11]; [18, 7]
19	524287	27594	[19, 6, 2, 1]
20	1048575	24000	[20, 17]; [20, 3]

Следовательно, после каждых M разных состояний регистр, находящийся в режиме генерирования, начинает повторять эти состояния. Иными словами, регистр генерирует дискретную последовательность элементов, структура которой периодически повторяется, через каждые M элементов. Величина M поэтому называется периодом последовательности.

Регистр с обратной связью, образованной суммированием по «модулю два» состояний некоторых заранее выбранных триггерных ячеек, называется линейным регистром сдвига или генератором псевдослучайной последовательности импульсов. Генерируемую последовательность можно снимать не только с выхода последнего триггера, но и с любого другого триггера. При этом последовательность имеет ту же структуру и отличается только сдвигом по времени.

На рис. 2.8 приведена структурная схема M -последовательности, соответствующая характеристическому многочлену $f(x) = x^3 + x^2 + 1$.

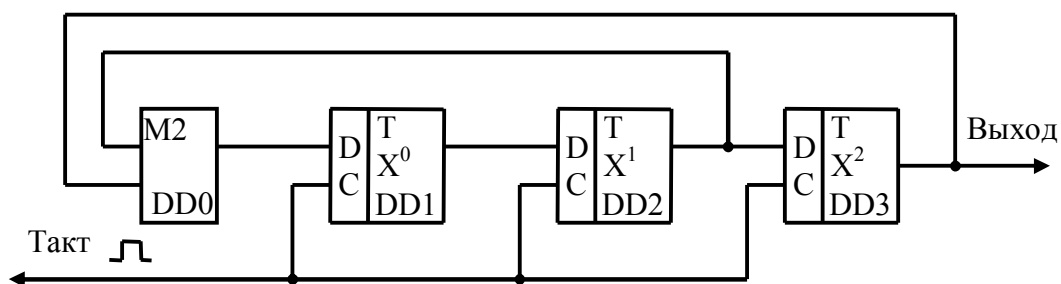


Рис. 2.8. Схема генератора с $f(x) = x^3 + x^2 + 1$

Состояние ячеек регистра при начальных условиях 100 приведено в табл. 2.2.

Таблица 2.2.

Процесс формирования $M = 7$ последовательности

Номер такта	Состояние ячеек			Выход		Номер такта	Состояние ячеек			Выход
	DD1	DD2	DD3				DD1	DD2	DD3	
0	0	0	1			6	0	1	1	1
1	1	0	0	1		7	1	1	1	1
2	0	1	0	0		8	1	0	0	1
3	1	0	1	0		9	0	1	0	0
4	1	1	0	1		10	1	0	1	0
5	1	1	1	0		•	•	•	•	•

Как следует из табл. 2.2 при указанных начальных условиях формируется последовательность, снятая с выхода ячейки DD3 $\gamma = (1001011, 1001011, 100\dots)$ с периодом $M = 7$.

Необходимым условием получения m -последовательности с помощью характеристического многочлена $F(x)$ является его неприводимость. Многочлен $F(x)$ степени m называется неприводимым, если он не может быть разложен на многочлены-сомножители меньшей степени. Например, многочлен $f(x) = x^5 + x + 1$ является приводимым, так как $x^5 + x + 1 = (x^3 + x^2 + 1)(x^2 + x + 1)$. Если $2^m - 1$ является простым числом, то неприводимый многочлен порождает M -последовательность.

Неприводимый многочлен $F(x)$ степени m называется примитивным, если период коэффициентов $1/F(x)$ равен 2^m-1 . Примитивность многочлена $F(x)$ является необходимым и достаточным условием получения М-последовательности. Примитивные многочлены существуют для всех $m>1$. Их количество определяется следующим выражением [3]:

$$N_p(m) = \frac{\Phi_p(L)}{m} = \frac{1}{m} \prod_{i=1}^k (p_i - 1) \cdot p_i^{m_i-1}, \quad (2.28)$$

где $\Phi_p(L)$ – функция Эйлера, определяющая количество целых чисел, взаимно простых и не превышающих L ; p_i – сомножители чисел 2^m-1 , т. е.

$$2^m - 1 = \prod_{i=1}^k p_i^{m_i}; \quad m_i - \text{целые числа.}$$

Например, при $m = 6$ имеем $L = 2^6-1 = 63$. Это число может быть представлено в виде произведения $3 \cdot 3 \cdot 7 = 3^2 \cdot 7$, причем $p_1 = 3$; $m_1 = 2$, $p_2 = 7$, $m_2 = 1$. Поэтому количество примитивных многочленов $N_p(m) = (1/6)[(3^{2-1})(3-1)] \times [(7^{1-1})(7-1)] = 6$. Если L может быть представлено в виде произведения не-кратных сомножителей, т. е. $m_i = 1$, то выражение (2.28) принимает вид

$$N_p(m) = \frac{1}{n} \prod_{i=1}^k (p_i - 1). \quad (2.29)$$

Например, при $m=8$ имеем $L=255=3 \cdot 5 \cdot 17$ и $N_p(m) = (1/8)(3-1)(5-1)(17-1) = 16$.

Если L – простое число, то количество примитивных многочленов $N_p(m) = 2(2^{m-1} - 1)/m$.

Например, при $m = 7$ имеем $L = 127$ и $N_p(m) = 2(64-1)/7 = 18$. Из этих 18 многочленов половина являются зеркальными по отношению к другой половине. Зеркальный многочлен степени m по отношению к исходному определяется с помощью выражения

$$f(x) = x^m f(x^{-1}). \quad (2.30)$$

Например, для $f(x) = x^4 + x + 1$

$$f_{\min}(x) = x^4(x^{-4} + x^{-1} + 1) = x^4 + x^3 + 1.$$

В табл. 2.3 приведены некоторые неприводимые многочлены до десятой степени, а в табл. 2.4 указаны периоды М-последовательностей.

Таблица 2.3

Неприводимые многочлены и их эквиваленты

Сте- пень	Многочлен	Двоичная по- следователь- ность	Сте- пень	Многочлен	Двоичная по- следователь- ность
1	$x+1$	11	7	x^7+x+1	100000011
2	x^2+x+1	111		x^7+x^3+1	10001001
3	x^3+x+1	1011		$x^7+x^3+x^2+x+1$	10001111
	x^3+x^2+1	1101		$x^7+x^4+x^2+x+1$	10011101
4	x^4+x+1	10011		$x^7+x^5+x^2+x+1$	10100111
	x^4+x^3+1	11001		$x^7+x^5+x^3+x+1$	10101011
5	x^5+x^2+1	100101		$x^7+x^6+x^3+x+1$	11001011
	x^5+x^3+1	101001	8	$x^7+x^6+x^4+x+1$	11010011
	$x^5+x^3+x^2+x+1$	101111		$x^8+x^4+x^3+x+1$	100011011
	$x^5+x^4+x^2+x+1$	110111		$x^8+x^4+x^3+x^2+1$	100011101
	$x^5+x^4+x^3+x+1$	111011		$x^8+x^5+x^3+x+1$	100101011
	$x^5+x^4+x^3+x^2+1$	111101		$x^8+x^5+x^3+x^2+1$	100101101
6	x^6+x+1	1000011	9	$x^8+x^6+x^5+x^2+1$	101100101
	$x^6+x^4+x^2+x+1$	1010111		$x^8+x^7+x^3+x+1$	110001011
	$x^6+x^4+x^3+x+1$	1011011		$x^8+x^7+x^5+x^3+1$	110101001
	x^6+x^5+1	1100001		x^9+x+1	1000000011
	$x^6+x^5+x^2+x+1$	1100111		x^9+x^4+1	1000010001
	$x^6+x^5+x^3+x^2+1$	1101101		$x^9+x^4+x^2+x+1$	1000010111
	$x^6+x^5+x^4+x+1$	1110011		$x^9+x^4+x^3+x+1$	1000011011
				$x^9+x^5+x^4+x+1$	1000110011
			10	$x^{10}+x^3+1$	1001100101
					10000001001

Таблица 2.4

Периоды M -последовательностей различной длины с тактовой частотой следования 1 МГц

Регистр длины m	Длина последовательности	Длительность периода последовательности
7	127	$1,27 \cdot 10^{-4}$ с
8	255	$2,55 \cdot 10^{-4}$ с
9	511	$5,11 \cdot 10^{-4}$ с
10	1 023	$1,023 \cdot 10^{-3}$ с
11	2 047	$2,047 \cdot 10^{-3}$ с
12	4 095	$4,095 \cdot 10^{-3}$ с
13	8 191	$8,191 \cdot 10^{-3}$ с
17	131 071	$1,31 \cdot 10^{-1}$ с
19	524 287	$5,24 \cdot 10^{-1}$ с
23	8 388 607	8,388 с
27	134 217 727	13,421 с
31	2 147 483 647	35,8 мин
43	879 609 302 207	101,7 дня
61	2 305 843 009 213 693 951	$7,3 \cdot 10^4$ лет
89	618 970 019 642 690 137 449 562 111	$1,95 \cdot 10^9$ лет

2.3.3. Некоторые свойства М-последовательностей. Рассмотрим в данном пункте только важнейшие свойства, которые приведены в [3].

Балансное свойство. Каждая М-последовательность содержит 2^{n-1} символов 1 и 2^{n-1} символов 0 или число единиц и нулей в ней должно отличаться не более чем на один символ. Это требование важно для исключения постоянной составляющей информационного сигнала.

Свойство полноты состояний. Состояние разрядов регистра сдвига, формирующего М-последовательность, можно представить полным набором m -разрядных двоичных чисел за исключением числа, содержащего нули во всех разрядах. Состояние «все нули» является запрещенным.

Свойство серий. В периоде М-последовательности половина серий имеет длину 1, одна четверть – длину 2, одна восьмая – длину 3 и так до тех пор, пока это продолжение имеет смысл. Под серией здесь понимается набор следующих друг за другом одинаковых символов 0 или 1. Это свойство проиллюстрировано табл. 2.5. Как следует из данных таблицы, исключение составляют серии, длина которых равна n и $(n-1)$.

Таблица 2.5

Свойства серий		
Длина серии	Количество серий единиц	Количество серий нулей
n	1	0
$n-1$	0	1
$n-2$	1	1
$n-3$	2	2
$n-4$	4	4
...	...	...
2	2^{n-4}	2^{n-4}
1	2^{n-3}	2^{n-3}

Свойство циклического сдвига при сложении. Сложение по mod2 m -последовательности и некоторого ее циклического сдвига дает в результате другой циклический сдвиг той же самой последовательности. Это свойство иллюстрируется на рис. 2.9.

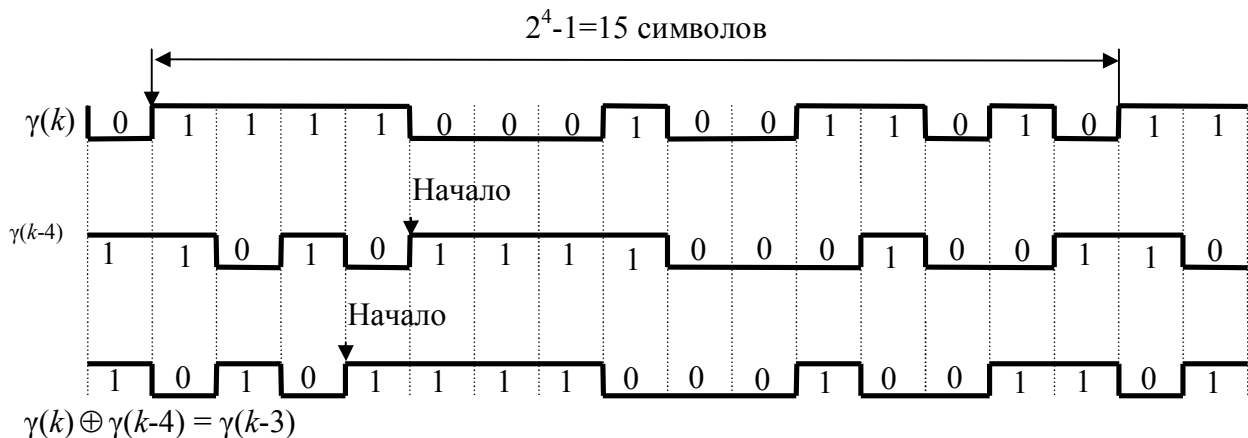


Рис. 2.9. Иллюстрация свойства циклического сдвига при сложении М-последовательности

Свойство децимации. Последовательность, образованная из взятых через один символ исходной M -последовательности, по структуре совпадает с исходной, но имеет в два раза ниже тактовую частоту. Это свойство иллюстрируется на рис. 2.10.

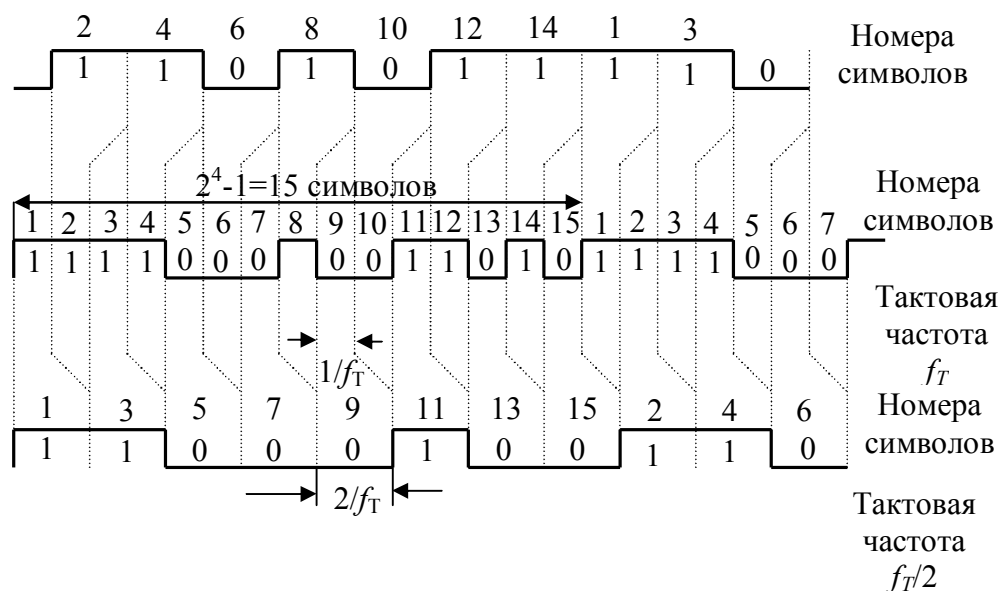


Рис. 2.10. Иллюстрация свойства децимации M -последовательности

Здесь средняя последовательность представляет собой m -последовательность с тактовой частотой F_T и периодом, равным 15. Верхняя последовательность образована из четных символов исходной m -последовательности с увеличением их длительности вдвое. Нижняя последовательность образована из нечетных символов исходной m -последовательности с увеличением их длительности вдвое. Нетрудно убедиться, что эти последовательности являются циклическими сдвигами исходной m -последовательности, но с тактовой частотой $F_T/2$. Сдвиг между ними равен 7,5 тактовым интервалам или половине длины последовательности. Этот принцип может быть распространен для индексов децимации более высокого порядка. Например, при индексе децимации R (R является степенью 2) можно получить R подпоследовательностей с тактовой частотой F_T/R . При четном индексе децимации R , но не равным степени 2, и если длина исходной m -последовательности есть простое число, то исходная m -последовательность может быть децимирована на R подпоследовательностей, являющихся ее зеркальными отображениями.

Корреляционные свойства. Корреляционные свойства кодовых последовательностей в широкополосных системах зависят от типа кодовой последовательности, ее длины, частоты следования символов и посимвольной структуры.

В общем виде автокорреляционная функция (АКФ) определяется интегралом

$$\Psi(\tau) = \int_0^{\tau_0} f(t)f(t-\tau)dt \quad (2.31)$$

и показывает связь сигнала со своей копией, смещенной во времени на величину τ .

Изучение АКФ играет важную роль при выборе кодовых последовательностей с точки зрения наименьшей вероятности установления ложной синхронизации.

Взаимокорреляционная функция (ВКФ) имеет большое значение для систем с кодовым разделением абонентов и отличается от АКФ только тем, что под знаком интеграла стоят разные функции:

$$\Psi(\tau) = \int f(t)g(t - \tau)dt. \quad (2.32)$$

ВКФ показывает, таким образом, степень соответствия одной кодовой последовательности другой. Чтобы упростить понятия АКФ и ВКФ, можно представить значение той или иной функции как разность между числом совпадений A и несовпадений B символов кодовых последовательностей при их посимвольном сравнении. Для иллюстрации данного примера рассмотрим автокорреляционную функцию кодовой последовательности Баркера длиной 11 чипов, имеющей следующий вид:

1 1 1 0 0 0 1 0 0 1 0.

Посимвольное сравнение этой последовательности с ее копией сведем в табл. 2.6.

Таблица 2.6

Сравнение автокорреляционной функции кодовой последовательности Баркера
(длиной 11 чипов) с её копией

Значение сдвига	Последовательность	Число совпадений A	Число несовпадений B	Значение разности
1	01110001001	5	6	-1
2	10111000100	5	6	-1
3	01011100010	5	6	-1
4	00101110001	5	6	-1
5	10010111000	5	6	-1
6	01001011100	5	6	-1
7	00100101110	5	6	-1
8	00010010111	5	6	-1
9	10001001011	5	6	-1
10	11000100101	5	6	-1
0	11100010010	11	0	11

Графическое изображение АКФ данной последовательности Баркера показано на рис. 2.11, а. Ее можно назвать идеальной, поскольку отсутствуют боковые пики, которые могли бы способствовать ложному обнаружению сигнала.

В качестве негативного примера рассмотрим любую произвольную кодовую последовательность, например

1 1 1 0 0 0 1 1 1 0 0.

Проведя соответствующие предыдущему примеру вычисления, получим следующее графическое изображение автокорреляционной функции, приведенное на рис. 2.11, б. Боковые пики величиной 7 и 3 единиц могут привести к ложному срабатыванию системы в случае применения такой последовательности для распределения сигнала.

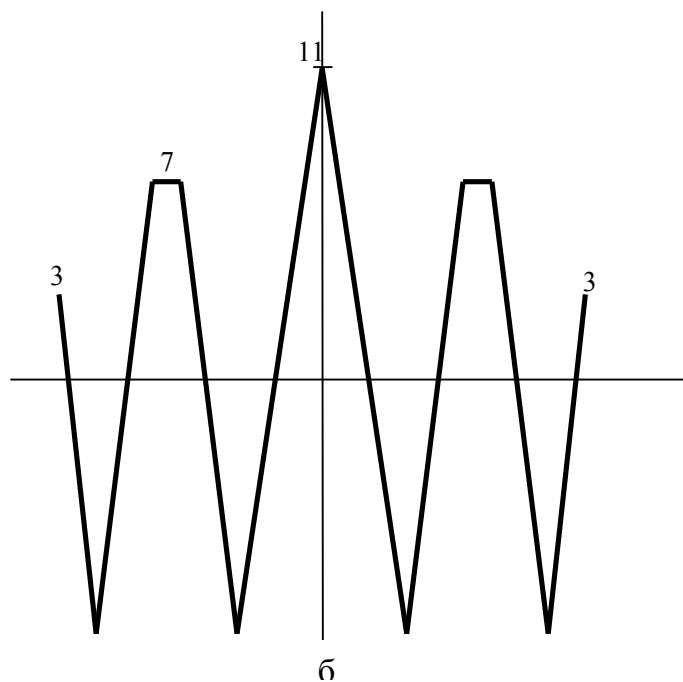
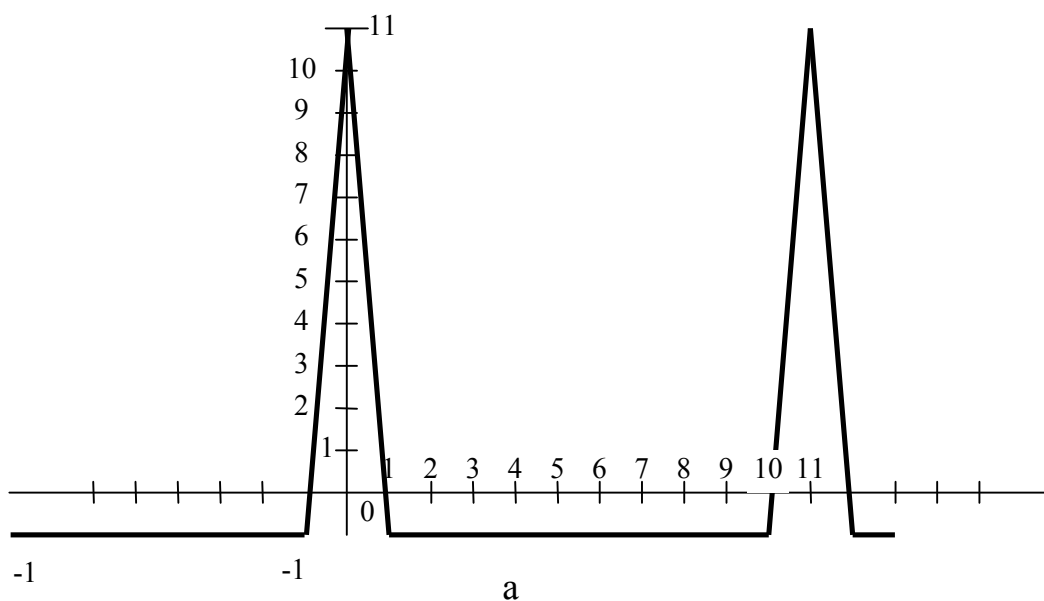


Рис. 2.11. Автокорреляционная функция:
а – последовательности Баркера; б – произвольной кодовой комбинации

На рис. 2.12 приведена взаимокорреляционная функция M -последовательностей $F(t) = 1\ 0\ 0\ 1\ 0\ 1\ 1$ и $G(t) = 1\ 1\ 1\ 0\ 1\ 0\ 0$. Посимвольное сравнение M -последовательностей $F(t)$ и $G(t-\tau)$ сведено в табл. 2.7.

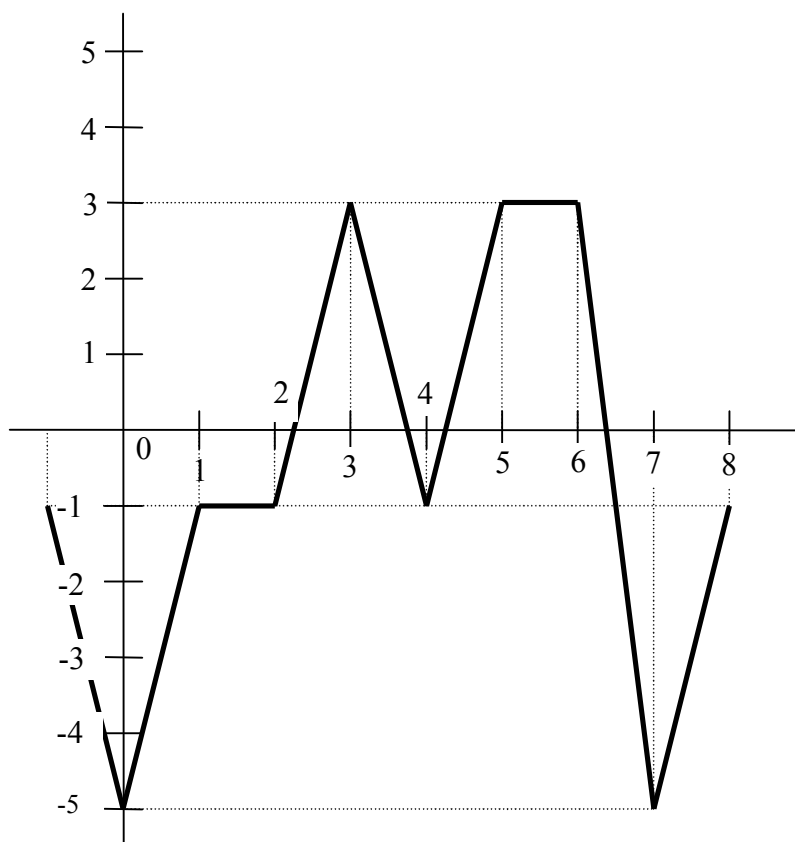


Рис. 2.12. Взаимокорреляционная функция последовательностей $F(t) = 1\ 0\ 0\ 1\ 0\ 1\ 1$ и $G(t) = 1\ 1\ 1\ 0\ 1\ 0\ 0$

Таблица 2.7

Вычисление взаимокорреляционной функции последовательностей $F(t) = 1\ 0\ 0\ 1\ 0\ 1\ 1$ и $G(t) = 1\ 1\ 1\ 0\ 1\ 0\ 0$

Значение сдвига	Последовательность	Число совпадений A	Число несовпадений B	Значение разности
0	1 1 1 0 1 0 0	1	6	-5
1	0 1 1 1 0 1 0	3	4	-1
2	0 0 1 1 1 0 1	3	4	-1
3	1 0 0 1 1 1 0	5	2	+3
4	0 1 0 0 1 1 1	3	4	-1
5	1 0 1 0 0 1 1	5	2	+3
6	1 1 0 1 0 0 1	5	2	+3

Спектральные свойства. Спектр биполярного псевдослучайного сигнала содержит постоянную и дискретные спектральные составляющие, следующие через интервал $\Delta f = 1/(L\Delta\tau_0) = 1/((2^m - 1)\Delta\tau_0)$. Огибающая дискретных спектральных составляющих определяется функцией $(\sin x/x)^2$. Поэтому амплитуда спектральных составляющих равна нулю на частотах $F = k\Delta\tau_0$. Спектральная плотность биполярного сигнала изображена на рис. 2.13. Путем увеличения периода M -последовательности можно уменьшить интервал между спектральными составляющими, сделав спектр практически сплошным. При этом спектральная плотность в пределах полосы частот, равной $2/\Delta\tau_0$ становится почти равномерной. Эти свойства спектральной плотности псевдослучайного сигнала позволяют его широко использовать в системах связи в качестве тестового и для получения аналогового шумового процесса с характеристиками, близкими к гауссовскому шуму [3].

Криптостойкость. Структура M -последовательности легко может быть раскрыта по ее неискаженному сегменту, содержащему $2m$ символов. Действительно, каждый символ сегмента удовлетворяет линейному рекуррентному соотношению (2.25) поэтому можно записать следующую систему линейных уравнений:

$$m \left\{ \begin{array}{l} \gamma_j = \gamma_{j-1}b_1 + \gamma_{j-2}b_2 + \dots + \gamma_{j-n}b_n \\ \gamma_{j+1} = \gamma_jb_1 + \gamma_{j-1}b_2 + \dots + \gamma_{j-n+1}b_n \\ \gamma_{j+2} = \gamma_{j+1}b_1 + \gamma_jb_2 + \dots + \gamma_{j-n+2}b_n \\ \dots\dots\dots \\ \gamma_{j+n} = \gamma_{j+n-1}b_1 + \gamma_{j+n-2}b_2 + \dots + \gamma_{j-n}b_n \end{array} \right. \quad (2.33)$$

$\xleftarrow{\qquad m \qquad} \xrightarrow{\qquad}$

относительно неизвестных коэффициентов $b_1, b_2, \dots, b_m$. Эта система уравнений может быть решена с помощью так называемого алгоритма Берлекампа-Мэсси [5] и в результате решения найдены номера отводов регистра сдвига, участвующих в формировании сигнала обратной связи.

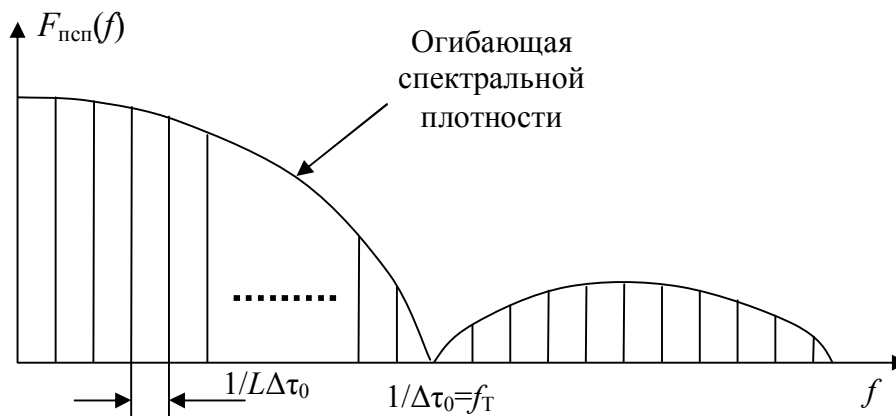


Рис. 2.13. Спектральная плотность двухуровневого псевдослучайного сигнала

Проиллюстрируем возможность решения системы (2.33) на следующем примере. Предположим, что получена последовательность $\{01100100\}$ и известно, что период последовательности равен 15. Таким образом, имеем систему из четырех уравнений

$$0 = 0 \cdot b_1 + 1 \cdot b_2 + 1 \cdot b_3 + 0 \cdot b_4, \quad (1)$$

$$1 = 0 \cdot b_1 + 0 \cdot b_2 + 1 \cdot b_3 + 1 \cdot b_4, \quad (2)$$

$$0 = 1 \cdot b_1 + 0 \cdot b_2 + 0 \cdot b_3 + 1 \cdot b_4, \quad (3)$$

$$0 = 0 \cdot b_1 + 1 \cdot b_2 + 0 \cdot b_3 + 0 \cdot b_4. \quad (4)$$

Сложением уравнений (1) и (4) получаем $b_3 = 0$. При подстановке $b_3 = 0$ в уравнение (1) имеем $b_2 = 0$. При подстановке $b_2 = 0$, $b_3 = 0$ в уравнение (4) получаем $b_4 = 0$, а при подстановке $b_2 = 0$, $b_3 = 0$, $b_4 = 1$ в уравнение (3) имеем $b_1 = 1$. В результате определяем характеристический многочлен M -последовательности: $f(x) = x^4 + x + 1$.

2.3.4. Ввод исходной информационной последовательности в регистр сдвига. Рассмотрим теперь работу линейного регистра сдвига, когда необходимо получить псевдослучайную последовательность импульсов, однозначно соответствующую исходной двоичной информационной последовательности символов.

Выше указывалось, что перед началом работы регистра необходимо задать начальные условия в его ячейках, т.е. ввести в регистр некоторую начальную комбинацию символов. Эта комбинация определяет начало последовательности и ее внутреннюю структуру в пределах периода. Если перед началом каждого периода M -последовательности изменять начальные условия, то структура внутри каждого периода M -последовательности также будет изменяться. При передаче двоичной цифровой информации начальные условия должны изменяться в соответствии с исходной информационной последовательностью двоичных символов.

Один из возможных вариантов ввода исходной информационной последовательности в регистр сдвига приведен на рис. 2.14. Двоичная информационная последовательность символов $\{x_i\}$, каждый из которых имеет длительность τ_0 , поступает на формирующее устройство (ФУ), вырабатывающее последовательность коротких импульсов ($\tau_{\Phi} \ll \tau_0$), которые задают начальные условия во всех триггерных ячейках регистра. Во время задания начальных условий, соответствующих исходному информационному символу, ключевая схема (КС) закрыта и тактовые импульсы на регистр не поступают. Через некоторое время, определяемое линией задержки (ЛЗ) и необходимое для записи начальных условий, с выхода устройства формирования строба (УФС) на ключевую схему подается строб, открывающий эту схему для тактовых импульсов, и регистр начинает вырабатывать последовательность импульсов. В момент окончания строба ключевая схема закрывается, и генерирование последовательности прекращается (регистр «останавливается»). После этого

вводятся начальные условия, соответствующие следующему информационному символу, и схема начинает работать так же, как описано выше.

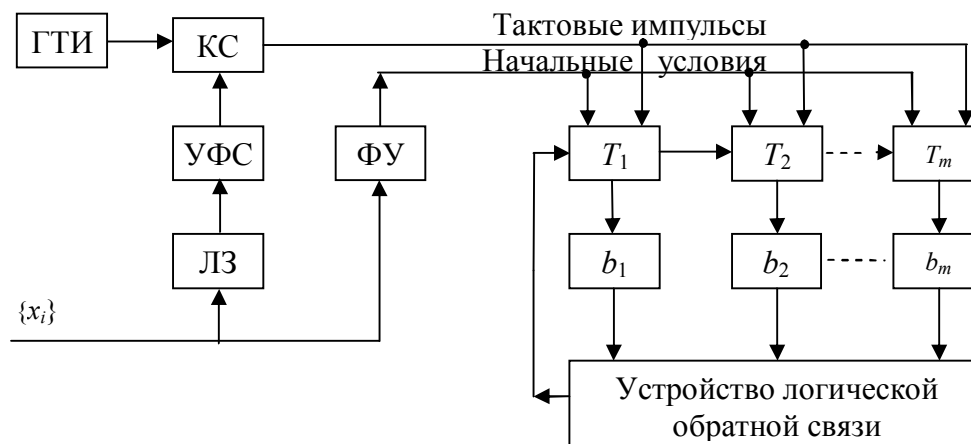


Рис. 2.14. Формирователь псевдослучайной последовательности в соответствии с передаваемым сообщением

Проведенное рассмотрение показывает, что линейный регистр сдвига преобразует информационную последовательность простых двоичных символов в последовательность сложных двоичных символов. При этом каждому простому информационному символу 0 или 1 однозначно соответствует сложный символ 0 или 1, представляющий собой определенную комбинацию из M двоичных элементов. Если правило ввода начальных условий изменить (например, поменять некоторые плечи триггеров, на которые задаются начальные условия; не подавать начальные условия на некоторые триггеры и т. п.), то исходным символам 0 и 1 станут соответствовать сложные символы 0 и 1 с другой внутренней структурой. Нетрудно видеть, что в зависимости от задаваемых начальных условий одну и ту же исходную информационную последовательность простых двоичных символов регистр сдвига будет преобразовывать в последовательность с различными сложными двоичными символами.

Так как в зависимости от начальных условий регистр сдвига может принимать M различных состояний, то, очевидно, каждому простому символу 0 или 1 можно поставить в соответствие один из M сложных символов 0 или 1. Эта особенность работы регистра сдвига открывает ряд возможностей для решения различных прикладных задач.

В заключение этого параграфа следует отметить, что для формирования ШПС широко используются последовательности: Баркера, Гоулда (G -последовательности), Касами (k -последовательности), двойственные коды БЧХ (B -последовательности), Уолша и генерируемые с помощью явления динамического хаоса. Необходимые сведения можно получить из [3, 6, 7].

2.4. Методы приёма сложных сигналов

Чтобы положительные свойства сложных сигналов проявлялись наиболее полно, необходимо применять способы приёма, близкие к оптимальным.

Рассмотрим сначала особенности оптимального приёма сложных сигналов, а затем, опираясь на результаты такого рассмотрения, дадим краткую характеристику особенностей реальных методов приёма. При этом для простоты ограничим рассмотрение случаем сложных равновероятных двоичных сигналов.

В зависимости от уровня априорных сведений о параметрах принимаемого сигнала (начиная от сигнала, известного точно и кончая сигналом со случайными параметрами) возможен ряд методов оптимального приёма сложных сигналов на фоне нормальной флуктуационной помехи типа белого шума. Остановимся на двух наиболее важных методах.

2.4.1. Оптимальный когерентный прием. В соответствии с теорией оптимального приема равновероятных двоичных сигналов, известных точно, правило работы приемника определяется выражением

$$\int_0^{\tau_0} y(t) s_{1cl}(t) dt \begin{matrix} > \\ < \end{matrix} \int_0^{\tau_0} y(t) s_{2cl}(t) dt \quad \begin{matrix} x_i = x_1, \\ x_i = x_2, \end{matrix} \quad (2.34)$$

где $S_{1cl}(t)$ и $S_{2cl}(t)$ – двоичные сложные сигналы, соответствующие передаваемым символам x_1 и x_2 ; $Y(t)$ – принимаемый сигнал; τ_0 – длительность передаваемого символа (см. рис. 2.1).

Из рассмотрения правила (2.34) следует, что в качестве «образцов» на приемной стороне системы связи необходимо располагать такими же сложными сигналами, которые используются для передачи. Процедура оптимального приема может быть несколько упрощена, если отказаться от непосредственного приема сложного сигнала «в целом» и вести когерентный поэлементный прием сигнала с последующей обработкой полученных результатов.

Приняв во внимание, что посылки сложного сигнала можно представить в виде

$$s_{icl}(t) = \sum_{k=1}^M s_{ik}(t - k\Delta\tau_0), \quad i = 1, 2, \dots, \quad (2.35)$$

где $S_{ik}(t)$ – элемент сложного сигнала, получим

$$\sum_{k=1}^M \int_0^{\tau_0} y(t) s_{1k}(t - k\Delta\tau_0) dt \begin{matrix} > \\ < \end{matrix} \sum_{k=1}^M \int_0^{\tau_0} y(t) s_{2k}(t - k\Delta\tau_0) dt \quad \begin{matrix} \hat{x}_i = x_1, \\ \hat{x}_i = x_2. \end{matrix} \quad (2.36)$$

Это выражение показывает, что оптимальный прием сводится к когерентной обработке всех элементов посылки сигнала, суммированию полученных результатов и принятию решения о переданном символе в соответствии с результатом сравнения образованных сумм.

Для конкретизации выражения (2.36) необходимо задать вид используемого сигнала. При наиболее часто применяемых противофазных сложных сигналах элементы $S_{1\text{сл}}(t)$ и $S_{2\text{сл}}(t)$ могут принимать только два возможных значения

$$s_{ik}(t) = \begin{cases} S_0 \sin \omega_0 t \\ S_0 \sin(\omega_0 t + \pi) \end{cases}, \quad 0 \leq t \leq \Delta\tau_0. \quad (2.37)$$

Тогда

$$s_{ik}(t - k\Delta\tau_0) = \varphi_{ik}(t) S_0 \sin \omega_0(t), \quad (2.38)$$

где $\varphi_{ik}(t)$ – двоичная функция, принимающая значения +1 и -1 и определяющая последовательность смены фаз элементов сложного сигнала.

Приняв во внимание (2.38), выражение (2.36) сложного двоичного сигнала с противофазной можно записать в виде

$$\sum_{k=1}^M \int_0^{\tau_0} y(t) \varphi_{1k}(t) \sin \omega_0 t dt \begin{cases} > \\ < \end{cases} \sum_{k=1}^M \int_0^{\tau_0} y(t) \varphi_{2k}(t) \sin \omega_0 t dt \begin{cases} x_i = x_1, \\ x_i = x_2. \end{cases} \quad (2.39)$$

Это правило можно реализовать различными оптимальными схемами приема. Одна из них приведена на рис. 2.15.

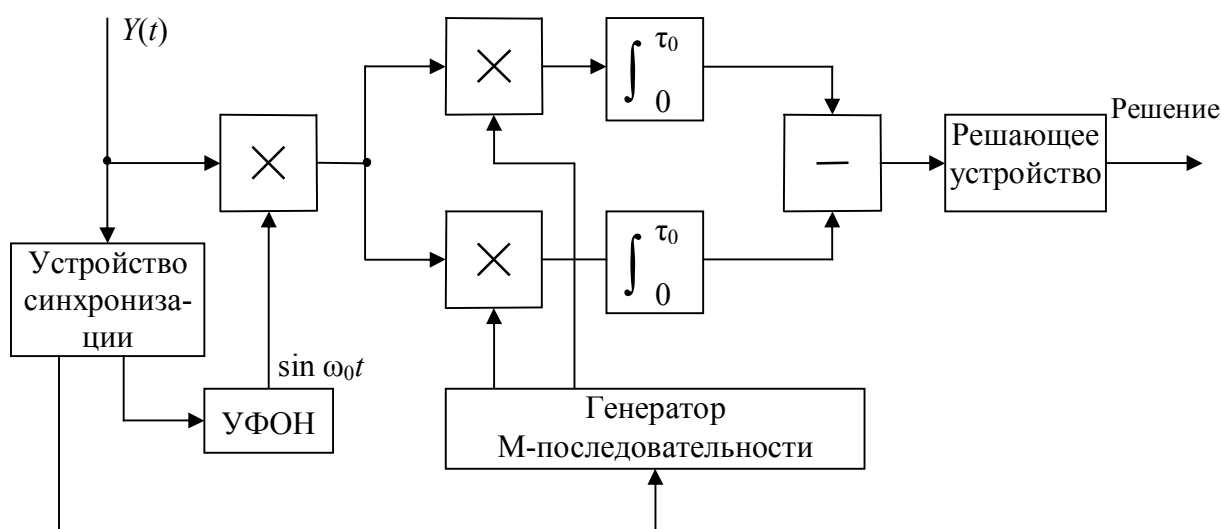


Рис. 2.15. Структурная схема оптимального приёмника двоичных сигналов с противофазной ФМН

Достоинством этой схемы является то, что она не требует «образцов» в виде сложных сигналов. Для приема достаточно иметь гармоническое опорное напряжение с нулевой фазой и псевдослучайную последовательность, формируемую по такому же правилу, как и на передающей стороне системы. В одном канале приемника используется последовательность, снимаемая с одного плеча выходной ячейки регистра сдвига, а в другом канале – с другого плеча.

Хотя принципиально все варианты оптимального приема обеспечивают одинаковые результаты, их значимость с точки зрения практической реализации различна. Дело в том, что при практической реализации всегда имеется некоторое рассогласование между параметрами принимаемого сигнала и параметрами приемника. С увеличением базы сигнала значительно повышаются требования к точности изготовления и стабильности узлов и элементов приемника. Невыполнение этих требований может привести к заметному снижению помехоустойчивости по сравнению с оптимальным приемом. В этой связи особое значение приобретает выбор такого варианта оптимального приема, реализация которого обеспечивала бы значительное ослабление влияния дестабилизирующих факторов на помехоустойчивость.

Если случайные рассогласования между параметрами сигнала и приемника таковы, что априорными сведениями о фазе сигнала воспользоваться нельзя (сигнал имеет случайную фазу), лучшие результаты обеспечивает оптимальный некогерентный прием.

2.4.2. Оптимальный некогерентный прием. Допустим, что цифровая информация передается сложными двоичными сигналами, определяемыми выражением (2.35). Если из-за различных факторов начальная фаза элементов таких сигналов приобретает случайный характер, то при противофазных сигналах каждый элемент сигнала можно записать в виде

$$s_{ik}(t) = s_{ik}(t - k\Delta\tau_0) = S_0\phi_{1k}(t) \sin[\omega_0 t + \varphi_x(t)]. \quad (2.40)$$

Это выражение отличается от (2.38) только тем, что фаза элемента сигнала имеет дополнительный случайный сдвиг $\varphi_x(t)$.

Полагая случайные изменения фазы медленными, т. е. удовлетворяющими условию

$$\varphi_x(t) \approx \varphi_x = \text{const}, \quad 0 \leq t \leq \tau_0,$$

можно показать, что правило работы оптимального приемника в этом случае определяется выражением

$$V_1 \begin{matrix} > \\ < \end{matrix} V_2 \begin{matrix} x_i = x_1, \\ x_i = x_2, \end{matrix} \quad (2.41)$$

где

$$V_i = \left\{ \left[\sum_{k=1}^M \int_0^{\tau_0} y(t) \phi_{ik}(t) \sin \omega_0 t dt \right]^2 + \left[\sum_{k=1}^M \int_0^{\tau_0} y(t) \phi_{ik}(t) \cos \omega_0 t dt \right]^2 \right\}^{1/2}, \quad i = 1, 2. \quad (2.42)$$

Так же, как и в предыдущем случае, возможен ряд вариантов схемы оптимального приемника, соответствующего правилу (2.41). Один из таких вариантов приведен на рис. 2.16.

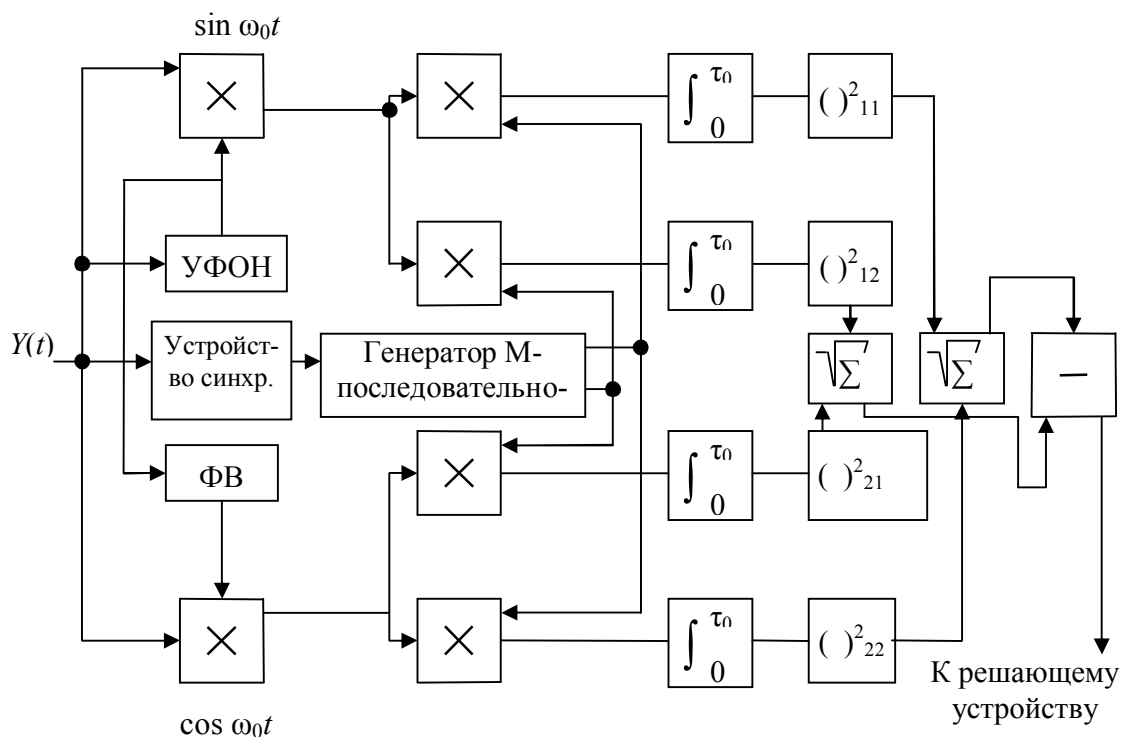


Рис. 2.16. Структурная схема некогерентного приёмника двоичных сигналов

Схема приемника оказывается довольно сложной. Особенность ее работы по сравнению со схемой оптимального приема сложного сигнала, известного точно, состоит в том, что принятое колебание $y(t)$ разлагается на два ортогональных, соответствующих синусоидальному косинусоидальному опорным напряжениям (рис. 2.17). При этом составляющие каждой проекции $Y_B(t)$ и $Y_C(t)$ оказываются когерентными с опорными напряжениями. Приемник определяет длину вектора каждой проекции, а затем, зная их, находит длину вектора принятого сигнала. Так как посылка сигнала может принять одно из двух возможных значений, то необходимо вычислять два возможных значения вектора. Решение принимается после сравнения этих значений. Переданным считается тот информационный символ, которому соответствует принятый сигнал с большей величиной вектора.

При практической реализации приема сложных сигналов возникает ряд трудностей, которые приводят к ухудшению показателей работы системы по сравнению с оптимальным приемом. Эти трудности обусловлены неидеальностью характеристик элементов и узлов реальных приемных устройств. Нестабильностью этих характеристик во времени из-за влияния различных дестабилизирующих факторов (случайных изменений температуры, питающих напряжений и т.п.), а также наличием технологического разброса параметров.

К перечисленным причинам добавляются такие факторы, как неидеальность синхронизации, невозможность иметь точное значение несущей частоты в пункте приема и т.п. Указанные обстоятельства приводят к тому, что даже сравнительно небольшие отклонения от оптимальных методов приема могут существенно ухудшить помехоустойчивость системы, особенно если база сигнала значительна. Все это делает проблему реализации приёмных устройств сложных сигналов очень трудной и многообразной.

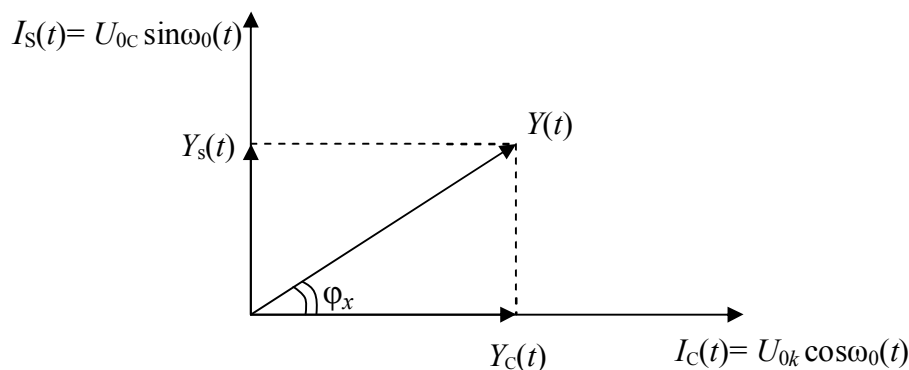


Рис. 2.17. Представление принятого сигнала в виде ортогональных сигналов

2.5. Примеры применения сложных сигналов в цифровых системах передачи данных

Применение сложных сигналов позволяет в ряде случаев решать задачи передачи информации более успешно по сравнению с традиционными методами. К таким случаям можно отнести передачу информации по каналам с многолучевым эффектом распространения; ослабление мешающего действия различных систем, работающих в том же диапазоне частот; повышение эффективности использования выделенного диапазона частот; ослабление влияния узкополосных помех и т. п. Кроме этого, сложные сигналы открывают ряд возможностей построения систем с такими свойствами, которые не могут быть получены в системах с простыми сигналами: скрытность, крипто- и имитостойкость связи; возможность одновременно с передачей информации измерять координаты движущегося объекта, на котором расположена передающая часть системы, и т.д.

Ниже кратко рассматриваются некоторые примеры применения сложных сигналов.

2.5.1. Применение сложных сигналов в многолучевых каналах. Эффект многолучевого распространения в каналах со случайными параметрами приводит к появлению интерференции между соседними посылками сигнала и их замираниям. При простых сигналах влияние интерференции можно уменьшить, только увеличивая длительность посылок, что, естественно, приводит к уменьшению скорости передачи. Для ослабления существенных за-

мираний уровня простые сигналы приходится принимать методами разнесенного приема.

Применение сложных сигналов в ряде случаев позволяет решить задачу ослабления влияния интерференции и замираний более успешно. Рассмотрим для этого рис. 2.18, на котором показаны огибающие одной посылки сложного сигнала, пришедшей по трем разным лучам. Полагаем, что замирания являются общими и за время, равное длительности посылки τ_0 , амплитуда и начальная фаза элементов сигнала не меняются.

Оптимальная обработка посылки сложного сигнала корреляторами или согласованными фильтрами дает выходное напряжение, вид которого совпадает с корреляционной функцией этой посылки. Выходные напряжения, соответствующие одной и той же посылке и разным лучам, показаны на рис. 2.18, б. Длительность основного лепестка этого напряжения приблизительно равна величине

$$\tau_{\text{блх}} \approx 1 / \Delta f_{\text{э}} = \tau_0 / \Delta f_{\text{э}} \tau_0 = \tau_0 / B. \quad (2.43)$$

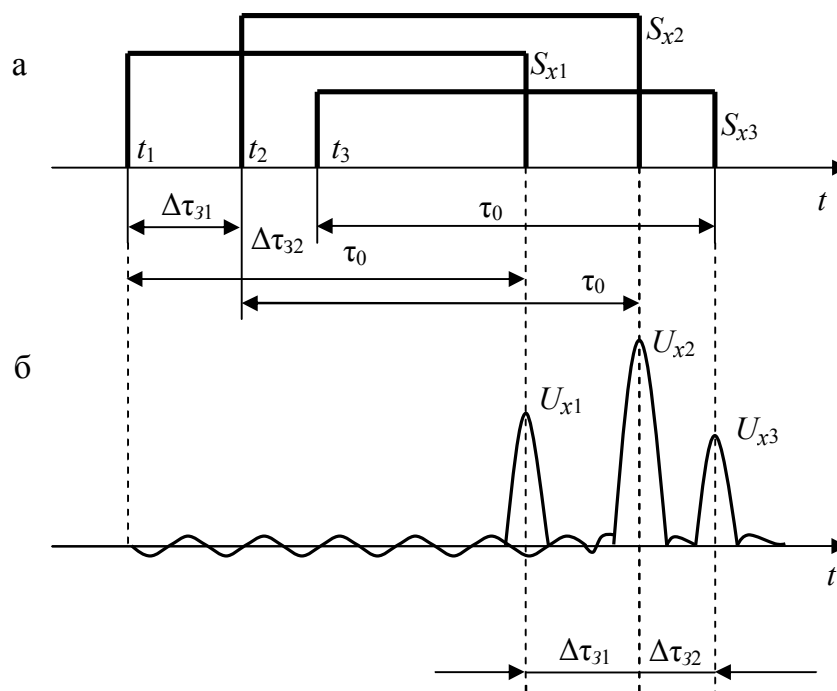


Рис. 2.18. Огибающие сложного сигнала:
а – пришедшего по трём разным лучам; б – на выходе приёмника

Если база сигнала значительна ($B \gg 1$), то $\tau_{\text{блх}} \ll \tau_0$ и, следовательно, в результате обработки посылки происходит ее «сжатие» по времени. Нетрудно видеть, что при условии

$$\Delta \tau_{3 \min} \geq \tau_{\text{блх}} \quad (2.44)$$

выходные напряжения разделяются друг от друга несмотря на то, что на входе посылки от разных лучей перекрываются. Из (2.44) и (2.43) следует, что

база сигнала, при которой обеспечивается разделение лучей с минимальной разностью запаздывания $\tau_{3\min}$, должна удовлетворять условию

$$B \geq \tau_0 / \Delta\tau_{3\min}. \quad (2.45)$$

Так как огибающие посылки, пришедшей по разным лучам S_{x1} , S_{x2} , S_{x3} , случайны, то случайны и выходные напряжения U_{x1} , U_{x2} , U_{x3} .

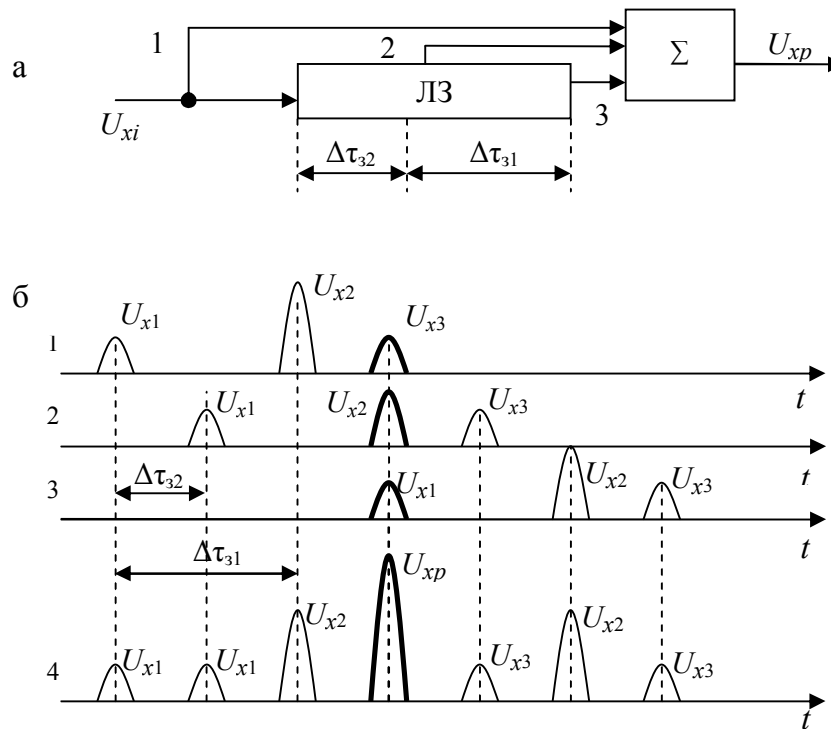


Рис. 2.19. Процесс обработки посылок:
а – схема устройства; б – временные диаграммы

Оптимальная обработка посылок представляет собой линейную операцию, поэтому закон распределения выходных напряжений остается таким же, как и у огибающих. Если выходные напряжения обработать далее схемой, показанной на рис. 2.19, а, то на ее выходе образуется последовательность импульсов со случайными амплитудами (рис. 2.19, б), среди которых будет импульс с амплитудой U_{xp} . Поскольку этот импульс представляет собой результат сложения трех импульсов со случайными амплитудами U_{x1} , U_{x2} и U_{x3} , закон распределения величины U_{xp} изменится по сравнению с распределениями случайных величин U_{x1} , U_{x2} и U_{x3} . В частности, если величины U_{x1} , U_{x2} и U_{x3} подчиняются распределению Релея, то распределение U_{xp} определяется композицией этих распределений и, следовательно, замирания величин U_{xp} менее глубоки, чем замирания случайных величин U_{x1} , U_{x2} и U_{x3} . Нетрудно видеть, что указанная процедура эквивалентна разнесенному приему с линейным сложением трех ветвей.

Так как время запаздывания лучей и разность хода между ними случайны, то в общем случае необходимо применять линию с временем задержки

$$\tau_{лз} = \Delta\tau_{3\max} - \Delta\tau_{3\min} = \tau_p,$$

равным времени многолучевого растяжения сигнала, а отводы брать через интервалы, равные минимальному времени запаздывания $\Delta\tau_{3\min}$. Значения $\Delta\tau_{3\min}$ и $\Delta\tau_{3\max}$ определяются при экспериментальных исследованиях свойств конкретного канала. Таким образом, применение сложных сигналов позволяет разделить перекрывающиеся сигналы, приходящие по разным лучам. Это дает возможность эффективно использовать энергию сигналов от отдельных лучей и существенно ослабить влияние замираний, обусловленных эффектом многолучевого распространения.

Изложенные выше идеи впервые были реализованы в системе «Рейк». Название системы происходит от английского слова *Rake* – грабли. Действительно, напряжение на выходе сумматора (рис. 2.19) напоминает зубья грабеля. Видимо, это и послужило основанием для такого термина. Эта система предназначалась для передачи двоичной информации на расстояние около 4500 км методом ЧМ_Н сигналов, манипулированных псевдослучайной бинарной последовательностью. Ширина спектра сигнала выбиралась равной 10 кГц, а база $B_c = 220$. В зависимости от времени суток использовались диапазоны 8, 12 и 17 МГц. При мощности передатчика 22 кВт система обеспечивала передачу информации с вероятностью искажения символа не более 10^{-6} и была эквивалентна системе примерно с 3 – 4 кратным пространственным разнесением.

2.5.2. Применение сложных сигналов для борьбы с помехами. Применение сложных сигналов иногда позволяет эффективно бороться с различными помехами, спектр которых сосредоточен в значительно более узком диапазоне частот по сравнению с диапазоном, занимаемым спектром сложного сигнала.

Как уже неоднократно указывалось, помехоустойчивость оптимального приема в гауссовских каналах зависит от величины

$$h_0^2 = E_0 / N_0 \cdot (P_c / P_{ш})_{вх} B \cdot \rho^2 \cdot B, \quad (2.46)$$

где E_0 – энергия посылки длительностью τ_0 ; N_0 – спектральная плотность шума; P_c – мощность широкополосного сигнала; $P_{ш}$ – мощность шума; B – база сигнала.

Для сложных сигналов с большим значением базы ($B \gg 1$) требуемая величина h_0 , при которой достигается заданная вероятность ошибки приема, может быть получена при малом входном отношении мощности сигнала к мощности шума. Это означает, что уровень сигнала на входе приемника оказывается значительно меньше уровня шума. Если ввести условную величину $N_{0с}$, характеризующую среднюю интенсивность мощности сигнала на единицу полосы, то для сигнала с большой базой можно записать условие

$$(P_C / P_{\text{ш}})_{\text{вх}} = P_C / N_0 \Delta f_{\text{э}} \quad (N_{0\text{с}} / N_0)_{\text{вх}} \ll 1.$$

Чем больше база сигнала, тем сильнее посылка «замаскирована» в шумах. При таких условиях обнаружить сам факт работы системы затруднительно. Так как мощность сигнала в такой системе распределена в широкой полосе частот, то ее интенсивность мала и такой сигнал практически не создает сколь-нибудь заметных помех системам, в которых используются значительно более узкополосные сигналы. Работа систем с узкополосными сигналами также не окажет существенного влияния на работу систем со сложными сигналами, если применять фильтрацию той части спектра сложного сигнала, где действует узкополосный сигнал. Потеря мощности сложного сигнала из-за такой фильтрации определяется соотношением

$$\Delta P_C / P_C \quad \Delta f_{\text{ус}} / \Delta f_{\text{э}} \equiv 1 / B \ll 1,$$

где $\Delta f_{\text{ус}}$ – ширина спектра узкополосного сигнала.

При большой базе эти потери незначительны и не приводят к заметному уменьшению величины напряжения после оптимальной обработки сложного сигнала.

Таким образом, применение сложных сигналов открывает возможность одновременной работы в одном и том же диапазоне частот узкополосных и широкополосных систем без заметного влияния их друг на друга (частотная совместимость систем).

На рис. 2.20 представлена зависимость базы ШПС B от отношения сигнал-помеха на входе ρ^2 дБ при значениях h_0^2 , равных 10, 20 и 30 дБ, построенные согласно (2.46).

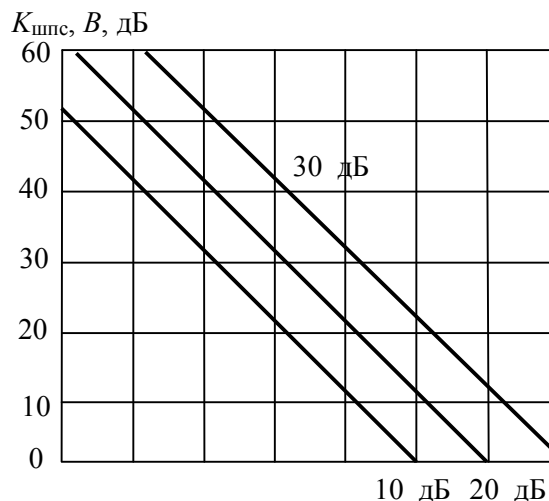


Рис. 2.20. Зависимость базы ШПС от отношения сигнал – помеха на выходе приёмника

Например, если необходимо иметь $h_0^2 = 20\text{дБ}$, а на входе приемника $\rho^2 = -40\text{ дБ}$, то требуемая база должна быть равна 60 дБ , т. е. $B = 10^6$.

Соотношение (2.46) является фундаментальным в теории систем связи с ШПС. Они получены для помехи в виде белого шума с равномерной спектральной плотностью мощности в пределах полосы частот, ширина которой равна ширине спектра ШПС. Вместе с тем эти соотношения справедливы для широкого круга помех (узкополосных, импульсных, структурных), что и определяет их фундаментальное значение.

2.5.3. Применение сложных сигналов для улучшения эффективности использования отведенного диапазона частот. Непрерывное увеличение числа различных систем связи делает особенно важной задачу рационального использования отведенных диапазонов частот. Наиболее распространенным до сих пор методом использования диапазона частот, выделенного для определенного класса систем, является метод частотной селекции сигналов. В соответствии с этим методом каждой системе выделяется определенный поддиапазон частот в пределах общего выделенного диапазона и разделение сигналов разных систем достигается их частотной селекцией.

Сложные сигналы можно разделять не только по частоте, но и по форме [11]. Разделение по форме (кодовое или структурное разделение) основано на корреляционных свойствах ансамбля сложных сигналов, различающихся тонкой структурой (формой), обусловленной расположением последовательности элементов, образующих эти сигналы. Хорошие корреляционные свойства ансамбля сложных сигналов позволяют успешно использовать такие сигналы в системах, работающих одновременно в одном и том же диапазоне частот.

Выясним, при каких условиях такое применение сложных сигналов может улучшить эффективность использования отведенного диапазона частот по сравнению со случаем частотной селекции.

Пусть в отведенном диапазоне частот $\Delta F_{\text{общ}}$ необходимо организовать разветвленную сеть связи, позволяющую каждому из N абонентов в любое время связаться с любым другим абонентом.

При частотном уплотнении каждому абоненту выделяется полоса частот ΔF_k , а для улучшения условий селекции этих полос вводятся защитные полосы $\Delta F_{\text{защ}}$. Тогда

$$\Delta f_k + \Delta f_{\text{защ}} = \Delta f_k (1 + \alpha_{\text{защ}}) = \Delta f_{\text{общ}} / N,$$

$$\Delta f_k = \Delta f_{\text{общ}} / N(1 + \alpha_{\text{защ}}),$$

где $\alpha_{\text{защ}}$ – коэффициент, учитывающий введение защитной полосы.

При уплотнении по форме (кодовое или структурное уплотнение) все N абонентов используют весь отведенный диапазон.

Для сравнения эффективности использования полосы в первом и втором случаях воспользуемся критерием удельных затрат полосы. Тогда

$$\beta_{I\Delta f} = \frac{\Delta f_{\text{общ}}}{\sum_{i=1}^N C_{Ii}}, \quad \beta_{II\Delta f} = \frac{\Delta f_{\text{общ}}}{\sum_{i=1}^N C_{IIi}},$$

где C_{Ii} и C_{IIi} – пропускная способность по i -му каналу в первом и втором случаях соответственно. Далее для простоты будем полагать, что эти значения не зависят от номера канала, и обозначим их C_{Ik} и C_{IIk} . Очевидно, что условием лучшей эффективности систем со сложными сигналами является

$$\frac{\beta_{II\Delta f}}{\beta_{I\Delta f}} = \frac{C_{Ik}}{C_{IIk}} < 1. \quad (2.47)$$

Для конкретизации этого условия воспользуемся формулой Шеннона. При этом будем считать, что мощность сигнала и спектральная плотность шума в обоих случаях одинаковы. Тогда

$$C_{Ik} = \frac{\Delta f_{\text{общ}}}{N(1 + \alpha_{\text{заш}})} \log_2 \left[1 + \frac{P_c N (1 + \alpha_{\text{заш}})}{N_0 \Delta f_{\text{общ}}} \right],$$

$$C_{IIk} = \Delta f_{\text{общ}} \log_2 \left[1 + \frac{P_c}{N_0 \Delta f_{\text{общ}}} \right]. \quad (2.48)$$

С учётом (2.48) условие (2.47) принимает вид

$$\frac{\log_2 [1 + AN(1 + \alpha_{\text{заш}})]}{N(1 + \alpha_{\text{заш}}) \log_2 (1 + A)} < 1, \quad (2.49)$$

где $A = P_c / N_0 \Delta f_{\text{общ}} = (P_c / P_{\text{ш}})_{\text{вх}}$.

Эта величина определяет отношение сигнал/шум на входе приемника системы со сложными сигналами. Для таких сигналов $A < 1$. Так как отношение логарифмов не зависит от их основания, то, перейдя к натуральным логарифмам и используя разложение $\ln(1 + A) \approx A$ (при $A \leq 0,2$), получим

$$\frac{\ln[1 + AN(1 + \alpha_{\text{заш}})]}{AN(1 + \alpha_{\text{заш}})} < 1. \quad (2.50)$$

Исследование этого условия показывает, что оно выполняется, начиная со значений

$$AN(1 + \alpha_{\text{заш}}) \geq 0,23. \quad (2.51)$$

Величину $\alpha_{\text{заш}}$ обычно выбирают в пределах $0,1 \dots 0,3$. Принимая $\alpha_{\text{заш}} = 0,2$ и учитывая выражение для A , получаем из (2.51) следующее условие:

$$N(P_c / P_{\text{ш}})_{\text{вх}} \geq 0,2. \quad (2.52)$$

Это условие определяет минимальное число каналов, начиная с которого система со сложными сигналами эффективнее использует полосу, чем системы с простыми сигналами. Чем меньше отношение сигнал/шум (чем больше база сигнала), тем при большем числе каналов начинается улучшение.

Примерами систем, эффективно использующих выделенный диапазон частот, могут служить различные дискретно-адресные системы с вызовом произвольного абонента, спутниковые системы связи со свободным доступом и т. п. Такие системы позволяют осуществлять связь между большим числом различных абонентов в любое необходимое время, т.е. эти системы являются несинхронными. Очень часто подобные системы называют асинхронно-адресными. Так как асинхронно-адресные системы, работающие в одном и том же диапазоне частот, могут создавать друг другу взаимные помехи, одной из основных проблем при их разработке является проблема выбора сигналов. Эти сигналы должны принадлежать к ансамблю с «хорошими» корреляционными свойствами (обеспечивать малые значения взаимокорреляционных функций по сравнению с основным пиком автокорреляционных функций). Для этого часто используют многочастотные составные сигналы, получаемые частотно-временным кодированием. Применение таких сигналов позволяет упростить приемно-передающую аппаратуру систем.

Обычно передачу цифровой информации в асинхронно-адресных системах ведут двоичными сигналами. Поэтому из выбранного ансамбля каждой системе выделяется один (при передаче с пассивной паузой) или два (при передаче с активной паузой) сложных сигнала. Эти же сигналы одновременно исполняют роль «адреса» данной системы. Каждое приёмное устройство может быть настроено на «адрес» любой системы, с которой необходимо установить связь.

Получить ансамбль большого числа сигналов, обладающих «хорошими» корреляционными свойствами, затруднительно. Поэтому некоторые сигналы могут давать заметные значения взаимокорреляционных функций. Если число одновременно работающих систем достаточно велико, то уровень помех на выходе коррелятора или оптимального фильтра приемника может быть значительным, так как он пропорционален сумме взаимокорреляционных функций сигналов работающих систем. Такие взаимные помехи являются основным фактором, ухудшающим качество передачи информации в адресных системах. Для ослабления их в подобных системах часто применяют определенную дисциплину работы. Если число активных систем возрастает настолько, что уровень взаимных помех становится недопустимым, то определенная часть систем должна прекращать работу.

2.5.4. Скрытность системы связи. Это способность противостоять обнаружению и измерению параметров. Скрытность – понятие очень ёмкое, так как включает в себя большое множество особенностей обнаружения ШПС и измерения их параметров. Поскольку обнаружение ШПС и измерение параметров возможны при различной первоначальной осведомленности (априорной неопределенности) о системе связи, то можно указать только основные

соотношения, характеризующие скрытность. Когда известно, что в данном диапазоне частот может работать система связи, но параметры ее неизвестны, то в этом случае можно говорить об энергетической скрытности системы связи, так как ее обнаружение возможно с помощью анализа спектра (энергетическое обнаружение). Характеристика обнаружения (вероятности ложной тревоги и пропуска сигнала) полностью определяется отношением сигнал-помеха на входе приемника-анализатора $\rho^2 = P_c/P_{ш}$, где помеха представляет собой собственный шум приёмника $P_{ш} = kT_0(N_{ш} - 1)F$, а k – постоянная Больцмана, T_0 – температура окружающей среды, $N_{ш}$ – коэффициент шума приёмника. Время обнаружения ШПС при условии $\rho^2 \ll 1$ приближенно определяется соотношением $T_{обн} = F^{-1}(\rho^2)^{-2} 2q^4$ или

$$T_{обн} \approx aF, \quad (2.53)$$

где размерная постоянная $a = 2[q^2 kT_0(N_{ш} - 1)/P_c]^2$ зависит как от шумовых свойств приемника, мощности сигнала на входе, так и от требуемого отношения сигнал-помеха на выходе Q^2 . Таким образом, чем шире ширина спектра ШПС, тем больше время обнаружения, тем выше энергетическая скрытность системы связи.

Таким образом, чем шире спектр ШПС и чем больше его база, тем выше как энергетическая, так и параметрическая скрытность. Для борьбы с радиоразведкой в помехозащищённых системах связи применяют также смену ШПС. Частота смены ШПС, их выбор из некоторого ансамбля (системы сигналов) определяется многими требованиями к системе связи и не может быть однозначно определен. Однако полагают, что число сигналов в системе (или объем системы сигналов) должно быть много больше базы ШПС. Можно предположить, что для помехозащищённых систем связи объем системы сигналов L определяется степенным законом:

$$L \sim B^m, \quad (2.54)$$

где m – некоторое число, по крайней мере удовлетворяющее условию $m \geq 2$, хотя для работы может использоваться гораздо меньшее число ШПС.

Следовательно, использование ШПС повышает помехоустойчивость и скрытность системы связи, т.е. её помехозащищенность. Как следует из материалов печати, ШПС используют в спутниковых системах связи, в авиационных системах связи, в радиорелейных линиях, в спутниковых навигационных системах и мобильных системах связи.

2.5.5. Кодовое разделение абонентов. Благодаря быстрому развитию микроэлектроники, ШПС нашли широкое применение в коммерческих системах связи. Основой этих систем является кодовое разделение абонентов за счет ШПС, отличающихся по форме.

При больших базах можно построить большое число различных ШПС. Например, пусть ШПС представляет собой фазоманипулированный сигнал, состоящий из радиоимпульсов, фазы которых 0 или π , а число их равно B . Можно построить множество сигналов (так называемый полный код), число

сигналов в котором равно 2^B , а сигналы между собой отличаются хотя бы в одном импульсе. Если положить $B = 100$, то имеем $2^{100} \sim 10^{30}$ различных сигналов. Из такого большого множества можно отобрать систему сигналов так, чтобы каждому абоненту в системе связи выделить свои собственные сигналы. При этом все абоненты могут работать в общей полосе частот, а разделение их возможно за счет различия ШПС по форме. Такое разделение абонентов называется кодовым. При этом ШПС является по сути дела адресом абонента и в этом случае принципиально нет необходимости в принудительной временной синхронизации абонентов. Поэтому подобные системы связи получили название асинхронных адресных систем связи (ААСС). Они основаны на применении ШПС и кодовом разделении абонентов.

В ААСС все абоненты работают в общей полосе частот. Поэтому при передаче информации ШПС различных абонентов перекрываются по времени и по частоте и создают взаимные помехи. Однако при использовании ШПС с большими базами возможно свести уровень взаимных помех до требуемого, чтобы обеспечить необходимое качество приема информации. Если предположить, что на входе одного из приемников системы связи действует L мешающих ШПС с одинаковыми мощностями, то отношение сигнал-помеха на выходе приемника

$$h_0^2 = B / L. \quad (2.53)$$

Таким образом, увеличивая базу ШПС, всегда можно добиться требуемого качества приема информации.

На рис. 2.21 представлены зависимости базы ШПС от числа активных абонентов, построенные согласно (2.53). Графики рис. 2.21 позволяют определить помехоустойчивость ААСС.

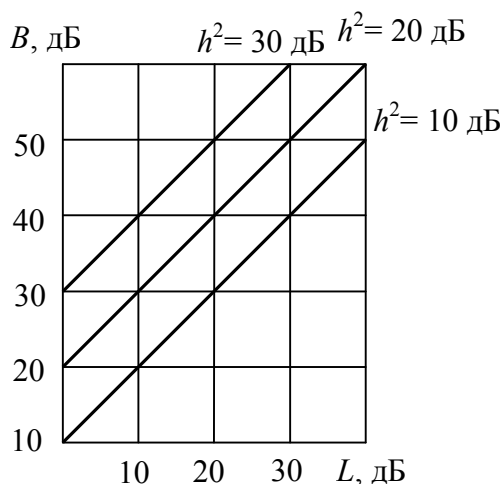


Рис. 2.21. Помехоустойчивость ААСС

В последние годы широкое распространение получили радиотелефонные системы подвижной связи. Вся обслуживаемая территория разбивается на большое число зон в виде сот. В каждой зоне радиосвязь ведется на частотах,

специально выделенных этой зоне. За счет территориального разнесения зон с одинаковыми частотными каналами возможно многократное использование одних и тех же частотных каналов. Такие системы связи получили название сотовых систем подвижной связи (ССПС). Прием сигналов в таких системах принципиально сопровождается взаимными помехами так же, как и в ААСС. Поэтому применение ШПС в ССПС перспективно, поскольку позволяет успешно бороться с взаимными помехами. Эффективность ССПС

$$\gamma_{\text{ССПС}} \approx 3,63(R_0 / D)^2 / F_k, \quad (2.54)$$

где R_0 – радиус зоны обслуживания; D – защитный интервал; F_k – ширина частотного канала. Если положить $R_0 = 30$ км, $D \approx 4,4$ км, радиус зоны $R = 0,85$ км, а $F_k = 50$ кГц, то $\gamma_{\text{ССПС}} \approx 3333$ аб/МГц.

2.5.6. Измерение координат подвижных объектов. Применение ШПС позволяет совместить системы передачи информации и системы траекторных измерений. При измерении параметров движения объекта наибольший интерес представляют расстояние между приёмником и передатчиком и их относительная скорость. Расстояние измеряется по задержке во времени, а скорость – по доплеровскому смещению частоты. Точность измерения и разрешающая способность по задержке определяются отношением сигнал-помеха h_0^2 (2.46) и шириной спектра сигнала и характеризуются ошибкой

$$\sigma_t = 1/(h_0 F). \quad (2.55)$$

Чем больше h_0 и F , тем меньше ошибка в измерении задержки, тем выше точность измерения и разрешающая способность по расстоянию. Точность измерения доплеровского смещения частоты определяется отношением сигнал-помеха h_0^2 (2.46) и длительностью сигнала и характеризуется ошибкой

$$\sigma_f = 1/(h_0 T). \quad (2.56)$$

Чем больше h_0 и T , тем меньше ошибка в измерении доплеровского сдвига частоты, тем выше точность измерения и разрешающая способность по скорости. Из (2.55), (2.56) следует, что при совместном измерении расстояния и скорости необходимо использовать ШПС, так как только для ШПС можно независимо изменять и ширину спектра F и длительность T . В системах связи длительность T обычно определяется скоростью передачи информации. Поэтому повышения точности измерения расстояния можно достигнуть расширением спектра F , т.е. используя ШПС.

2.5.7. Электромагнитная совместимость. Шумоподобные сигналы обеспечивают хорошую электромагнитную совместимость ШСС (ЭМС) с узкополосными системами радиосвязи и вещания. На рис. 2.22 изображены спектры ШСС с ШПС с шириной спектра F и узкополосной системы связи с шириной спектра сигнала F_γ . Соответственно для ШПС спектральная плот-

ность мощности $N_{\text{шпс}} = P_{\text{шпс}}/F$, для узкополосного сигнала $N_y = P_y/F_y$. Помехоустойчивость системы связи с ШПС определяется фундаментальным соотношением (2.46), в котором $\rho^2 = P_{\text{шпс}}/P_y$. Если узкополосная система связи постоянно занимает определенный интервал, то можно её спектр полностью подавить, используя режекторный фильтр, настроенный на частоту узкополосной системы связи. Таким образом, воздействие узкополосной системы связи на широкополосную незначительно. В свою очередь, широкополосная система связи также слабо влияет на узкополосную систему связи. Мощность ШПС, проходящего на выход приемника, $N_{\text{шпс}}F_y = P_{\text{шпс}}F_y/F$. Поэтому отношение сигнал-помеха на выходе узкополосного приемника будет определяться соотношением (2.46), в котором $\rho^2 = P_y/P_{\text{шпс}}$, а $B = F/F_y$. Поэтому чем больше отношение (2.46), тем лучше фильтрация ШПС в узкополосной системе связи. Следовательно, чем больше база ШПС, тем выше ЭМС широкополосной и узкополосной систем связи.

Системы связи с ШПС можно совмещать и с радиотелевизионными системами. На рис. 2.23 изображен спектр телевизионного сигнала $N_{\text{ТВ}}$. Программы телевидения в одной территориальной зоне передаются по нескольким каналам с большими защитными частотными интервалами. Обычно в этих частотных защитных интервалах не допускается работа каких-либо радиотехнических систем, чтобы не создавать помех телевизионным передачам. Однако можно в этих частотных интервалах разместить системы связи с ШПС так, как это показано на рис. 2.23. Спектр ШПС расположен вблизи спектра телевизионного сигнала, там, где спектральная плотность последнего резко уменьшается. При этом взаимные помехи и той, и другой системе будут малыми. Следует отметить, что если вместо ШПС использовать сигналы с частотной модуляцией, то уровень взаимных помех возрастает, так как сигналы системы связи и телевидения относятся к одинаковому классу и демодулируются частотным детектором.

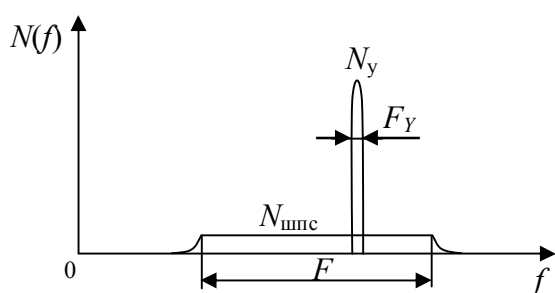


Рис. 2.22. Спектры широкополосной и узкополосной систем связи

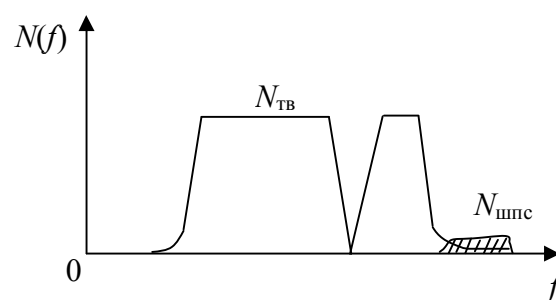


Рис. 2.23. Спектры телевизионного сигнала и ШПС

Таким образом, системы связи с ШПС обладают хорошей ЭМС с системами радиосвязи, вещания и телевидения. Ранее было упомянуто, что ШПС обеспечивают высокую эффективность использования радиоспектра в ССПС. Если рассматривать действие систем связи в некотором замкнутом простран-

стве, то оказывается, что наилучшую ЭМС при ограниченном диапазоне частот обеспечивают ШПС, хотя сами по себе они требуют более широкой полосы, чем традиционные узкополосные системы. В то же время общая полоса частот при использовании ШПС будет меньше. Из рассмотрения основных свойств ШПС следует, что применение ШПС в системах передачи данных позволяет обеспечивать высокую помехоустойчивость относительно мощных помех, скрытность, адресность, работоспособность в общей полосе частот, борьбу с многолучевостью, высокие точности измерений и разрешающие способности, хорошую ЭМС со многими радиотехническими системами.

3. ФОРМИРОВАНИЕ ТРЕБУЕМОГО ЭНЕРГЕТИЧЕСКОГО СПЕКТРА ДАННЫХ

3.1. Линейные коды для передачи информации по последовательным каналам связи

Данные пользователя, поступающие от источников информации уже являются цифровыми, представленными в униполярном или биполярном коде без возврата к нулю – *NRZ (NonRetURn to ZeRo)*. При передаче данных на большие расстояния в коде *NRZ* возникают следующие проблемы:

- с течением времени нарастает постоянный ток, блокируемый некоторыми электрическими устройствами цифрового тракта, например, трансформаторами, что приводит к искажению передаваемых импульсов;
- изменение постоянного тока в цепи отрицательно сказывается на функционировании устройств, получающих питание от линии (репитеры или *CSU*);
- передача длинных серий нулей или единиц приводит к нарушению правильной работы устройств синхронизации;
- отсутствует возможность контроля возникающих ошибок на уровне физического канала.

Перечисленные проблемы решаются при помощи методов линейного кодирования. Параметры получаемого линейного сигнала должны быть согласованы с характеристикой используемой линии и отвечать ряду следующих требований:

- энергетический спектр линейного сигнала должен быть как можно уже. В нем должна отсутствовать постоянная составляющая, что позволяет повысить верность либо дальность передачи;
- структура линейного сигнала должна обеспечивать возможность выделения тактовой частоты на приемной стороне;
- обеспечение возможности постоянного контроля за ошибками на уровне физической линии;
- линейный код должен иметь достаточно простую техническую реализацию;

– сигнал должен быть неполярным, т.е. полярность сигнальных проводников не должна иметь значения, в таком случае система передачи не боится ошибок типа «зеркальный прием», или «обратная работа» (инверсия знаков), а также переплюсовки контактов физической линии или используемых разъемов.

Формирование требуемого энергетического спектра может быть осуществлено соответствующим изменением структуры импульсной последовательности и выбором нужной формы импульсов. Например, даже сокращение длительности импульсов в два раза (биимпульсный код с возвратом к нулю, *RZ*) вдвое уменьшает уровень постоянной составляющей и увеличивает уровень тактовой составляющей в спектре такого сигнала.

Различают неалфавитные (*1B1T*, *1B1Q*) и алфавитные (*mBnB*, *mBnT*, *mBnQ*) коды (*B* – двоичное, *T* – троичное, *Q* – четверичное основание кода). В случае алфавитных кодов кодирующее устройство преобразует каждую группу из *m* символов (исходного алфавита) в новую группу из *n* символов (выходного алфавита), причем таким образом, чтобы примерно уравнивать число нулей и единиц в передаваемой последовательности. Для одного и того же кода (с одинаковым обозначением) может существовать несколько отображений исходного алфавита на выходной. Предельной помехоустойчивостью обладают сигналы, элементы которых равны, но противоположны по полярности. Примеры популярных простых линейных кодов приведены на рис. 3.2 и рис. 3.3. Остановимся на некоторых из них детальней, но прежде рассмотрим структуру последовательного канала связи.

3.1.1. Структура последовательного канала связи. Передача информации между достаточно удаленными устройствами требует представления ее в виде последовательного потока битов, характеристики которого зависят от особенностей конкретной системы. Физической основой такой системы является линия связи, которая обычно выполняется в виде витой пары проводов, коаксиального кабеля либо оптического кабеля.

В зависимости от расстояния данные, передаваемые по линии, могут однократно или многократно подвергаться ретрансляции с целью восстановления амплитуды и временных характеристик (рис. 3.1) [21, 71]

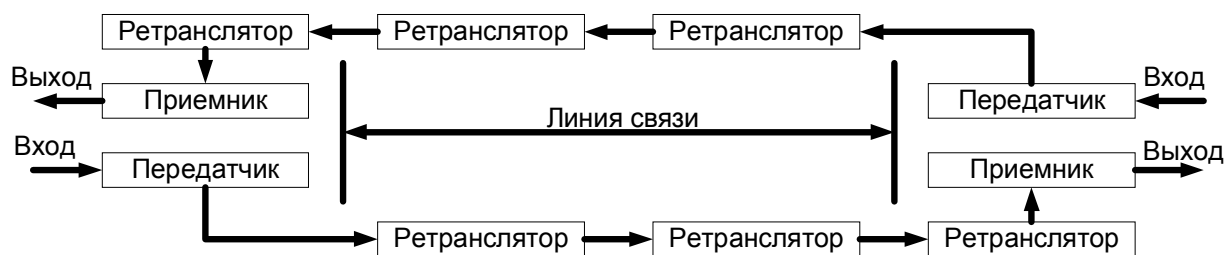


Рис. 3.1. Структура типового последовательного канала связи

Алгоритмы работы передатчика, ретранслятора и приемника определяются выбранным кодом, предназначенным для передачи по линии, который называют линейным кодом.

3.1.2. Униполярный код NRZ. Простейшим линейным кодом является униполярный код типа NRZ (Non Return to Zero), показанный на рис. 3.2, а. В этом коде нули представлены отсутствием импульса (напряжение, близкое нулю), а единицы - наличием импульса (некоторое положительное напряжение). Этот код имеет четыре недостатка.

1. Средняя мощность, выделяемая на нагрузочном резисторе R (на рисунке не показан), равна $A^2/2R$, где A – амплитуда импульса напряжения. Число 2 в знаменателе дроби соответствует равновероятному появлению лог. 0 и лог. 1 в потоке данных. Результат неутешительный. Резистор R рассеивает тепловую энергию в два раза интенсивнее, чем при биполярном кодировании (см. рис. 3.2, б) при той же амплитуде сигнала, равной A !

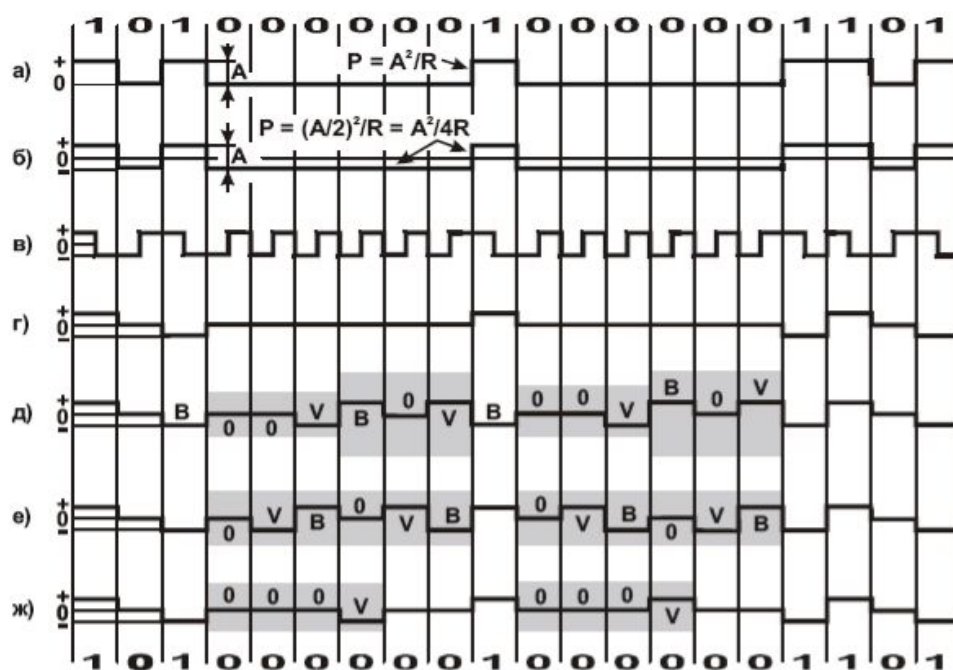


Рис. 3.2. Наиболее распространенные линейные коды:
а - униполярный код NRZ; б-биполярный код NRZ; в-код Манчестер-П;
г-код AMI; д-код BZS; е-код BZS; ж - код HDB3;
затемненными прямоугольниками выделены «заготовки»

2. Униполярные сигналы всегда содержат постоянную составляющую и значительную долю низкочастотных компонентов в спектре при передаче длинных последовательностей единиц. Это препятствует передаче сигналов через трансформаторы или конденсаторы.

3. Ретрансляторы и приемники надежно восстанавливают синхронизирующую временную сетку только тогда, когда паузы между изменениями сигнала не слишком велики. Изменение сигнала после незначительной паузы позволяет всякий раз корректировать «ход часов» ретранслятора или приемника. С увеличением паузы надежность «службы времени» падает. Например, после передачи серии из 10 тыс. нулей приемник, вероятнее всего, не сможет

точно определить, находится ли последующая единица на позиции 9999, 10000 или 10001. То же относится и к передаче длинных цепочек из лог. 1. Другими словами, при передаче достаточно большой последовательности нулей или единиц приемник (или ретранслятор) теряет синхронизацию с передатчиком (или ретранслятором).

4. Отсутствует возможность оперативной регистрации ошибок, таких как пропадание или появление лишних импульсов из-за помех.

3.1.3. Биполярный код NRZ. Биполярный сигнал NRZ (рис. 3.2, б) по сравнению с униполярным обладает лучшими энергетическими характеристиками. Единица представлена положительным уровнем напряжения, нуль – отрицательным. Нагрузочный резистор R в данном случае постоянно рассеивает тепло, так как на нем независимо от передаваемого кода присутствует напряжение $A/2$ той или иной полярности. Средняя мощность, выделяемая на нагрузочном резисторе, равна $(A/2)^2/R = A^2/4R$, т. е. половине средней мощности униполярного сигнала, хотя перепад уровней тот же самый.

Так что первый из отмеченных ранее недостатков униполярного сигнала NRZ в какой-то мере удалось устранить. Остальные три недостатка сохраняются. Для их ликвидации необходимо введение избыточности одним из двух способов:

1) скорость передачи сигналов по линии выбирается большей, чем скорость передачи информации, без использования дополнительных электрических уровней сигналов;

2) скорость передачи сигналов по линии выбирается равной скорости передачи информации, но вводятся дополнительные электрические уровни сигналов.

3.1.4. Код Манчестер-П (код PE). Примером кода с избыточностью, введенной согласно только что упомянутому первому способу, является код Манчестер-П. Форма биполярного сигнала при передаче кода Манчестер-П показана на рис. 3.2, в. Единица кодируется отрицательным перепадом сигнала в середине битового интервала, нуль – положительным перепадом. На границах битовых интервалов сигнал, если это необходимо, изменяет значение, готовясь к отображению очередного бита в середине следующего битового интервала.

С помощью кода Манчестер-П решаются сразу все отмеченные ранее проблемы. Поскольку число положительных и отрицательных импульсов на любом достаточно большом отрезке времени равно (отличается не более чем на один импульс, что не имеет значения), постоянная составляющая равна нулю.

Подстройка часов приемника или ретранслятора производится при передаче каждого бита, т. е. снимается проблема потери синхронизации при передаче длинных цепочек нулей или единиц.

Спектр сигнала содержит только две логические составляющие: F и $F/2$, где F – скорость передачи информационных битов. Наличие лишь двух (а не трех или более) электрических уровней сигнала позволяет надежно их распознавать (хорошая помехозащищенность).

Критерием ошибки может являться «замораживание» сигнала на одном уровне на время, превышающее время передачи одного информационного бита, поскольку независимо от передаваемого кода сигнал всегда «колеблется» и никогда не «замирает». Но за эти чрезвычайно полезные качества приходится платить расширением полосы пропускания связной аппаратуры. Поэтому код Манчестер-II широко используется там, где частотные ограничения не являются определяющими.

3.1.5. Код АМІ. Второй способ введения избыточности связан с добавлением дополнительных электрических уровней, в простейшем случае – третьего, «нулевого», уровня.

На рис. 3.2, г представлена форма сигнала с попеременной инверсией знака, так называемого АМІ сигнала (*ALteRnatiVe MaRk InVeRSion*). Нули кодируются отсутствием импульсов, а единицы - попеременно положительными и отрицательными импульсами. Постоянная составляющая сигнала АМІ равна нулю. Поэтому при передаче длинной последовательности единиц синхронизация не теряется. Обнаруживаются ошибки, нарушающие правильную последовательность знакочередующихся сигналов.

Синхронизация нарушается при передаче длинной последовательности нулей, как и в коде *NRZ*.

3.1.6. Коды BNZS, HDB3. Потеря синхронизации при передаче длинной последовательности нулей предотвращается так: цепочки нулей передатчик заменяет определенными «заготовками», которые представляют собой «отрезки» стандартных временных диаграмм. Коды АМІ, в которых цепочка из N нулей заменяется определенной подстановкой, называются BNZS-кодами (*BiPoLaR WITH N ZeRoeS SUBStITUtion*).

В коде *B3ZS* (рис.3.2, д) каждые три последовательных нуля подменяются либо комбинацией *B0V*, либо *00V*. Символ *B* обозначает импульс, который отвечает правилам кодирования АМІ, символ *V* – импульс, который нарушает правила кодирования АМІ (совпадает по полярности с предыдущим). Выбор одной из этих двух «заготовок» проводится так, чтобы, во-первых, число импульсов *B* между двумя последовательно расположенными импульсами *V* было нечетным, и, во-вторых, чтобы полярность импульсов *V* чередовалась.

В коде *B6ZS* (рис. 3.2, е) каждые шесть последовательных нулей подменяются комбинацией *0VB0VB*.

Коды BNZS получили широкое распространение в компьютерных сетях США и Канады: линии T1 – 1,544 Мбит/с, T1C – 3,152 Мбит/с, LD-4 - 274,176 Мбит/с, T4-274,176 Мбит/с. В странах Западной Европы широко используется код *HDB3* для работы на скоростях 2,048 и 8,448 Мбит/с. Этот код очень похож на BNZS, поскольку максимально допустимое число нулей, стоящих в цепочке, равно трем.

Каждые четыре последовательных нуля подменяются комбинацией *000V* либо *B00V*. Выбор той или иной комбинации проводится так, чтобы, во-первых, число импульсов *B* между двумя последовательными импульсами *V*

было нечетным, и, во-вторых, чтобы полярность импульсов V чередовалась (рис. 3.2, ж).

Существуют также другие распространенные коды, такие как CMI, PST, 4B3T и т. п. Все они являются разновидностями кодов AMI и созданы с целью минимизации требований к полосе пропускания каналов связи и увеличения обнаруживающей способности по отношению к ошибкам при передаче информации.

3.1.7. RZ – RetURn to ZeRo (возврат к нулю). Цифровые данные (рис. 3.3) представляются следующим образом:

- биты 0 представляются нулевым напряжением (0 В);
- биты 1 представляются значением $+U$ в первой половине и нулевым напряжением во второй, т.е. единице соответствует импульс напряжения продолжительностью передачи одного бита данных.

Этот способ имеет два преимущества по сравнению с кодированием NRZ:

- средний уровень напряжения в линии составляет $1/4U$ (вместо $1/2U$);
- при передаче непрерывной последовательности 1 сигнал в линии не остаётся постоянным.

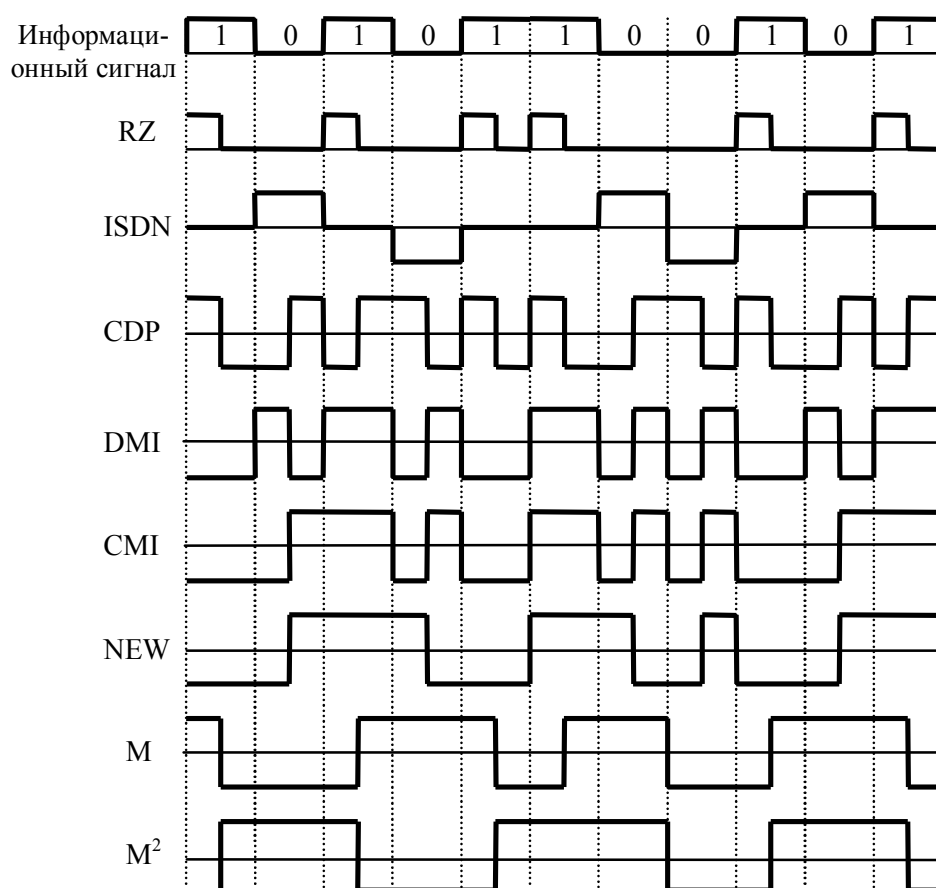


Рис. 3.3. Примеры линейных кодов

Однако при использовании кодирования RZ полоса сигнала может достигать значений, равных скорости передачи данных (при передаче последовательности 1).

3.1.8. CDP - ConDITional DiphaSe. Этот способ является комбинацией алгоритмов NRZI и PE и использует следующие представления битов цифрового потока:

- биты 0 представляются переходом напряжения в том же направлении, что и для предшествующего бита (от $+U$ к $-U$ или от $-U$ к $+U$);

- биты 1 представляются переходом напряжения в направлении, противоположном предшествующему биту (от $+U$ к $-U$ или от $-U$ к $+U$).

Такой код стандартизован для использования в отечественном стыке С1И (С1-ФЛ-БИ).

Этот способ обеспечивает формирование неполярного сигнала, который занимает достаточно широкую полосу.

Наиболее широкое распространение получили двухуровневые линейные коды с удвоением скорости передачи класса 1B2B (преобразование группы из одного двухуровневого символа в группу из двух двухуровневых символов), обладающие высокой помехозащищенностью, простотой преобразования и выделения тактовой частоты. Однако частота следования импульсов таких кодов, а, следовательно, и требуемая полоса частот передачи вдвое превышает частоту следования исходной двоичной последовательности. К таким кодам кроме перечисленных выше относятся коды DMI, CMI, NEW, код Миллера (M), M^2 , и ряд других менее популярных.

В последнее время к способам цифровой модуляции (впрочем, как и к аналоговой) предъявляются повышенные требования по эффективности использования задействованной полосы частот. Другими словами, такие способы должны обеспечивать передачу большего количества бит на 1 Бод и, следовательно, на 1 Гц используемой полосы. С такой точки зрения наиболее предпочтительно использование алфавитных кодов (mBnB, mBnT, mBnQ) с большими значениями отношения бит/Бод (m/n), например, 2B1Q.

3.1.9. Логический код 4B/5B. Для улучшения потенциальных кодов типа AMI, NRZI или 2B1Q используют другие избыточные логические коды. Логическое кодирование должно заменять длинные последовательности бит, приводящие к постоянному потенциалу в среде передачи данных, вкраплениями единиц. Как отмечалось выше, для логического кодирования характерны два метода – избыточные коды и скремблирование. Например, избыточный логический код 4B/5B, используемый в технологиях FDDI и FaSt EtheRnet, заменяет исходные символы длиной 4 бит на символы длиной в 5 бит. Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 битовых комбинации, в то время как исходные символы – только 16. Поэтому в результирующем коде можно отобрать 16 таких комбинации, которые не содержат большого количества нулей, а остальные считать **запрещенными кодовыми комбинациями**. Кроме устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Соответствие двоичного кода коду 4B/5B представлено в табл. 3.1.

Код $4B/5B$ передается по линии с помощью физического кодирования по одному из методов потенциального кодирования, чувствительному только к длинным последовательностям нулей. Символы кода $4B/5B$ длиной 5 бит гарантируют, что при любом их сочетании на линии не могут встретиться более трех нулей подряд. Буква B в названии кода означает, что элементарный сигнал имеет 2 состояния (от английского *biNaRY* – двоичный). Существуют коды и с тремя состояниями сигнала, например, в коде $8B/6T$ для кодирования 8 бит исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода $8B/6T$ выше, чем у кода $4B/5B$, так как на 256 исходных кодов приходится $3^6 = 729$ результирующих символов, поэтому 473 состояния считаются запрещенными.

Использование для перекодировки таблицы, аналогичной табл. 3.1, является простой операцией, поэтому это не усложняет сетевые адаптеры и интерфейсные блоки коммутаторов и маршрутизаторов.

Таблица 3.1

Соответствие двоичного кода коду $4B/5B$

Двоичный код	Код $4B/5B$	Двоичный код	Код $4B/5B$
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
1111	01111	1111	11101

Для обеспечения заданной пропускной способности линии передатчик, использующий избыточный код, должен работать с повышенной тактовой частотой. Так, для передачи кодов $4B/5B$ со скоростью 100 Мбит/с необходима тактовая частота передатчика 125 МГц. При этом спектр сигнала на линии расширяется по сравнению со случаем, когда по линии передается чистый, неизбыточный код. Тем не менее, спектр избыточного потенциального кода оказывается уже спектра манчестерского кода, что оправдывает дополнительный этап логического кодирования, а также работу приемника и передатчика на повышенной тактовой частоте.

3.2. Трехуровневое кодирование сигнала с гарантированным изменением уровней между соседними битовыми интервалами

Как следует из ранее сказанного, для надежного восстановления синхросигнала приемником желательно так закодировать данные, чтобы сигнал в линии изменялся как можно чаще, в идеальном случае - в каждом битовом интервале. Одно из таких решений с использованием трехуровневого кодирования сигнала предложено в [64]. Между двумя проводами линии может присутствовать отрицательное, нулевое или положительное напряжение или $U = -1$, $U = 0$, $U = +1$. Данное решение интересно тем, что созданы гарантии изменения уровня сигнала при переходе от одного битового интервала к другому независимо от вида передаваемой последовательности битов, что подтверждается временной диаграммой рис. 3.4.

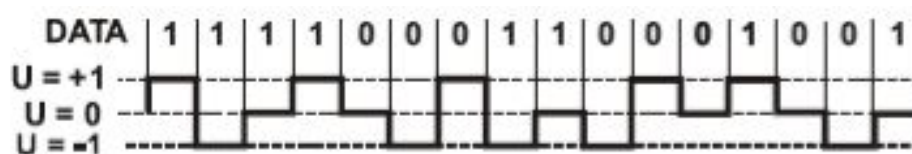


Рис. 3.4. Временная диаграмма сигнала в линии

В этой диаграмме встречаются все сочетания соседних битов (00, 01, 10, 11) и их однородные цепочки (1111 и 000). Тем не менее сигнал всегда изменяется при переходе от одного битового интервала к другому. На первый взгляд, неясно, каким образом достигнут столь примечательный результат.

Переходы передатчика между тремя возможными состояниями можно проследить по диаграмме, приведенной на рис. 3.5.



Рис. 3.5. Диаграмма состояний передатчика

Передатчик может находиться в трех состояниях, выделенных кружками. Эти состояния обозначены в соответствии с принятыми ранее сокращениями (см. рис. 3.4).

Стрелками обозначены переходы из одного состояния в другое. Цифра 0 или 1 около стрелки соответствует значению очередного бита *DATA*. Из рисунка следует, что при передаче цепочки битов 111...1 траектория переходов по диаграмме соответствует движению по часовой стрелке, а при передаче цепочки 000...0 – движению в обратном направлении. Передача случайных данных сопровождается «блужданием» между тремя состояниями. Существенно, что не бывает ситуаций, при которых одно и то же состояние повторяется в соседних тактах.

Декодирование сигналов в приемнике поясняется той же диаграммой, что и предыдущая, но с несколько иной интерпретацией событий (рис. 3.6).



Рис. 3.6. Диаграмма состояний приемника

Предположим, что ранее принятое и текущее состояния трехуровневого сигнала соответствуют показанным на рисунке. Непосредственный переход между этими состояниями возможен только по одному пути, который соответствует приему единичного бита. Поэтому на выходе логической схемы *L2* формируется сигнал $DATA' = 1$.

Особенность этой схемы кодирования - декодирования состоит в том, что при передаче цепочки битов вида 010101... все импульсы будут иметь одинаковую полярность, зависящую от предыстории. Это означает, что в сигнале появится постоянная составляющая, что для многих систем недопустимо. Чтобы избежать этого, можно применить скремблирование данных на входе передатчика и их дескремблирование на выходе приемника. Напомним, что применение этих операций позволяет получить псевдослучайный поток битов, в котором устранены нежелательные закономерности их чередования.

3.3. Кодирование РАМ-5

Разобравшись с тем, какие коды используются для представления данных, и рассмотрев методы улучшения самосинхронизирующих и спектральных свойств этих кодов, попробуем выяснить, достаточно ли этих мер, чтобы обеспечить передачу данных на скорости 1000 Мбит/с с использованием четырехпарного кабеля 5-й категории.

Как уже отмечалось, манчестерское кодирование обладает хорошими самосинхронизирующими свойствами и в этом смысле не требует каких-либо доработок, однако максимальная частота основной гармоники численно равна скорости передачи данных, то есть количеству переданных бит в секунду. Этого достаточно для передачи данных со скоростью 10 Мбит/с, так как кабель 3-й категории (а в стандарте 10BaSe-T может использоваться такой кабель) ограничен частотами в 16 МГц. Однако манчестерское кодирование не годится для передачи данных со скоростью 100 Мбит/с и выше.

Использование кода *NRZI* после дополнительной доработки с помощью избыточного блочного кода 4B/5B и скремблирования, а также трехпозиционного кода *MLT-3* (с целью уменьшения максимальной частоты основной гармоники) позволяет передавать данные со скоростью 100 Мбит/с по кабелю 5-й категории. Действительно, при использовании кода *MLT-3* максимальная частота основной гармоники численно равна одной четвертой от скорости передачи данных, то есть при скорости передачи 100 Мбит/с частота основной гармоники не превосходит 25 МГц, что вполне достаточно для кабеля 5-й категории. Однако такой способ не годится для передачи данных на скорости 1000 Мбит/с.

Поэтому в стандарте 1000BaSe-T используется принципиально иной способ кодирования. Для уменьшения тактовой частоты до величин, позволяющих передавать данные по витым парам категории 5, данные в линии представляются в так называемом коде PAM-5 (рис. 3.7). В нем передаваемый сигнал имеет набор из пяти фиксированных уровней $\{-2, -1, 0, +1, +2\}$. Четыре из них используются для кодирования информационных битов, а пятый предназначен для коррекции ошибок. На наборе из четырех фиксированных уровней одним дискретным состоянием сигнала можно закодировать сразу два информационных бита, поскольку комбинация из двух бит имеет четыре возможные комбинации (так называемые дибиты) — 00, 01, 10 и 11.

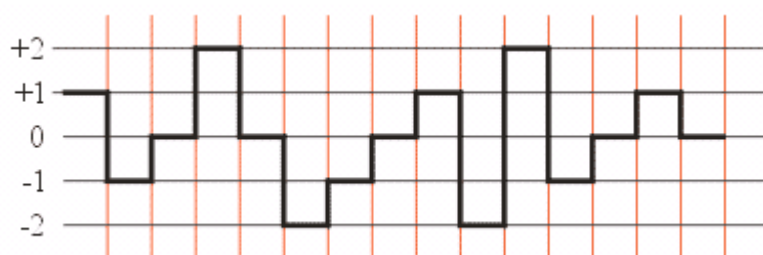


Рис. 3.7. Кодирование пятиуровневым сигналом PAM-5

Переход к дибитам позволяет в два раза повысить битовую скорость. Чтобы различать битовую, или информационную, скорость и скорость различных дискретных состояний сигнала, вводят понятие бодовой скорости. Бод — это количество различных дискретных состояний сигнала в единицу времени. Поэтому, если в одном дискретном состоянии кодируется два бита, битовая скорость в два раза больше бодовой, т. е. 1 Бод = 2 бит/с.

Если учесть, что кабель 5-й категории рассчитан на частоту 125 МГц, то есть способен работать с бодовой скоростью 125 МБод, то информационная скорость по одной витой паре составит 250 Мбит/с. Вспомним, что в кабеле имеется четыре витые пары, поэтому если задействовать все четыре пары, то можно повысить скорость передачи до $250 \text{ Мбит/с} \times 4 = 1000 \text{ Мбит/с}$, то есть достичь желаемой скорости.

Как уже отмечалось, в кодировании PAM-5 имеется пять дискретных уровней, однако для передачи дибитов используется только четыре уровня. Пятый избыточный уровень кода (*ForWARD ERRoR CoRRection*, *FEC*) используется для механизма построения коррекции ошибок. Он реализуется кодером Треллиса и декодером Витерби. Применение механизма коррекции ошибок позволяет увеличить помехоустойчивость приемника на 6 дБ.

3.4. Способы кодирования сигнала для уменьшения излучаемых помех при его передаче по витой паре проводов

3.4.1. Скремблирование полярностей импульсов. Передача сигнала по линии сопровождается излучением энергии в окружающее пространство. Наибольшему влиянию со стороны активной линии подвержены соседние линии многожильного кабеля. Это влияние проявляется в том, что в них появляются помехи, обусловленные в основном индуктивными и емкостными паразитными связями между линиями.

Энергия передаваемого по линии сигнала сосредоточена в некоторой спектральной полосе. Для уменьшения влияния на соседние линии желательно как можно более равномерно распределить энергию в этой полосе, без выраженных спектральных пиков. Если это условие выполнено, то источник сигнала можно грубо представить в виде бесконечно большого числа генераторов разной частоты, причем каждый генератор имеет бесконечно малую мощность. Результирующий сигнал помехи имеет характер шума.

Однако если источник формирует сигнал, близкий к периодическому, или, тем более, периодический, то на соседние линии вместо широкополосного шума действуют несколько сигналов или даже один сигнал, близкий по форме к синусоидальному. Так как основная энергия сигнала уже не распределена, а сосредоточена в нескольких или одной пиковой спектральной составляющей, то амплитуда помех может превысить допустимую. Таким образом, для уменьшения амплитуды помех, наводимых на соседние линии, следует по возможности исключить из передаваемого сигнала выраженные периодические компоненты.

Эти компоненты могут появляться, например, в сигналах *AMI*, *T1* или *MLT-3* при передаче длинной последовательности лог. 1, как показано затененными областями на рис. 3.8.

В этих областях невооруженным глазом просматриваются прообразы синусоидальных сигналов, несущих основную энергию. Периоды сигналов *AMI*

и T_1 при передаче длинной последовательности лог. 1 равны двум битовым интервалам. Период сигнала $MLT-3$ равен четырем битовым интервалам.

Длинные последовательности лог. 1 можно «разрушить» применением скремблирования, т. е. особой шифрации данных, после которой любые исходные последовательности выглядят как случайные (см. п. 3.4). Для восстановления исходных данных приемник должен выполнить обратную операцию (дескремблирование). При этом необходима синхронная работа шифратора и дешифратора, что несколько усложняет задачу.

Предлагаемое в [18] решение также предусматривает разрушение периодического сигнала при передаче длинной последовательности лог. 1, но выполняется оно иначе. Скремблируются не данные, а полярности передаваемых по линии импульсов. В зависимости от значения некоторого псевдослучайного бита выбирается либо положительная, либо отрицательная полярность. Приемник безразличен к полярности импульса и реагирует только на его наличие. Поэтому для восстановления данных приемнику не нужно знать вид псевдослучайной последовательности, использованной при шифрации полярностей! Иными словами, осуществляется некое «скремблирование без последующего дескремблирование» (что на первый взгляд представляется лишним смыслом). В итоге упрощается аппаратура, предназначенная для уменьшения излучаемых помех.

Чтобы перейти к существу вопроса, рассмотрим временные диаграммы, приведенные на рис. 3.8 более подробно.

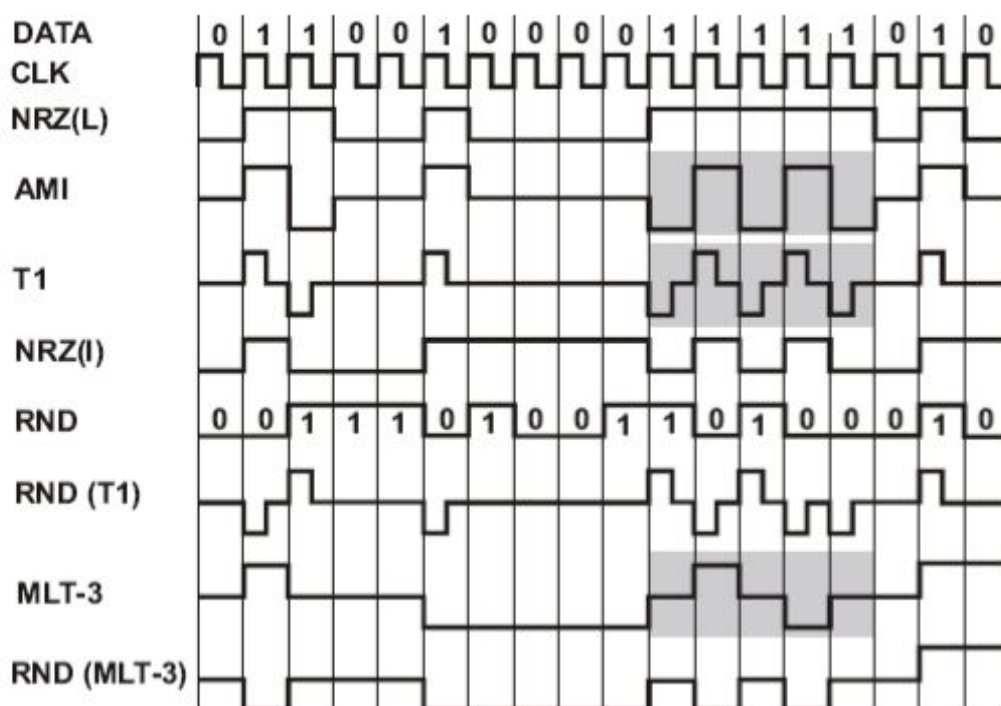


Рис. 3.8. Временные диаграммы передачи данных *DATA* с использованием различных кодов; *RND* - сигнал на выходе генератора псевдослучайной последовательности битов

Как уже отмечалось в п. 3.1, код NRZ (в данном случае он обозначен как $NRZ(L)$) отображает лог. 0 и лог. 1 соответственно низким и высоким уровнями напряжения. В коде AMI лог. 0 отображается отсутствием напряжения, а лог. 1 – положительным или отрицательным импульсом, причем полярности соседних импульсов чередуются. Код $T1$ отличается от AMI длительностью импульса.

В коде $NRZ(I)$ любой фронт сигнала несет информацию о том, что замыкающий к нему справа битовый интервал соответствует лог. 1. Если фронта нет, то битовый интервал отображает лог. 0.

Код $MLT-3$ можно получить из кода $NRZ(I)$ следующим образом. В интервалах, где код $NRZ(I)$ принимает нулевое значение, код $MLT-3$ также должен быть нулевым. Положительные импульсы кода $NRZ(I)$ должны соответствовать знакопереключающимся импульсам кода $MLT-3$. При этом не имеет значения, какую полярность имеет первоначальный импульс.

Схема преобразования кода $NRZ(L)$ в коды $NRZ(I)$ и $MLT-3$ приведена на рис. 3.9, а. Каждый из двух последовательно соединенных D -триггеров включен в режиме делителя частоты. На выходе Q первого триггера формируется код $NRZ(I)$. На входы передатчика подаются сигналы «+» и «-», которые преобразуются соответственно в положительные и отрицательные импульсы трехуровневого сигнала $MLT-3$.

Строго говоря, в эту и последующие схемы нужно ввести компенсирующие элементы для предотвращения некорректных ситуаций – так называемых «гонок» или «состязаний» сигналов. Пример гонки: из-за того, что второй триггер изменяет состояние и опрашивается под действием одного и того же сигнала $NRZ(I)$, на выходах «+» и «-» элементов И в процессе переключения триггера будут наблюдаться кратковременные ложные импульсы. Но на эти «мелочи» сейчас не будем обращать внимания, чтобы не усложнять рисунки и не потерять основную идею реализации скремблирования полярностей импульсов.

Схема, показанная на рис. 3.9, б, отличается от предыдущей тем, что на D -вход второго триггера (первый триггер не показан) подается псевдослучайная последовательность битов RND . При $RND = 1$ в момент формирования положительного фронта сигнала $NRZ(I)$ выбирается положительная полярность импульса в линии, при $RND = 0$ – отрицательная. Последовательность битов RND синхронизирована сигналом CLK и формируется, например, генератором на основе сдвигового регистра с логическими элементами Иключающее ИЛИ в цепях обратных связей [18]. Такое решение приводит к случайному чередованию полярностей импульсов кода $RND(MLT-3)$ в отличие от их регулярного чередования в коде $MLT-3$. Схема формирования сигнала $RND(TL)$, показанная на рис. 3.9, в, построена аналогично и отличается наличием дополнительного логического элемента И, предназначенного для укорочения положительных импульсов кода $NRZ(I)$.

Схема, представленная на рис. 3.9, г, позволяет дешифровать коды $MLT-3$ или $RND(MLT-3)$, т. е. преобразовывать их в обычный код $NRZ(L)$. На

выходе приемника формируются положительные импульсы «+» и «-», которые соответствуют разнополярным входным сигналам. Приемник также формирует синхросигнал CLK , например, с помощью генератора с фазовой автоподстройкой частоты.

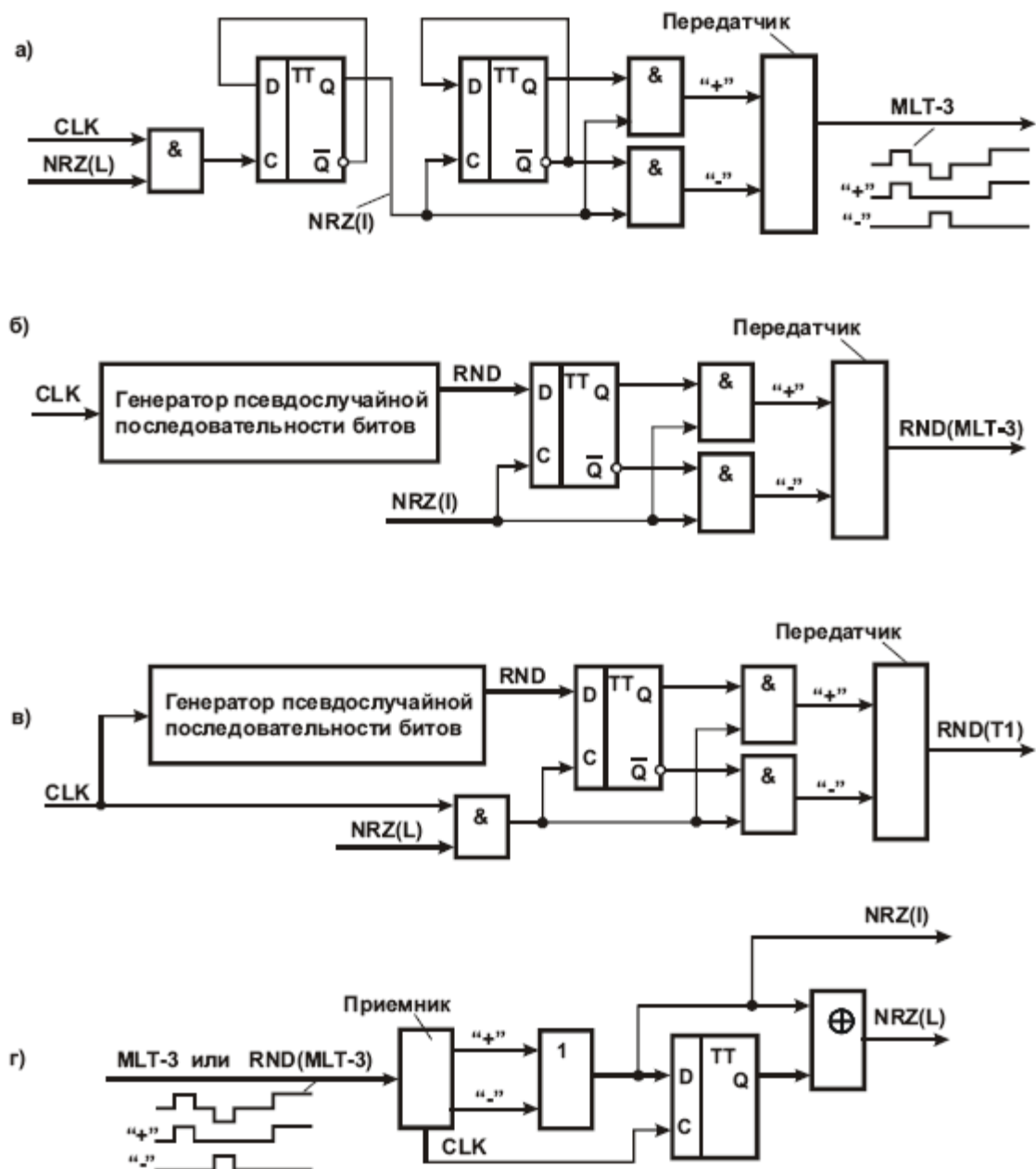


Рис.3.9. Упрощенные схемные решения: а - формирователь кодов $NRZ(I)$, $MLT-3$; б-формирователь кода $RND(MLT-3)$ с псевдослучайным чередованием полярностей импульсов; в - формирователь кода $RND(TL)$ с псевдослучайным чередованием полярностей импульсов; г - дешифратор кода $MLT-3$ или $RND(MLT-3)$

Логический элемент ИЛИ суммирует импульсы «+» и «-», так что их первоначальная полярность не учитывается. В этом, пожалуй, и заключена основная предпосылка создания рассмотренного решения: полярность импульсов в линии может быть произвольной, так как приемник не обращает на нее внимания. А если это так, то можно случайным образом распределить полярности передаваемых импульсов и тем самым подавить периодические составляющие сигнала. Единственное ограничение состоит в том, что для исключения постоянной составляющей сигнала в линии среднее число положительных и отрицательных импульсов в любом достаточно большом интервале времени должно быть одинаковым. Это условие в данном случае выполнено.

3.4.2. Двубинарное кодирование. Еще одно решение задачи уменьшения уровня излучаемых помех основано на применении двубинарного кодирования.

В схеме, показанной на рис. 3.10, потребитель данных находится на некотором удалении от оптоволоконной линии связи. Для приема данных потребителю выделена витая пара проводов в многожильном кабеле (рассматриваем только одно направление передачи). На выходе интерфейса *FDDI* (*FIbeR DIStribUteD DATA InteRFace* - распределенный интерфейс передачи данных по волоконно-оптическим каналам) данные представлены кодом *NRZ(I)* и сопровождающим его синхросигналом *CLK* (см. рис. 3.8).

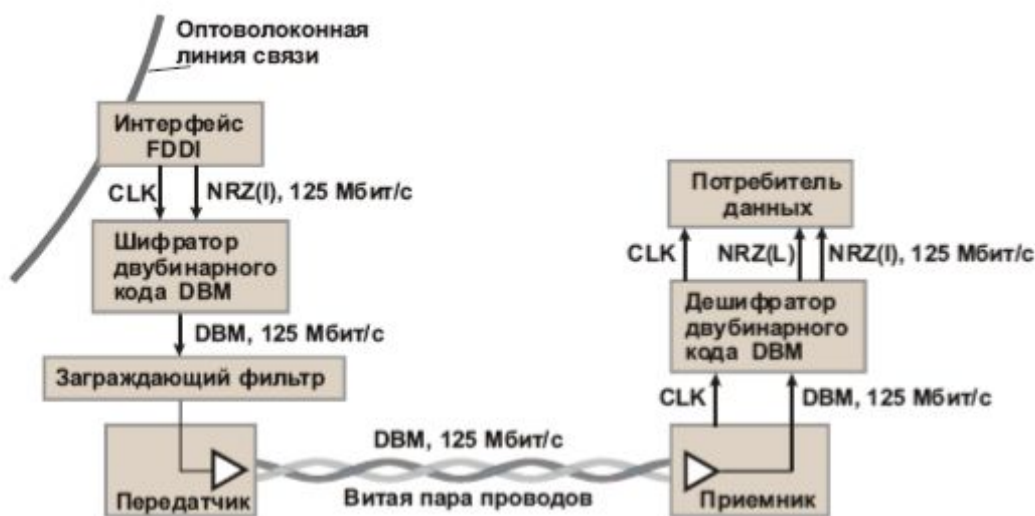


Рис. 3.10. Схема высокоскоростной передачи данных в двубинарном коде с использованием витой пары проводов

Проблема заключается в том, что непосредственная передача сигнала *NRZ(I)* со скоростью 125 Мбит/с по витой паре проводов создает повышенный уровень помех на соседних жилах кабеля. Ситуация усугубляется в отсутствие полезных данных, когда передается заполняющая паузу непрерывная последовательность лог. 1. Эта последовательность соответствует частоте сигнала *NRZ(I)*, равной половине скорости передачи данных или 62,5 МГц. На

этой частоте сигнал легко преодолевает паразитные емкостные и индуктивные связи и наводится на соседние провода кабеля. Поэтому следовало бы применить какой-либо дополнительный способ кодирования для снижения частоты сигнала в отсутствие данных и разравнивания его спектра при наличии данных. Рассмотренное далее трехуровневое двубинарное кодирование *DBM* (*DUobInaRY moDULatIon*) и включение заграждающего фильтра позволяют в значительной мере снизить уровень излучаемых помех. По способу построения код *DBM* во многом схож с описанными в п. 3.4.1 кодами *MLT-3* и *RND(MLT-3)*.

Как показано на рис. 3.10, код *NRZ(I)* с выхода интерфейса *FDDI* преобразуется шифратором в код *DBM*. Сигнал с выхода шифратора проходит через заграждающий *R-L-C*-фильтр, разравнивающий спектр сигнала, передатчик и по линии связи (витой паре проводов) поступает в приемник. Приемник выделяет из него синхросигнал *CLK* и данные, представленные в коде *DBM*. Дешифратор кода *DBM* формирует коды *NRZ(I)* и *NRZ(L)*. Скорость передачи данных во всем тракте постоянна и равна 125 Мбит/с.

Шифратор двубинарного кода (рис. 3.11) [18] содержит инвертор, логический элемент Иключающее ИЛИ (XOR), тактируемый элемент Т задержки, дешифратор *DC* со структурой 2х4, элемент ИЛИ, электронные ключи *SWM1* - *SWM3* и два источника *U1* и *U2* постоянного напряжения. Временные диаграммы формирования кода *DBM* показаны на рис. 3.12.

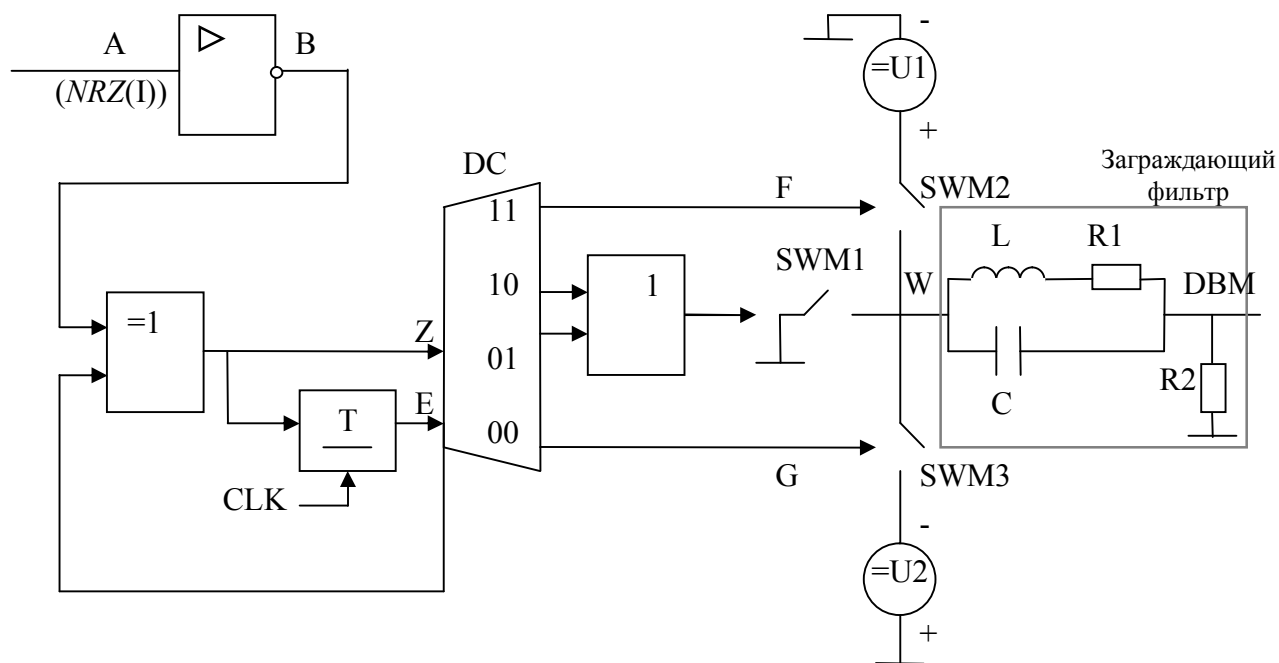


Рис. 3.11. Схема шифратора двубинарного кода *DBM* и структура заграждающего фильтра

Входной сигнал A инвертируется и поступает на первый вход элемента XOR . Сигнал Z с выхода этого элемента задерживается на один период сигнала CLK (например, с помощью D -триггера) и подается на второй вход элемента XOR . Дешифратор DC в зависимости от сочетания сигналов Z и E формирует сигнал на одном из четырех выходов. При $Z = E = 0$ сигнал $G = 1$ замыкает ключ $SWM3$, поэтому на выход W шифратора поступает отрицательное напряжение от источника $U2$. При Z ПЕ сигнал $J = 1$ замыкает ключ $SWM1$, на выход шифратора поступает нулевое напряжение. При $Z = E = 1$ сигнал $F = 1$ замыкает ключ $SWM2$, на выход шифратора поступает положительное напряжение от источника $U1$.

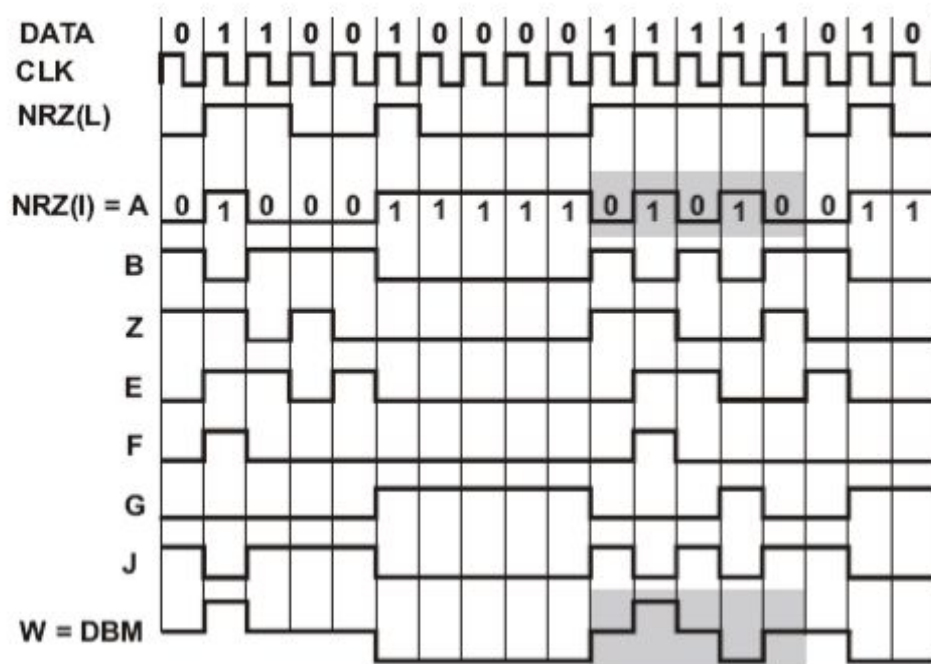


Рис. 3.12. Временные диаграммы формирования двубинарного кода DBM

Процесс шифрации удобно проследить с помощью диаграммы состояний, приведенной на рис. 3.13.

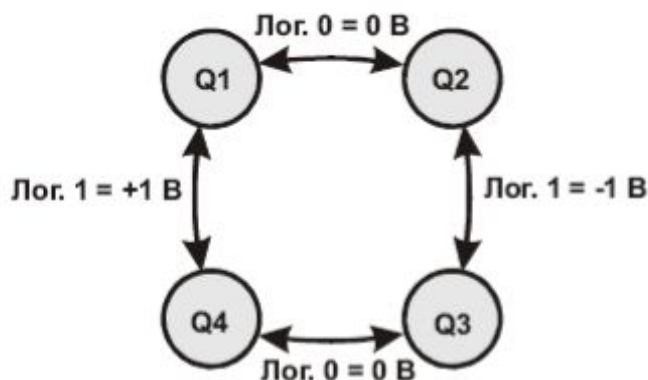


Рис. 3.13. Диаграмма состояний шифратора двубинарного кода DBM

Шифратор может находиться в одном из четырех состояний $Q_L - Q_4$. Если, например, шифратор пребывает в состоянии Q_1 , то при поступлении на вход A сигнала лог. 1 на его выходе W формируется положительное напряжение $+1$ В (величина условная). Этот факт отражен обозначением «Лог.1 = $+1$ В» около двунаправленной связи между узлами Q_1 и Q_4 . В этой ситуации шифратор переходит в состояние Q_4 .

Если шифратор находится в состоянии Q_1 , то при поступлении на вход A сигнала лог. 0 на его выходе W формируется нулевое напряжение 0 В. Этот факт отражен обозначением «Лог. 0 = 0 В» около двунаправленной связи между узлами Q_1 и Q_2 . В данной ситуации шифратор переходит в состояние Q_2 . Переходы между состояниями Q_2 и Q_3 возможны при поступлении на вход A сигналов лог. 1, но эти переходы сопровождаются выдачей отрицательного напряжения (-1 В) на выход W . Переходы между состояниями Q_3 и Q_4 возможны при поступлении на вход A шифратора сигналов лог. 0.

Из диаграммы состояний следует, что если на вход A подана последовательность лог. 0, то шифратор последовательно переходит из состояния Q_1 в состояние Q_2 и обратно либо из состояния Q_3 в состояние Q_4 и обратно. Эти ситуации внешне неразличимы, так как на выходе шифратора в любом случае сформировано нулевое напряжение. Если на вход A подана последовательность лог. 1, то шифратор последовательно переходит из состояния Q_1 в состояние Q_4 и обратно либо из состояния Q_2 в состояние Q_3 и обратно. Эти ситуации различаются полярностью выходного напряжения.

Если на вход A подана последовательность $\dots 010101\dots$, то шифратор последовательно циклически проходит все состояния в направлении по часовой или против часовой стрелки в зависимости от начальных условий. Нулевые биты отображаются нулевым напряжением, единичные - попеременно положительным и отрицательным.

В общем случае данные кодируются следующим образом. Нулевые биты ($A = 0$) отображаются нулевым напряжением ($W = 0$ В), единичные - положительным или отрицательным в соответствии со следующими правилами.

Правило 1. При нечетном числе нулевых битов между двумя единичными (например, в коде $\dots 10001\dots$) полярности импульсов, отображающих единичные биты, взаимно-обратны ($\dots -000+ \dots$ или $\dots +000- \dots$).

Правило 2. При четном числе нулевых битов между двумя единичными (например, в коде $\dots 1001\dots$) полярности импульсов, отображающих единичные биты, одинаковы ($\dots -00- \dots$ или $\dots +00+ \dots$).

Правило 3. В группе единичных битов ($\dots 111\dots$) сигналы имеют одинаковую полярность ($\dots +++ \dots$ или $\dots --- \dots$).

В соблюдении приведенных правил можно убедиться при сопоставлении временных диаграмм сигналов A и W на рис. 3.12. Из этих диаграмм также следует, что при передаче непрерывной последовательности лог.1 ($DATA = 11\dots 1$) частота основной гармоники сигнала $NRZ(I)$ равна половине скорости передачи данных или 62,5 МГц. При этих же условиях частота основной гармоники сигнала DBM равна четверти скорости передачи данных или 31,25

МГц. (Интересующие нас области временных диаграмм выделены серым фоном.) Амплитуда этой гармоники достаточно высока по сравнению с остальными, поэтому без заметного искажения формы сигнала ее можно несколько снизить с помощью заграждающего фильтра.

Заграждающий фильтр настроен на частоту 31,25 МГц. Значения емкости и индуктивности удовлетворяют соотношению $LC = 2,6 \times 10^{-17}$. Например, при $L = 2,6$ мкГн $C = 10$ пФ. Резонансный импеданс цепи $R1-L-C$ равен $Z_F = L/R1C$. Коэффициент подавления сигнала на резонансной частоте равен $(ZF + R2)/R2$ и может регулироваться выбором параметров фильтра.

Двубинарное кодирование с фильтрацией выходного сигнала позволяет сместить его энергетический спектр в область более низких частот по сравнению с другими решениями. Так, 78% энергии сигнала сосредоточено в полосе частот ниже 30 МГц, а 90% энергии - в полосе частот ниже 42,6 МГц. Напомним, что скорость передачи данных составляет 125 Мбит/с!

Дешифратор двубинарного кода (см. рис. 3.10) можно выполнить по схеме, приведенной на рис. 3.9, г. Эта схема нечувствительна к полярности импульсов и в равной мере применима для дешифрации кодов *MLT-3*, *RND(MLT-3)* и *DBM*.

3.5. Передача данных с использованием скремблера и дескремблера

Скремблирование может выполняться с различными целями. Наиболее распространенная цель - защита передаваемых данных от несанкционированного доступа. Для ее достижения разработано множество методов кодирования и схемных решений. Но нас интересует иная задача, связанная с «разравниванием» спектра сигнала и повышением надежности синхронизации приемника с источником передаваемых по линии данных. Применительно к этой задаче цель скремблирования состоит в исключении из потока данных длинных последовательностей лог. 0, лог. 1 и периодически повторяющихся групп битов. Для этого необходимо преобразовать данные так, чтобы они выглядели как случайные, т. е. лишенные какой-либо видимой закономерности.

Обычно скремблирование осуществляется непосредственно перед модуляцией.

Скремблирование (от англ. слова *to SCRAMbLe* – перемешивать) производится на передающей стороне с помощью устройства – **скремблера**, реализующего логическую операцию суммирования по модулю 2 исходного и преобразующего псевдослучайного двоичных сигналов. Например, скремблер может реализовать соотношение:

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}, \quad (3.1)$$

где B_i – двоичная цифра результирующего кода, полученная на I -м такте работы скремблера, A_i – двоичная цифра исходного кода, поступающая на I -м

такте на вход скремблера; B_{i-3} и B_{i-5} – двоичные цифры результирующего кода, полученные на предыдущих тактах работы скремблера, соответственно на 3 и на 5 тактов ранее текущего такта; $\oplus$ – операция исключающего ИЛИ (сложение по $\text{mod } 2$).

Например, для исходной последовательности 110110000001 скремблер даст следующий результирующий код:

$B_1 = A_1 = 1$ (первые три цифры результирующего кода будут совпадать с исходным, так как на вход ещё не поступили необходимые цифры)

$$\begin{aligned} B_2 &= A_2 = 1; \\ B_3 &= A_3 = 0; \\ B_4 &= A_4 \oplus B_1 = 1 \oplus 1 = 0; \\ B_5 &= A_5 \oplus B_2 = 1 \oplus 1 = 0; \\ B_6 &= A_6 \oplus B_3 \oplus B_1 = 0 \oplus 0 \oplus 1 = 1; \\ B_7 &= A_7 \oplus B_4 \oplus B_2 = 0 \oplus 0 \oplus 1 = 1; \\ B_8 &= A_8 \oplus B_5 \oplus B_3 = 0 \oplus 0 \oplus 0 = 0; \\ B_9 &= A_9 \oplus B_6 \oplus B_4 = 0 \oplus 1 \oplus 0 = 1; \\ B_{10} &= A_{10} \oplus B_7 \oplus B_5 = 0 \oplus 1 \oplus 0 = 1; \\ B_{11} &= A_{11} \oplus B_8 \oplus B_6 = 0 \oplus 0 \oplus 1 = 1; \\ B_{12} &= A_{12} \oplus B_9 \oplus B_7 = 1 \oplus 1 \oplus 1 = 1. \end{aligned}$$

Таким образом, на выходе скремблера появится последовательность 110001101111, в которой нет шести нулей подряд, присутствовавших в исходном коде. На приемной стороне осуществляется обратная операция – дескремблирование устройством, называемым **дескремблером**. Дескремблер выделяет из принятой исходную последовательность на основании обратного соотношения

$$C_i = B_i \oplus B_{i-3} \oplus B_{i-5} = (A_i \oplus B_{i-3} \oplus B_{i-5}) \oplus B_{i-3} \oplus B_{i-5} = A_i. \quad (3.2)$$

Различные алгоритмы скремблирования отличаются количеством слагаемых, которые определяют цифру результирующего кода, и сдвигом между слагаемыми. Основной частью скремблера является генератор псевдослучайной последовательности (ПСП) в виде линейного n -каскадного регистра с обратными связями, формирующий последовательность максимальной длины $2^n - 1$.

Различают два основных типа скремблирования: самосинхронизирующееся (СС) и с установкой (аддитивное).

3.5.1. Самосинхронизирующиеся скремблер и дескремблер. Особенностью СС скремблера (рис. 3.14) является то, что он управляется скремблированной последовательностью, т.е. той, которая передается в канал. Поэтому при данном виде скремблирования не требуется специальная установка со-

стояний скремблера и дескремблера; скремблированная последовательность записывается в регистры сдвига скремблера и дескремблера, устанавливая их в идентичное состояние. При потере синхронизма между скремблером и дескремблером время восстановления синхронизма не превышает числа тактов, равного числу ячеек регистра скремблера.

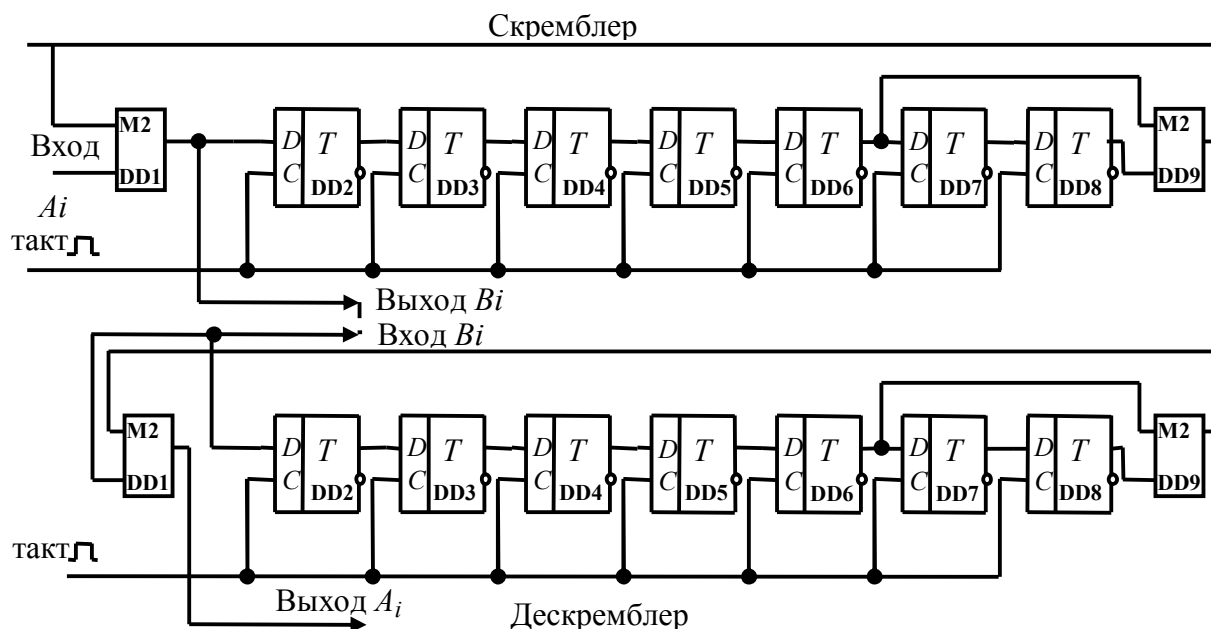


Рис. 3.14. Функциональная схема самосинхронизирующихся скремблера и дескремблера

На приемном конце выделение исходной последовательности происходит путем сложения по модулю 2 принятой скремблированной последовательности с ПСП регистра. Например, для схемы рис. 3.14 входная последовательность A_i помощью скремблера в соответствии с соотношением $B_i = A_i \oplus (B_{i-5} \oplus B_{i-7})$ преобразуется в посылаемую двоичную последовательность B_i . В приёмнике из этой последовательности таким же регистром сдвига, как на передаче формируется последовательность $A_i = (B_{i-5} \oplus B_{i-7}) \oplus B_i$. Эта последовательность на выходе дескремблера идентична первоначальной последовательности.

Как следует из принципа действия схемы, при одной ошибке в последовательности B_i ошибочными получаются также последующие пятый и седьмой символы (в данном примере). В общем случае влияние ошибочно принятого бита будет сказываться $(A+1)$ раз, где A – число обратных связей. Таким образом, СС скремблер обладает свойством размножения ошибок. Данный недостаток ограничивает число обратных связей в регистре сдвига; практически это число не превышает двух. Второй недостаток СС скремблера связан с возможностью появления на его выходе при определенных условиях, так на-

зываемых критических ситуаций, когда выходная последовательность приобретает периодический характер с периодом, меньшим длины ПСП. Чтобы предотвратить это, в скремблере и дескремблере согласно рекомендациям *ITU-T* предусматриваются специальные дополнительные схемы контроля, которые выявляют наличие периодичности элементов на входе и нарушают ее.

3.5.2. Адаптивные скремблер и дескремблер. Недостатки, присущие СС скремблеру, практически отсутствуют при аддитивном скремблировании (рис. 3.15), однако здесь **требуется** предварительная идентичная установка состояний регистров скремблера и дескремблера. В скремблере с установкой (АД скремблере) производится суммирование входного сигнала и ПСП, но результирующий сигнал не поступает на вход регистра. В дескремблере скремблированный сигнал также не проходит через регистр сдвига, поэтому размножения ошибок не происходит.

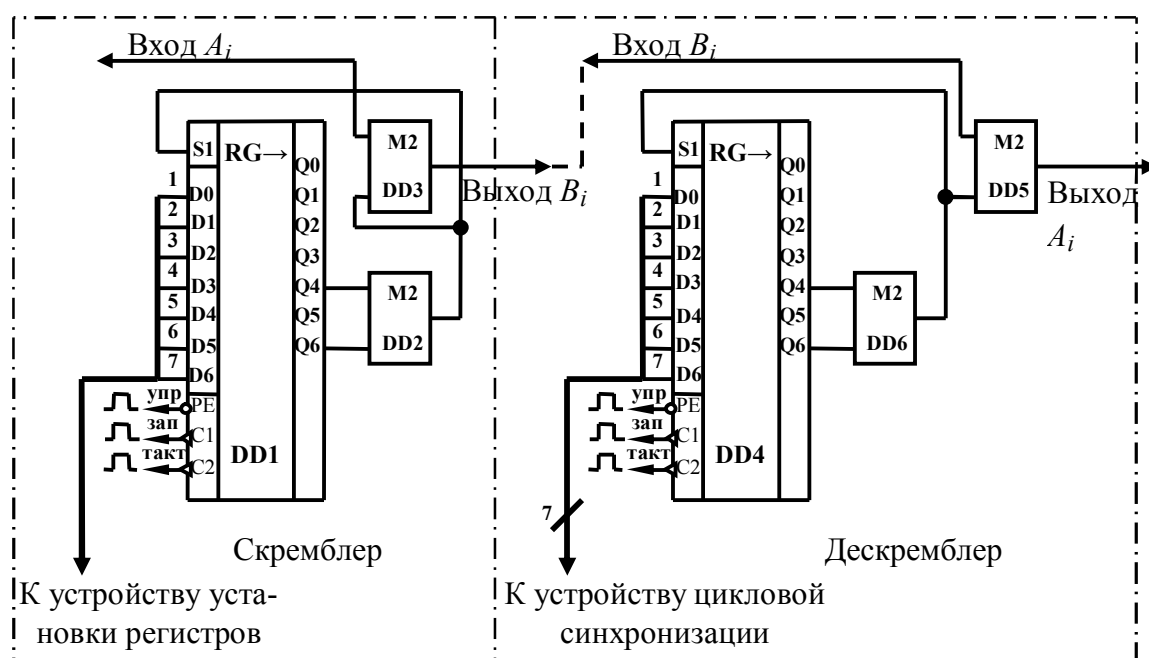


Рис. 3.15. Функциональная схема аддитивных скремблера и дескремблера

Суммируемые в скремблере последовательности независимы, поэтому их период всегда равен наименьшему общему кратному длительности периодов входной последовательности и ПСП, и критическое состояние отсутствует. Отсутствие эффекта размножения ошибок и необходимости в специальной логике защиты от нежелательных ситуаций делают способ аддитивного скремблирования предпочтительнее, если не учитывать затраты на решение задачи фазирования скремблера и дескремблера. В качестве сигнала предварительной установки используют сигнал цикловой синхронизации.

3.5.3. Улучшенный вариант самосинхронизирующихся скремблера и дескремблера. Рассмотрим улучшенный вариант скремблера – дескремблера, построенного на основе двух одинаковых генераторов псевдослучайных последовательностей битов, рис. 3.16 [18]. Улучшение состоит в устранении упоминавшихся в п. 3.5.1 неблагоприятных кодовых ситуаций. В отличие от схемы, приведенной на рис. 3.14, применены средства коррекции состояний генераторов для устранения нежелательных последовательностей битов.

Скремблер содержит сдвиговый регистр $RG1$ с логическими элементами Иключающее ИЛИ ($DD1$ и $DD2$) в цепи обратной связи, а также два двоичных счетчика.

Счетчик лог. 0 устанавливается в нуль всякий раз, когда скремблированный сигнал данных $SCRD = 1$. Если $SCRD = 0$, то содержимое счетчика увеличивается на единицу по фронту сигнала $CLK1$. При накоплении заданного числа единиц (например, пяти) счетчик автоматически устанавливается в нулевое состояние и формирует импульс SET установки в единицу некоторого разряда (или группы разрядов) сдвигового регистра. Таким образом, счетчик лог. 0 служит детектором цепочек лог. 0 заданной длины. При обнаружении такой цепочки корректируется код в сдвиговом регистре.

Счетчик лог. 1 построен симметрично. Он устанавливается в нуль всякий раз, когда скремблированный сигнал данных $SCRD = 0$. Если $SCRD = 1$, то содержимое счетчика увеличивается на единицу по фронту сигнала $CLK1$. При накоплении заданного числа единиц (например, пяти) счетчик автоматически устанавливается в нулевое состояние и формирует импульс $RESET$ установки в нуль некоторого разряда (или группы разрядов) сдвигового регистра. Счетчик лог. 1 служит детектором цепочек лог. 1 заданной длины.

Дескремблер построен аналогично. Он дополнительно содержит схему выделения синхросигнала $CLK2$ из скремблированного сигнала $SCRD$. Эта схема может быть выполнена на основе петли фазовой автоподстройки частоты PLL (PhaSe LockeD Loop).

Система передачи данных функционирует следующим образом. Источник данных формирует синхронный битовый поток SD и соответствующий синхросигнал $CLK1$. Этот поток проходит через логический элемент $DD2$. На второй вход этого элемента поступает последовательность скремблирующих битов $SC1$. Суммарный (скремблированный) поток $SCRD$ передается по линии и поступает в дескремблер.

После заполнения регистра $RG2$ информация в нем в точности совпадает с той, которая присутствует в регистре $RG1$. В дальнейшем все изменения информации в этих регистрах происходят синхронно, так как на их входы подается один и тот же сигнал $SCRD$ (разумеется, с учетом задержки передачи по линии связи). Благодаря этому, $SC2 = SC1$. Логический элемент $XOR4$ формирует сигнал принимаемых данных RD , который повторяет исходный сигнал SD . Это следует из того, что

$$RD = SCRD \oplus SC2 = SCRD \oplus SC1 = SD \oplus SC1 \oplus SC1 = SD.$$

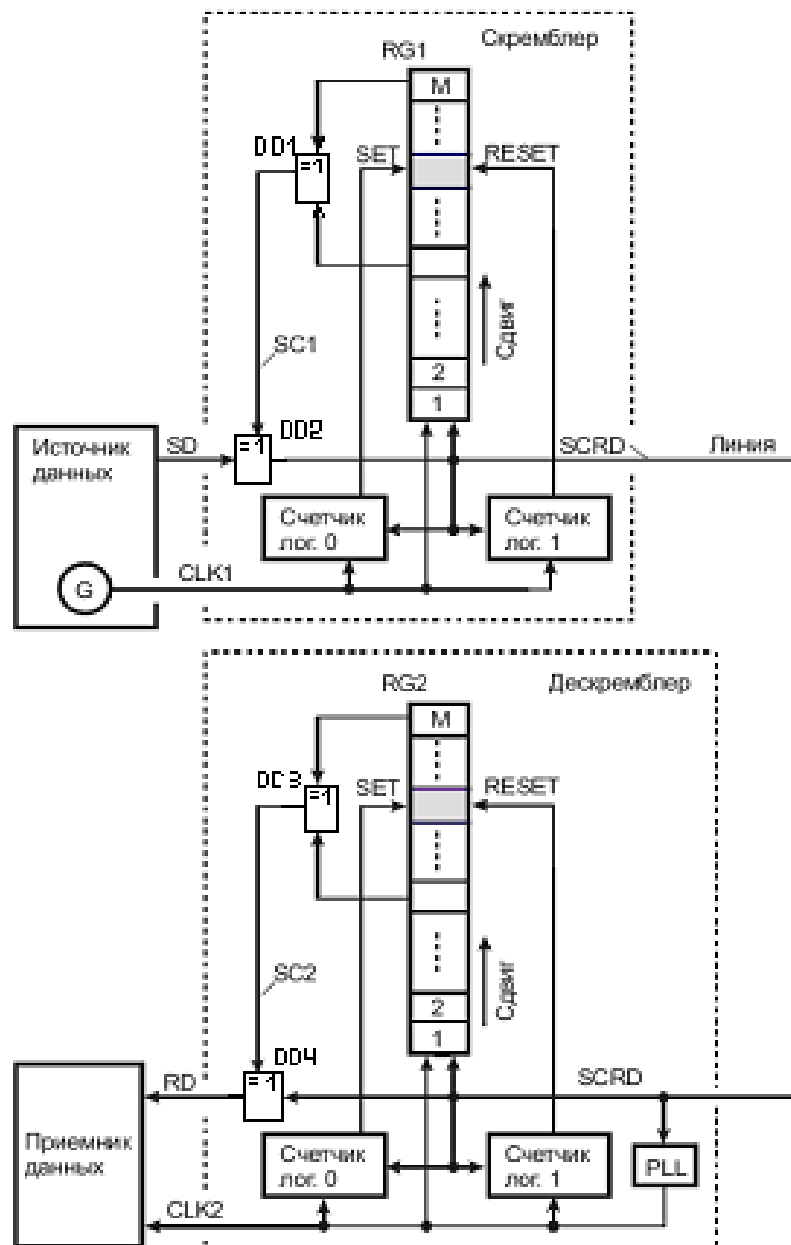


Рис. 3.16. Система передачи данных, в которой скремблер и дескремблер содержат неизолированные генераторы псевдослучайных битовых последовательностей (улучшенный вариант)

Уточним роль счетчиков лог. 0 и лог. 1, о которых уже кратко упоминалось. Предположим, что эти счетчики исключены из схем скремблера и дескремблера. Схема остается работоспособной при условии, что поток *SD* не содержит некоторых опасных последовательностей сигналов. Рассмотрим эти последовательности.

При работе системы не исключено, что поступающие от источника данные *SD* таковы, что логический элемент *DD2* скремблера в *M* последовательных тактах сформирует сигнал лог. 0 (*M*-разрядность сдвигового регистра). Тогда сдвиговый регистр *RG1* (а синхронно с ним и регистр *RG2*) заполнится ну-

левыми битами. Если после этого источник сигнала начнет передавать длинную последовательность лог. 0, то на обоих входах логического элемента *DD2* будут постоянно присутствовать нулевые сигналы, сигнал *SCRD* также в течение длительного времени будет оставаться нулевым, что крайне нежелательно.

Аналогичная ситуация возможна и после случайного заполнения сдвигового регистра единичными битами. При последующей передаче длинной последовательности сигналов $SD = 1$ на выходе логического элемента *DD2* поддерживается сигнал лог. 1, который в каждом такте записывается в регистр, подтверждая его состояние «Все единицы».

Введение счетчиков позволяет исключить возможность заполнения регистра *RG1* одинаковыми битами (лог. 0 или лог. 1). Поэтому нет опасности фиксации уровня сигнала в линии при последующей выдаче источником данных длинной последовательности лог. 0 или лог. 1. Но это, к сожалению, не означает, что задача получения гарантированно изменяющегося сигнала *SCRD* решена «полностью и окончательно». Действительно, теоретически можно преднамеренно синтезировать сколь угодно длинную последовательность сигналов *SD*, совпадающую или противофазную последовательности сигналов *SC1*, какой бы сложной она ни была (ведь ее можно заранее вычислить, зная структуру скремблера и его начальное состояние). В результате такого синтеза получим неизменный сигнал *SCRD* на протяжении любого желаемого интервала времени! Точно так же можно было бы синтезировать периодический сигнал *SCRD* вида 010101... для создания максимального уровня перекрестных помех в соседних проводах многожильного кабеля (например, с целью тестирования системы). Но так как начальное состояние регистра *RG1* источнику данных не известно, на практике такой синтез невозможен.

Вероятность случайного формирования нескремблируемых последовательностей битов источником данных зависит от разрядности скремблера и может быть небольшой, но с ней нельзя не считаться при проектировании телекоммуникационных устройств.

3.5.4. Синхронизация адаптивных скремблера и дескремблера. В системе передачи данных, показанной на рис. 3.17, применены изолированные генераторы псевдослучайных битовых последовательностей. Их синхронизация осуществляется с использованием аппаратных и программных средств приемной стороны.

К этим средствам относятся мультиплексор *MUX* и программно-управляемый выход приемника данных, на котором формируется сигнал *F*. При нормальной работе системы приемник данных постоянно поддерживает на выходе сигнал $F = 0$. На выход мультиплексора транслируется сигнал *Z2*, генератор псевдослучайной битовой последовательности на основе регистра *RG2* изолирован от внешних воздействий. Поэтому схема эквивалентна рассмотренной ранее (см. рис. 3.15).

Предположим теперь, что в исходном состоянии дескремблер не синхронизирован со скремблером. Такая ситуация может возникнуть, например, после включения напряжения питания аппаратуры приемной стороны, после

сбоя тактового генератора дескремблера из-за воздействия помех на линию связи или по иным причинам. В отсутствие синхронизации между скремблером и дескремблером содержимое регистров $RG1$ и $RG2$ не совпадает, поток принимаемых данных RD ошибочен и не совпадает с потоком передаваемых данных SD .

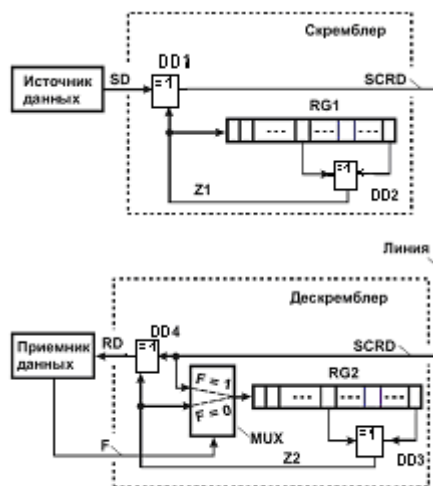


Рис. 3.17. Средства синхронизации аддитивных скремблера и дескремблера.

При обнаружении устойчивого хаотического потока данных RD (в котором нет обусловленного протоколом разделения на кадры и т. п.), приемник формирует сигнал $F = 1$. Вследствие этого на вход регистра $RG2$ транслируется сигнал скремблированных данных $SCRD$, как в схеме на рис. 3.14.

Протокол обмена предусматривает пересылку данных в виде последовательности кадров. Группы обычных кадров перемежаются со служебными кадрами. Например, после группы из 63 обычных кадров следует один служебный. Он, в частности, содержит синхронизирующую последовательность из нулевых битов. При выдаче этих битов ($SD = 0$) в скремблер элемент 1 выполняет функцию повторителя сигнала $Z1$. Поэтому в данном случае скремблированный сигнал $SCRD$ представляет собой фрагмент «истинной» псевдослучайной битовой последовательности, в том смысле, что она не смешана с потоком произвольных данных и порождается только генератором скремблера.

Эта последовательность загружается в регистр $RG2$, так как $F = 1$. После того как содержимое регистров $RG1$ и $RG2$ оказывается одинаковым, сигнал $Z2$ начинает повторять сигнал $Z1$. Синхронизация достигнута. После заполнения регистра $RG2$ на вход приемника данных подается непрерывная последовательность лог. 0, так как $RD = SD$. После уверенного обнаружения последовательности лог. 0 приемник формирует сигнал $F = 0$ и тем самым переключает генератор дескремблера в режим изолированной работы. Теперь синхронизация не только достигнута, но и «сохранена». Для гарантии окончания процесса установления синхронизации источник данных еще некоторое время продолжает выдачу последовательности лог. 0, а затем приступает к передаче данных согласно принятому в системе протоколу.

У читателя могут возникнуть сомнения в том, что *BER*-генератор третьего модема выдает в линию псевдослучайную тестовую последовательность битов со скоростью, задаваемой генератором *G1*. Действительно, данные в смеси с синхросигналом передаются по «верхнему» каналу справа налево, а генератор *G1* находится слева. Каким образом синхросигнал от генератора *G1* может распространяться по линии против течения потока данных? - Ответ на этот вопрос мы уже рассматривали, см. п. 3.5.2.

3.6. Кодирование данных и синхросигнала для параллельной передачи по группе витых пар проводов

Далее приведен ряд схемных решений, обеспечивающих передачу данных по группе витых пар проводов. Сначала рассмотрена основная схема, а затем предложены ее усовершенствованные варианты.

3.6.1. Схема кодирования-декодирования. Сигналы можно передавать по нескольким витым парам проводов без использования кодирования, выделив для каждого сигнала одну витую пару. В схеме, приведенной на рис. 3.18, трехразрядный параллельный код *DIN* и сопровождающий его синхросигнал *CIN* передаются по четырем витым парам проводов к удаленному абоненту. Выходной код *DOUТ* и принятый синхросигнал *COUТ* совпадают с входными с точностью до задержки передачи.



Рис. 3.18. Схема параллельной передачи данных и синхросигнала по витым парам проводов без применения кодирования

При увеличении разрядности кода *DIN* такое решение становится слишком неэкономичным. Для уменьшения числа витых пар проводов можно воспользоваться схемой, приведенной на рис. 3.19 [18].

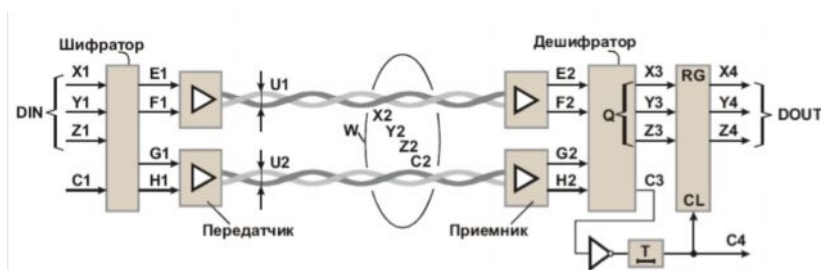


Рис. 3.19. Схема параллельной передачи кодированных данных и синхросигнала по витым парам проводов

В этой схеме применено трехуровневое кодирование сигнала в линии. Напряжение *U1* (*U2*) между проводами витой пары может быть отрицатель-

ным, нулевым или положительным (сокращенно: $-$, 0 , $+$). Это напряжение формируется передатчиком в зависимости от сочетания битов $E1\ F1\ (G1\ H1)$ на его входах следующим образом. При $E1 = F1 = 0\ (G1 = H1 = 0)$ напряжение $U1\ (U2)$ нулевое, при $E1 = 0, F1 = 1\ (G1 = 0, H1 = 1)$ – отрицательное, при $E1 = 1, F1 = 0\ (G1 = 1, H1 = 0)$ – положительное. Код $E1\ F1\ G1\ H1$ формируется шифратором из входного кода $X1\ Y1\ Z1\ C1$ (данные, синхросигнал), как показано в левой части табл. 3.2.

Таблица 3.2

Состояния шифратора, линии связи и дешифратора при параллельной передаче трех битов данных и синхросигнала

Коды на входах и выходах шифратора								Сигналы в линии		Коды на входах и выходах дешифратора							
C1	X1	Y1	Z1	E1	F1	G1	H1	U1	U2	H2	G2	F2	E2	Z3	Y3	X3	C3
1	x	x	x	0	0	0	0	0	0	0	0	0	0	x	x	x	1
0	0	0	0	0	0	0	1	0	–	1	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	+	0	1	0	0	1	0	0	0
0	0	1	0	0	1	0	0	–	0	0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1	–	–	1	0	1	0	1	1	0	0
0	1	0	0	0	1	1	0	–	+	0	1	1	0	0	0	1	0
0	1	0	1	1	0	0	0	+	0	0	0	0	1	1	0	1	0
0	1	1	0	1	0	0	1	+	–	1	0	0	1	0	1	1	0
0	1	1	1	1	0	1	0	+	+	0	1	0	1	1	1	1	0

Примечание. Символ «х» соответствует произвольному значению бита

Приемники и дешифратор осуществляют обратное преобразование сигналов, представленных напряжениями $U1$ и $U2$ в линии. В этом можно убедиться сопоставлением левой и правой частей таблицы – они симметричны. В центральной части таблицы показаны девять возможных комбинаций сигналов в линии. Первая комбинация $(0\ 0)$ выбрана для отображения единичного состояния синхросигнала. Символы «х» в левой части таблицы означают, что при $C1 = 1$ входные сигналы DIN не воспринимаются шифратором. Эти же символы в правой части таблицы показывают, что при выдаче синхросигнала $C3$ выходные данные $X3\ Y3\ Z3$ не определены и не могут использоваться для выдачи абоненту. Остальные восемь комбинаций сигналов в линии распределены для отображения восьми состояний трехразрядного кода $X1\ Y1\ Z1$ при условии, что $C1 = 0$. Шифратор и дешифратор могут быть выполнены на основе ПЗУ, программируемой логики или построены из обычных логических элементов.

Как следует из временных диаграмм, показанных на рис. 3.20, сигналы, передаваемые по линии, поочередно отображают синхросигнал и данные. Преобразования этих зашифрованных тремя уровнями напряжения сигналов ($C2, X2, Y2, Z2$) в группе W являются сигналы на входах шифратора. Как уже

отмечалось, группа выходных сигналов дешифратора не определена при наличии синхросигнала. Этот факт отмечен затемненными областями на временной диаграмме группы Q сигналов $X3$, $Y3$, $Z3$ на выходе шифратора.

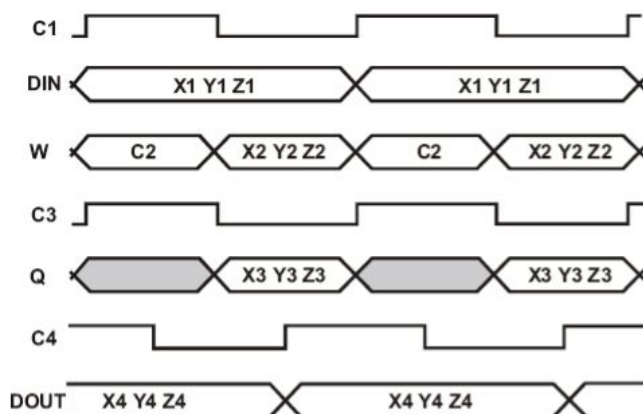


Рис.3.20. Временные диаграммы передачи данных и синхросигнала

Для устранения областей неопределенности этих сигналов применен параллельный регистр RG . Данные принимаются в него по положительному фронту сигнала $C4$, который формируется из сигнала $C3$ после его инвертирования и задержки на четверть периода. В результате сигналы $DOUT$ и $C4$ с некоторой задержкой повторяют сигналы DIN и $C1$.

3.6.2. Усовершенствованная схема. Описанный способ уменьшения числа витых пар проводов линии применим и к схемам с большей разрядностью. В общем случае при трехуровневом кодировании число состояний линии, содержащей N витых пар неводов; равно $3N$. С помощью этих состояний можно передать M -разрядный двоичный код DIN (см. рис. 3.19), где $M = \lfloor \log_2 3N \rfloor$, прямые скобки обозначают целую часть заключенного в них числа. Так, при $N = 2, 3, 4, \dots, 10$ разрядность M передаваемого двоичного кода составляет 3, 4, 6, 7, 9, 11, 12, 14, 15 (табл. 3.3). Напомним, что одно из избыточных состояний линии в приведенном ранее примере использовалось для кодирования единичного состояния синхросигнала.

Из табл. 3.3 следует, что при $N > 2$ в троичном представлении двоичного кода имеется избыточность, в основном, значительная. Ее можно использовать для повышения информационной нагрузки на линию связи при передаче потока данных.

Первый вариант усовершенствованной схемы предусматривает использование избыточных состояний линии для построения дополнительного канала связи. Телекоммуникационная система (рис. 3.21), в которую включена схема передачи данных, содержит четыре оконечных устройства $DTE1 - DTE4$, например четыре компьютера. В процессе работы системы данные передаются из устройства $DTE1$ в устройство $DTE3$ по основному (первому) каналу. Одновременно с этим по дополнительному (второму) каналу данные передаются из устройства $DTE2$ в устройство $DTE4$. (Для передачи данных в об-

ратном направлении необходима вторая схема передачи данных, включенная встречно.)

Таблица 3.3
 Параметры перехода от троичного к двоичному представлению кодов

Число витых пар про- водов N	Число со- стояний линии 3^N	Допустимая разрядность двоичного кода M	Число со- стояний двоичного кода 2^M	Избыточность троичного пред- ставления двоичного кода	
				абсолютная $3^N - 2^M$	относительная $((3^N - 2^M) / 3^N) \cdot 100\%$
2	9	3	8	1	11,1
3	27	4	16	11	40,7
4	81	6	64	17	21,0
5	243	7	128	115	47,3
6	729	9	512	217	29,8
7	2187	11	2048	139	6,3
8	6561	12	4096	2465	37,5
9	19683	14	16384	3299	16,8
10	59049	15	32768	26281	44,5

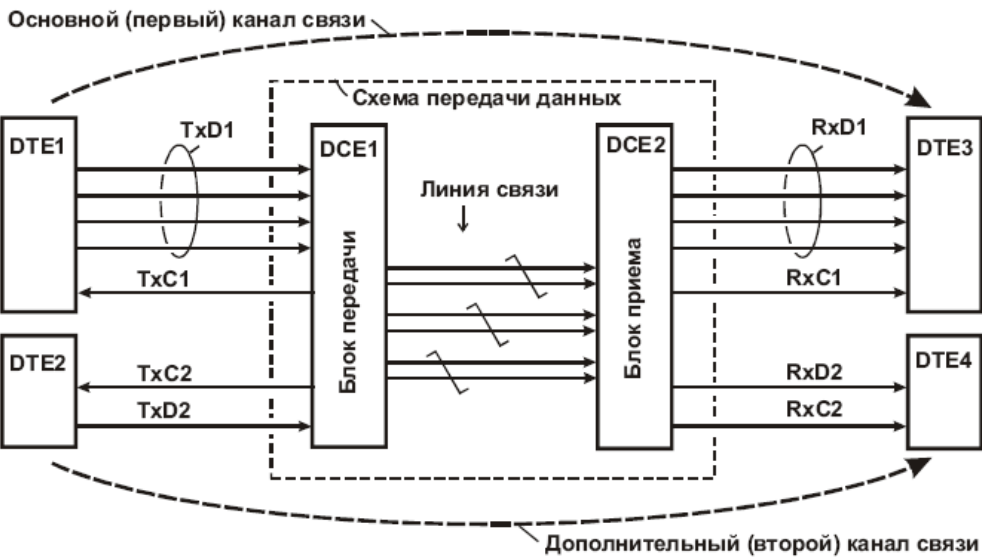


Рис. 3.21. Пример включения усовершенствованной схемы передачи данных (первый, третий варианты) в телекоммуникационную систему

Сигнал *TxC1* обеспечивает синхронное поступление данных *TXDL* первого канала из устройства *DTE1* в блок передачи данных *DCE1*. Сигнал *TxC2* синхронизирует поступление данных *TXD2* второго канала из устройства *DTE2* в блок *DCE1*. Положительные фронты сигнала *TxC1* задают границы битовых интервалов для каждого входного сигнала из группы *TXDL*. Поло-

жительные фронты сигнала $TxC2$ задают границы битовых интервалов сигнала $TXD2$. Отрицательные фронты сигналов $TxC1$ и $TxC2$ гарантируют истинность соответствующих данных $TXD1$ и $TXD2$. Аналогичные функции выполняют синхросигналы $RXC1$ и $RXC2$ сопровождения данных $RXD1$ и $RXD2$. Как будет показано, поток данных по первому каналу связи имеет постоянную скорость, в отличие от потока данных по второму каналу. Скорость передачи битов по второму каналу зависит от кодов в первом канале и при случайном равномерном распределении этих кодов составляет 62,5% скорости передачи четырехразрядных слов данных по первому каналу.

На рис. 3.22 – 3.24 приведены схемы и временные диаграммы, поясняющие детали предлагаемого решения.

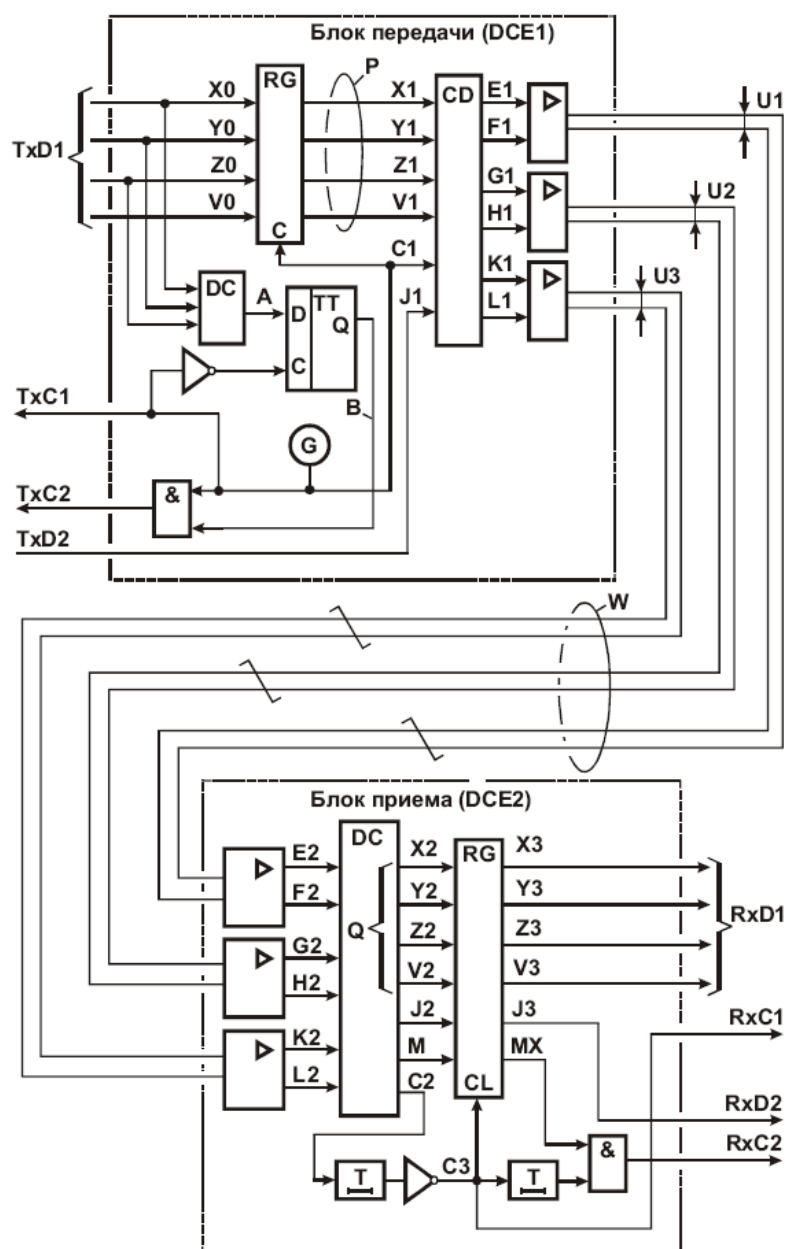


Рис. 3.22. Усовершенствованная схема параллельной передачи кодированных данных и синхросигнала по витым парам проводов (первый вариант)

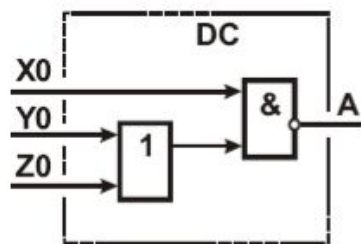


Рис. 3.23. Схема дешифратора блока *DCE1* передачи данных

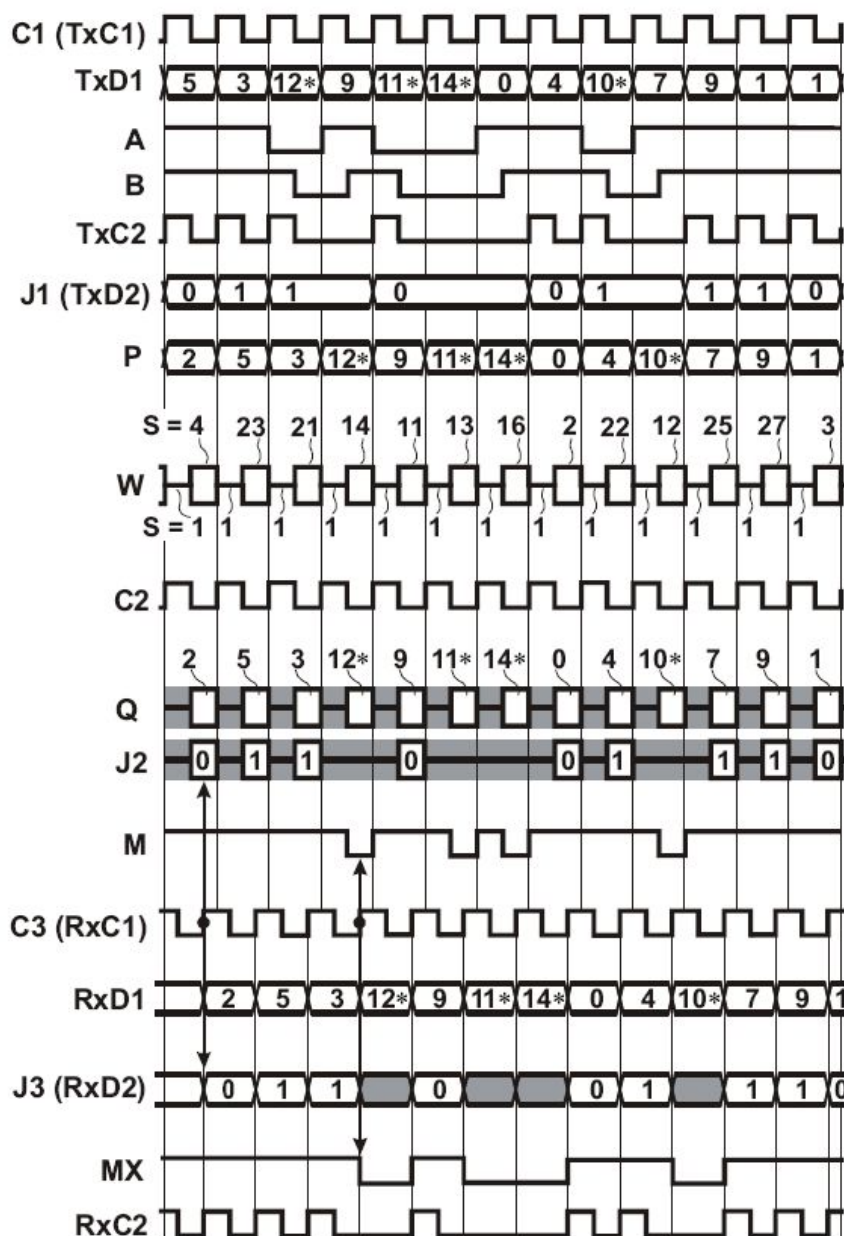


Рис. 3.24. Временные диаграммы передачи данных

Все процессы, протекающие при передаче данных, синхронизируются от генератора G блока $DCE1$ (см. рис. 3.22). По положительному фронту сигнала $TxC1$ устройство $DTE1$ формирует очередной четырехразрядный двоичный код $X0\ Y0\ Z0\ V0$ данных $TXDL$ ($X0$ - старший разряд). В примере, приведенном на временных диаграммах (см. рис. 3.24), от устройства $DTE1$ под управлением сигнала $TxC1$ поступает такая последовательность кодов $TXDL$: $01012 = 5$, $00112 = 3$, $11002 = 12$, ..., $00012 = 1$. При этом коды, помеченные символами «*», препятствуют распространению данных по дополнительному каналу в соответствующих битовых интервалах.

Регистр RG блока $DCE1$ принимает данные по положительному фронту сигнала $TxC1$, поэтому в этом регистре фиксируется «старый» код $TXDL$, прежде чем на его входы поступит «новый» код. (По аналогичной схеме построены обычные сдвиговые регистры с общей цепью синхронизации.) В результате данные P на выходах регистра формируются с задержкой на один такт по сравнению с данными $TXDL$ на его входах (см. диаграмму сигналов P , повторяющую диаграмму сигналов $TXDL$ со сдвигом вправо на один период сигнала $TxC1$).

Данные $TXDL$ анализируются дешифратором DC блока $DCE1$. Если данные принадлежат диапазону 0, 1, 2, ..., 9, то на выходе дешифратора формируется сигнал $A = 1$. Если данные принадлежат диапазону 10, 11, 12, ..., 15, то на выходе дешифратора формируется сигнал $A = 0$ (см. временную диаграмму этого сигнала). Как показано на рис. 3.23, дешифратор может быть выполнен на двух логических элементах. Сигнал A с выхода дешифратора принимается в триггер по положительному фронту инвертированного сигнала $TXCL$. Сигнал $B = 0$ с выхода триггера закрывает элемент И по нижнему входу и препятствует формированию сигнала $TxC2$ в тех ситуациях, когда необходимо приостановить передачу данных по второму каналу. Поэтому данные $TXD2$ запрашиваются у устройства $DTE2$ и поступают от него с переменным темпом по мере возможности их передачи по второму каналу.

Сигналы P , $C1$ и L ($TXD2$) поступают на входы шифратора CD . Цепь, состоящая из шифратора, выходных и входных усилителей, обслуживающих линию связи, и дешифратора блока $DCE2$ выполняет преобразование кодов в соответствии с табл. 3.4.

Как и в предыдущей схеме (см. рис. 3.19), в линии связи применено трехуровневое кодирование сигналов. Напряжение $U1$ ($U2$, $U3$) между проводами витой пары может быть отрицательным, нулевым или положительным (сокращенно: -, 0, +). Это напряжение формируется выходным усилителем блока $DCE1$ в зависимости от сочетания битов $EL\ FL$ ($G1\ HI$, $G1\ L1$) на его входах следующим образом. При $EL = F1 = 0$ ($G1 = HI = 0$, $K1 = L1 = 0$) напряжение $U1$ ($U2$, $U3$) нулевое, при $E1 = 0$, $F1 = 1$ ($G1 = 0$, $HI = 1$; $K1 = 0$, $L1 = 1$) - отрицательное, при $E1 = 1$, $F1 = 0$ ($G1 = 1$, $HI = 0$; $K1 = 1$, $L1 = 0$) - положительное. Код $EL\ FL\ G1\ HI\ K1\ LI$ формируется шифратором CD из входного кода $X1\ Y1\ Z1\ VI\ CI\ L$ (данные P , синхросигнал, данные $TXD2$), как показано в левой части таблицы.

Таблица 3.4

Состояния шифратора, линии связи и дешифратора при передаче данных и синхросигнала схемой, приведенной на рис. 3.22

[illegible]

Входные усилители и дешифратор блока *DCE2* выполняют обратное преобразование сигналов, представленных напряжениями $U1$, $U2$ и $U3$ в витых парах проводов линии связи. В графе S таблицы перечислены $33 = 2^7$ возможных состояний трехуровневых сигналов $U1 - U3$ в линии связи. Первое состояние ($S = 1$) соответствует нулевым напряжениям в линии: $(U1\ U2\ U3) = (0\ 0\ 0)$. Это состояние выбрано для отображения единичного значения синхросигнала $C1$. Символы «х» в левой части первой строки таблицы означают, что при $C1 = 1$ входной код $X1\ Y1\ Z1\ V1\ J1$ не воспринимается шифратором. Эти же символы в правой части первой строки таблицы показывают, что при выдаче восстановленного синхросигнала $C2 = 1$ выходные данные $X2\ Y2\ Z2\ V2\ J2$ также не определены. При этом сигнал M принимает значение лог. 1.

Состояния 2, 3, ..., 27, соответствующие условию $C1 = 0$, используются для отображения данных первого и, в дополнение к этому, по возможности, второго каналов. Как следует из таблицы, коды P , принадлежащие диапазону 0-9, представлены двумя равноценными копиями. Первая копия размещена в строках с номерами 2 - 11 и сопровождается нулевым значением сигнала $J1$. Вторая копия размещена в строках с номерами 18 - 27 и сопровождается единичным значением сигнала $J1$. Передача первой или второй копии кода по первому каналу совмещена с передачей одного бита данных ($TXD2 = J1$) по второму каналу. Коды P , принадлежащие диапазону 10 - 15, представлены в единственном экземпляре и размещены в строках 12 - 17. При передаче этих кодов состояние сигнала $J1$ не учитывается (что отражено символами «х» в столбце « $J1$ » таблицы). Иными словами, в данных ситуациях параллельная передача данных по второму каналу приостанавливается.

Как показано на временной диаграмме сигналов W линии, в первых половинах тактов состояние линии $S = 1$, так как $C1 = 1$. Во второй половине первого такта $C1 = 0$, $P = 2$, $J = 0$. Этому сочетанию сигналов соответствует четвертая строка таблицы. Из нее следует, что состояние линии $S = 4$. Аналогично, во второй половине второго такта $C1 = 0$, $P = 5$, $J = 1$, в результате в линии формируется состояние $S = 23$ (+ - -) и т. д.

Дешифратор блока *DCE2* восстанавливает синхросигнал $C2$ (совпадающий с синхросигналом $C1$ с точностью до задержки передачи) и формирует сигналы Q , $J2$ и M (см. соответствующие временные диаграммы). Сигналы Q и $J2$ содержат области неопределенности, выделенные на диаграммах затемненными участками.

Для устранения неопределенности сигналов Q применен параллельный регистр. Данные принимаются в него по положительному фронту сигнала $C3$, который формируется из сигнала $C2$ после его задержки на четверть периода и последующего инвертирования. В результате сигналы $RXDL$ и $RXCL$ с некоторой задержкой повторяют сигналы $TXDL$ и $TXC1$.

Сигнал $C3$ далее проходит через элемент задержки для исключения «гонок». При наличии единичного сигнала MX элемент И формирует периодический сигнал $RXC2$. При $MX = 0$ из сигнала $RXC2$ исключаются положительные

импульсы, соответствующие областям неопределенности сигнала $J3$. Таким образом регулируется поток данных по дополнительному каналу.

При равномерном случайном распределении кодов $TXDL$ вероятность их попадания в диапазон 0-9 равна $10/16 = 0,625$. Каждое попадание в этот диапазон сопровождается передачей одного бита данных по дополнительному каналу. Поэтому скорость передачи битов по дополнительному каналу составляет 62,5% скорости передачи четырехразрядных кодов $TXDL$ по основному каналу. Протокол передачи данных может предусматривать передачу нулевых кодов $TXDL$ в незанятом состоянии первого канала. Тогда дополнительный канал используется в каждом такте синхросигнала, т. е. в нем полностью исключаются приостановки передачи данных.

Для исключения неблагоприятных ситуаций, при которых в течение длительных промежутков времени коды попадают в диапазон 10-15, можно применить скремблирование одного или двух старших разрядов данных $TXDL$ перед их подачей на входы блока $DCE1$.

4. КАНАЛЫ, РЕЖИМЫ И ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ

4.1. Классификация каналов

На рис. 4.1 представлен вариант классификации каналов передачи данных.

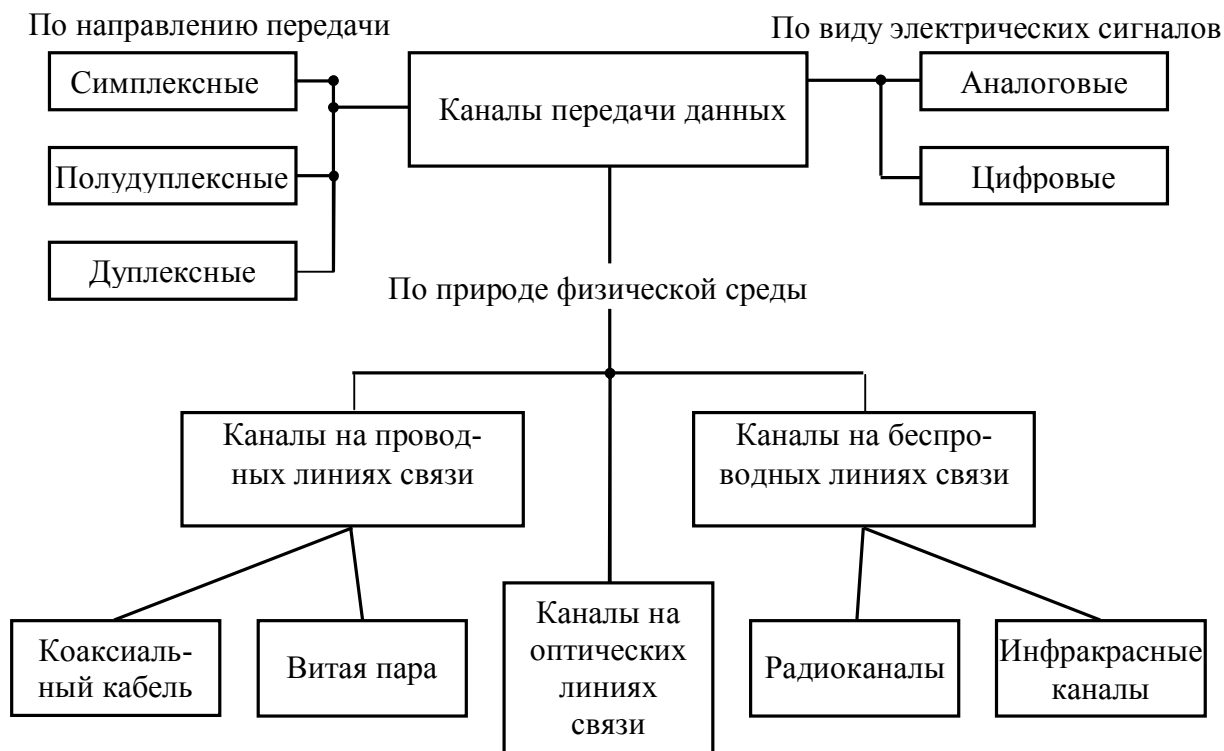


Рис. 4.1. Классификация каналов передачи данных

Цифровой канал является битовым трактом с цифровым сигналом на входе и выходе. На вход аналогового канала поступает непрерывный сигнал, и с его выхода снимается также непрерывный сигнал.

Различают каналы: коммутируемые и выделенные, двух- и четырёхпроводные. Коммутируемые каналы предоставляются потребителям на время соединения по их требованию. Такие каналы принципиально содержат в своем составе коммутационное оборудование телефонных станций (АТС).

Выделенные (арендованные) каналы арендуются у телефонных компаний или (очень редко) прокладываются заинтересованной организацией. Такие каналы являются принципиально двухточечными. Их качество выше, чем у коммутируемых каналов, так как на них не влияет коммутационная аппаратура АТС.

Каналы, как правило, имеют двухпроводное или четырёхпроводное окончание. Их называют, соответственно, двухпроводными и четырёхпроводными.

Четырёхпроводные каналы предоставляют два провода для передачи сигнала и еще два провода для приема. В таких каналах практически полностью отсутствует влияние сигналов, передаваемых во встречном направлении,

Двухпроводные каналы используют два провода как для передачи, так и для приема сигналов. Такие каналы позволяют экономить на стоимости кабелей, но требуют усложнения каналообразующей аппаратуры и аппаратуры пользователя. Двухпроводные каналы требуют разделения принимаемого и передаваемого сигналов, что реализуется при помощи дифференциальных систем, обеспечивающих необходимое затухание по встречным направлениям передачи. Наличие дифференциальных систем приводит к искажениям амплитудно-частотных и фазочастотных характеристик канала и к специфической помехе в виде эхо-сигнала.

Физические каналы (среды) передачи данных бывают двух типов: открытые и закрытые. В закрытом канале распространение сигналов ограничено данным каналом, и, за исключением небольших утечек, сигнал не покидает среду передачи. К числу закрытых сред передачи относятся пара проводов, коаксиальный кабель, волновод, волоконно-оптический кабель.

При использовании пары проводов по схеме, приведенной на рис. 4.2, а, линия получилась, не согласованная с землей [12], почти все устройства, генерирующие шум, являются причиной возникновения в ней помех (наводок). Вот почему для существенного уменьшения уровня шума в канале связи стало применяться согласованная двухпроводная линия (рис. 4.2, б).

Наиболее широко используются закрытые среды передачи, состоящие из скрученных вместе пар проводов – в одном кабеле может быть от 4 до 3000 таких пар. Поскольку провод функционирует, как антенна, то для уменьшения электромагнитных помех применяют экранирование и скручивание (переплетение) проводов друг с другом с определенным шагом.

При увеличении частоты резко возрастает затухание сигнала в витой паре, а так же увеличиваются перекрестные наводки в соседних парах кабеля. Максимально возможная частота передаваемого сигнала по симметричному кабелю без применения специальных технологий равна примерно 1 МГц.

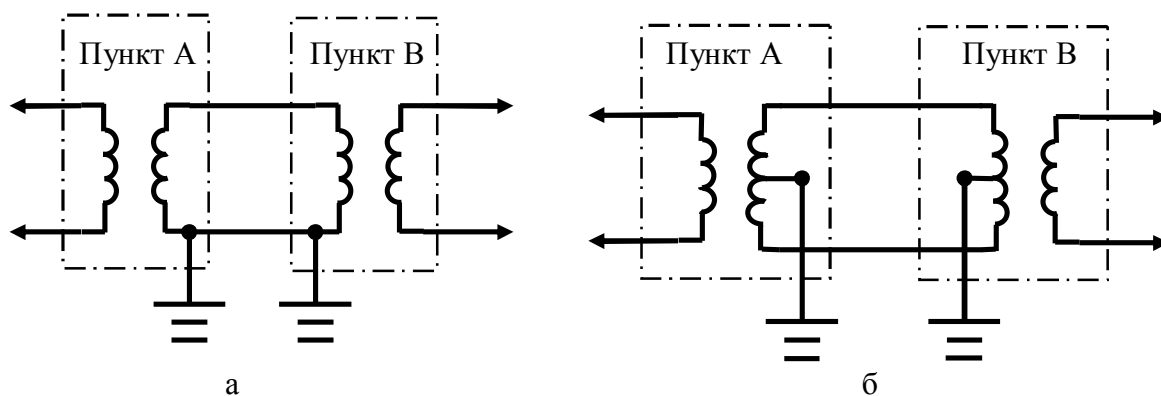


Рис. 4.2. Виды проводных схем передачи:
а – несогласованная; б – согласованная

4.1.1. Витая пара. Витая пара (англ. *twisted pair*) — вид кабеля связи, представляет собой одну или несколько пар изолированных проводников, скрученных между собой (с небольшим числом витков на единицу длины), для уменьшения взаимных наводок при передаче сигнала, и покрытых пластиковой оболочкой. Один из компонентов современных структурированных кабельных систем. Используется в телекоммуникациях и в компьютерных сетях в качестве сетевого носителя во многих технологиях, таких как *EtherNet*, *ARCNet* и *Token RING*. В настоящее время, благодаря своей дешевизне и лёгкости в установке, является самым распространённым для построения локальных сетей.

Кабель подключается к сетевым устройствам при помощи соединителя *RJ45*, немного большим, чем телефонный соединитель *RJ11*.

В зависимости от наличия защиты — электрически заземлённой медной оплетки или алюминиевой фольги вокруг скрученных пар, определяют разновидности данной технологии:

- неэкранированная витая пара (*UTP – UNShleLDeD twisted pair*);
- экранированная витая пара (*STP – ShleLDeD twisted pair*);
- фольгированная витая пара (*FTP – FoILeD twisted pair*);
- фольгированная экранированная витая пара (*SFTP – ShleLDeD FoILeD twisted pair*).

В некоторых типах экранированного кабеля, защита может использоваться ещё и вокруг каждой пары, индивидуальное экранирование. Экранирование обеспечивает лучшую защиту от электромагнитных наводок как внешних, так и внутренних, и т. д.

Из курса физики известно, что любой кабель обладает, кроме активного, также емкостным и индуктивным сопротивлениями, причем два последних зависят от частоты сигнала. Все три типа сопротивления определяют так называемый импеданс цепи. Наличие импеданса приводит к тому, что при распространении сигнала по кабелю он постепенно затухает, теряя часть своей первоначальной мощности.

Наряду с затуханием при распространении сигнала возникают различного рода помехи. Первый род помех – это индуцированные наводки паразитных сигналов, которые являются следствием хорошо известного явления электромагнитной индукции: изменение тока в замкнутом проводящем контуре порождает меняющийся во времени магнитный поток. Изменение магнитного потока, в свою очередь, индуцирует возникновение электрического тока в соседнем проводящем контуре. С учетом того, что в кабеле близко расположены пары проводов, ясно, что изменение тока в одном из них приводит к появлению индуцированного тока в остальных парах проводов. Явление взаимной индукции носит распределенный характер, однако его более удобно характеризовать в двух локализованных точках – в начале и в конце кабеля.

Если взаимная индукция вычисляется в начале кабеля, то соответствующий тип помех будет называться *NEXT* (*NeaR-eND cRoSStaLk LoSS*). Если же помехи, вызванные взаимной индукцией, рассматриваются в конце кабеля, то они называются *FEXT* (*FaR-eND cRoSStaLk LoSS* – рис. 4.3).

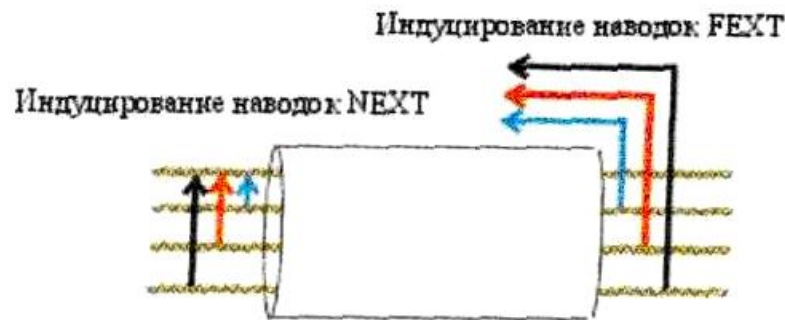


Рис. 4.3. Возникновение взаимных наводок в кабеле

Величина переходного затухания на ближнем конце $A_{\text{бк}}$ (*NEXT*) может быть выражена следующим образом:

$$A_{\text{бк}} = 20 \log_{10}(U_{10}/U_{20}),$$

где U_{10} – уровень напряжения сигнала на передающей стороне; U_{20} – уровень напряжения на принимающей паре во время передачи сигнала по передающей паре.

Линейное затухание витой пары (A_L) рассчитывается по формуле:

$$A_L = 20 \log_{10}(U_{10}/U_{11}),$$

где U_{11} – уровень напряжения сигнала на приемной стороне.

Кроме того, при распространении сигнала возникает и другой тип помех, связанный с рассогласованием входного импеданса сетевого адаптера и кабеля. В результате подобного рассогласования возникает отражение сигнала, что также приводит к образованию шума.

Передача сигналов в описанных выше условиях помех требует использования хитроумных способов, позволяющих обеспечить необходимую скорость передачи и в то же время гарантировать безошибочное распознавание передаваемых сигналов.

Существует несколько категорий кабеля витая пара, которые нумеруются от *CAT1* до *CAT7* и определяют эффективный пропускаемый частотный диапазон. Кабель более высокой категории обычно содержит больше пар проводов и каждая пара имеет больше витков на единицу длины. Категории неэкранированной витой пары описываются в стандарте *EIA/TIA 568* (Американский стандарт проводки в коммерческих зданиях).

CAT1 – (полоса частот 0,1 МГц) телефонный кабель, всего одна пара, известный в России как «лапша». В США использовался ранее, и проводники были скручены между собой. Используется только для передачи голоса или данных при помощи модема;

CAT2 – (полоса частот 1 МГц) старый тип кабеля, 2-е пары проводников, поддерживал передачу данных на скоростях до 4 Мбит/с, использовался в сетях *token RING* и *ARCNet*. Сейчас иногда встречается в телефонных сетях;

CAT3 – (полоса частот 16 МГц) 2-х парный кабель, использовался при построении локальных сетей *10BASE-T* и *token RING*, поддерживает скорость передачи данных только до 10 Мбит/с. В отличие от предыдущих двух, отвечает требованиям стандарта *IEEE 802.3*. Также до сих пор встречается в телефонных сетях;

CAT4 – (полоса частот 20 МГц) кабель состоит из 4-х скрученных пар, использовался в сетях *token RING*, *10BASE-T*, *10BASE-T4*, скорость передачи данных не превышает 16 Мбит/с, сейчас не используется;

CAT5 – (полоса частот 100 МГц) 4-х парный кабель, это и есть, то, что обычно называют кабель «витая пара», благодаря высокой скорости передачи, до 100 Мбит/с при использовании 2-х пар и до 1000 Мбит/с, при использовании 4-х пар, является самым распространённым сетевым носителем, использующимся в компьютерных сетях до сих пор. При прокладке новых сетей используются несколько усовершенствованным кабелем *CAT5e* (полоса частот 125 МГц), который лучше пропускает высокочастотные сигналы;

CAT6 – (полоса частот 250 МГц) применяется в сетях *FaSt EtherNet* и *GIGabit EtherNet*, состоит из 4-х пар проводников и способен передавать данные на скорости до 10000 Мбит/с. Добавлен в стандарт в июне 2002 года. Существует категория *CAT6e*, в которой увеличена частота пропускаемого сигнала до 500 МГц. По данным *IEEE* 70% установленных сетей в 2004 году, использовали кабель категории *CAT6*, однако возможно это просто дань моде, поскольку кабель *CAT5* и *CAT5e* вполне справляется в сетях *10GBASE-T*;

CAT7 – Спецификация на данный тип кабеля пока не утверждена, скорость передачи данных до 10000 Мбит/с, частота пропускаемого сигнала до 600-700 МГц. Кабель этой категории экранирован.

Как уже упоминалось, витая пара широко применяется в сетевых технологиях и коммуникациях, сейчас кабелем категории 6, во многих местах заменяют

коаксиальный кабель. Несмотря на большую защищённость экранированной витой пары, она не получила широкого распространения из-за сложности в установке — требуется заземление, а кабель по сравнению с неэкранированной витой парой более жёсткий.

4.1.2. Коаксиальный кабель. Коаксиальный кабель (от лат. со – совместно и *axis* – ось, то есть «соосный») — вид электрического кабеля. Состоит из двух цилиндрических проводников, соосно вставленных один в другой (рис. 4.4). Чаще всего используется центральный медный проводник, покрытый пластиковым изолирующим материалом, поверх которого идёт второй проводник - медная сетка или алюминиевая фольга. Благодаря совпадению центров обоих проводов потери на излучение практически отсутствуют; одновременно обеспечивается хорошая защита от внешних электромагнитных помех. Поэтому такой кабель обеспечивает передачу данных на большие расстояния и использовался при построении компьютерных сетей (пока не был вытеснен витой парой). Используется в сетях кабельного телевидения и во многих других областях. Основной характеристикой кабеля является волновое сопротивление. В зависимости от этой величины и толщины коаксиальный кабель делится на несколько категорий. Компьютерные сети на основе этого кабеля обычно требуют наличия терминаторов на оконечных точках.

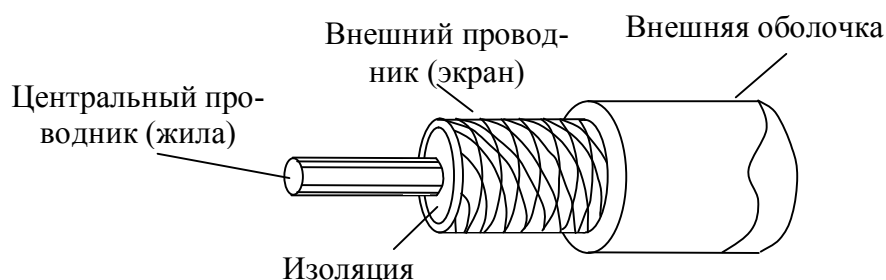


Рис. 4.3. Коаксиальный кабель

Наиболее распространенные категории кабеля:

RG-8 и *RG-11* – «Толстый *EtherNet*» (*ThickNet*), 50 Ом. Стандарт *10BASE5*;

RG-58 – «Тонкий *EtherNet*» (*ThinNet*), 50 Ом. Стандарт *10BASE2*;

RG-58/U – сплошной центральный проводник;

RG-58A/U – многожильный центральный проводник;

RG-58C/U – военный кабель;

RG-59 – телевизионный кабель (*BRoADBaND/CabLe TeLeVISIoN*), 50 Ом.
Российский аналог РК-50;

RG-59/U – телевизионный кабель (*BRoADBaND/CabLe TeLeVISIoN*), 75 Ом.
Российский аналог РК-75

RG-62 – *ARCNet*, 93 Ом.

Тонкий *EtherNet*. Был наиболее распространённым кабелем для построения локальных сетей. Диаметр – примерно 6 миллиметров – и значительная гибкость позволяли ему быть проложенным практически в любых местах. Ка-

бели соединялись друг с другом и с сетевой платой в компьютере при помощи *T*-коннектора *BNC* (*BRITISH NaVaL CoNNectoR*). Между собой кабели могли соединяться с помощью *I*-коннектора *BNC* (прямое соединение). На обоих концах сегмента должны быть установлены терминаторы. Поддерживает передачу данных до 10 Мб/с на расстояние до 185 метров.

Толстый *EtheRNet*. Более толстый, по сравнению с предыдущим кабель — около 12 миллиметров в диаметре, имел более толстый центральный проводник. Плохо гнулся и имел значительную стоимость. Кроме того в присоединении к компьютеру были некоторые сложности — использовались трансиверы *AUI* (*AttachmeNt UNIT INteRFace*), присоединённые к сетевой карте с помощью ответвления, пронизывающего кабель, т.н. «вампирики». За счёт более толстого проводника передачу данных можно было осуществлять на расстояние до 500 метров со скоростью 10 Мб/с. Однако сложность и дороговизна установки не дали этому кабелю такого широкого распространения, как *RG-58*. Исторически, фирменный кабель *RG-8* имел жёлтую окраску, и поэтому иногда можно встретить название «Жёлтый *EtheRNet*» (англ. *YeLLoW EtheRNet*).

4.1.3. Волновод. Если частота передачи высока, то электрическая и магнитная составляющие сигнала могут распространяться в свободном пространстве (не требуется сплошной проводник). Для того чтобы сигнал распространялся в нужном направлении с наименьшими помехами и потерями, иногда используют волновод. Обычно волноводы применяют на частотах от 2 до 110 ГГц для соединения сверх высокочастотных (СВЧ) передатчиков и приемников с антеннами. В волновод под повышенным давлением закачивается сухой воздух или чистый азот с целью снижения влажности, поскольку в СВЧ — диапазоне она существенно увеличивает затухание. Поперечное сечение может быть круглым или прямоугольным.

4.1.4. Оптоволокно. Современные коммуникационные системы предназначены для передачи высокочастотных сигналов, поэтому важным параметром среды является ее полоса пропускания. Применение для передачи бинарных сигналов стеклянных волокон показало, что такие волокна могут быть использованы в широкополосных системах. Принцип работы и основные характеристики приведены в [11].

Оптическое волокно — это диэлектрический (то есть не проводящий электрический ток) волновод, выполненный из стекла или пластика. Оптическое волокно состоит из трех компонентов: сердцевины, оболочки и защитного покрытия (рис. 4.5).

Показатель преломления у сердцевины выше, чем у оболочки. Поэтому, даже если оптическое волокно изогнуто или завязано в узел, введенный в него световой поток проходит по сердцевине волокна. Лучи, входящие в волокно под разными углами, называются **модами**, а волокно, поддерживающее несколько мод, — **многомодовым**.

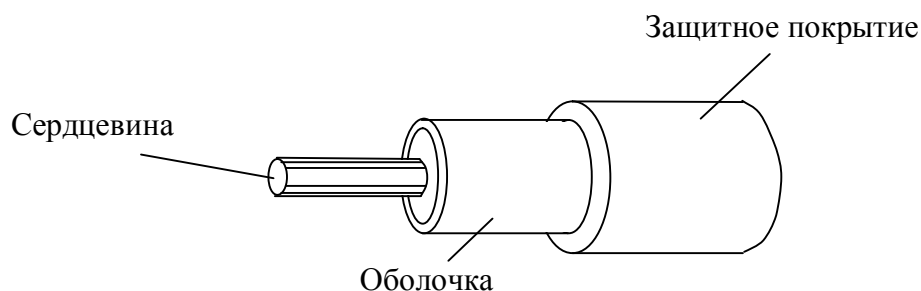


Рис. 4.5. Структура оптического волокна

Простой принцип действия позволяет использовать различные методы, дающие возможность создавать самые разнообразные оптоволоконна:

- Мультиимодовые;
- Одноимодовые оптоволоконна;
- Оптоволоконна с градиентным показателем преломления;
- Оптоволоконна со ступенчатым профилем распределения показателей преломления.

Из-за физических свойств оптоволоконна необходимы специальные методы для их склеивания и соединения с оборудованием. Оптоволоконна являются базой для различных типов кабелей, в зависимости от того, где они будут использоваться.

Многоимодовое распространение лучей вызывает эффект имодовой дисперсии. Он заключается в том, что лучи, одновременно вошедшие в оптическое волокно под разными углами, выходят из него в разное время, что приводит к расширению импульсов света. В результате чего соседние импульсы накладываются друг на друга (рис. 4.6).

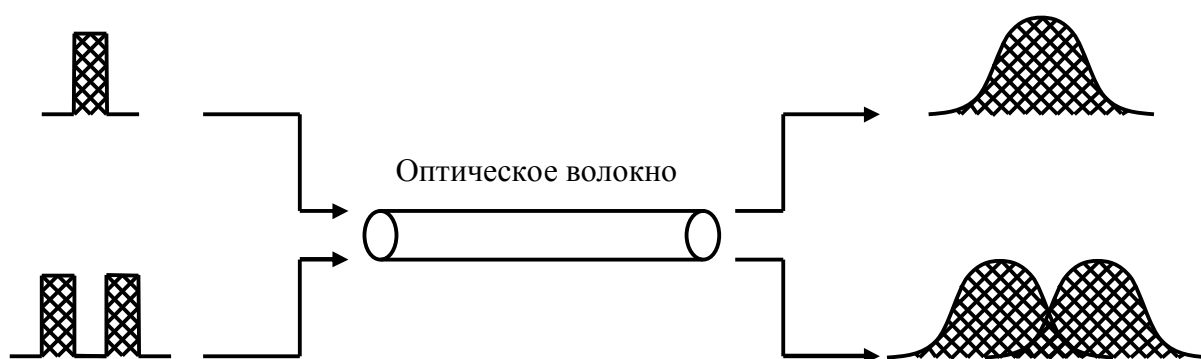


Рис. 4.6. Расширение импульсов

Поэтому, начиная с определенного расстояния, при передаче данных по многоимодовому волокну возникает слишком большое количество ошибок либо передача данных вообще становится невозможной.

Если диаметр сердцевины волокна незначительно (скажем, в 3 раза) превышает длину волны передаваемого света, то по волокну может передаваться только один луч, и модовая дисперсия будет отсутствовать. Такие волокна, называемые одномодовыми, используются в большинстве систем передачи сигналов.

Ширину полосы пропускания волоконно-оптических систем ограничивают модовая и материальная дисперсии. Описанная выше модовая дисперсия проявляется в многомодовом волокне. Материальная дисперсия обусловлена тем, что лучи света разной длины распространяются в одной и той же среде с разной скоростью, из-за чего происходит расширение импульсов.

Потери в оптическом волокне обусловлены в основном четырьмя факторами: излучением передаваемого света из сердцевины в оболочку (рассеянием), преобразованием световой энергии в тепловую (поглощением), неточным совмещением при соединении волокон и потерями на изгибах волокна.

Области минимальных потерь вблизи длин волн 1,3 и 1,5 мкм называются окнами прозрачности. Каждое окно представляет собой диапазон длин волн, в пределах которого затухание минимально. Поэтому производители разрабатывают лазеры и фотодетекторы, функционирующие на длинах волн, соответствующих этим окнам.

Развитие современных оптоволокон началось в 1950-х годах. Они стали использоваться в связи несколько позже, в 1970-х; с этого момента технический прогресс значительно увеличил диапазон действия и скорость оптоволокон, а также уменьшил стоимость систем оптоволоконной связи.

Оптоволокно может быть использовано как средство для дальней связи и построения компьютерной сети, вследствие своей гибкости и возможности завязываться в узел как кабель. Несмотря на то, что волокна могут быть сделаны из прозрачного пластичного оптоволокон или силикагелевого волокна, волокна, использующиеся для передачи информации на большие расстояния, всегда сделаны из стекла, из-за низкого оптического ослабления электромагнитного излучения. В связи используются многомодовые и одномодовые оптоволокон; мультимодовое оптоволокно обычно используется на небольших расстояниях (до 500 м), а одномодовое оптоволокно — на длинных дистанциях. Из-за строгого допуска между одномодовым оптоволокон, передатчиком, приемником, усилителем и другими одномодовыми компонентами, их использование обычно дороже, чем применение мультимодовых компонент.

Оптоволокно может быть использовано как датчик для измерения напряжения, температуры, давления и других параметров. Малый размер и фактическое отсутствие необходимости в электрической энергии, дает оптоволоконным датчикам преимущество перед традиционными электрическими в определенных областях.

Оптоволокно используется в гидрофонах в сейсмических или гидролокационных приборах. Созданы системы с гидрофонами, в которых на волоконный кабель приходится более 100 датчиков. Системы с гидрофоновым датчиком используются в нефтедобывающей промышленности, а также флотом некоторых

стран. Немецкая компания *SeNNheISeR* разработала лазерный микроскоп, работающий с лазером и оптоволоконном.

Оптоволоконные датчики, измеряющие температуры и давления, разработаны для измерений в нефтяных скважинах. Оптоволоконные датчики хорошо подходят для такой среды, работая при температурах, слишком высоких для полупроводниковых датчиков. Другое применение оптоволоконна – в качестве датчика в лазерном гироскопе, в некоторых моделях машин для навигации.

Оптоволоконно применяется в охранной сигнализации на особо важных объектах (например, ядерное оружие). Когда злоумышленник пытается переместить боеголовку, условия прохождения света через световод изменяются, и срабатывает сигнализация.

Оптоволоконна широко используются для освещения. Они используются как световоды в медицинских и других целях, где яркий свет необходимо доставить в труднодоступную зону. В некоторых зданиях оптоволоконна используются для обозначения маршрута с крыши в какую-нибудь часть здания. Оптоволоконное освещение также используется в декоративных целях, включая коммерческую рекламу, искусство и искусственные ёлки.

Оптоволоконно также используется для формирования изображения. Когерентный пучок, создаваемый оптоволоконном, иногда используется совместно с линзами – например, в эндоскопе, который используется для просмотра объектов через маленькое отверстие.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует много типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью связи. Диапазоны коротких, средних и длинных волн (КВ, СВ и ДВ), называемые также диапазонами амплитудной модуляции (*AM – AmpLITUDe MoDULatIoN*) по типу используемого в них метода модуляции сигнала, обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн (УКВ), для которых характерна частотная модуляция (*FM – FReQUeNcY MoDULatIoN*), а также на диапазонах сверхвысоких частот (СВЧ или *mIcRoWaVeS*). В диапазоне СВЧ (свыше 4 ГГц) сигналы уже не отражаются ионосферой Земли, и для устойчивой связи необходимо наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

Способы организации связи с помощью радиоканалов рассмотрено в [11]. Здесь остановимся лишь на системах сотовой радиосвязи.

На рис. 4.7. представлена упрощенная схема организации сотовой связи. Площадь крупного города и его пригородов разбивается на несколько ячеек, называемых сотами, каждая из которых обслуживается базовой станцией (БС) с маломощным передатчиком и соответствующим ему приемником.

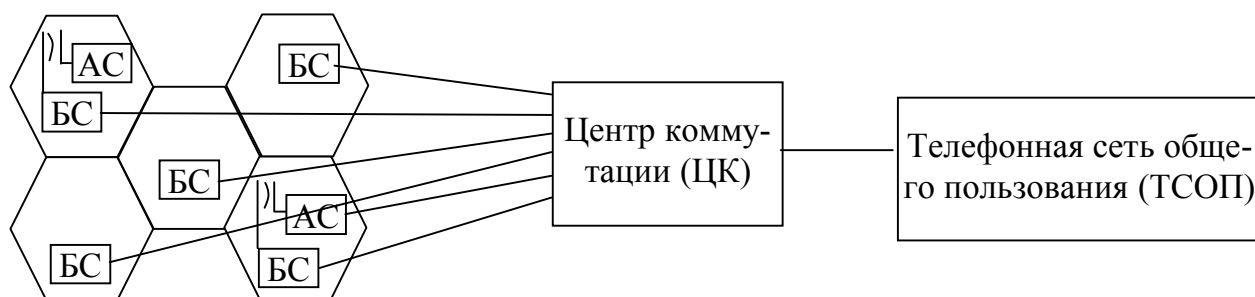


Рис. 4.7. Схема организации сотовой связи

Радиоканалы поддерживают передачи речи и данных. Количество каналов в каждой соте вполне достаточно для обслуживания прогнозированного числа абонентских станций (АС), находящихся в зоне работы данной соты в любой момент времени. Когда абонент делает звонок, его мобильное устройство автоматически занимает свободный канал в данной соте, а после того как он покидает соту, базовая станция соты автоматически переключает его на канал той соты, в которую он переместился, причем для абонента переключение с одного канала на другой остается совершенно незаметным. Соединение происходит через БС и центр коммутации. ЦК соединяет вызывающего абонента через радиоканал с другим мобильным абонентом или через ТСОП с абонентом, имеющим стационарный телефонный аппарат. В последующих разделах рассмотрим более подробно технологии сотовой связи.

4.1.5. Инфракрасное излучение. Инфракрасные беспроводные сети используют для передачи данных инфракрасные лучи. В подобных системах необходимо генерировать очень сильный сигнал, так как в противном случае значительное влияние будут оказывать другие источники.

Сети на рассеянном инфракрасном излучении. При этой технологии сигналы, отражаясь от стен и потолка, в конце концов достигают приемника. Эффективная область ограничивается примерно 30 м. Скорость передачи невелика (так как все сигналы отраженные).

Сети на отраженном инфракрасном излучении. В таких сетях оптические трансиверы, расположенные рядом с компьютером, передают сигналы в определенное место, из которого они транслируются соответствующему компьютеру.

Широкополосные оптические сети. Эти инфракрасные беспроводные сети предоставляют широкополосные услуги магистральной связи, соответствуют жестким требованиям мультимедийной среды и практически не уступают кабельным сетям. Хотя скорость и удобство использования инфракрасных сетей очень привлекательны, возникают трудности при передаче сигналов на расстояние более 10 м. К тому же такие сети подвержены помехам со стороны сильных источников света, которые есть в большинстве помещений.

4.2. Виды связи и режимы передачи данных

При передаче данных между двумя взаимодействующими объектами возможны три вида связи:

симплексный – используется, когда передача данных должна осуществляться только в одном направлении, например в системах контроля, в которых информация с датчиков передается в управляющий компьютер через регулярные промежутки времени;

полудуплексный – применяется, когда два взаимодействующих объекта хотят обмениваться информацией поочередно, т.е. канал используется поочередно для передачи данных в обоих направлениях. Ясно, в таком режиме каждый объект должен иметь возможность переключаться от состояния передачи к состоянию приема;

дуплексный – используется для обмена данными между двумя взаимодействующими объектами (устройствами) в обоих направлениях одновременно, например, когда пропускная способность канала позволяет потоку данных осуществляться в обоих направлениях независимо.

При обмене данными на физическом уровне единицей информации является бит, поэтому средства физического уровня всегда поддерживают побитовую синхронизацию между приемником и передатчиком. Чтобы приемник мог правильно декодировать и интерпретировать получаемый набор битов, он должен знать:

- скорость передачи битов, определяемую интервалом времени, выделяемым на один битовый разряд;
- начало и конец каждого элемента (символа или байта);
- начало и конец каждого полного блока сообщения или кадра.

Эти три фактора называют соответственно побитной или тактовой синхронизацией, побайтной или посимвольной синхронизацией и поблочной или покадровой синхронизацией.

Канальный уровень оперирует кадрами данных и обеспечивает синхронизацию между приемником и передатчиком на уровне кадров. В обязанности приемника входит распознавание начала первого байта кадра, границ полей кадра и признака окончания кадра. Обычно достаточно обеспечить синхронизацию на указанных двух уровнях – битовом и кадровом, чтобы передатчик и приёмник обеспечили устойчивый обмен информацией. Однако при плохом качестве линии связи (как правило это относится к телефонным коммутируемым каналам) для удешевления аппаратуры и повышения надежности передачи данных вводят дополнительные средства синхронизации на уровне байт. Такой режим работы называется **асинхронным** или **старт-стопным**. Использование такого режима работы обусловлено наличием устройств, которые генерируют байты данных в случайные моменты времени. Так работает клавиатура дисплея или другого терминального устройства, с которого человек вводит данные для обработки их компьютером.

В асинхронном режиме каждый байт данных сопровождается специальными сигналами «старт»-стартовый бит и «стоп»-стоповый(ые) бит(ы) (рис. 4.8). Назначение этих сигналов состоит в том, чтобы, во-первых, известить приемник о приходе данных и, во-вторых, чтобы дать приемнику достаточно времени для выполнения некоторых функций, связанных с синхронизацией, до поступления следующего байта. Сигнал «старт» имеет продолжительность тактовый интервал, а сигнал «стоп» может длиться один, полтора или два такта, поэтому говорят, что используется один, полтора или два бита в качестве стопового сигнала, хотя эти сигналы не несут информации. Асинхронным данный режим называют потому, что каждый байт может быть несколько смещен во времени относительно побитовых тактов предыдущего байта. Такая асинхронность передачи байт не влияет на корректность принимаемых данных, как в начале каждого байта происходит дополнительная синхронизация приемника с источником за счет стартового бита. Более «свободные» временные допуски определяют низкую стоимость оборудования асинхронной системы. При **синхронном** режиме передачи старт-стопные биты между каждой парой байт отсутствуют и весь блок или кадр данных передается как одна цепочка битов без каких-либо задержек между 8-битными элементами.

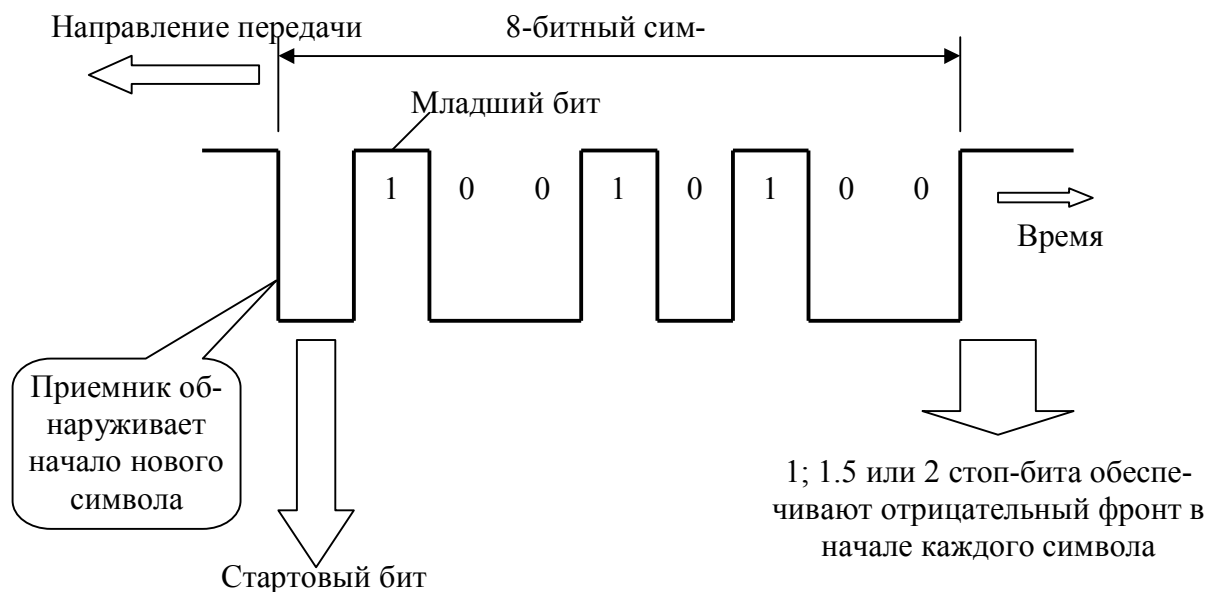


Рис. 4.8. Асинхронная передача

Чтобы приёмник обеспечивал различные уровни синхронизации, необходимо выполнение следующих требований:

- передаваемая цепочка битов должна быть закодирована так, чтобы приемник мог осуществлять побитовую синхронизацию;
- каждому кадру должен предшествовать один или более зарезервированных байтов или символов, благодаря чему приемник может надежно разделить

полученную цепочку битов по границам байтов или символов (побайтная или посимвольная синхронизация);

– содержимое каждого кадра обрамляется парой зарезервированных байтов или символов.

Благодаря последнему требованию приемник оповещается о поступлении кадра данных и об окончании кадра (рис. 4.9). При наличии промежутков времени между передачей двух последовательных кадров в этот период либо непрерывно передаются синхробайты бездействия (простоя), что позволяет приемнику поддерживать побитную или побайтную синхронизацию, либо каждому кадру предшествует один или несколько специальных синхронизирующих байтов или символов, например 01111110, что позволяет приемнику вновь войти в байтовый синхронизм с передатчиком.

Для обеспечения побитовой синхронизации используют самосинхронизирующиеся коды.

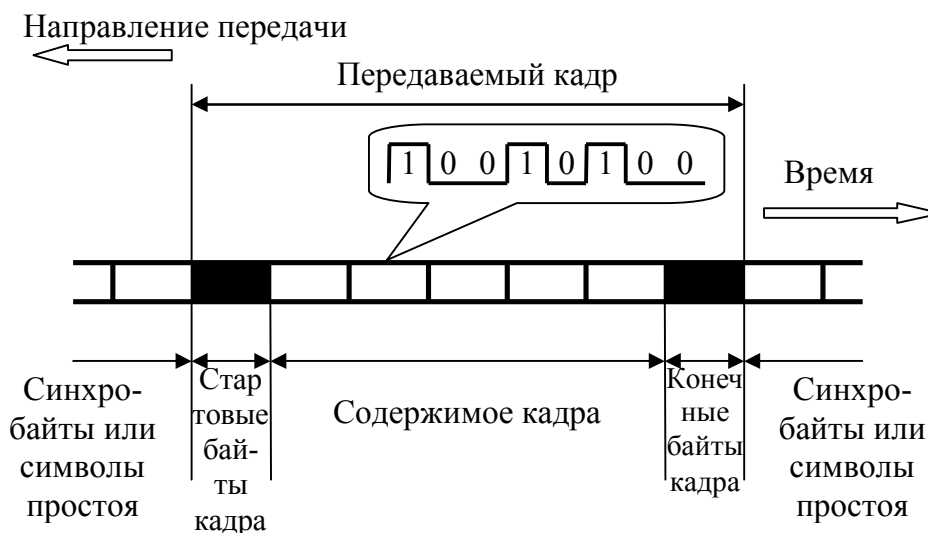


Рис. 4.9. Синхронная передача

4.2.1. Методы передачи данных канального уровня. Канальный уровень обеспечивает передачу пакетов данных, поступающих от протоколов верхних уровней, узлу назначения, адрес которого также указывает протокол верхнего уровня. Протоколы канального уровня оформляют переданные им пакеты в кадры собственного формата, помещая указанный адрес назначения в одно из полей такого кадра, а также сопровождая кадр контрольной суммой. Протокол канального уровня предназначен для доставки кадров данных, как правило, в пределах сетей с простой топологией связей и однотипной или близкой технологией. Другой областью действия протоколов канального уровня являются связи типа «точка-точка» глобальных сетей, когда протокол канального уровня ответственен за доставку кадра непосредственному соседу. Адрес в этом случае

не имеет принципиального значения, а на первый план выходит способность протокола восстанавливать искаженные и утерянные кадры, так как плохое качество территориальных каналов, особенно коммутируемых телефонных, часто требует выполнения подобных действий. Основными характеристиками метода передачи, работающего на канальном уровне, являются следующие:

- асинхронный/синхронный;
- байт-ориентированный/бит-ориентированный;
- с предварительным установлением соединения/дейтграммный;
- с обнаружением искаженных данных/без обнаружения;
- с обнаружением потерянных данных/без обнаружения;
- с восстановлением искаженных и потерянных данных/без восстановления;
- с поддержкой динамической компрессии данных/без поддержки.

Многие из них характерны не только для протоколов канального уровня, но и для протоколов более высоких уровней.

4.2.2. Асинхронные протоколы. Асинхронные протоколы представляют собой один из первых способов связи. Эти протоколы оперируют не с кадрами, а с отдельными символами, которые представлены байтами со старт-стоповым обрамлением.

В асинхронных протоколах применяются стандартные наборы символов, чаще всего *ASCII* или *EBCDIC*. Первые 32 или 27 кодов в этих наборах являются специальными. Они не отображаются на дисплее или принтере и используются асинхронными протоколами для управления режимом обмена данными. В самих пользовательских данных, которые представляют собой буквы, цифры, а также такие знаки, как @, %, \$ и т. п., специальные символы никогда не встречаются, так что проблемы их отделения от пользовательских данных не существует.

Постепенно асинхронные протоколы усложнялись и стали наряду с отдельными символами использовать целые блоки данных, т. е. кадры. Примерами асинхронных протоколов являются популярный протокол *X MODEM*, который передает файлы между двумя компьютерами по асинхронному модему и протокол коррекции ошибок в модемной связи *MNP2*. В этих протоколах часть управляющих операций выполняется посылкой в асинхронном режиме отдельных символов, а часть данных – блоками, что более характерно для синхронных протоколов.

4.2.3. Байт-ориентированные и бит-ориентированные протоколы. В синхронных протоколах между пересылаемыми символами (байтами) нет стартовых и стоповых сигналов, поэтому отдельные символы в этих протоколах пересылать нельзя. Все обмены данными осуществляются кадрами, которые имеют в общем случае заголовок, поле данных и концевик. Все биты кадра передаются непрерывным синхронным потоком, что значительно ускоряет передачу данных. Так как байты в этих протоколах не отделяются друг от друга служебными сигналами, то прежде всего приемник должен распознать границы байт. Затем приемник должен найти начало и конец кадра, а также определить границы каждого

поля кадра – адреса назначения, адреса источника, служебных полей заголовка, поля данных и контрольной суммы, если она имеется.

Большинство протоколов допускает использование в кадре поля данных переменной длины. Иногда и заголовок может быть переменной длины. Обычно протоколы определяют максимальное значение длины поля данных – максимальную единицу передачи данных (*MTU – MaXImUm tRaNSFeR UNIT*). В некоторых протоколах задается также минимальное значение длины поля данных. Например, протокол *EtheRNet* требует, чтобы поле данных содержало не менее 46 байт данных (если приложение хочет отправить меньшее количество байт, то оно обязано дополнить их до 46 байт любыми значениями). Другие протоколы разрешают использовать поле данных нулевой длины, например *FDDI*. Существуют протоколы с кадрами фиксированной длины, например, в сетях АТМ кадры имеют фиксированный размер 53 байт, включая служебную информацию. Для таких протоколов необходимо решить только первую часть задачи – распознать начало кадра.

Синхронные протоколы канального уровня бывают двух типов: байт-ориентированные (иногда их называют символьно-ориентированные или знак-ориентированные) и бит-ориентированные. Для них характерны одни и те же методы синхронизации бит. Главное различие между ними заключается в методе синхронизации символов и кадров.

4.2.4. Байт-ориентированные протоколы. Эти протоколы используют в основном для передачи блоков отображаемых символов, например текстовых файлов. При синхронной передаче стоповые и стартовые биты отсутствуют. Синхронизация здесь осуществляется за счет того, что передатчик добавляет два или более управляющих символа – символы синхронизации (*SYN*) – перед каждым блоком символов. В коде *ASCII* символ *SYN* имеет двоичное значение 00010110 (16h), это несимметричное относительно начала символа значение позволяет легко разграничивать отдельные символы *SYN* при их последовательном приеме. Символы *SYN* выполняют две функции: во-первых, они обеспечивают приемнику побитную синхронизацию, во-вторых, как только битовая синхронизация достигается, они позволяют приемнику начать распознавание границ символов *SYN*. После того как приемник начал отделять один символ от другого, можно задавать границы начала кадра с помощью другого специального символа. Обычно в символьных протоколах для этих целей используется символ *STX* (*StaRt oF TeXt*)-02h. Другой символ – *ETX* (*END oF TeXt*)-03h – отмечает окончание кадра.

Однако такой простой способ выделения начала и конца кадра хорошо работает только в том случае, если внутри кадра нет символов *STX* и *ETX*. При подключении к компьютеру алфавитно-цифровых терминалов такая задача действительно не возникает. Тем не менее, синхронные байт-ориентированные протоколы позднее стали использовать и для связи компьютера с компьютером, а в этом случае данные внутри кадра могут быть любые, когда, например, между компьютерами передается программа. Наиболее популярными протоколами такого типа являются протоколы ЮМ-2848 и *BSC* компании ЮМ. Протокол

BSC работает в двух режимах - непрозрачном (некоторые специальные символы внутри кадра запрещены) и прозрачном (разрешена передача внутри кадра любых символов, в том числе и *ETX*). Прозрачность достигалась за счет байтстаффинга – перед управляющими символами *STX* и *ETX* всегда вставлялся символ *DLE* (*DATA LINK ESCape*). Такая процедура называется вставкой символов или. Если в поле данных кадра встречается последовательность *DLE ETX*, то передатчик удваивает символ *DLE*, т. е. порождает последовательность *DLE DLE ETX*. Приемник, встретив подряд два символа *DLE DLE*, всегда удаляет первый, но оставшийся *DLE* уже не рассматривает как начало управляющей последовательности, т. е. оставшиеся символы *DLE ETX* воспринимается как пользовательские данные.

4.2.5. Бит-ориентированные протоколы. Потребность в паре символов в начале и конце каждого кадра вместе с дополнительными символами *DLE* означает, что байт-ориентированная передача не эффективна для передачи двоичных данных, так как приходится в поле данных кадра добавлять достаточно много избыточных данных. Кроме того, формат управляющих символов для разных кодировок различен, например, в коде *ASCII* символ *S YN* равен 00101110, а в коде *EBCDIC* – 00110010. Следовательно, этот метод допустим только с определенным типом кодировки, даже если кадр содержит только двоичные данные. Чтобы это преодолеть используют более универсальный метод бит-ориентированной передачи. Этот метод в настоящее время применяется при передаче как двоичных, так и символьных данных.

На рис. 4.10 показаны схемы бит-ориентированной передачи, отличающиеся способом обозначения начала и конца каждого кадра. Схема, представленная на рис. 4.10, а, похожа на схему с символами *STX* и *ETX* в байт-ориентированных протоколах. Начало и конец каждого кадра отмечены одной и той же 8-битовой последовательностью – 01111110, называемой флагом. Термин «бит-ориентированный» используется потому, что принимаемый поток битов сканируется приемником на побитовой основе для обнаружения стартового флага, а затем во время приема для обнаружения стопового флага. Поэтому длина кадра в этом случае не обязательно должна быть кратна байту.

Чтобы обеспечить синхронизацию приемника, передатчик посылает последовательность байтов простоя (11111111), предшествующую стартовому флагу. Для достижения прозрачности данных в этой схеме необходимо, чтобы флаг не присутствовал в поле данных кадра. Это достигается с помощью приема, известного как вставка 0-го бита, – битстаффинга. Схема вставки бита работает только во время передачи поля данных кадра. Если эта схема обнаруживает, что подряд передано пять единиц, то она в любом случае автоматически вставляет дополнительный ноль. Поэтому последовательность 01111 никогда не появится в поле данных кадра. Аналогичная схема работает в приемнике и выполняет обратную функцию. Когда после пяти единиц обнаруживается ноль, он автоматически удаляется из поля данных кадра. Битстаффинг экономичнее байтстаффинга, так как вместо лишнего байта вставляется один бит, следовательно, скорость передачи пользовательских данных в этом случае снижается медленнее.

Во второй схеме (см. рис. 4.10, б) для обозначения начала кадра предусмотрен только стартовый флаг, а для определения конца кадра используется поле длины кадра, которое при фиксированных размерах заголовка и концевика чаще всего имеет смысл длины поля данных кадра. Эта схема наиболее применима в локальных сетях, в которых для обозначения факта незанятости среды вообще не передается никаких символов. Чтобы все остальные станции вошли в битовую синхронизацию, посылающая станция предваряет содержимое кадра последовательностью бит, известной как преамбула, которая состоит из чередования единиц и нулей 101010... Войдя в битовую синхронизацию, приемник исследует входной поток на побитовой основе, пока не обнаружит байт начала кадра 10101011, который выполняет роль символа *STX*. За этим байтом следует заголовок кадра, в котором в определенном месте находится поле длины поля данных. Таким образом, в этой схеме приемник просто отсчитывает заданное количество байт, чтобы определить окончание кадра.

Третья схема (см. рис. 4.10, в) для обозначения начала и конца кадра использует флаги, которые включают запрещенные для данного кода сигналы (*coDe VIoLatIoNS*, *V*). Например, при манчестерском кодировании вместо обязательного изменения полярности сигнала в середине тактового интервала уровень сигнала остается неизменным и низким (запрещенный сигнал *J*) или неизменным и высоким (запрещенный сигнал *K*). Начало кадра отмечается последовательностью *JK0JK000*, а конец – последовательностью *JK1JK111*. Этот способ очень экономичен, так как не требует ни битстаффинга, ни поля длины. Недостатком этого способа является то, что он зависит от принятого метода физического кодирования. При использовании избыточных кодов роль сигналов *J* и *K* играют запрещенные символы, например, в коде 4В/5В этими символами являются коды 11000 и 10001.

Каждая из трех схем имеет свои преимущества и недостатки. Флаги позволяют отказаться от специального дополнительного поля, но требуют специальных мер: либо по разрешению размещения флага в поле данных за счет битстаффинга, либо по использованию в качестве флага запрещенных сигналов, что делает эту схему зависимой от способа кодирования.

4.2.6. Протоколы с переменным форматом кадра. Существует ряд протоколов, в которых кадры имеют гибкую структуру. К таким протоколам относятся популярный прикладной протокол управления сетями *SNMP* и протокол канального уровня *PPP*, используемый для соединений типа «точка-точка». Кадры таких протоколов состоят из неопределенного количества полей, каждое из них может иметь переменную длину.

Для большей части протоколов характерны кадры, состоящие из служебных полей фиксированной длины. Исключение делается только для поля данных, так как возможны пересылки как небольших квитанций, так и больших файлов. Способ определения окончания кадра путем задания длины поля данных, рассмотренный выше, как раз рассчитан на такие кадры с фиксированной структурой и фиксированными размерами служебных полей.

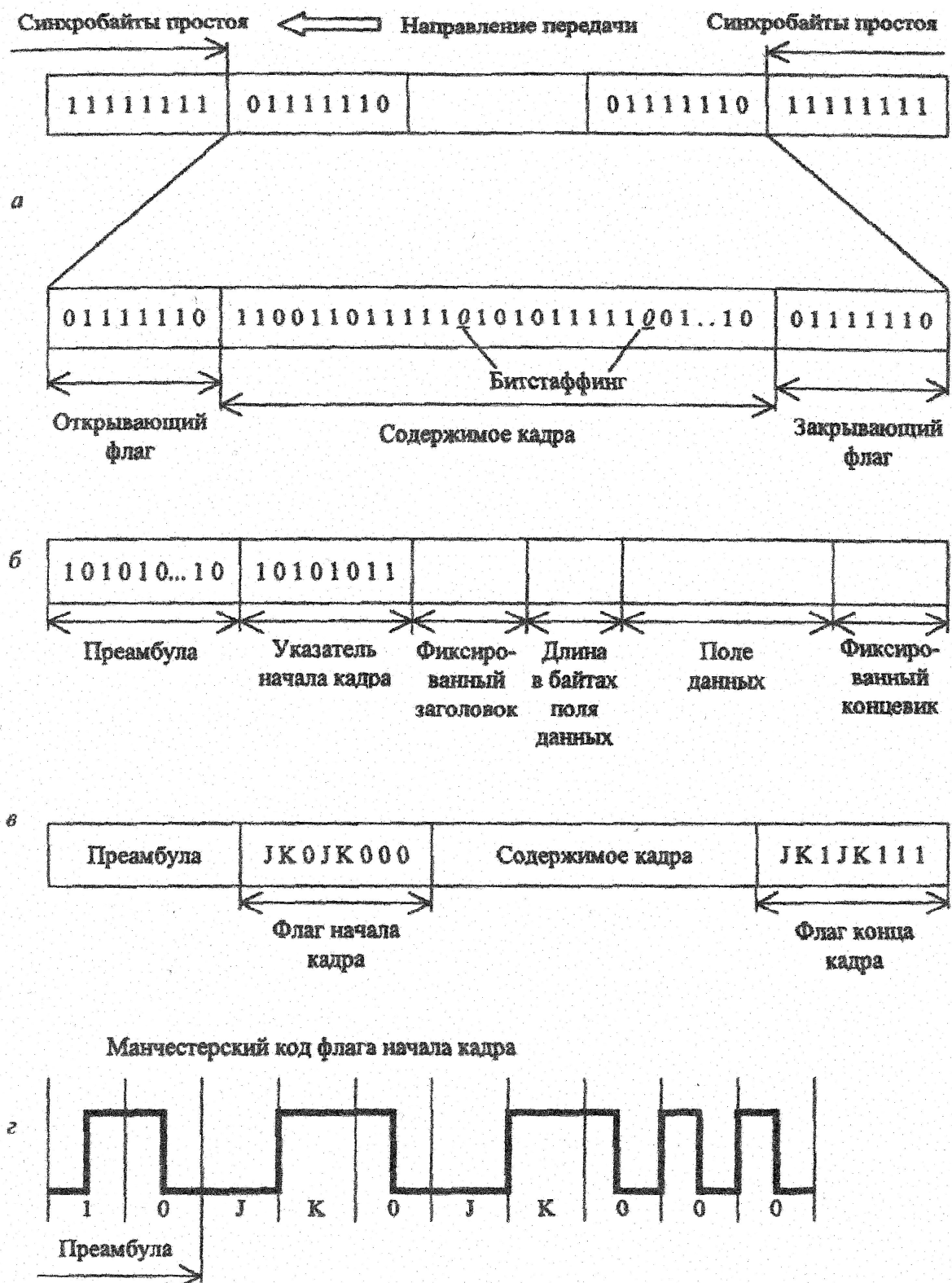


Рис. 4.10. Способы выделения начала и конца кадра в бит-ориентированных протоколах

4.2.7. Передача с установлением логического соединения и без установления логического соединения. При передаче кадров данных на канальном уровне используют дейтаграммные процедуры, работающие без установления логического соединения (*coNNectionLeSS*), и процедуры с предварительным установлением логического соединения (*coNNection-oRIented*) (рис. 4.11).

При дейтаграммной передаче кадр посылается в сеть «без предупреждения», и никакой ответственности за его утерю протокол не несет (см. рис. 4.11, а). Предполагается, что сеть всегда готова принять кадр от конечного узла. Дейтаграммный метод работает быстро, так как никаких предварительных действий перед отправкой данных не выполняется. Однако при таком методе трудно организовать в рамках протокола отслеживание факта доставки кадра узлу назначения. Этот метод не гарантирует доставку пакета.

Передача с установлением соединения более надежна, но требует больше времени для передачи данных и вычислительных затрат от конечных узлов. При такой передаче узлу-получателю отправляется служебный кадр специального формата с предложением установить соединение (рис. 4.11, б).

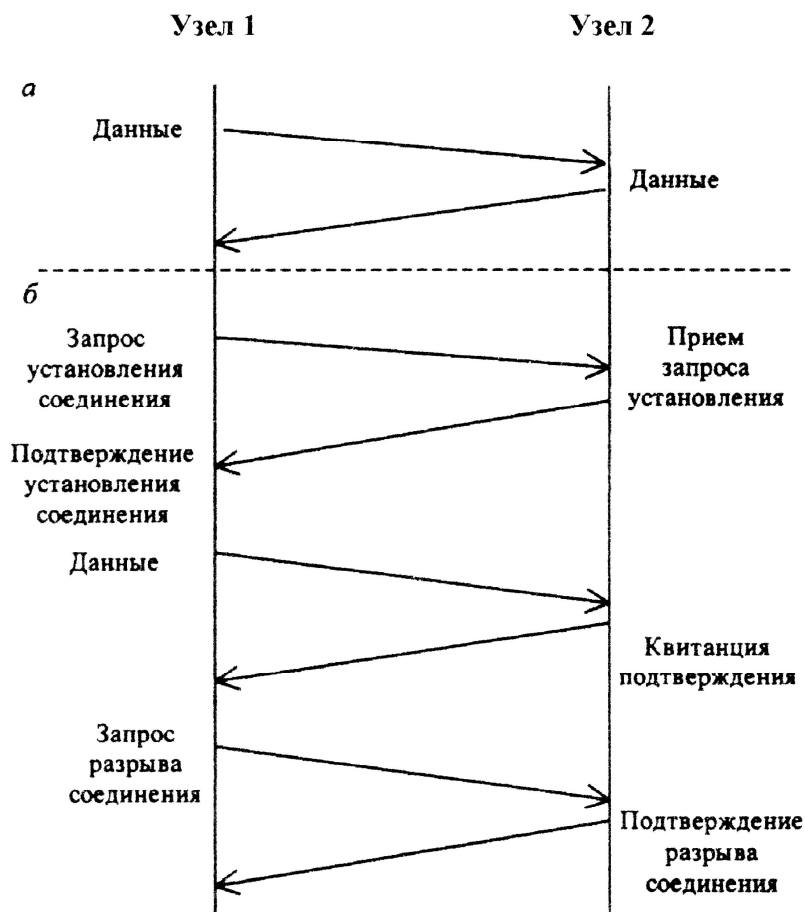


Рис. 4.11. Протоколы без установления соединения (а) и с установлением соединения (б)

Если узел-получатель согласен с этим, то он посылает в ответ другой служебный кадр, подтверждающий установление соединения и предлагающий для данного логического соединения некоторые параметры, например идентификатор соединения, максимальное значение поля данных кадров, которые будут использоваться в рамках данного соединения, и т. п. Узел-инициатор соединения может завершить процесс установления соединения отправкой третьего служебного кадра, в котором сообщит, что предложенные параметры ему подходят. На этом логическое соединение считается установленным, и в его рамках можно передавать информационные кадры с пользовательскими данными. После передачи некоторого законченного набора данных, например определенного файла, узел инициирует разрыв данного логического соединения, посылая соответствующий служебный кадр.

В отличие от протоколов дейтаграммного типа, которые поддерживают только один тип кадра – информационный, протоколы, работающие по процедуре с установлением соединения, должны поддерживать несколько типов кадров – служебные, для установления (и разрыва) логического соединения, и информационные, переносящие собственно пользовательские данные.

Процедура установления соединения используется:

- для взаимной аутентификации либо пользователей, либо оборудования (маршрутизаторы тоже могут иметь имена и пароли, которые нужны для уверенности в том, что злоумышленник не подменил корпоративный маршрутизатор и не отвел поток данных в свою сеть для анализа);
- для согласования изменяемых параметров протокола: MTU, различных тайм-аутов и т. п.;
- для обнаружения и коррекции ошибок. Установление логического соединения дает точку отсчета для задания начальных значений номеров кадров. При потере нумерованного кадра приемник, во-первых, получает возможность обнаружить этот факт, а во-вторых, может сообщить передатчику, какой в точности кадр нужно передать повторно.

В некоторых технологиях процедуру установления логического соединения используют при динамической настройке коммутаторов сети для маршрутизации всех последующих кадров, которые будут проходить через сеть в рамках данного логического соединения. Так работают сети технологий X.25, *FRAMe ReLaY* и *ATM*.

4.2.8. Обнаружение и коррекция ошибок. Канальный уровень должен обнаруживать ошибки передачи данных, связанные с искажением бит в принятом кадре данных или с потерей кадра, и по возможности их корректировать. Большая часть протоколов канального уровня выполняет только одну задачу – обнаружение ошибок, считая, что корректировать ошибки, т. е. повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают известные протоколы локальных сетей *EtheRNet*, *Token RING*, *FDDI* и др. Однако существуют протоколы канального уровня, например *LLC2* или *LAP-B*, которые самостоятельно решают задачу восстановления искаженных или потерянных кадров.

Очевидно, что протоколы должны работать наиболее эффективно в типичных условиях работы сети. Поэтому для сетей, в которых искажения и потери кадров являются очень редкими событиями, разрабатываются протоколы типа *EtheRNet*, где не предусмотрены процедуры устранения ошибок. Действительно, наличие процедур восстановления данных потребовало бы от конечных узлов дополнительных вычислительных затрат, которые в условиях надежной работы сети являлись бы избыточными.

Напротив, если в сети искажения и потери происходят часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок, а не оставлять эту работу протоколам верхних уровней. Протоколы верхних уровней, например транспортного или прикладного, работая с большими тайм-аутами, восстановят утерянные данные с большой задержкой. В глобальных сетях первых поколений, например сетях X.25, которые использовали ненадежные каналы связи, протоколы канального уровня всегда выполняли процедуры восстановления потерянных и искаженных кадров. Поэтому нельзя считать, что один протокол лучше другого потому, что он восстанавливает ошибочные кадры, а другой протокол нет. Каждый протокол должен работать в тех условиях, для которых он разработан.

Методы коррекции ошибок в вычислительных сетях основаны на повторной передаче кадра данных в случае, если кадр теряется и не доходит до адресата или приемник обнаружил в нем искажение информации. Чтобы убедиться в необходимости повторной передачи данных, отправитель нумерует отправляемые кадры и для каждого кадра ожидает от приемника так называемой *положительной квитанции* – служебного кадра, извещающего о том, что исходный кадр был получен и данные в нем оказались корректными. Время этого ожидания ограничено – при отправке каждого кадра передатчик запускает таймер, и, если по его истечении положительная квитанция не получена, кадр считается утерянным. Приемник в случае получения кадра с искаженными данными может отправить *отрицательную квитанцию* – что указывает на то, что данный кадр нужно передать повторно. Процесс обмена квитанциями называется автоматическим запросом повторения – *ARQ (AUtomatIc Repeat ReGUeSt)*.

Существуют два подхода к организации процесса обмена квитанциями: с простоями и с организацией «скользящего окна».

Метод с простоями (IDLE RQ) требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется. В этом случае производительность обмена данными существенно снижается, так как передатчик не может послать следующий кадр сразу же после отправки предыдущего, он обязан ждать прихода квитанции. Снижение производительности этого метода коррекции особенно заметно на низкоскоростных каналах связи (в территориальных сетях).

В методе «скользящего окна» (*SLIDING WINDOW*) для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, т. е. в максимально возможном для источника темпе, без получения на эти кадры положительных ответных квитанций. Количество кадров, которые разрешается передавать таким образом, называется размером окна. Рис. 4.12, а иллюстрирует данный метод для передающего окна размером в W кадров. В начальный момент, когда еще не послано ни одного кадра, окно определяет диапазон кадров с номерами от 1 до W включительно. Источник начинает передавать кадры и получать в ответ квитанции. Квитанции канального уровня поступают в той же последовательности, что и кадры, которым они соответствуют. В момент t_1 при получении первой квитанции K_1 окно сдвигается на одну позицию, определяя новый диапазон от 2 до $(W+1)$.

Процессы отправки кадров и получения квитанций происходят независимо друг от друга. Рассмотрим произвольный момент времени t_N , когда источник получил квитанцию на кадр с номером N . Окно сдвинулось вправо и определило диапазон разрешенных к передаче кадров от $(N+1)$ до $(W+N)$. Все множество кадров, выходящих из источника, можно разделить на перечисленные ниже группы (см. рис. 4.12, а).

Кадры с номерами от 1 до N уже были отправлены и квитанции на них получены, т.е. они находятся за пределами окна слева.

Кадры, начиная с номера $(N+1)$ и кончая номером $(W+N)$, расположены в пределах окна и потому могут быть отправлены, не дожидаясь прихода какой-либо квитанции. Этот диапазон можно разделить еще на два поддиапазона;

кадры с номерами от $(N+1)$ до m , которые уже отправлены, но квитанции на них еще не получены;

кадры с номерами от m до $(W+N)$, которые пока не отправлены, хотя запрета на это нет.

Все кадры с номерами, большими или равными $\{W+N+1\}$, находятся за пределами окна справа и поэтому пока не могут быть отправлены.

Перемещение окна вдоль последовательности номеров кадров показано на рис. 4.12, б. Каждый раз, когда приходит положительная квитанция, окно сдвигается, но его размер при этом не меняется и остается равным W . При отправке кадра с номером N источнику разрешается передать еще $W-1$ кадров до получения квитанции на кадр N , так что в сеть последним уйдет кадр с номером $(W+N-1)$. Если же за это время квитанция на кадр N так и не пришла, то процесс передачи приостанавливается, и по истечении некоторого тайм-аута кадр N (или квитанция на него) считается утерянным, и его передают снова. Если поток подтверждений поступает регулярно, в пределах допуска в W кадров, то скорость обмена достигает максимально возможной для данного канала и принятого протокола.

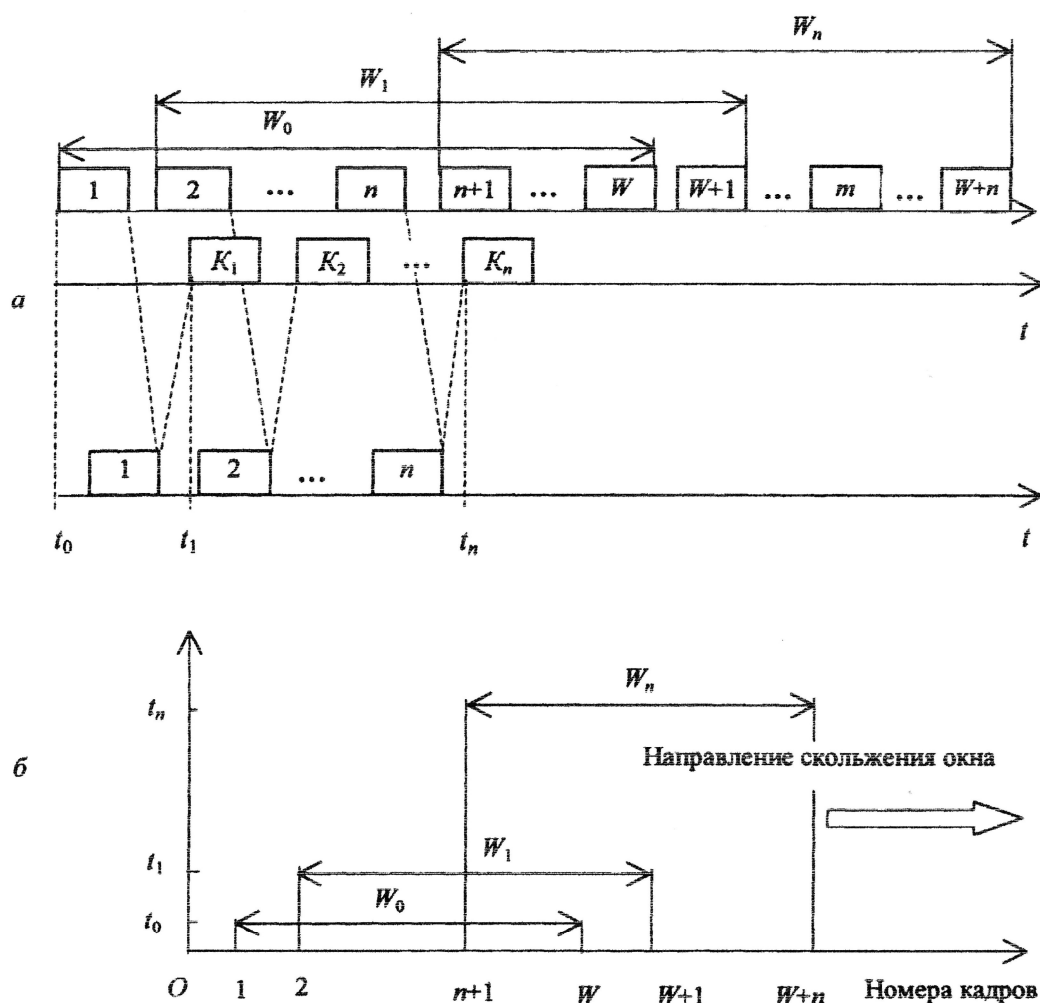


Рис. 4.12. Метод скользящего окна: t_0 – исходный момент, t_1 и t_N – моменты прихода квитанций на 1-й и N -й кадр соответственно

Метод «скользящего окна» более сложен в реализации, чем метод с простоями, так как передатчик должен хранить в буфере все кадры, на которые пока не получены положительные квитанции. Кроме того, требуется отслеживать несколько параметров алгоритма: размер окна W , номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции. Приемник может не посылать квитанции на каждый принятый корректный кадр. Если несколько кадров пришли почти одновременно, то приемник может послать квитанцию только на последний кадр. При этом подразумевается, что все предыдущие кадры также дошли благополучно.

Некоторые методы используют отрицательные квитанции, которые бывают двух типов - групповые и избирательные. Групповая квитанция содержит номер кадра, начиная с которого нужно повторить передачу всех кадров, отправленных передатчиком в сеть. Избирательная квитанция требует повторной передачи только одного кадра.

Метод с простоями является частным случаем метода «скользящего окна», когда размер окна равен единице. Метод «скользящего окна» имеет два параметра, которые могут заметно влиять на эффективность передачи данных меж-

ду передатчиком и приемником, - размер окна и величина тайм-аута ожидания квитанции. В надежных сетях, где кадры искажаются и теряются редко, для повышения скорости обмена данными размер окна можно увеличивать, так как при этом передатчик будет посылать кадры с меньшими паузами. В ненадежных сетях размер окна следует уменьшать, так как при частых потерях и искажениях кадров резко возрастает объем вторично передаваемых через сеть кадров, а значит, пропускная способность сети будет расходоваться во многом впустую - полезная пропускная способность сети будет падать.

Выбор тайм-аута зависит не от надежности сети, а от задержек передачи кадров сетью. Во многих реализациях метода «скользящего окна» величина окна и тайм-аут выбираются адаптивно, в зависимости от текущего состояния сети.

4.3. Технологии *DSL*

Поскольку в ближайшем будущем в подавляющем большинстве линий, служащих для соединения модемов пользователей с телефонной сетью, будет по-прежнему использоваться металлическая витая пара, основные усилия разработчиков направлены на создание методов, позволяющих осуществлять высокоскоростную передачу данных по локальным линиям связи. Результатом этих усилий стало появление семейства технологий, которые получили название *DSL (DIGITaL SUbSCRibeR LiNeS* — цифровые абонентские линии). Как известно, полоса пропускания речевого канала равна приблизительно 3000 Гц. Однако в действительности диапазон частот, поддерживаемый витой парой, с помощью которой обычно соединяется модем пользователя с АТС, значительно шире: в некоторых линиях доступа его размер достигает 1 МГц. Данной особенностью и решили воспользоваться разработчики технологий *DSL*. При передаче сигналов речевого диапазона на АТС применяются фильтры, поэтому можно использовать более широкий диапазон частот, которые поддерживаются линиями связи, а перед поступлением сигнала в телефонную сеть эти частоты отфильтровывать. Такая технология для передачи речи по-прежнему позволяет задействовать канал речевого диапазона.

При использовании технологий *DSL* скорость передачи данных зависит от таких факторов, как расстояние между модемом пользователя и АТС, категория витой пары (диаметр провода), а также метод модуляции, лежащий в основе работы *DSL*-модемов.

С начала 1997 года была разработана целая серия технологий *DSL*, которые можно разбить на шесть основных категорий (табл. 4.1).

Основные категории *DSL*

Название категории	Скорость передачи данных от модема пользователя	Скорость передачи данных к модему пользователя
Ассиметричная <i>DSL</i> (<i>ASymmetRIcaL DIGITaL SUbSCRibeR LiNe</i> , <i>ADSL</i>)	1 Мбит/с	8 Мбит/с
Высокоскоростная <i>DSL</i> (<i>HIGH BIT Rate DIGITaL SUbSCRibeR LiNe</i> , <i>HDSL</i>)	1,544/2,048 Мбит/с	1,544/2,048 Мбит/с
<i>G.LiTe</i>	512 кбит/с	1,4 Мбит/с
<i>DSL</i> с настраиваемой скоростью передачи (<i>Rate ADaptIVe DIGITaL SUbSCRibeR LiNe</i> , <i>RDSL</i>)	784 кбит/с	4 Мбит/с
Симметричная <i>DSL</i> (<i>SYmmetRIc DIGITaL SUbSCRibeR LiNe</i> , <i>SDSL</i>)	2 Мбит/с	2 Мбит/с
Сверхскоростная <i>DSL</i> (<i>VeRY-hIGH-bI-Rate DIGITaL SUbSCRibeR LiNe</i> , <i>VDSL</i>)	1,5 Мбит/с	52 Мбит/с

Далее каждая из технологий описывается более подробно.

- При использовании асимметричной *DSL* (*ADSL*) для передачи сигналов по направлению от модема пользователя к АТС выделяется полоса частот, которая несимметрична по отношению к полосе частот, предназначенной для передачи данных в противоположном направлении;

- Технология *G.LiTe* представляет собой более низкоскоростную версию асимметричной *DSL*. В ее основе лежит та же схема модуляции, что и в *ADSL*. Теоретически нет необходимости в установке пользователем разделителя, поскольку данной технологией предусмотрена поддержка низких значений скорости. Но в связи с тем, что соединение модема пользователя с АТС иногда все еще производится посредством устаревших медных проводов, для применения этой технологии обычно требуется наличие у пользователя разделителя;

- Высокоскоростную *DSL* (*HDSL*) можно рассматривать как эквивалент четырехпроводных каналов Т1 и Е1. В ней полоса пропускания, выделенная для передачи данных в одном направлении, симметрична полосе пропускания, которая выделяется для передачи данных в другом направлении. При использовании технологии *HDSL* модемы пользователей могут находиться на большом расстоянии от АТС, причем наличия повторителей для этого не требуется. Поэтому данная технология быстро получила признание у телефонных компаний, которые ранее в локальных линиях связи были вынуждены использовать повторители, за счет чего повышалась стоимость предоставляемых услуг;

- *DSL* с настраиваемой скоростью передачи (*RDSL*) является одной из последних версий *DSL* и позволяет модемам не только автоматически настраиваться на максимально допустимую для имеющейся локальной линии связи скорость передачи, но и работать с фиксированной скоростью;

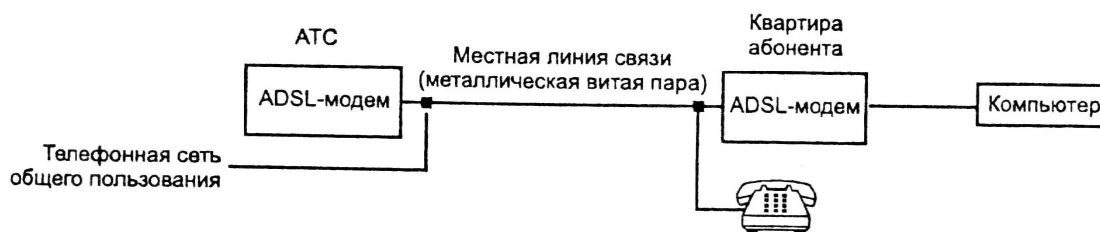
- В симметричной *DSL* (*SDSL*), так же как и в *HDSL*, полоса пропускания, выделенная для передачи данных в одном направлении, симметрична полосе пропускания, выделенной для передачи данных в другом направлении. Однако в отличие от *HDSL*, где используется четыре пары проводов, в *SDSL* данные передаются по одной паре проводов;

- При использовании сверхскоростной *DSL* (*VDSL*) достигается наивысшая скорость передачи данных по направлению от АТС к модему пользователя — 52 Мбит/с. Однако эта технология применима только в тех случаях, когда данные передаются на небольшие расстояния — меньше тысячи футов.

Из шести перечисленных выше технологий наибольшее распространение получила *ADSL*, так как она идеально подходит и для путешествий по *Web*, и для работы с асимметричными приложениями (приложениями, при функционировании которых используется асимметричная передача). В настоящее время большинство телефонных компаний предлагают абонентам именно эту технологию, поэтому в оставшейся части данного раздела мы рассмотрим ее более детально.

4.3.1. Технология *ADSL*. Для использования *ADSL*, равно как и для применения других технологий *DSL*, требуется наличие двух модемов, соединенных локальной линией связи — модема пользователя и модема, расположенного на АТС. В настоящее время на рынке имеется несколько моделей модемов, поддерживающих технологию *ADSL*. Основное различие между ними состоит в применяемых методах модуляции, а также в том, что для использования некоторых модемов требуется установить специальные фильтры, с помощью которых устраняется негативное влияние на речевой сигнал частот, предназначенных для передачи данных.

Схема соединения через местную линию связи модемов, работающих на основе технологии *ADSL*, изображена на рис. 4.12. Ниже показаны полосы частот, выделенные из диапазона, поддерживаемого локальной линией связи. При этом местную линию связи можно разделить на три канала: высокоскоростной, посредством которого данные передаются в направлении к модему пользователя (для него выделена большая часть полосы пропускания), среднескоростной, по которому данные пересылаются в обратном направлении, и стандартный канал речевого диапазона (с полосой пропускания 4 кГц).



а



б

Рис. 4.12. Технология ADSL: а – схема соединения модемов;
б – используемые полосы частот

Фактическая скорость передачи данных, которую можно достичь в местной телефонной линии (рис. 4.12), зависит от длины абонентской линии, диаметра используемого в ней провода, наличия или отсутствия отводов, а также от уровня помех. Поскольку коэффициент затухания сигнала прямо пропорционален длине абонентской линии и частоте и обратно пропорционален диаметру используемого провода, при сравнении производительности двух абонентских линий следует учитывать эти параметры. В табл. 4.2 перечислены значения скорости, с которой могут обмениваться данными модемы, поддерживающие технологию ADSL, и параметры, необходимые для достижения этих значений.

Таблица 4.2

Поддерживаемые технологией ADSL параметры передачи данных

Скорость передачи данных		Тип провода	Длина абонентской линии, футов (метры)
от модема к АТС, кбит/с	от АТС к модему, Мбит/с		
64-384	2,0	24 AWG	18 000 (6 000)
64-384	2,0	26 AWG	15 000 (4 500)
64-384	8,0	24 AWG	12 000 (3 600)
64-384	8,0	26 AWG	9 000 (2 700)

Хотя на рис. 4.12 в качестве оборудования АТС изображен только один модем, поддерживающий технологию *ADSL*, на самом деле для обеспечения максимальной эффективности работы используется несколько модемов, смонтированных в стойке. Кроме того, на узле связи с помощью устройства, которое получило название мультиплексор доступа через цифровые абонентские линии (*DIGITaL SUBSCRibeR LiNe AcceSS MULtIpLeXeR, DSLAM*), выполняется объединение трафика, поступающего от группы модемов. Большинство устройств *DSLAM* поддерживается сразу несколько категорий *DSL*, таких как *ADSL*, *HDSL*, *G.LiTe* и *SDSL*, что дает этим устройствам возможность работать с трафиком, поступающим от модемов абонентов всех перечисленных служб. Во многих устройствах *DSLAM* используется протокол *ATM*, но в отдельных из них для передачи данных применяется протокол *IP*. Однако независимо от используемого протокола основной функцией устройства *DSLAM* является объединение потоков данных, которые поступают из нескольких абонентских линий, и передача их в виде одного потока Всемирной сети, к которой устройство *DSLAM* последовательно подключено через высокоскоростной канал.

В *ADSL*-модемах применяются две конкурирующие технологии модуляции: дискретная многотональная модуляция (*DISCRete MIRLtIToNe, DMT*) и амплитудно-фазовая модуляция без несущей (*CaRRieRLeSS AmpLiTUDe PhaSe, CAP*). Модуляция *DMT* является стандартом *ANSI*, в то время как модуляция *CAP* представляет собой запатентованную технологию, разработанную корпорацией *PaRaDYNe*. Применение каждой из этих технологий позволяет вести высокоскоростную передачу данных, используя мультиплексирование с разделением частот (*FDM*), посредством которого диапазон частот, поддерживаемый местной линией связи, подразделяется по частотам на независимые каналы. (Каналы, полученные при таком разделении, показаны на рис. 4.12, б.) При использовании *FDM* создается три канала: один - для передачи данных в направлении от АТС к модему пользователя, другой - для передачи информации в обратном направлении, третий — для передачи речевой информации. В свою очередь, оба канала, предназначенные для пересылки данных, могут дополнительно подразделяться на подканалы с помощью мультиплексирования с разделением времени. Эти подканалы можно использовать для одновременной передачи данных нескольким цифровым устройствам, например компьютеру и системе видеоконференций.

4.4. Технологии мультиплексирования

До этого момента мы в основном рассматривали процесс обмена данными между двумя устройствами, например персональными компьютерами или телефонными аппаратами. Но понятно, что если владельцы коммуникационных каналов для каждой пары пользователей проложат отдельную линию, то стоимость предоставляемых служб будет непомерно высокой, а окупаемость каналов очень низкой. Поэтому на всех участках каналов, где несколько потоков данных передаются в одном направлении, используется мультиплексирование -

операция, позволяющая существенно повысить производительность канала. Мультиплексирование предоставляет механизм для совместного использования одного канала несколькими пользователями.

В начале этого подраздела будут рассмотрены используемые в коммуникационных каналах технологии мультиплексирования — мультиплексирование с разделением частоты, или частотным разделением (*FReQUeNcY-DIVISIoN MULtIpLeXING*, *FDM*) и мультиплексирование с разделением времени, или временным разделением (*TIme-DIVISIoN MULtIpLeXING*, *TDM*). Затем мы проанализируем варианты применения этих технологий, в частности использование статистических мультиплексоров с разделением времени (*StatIStIcaL TIme-DIVISIoN MULtIpLeXING*, *STDM*), а в завершение расскажем об относительно новой категории устройств, которые получили название низкоскоростных мультиплексоров речь/данные, позволяющих передавать по одному каналу связи и речевую информацию, и данные. Чтобы вы лучше поняли, как работают такие мультиплексоры, в конце главы мы рассмотрим несколько методов оцифровки речи.

4.4.1. Мультиплексирование с разделением частоты. Пример использования мультиплексирования с разделением частоты для передачи по коммуникационному каналу трех телефонных разговоров приведен на рис. 6.1. Частоты каждого из трех мультиплексируемых каналов речевого диапазона смещаются таким образом, чтобы в широкополосном канале связи их можно было передавать одновременно в виде одной группы. На другом конце канала связи выполняется восстановление первоначальных частот. В течение многих лет для мультиплексирования телефонных разговоров применялась именно технология *FDM*, поскольку она позволяет использовать имеющуюся полосу частот более эффективно, чем это делается в системах цифровой передачи. Но системы *FDM* имеют существенный недостаток: при усилении речевого сигнала одновременно повышается и уровень шума. По этой причине, а также в связи со снижением стоимости используемых в цифровой связи аппаратных средств системы *FDM* стали повсеместно вытесняться системами мультиплексирования с разделением времени.

Но все-таки главной причиной, по которой в течение последних трех десятилетий наблюдается спад интереса к *FDM*, является повсеместная замена аналоговых линий связи цифровыми. Следствием таких замен стала необходимость оцифровки речи для ее передачи в цифровом формате от одного абонента другому (в пункте назначения выполняется обратная процедура — преобразование цифрового сигнала в аналоговый). К 1990 году компании *AT&T* и *MCI* заменили 90% своих аналоговых линий цифровыми, а компания *SpRINt* заменила таковыми все свои аналоговые линии. В середине 90-х годов в США через цифровые линии осуществлялись все междугородние коммуникации, такая же тенденция наблюдается и в Европе.

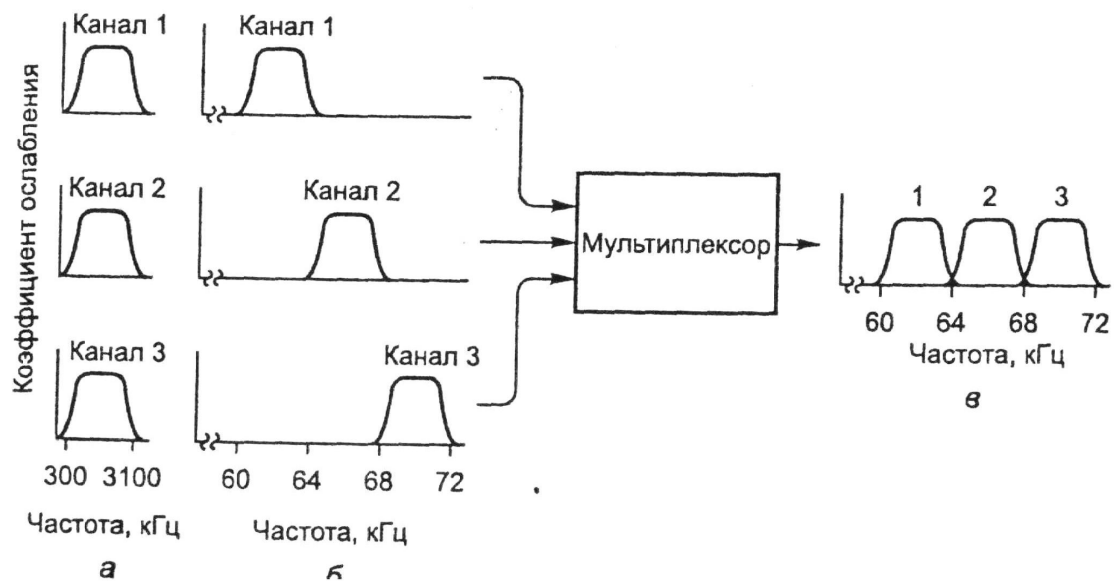


Рис. 4.13. Мультиплексирование с разделением частоты: а – первоначальные полосы частот каждого канала; б – смещение по частоте; в – расположение частот после мультиплексирования

Однако, перефразировав известное изречение Марка Твена еще раз, можно сказать, что слухи о смерти мультиплексирования с частотным разделением несколько преувеличены. Хотя в настоящее время и наблюдается очевидный спад интереса к *FDM*, но появилась новая технология — оптическая передача, — в основе которой лежит использование именно мультиплексирования с разделением частоты. В соответствии с этой технологией, *FDM* используется для передачи сигналов, имеющих различную длину волн (речь идет об электромагнитных сигналах видимого диапазона), по единому волоконно-оптическому кабелю. Чтобы понять, как это происходит, достаточно вспомнить опыт из школьного курса физики: если луч дневного света пропустить через призму, то на стоящем за призмой экране, появится полоса (спектр), состоящая из всех цветов радуги. То есть дневной свет состоит из электромагнитных волн различной длины (иначе говоря, сигналов, имеющих различные частоты). Нужную волну нетрудно выделить и передавать данные с ее помощью по волоконно-оптическому кабелю.

Это, конечно, упрощенное объяснение технологии *FDM*, но применительно к волоконно-оптической среде передачи оно в целом довольно точно отражает суть происходящих процессов. Используемое в волоконно-оптической среде передачи мультиплексирование с частотным разделением называется мультиплексированием с разделением по «длине волн» (*WaVeLeNGth-DIVISIoN MULtIpLeXING*, *WDM*). Но поскольку длина волны является величиной, обратно пропорциональной частоте ($\lambda = 1/f$), передача по оптоволокну сигналов различной длины происходит с использованием различных частот, то есть применяется технология *FDM*.

4.4.2. Мультиплексирование с разделением времени.

Уже само название устройств этого типа указывает на то, что во время работы они используют мультиплексирование по времени. Рисунок 6.10 иллюстрирует процесс демультиплексирования данных, полученных мэйнфреймом, который расположен в одном пункте, от трех удаленных терминалов, расположенных в другом пункте. Для того чтобы упростить изложенный материал, примем, что терминал 1 передает последовательность символов *BA*, терминал 2 – последовательность *DC*, а терминал 3 – последовательность *FE*. После первого сканирования портов, к которым подключены терминалы, *TDM*-мультиплексор 1 обнаруживает, что на порт 1 поступил символ *A*, на порт 2 – символ *C*, а на порт 3 – символ *E*. Далее *TDM*-мультиплексор принимает данные каждого порта и формирует передаваемый кадр, содержащий последовательность символов *ECA*. Затем процесс повторяется – с портов считываются символы *F*, *D* и *B* и формируется второй кадр. Фактически каждый *TDM*-кадр содержит также символы синхронизации, предшествующие пользовательским данным, и данные, полученные в результате одного или нескольких сканирований.

На принимающем *TDM*-мультиплексоре 2 (также может использоваться де-мультиплексор) демультиплексирование данных выполняется с учетом их позиции в кадре. Поскольку символ *A* находится в кадре 1 на первой позиции (рис. 4.14), этот байт информации подается на порт 1 мультиплексора. Аналогичным образом обрабатываются и два оставшихся символа первого кадра – символ *C* подается на порт 2, а символ *E* – на порт 3. Затем *TDM*-мультиплексор 2 проверяет данные, содержащиеся во втором кадре, и пересылает их на соответствующие порты: первый байт (*B*) на порт 1, второй байт (*D*) на порт 2, а байт *F* на порт 3.

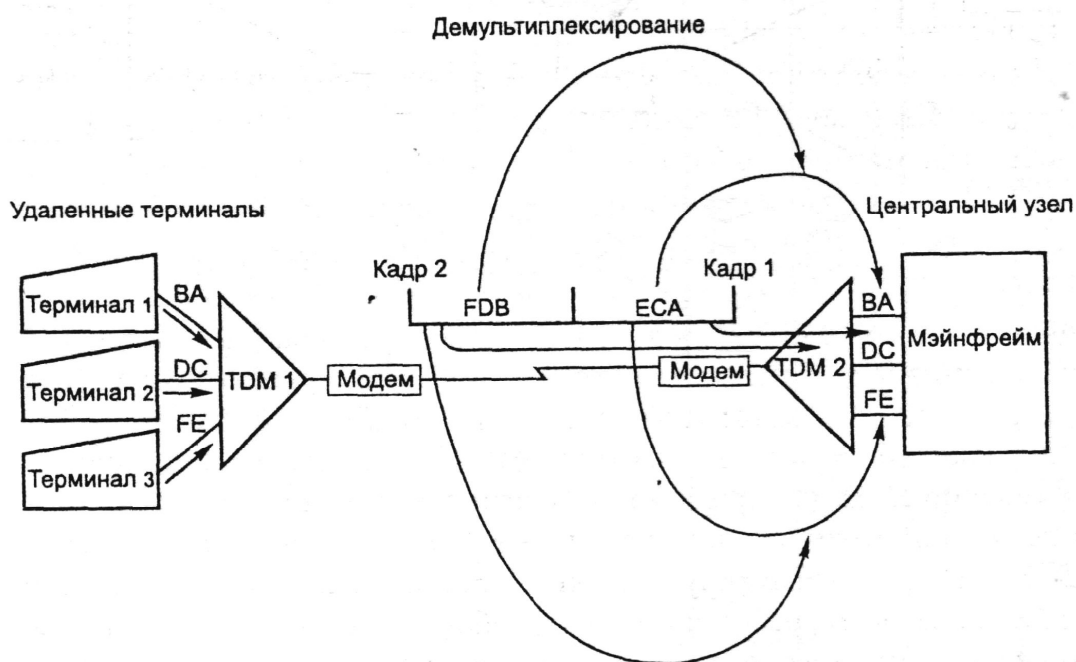


Рис. 4.14. Мультиплексирование и демультиплексирование данных

Магистраль, показанная на рис. 4.15, называется системой T1. Эта система цифровой передачи изначально предусматривала использование операции мультиплексирования для передачи данных 24 речевых каналов по одной высокоскоростной линии.

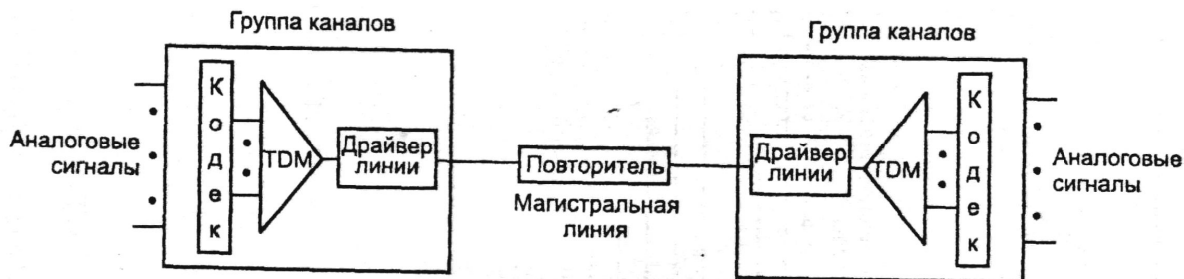


Рис. 4.15. Система передачи данных посредством групп каналов

Выборка каждого речевого сигнала выполняется с частотой 8000 раз в секунду, а для представления данных каждой выборки используется 8 бит. Значит, скорость оцифровки в каждом речевом канале составляет $8000 \cdot 8 = 64$ кбит/с (суммарно за 1 секунду по 24 каналам передается $64 \text{ кбит/с} \cdot 24 = 1,536$ Мбит пользовательских данных). А поскольку кадрирующие биты передаются по каналу со скоростью 8000 бит/с, рабочая скорость системы T1 составляет $1,536 \text{ Мбит/с} + 8 \text{ кбит/с} = 1,544 \text{ Мбит/с}$.

До начала 80-х годов система T1 использовалась только в больших магистральных коммуникационных линиях, но впоследствии стала применяться и для прокладки коммуникаций между коммерческими организациями.

Принцип использования T1-мультиплексора иллюстрирует рис. 4.16. Обратите внимание на наличие в нем PCM-карт, с помощью которых выполняется оцифровка аналоговых речевых сигналов, поступающих с офисной АТС. При наличии цифровой офисной АТС речевой сигнал может подаваться на мультиплексор, минуя PCM-карту.

В T1-мультиплексоре данные обрабатываются с применением однополярного NRZ-кодирования. И хотя этот формат представления данных широко используется в компьютерах, электронном оборудовании и для передачи Данных между терминальным оборудованием и оборудованием передачи данных, Для передачи по системе T1 он не пригоден.

При прокладке первых линий между группами каналов у компаний, владеющих коммуникационными средствами, появилось вполне естественное желание передавать по одной линии и цифровые сигналы, и напряжение питания (электричество). Было предложено для этой цели использовать трансформаторы, которые должны располагаться в каждой группе каналов (на каждом конце линии) и отделять сигналы от передаваемого напряжения питания. Однако для реализации этой технологии необходимо, чтобы при передаче сигналов в линии отсутствовало остаточное постоянное напряжение. Поэтому был разработан

новый метод линейного кодирования - инверсия чередующихся единиц (*ALteRNate MaRk INVeRSIoN*, *AMI*), или биполярное кодирование.

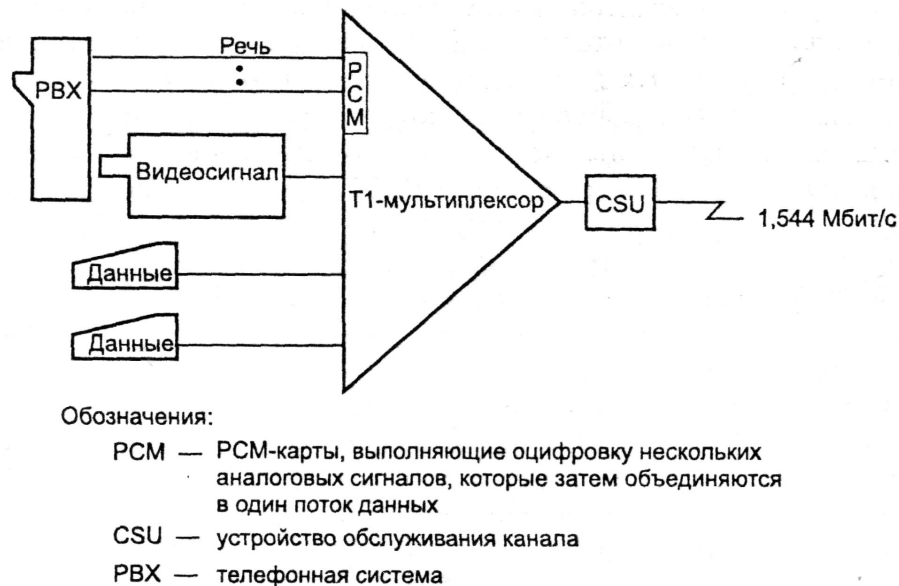


Рис. 4.16. Мультиплексор в системе T1

При прокладке первых линий между группами каналов у компаний, владеющих коммуникационными средствами, появилось вполне естественное желание передавать по одной линии и цифровые сигналы, и напряжение питания (электричество). Было предложено для этой цели использовать трансформаторы, которые должны располагаться в каждой группе каналов (на каждом конце линии) и отделять сигналы от передаваемого напряжения питания. Однако для реализации этой технологии необходимо, чтобы при передаче сигналов в линии отсутствовало остаточное постоянное напряжение. Поэтому был разработан новый метод линейного кодирования – инверсия чередующихся единиц (*ALteRNate MaRk INVeRSIoN*, *AMI*), или биполярное кодирование.

В большинстве T1-мультиплексоров выполняется преобразование однополярных NRZ-сигналов в биполярные AMI-сигналы для их последующей передачи по магистрали. Однако непосредственно перед передачей цифровой сигнал должен быть помещен в кадр того формата кадрирования (*D4* или *ESF*), который поддерживается системой цифровой передачи. Эту функцию выполняет устройство обслуживания канала (*CSU*), которое, кроме того, предназначено для хранения эксплуатационных данных (при использовании формата кадрирования *ESF*), распознавания сетевых кодов, передаваемых коммуникационной системой или устройствами, подключенными к системе T1, и формирования ответа на эти коды, а также для передачи минимального количества двоичных единиц (обеспечения плотности единиц).

Системы TDM имеют свои ограничения, для понимания природы которых необходимо рассмотреть работу терминальных устройств. Предположим, что в

филиале фирмы имеется несколько терминалов. В какой-то промежуток времени может сложиться ситуация, когда одни пользователи читают файлы справки, пытаюсь определить, как работать с установленным программным обеспечением или как ответить на сообщение, требующее ввода данных в определенном формате, другие пользователи старательно вводят данные или принимают их, а третьи просто устроили себе небольшой перерыв. То есть в любой момент времени существует большая вероятность того, что на каком-то компьютере (компьютерах) не будет производиться ни передача данных, ни их прием. Поскольку в соответствии с технологией *TDM* демультиплексирование данных выполняется на основе их позиции в кадре, отсутствие активности на некоторых портах может привести к неправильной интерпретации данных. Чтобы избежать таких ситуаций, при сканировании передающий *TDM*-мультиплексор вставляет в соответствующие неактивным терминалам позиции в кадре символы *NULL*.

На принимающем мультиплексоре символы *NULL* отбрасываются. А соответствующим подключенным устройствам просто ничего не передается.

Ввод символа *NULL* позволяет корректно выполнять демультиплексирование, однако канал связи при этом используется неэффективно. Например, если несколько пользователей терминалов не передают данные, большинство пересылаемых кадров содержат избыточное количество символов *NULL*. Следовательно, коммуникационный канал между *TDM*-мультиплексорами большую часть времени используется очень неэффективно. Этот же недостаток присущ и *T1*-мультиплексорам, однако благодаря высокой скорости передачи (1,544 Мбит/с) они могут обслуживать одновременно несколько сотен устройств, что несколько компенсирует неэффективность использования канала. *TDM*-мультиплексоры, работающие в аналоговых сетях или сетях службы *DDS*, поддерживают передачу данных на значительно меньших скоростях и могут одновременно обслуживать гораздо меньшее количество устройств. Поэтому, стремясь добиться большей производительности устройств при функционировании в этих сетях, фирмы-поставщики разработали еще один тип мультиплексоров — статистические мультиплексоры с разделением времени (*Statistical Time-DIVISION MULTIpLeXer, STDM*). При их применении достигается значительно большая эффективность передачи данных, чем при использовании *TDM*-мультиплексоров.

4.4.3. Статистическое мультиплексирование с разделением времени.

Если в технологии *TDM* кадры имеют фиксированную длину, а демультиплексирование производится на основе позиций данных в кадре, то в технологии *STDM* используются кадры переменной длины. Для того чтобы вы поняли, как работает *STDM*-мультиплексор, мы опишем процесс создания таких кадров, а затем рассмотрим один из нескольких методов формирования информации в мультиплексированном кадре.

Предположим, что к *STDM*-мультиплексору подключено восемь терминалов (рис. 4.17, а) и в течение первой операции сканирования активными были терминалы 1, 2, 3 и 6, с портов которых считаны символы *X*, *Y*, *Z* и *Q*, соответ-

ственно. Во время выполнения сканирования в кадр *STDM*-мультиплексора помещается битовая карта. Если при сканировании было обнаружено, что у терминала 1 есть данные, подлежащие передаче, первому биту битовой карты присваивается значение 1. Если же терминал не проявляет активности, соответствующему биту битовой карты присваивается значение 0.

Структура кадра, используемого при *STDM*-мультиплексировании, показана на рис. 4.17, б. Благодаря внедрению битовой карты размер передаваемого кадра в тех случаях, когда некоторые из терминалов не проявляют активности, можно значительно уменьшить. Так, в рассматриваемом нами случае при сканировании выяснилось, что активными являются терминалы 1, 2, 3 и 6, следовательно, в битовой карте битам, занимающим позиции 1, 2, 3 и 6, присваиваются значения 1. Одновременно с присвоением биту этого значения в кадр переменной длины добавляется символ, считанный с порта, к которому подключен активный терминал. То есть при использовании битовых карт абсолютно точно указывается, какой терминал какие данные передает, а значит, нет необходимости добавлять в кадр символы *NULL*.

В целом *STDM*-мультиплексоры могут обслуживать в два, а то и в четыре раза больше источников асинхронных данных, чем традиционные *TDM*-мультиплексоры. При оценке производительности работы мультиплексоров используется термин коэффициент нагрузки (*SeRVIce RatIo*). *STDM*-технология предусматривает не только сокращение размеров передаваемых кадров при слабой активности терминалов, но и удаление из каждого передаваемого символа стартовых и стоповых битов, а также битов четности с последующим восстановлением прежней структуры переданных данных при демультиплексировании. Кроме перечисленных выше функций, некоторые *STDM*-мультиплексоры выполняют сжатие данных, что еще больше повышает их коэффициент обслуживания.

Рассмотрим наихудший из возможных сценариев, при котором каждый из терминалов является персональным компьютером. Предположим, что в какой-то момент времени пользователи одновременно начали передавать по каналу файлы больших размеров. В этой ситуации суммарная скорость поступления данных на вход *STDM*-мультиплексора составит $2\,400 \times 8 = 19\,200$ бит/с, и он будет непрерывно получать все новые и новые данные. Даже если отбросить все стартовые и стоповые биты, а также биты четности, суммарная скорость поступления данных на мультиплексор будет заметно превышать пропускную способность канала. В *STDM*-мультиплексоре имеется буфер памяти, предназначенный для хранения подлежащих передаче данных, но если каждый из терминалов будет продолжать вести интенсивную передачу данных, буфер переполнится, что приведет к потере части данных. Чтобы предотвратить такую, *STDM*-мультиплексор начинает управлять потоком входных данных, то есть при определенной степени заполнения буфера временно отключает один или более терминалов (передает команду остановить передачу). После того как часть данных будет передана в канал и в буфере появится свободное простран-

ство, *STDM*-мультиплексор разрешит ожидающим терминалам возобновить передачу своих данных.

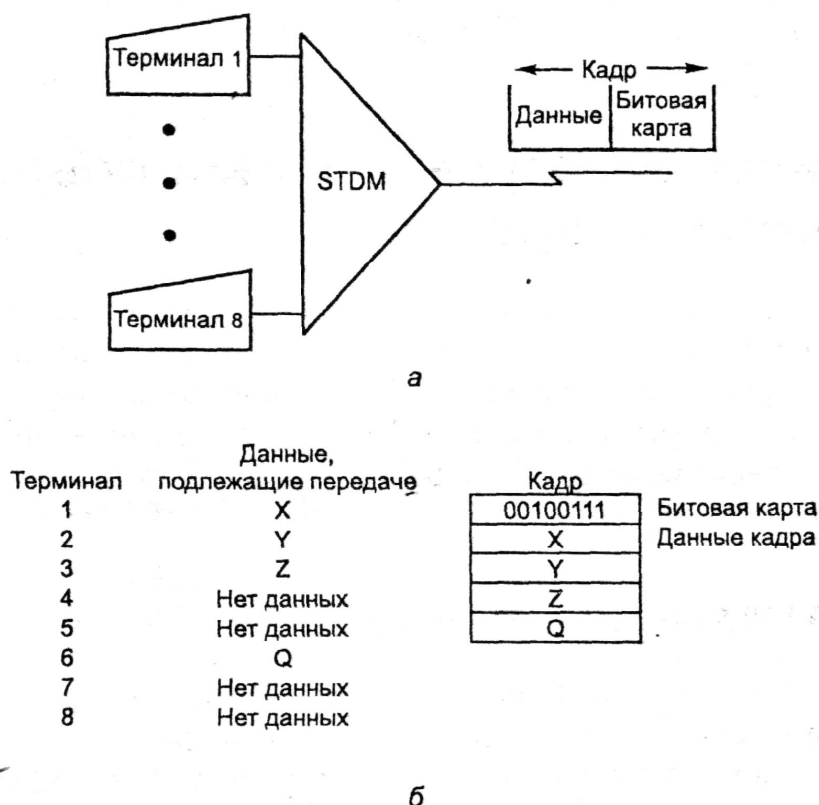


Рис. 4.17. *STDM*-мультиплексирование: а – передача кадра; б – структура кадра

Благодаря тому что на входе *STDM*-мультиплексоры могут получать данные со скоростью, которая в два-четыре раза превышает скорость передачи данных по коммуникационному каналу, они стали очень популярны. Обычные же *TDM*-мультиплексоры (например, *T1*-мультиплексоры) используются преимущественно в тех случаях, когда приходится работать с приложениями, очень чувствительными к задержкам при передаче данных.

Как уже говорилось ранее, почти все *TDM*-мультиплексоры были вытеснены *STDM*-мультиплексорами. Но прогресс не стоит на месте - со временем появились новые технологии, и в настоящее время мы являемся свидетелями того, как вытесняются маршрутизаторами устройства статического мультиплексирования. Правда, некоторыми *STDM*-мультиплексорами маршрутизация поддерживается, однако это лишь частичная поддержка, поэтому возможности таких мультиплексоров крайне ограничены. В частности, *STDM*-мультиплексоры не пригодны для создания сложных сетей. Кроме того, маршрутизаторы поддерживают соединения между локальными сетями, в то время как *STDM*-мультиплексоры разрабатывались прежде всего для поддержки прямых или коммутируемых соединений. На сегодняшний день технология *STDM* в основном используется в устройствах, где должно производиться мультиплексирование речевого сигнала. То есть в данном случае мультиплексирование применя-

ется как дополнительная функция устройств, которые установлены в каналах, соединяющих офисы небольших фирм.

4.4.4. Кадрирование. За последние четыре десятилетия для обеспечения синхронизации при передаче между группами каналов было разработано несколько форматов кадрирования. До конца 1980-х годов самым популярным был формат кадрирования *D4*, но затем в системах T1 стал использоваться формат *ESF*, который в настоящее время является наиболее распространенным.

В соответствии с форматом *D4* в начале каждого кадра находится кадрирующий бит (*F* на рис. 4.18, а). Так как у нас имеется 24 байта по 8 бит, каждый байт представляет собой оцифрованное значение выборки аналогового сигнала для 24 речевых каналов. Следовательно, кадр формата *D4* состоит из $1+24*8=193$ бит. Кадрирующие биты каждых 12 кадров используются для формирования специального значения, с помощью которого группы каналов выполняют синхронизацию при приеме данных, передаваемых группой каналов, которая расположена на другом конце магистральной линии.

В суперкадре формата *D4* кадрирующими битами могут поочередно назначаться терминальные кадрирующие биты (*FT*) и сигнальные кадрирующие биты (*FS*). Как показано в нижней части рис. 4.18, терминальные кадрирующие биты состоят из кадрирующих битов, предшествующих данным нечетных кадров. Используемые биты *FT* формируют состоящее из чередующихся единиц и нулей альтернативное специальное значение, с помощью которого можно определить, где заканчивается один кадр и начинается другой. Поэтому последовательность терминальных кадрирующих битов называется также сигналом выравнивания кадра.

F1	192D	F2	192D	...	F11	192D	F12	192D	F1	192D

$1 \text{ кадр} = 1 \text{ кадрирующий бит} + 192 \text{ бита данных}$									

Суперкадр = кадры с F1 по F12

а

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
1	0	0	0	1	1	0	1	1	1	0	0

Значение терминальных кадрирующих битов (FT) равно 101010

Значение сигнальных кадрирующих битов (FS) равно 001110

б

Рис. 4.18. Кадр формата *D4*: а – структура кадра; б – значение кадрирующих битов

Сигнальные кадрирующие биты, формирующие значение, например 001110 (рис. 4.18), предшествуют данным четных кадров. Эти биты предназначены для определения границ кадров, а также для извлечения из кадров управляющих (сигнальных) битов. Сигнальные биты используются для передачи информации о том, снята ли трубка, или, скажем, о ходе установления соединения.

Еще один формат, называемый расширенным суперкадром (*EXteNDeD SuPeR-FRAME, ESF*), охватывает 24 кадра и представляет собой дополненный формат *D4*. Однако в отличие от *D4*, в котором используются фиксированные значения кадрирующих битов, в *ESF* применяются как фиксированные, так и переменные значения. Из 24 кадрирующих битов расширенного суперкадра только 6 бит предназначены для формирования значения кадрирования и синхронизации, остальные служат для передачи данных, используемых при мониторинге сети и определении частоты ошибок передачи.

В табл. 4.3 указано назначение каждого из 24 кадрирующих битов формата *ESF*. Обратите внимание на символы *D* в столбце - «Используемый бит». Они обозначают биты, предназначенные для формирования канала передачи данных между группами каналов или двумя устройствами *CSU* (устройствами обслуживания канала), подключенными к каналу *T1*. Этот канал используется для активизации или деактивизации петлевого контроля и ретрансляции запроса на передачу удаленными устройствами *CSU* эксплуатационных данных, а также для передачи этих данных запрашивающим устройствам. Такие *D*-биты присутствуют в начале только нечетных кадров (1, 3, 5,... 21, 23), то есть из 24 кадрирующих битов формата *ESF* *D*-битами являются только 12. Они обеспечивают скорость передачи 4 кбит/с.

Таблица 4.3.

Назначения кадрирующих битов формата *ESF*

Номер кадра	Используемый бит	Номер кадра	Используемый бит
1	<i>D</i>	13	<i>D</i>
2	c1	14	c4
3	<i>D</i>	15	<i>D</i>
4	0	16	0
5	<i>D</i>	17	<i>D</i>
6	c2	18	c5
7	<i>D</i>	19	<i>D</i>
8	0	20	1
9	<i>D</i>	21	<i>D</i>
10	c3	22	c6
11	<i>D</i>	23	<i>D</i>
12	1	24	1

В том же столбце таблицы символы с указывают на биты, которые используются для вычисления контрольного циклического избыточного кода (*Cyclic Redundancy Check, CRC*) для всех 4632 бит предыдущего расширенного суперкадра. Эти кадрирующие биты находятся перед 2, 6, 10, 14, 18 и 22 кадрами. С их помощью приемник может вычислить свой *CRC*-код и сравнить его с полученным значением *CRC*. Если эти значения не совпадают, значит, кадр содержит ошибочные биты. Такая проверка позволяет приемнику определить частоту ошибок, возникающих при передаче кадров, и качество канала *T1*. Фактически вычисление *CRC*-кода выполняют *ESF*-совместимые блоки *CSU*. А поскольку для этого используются только 6 кадрирующих битов, пропускная способность канала, предназначенного для отслеживания частоты ошибок, составляет 2000 бит/с.

Третьим типом кадрирующих битов формата *ESF* являются биты, используемые в качестве значения кадрирования для синхронизации принимаемых данных. Как следует из табл. 4.3, это кадрирующие биты, расположенные перед 4, 8, 12, 16, 20 и 24 кадрами. При их передаче генерируется значение 001011, и поскольку 6 бит составляют четвертую часть общего количества кадрирующих битов, можно сказать, что для передачи данных синхронизации применяется канал, имеющий пропускную способность 2 кбит/с (то есть под этот канал выделена четвертая часть канала *ESF*, пропускная способность которого равна 8 кбит/с).

4.4.5. Низкоскоростные мультиплексоры речь/данные. В результате объединения некоторых возможностей, предоставляемых обеими технологиями мультиплексирования (*TDM* и *STDM*), был создан новый тип мультиплексоров, предназначенных для передачи как данных, так и речевой информации, - низкоскоростные мультиплексоры речь/данные. Главной отличительной чертой таких мультиплексоров является их способность оцифровывать речь на скоростях, которые значительно меньше скорости, достигаемой при использовании РСМ (64 кбит/с). Поэтому, рассматривая данную тему, мы вначале сосредоточим внимание на нескольких методах, используемых этими устройствами для оцифровки речи.

За последние четыре десятилетия было разработано немало технологий оцифровки речи, которые можно разделить на три категории в зависимости от используемого метода кодирования: кодирование формы сигнала, вокодерное кодирование и смешанное (гибридное) кодирование. Мы рассмотрим наиболее популярные из этих технологий.

4.4.5.1. Кодирование формы сигнала. Этот метод предназначен для обработки только аналоговых сигналов. Применяется он с целью максимально точно передать информацию о форме сигнала. Значения напряжений сигнала, полученные в результате выборки, при кодировании преобразуются в дискретные значения, двоичные представления которых затем используются при модуляции сигнала. И хотя для передачи сигналов, полученных этим методом, требуется наибольшая ширина полосы пропускания, он обеспечивает самое высокое

качество передачи речи. Метод кодирования формы аналогового сигнала применяется в таких технологиях, позволяющих оцифровывать речь с качеством телефонной связи, как PCM, ADPCM и CVSD.

4.4.5.2. Вокодерное кодирование. При использовании метода вокодерного кодирования речи восстановление сигналов производится на основе анализа особенностей образования речи и ее восприятия человеком, однако форма восстановленного сигнала при этом может отличаться от формы исходного сигнала. Принципиальное отличие вокодерного кодирования от кодирования формы состоит в том, что по каналу связи передается не сам сигнал, а параметры модели его образования. Руководствуясь этими параметрами, приемник восстанавливает сигнал.

Обычно при использовании вокодерного кодирования оцифровка речи осуществляется на очень низких скоростях, к тому же в восстановленной речи появляются металлические нотки. Одной из наиболее популярных технологий, в которых применяется данный метод, является линейное предсказывающее кодирование, согласно которому выборка сегментов речи выполняется через каждые 20 мс и кодируются параметры, необходимые для ее восстановления. Такое кодирование применяется для передачи речевых сигналов со скоростью 2400 или 4800 бит/с.

В 70-80-е годы устройства, поддерживающие вокодерное кодирование речи, стали широко применяться в частных сетях. С их помощью и с использованием мультиплексирования по аналоговым каналам, пропускная способность которых равна 9600 бит/с, можно было одновременно передавать четыре телефонных разговора. Однако такие устройства имеют высокую стоимость.

4.4.5.3. Гибридное кодирование. Определив достоинства и недостатки описанных выше методов кодирования, разработчики взяли все лучшее и создали новый метод, получивший название гибридное кодирование. Устройства, выполняющие кодирование в соответствии с этим методом (далее мы будем называть их гибридными кодерами), вначале осуществляют выборку речевого сигнала, а затем анализируют ее, как устройства вокодерного кодирования речи. Но вместо того чтобы немедленно передать проанализированные параметры речевого сигнала, гибридные кодеры используют их для его восстановления, а затем сравнивают полученный сигнал с первоначальным, и если обнаруживаются существенные расхождения, то они ликвидируются путем подбора соответствующих параметров. В результате такого подхода удастся создать более достоверную модель речи, чем при использовании метода вокодерного кодирования речи. В методе гибридного кодирования удалось объединить лучшие черты двух предыдущих методов и добиться высококачественной передачи речи по низкоскоростным каналам.

Сейчас метод гибридного кодирования широко используется в цифровой мобильной связи. Возможно, наиболее популярная технология, основанная на данном методе, применяется в мобильной телефонной связи стандарта GSM (*GeNeRaL SYStemS FoR MobILe*).

Еще одной технологией по методу смешанного кодирования является линейное предсказание с кодовым возбуждением (*CoDeD EXcITeD LINeaR PRe-DICTIoN*, CELP), которая была недавно стандартизирована *ITU*. Данная технология обеспечивает оцифровку речи, передаваемой со скоростью от 5 до 16 кбит/с. В соответствии с этой технологией, была создана речевая кодовая книга, с помощью которой вместо целой серии параметров речи можно передать только номер записи, содержащий данные параметры. Благодаря использованию кодовой книги речь можно передавать с высоким качеством и по каналам, имеющим малую пропускную способность.

Гибридные кодеры были разработаны еще десять лет назад, однако широкое применение получили недавно, поскольку коммерчески выгодными стали только после создания недорогого цифрового процессора сигналов (*DIGITaL SIGNaL PRoceSSoR*, *DSP*), способного обрабатывать до 200 миллионов операций в секунду. На сегодняшний день различные гибридные кодеры, поддерживающие технологию CELP, встраиваются в мультиплексоры, устройства доступа к сети *FRaMe ReLaY* (*FRaMe ReLaY AcceSS DeViCeS*, *FRAD*), а также речевые шлюзы. Благодаря этому обеспечивается высококачественная передача оцифрованной речи по низкоскоростным каналам частных сетей, сетей *FRaMe ReLaY* и Интернета.

4.4.5.4. Адаптивная дифференциальная импульсно-кодовая модуляция. Адаптивная дифференциальная импульсно-кодовая модуляция (*ADaptIVe DIFFeReNtIaL PULSe-CoDe MoDULatIoN*, *ADPCM*) представляет собой технологию оцифровки и сжатия речи, в которой вместо кодирования значений выборки с помощью 8 бит, используемых в соответствии с *PCM*, применяются 4-битовые слова, каждое из которых служит для представления одного из 16 уровней оцифровки. В отличие от *PCM*, где 8 бит служат для определения амплитуды сигнала, в *ADPCM* 4-битовые слова предназначены для представления разности между значениями двух последовательных выборок. При использовании технологии *ADPCM* оцифрованную речь можно передавать со скоростью 32 кбит/с (для сравнения: в случае применения *PCM* данные должны передаваться со скоростью 64 кбит/с).

Создавая эту технологию, разработчики учли тот факт, что при выборке речевого сигнала с частотой 8000 раз в секунду амплитуды сигналов двух соседних выборок не очень сильно отличаются друг от друга. В устройства, поддерживающие технологию *ADPCM*, встраиваются микросхемы с адаптивным предсказывающим кодированием, которые, исходя из значения предыдущей выборки, пытаются определить значение следующей. Кроме того, благодаря использованию в предсказывающем блоке цепи обратной связи существенно уменьшается вероятность появления большой разности между предсказываемым и истинным значениями выборок. Анализ разности между этими значениями замеров показал, что для их представления вполне достаточно четырех битов. При использовании *ADPCM* уменьшается отношение сигнал/шум и снижается точность восстановления сигнала. Но когда технология *ADPCM* приме-

няется для оцифровки обычной речи, восстановленный сигнал очень трудно отличить от высококачественного сигнала, полученного при использовании *PCM*.

Технологии *ADPCM* и *PCM* различаются прежде всего характеристиками модемов, используемых для передачи оцифрованной речи. Технология *PCM* требует применения модемов, работающих со скоростями до 33,6 кбит/с, тогда как технология *ADPCM* благодаря использованию адаптивного предсказания поддерживает модемы, передающие данные со скоростью до 9,6 кбит/с. При скорости передачи выше 33,6 кбит/с сигналы изменяются слишком быстро, чтобы быть корректно предсказанными, поэтому уровень ошибок значительно возрастает. Вот почему *ADPCM* в основном используется в частных сетях, где каналы речевого диапазона могут быть отведены для передачи речи, оцифрованной с помощью *ADPCM*, а остальные каналы могут предназначаться для передачи данных.

Некоторые из используемых в настоящее время технологий *ADPCM* поддерживают передачу данных со скоростью 32 кбит/с, другие, стандартизированные версии этой технологии оцифровки речи, предназначены для работы со скоростями 40 и 16 кбит/с (причем при передаче со скоростью 40 кбит/с качество восстанавливаемого сигнала выше).

4.4.5.5. Оцифровка речи по технологии CVSE. Дельта-модуляция с переменной крутизной входного аналогового сигнала (*CoNtINUoUSLY VaRIabLe Slope Data, CVSD*) представляет собой низкоскоростную технологию оцифровки речи, которая первоначально разрабатывалась для применения военным ведомством, но заняла достойное место среди методов оцифровки речи и на коммерческом рынке.

В основе технологии *CVSD* лежит следующая закономерность: чем выше частота выборки сигнала, тем меньше разность между значениями амплитуд двух соседних выборок. То есть при достаточном увеличении частоты выборки можно добиться того, что для двоичного представления разности между значениями амплитуд двух соседних выборок может быть использован всего лишь один бит. Более того, с помощью одного бита указывается направление изменения крутизны входного аналогового сигнала. Если этот бит имеет значение 1, разность между значениями амплитуд соседних выборок (крутизна) увеличивается, а если этот бит имеет значение 0 – уменьшается. Как производится кодирование в данном случае, можно понять из рис. 4.19.

Технология *CVSD* поддерживает две частоты выборки, 32 000 и 16 000 раз в секунду, а следовательно, и две скорости передачи данных, 32 кбит/с и 16 кбит/с. Если при работе со скоростью 16 кбит/с для оцифровки речи использовать эту технологию, эффективность передачи будет в четыре раза выше, чем при оцифровке речи с помощью *PCM* (при той же скорости), и в два раза выше, чем в случае применения *ADPCM*. Однако поскольку в *CVSD* изменение крутизны указывается посредством только одного бита, для передачи речи, оцифрованной с помощью этой технологии, не рекомендуется использовать модемы, предназначенные для передачи данных.

Технология *CVSD*, равно как и технология *ADPCM*, предназначена в первую очередь для передачи оцифрованной речи с применением низкоскоростных мультиплексоров речь/данные. Подробнее об этом будет рассказано в следующем разделе.

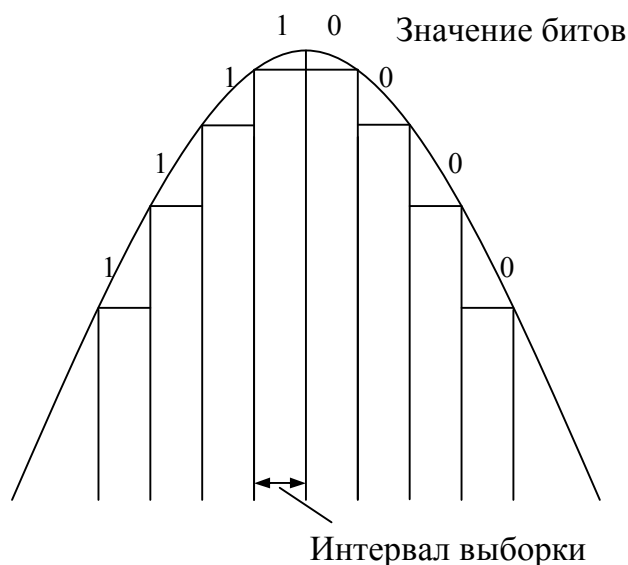


Рис. 4.19. Кодирование речи с помощью технологии *CVSO*

4.4.5.6. Использование мультиплексоров речь/данные. Как уже говорилось выше, низкоскоростные мультиплексоры речь/данные можно рассматривать как устройства, которые выполняют функции, присущие *TDM*- и *STDM*-мультиплексорам. Блок устройства, выполняющий функции *TDM*-мультиплексора, используется исключительно для мультиплексирования двух или более оцифрованных речевых сигналов и передачи их по высокоскоростному каналу. Хотя мы и ранее использовали термин «высокоскоростной канал», на самом деле речь шла всего лишь о скорости 56 или 64 кбит/с, что выглядит довольно скромно по сравнению с показателями *T1*-мультиплексора. Предположим, низкоскоростной мультиплексор речь/данные соединен с цифровой выделенной линией, обеспечивающей скорость передачи 64 кбит/с. Он принимает три речевых сигнала, и речь оцифровывается с помощью технологии *CVSD*. Суммарная скорость передачи оцифрованных речевых сигналов составит 48 кбит/с. Это означает, что блок статического мультиплексирования сможет передавать свои данные только со скоростью 16 кбит/с ($64 - 48$).

Рассмотрим рис. 4.20, на котором показано распределение каналов цифровой выделенной линии с пропускной способностью 64 кбит/с. Каждому речевому сигналу выделено по одному временному слоту, то есть можно сказать, что каждый мультиплексированный оцифрованный речевой сигнал передается по отдельному каналу со скоростью 16 кбит/с. Что же касается данных, то для них выделен всего лишь один временной слот, что равносильно суммарной скорости их передачи 16 кбит/с. Данная технология гарантирует, что каждый оцифрованный речевой сигнал будет передан независимо от статистически

мультиплексированных данных и что при этом не возникнут задержки, способные привести к некорректному восстановлению сигнала.

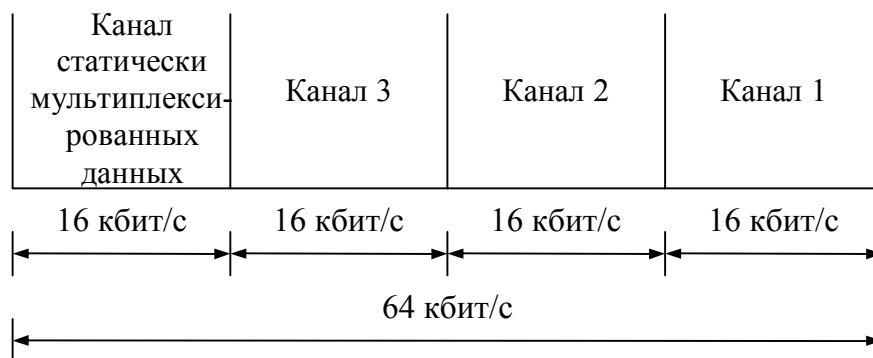


Рис. 4.20. Распределение каналов мультиплексора

Рассмотрим следующий пример использования низкоскоростного мультиплексора речь/данные. Предположим, что в центральном офисе и в филиале фирмы имеются частные АТС и что в среднем в течение рабочего дня постоянно используются три телефонных канала. Также предположим, что 10 низкоскоростных (2400 бит/с) терминалов, расположенных в филиале, время от времени обмениваются данными с компьютером, находящимся в центральном офисе.

Схема соединения устройств, соответствующая описанному выше сценарию, изображена на рис. 4.21. Здесь для обеспечения одновременной передачи речевой информации и данных по каналу с пропускной способностью 64 кбит/с, проложенному между центральным офисом фирмы и ее филиалом, используется пара низкоскоростных мультиплексоров речь/данные. Каждый из этих мультиплексоров имеет три встроенные адаптерные платы *CVSD*, на которые по трем телефонным линиям офисных АТС поступают аналоговые сигналы. После оцифровки для каждого из сигналов выделяется временной слот, в котором можно за одну секунду передать 16 кбит данных (см. рис. 4.20.). Для мультиплексирования данных, поступающих от десяти низкоскоростных терминалов, применяется технология *STDM*. Причем, как уже говорилось ранее, для их передачи выделен только один временной слот (то есть они пересылаются со скоростью 16 кбит/с). Если все десять терминалов одновременно начнут передавать файлы компьютеру центрального офиса, данные будут поступать на мультиплексор с суммарной скоростью 24 кбит/с, что превысит объем данных, который может быть передан с помощью одного временного слота. При возникновении такой ситуации мультиплексор инициирует функцию управления потоком, с помощью которой указывает одному или нескольким терминалам на необходимость приостановить передачу.

Что касается экономической целесообразности использования мультиплексора речь/данные, то при оценке его окупаемости следует принимать во внимание следующие факторы: расстояние между связываемыми сторонами, те-

кущий и ожидаемый трафики коммутируемой телефонной сети, количество подлежащих мультиплексированию речевых каналов и каналов данных, а также стоимость самого мультиплексора. Заметим, что в среднем мультиплексор окупает себя за два месяца.

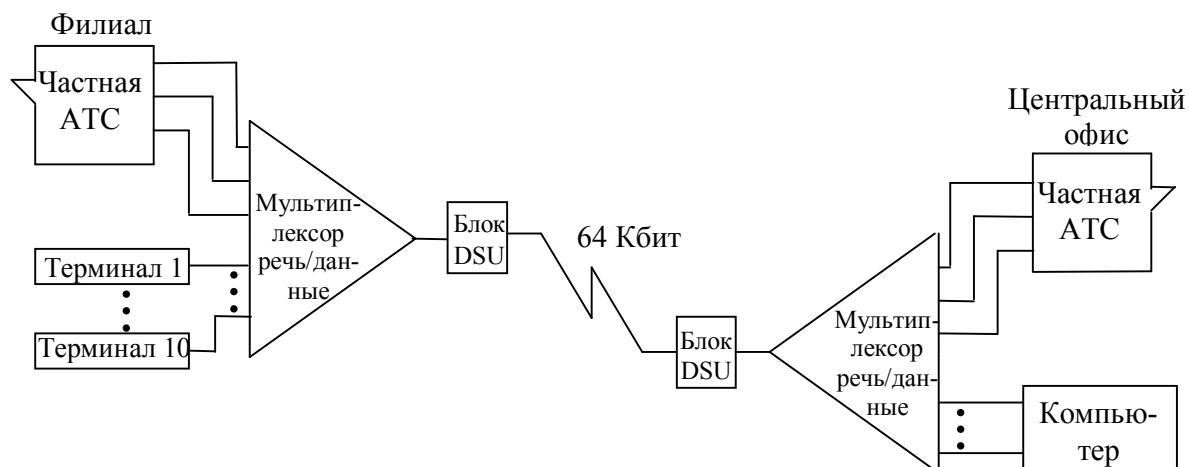
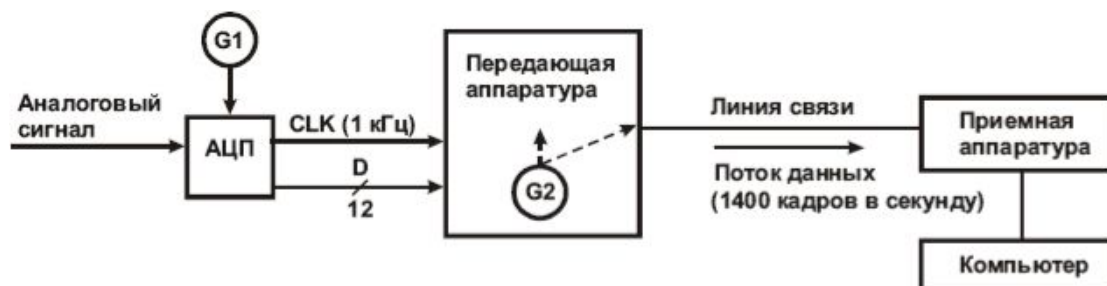


Рис.4.21. Использование низкоскоростных мультиплексоров речь/данные

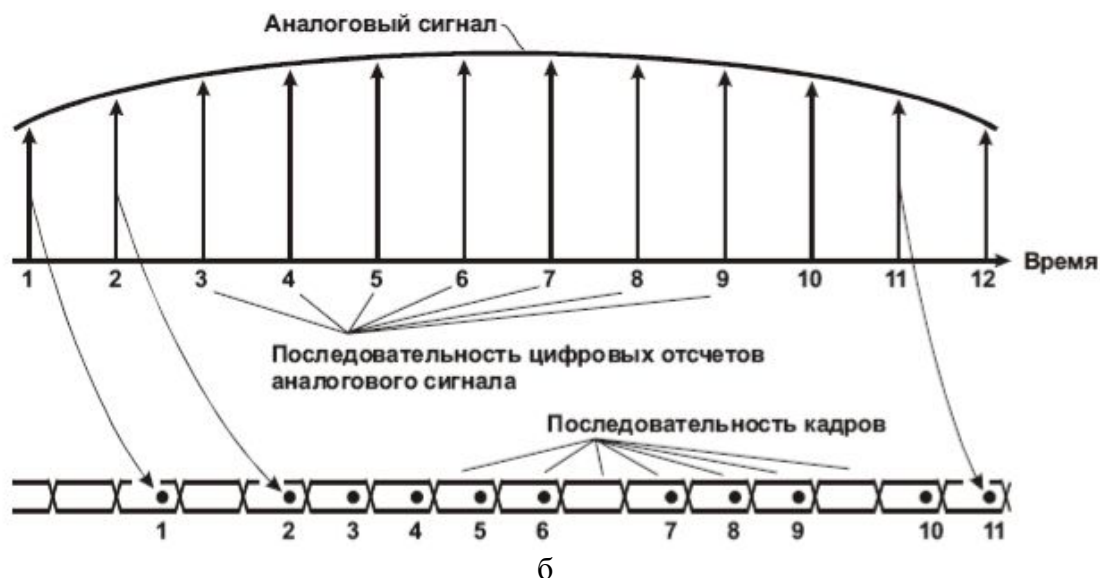
4.4.6. Способы размещения низкоскоростного потока данных в высокоскоростном потоке кадров. Для передачи данных от некоторого источника, например, от низкоскоростного аналого-цифрового преобразователя (АЦП), к удаленному регистрирующему устройству, например, компьютеру, может использоваться высокоскоростной канал связи (рис. 4.22). Канал содержит передающую и приемную аппаратуру (например, два мультиплексора), подключенную к противоположным сторонам линии связи. Разумеется, канал связи передает потоки данных и от других источников (на рисунке не показаны). Задача состоит в рациональном размещении низкоскоростного потока данных D в высокоскоростном потоке кадров, передаваемых по линии. Рассмотрим два способа решения этой задачи [29].

Первый способ. Как показано на рис. 4.22, АЦП формирует непрерывную последовательность 12-разрядных цифровых отсчетов D аналогового сигнала. Синхронизация АЦП осуществляется от генератора синхросигналов $G1$. Отсчеты следуют с частотой 1кГц и сопровождаются синхросигналом CLK .

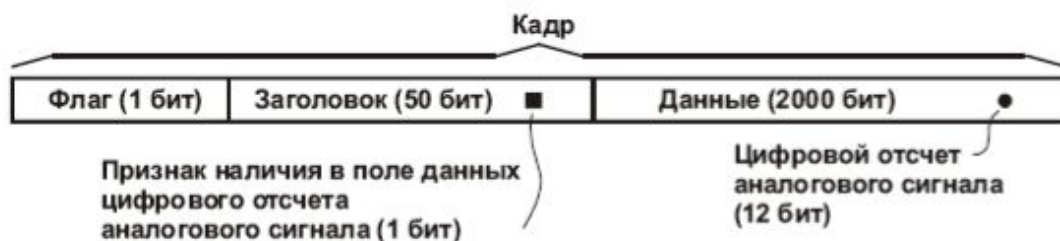
Передающая аппаратура синхронизируется сигналом от генератора $G2$. Данные поступают в линию связи в виде непрерывной последовательности кадров со скоростью 1400 кадров в секунду. Кадр содержит одноразрядный флаг, обозначающий его начало, 50-разрядный заголовок со служебными данными и поле данных (2000 бит).



а



б



в

Рис. 4.22. Передача низкоскоростного потока данных от аналого-цифрового преобразователя к удаленному компьютеру (вариант 1):

а - схема; б - размещение потока данных в потоке кадров; в - структура кадра

Передающая аппаратура принимает очередной отсчет D аналогового сигнала и размещает его в определенной группе разрядов поля данных ближайшего формируемого кадра. В каждый кадр может быть помещен только один отсчет. Факт его размещения помечается установкой в состояние лог. 1 определенного бита в заголовке кадра.

Как следует из рис. 4.22, б, отсчеты переносятся в поток кадров с нерегулярным темпом, по мере их поступления от АЦП и наличия «свободных мест» в транспортной системе.

Приемная аппаратура анализирует поступающие кадры. При обнаружении в заголовке кадра признака наличия цифрового отсчета аналогового сигнала соответствующий 12-разрядный код извлекается из поля данных и передается в компьютер для дальнейшей обработки. Если кадр не содержит данных от АЦП, то отведенное для них место используется для передачи данных от других источников.

Второй способ. Передача данных от АЦП в компьютер может синхронизироваться одним генератором G (рис. 4.23), при этом скорость поступления данных, как и в предыдущем примере (рис. 4.22, а), не совпадает со скоростью потока кадров. Однако в данном случае возможно регулярное размещение данных в потоке кадров. Это означает, что в каждой группе из M кадров размещаются N цифровых отсчетов аналогового сигнала. В примере, приведенном на рис. 4.23, поток разбивается на группы по девять кадров ($M = 9$), а в каждой группе размещаются четыре отсчета аналогового сигнала ($N = 4$).

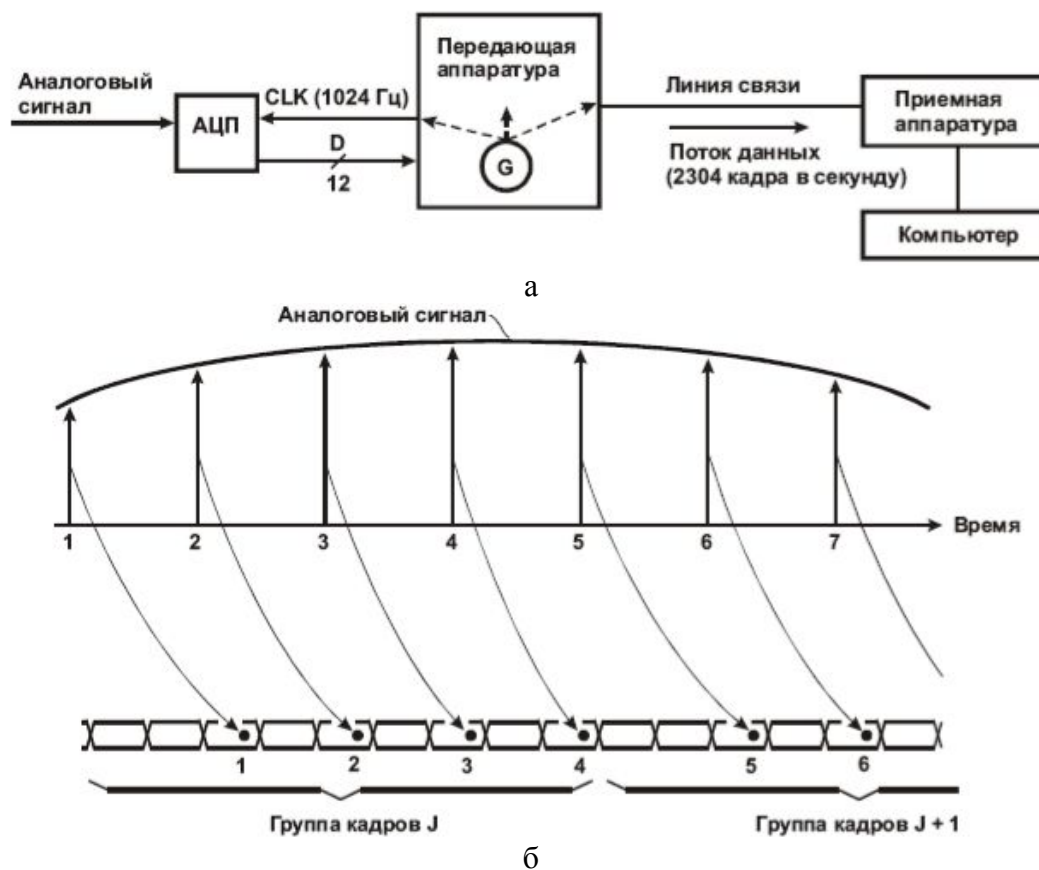


Рис. 4.23. Передача низкоскоростного потока данных от аналого-цифрового преобразователя к удаленному компьютеру (вариант 2):
а - схема; б - размещение потока данных

Убедимся в правильности такого разбиения. За одну секунду по линии связи передаются 2304 кадра или 256 групп по девять кадров. В каждой группе содержатся четыре отсчета, поэтому их общее число составит $256 \times 4 = 1024$, что соответствует скорости поступления данных от АЦП.

В общем случае при некрatном отношении частоты F_F следования кадров к частоте F_S следования цифровых отсчетов аналогового сигнала выполняется условие $F_F/F_S = A + b/c$, где $F_F/F_S > 1$, A – целая часть отношения частот, b/c – несократимая дробь. Это условие можно записать в следующем виде: $F_F/F_S = (Ac + b)/c$, где $Ac + b = M$, $c = N$.

Разметку потока кадров на группы можно выполнить по меньшей мере двумя способами.

1. Первый кадр каждой группы может помечаться установкой в единичное состояние некоторого служебного бита в его заголовке. Приемная аппаратура, обнаружив признак начала группы кадров, получает ориентир для поиска и выделения данных от АЦП. Это возможно благодаря тому, что размещение этих данных в каждой группе кадров одинаково и заранее известно. Так, в приведенном на рис. 4.23 примере первый и второй кадры каждой группы не содержат данных от АЦП, в третьем кадре данные имеются и т. д.

2. Для более экономной разметки потока кадров на группы можно использовать псевдослучайные одноразрядные флаги. Экономия заключается в том, что применение псевдослучайной последовательности флагов позволяет приемной аппаратуре находить начало группы кадров только на основе анализа этой последовательности. При этом нет необходимости помечать первый кадр группы служебным битом в заголовке.

Применительно к рассматриваемой системе передачи данных (рис. 4.23) следует генерировать псевдослучайную последовательность битов с периодом повторения, равным или кратным девяти. Точнее, последовательность не обязательно должна быть псевдослучайной в строгом смысле слова. В данном случае достаточно принять за основу классическую псевдослучайную последовательность с периодом повторения, равным $2^4 - 1 = 15$ бит и выделить из нее только девять следующих подряд элементов, отбросив шесть остальных. Один из вариантов формирования такой «усеченной» последовательности поясняется рис. 4.24.

После начальной установки в регистре RG некоторого ненулевого кода (цепи установки на рисунке не показаны) классический генератор псевдослучайной битовой последовательности (рис. 4.24, а) может находиться в 15 состояниях (см. таблицу в левой верхней части рисунка), которые периодически повторяются. Копии этих состояний можно получить при просмотре выходной последовательности битов через четырехразрядное движущееся окно, как показано в правой верхней части рисунка.

Чтобы исключить «лишние» состояния регистра RG , в схему генератора введены дополнительные элементы (рис. 2.24, б). Элемент И формирует сигнал лог. 1 при обнаружении в регистре кода $0011_2 = 3_{10}$. В начале следующего такта, по положительному фронту сигнала CLK триггер $D1$ устанавливается в единицу, что вызывает безусловный переход регистра в состояние $1111 = 15$. При переходе синхросигнала CLK в состояние лог. 0 устанавливается в единицу триггер $D2$, что вызывает установку в нуль триггера $D1$ и снятие сигнала установки кода 1111 с входа регистра. В начале следующего такта в регистре формируется

код 1110 = 14, а в середине этого такта сигнал лог. 0 переписывается из триггера $D1$ в триггер $D2$.

В последующих тактах регистр последовательно проходит состояния 12, 8, 1, 2, 4, 9, 3, затем описанный цикл генерации кодов повторяется. В результате число различных состояний регистра сокращено до девяти, что и требовалось. Отметим, что в данном случае последовательность состояний регистра не соответствует кодам, наблюдаемым при просмотре выходной последовательности битов через движущееся четырехразрядное окно. Однако это не приводит к каким-либо неопределенностям, так как эти коды можно однозначно сопоставить с состояниями регистра.

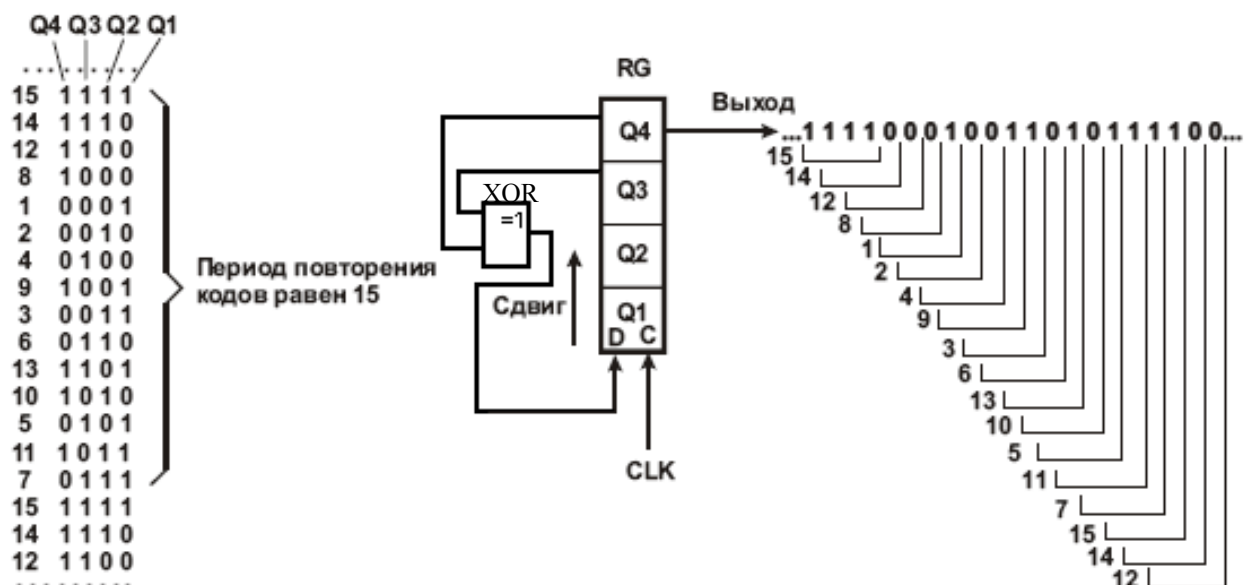
Последовательность генерируемых таким способом битов размещается передающей аппаратурой во флаговых позициях формируемых кадров - по одному биту на каждый кадр. Эта последовательность выделяется из кадров приемной аппаратурой и просматривается ею через четырехразрядное окно (см. рис. 4.24, б).

Согласно протоколу обмена, один из кодов в окне, например код 1111, является признаком начала группы из девяти кадров. Обнаружив такой код, приемная аппаратура «узнаёт», что последний принятый кадр является первым в группе (возможны и иные договоренности). Разумеется, что передающая аппаратура должна знать, что к этому моменту она действительно переслала именно первый кадр группы. Таким образом, осуществляется кадровая синхронизация между приемной и передающей аппаратурой.

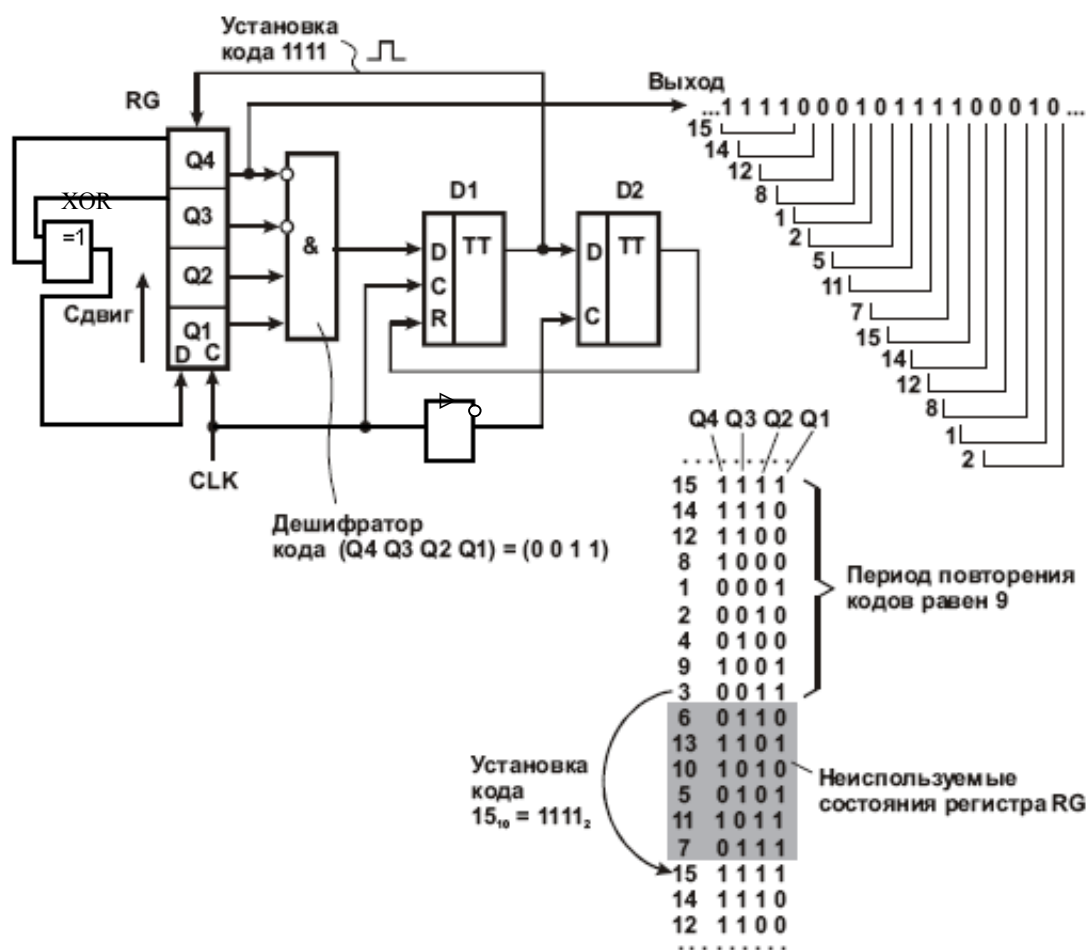
Рассмотренные решения позволяют согласовать темп поступления данных от низкоскоростного источника с темпом их передачи по высокоскоростной транспортной системе. Для такого согласования можно выделить один служебный бит в каждом кадре. Но можно и не вводить служебные биты, если использовать псевдослучайные флаги начала кадров.

4.4.7. Резюме.

- Мультиплексирование представляет собой процесс совместного использования средств передачи с применением деления частоты или времени.
- Замена большинства аналоговых коммуникационных каналов цифровыми привела к замене систем *FDM* системами *TDM*.
- Для корректного восстановления аналогового сигнала частота его выборки должна быть в два раза больше ширины его спектра.
- Повторители регенерируют цифровые импульсы, удаляя при этом любые искажения. Этим они отличаются от используемых в аналоговых каналах усилителей, которые усиливают не только сигнал, но и искажения.
- Кадрирование - это метод синхронизации групп каналов.



а



б

Рис. 4.24. Генераторы, формирующие а - псевдослучайную битовую последовательность максимальной длины с периодом повторения, равным 15;
б - битовую последовательность с периодом повторения, равным 9

- При использовании формата кадрирования *D4* принимающему устройству предоставляются только данные синхронизации, а при использовании формата *ESF* ему передаются также статистические данные, применяемые для мониторинга производительности.

- В соответствии с форматом *ESF*, вместе с каждым суперкадром передаются 24 кадрирующих бита, из которых 12 являются битами канала данных, 6 - битами канала, служащего для отслеживания частоты ошибок, а 6 - битами формирования значения синхронизации.

- Использование биполярного кодирования позволяет по одному каналу передавать одновременно и сигналы, и напряжение питания - для их разделения в каждой группе каналов должен быть установлен трансформатор.

- Для обеспечения возможности синхронизации в передаваемых данных должно содержаться некоторое минимальное количество двоичных единиц, или, другими словами, должна поддерживаться необходимая плотность единиц.

- Если передающий мультиплексор при сканировании собственного порта не обнаружит там активности, для корректного позиционирования данных. он должен внедрить в формируемый кадр символ *NULL*.

- В процессе эволюции новые технологии всегда вытесняют старые. Статистические мультиплексоры с разделением времени, *STDM*, почти повсеместно вытеснили *TDM*-мультиплексоры. Однако их ждет аналогичная судьба: в настоящее время многие организации заменяют *STDM* маршрутизаторами.

- Существуют три категории технологий оцифровки речи: кодирование формы сигнала, вокодерное кодирование и гибридное кодирование.

- При использовании технологий *ADPCM* и *CVSD* речь можно оцифровывать со скоростями, которые в два-четыре раза меньше скорости, требуемой в соответствии с технологией *PCM*.

- В мультиплексоре речь/данные технология *TDM* используется для мультиплексирования нескольких речевых каналов, а технология *STDM* – для мультиплексирования потоков данных, поступающих от нескольких источников.

4.5. Асинхронный режим передачи

В этом подразделе описана быстро развивающаяся сетевая технология *ATM*, предоставляющая возможность передавать данные, оцифрованную видео- и аудиоинформацию, а также изображения. *ATM* – это аббревиатура от *ASYNChRoNoUS TRaNSFeR MoDe*, что в переводе с английского означает «асинхронный режим передачи». Используя один формат данных, абоненты на обоих концах соединения могут посредством *ATM* обмениваться информацией любого типа. Рассмотрим эту технологию более подробно.

Технологию *ATM* можно считать универсальной, так как она разработана для передачи данных, аудио- и видеоинформации (включая графические изображения) как по локальным, так и по глобальным сетям. До появления *ATM* выбор пользователем типа сети определялся спецификой передаваемой информации. Например, сети с коммутацией каналов, которые включают телефонные

коммутируемые линии общественного пользования и в которых используются высокоскоростные средства передачи данных, применялись в основном для пересылки аудио- или видеоинформации, то есть информации, при передаче которой задержки не желательны. При помощи сетей с коммутацией пакетов, таких как *X.25* и *FRAMe ReLaY*, *E* пересылалась информация, допускающая некоторые задержки при передаче. Поэтому отдельные пользователи выбирали технологию передачи в зависимости от специфики информации, но большинство организаций были вынуждены устанавливать сети обоих типов. Это привело к снижению эффективности применения сетей и к увеличению стоимости их обслуживания. С появлением АТМ стало возможным использовать преимущества как первой, так и второй технологии передачи информации и пересылать данные, аудио- и видеоинформацию по одной сети.

В основе технологии АТМ лежит передача данных в виде ячеек фиксированной длины (53 байта) из одной локальной сети в другую с помощью глобальной сети. При этом могут использоваться и частные, и общественные глобальные сети с разными скоростями. В настоящее время при передаче данных по локальным сетям АТМ поддерживаются скорости 25 и 155 Мбит/с, а при использовании волоконно-оптических сетей (*SONET*) скорость передачи составляет 622 Гбит/с. Режим АТМ является столь универсальным благодаря тому, что данные при пересылке разбиваются на ячейки, длина которых остается постоянной – 53 байта – независимо от используемых средств, скорости передачи и способа кадрирования.

Применение ячеек фиксированной длины позволяет разрабатывать сравнительно недорогое оборудование, обеспечивающее коммутацию ячеек в соответствии с содержимым их заголовков. Поэтому режим АТМ считается универсальной и экономически очень выгодной технологией.

Хотя многие организации для пересылки аудиоинформации и данных с помощью мультиплексоров применяют один общий канал, такой способ передачи не получил широкого распространения. Например, при использовании маршрутизатора, подключенного к локальной сети, передаваемые данные должны поступать на порт высокоскоростного мультиплексора, соединенного с мультиплексором, который установлен на частной телефонной станции. Несмотря на то что общая глобальная сеть применяется для пересылки как речевой информации, так и данных, такой метод передачи данных не является оптимальным. Кроме того, в случае использования существующих локальных и глобальных сетей новые мультимедийные приложения, передающие видеоинформацию, могут оказаться неработоспособными, так как требуют большой пропускной способности канала при незначительных временных задержках. АТМ — это технология, которая обеспечивает необходимую пропускную способность при передаче аудио- и видеоинформации, а также цифровых данных. Сравнив возможности АТМ с характеристиками других технологий, вы сможете оценить все ее преимущества. Для этого достаточно обратиться к табл. 4.4.

Таблица 4.4.

Сравнительные характеристики сетей

Характеристика	Сеть передачи данных	Сеть связи	АТМ-сеть
Поддерживается обмен	Данными	Речевой информацией	Данными, речевой и видеоинформацией
Единица передаваемых данных	Пакет	Кадр	Ячейка
Длина единицы передаваемых данных	Изменяется	Фиксирована	Фиксирована
Тип коммутации	Пакетов	Каналов	Ячеек
Тип связи	Ориентированная или не ориентированная на соединение	Ориентированная на соединение	Ориентированная на соединение
Чувствительность к задержкам	Зависит от передаваемой информации	При передаче всех данных	Адаптивная
Среда и скорость передачи	Определяется протоколом	Определяется классом	Изменяется
Тип доступа к среде передачи	Выделенный или общий	Выделенный	Выделенный

В настоящее время применяются сети различных типов - от локальных (например, *EtherNet*, *Token-RING*) до глобальных (*X.25* и *FRaMe ReLaY*). Для каждого типа сетей характерно определенное оборудование. И даже если используемое оборудование во многом является схожим, в принципах функционирования таких сетей имеется много различий. Данные, как правило, пересылаются в виде пакетов переменной длины. Большинство протоколов передачи информации как по глобальным, так и по локальным сетям ориентировано на соединение, но существуют также протоколы, не ориентированные на соединение. То, насколько время задержки влияет на качество передачи, зависит от вида передаваемой информации. Если задержки при передаче не слишком продолжительны, то оказываемое ими отрицательное влияние мало ощущается пользователями. Но такие же задержки, например, при телефонном разговоре, могут значительно ухудшить его качество. Технология АТМ является адаптивной по отношению к передаваемым данным: величина задержки всегда зависит от типа информации.

Некоторые службы АТМ обеспечивают постоянную скорость передачи битов (*CoNStaNt BIT Rate*, *CBR*). При этом ячейки пересылаются строго по порядку, что гарантирует качество сервиса (*QUaLITY oF SeRVice*, *QoS*), необходимое для передачи информации в реальном масштабе времени. Для сравнения: в дру-

гих технологиях, например в IP, для обеспечения такого же качества сервиса приходится использовать множество протоколов.

Передача данных в АТМ основана на коммутации ячеек, имеющих длину 53 байта. В каждой ячейке 5 байт отводится для заголовка и 48 байт – для данных пользователя. Формат ячейки АТМ и ее структура представлены на рис. 4.25.

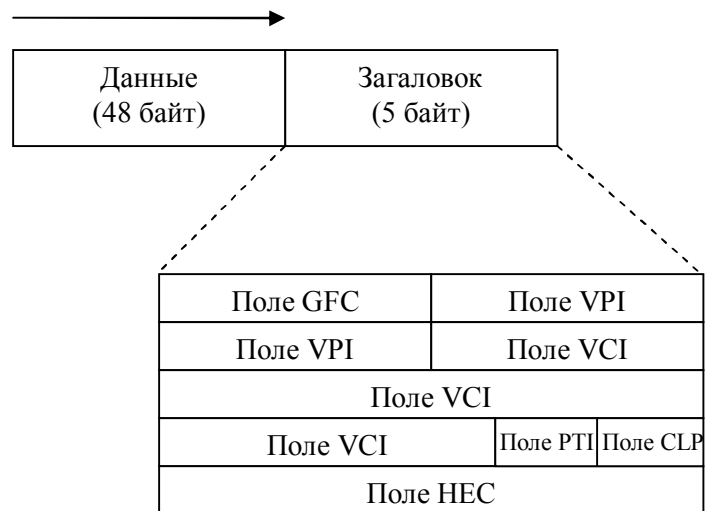


Рис. 4.25. Формат и структура ячейки АТМ

Поле *GFC* (*GeNeRiC FLoW CoNtRoL* – общее управление потоком) занимает первые 4 бита в первом байте заголовка и предназначено для управления потоком между АТМ-сетью и пользователем. В настоящее время механизм использования этого поля все еще дорабатывается. Режим АТМ поддерживает два основных типа интерфейса: интерфейс пользователь-сеть (*UNI*) и межсетевой интерфейс (*NNI*). При передаче ячейки от пользователя к сети или в обратном направлении в поле *GFC* содержится значение. Если же поток данных направлен за пределы сети либо передается между сетями, значение в этом поле отсутствует, а свободное место используется для увеличения длины поля *VPI*.

Поле *VPI* (*VIRtUaL Path IDeNtIFieR* – идентификатор виртуального маршрута) имеет длину 8 бит и является одной из двух частей идентификатора соединения, используемого в АТМ. В этом поле определяется виртуальный маршрут, представляющий собой набор каналов, по которым передается информация. При использовании интерфейса *UNI* размер поля *VPI* в ячейке составляет 8 бит, а при передаче потока за пределы сети (интерфейс *NNI*) он увеличивается до 12 бит. Ниже этот механизм будет описан более подробно.

Второй частью идентификатора соединения является поле *VCI* (*VIRtUaL ChaNNeL IDeNtIFieR* – идентификатор виртуального канала). Поле *VCI* имеет длину 16 бит и предназначено для идентификации соединения двух АТМ-станций, обменивающихся данными при помощи специального приложения.

Данные передаются по нескольким виртуальным каналам (*VC*), но по одному виртуальному маршруту. Например, один виртуальный канал можно применять для передачи резервных копий, а другой – для передачи с помощью протокола *TCP/IP*. Виртуальный канал предоставляет возможность передачи ячеек по одноподключенному маршруту. Во время проведения каждой из описанных выше операций другой набор виртуальных каналов устанавливается на втором конце соединения. Виртуальный канал можно рассматривать как сквозную одноподключенную линию связи, а виртуальный маршрут – как набор виртуальных каналов. При передаче данных за пределы сети поле *VPI* применяется для определения всего набора виртуальных каналов между коммутаторами, что позволяет АТМ-коммутаторам просто считывать значение поля *VPI*. Использование поля *VCI* будет описано ниже.

Поле *PTI* (*PaYLOAD TYpe IDeNtIFIeR* – идентификатор типа полезной нагрузки) имеет длину 3 бита и определяет тип информации, передаваемой ячейкой АТМ. Значение в поле *PTI* показывает, какая информация передается – управляющая или данные пользователя. В настоящее время это поле не выполняет никаких других задач, однако предусмотрена возможность расширения его функций.

Значение в поле *CLP* (*CeLL LoSS PRIoRITY* – приоритет ячейки) имеет длину 1 бит и определяет относительный приоритет данных, передаваемых в ячейке. Если в этом поле содержится значение 1, ячейка при перегрузке канала связи отбрасывается, если 0 – передается дальше.

В поле *HEC* (*HeaDeR ERRoR CoNtRoL* – контроль ошибок в заголовке), имеющее длину 8 бит, записывается результат контроля ошибок в заголовке, вычисленный при помощи избыточного циклического кода (*CRC*). Поле *HEC* позволяет обнаружить ошибки, появляющиеся в 40-битовом заголовке ячейки АТМ, в отдельных или в нескольких сразу разрядах.

В АТМ предусмотрено два типа сетевых интерфейсов. Интерфейс пользователь-сеть (*USeR-NetWoRk INteRFace, UNI*) описывает способ подключения пользователей к АТМ-сети. В свою очередь, междоузелный интерфейс (*NetWoRk-NetWoRk INteRFace, NNI*) определяет способ соединения двух сетей АТМ. Так как частные и общие АТМ-сети должны взаимодействовать между собой, спецификации *UNI* и *NNI* также были разделены на частные и общие.

Интерфейс *UNI* для общих сетей определяет способ подключения пользователя к коммутатору поставщика услуг АТМ. Аналогичным образом, интерфейс *UNI* для частных сетей задает способ подключения пользователя к частному АТМ-коммутатору. Интерфейс *UNI*, предназначенный для частных сетей и сетей общего пользования, показан на рис. 4.26.

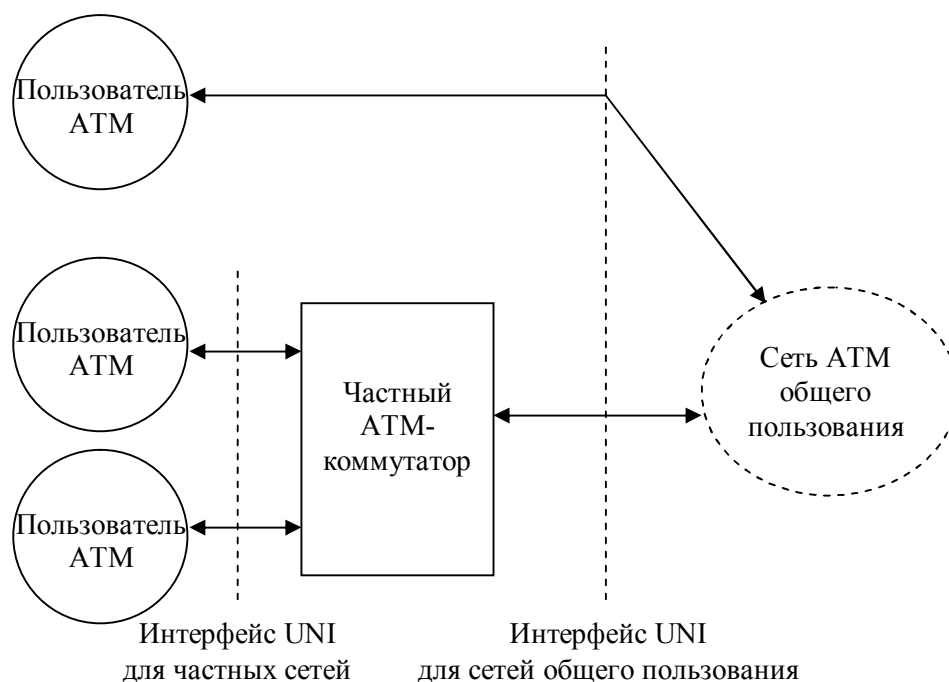


Рис. 4.26. Применение интерфейса *UNI*

Использование технологии коммутации ячеек в ЛВС дает некоторые преимущества перед использованием обычных технологий, применяемых в сетях *EtherNet*, *Token-RING* и *FDDI*. К их числу относятся, в частности, предоставление отдельным рабочим станциям доступа с полной пропускной способностью к ATM-коммутаторам и возможность устройств, подключенных к сети, работать с потоками данных, поступающих с разной скоростью. О преимуществах этой технологии можно судить по рис. 4.27, где изображен ATM-коммутатор, позволяющий обрабатывать потоки данных, которые поступают с тремя различными скоростями. В рассматриваемом случае рабочие станции, соединенные с коммутатором, способны обмениваться с ним данными со скоростью 25 Мбит/с, а локальный сервер — со скоростью 155 Мбит/с. Коммутатор может быть соединен с другими коммутаторами в локальной сети или с телефонной станцией.

В отличие от обычных пакетных технологий ЛВС (например, *EtherNet*, где кадр имеет длину до 1526 байт), ATM-технология, основанная на применении ячеек длиной 53 байта, позволяет заметно сократить задержки при передаче данных. Поскольку длина ячеек фиксирована, при передаче они не оказывают влияния друг на друга. Позиция ячейки в потоке определена в ее заголовке, поэтому для коммутации ячеек можно использовать аппаратные средства. В обычных же сетях функции коммутации и маршрутизации выполняются, как правило, программными или программно-аппаратными средствами, действующими намного медленнее.

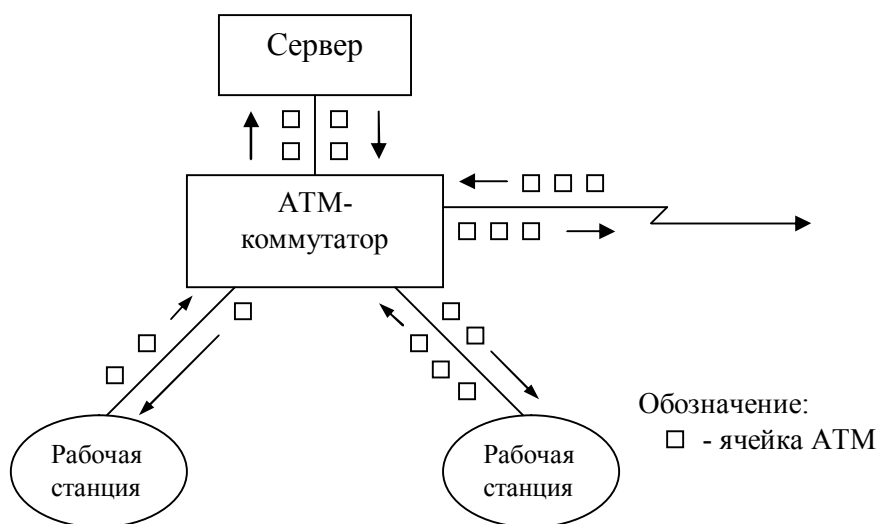


Рис. 4.27. Технология АТМ основана на коммутации 53-байтовых ячеек

Существуют еще две особенности АТМ-технологии, на которые следует обратить внимание: во-первых, данная технология работает в асинхронном режиме, а во-вторых, она является ориентированной на соединение. В процессе обмена данными ячейки АТМ при уплотнении потока смешиваются с ячейками, передаваемыми по другим адресам, и пересылаются в общем потоке. Однако мультиплексирование ячеек осуществляется при асинхронной передаче и лишь при условии, что ячейка содержит полезную информацию. При обычном же мультиплексировании с разделением времени синхронизирующий байт посылается даже в отсутствие информации, которую нужно передать. Ориентация технологии АТМ на соединение проявляется в том, что связь между станциями устанавливается до передачи данных. В процессе ее установления определяется маршрут передачи данных от АТМ-коммутаторов к конечным станциям, что позволяет использовать заголовок ячейки АТМ для передачи таковой в нужное место по сети АТМ.

4.6. Соотношение между скоростью передачи и шириной полосы

В первом приближении часто считают, что для одного двоичного символа в секунду (бит/с) достаточно полосы частот в 1 Гц. Исходя из этого предположения в телефонном канале с шириной 3 100 Гц можно передать не более 3 100 бит/с. В то же время известно, что по тем же каналам современные модемы успешно работают до скорости 56 кбит/с. Очевидно, что спектральная эффективность применяемых при этом технологий значительно выше, чем 1 бит/(с·Гц). Найквист теоретически доказал, что в полосе частот 1 Гц аналогового канала можно организовать безошибочную передачу данных со скоростью 2 бит/с (1.2). Этот результат часто называют скоростью или границей Найкви-

ста. Другими словами по телефонному каналу возможна безошибочная передача со скоростью 6200 бит/с. Утверждение Найквиста учитывало только эффект межсимвольной интерференции при передаче двоичных символов через фильтр нижних частот (ФНЧ). Если имеется ФНЧ с верхней частотой среза F_B (Гц), то согласно Найквисту без искажений возможна передача двоичных символов со скоростью, численно раной $2F_B$ (бит/с).

Клод Шеннон предложил другую модель для оценки возможностей передачи данных по каналу с белым шумом и ограниченной энергетикой. Классическая формула Шеннона выглядит следующим образом:

$$C = \Delta F \cdot \log_2(1 + P_c / P_{\text{ш}}), \quad (4.1)$$

где $P_c = E_b \cdot V$ – средняя мощность сигнала; E_b – энергия, затрачиваемая на передачу одного бита информации; V – скорость передачи информации; $P_{\text{ш}} = N_0 \Delta F$ – средняя мощность шума в полосе частот ΔF , N_0 – спектральная плотность мощности шума.

Соотношение сигнал/шум $P_c / P_{\text{ш}}$ в формуле присутствует в относительных единицах. Ограничившись отношением сигнал/шум, равным 30 дБ, для всё того же телефонного канала получим $V = 3100 \cdot \log_2(1 + 10^3) \approx 30$ кбит/с.

Следует заметить, что модель Шеннона не учитывает никаких других помех, кроме белого гауссовского (теплого) шума и не даёт конструктивных предложений по тем методам передачи, при помощи которых можно достичь таких скоростей.

Для достижения высоких значений спектральной эффективности (более чем 2 бит/(с·Гц)) широкое распространение получили многоуровневые (или многопозиционные) методы модуляции, такие как ФМ-4, КАМ-16 и другие, которые рассмотрены в разделе 1. Так при использовании ФМ-4 одна позиция сигнала или один сигнальный отсчёт переносит два бита (рис. 4.28).

Применение шестнадцатипозиционной квадратурной амплитудной модуляции КАМ-16 позволяет достичь спектральной эффективности 4 бит/(с·Гц) за счёт формирования большого цикла сигналов (сигнальных символов) и передачи их же с той же скоростью в той же полосе частот.

Скорость изменения позиций сигнала (сигнальных позиций или символов) носит название скорости модуляции или модуляционной скорости и измеряется опять же в Бодах. При этом скорость передачи данных отличается от скорости модуляции в число раз, количественно равное спектральной эффективности. Например, в случае протокола модуляции *V.29* (рис. 4.29) используется скорость модуляции 2 400 Бод и при этом достигается скорость передачи данных, равная 9 600 бит/с.

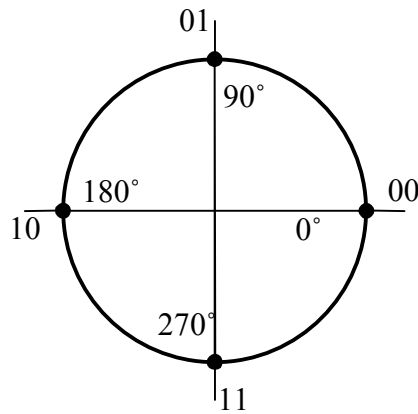


Рис. 4.28. Сигнальная диаграмма для модуляции методом ФМ-4

Ошибочно было бы думать, что достигнуть высокой спектральной эффективности можно только за счет усложнения формирующих схем многоуровневых модуляторов. Истинная цена за это состоит в необходимости повышения соотношения сигнал/шум для удержания того же значения вероятности ошибки на один передаваемый бит данных (что следует из формулы Шеннона).

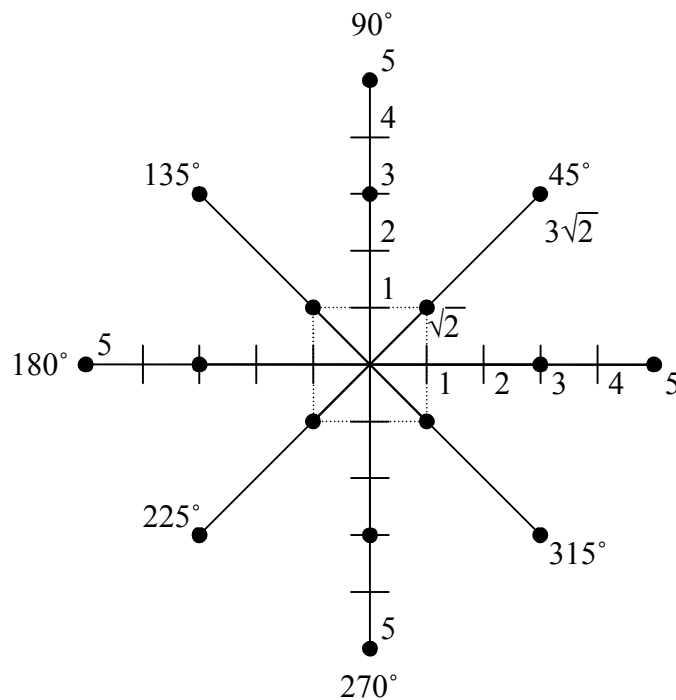


Рис. 4.29. Сигнальная диаграмма КАМ-16 согласно протокола V.29

Бод и бит/с (в контексте модуляции) количественно совпадают только в простейшем случае, когда один сигнал (сигнальный отсчет) переносит 1 бит данных, т.е. когда используются два сигнала или двухуровневая модуляция.

Соотношение скорости передачи и используемой полосы частот носит неоднозначный характер и в случае применения линейных методов кодирования (в цифровых модемах). Даже такой простой линейный код как *RZ* (*RetURN-to-*

ZeRo) использует значительно большее число переходов уровня сигнала (рис. 4.30), чем исходный информационный сигнал в соответствующем коде *NRZ* (*NoN-RetURN-to-ZeRo*). Для информационной последовательности примера, представленного на рис. 4.11 в коде *NRZ*, имеется всего 5 переходов, в то время как в *RZ* уже насчитывается 18 переходов уровня сигнала. Частота таких переходов непосредственно влияет на ширину спектра сигнала и соответствующие потребности в полосе частот среды передачи.

Обычно нет возможности расширить полосу частот используемой среды передачи, так как она определяется конкретными фазочастотными характеристиками, главными из которых являются амплитудные затухания и задержка группового времени прохождения. Тем не менее, существуют ограниченные возможности компенсации негативных эффектов путём применения всякого рода корректоров или эквалайзеров.

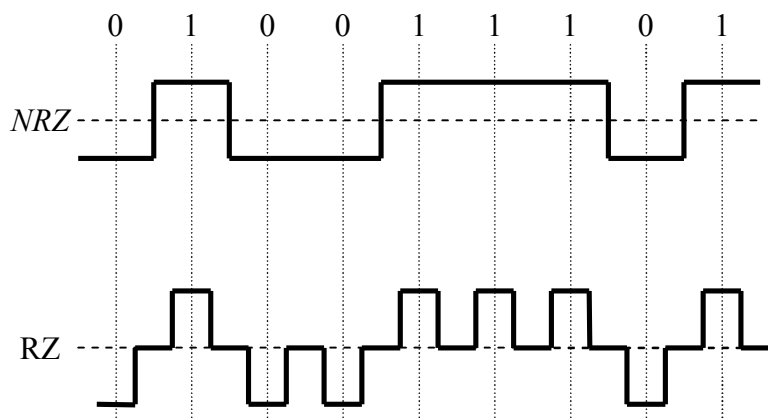


Рис. 4.30. Форма линейных сигналов в кодах *NRZ* и *RZ*

В настоящее время имеются три отдельные области телекоммуникаций, где очень актуальна задача повышения спектральной эффективности:

- телефонный канал, где доступная ширина полосы частот равна только 3100 Гц;
- абонентская линия – скорости доступа в *INteRNet*. Различные методы позволяют повысить скорость передачи данных в абонентской линии до 6 Мбит/с и более;
- радиорелейные СВЧ-линии. Дефицит радиочастотного диапазона вынуждает использовать многоуровневые методы модуляции, такие как КАМ-512, что позволяет достичь спектральной эффективности до 9 бит/(с·Гц).

Часто нужно разделять, понятия символьной и модуляционной скорости. Например, в системе передачи, изображенной на рис. 4.31, на выходе помехоустойчивого кодера скорость должна измеряться в кодовых символах в секунду, на выходе модулятора – в Бод. При этом каждый сигнальный символ переносит не только информационные, но и избыточные биты, добавленные ранее помехоустойчивым кодером. В таких случаях часто скорость выходной после-

довательности многоуровневого модулятора также выражается не в Бод, а в символах (уже сигнальных) в секунду. Так, например, принято в описании протокола модуляции *V.34*.



Рис. 4.31. Пример системы передачи, требующей для своего описания различные единицы скорости передачи

В СПД, работающих по технологии с расширением спектра (например *CDMA*), картина сложнее за счет того, что каждый бит передается в виде расширяющейся последовательности, достигающей длины $10^5 \dots 10^6$ двоичных символов. Такие двоичные символы называют часто элементарными символами, дискретами последовательности или по-английски – *chIp*.

5. МЕТОДЫ ПОВЫШЕНИЯ КАЧЕСТВА РАБОТЫ СИСТЕМ ПЕРЕДАЧИ ДИСКРЕТНОЙ ИНФОРМАЦИИ

5.1. Общие сведения

При создании систем передачи дискретной (цифровой) информации необходимо учитывать не только её назначение, условия работы, характер решаемых задач и требуемые показатели качества передаваемой информации, но и ряд других факторов. К их числу относятся неизбежные физические ограничения, связанные с типом используемого канала, шириной выделенного диапазона рабочих частот, техническими возможностями реализации данной системы, экономическими соображениями и т. д.

Полное решение задачи оптимизации системы передачи информации при ряде наложенных ограничений до сих пор не известно. Поэтому на практике обычно стремятся получить требуемые показатели только с учетом тех ограничений, которые наиболее существенны для рассматриваемой системы.

В настоящее время известны следующие пути повышения качества работы системы передачи цифровой информации:

- применение многопозиционных сигналов, эффективно использующих отведенную полосу частот (цифровые виды модуляции, рассмотренные в разделе 1);
- применение сложных сигналов, эффективно использующих излучаемую энергию (широкополосные сигналы – раздел 2);
- применение корректирующих кодов [14];

- применение перемежения;
- применение систем с обратной связью [15].

При создании систем часто комбинируют указанные выше методы, что позволяет получить высокие качественные показатели работы систем.

Учитывая, что ряд методов уже рассмотрены в предыдущих разделах, остановимся ниже на методе перемежения и некоторых алгоритмах передачи сообщений в системе с автоматическим запросом повторной передачи.

5.2. Методы перемежения

Изменение по определенному правилу естественного порядка следования символов в некоторой кодовой последовательности называют процедуру **перемежением** (*INteRLeaVING*), обратную перемежению, принято называть **деперемежением** (*DeINteRLeaVING*). В результате выполнения процедуры деперемежения восстанавливается естественный порядок следования символов.

Методы перемежения–деперемежения обычно используются для разрушения пакетов ошибок, вызванных замираниями уровня принимаемого сигнала, и, следовательно, уменьшения степени группирования ошибок в последовательности символов, поступающих на вход канального декодера. При перемежении передаваемое кодовое слово формируется из символов различных кодовых слов. Поэтому при деперемежении возникающий пакет ошибок разбивается на отдельные ошибки, принадлежащие различным кодовым словам. Иначе говоря, при деперемежении пакет ошибок трансформируется в последовательность независимых ошибок, для исправления которых, как правило, можно использовать менее мощный код. С увеличением глубины перемежения можно ожидать улучшения характеристик помехоустойчивости, поскольку при этом происходит ослабление корреляции ошибок. Но при этом возрастает задержка в доставке сообщения, связанная с выполнением процедур перемежения–деперемежения. Поэтому приходится принимать компромиссное решение между степенью улучшения характеристик помехоустойчивости и возможной задержкой.

Рассмотрим некоторые эффективные методы перемежения.

5.2.1. Блочное перемежение. При блочном перемежении кодовые слова длиной n символов записываются в виде таблицы шириной W и глубиной D символов, как показано на рис. 5.1.

Предположим, что $W = N$. Тогда строки таблицы представляют собой кодовые слова, содержащие k информационных символов и $(N-k)$ проверочных символов. После заполнения таблицы осуществляется последовательное считывание символов по столбцам и их передача по каналу связи. В приемнике выполняется обратная процедура – последовательная запись символов по столбцам до полного заполнения таблицы. Затем производится считывание символов по строкам таблицы и их декодирование. Такой перемежитель позволяет разрушить пакет ошибок длиной $I < D$, в результате чего в каждом кодовом слове будет не более одной ошибки.



Рис. 5.1. Блоковый перемежитель

Однако периодическая последовательность одиночных ошибок, отстоящих друг от друга на D символов, будет вызывать полное поражение ошибками некоторого одного слова. Задержка при выполнении процедур перемежения-деперемежения равна $2WD$ символов. Объем памяти и перемежителя и деперемежителя составляет WD символов.

Другой возможный вариант выполнения перемежителя изображен на рис. 5.2. Здесь информационные символы последовательно записываются по столбцам. Причем проверочные символы формируются из k информационных символов, разнесенных друг от друга в исходной последовательности на D символов. Считывание символов также осуществляется по столбцам. Преимуществом этого метода является передача информационных символов в естественном порядке следования и отсутствие задержки в перемежителе. Общая задержка составляет WD символов и обусловлена выполнением процедуры деперемежения. Параметры D и W перемежителя должны выбираться с таким расчетом, чтобы наиболее вероятные значения длины пакетов ошибок оказались меньше D .

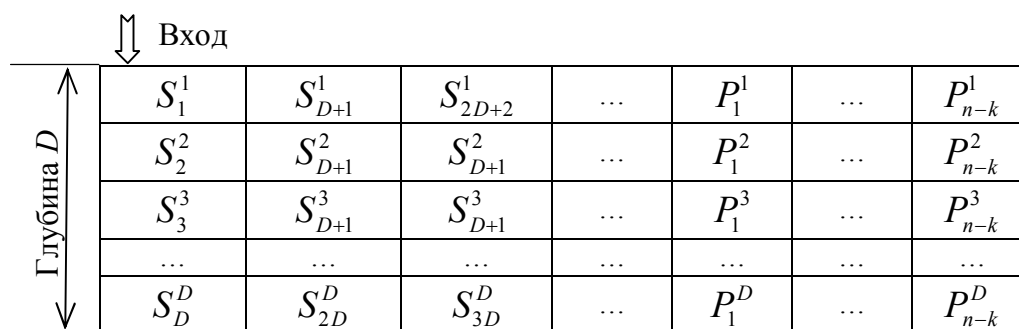


Рис. 5.2. Модифицированный блочный перемежитель

Однако этот тип перемежителя не обладает устойчивостью по отношению к периодической последовательности одиночных ошибок, разнесенных на D символов. В этой ситуации все символы в строке оказываются ошибочными, и каналный декодер переполняется.

5.2.2. Межблоковое перемежение. При межблоковом перемежении в качестве входного блока принимается блок из NB символов, и каждый блок из N символов распределяется между следующими B выходными блоками. Пусть x и y представляют собой соответственно входной и выходной символы перемежителя. Тогда правило отображения m -го символа I -го входного блока в $(J+Bt)$ -й символ $(I+J)$ -го выходного блока можно определить следующим образом:

$$y(i+j, j+Bt) = x(i, m)$$

для всех I и при $j = m \bmod B, t = m \bmod N$.

Пример межблокового перемежения при $B=3$ и $N=2$ показан на рис. 5.3.



Рис. 5.3. Пример межблокового перемежения при $B=3$ и $N=2$

Здесь символы I -го, $(I+1)$ -го и $(I+2)$ -го входных кодовых блоков обозначены соответственно a , b , c . Согласно приведённому правилу отображения

$$y(i+j, j+3t) = x(i, m)$$

для всех I и при $j = m \bmod 3, t = m \bmod 2$.

При

$$m = 0 \text{ имеем } y(i, 0) = x(i, 0),$$

$$m = 1 \text{ имеем } y(i+1, 4) = x(i, 1),$$

$$m = 2 \text{ имеем } y(i+2, 2) = x(i, 2) \text{ и т. д.}$$

Отметим, что последовательно следующие символы I -го входного блока отображаются в символах очередных в выходных блоках с нерегулярным смещением позиций $(J+Bt)$ в каждом блоке. Такое нерегулярное смещение позволяет рандомизировать влияние периодической помехи, действующей в канале

связи. Для однозначного отображения символов необходимо, чтобы B и N не имели общего делителя. Это ограничивает свободу выбора длины блока из BN символов. Недостатки такого метода перемежения состоят в том, что здесь выходные символы распределяются в пределах B блоков и общая задержка составляет B^2N символов (BN символов из-за необходимости запоминания $(B-1)BN$ входных блоков для выполнения процедуры распределения символов).

5.2.3. Свёрточное перемежение. Структурная схема сверточного перемежителя-деперемежителя приведена на рис. 5.4. Предполагается, что имеется синхронизация мультиплексоров и демультимплексоров передатчика и приёмника.

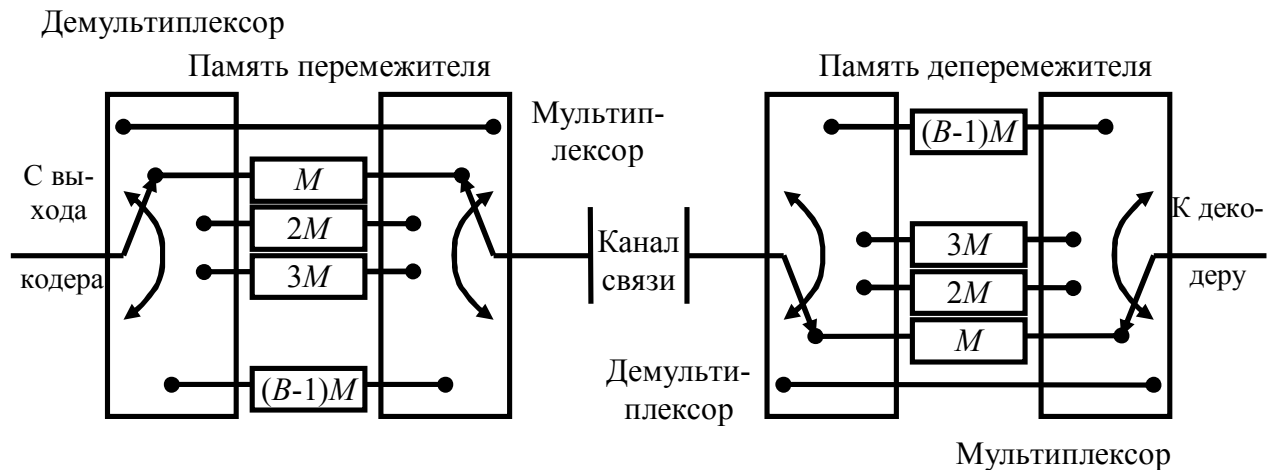


Рис. 5.4. Структурная схема свёрточного перемежителя - деперемежителя

Демультимплексор осуществляет последовательное подключение выхода кодера к различным строкам памяти перемежителя. Мультиплексор соответственно подключает вход декодера к различным строкам памяти деперемежителя. Каждая строка памяти представляет собой регистр сдвига, количество элементов задержки которого указано соответствующим числом, вписанным в прямоугольник. Первый элемент кодированной последовательности записывается в верхнюю строку и сразу же передается по каналу связи. Записывается он также в первую строку памяти деперемежителя, обеспечивающей задержку на $(B-1)M$ символов. Второй элемент кодированной последовательности записывается во вторую строку памяти перемежителя, обеспечивающей задержку на M символов. Таким образом, смежные символы кодированной последовательности оказываются разнесенными на M символов. Поэтому на них не оказывают влияние пакеты ошибок, длина которых не превышает M . При приеме второй символ дополнительно задерживается на $(B-2)M$ символов, так что общая задержка символов составляет $(B-1)M$ символов. Следует отметить, что все символы кодовой последовательности после перемежения и деперемежения имеют одинаковую задержку, поэтому порядок следования символов на выходе кодера и входе декодера сохраняется одним и тем же.

5.3. Автоматический запрос повторной передачи

В некоторых системах связи между передатчиком и приемником имеется двусторонняя линия связи, которую можно использовать для уведомления передающей стороны о наличии ошибок в принятом сообщении и для запроса повторной передачи искаженного сообщения.

Системы с исправлением ошибок, в которых принятые сообщения, содержащие ошибки, многократно повторяются передающей стороной по запросам приемной стороны, принято называть **системами с автоматическим запросом повторной передачи (АЗПП)**.

В системах с АЗПП на приемной стороне, как правило, выполняется только процедура обнаружения ошибок в принятом сообщении и не предпринимается никаких попыток их исправить. Поэтому в передаваемое сообщение вводится определенное количество избыточных битов или бит проверки на четность, образующих вместе с битами сообщения код с обнаружением ошибок. С увеличением количества избыточных битов возрастает обнаруживающая способность кода. Так как кодирование с обнаружением ошибок является простым, эффективным и достаточно устойчивым методом выявления искаженных сообщений, то системы с АЗПП позволяют простыми средствами обеспечить надежную передачу сообщений от источника к потребителю по каналам связи с различными характеристиками. Поэтому АЗПП широко используется в системах передачи данных, в которых предъявляются повышенные требования к верности, например в компьютерной связи.

В соответствии с семиуровневой моделью взаимодействия открытых систем контроль ошибок относится к числу основных операций, которые могут выполняться на нескольких уровнях, в частности на транспортном уровне (4-й уровень) и уровне линий передачи данных (2-й уровень).

На уровне линий передачи данных пакеты сообщений, поступающие с выхода сетевого уровня (3-й уровень), перед передачей их в канал связи преобразуются в блоки (фреймы) путем добавления вспомогательных битов, образующих поля управления и проверки ошибок.

На рис. 5.5 приведена примерная структура блока.

Длительность блока зависит от используемого протокола в линии передачи данных и может изменяться в пределах от 50 до 200 байт.



Рис. 5.5. Примерная структура блока

В общем случае система с АЗПП работает следующим образом. Принятые блоки, содержащие ошибки, повторно передаются до тех пор, пока они будут

приняты без ошибок или ошибки в них не будут обнаружены, При этом обнаружение ошибок производится с помощью простого кода, обнаруживающего ошибки. Приемная сторона по надежному обратному каналу передающей стороне подтверждает: положительные (ПП) или отрицательные (ОП) для указания отсутствия или наличия ошибок в ранее переданном блоке. При получении положительного подтверждения передающая сторона передает очередной блок, а при получении отрицательного – повторяет передачу ранее переданного блока. В принципе, повторная передача одного и того же сообщения может происходить до тех пор, пока оно не будет принято без ошибок.

Из различных вариантов АЗПП можно выделить три базовых: АЗПП с остановом и ожиданием, АЗПП с возвращением на N блоков, АЗПП с селективным повторением.

5.3.1. Автоматический запрос повторной передачи с остановом и ожиданием. При автоматическом запросе повторной передачи с остановом и ожиданием работа происходит следующим образом. Передающая и приемная стороны обмениваются сообщениями по полудуплексной линии связи. После передачи одного блока передающая сторона ожидает подтверждения со стороны приемной для того, чтобы передать очередной блок или повторить прежний. При получении блока приемная сторона передает в сторону передающей ПП, если принятый блок не содержит ошибок, и ОП, если принятый блок содержит ошибки.

На рис. 5.6 изображена временная диаграмма, иллюстрирующая процедуру передачи сообщений в системе с АЗПП с остановом и ожиданием.

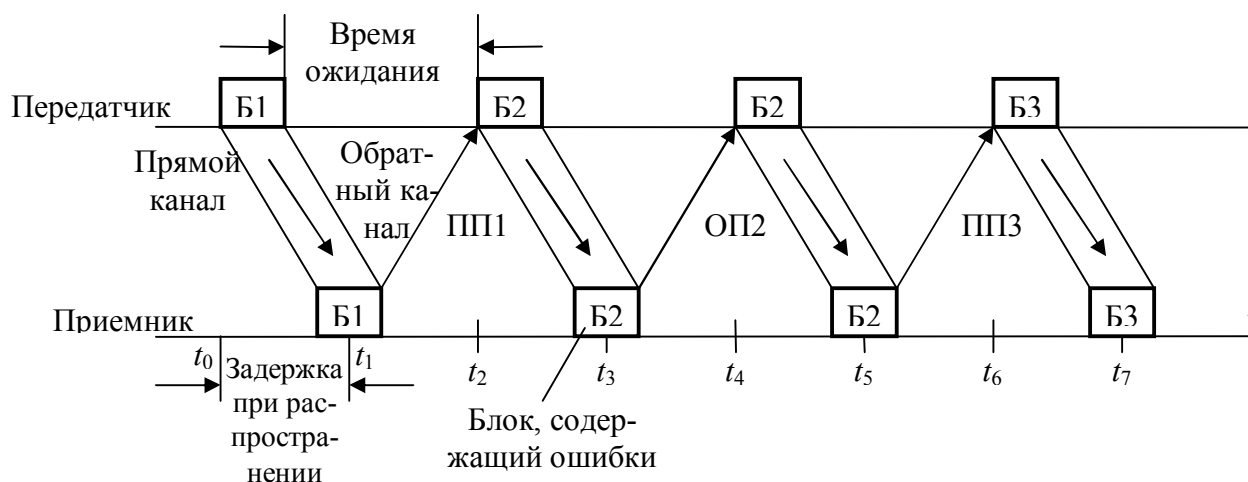


Рис. 5.6. Диаграмма передачи сообщений в системе АЗПП с остановом и ожиданием

Блоки сообщений Б1, Б2, Б3, ... подлежат передаче. В момент времени t_0 передатчик передает блок Б1 и ожидает подтверждения со стороны приемника о качестве приема этого блока. Если в момент времени t_1 , блок Б1 был принят без ошибок, то посылается ПП1. Получив это подтверждение, передатчик в мо-

Передатчик передаст последовательность блоков с номерами Б0, Б1, Б2,...,Б6. Как показано на рис. 5.7, в приемнике обнаруживается блок, содержащий ошибки (Б3), посылается ОПЗ передатчику и отбрасываются этот и все последующие блоки Б4, Б5, Б6, которые уже были переданы передатчиком до получения ОПЗ из-за задержки при распространении. Получив ОПЗ, передатчик повторно передает блоки Б3, Б4, Б5, Б6 и далее последующие блоки Б0,Б1,Б2,... (здесь нумерация блоков производится по $\text{mod } 2^k$). При обнаружении ошибок в любом последующем блоке эта процедура повторяется, начиная с блока, содержащего ошибки.

АЗПП с возвращением на N блоков является более эффективным по сравнению с АЗПП с остановом и ожиданием, поскольку здесь исключаются потери времени на ожидание подтверждений. Потери эффективности здесь связаны с отбрасыванием всех блоков, следующих за ошибочным блоком, даже если эти блоки не содержат ошибок.

5.3.3. Автоматический запрос повторной передачи с селективным повторением. Чтобы исключить потери эффективности, связанные с необходимостью повторной передачи безошибочных блоков, можно использовать процедуру селективного повторения. Здесь, как и в предыдущем варианте, необходима полнодуплексная линия связи.

На рис. 5.8 изображена временная диаграмма, иллюстрирующая процедуру передачи сообщений в системе с АЗПП и селективным повторением. Передатчик передает последовательность блоков Б1, Б2, Б3.... В приемнике обнаруживаются ошибки в блоке Б3 и посылается ОПЗ передатчику. Однако до его получения блоки Б4, Б5 и Б6 оказываются уже переданными. При получении ОПЗ в передатчике прекращается передача и вместо текущего блока Б7 повторно передается блок Б3, а затем передача блоков возобновляется в естественном порядке.

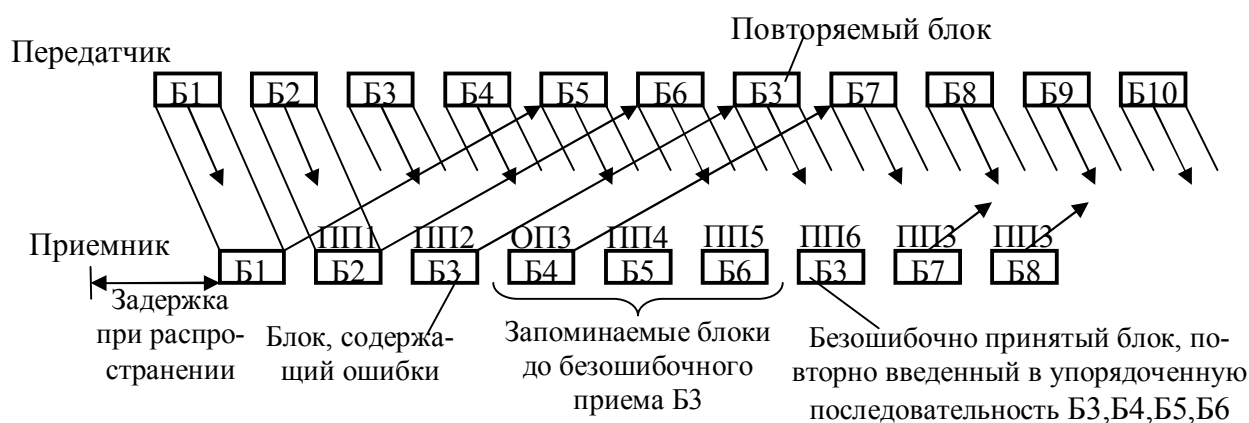


Рис. 5.8. Диаграмма передачи сообщений в системе с селективным повторением

Последовательность принятых блоков Б4, Б5, Б6, не содержащих ошибок, запоминается в приемнике до тех пор, пока не будет безошибочно принят блок Б3 и введен в соответствующее место для восстановления естественного порядка следования блоков в последовательности Б3, Б4, Б5, Б6,....

Если блок Б3 будет безошибочно принят после первой повторной передачи, то запоминаются блоки Б4, Б5 и Б6. Однако, если в блоке Б3 будут обнаружены ошибки при повторной передаче, то возникает необходимость повторить его передачу еще раз. В этом случае надо запоминать блоки Б4, Б5, Б6, Б7, Б8, Б9. Требование многократных повторных передач приводит к увеличению объема буферного запоминающего устройства. Поскольку теоретически повторные передачи могут быть неограниченными, то при конечном объеме буферного запоминающего устройства возможно его переполнение. В практически используемых системах с АЗПП и селективным повторением для предотвращения этого события предусматривается специальный механизм прерываний.

Характеристики систем с АЗПП принято оценивать двумя величинами: относительной пропускной способностью и вероятностью необнаруженной ошибки в битах данных. Относительная пропускная способность определяется как отношение среднего числа битов сообщения, доставляемых получателю в единицу времени, к среднему числу битов, передаваемых в системе в единицу времени. Таким образом, значение относительной пропускной способности всегда меньше 100 %. Например, при использовании кода, обнаруживающего ошибки, имеющего кодовую скорость $R_k = 0,98$, относительная пропускная способность при безошибочной передаче должна составлять 98%. Очевидно, что повторные передачи будут снижать значение относительной пропускной способности.

6. СЖАТИЕ СООБЩЕНИЙ

6.1. Типы систем сжатия

Передача и хранение информации требуют достаточно больших затрат. И чем с большим количеством информации нам приходится иметь дело, тем дороже это стоит. К сожалению, большая часть данных, которые нужно передавать по каналам связи и сохранять, имеет не самое компактное представление. Скорее, эти данные хранятся в форме, обеспечивающей их наиболее простое использование, например: обычные книжные тексты, *ASCII* коды текстовых редакторов, двоичные коды данных ЭВМ, отдельные отсчеты сигналов в системах сбора данных и т. д. Однако такое наиболее простое в использовании представление данных требует вдвое - втрое, а иногда и в сотни раз больше места для их сохранения и полосу частот для их передачи, чем на самом деле нужно. Поэтому сжатие данных — это одно из наиболее актуальных направлений современной телемеханики.

Таким образом, цель сжатия данных - обеспечить компактное представление данных, вырабатываемых источником, для их более экономного сохранения и передачи по каналам связи.

Ниже приведена условная структура системы сжатия данных:

Данные источника → Кодер → Сжатые данные → Декодер → Восстановленные данные

В этой схеме вырабатываемые источником данные определим как *данные источника*, а их компактное представление – как *сжатые данные*. Система сжатия данных состоит из *кодера* и *декодера источника*. Кодер преобразует данные источника в сжатые данные, а декодер предназначен для восстановления данных источника из сжатых данных. *Восстановленные* данные, вырабатываемые декодером, могут либо абсолютно точно совпадать с исходными *данными источника*, либо незначительно отличаться от них.

Существуют два типа систем сжатия данных:

- системы сжатия без потерь информации (неразрушающее сжатие);
- системы сжатия с потерями информации (разрушающее сжатие).

В системах сжатия без потерь декодер восстанавливает данные источника абсолютно точно, таким образом, структура системы сжатия выглядит следующим образом:

Вектор данных $X \rightarrow$ Кодер $\rightarrow B(X) \rightarrow$ Декодер $\rightarrow X$

Вектор данных источника X , подлежащих сжатию, представляет собой последовательность $X = (X_1, X_2, \dots, X_N)$ конечной длины. Отсчеты X_i - составляющие вектора X - выбраны из конечного алфавита данных A . При этом размер вектора данных N ограничен, но он может быть сколь угодно большим. Таким образом, источник на своем выходе формирует в качестве данных X последовательность длиной N из алфавита A .

Выход кодера – сжатые данные, соответствующие входному вектору X – представим в виде двоичной последовательности $B(X) = (b_1, b_2, \dots, b_k)$, размер которой k зависит от X . Назовем $B(X)$ кодовым словом, присвоенным вектору X кодером (или кодовым словом, в которое вектор X преобразован кодером). Поскольку система сжатия – неразрушающая, одинаковым векторам $X_L = X_m$ должны соответствовать одинаковые кодовые слова $B(X_L) = B(X_m)$.

При решении задачи сжатия естественным является вопрос, насколько эффективна та или иная система сжатия. Поскольку, как мы уже отмечали, в основном используется только двоичное кодирование, то такой мерой может служить коэффициент сжатия R , определяемый как отношение

$$r = \frac{\text{размер данных источника в битах}}{\text{размер сжатых данных в битах}} \quad (6.1)$$

Таким образом, коэффициент сжатия $R = 2$ означает, что объем сжатых данных составляет половину от объема данных источника. Чем больше коэффициент сжатия R , тем лучше работает система сжатия данных.

Наряду с коэффициентом сжатия R эффективность системы сжатия может быть охарактеризована *скоростью сжатия* R , определяемой как отношение

$$R = k/N \quad (6.2)$$

и измеряемой в количестве кодовых бит, приходящихся на отсчет данных источника». Система, имеющая *большой* коэффициент сжатия, обеспечивает *меньшую* скорость сжатия.

В системе сжатия с потерями (разрушением) кодирование производится таким образом, что декодер не в состоянии восстановить данные источника в первоначальном виде. Структурная схема системы сжатия с разрушением выглядит следующим образом:

$$X \rightarrow \text{Квантователь} \rightarrow X^Q \rightarrow \text{Неразрушающий кодер} \rightarrow B(X^Q) \rightarrow \text{Декодер} \rightarrow X^*$$

Как и в предыдущей схеме, $X = (X_1, X_2, \dots, X_N)$ - вектор данных, подлежащих сжатию. Восстановленный вектор обозначим как $X^* = (X_1, X_2, \dots, X_N)$. Отметим наличие в этой схеме сжатия элемента, который отсутствовал при неразрушающем сжатии, - *квантователя*.

Квантователь применительно к вектору входных данных X формирует вектор X^Q , достаточно близкий к X в смысле среднеквадратического расстояния. Работа квантователя основана на понижении размера алфавита (простейший квантователь производит округление данных до ближайшего целого числа).

Далее кодер подвергает неразрушающему сжатию вектор квантованных данных X^Q таким образом, что обеспечивается однозначное соответствие между X^Q и $B(X^Q)$ (для $X_L^Q = X_m^Q$ выполняется условие $B(X_L^Q) = B(X_m^Q)$). Однако система в целом остается разрушающей, поскольку двум различным векторам X может соответствовать один и тот же вектор X^* .

Разрушающий кодер характеризуется двумя параметрами - скоростью сжатия R и величиной искажений D , определяемых как

$$R = k/n, \quad D = (1/n) \sum (x_i - x_i^*)^2. \quad (6.3)$$

Параметр R характеризует скорость сжатия в битах на один отсчет источника, величина D является мерой среднеквадратического различия между X^* и X .

Если имеются система разрушающего сжатия со скоростью и искажениями R_1 и D_1 соответственно и вторая система со скоростью R_2 и искажениями D_2 , то первая из них лучше, если $R_1 < R_2$ и $D_1 < D_2$. Однако, к сожалению, невозможно построить систему разрушающего сжатия, обеспечивающую одновременно снижение скорости R и уменьшение искажений D , поскольку эти два параметра свя-

заны обратной зависимостью. Поэтому целью оптимизации системы сжатия с потерями может быть либо минимизация скорости при заданной величине искажений, либо получение наименьших искажений при заданной скорости сжатия.

Выбор системы неразрушающего или разрушающего сжатия зависит от типа данных, подлежащих сжатию. При сжатии текстовых данных, компьютерных программ, документов, чертежей и т.п. совершенно очевидно, что нужно применять неразрушающие методы, поскольку необходимо абсолютно точное восстановление исходной информации после ее сжатия. При сжатии речи, музыкальных данных и изображений, наоборот, чаще используется разрушающее сжатие, поскольку при практически незаметных искажениях оно обеспечивает на порядок, а иногда и на два меньшую скорость R . В общем случае разрушающее сжатие обеспечивает, как правило, существенно более высокие коэффициенты сжатия, нежели неразрушающее.

Ниже приведены ряд примеров, иллюстрирующих необходимость процедуры сжатия.

Пример 6.1. Предположим, что источник генерирует цифровое изображение (кадр) размером 512×512 элементов, содержащее 256 цветов. Каждый цвет представляет собой число из множества $\{0, 1, 2, \dots, 255\}$. Математически это изображение представляет собой матрицу 512×512 , каждый элемент которой принадлежит множеству $\{0, 1, 2, \dots, 255\}$. (Элементы изображения называют пикселями).

В свою очередь, каждый пиксел из множества $\{0, 1, 2, \dots, 255\}$ может быть представлен в двоичной форме с использованием 8 бит. Таким образом, размер данных источника в битах составит $8 \times 512 \times 512 = 2^{21}$, или 2,1 Мегабита.

На жесткий диск объемом в 1 Гигабайт поместится примерно 5000 кадров изображения, если они не подвергаются сжатию (видеоролик длительностью примерно в пять минут). Если же это изображение подвергнуть сжатию с коэффициентом $R = 10$, то на этом же диске мы сможем сохранить уже почти часовой видеофильм!

Предположим далее, что мы хотим передать исходное изображение по телефонной линии, пропускная способность которой составляет 14000 бит/с. На это придется затратить $21000000 \text{ бит} / 14000 \text{ бит/с}$, или примерно 3 минуты. При сжатии же данных с коэффициентом $R = 40$ на это уйдет всего 5 секунд!

Пример 6.2. В качестве данных источника, подлежащих сжатию, выберем фрагмент изображения размером 4×4 элемента и содержащее 4 цвета: R = «красный», O = «оранжевый», Y = «синий», G = «зеленый»:

R	R	O	Y
R	O	O	Y
O	O	Y	G
Y	Y	Y	G

Просканируем это изображение по строкам и каждому из цветов присвоим соответствующую интенсивность, например, $R = 3$, $O = 2$, $Y = 1$ и $G = 0$, в результате чего получим вектор данных $X = (3, 3, 2, 1, 3, 2, 2, 1, 2, 2, 1, 0, 1, 1, 1, 0)$.

Для сжатия данных возьмем кодер, использующий следующую таблицу перекодирования данных источника в кодовые слова (вопрос о выборе таблицы оставим на будущее):

Кодер	
Отсчет	Кодовое слово
3	001
2	01
1	1
0	000

Используя таблицу кодирования, заменим каждый элемент вектора X соответствующей кодовой последовательностью из таблицы (так называемое *кодирование без памяти*). Сжатые данные (кодовое слово $B(X)$) будут выглядеть следующим образом:

$$B(X) = (0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0).$$

Коэффициент сжатия при этом составит $R = 32/31$, или 1,03. Соответственно скорость сжатия $R = 31/16$ бит на отсчет.

6.2. Основные алгоритмы сжатия без потерь информации

Сжатие осуществляется либо на прикладном уровне с помощью программы сжатия, либо с помощью устройств защиты от ошибок непосредственно в составе модемов.

Основными методами сжатия являются: вероятностные, статические, арифметические, словарей и кодирование повторов.

К методам сжатия также относятся методы разностного кодирования, поскольку разности амплитуд представляется меньшим числом разрядов. Разностное кодирование реализовано в методах дельта-модуляции и её разновидностях.

Кодирование повторов (*RUN LeNGth ENcoDING, RLE*) применяется в основном для сжатия растровых изображений (графических файлов). Один из вариантов метода *RLE* предусматривает замену последовательности повторяющихся символов на строку, содержащую этот символ, и число, соответствующее количеству его повторений. Применение метода кодирования повторов для сжатия текстовых файлов оказывается неэффективным. Поэтому в современных системах передачи кодированной цифробуквенной информации алгоритм *RLE* используется мало.

Вероятностные методы сжатия используют кодовые слова переменной длины. В основе вероятностных методов сжатия (алгоритмов Шеннона-Фано

и Хаффмена) лежит идея построения «дерева», на «ветвях» которого положение символа определяется частотой его появления. Каждому символу присваивается код, длина которого обратно пропорциональна частоте появления этого символа. Существуют две разновидности вероятностных методов, различающихся способом определения вероятности появления каждого символа:

- статические методы, использующие фиксированную таблицу частоты появления символов, рассчитываемую перед началом процесса сжатия,
- динамические или адаптивные методы, в которых частота появления символов все время меняется и по мере считывания нового блока данных происходит перерасчет начальных значений частот.

Статические методы имеют значительное быстродействие и не требуют большой оперативной памяти. Они нашли широкое применение в многочисленных программах-архиваторах, например *ARC*, *PKZIP* и др., но для сжатия передаваемых модемами данных используются редко – предпочтение отдается арифметическому кодированию и методу словарей, обеспечивающим большую степень сжатия.

Арифметические методы. При арифметическом кодировании строка символов заменяется действительным числом больше нуля и меньше единицы. Арифметическое кодирование позволяет обеспечить высокую степень сжатия, особенно в случаях, когда сжимаются данные, где частота появления различных символов сильно варьируется. Однако сама процедура арифметического кодирования требует мощных вычислительных ресурсов, так как активно использует нецелочисленную арифметику, и до недавнего времени этот метод мало применялся при сжатии передаваемых данных. Лишь появление мощных процессоров, особенно с *RISC*-архитектурой, позволило создать эффективные устройства арифметического сжатия данных.

Метод словарей. Алгоритм для метода словарей описан в работах Зива и Лемпеля, которые впервые опубликовали его в 1977 г. В последующем алгоритм был назван *Lempel-ZIV*, или сокращенно *LZ*. На сегодня *LZ*-алгоритм и его модификации получили наиболее широкое распространение по сравнению с другими методами сжатия. В его основе лежит идея замены наиболее часто встречающихся последовательностей символов (строк) в передаваемом потоке ссылками на «образцы», хранящиеся в специально создаваемой таблице (словаре).

6.2.1. Вероятностные методы сжатия. Согласно методу Шеннона-Фано для каждого символа формируется битовый код, причем символы с различными частотами появления имеют коды различной длины [16]. Чем меньше частота появления символов в файле, тем больше размер его битового кода. Соответственно, чаще появляющийся символ имеет меньший размер кода.

Код строится следующим образом. Все символы, встречающиеся в файле, выписывают в таблицу в порядке убывания частоты их появления. Затем их разделяют на две группы так, чтобы в каждой из них были примерно равные суммы частот символов. Первые биты кодов всех символов одной половины устанавливаются в 0, а второй – в 1. После этого каждую группу делят еще раз пополам и так до тех пор, пока в каждой группе не останется по одному симво-

лу. Допустим, файл состоит из некоторой символьной строки *aaaaaaaaabbbbbbbccccccDDDDDeeeeFFF*, тогда каждый символ этой строки можно закодировать так, как показано в табл. 6.1.

Можно видеть, что если раньше каждый символ кодировался 8 битами, то теперь требуется максимум три бита.

Однако способ Шеннона-Фано не всегда приводит к построению однозначного кода. Более удачен в данном отношении метод Хаффмена, позволяющий однозначно построить код с наименьшей средней длиной, приходящейся на символ.

Однако способ Шеннона-Фано не всегда приводит к построению однозначного кода. Более удачен в данном отношении метод Хаффмена, позволяющий однозначно построить код с наименьшей средней длиной, приходящейся на символ.

Суть метода Хаффмена сводится к следующему. Символы, встречающиеся в файле, выписываются в столбец в порядке убывания вероятностей (частоты) их появления. Два последних символа объединяются в один с суммарной вероятностью. Из полученной новой вероятности m вероятностей новых символов, не использованных в объединении, формируется новый столбец в порядке убывания вероятностей, а две последние вновь объединяются. Это продолжается до тех пор, пока не останется одна вероятность, равная сумме вероятностей всех символов, встречающихся в файле.

Таблица 6.1

Пример построения кода Шеннона – Фано

Символ	Частота появления	Код
<i>a</i>	10	11
<i>b</i>	8	10
<i>c</i>	6	011
<i>D</i>	5	010
<i>e</i>	4	001
<i>F</i>	3	000

Процесс кодирования по методу Хаффмена поясняется табл. 6.2. Для составления кода, соответствующего данному символу, необходимо проследить путь перехода знака по строкам и столбцам таблицы кода.

Недостатки метода Хаффмена. Самой большой сложностью с кодами Хаффмена, как следует из предыдущего обсуждения, является необходимость иметь таблицы вероятностей для каждого типа сжимаемых данных. Это не представляет проблемы, если известно, что сжимается английский или русский текст; мы просто предоставляем кодеру и декодеру подходящее для английского или русского текста кодовое дерево. В общем же случае, когда вероятность символов для входных данных неизвестна, статические коды Хаффмена работают неэффективно.

Процесс кодирования по методу Хаффмена

Сим-вол	Частость появления								Кодовое слово
c	22	22	22	22	26	32	42	1	01
e	20	20	20	20	22	26	32	1	00
h	16	16	16	16	20	22	26	0	111
I	16	16	16	16	16	20	26	0	110
a	10	10	10	16	16	20	26	0	100
k	10	10	10	10	16	20	26	0	1011
m	4	1	6	10	16	20	26	0	10101
b	2	0	0	10	16	20	26	0	10100

Решением этой проблемы является статистический анализ кодируемых данных, выполняемый в ходе первого прохода по данным, и составление на его основе кодового дерева. Собственно кодирование при этом выполняется вторым проходом.

Существует, правда, динамическая версия сжатия Хаффмена, которая может строить дерево Хаффмена «на лету» во время чтения и активного сжатия. Дерево постоянно обновляется, чтобы отражать изменения вероятностей входных данных. Однако и она на практике обладает серьезными ограничениями и недостатками и, кроме того, обеспечивает меньшую эффективность сжатия.

Еще один недостаток кодов Хаффмена – это то, что минимальная длина кодового слова для них не может быть меньше единицы, тогда как энтропия сообщения вполне может составлять и 0,1, и 0,01 бит/букву. В этом случае код Хаффмена становится существенно избыточным. Проблема решается применением алгоритма к блокам символов, но тогда усложняется процедура кодирования/декодирования и значительно расширяется кодовое дерево, которое нужно в конечном итоге сохранять вместе с кодом.

Наконец, код Хаффмена обеспечивает среднюю длину кода, совпадающую с энтропией, только в том случае, когда вероятности символов источника являются целыми отрицательными степенями двойки: $1/2 = 0,5$; $1/4 = 0,25$; $1/8 = 0,125$; $1/16 = 0,0625$ и т. д. На практике же такая ситуация встречается очень редко или может быть создана блокированием символов со всеми вытекающими отсюда последствиями.

6.2.2. Арифметическое кодирование. При арифметическом кодировании, в отличие от рассмотренных нами методов, когда кодируемый символ (или группа символов) заменяется соответствующим им кодом, результат кодирования всего сообщения представляется одним или парой вещественных чисел в интервале от 0 до 1. По мере кодирования исходного текста отображающий его интервал уменьшается, а количество десятичных (или двоичных) разрядов,

служащих для его представления, возрастает. Очередные символы входного текста сокращают величину интервала исходя из значений их вероятностей, определяемых моделью. Более вероятные символы делают это в меньшей степени, чем менее вероятные, и, следовательно, добавляют меньше разрядов к результату.

Поясним идею арифметического кодирования на простейшем примере. Пусть нам нужно закодировать следующую текстовую строку: *РАДИОВИЗИР*.

Перед началом работы кодера соответствующий кодируемому тексту исходный интервал составляет $[0; 1)$.

Алфавит кодируемого сообщения содержит следующие символы (буквы): $\{ P, A, Д, И, О, В, З \}$.

Определим количество (встречаемость, вероятность) каждого из символов алфавита в сообщении и назначим каждому из них интервал, пропорциональный его вероятности. С учетом того, что в кодируемом слове всего 10 букв, получим табл. 6.3.

Располагать символы в таблице можно в любом порядке: по мере их появления в тексте, в алфавитном или по возрастанию вероятностей – это совершенно не принципиально. Результат кодирования при этом будет разным, но эффект – одинаковым.

Итак, перед началом кодирования исходный интервал составляет $[0 - 1)$.

Таблица 6.3.

Вероятности появления символов

Символ	Вероятность	Интервал
<i>А</i>	0,1	0 – 0,1
<i>Д</i>	0,1	0,1 – 0,2
<i>В</i>	0,1	0,2 – 0,3
<i>И</i>	0,3	0,3 – 0,6
<i>З</i>	0,1	0,6 – 0,7
<i>О</i>	0,1	0,7 – 0,8
<i>Р</i>	0,2	0,8 – 1

После просмотра первого символа сообщения *Р* кодер сужает исходный интервал до нового - $[0,8; 1)$, который модель выделяет этому символу. Таким образом, после кодирования первой буквы результат кодирования будет находиться в интервале чисел $[0,8- 1)$.

Следующим символом сообщения, поступающим в кодер, будет буква *А*. Если бы эта буква была первой в кодируемом сообщении, ей был бы отведен интервал $[0 - 0,1)$, но она следует за *Р* и поэтому кодируется новым *подынтервалом внутри уже выделенного для первой буквы*, сужая его до величины $[0,80 - 0,82)$. Другими словами, интервал $[0 - 0,1)$, выделенный для буквы *А*, располагается теперь внутри интервала, занимаемого предыдущим символом

(начало и конец нового интервала определяются путем прибавления к началу предыдущего интервала произведения ширины предыдущего интервала на значения интервала, отведенные текущему символу). В результате получим новый рабочий интервал $[0,80 - 0,82)$, так как предыдущий интервал имел ширину в 0,2 единицы и одна десятая от него есть 0,02.

Следующему символу *Д* соответствует выделенный интервал $[0,1 - 0,2)$, что применительно к уже имеющемуся рабочему интервалу $[0,80 - 0,82)$ сужает его до величины $[0,802 - 0,804)$.

Следующим символом, поступающим на вход кодера, будет буква *И* с выделенным для нее фиксированным интервалом $[0,3 - 0,6)$. Применительно к уже имеющемуся рабочему интервалу получим $[0,8026 - 0,8032)$.

Продолжая в том же духе, имеем:

вначале		$[0,0 - 1,0)$
после просмотра	<i>Р</i>	$[0,8 - 1,0)$
	<i>А</i>	$[0,80 - 0,82)$
	<i>Д</i>	$[0,802 - 0,804)$
	<i>И</i>	$[0,8026 - 0,8032)$
	<i>О</i>	$[0,80302 - 0,80308)$
	<i>В</i>	$[0,803032 - 0,803038)$
	<i>И</i>	$[0,8030338 - 0,8030356)$
	<i>З</i>	$[0,80303488 - 0,80303506)$
	<i>И</i>	$[0,803034934 - 0,803034988)$
	<i>Р</i>	$[0,8030349772 - 0,8030349880)$

Результат кодирования: интервал $[0,8030349772 - 0,8030349880]$. На самом деле, для однозначного декодирования теперь достаточно знать только одну границу интервала – нижнюю или верхнюю, т. е. результатом кодирования может служить начало конечного интервала – 0,8030349772. Если быть еще более точным, то любое число, заключенное *внутри* этого интервала, однозначно декодируется в исходное сообщение. К примеру, это можно проверить с числом 0,80303498, удовлетворяющим этим условиям. При этом последнее число имеет меньшее число десятичных разрядов, чем числа, соответствующие нижней и верхней границам интервала, и, следовательно, может быть представлено меньшим числом двоичных разрядов.

Нетрудно убедиться в том, что, чем шире конечный интервал, тем меньшим числом десятичных (и, следовательно, двоичных) разрядов он может быть представлен. Ширина же интервала зависит от распределения вероятностей кодируемых символов – более вероятные символы сужают интервал в меньшей степени и, следовательно, добавляют к результату кодирования меньше бит. Покажем это на простом примере.

Допустим, нам нужно закодировать следующую строку символов: *А А А А А А А А А А #*, где вероятность буквы *А* составляет 0,9. Процедура кодирования этой строки и получаемый результат будут выглядеть в этом случае следующим образом:

Входной символ	Нижняя граница	Верхняя граница
	0,0	1,0
<i>A</i>	0,0	0,9
<i>A</i>	0,0	0,81
<i>A</i>	0,0	0,729
<i>A</i>	0,0	0,6561
<i>A</i>	0,0	0,59049
<i>A</i>	0,0	0,531441
<i>A</i>	0,0	0,4782969
<i>A</i>	0,0	0,43046721
<i>A</i>	0,0	0,387420489
#	0,3486784401	0,387420489

Результатом кодирования теперь может быть, к примеру, число 0,35, целиком попадающее внутрь конечного интервала 0,3486784401 – 0,387420489. Для двоичного представления этого числа нам понадобится 7 бит (два десятичных разряда соответствуют примерно семи двоичным), тогда как для двоичного представления результатов кодирования из предыдущего примера – 0,80303498 – нужно 27 бит !!!

При декодировании предположим, что все что декодер знает о тексте, – это конечный интервал [0,8030349772 - 0,8030349880]. Декодеру, как и кодеру, известна также таблица распределения выделенных алфавиту интервалов. Он сразу же понимает, что первый закодированный символ есть *P*, так как результат кодирования целиком лежит в интервале [0,8 - 1), выделенном моделью символу *P* согласно таблице.

Теперь повторим действия кодера:

вначале [0,0 - 1,0);
после просмотра [0,8 - 1,0).

Исключим из результата кодирования влияние теперь уже известного первого символа *P*, для этого вычтем из результата кодирования нижнюю границу диапазона, отведенного для *P*, – 0,8030349772 – 0,8 = 0,0030349772 – и разделим полученный результат на ширину интервала, отведенного для *P*, – 0,2. В результате получим 0,0030349772 / 0,2 = 0,015174886. Это число целиком помещается в интервал, отведенный для буквы *A*, – [0 – 0,1), следовательно, вторым символом декодированной последовательности будет *A*.

Поскольку теперь мы знаем уже две декодированные буквы - *PA*, исключим из итогового интервала влияние буквы *A*. Для этого вычтем из остатка 0,015174886 нижнюю границу для буквы *A* 0,015174886 – 0,0 = 0,015174886 и разделим результат на ширину интервала, отведенного для буквы *A*, т. е. на 0,1. В результате получим 0,015174886/0,1=0,15174886. Результат лежит в диапазоне, отведенном для буквы *Д*, следовательно, очередная буква будет *Д*.

Исключим из результата кодирования влияние буквы *Д*. Получим (0,15174886 – 0,1)/0,1 = 0,5174886. Результат попадает в интервал, отведенный для буквы *И*, следовательно, очередной декодированный символ – *И*, и так далее, пока не декодируем все символы:

Декодированное число	Символ на выходе	Границы		Ширина интервала
		нижняя	верхняя	
0,8030349772	P	0.8	1.0	0.2
0,015174886	A	0.0	0.1	0.1
0,15174886	D	0.1	0.2	0.1
0,5174886	I	0.3	0.6	0.3
0,724962	O	0,7	0,8	0,1
0,24962	B	0,2	0,2	0,1
0,4962	I	0,3	0,6	0,3
0,654	3	0,6	0,7	0,1
0,54	I	0,3	0,6	0,3
0,8	P	0,6	0,8	0,2
0.0	Конец декодирования			

Это основная идея арифметического кодирования, его практическая реализация несколько сложнее. Некоторые проблемы можно заметить непосредственно из приведенного примера.

Первая состоит в том, что декодеру нужно обязательно каким-либо образом дать знать об окончании процедуры декодирования, поскольку остаток 0,0 может означать букву а или последовательность *aa*, *aaa*, *aaaa* и т. д. Решить эту проблему можно двумя способами.

Во-первых, кроме кода данных можно сохранять число, представляющее собой размер кодируемого массива. Процесс декодирования в этом случае будет прекращен, как только массив на выходе декодера станет такого размера.

Другой способ – включить в модель источника специальный символ конца блока, например #, и прекращать декодирование, когда этот символ появится на выходе декодера.

Вторая проблема вытекает из самой идеи арифметического кодирования и состоит в том, что окончательный результат кодирования – конечный интервал – станет известен только по окончании процесса кодирования. Следовательно, нельзя начать передачу закодированного сообщения, пока не получена последняя буква исходного сообщения и не определен окончательный интервал? На самом деле в этой задержке нет необходимости. По мере того, как интервал, представляющий результат кодирования, сужается, старшие десятичные знаки его (или старшие биты, если число записывается в двоичной форме) перестают изменяться (посмотрите на приведенный пример кодирования). Следовательно, эти разряды (или биты) уже могут передаваться. Таким образом, передача закодированной последовательности осуществляется, хотя и с некоторой задержкой, но последняя незначительна и не зависит от размера кодируемого сообщения.

И третья проблема – это вопрос точности представления. Из приведенного примера видно, что точность представления интервала (число десятичных разрядов, требуемое для его представления) неограниченно возрастает при увеличении длины кодируемого сообщения. Эта проблема обычно решается использованием арифметики с конечной разрядностью и отслеживанием переполнения разрядности регистров.

6.2.3. Сжатие данных по алгоритму словаря. Алгоритм словаря построен вокруг так называемой таблицы фраз (словаря), которая отображает строки символов сжимаемого сообщения в коды фиксированной длины, равные 12 бит. Таблица обладает свойством предшествования.

В настоящее время методы сжатия данных, включенные в протоколы *MNP5* и *MNP7*, целенаправленно заменяются на метод, основанный на алгоритме словарного типа Лемпеля–Зива–Вэлча (*LZW*-алгоритме), который имеет два главных преимущества:

- обеспечивает достижение коэффициента сжатия 4:1 файлов с оптимальной структурой;
- *LZW*-метод утвержден *ITU-T* как составная часть стандарта *V.42bis*.

Метод сжатия данных *LZW* основан на создании древовидного словаря последовательностей символов, в котором каждой последовательности соответствует единственное кодовое слово. Входящий поток данных последовательно, символ за символом, сравнивается с имеющимися в словаре последовательностями. После того как в словаре будет найдена кодируемая последовательность, идентичная входной, модем передает соответствующее ей кодовое слово. Алгоритм динамически создает и обновляет словарь символьных последовательностей.

Согласно кодировке, приведённой в табл. 6.4, в двоичном коде с помощью 8 бит можно закодировать 256 символов.

Таблица 6.4.

Кодирование буквенно-цифровых знаков

				<i>b8</i>	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
				<i>b7</i>	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	
				<i>b6</i>	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	
				<i>b5</i>	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
<i>b4</i>	<i>b3</i>	<i>b2</i>	<i>b1</i>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0			Пробел	0	@	P		p					ю	п	Ю	П
0	0	0	1	1	┐		!	1	A	<i>Q</i>	a	<i>Q</i>			<i>I</i>	±	a	я	A	Я
0	0	1	0	2			«	2	B	<i>R</i>	b	<i>R</i>			¢	2	б	р	Б	Р
0	0	1	1	3			#	3	C	<i>S</i>	c	<i>S</i>			£	3	ц	с	Ц	С
0	1	0	0	4	↑		□	4	<i>D</i>	T	<i>D</i>	t			\$	<i>X</i>	д	т	Д	Т
0	1	0	1	5	↓		%	5	E	<i>U</i>	e	<i>U</i>			¥		е	у	Е	У
0	1	1	0	6	→		&	6	<i>F</i>	<i>V</i>	<i>F</i>	<i>V</i>			#		Ф	ж	Ф	Ж
0	1	1	1	7	←		,	7	<i>G</i>	<i>W</i>	<i>G</i>	<i>W</i>			§	,	г	в	Г	В
1	0	0	0	8			(	8	H	<i>X</i>	h	<i>X</i>			□	• <i>R</i>	х	ь	<i>X</i>	Ь
1	0	0	1	9			)	9	<i>I</i>	<i>Y</i>	<i>I</i>	<i>Y</i>					и	ы	<i>I</i>	Ы
1	0	1	0	10			*	:	<i>J</i>	<i>Z</i>	<i>J</i>	<i>Z</i>					й	з	<i>Й</i>	З
1	0	1	1	11	┌		+	;	K	[	k	{			«	«	к	ш	K	Ш
1	1	0	0	12			,	<	<i>L</i>	\	<i>L</i>					¼	л	э	<i>L</i>	Э
1	1	0	1	13			-	=	M	]	m	}				½	м	щ	M	Щ
1	1	1	0	14			.	>	<i>N</i>	¬	<i>N</i>	-				¾	н	ч	N	Ч
1	1	1	1	15				?	O		o					¿	о	ъ	O	

Эти символы (вернее, их коды изначально заносятся в словарь программы, реализующей *LZW*). Во время работы программа посимвольно перебирает строку, подлежащую сжатию и передаче. При этом выполняется такая последовательность действий.

- Считываемый символ добавляется в формирующую строку. Если полученная строка уже присутствует в словаре, проверяется следующий символ.

- если полученной строки в словаре нет, передается предыдущая сформированная строка, а новая заносится в словарь.

Таким образом, считываемые символы используются для формирования отсутствующих в словаре строк, длина которых с каждым выполнением цикла сжатия увеличивается. Если обнаруживается, что такой последовательности символов в словаре еще нет, последняя сформированная строка передается на выход, а новая строка добавляется в словарь. Для указания положения строки в таблице строк словаря в алгоритме *LZW* используется числовой код. Если сформированную строку условно назвать префиксом, а считываемый символ – суффиксом, то работу алгоритма можно описать следующим образом:

префикс + суффикс = новая строка

После формирования новой строки суффикс становится префиксом:

префикс = суффикс

В качестве примера рассмотрим, как с помощью алгоритма *LZW* выполняется сжатие строки *ababc*, которая была передана модему терминалом. Вначале каждому символу словаря назначается числовое кодовое значение, соответствующее десятичному представлению этого символа в кодировке *ASCII*. То есть кодовое значение символа *a* равно 97, кодовое значение символа *b* – 98 и т. д.

В соответствии с алгоритмом *LZW*, при первой выполняемой операции (первом цикле) принимается, что префиксом является пустая строка, которую мы обозначим символом *F*. Поэтому при выполнении первой операции первый считываемый символ *a* добавляется к пустой строке, в результате чего формируется новая строка *a*. Поскольку *a* присутствует в словаре, на выход ничего не передается. Далее, согласно алгоритму, суффикс становится префиксом – *a* становится префиксом при формировании новой строки (этот этап сжатия строки *ababc* отображен в первой строке табл. 6.5).

Следующим шагом выполнения алгоритма *LZW* является считывание из строки ввода второго символа – *b*, который становится суффиксом. В ходе его обработки он добавляется к префиксу *a*, и в результате образуется новая строка *ab*. Этой строки нет в словаре программы, поэтому вступает в силу второе правило, согласно которому на выход передается последняя сформированная строка *a*, кодовое значение которой равно 97, а новая строка *ab* добавляется в словарь. Ранее уже говорилось, что для представления символов в кодировке *ASCII* используется 8 бит, что позволяет работать с 255 символами. Из этого следует, что новым строкам можно присвоить кодовые значения, которые будут больше

255 (256 и т. д.) и которые в двоичном представлении требуют большего количества битов. Первоначальный размер лексемы, используемый для представления новых строк, согласовывается модемами во время процесса согласования, выполняемого в соответствии со стандартом *V.42bis*.

Таблица 6.5.

Сжатие строки *ababc* в соответствии с алгоритмом *LZW*

Префикс	Суффикс	Новая строка	Выход
<i>F</i>	<i>A</i>	<i>A</i>	—
<i>a</i>	<i>B</i>	<i>ab</i>	97
<i>b</i>	<i>A</i>	<i>ba</i>	98
<i>a</i>	<i>B</i>	<i>ab</i>	—
<i>ab</i>	<i>C</i>	<i>abc</i>	256
<i>c</i>	<i>F</i>	<i>c</i>	99

Однако вернемся к рассмотрению процесса сжатия. Символ *b*, который был суффиксом при формировании строки *ab*, стал префиксом для следующей операции (это отображено в третьей строке табл. 6.5).

Далее считывается следующий символ — *a*, который тут же используется как суффикс при создании новой строки *ba*. Поскольку этой строки нет в словаре, на выход передается предыдущая строка из числа еще не переданных, *b*, кодовое значение которой равно 98 (в соответствии с кодировкой *ASCII*). Заметьте, что сформированная перед этим строка *ab* была добавлена в словарь, а не отправлена на выход. При добавлении в словарь строки *ba* ей присваивается следующий код — 257, а символ *a*, который был суффиксом при формировании этой строки, при выполнении следующей операции становится префиксом, что отражено в четвертой строке табл. 6.4. Затем считывается очередной (четвертый) символ строки ввода — *b*, при добавлении которого в качестве суффикса к предыдущей строке (*a*) образуется новая строка *ab*. Однако поскольку она уже была добавлена в таблицу строк (словарь), на выход ничего не передается, а сама строка становится префиксом при создании следующей строки.

Данный этап процесса сжатия отражен в пятой строке табл. 6.5 сформированная на предыдущем этапе строка *ab*, которая ранее была занесена в таблицу строк, стала префиксом при создании следующей строки, а последний символ *c* стал суффиксом. Полученная новая строка *abc* отсутствует в словаре, поэтому на выход передается последняя сформированная и не переданная строка — *ab*, точнее, передается присвоенное ей кодовое значение — 256. Символ *c* становится префиксом для создаваемой очередной строки, но так как он является последним символом строки ввода, его кодовое значение (99) передается на выход.

Декодер *LZW* должен использовать тот же словарь, что и кодер, строя его по аналогичным правилам при восстановлении сжатых данных. Каждый считываемый код разбирается с помощью словаря на предшествующую фразу *W* и

символ K . Затем рекурсия продолжается для предшествующей фразы W до тех пор пока она не окажется кодом одного символа. При этом завершается декомпрессия этого кода. Обновление словаря происходит для каждого декодируемого кода, кроме первого. После завершения декодирования кода его последний символ, соединенный с предыдущей фразой, добавляется в словарь. Новая фраза получает то же значение кода (позицию в словаре), которое присвоил ей кодер. В результате такого процесса декодер шаг за шагом восстанавливает тот словарь, который построил кодер.

Важное значение имеют алгоритмы сжатия LZ и LZW при архивации данных. Популярные архиваторы ARJ , PAK , $LHARC$ $PKZIP$ работают на основе этих алгоритмов.

6.2.4. Кодирование повторов. Кодирование последовательностей повторяющихся символов, метод RLE предусматривает замену последовательности повторяющихся символов на строку, содержащую этот символ, и число, соответствующее количеству его повторений. В качестве примера рассмотрим сжатие последовательности символов $ACCOUNTbbbbbbMOUNT$, в которой b означает символ пробела. Если для обозначения выполненного сжатия символов пробела модем использует специальный символ Sc , то между модемами будет передана последовательность символов $ACCONTSC7MOUNT$. Символ Sc в этой последовательности означает, что было произведено сжатие символов пробела, а число 7 указывает, сколько именно символов пробела заменено символом Sc . С помощью этой информации принимающий модем может восстановить данные.

Однако в последовательности передаваемых символов может встретиться пара символов S и c , которые являются частью данных, а не специальным символом Sc , обозначающим сжатие. Чтобы принимающий модем воспринимал эти символы как данные, передающий модем при обнаружении пары символов Sc добавляет в передаваемую последовательность еще одну такую пару. Таким образом, если модем принял от терминала поток данных $XYZSCABC$, то по телефонному каналу он передаст следующую последовательность символов: $XYZSCSCABC$. На принимающем модеме при обнаружении первого специального символа SC проверяется следующий символ. Если им окажется не число, а еще один такой символ, модем отбросит второй символ и восстановит первоначальный поток данных.

Сжатие позволяет увеличить пропускную способность систем передачи данных, но если один или более символов будут переданы с ошибкой, это может привести к очень печальным последствиям при восстановлении потока данных. В качестве примера покажем, к чему может привести ошибка при передаче последовательности символов $AAAAAAAA$. Предположим, что используется алгоритм сжатия RLE , в котором символ, означающий сжатие, представлен последовательностью битов 11111111, а символ A — последовательностью 01000001 (табл. 6.6.).

На рис. 8.6 демонстрируется, к чему может привести ошибка всего лишь в одном символе переданной последовательности битов.

Как видно из рассмотренного примера, последствия ошибки в символе при передаче сжатых данных намного серьезнее, чем при наличии такой же ошибки в случае обычной передачи. Поэтому во всех модемах, выполняющих сжатие данных, имеются встроенные программы, осуществляющие кодирование и декодирование сжатой последовательности в одном из корректирующих кодов.

– Последовательность одинаковых символов:	AAAAAAAA		
– Последовательность символов после сжатия:	Sc	8	A
– Двоичное представление передаваемых данных:	11111111	00001000	01000001
– Ошибка в символе:	11111111	00001000	01000011
– Принятая последовательность символов:	Sc	8	C
– Распакованная последовательность символов:	CCCCCCCC		

Рис. 6.6. Последствие ошибки в одном бите переданной сжатой строки

7. УСТРОЙСТВО ЗАЩИТЫ ОТ ОШИБОК (УЗО)

7.1. Основные типы УЗО

Наибольшей эффективностью в симплексных системах передачи обладает способ защиты от ошибок, основанный на использовании кодов с исправлением ошибок. В таких системах передаваемый блок кроме информационных единичных элементов, полученных от источника информации содержит и проверочные биты, которые формируются кодирующим устройством на основании информационных разрядов по определенным правилам. На приемной стороне декодером по тем же правилам осуществляются аналогичные проверки, при которых учитываются и проверочные элементы. В результате проверки определяется номер позиции в принятом блоке, значение которой необходимо в процессе исправления проинвертировать. Вероятность ошибочного приема символа зависит не только от вероятности ошибки в дискретном канале, но и от применяемого кода.

В односторонних системах передачи при выборе корректирующего кода для УЗО приходится рассчитывать на худшее состояние канала, т.е. избыточность в таких системах является постоянной, независимо от того, имеются ошибки в канале или нет, и пропускная способность канала используется неэффективно. Вторым недостатком систем с исправлением ошибок является резкое возрастание сложности аппаратуры с увеличением числа исправляемых ошибок. Существенное снижение аппаратурных затрат достигается применением в качестве кодирующих и декодирующих устройств микропроцессоров.

К дуплексным УЗО относятся устройства, в которых повышение верности принимаемой информации достигается за счет введения обратной связи. Они, в

свою очередь, делятся на системы с решающей (РОС), информационной (ИОС) и комбинированной (КОС) обратной связью. Механизм повышения верности в этих системах состоит в том, что при обнаружении искажений в принятом сообщении происходит запрос блока, в котором один или несколько неправильно принятых знаков. В системах с РОС передаваемые данные кодируются избыточными кодами, позволяющими обнаруживать одиночные ошибки или пачки (группы) ошибок. Решение о необходимости повторения передачи блока информации, в котором обнаружена ошибка, принимается приемником на основании анализа поступившей последовательности. В случае обнаружения в принятом блоке ошибок он стирается, и по каналу обратной связи (ОС) приемная станция посылает сигнал «Запрос», на основании которого передатчик высылает этот же блок. При безошибочном приеме блока данные поступают потребителю, а по каналу ОС передается сигнал «Подтверждение».

В УЗО с ИОС нет необходимости вводить избыточность в передаваемые данные. Двоичная последовательность, зафиксированная приёмником, запоминается и затем по каналу ОС передается (вся или в виде укороченной кодовой комбинации, содержащей определенные признаки всей последовательности) на передающую сторону. Полученная по каналу ОС информация анализируется передающей станцией, которая по результатам анализа принимает решения о передаче следующего блока либо о повторении ошибочно принятого. Это решение сообщается на приемную сторону и на его основании полученная информация выдается потребителю или стирается.

УЗО с КОС представляют собой сочетание информационной и решающей ОС. В них решения о необходимости повторной передачи могут приниматься как на передающей, так и на приемной сторонах, а по каналу обратной связи передаются информационные элементы или сигналы «Запрос» и «Подтверждение».

7.2. Алгоритм функционирования УЗО

Если реализация УЗО предполагается аппаратно, то разработку алгоритма функционирования целесообразно производить параллельно с разработкой структурной схемы УЗО. Алгоритм определяет основные функции устройства и последовательность их выполнения, а структурная схема представляет собой его техническую реализацию. Основные функции УЗО определяются сначала в общем виде. Приведем перечень основных функций, которые являются типовыми для всех видов УЗО:

- начальная установка блоков УЗО;
- прием, преобразование и контроль информации, поступающей от источника на передающей стороне, и выдача ее потребителю на приемной;
- обмен управляющими сигналами между отправителем и потребителем;
- генерирование тактовых импульсов и синхронизация (в случае работы без УПС);
- групповое фазирование (по циклам);

- формирование служебных символов начала и конца блока, номера блока, сигналов «Подтверждение», «Запрос», «Стирание» и др.;
- подсчёт числа бит в блоке;
- формирование номеров блоков при передаче и проверка соответствия очередности их поступления на приемной стороне;
- кодирование и декодирование сообщений;
- формирование информационных блоков и хранение их в буферных накопителях передатчика и приемника до принятия решения о приеме их с заданной верностью;
- подсчет числа переданных подряд одних и тех же блоков;
- формирование сигналов аварийной ситуации и их индикация;
- индикация состояния аппаратуры.

В зависимости от конкретного типа устройства перечень функций может быть дополнен и расширен, а также возможно исключение части функций. Очевидно, что для реализации этих функций в УЗО с «жесткой логикой» должны находиться соответствующие блоки, а в программируемом – соответствующие подпрограммы.

Первую функцию реализует, например, блок начальной установки устройства, который формирует импульс установки всех остальных блоков в исходное состояние. Начальная установка аппаратуры производится после включения питания или при переключении режимов работы. Вторую функцию реализует блок приема и преобразования вводимого сообщения, который обеспечивает кратковременное хранение поступающих кодовых комбинаций (байтов) и преобразование их в соответствующую форму (чаще всего в последовательный код). В этом блоке может происходить также согласование уровней сигналов, поступающих с ООД, с уровнями УЗО.

Устройство защиты от ошибок состоит из передающей (рис. 7.1) и приемной (рис. 7.2) частей.

7.3. Структурная схема передающей части УЗО

Передающая часть содержит следующие устройства: блок проверки преобразования информации (БППИ), буферный накопитель (БН), датчик номера блока (ДНБ), датчик служебных комбинаций (ДСК), кодер, формователи сигналов обмена с ООД и УПС (ФС01 и ФС02), блок начальной установки (БНУ), устройство управления (УУ), формователь тактовых импульсов (ФТИ), блок аварийной сигнализации и индикации (БАСИ), счетчик числа повторных запросов (СПЗ), анализатор обратного канала связи (АОКС).

Основным блоком УЗО является УУ, которое управляет работой всех остальных блоков. Управляющие воздействия на выходе УУ вырабатываются на основе анализа входящих сигналов и зависят от режима работы УЗО и временной позиции в пределах синхронизирующего или информационного блока. Оно представляет собой управляющий автомат, выполненный на основе жесткой или программируемой логики. УУ с жесткой логикой обычно строится на осно-

9 9 1 1 1 ()



Устройство защиты от ошибок с решающей обратной связью и безмаркерным способом группового фазирования работает следующим образом. В исходном состоянии обе части УЗО находятся в состоянии ожидания вызова. При поступлении запроса на передачу от источника информации УЗО обменивается управляющими сигналами с ООД и УПС в соответствии с техническими требованиями на стандартный стык. Для выработки необходимых сигналов обмена используются соответствующие формователи ФС01 и ФС02. С помощью УПС передающей части в канал связи посылается сигнал вызова, после приёма которого линия связи удаленной аппаратуры передачи данных (АПД) переключается с устройства автоматического выхода на вход УПС. Если АПД или ООД не готовы к процессу передачи данных, то включается сигнализация «Авария АПД» или «Авария ООД», а аппаратура переключается снова в режим ожидания вызова. При готовности АПД и ООД передатчик УЗО посылает в канал синхросигналы, которые используются для фазирований по циклу УУ приемника. Фазирующая последовательность формируется в передатчике датчиком ДСК, затем передается в канал связи и после приема фиксируется в РгСК. Сигнал регистрации обнаружения фазирующей (маркерной) комбинации вырабатывается ДшСК, поступает в БЦФ, который формирует соответствующие управляющие воздействия, смещающие по фазе УУ приемника до тех пор, пока оно не будет переключаться синфазно с УУ передатчика. Этот момент фиксируется ДшСК, который запрещает дальнейшее смещение фазы УУ приемника, а также вырабатывает сигнал подтверждения приема синхроблока и с помощью ФСОС передает его в канал ОС. В конце каждого цикла работы УУ передатчика опрашивает АОКС и в случае обнаружения сигнала подтверждения переключает АПД из режима фазирования в режим передачи данных. Если в течение заданного числа циклов (например, пяти) фазирование не достигнуто, то включается аварийная сигнализация «Нет фазы» и АПД переключается в режим ожидания вызова.

После завершения процесса группового фазирования ФС01 передатчика генерирует сигнал запроса данных от ООД отправителя сообщения.

Под действием управляющих сигналов УУ передатчика и информационной последовательности, поступающей с ООД и преобразуемой в последовательный код в БППИ, на соответствующих временных позициях добавляются кодовые комбинации номера блока, формируемые датчиком ДНБ, а также другие служебные символы (например, начало и конец блока), считываемые с ДСК. Данные, передаваемые в дискретный канал, кодируются помехоустойчивым кодом. Сформированные кодером проверочные элементы добавляются к информационным и служебным символам в конце блока. Поступившая от ООД информационная последовательность одновременно с передачей в канал записывается в буферный накопитель (БН), объем которого зависит от типа и алгоритма работы УЗО, а также от времени распространения сигналов по каналу связи. Аналогичным образом формируются последующие блоки. В конце каждого блока УУ передатчика спрашивает о состоянии анализатора обратную канал связи и в случае наличия сигнала «Подтверждение» осуществляет даль-

нейшую передачу последующих блоков либо при наличии сигнала «Запрос» прекращает ввод информации и повторно выдает из БН блок информации, в котором обнаружена ошибка. СПЗ контролирует число повторных запросов одного и того же блока и при достижении больше установленного числа переключает УЗО в режим циклового фазирования.

7.4. Структурная схема приёмной части УЗО

В состав приемной части входят блоки: регистр служебных комбинаций (РгСК), декодер (ДК), входной регистр (ВхРг), формирователь сигналов обратной связи (ФСОС), дешифратор служебных комбинаций (ДшСК), буферный накопитель (БН), блок преобразования и выдачи информации (БПВИ), формирователь тактовых импульсов (ФТИ), устройство управления (УУ), блок начальной установки (БНУ), формирователи сигналов обмена с ООД и УПС (ФС01 и ФС02), блок циклового фазирования (БЦФ), блок аварийной, сигнализации и индикации (БАСИ).

На приемной стороне с помощью УУ приемника происходит разделение служебных и информационных элементов. Первые записываются в РгСК, а вторые – во входной регистр ВхРг. ДшСК проверяет очередность и правильность поступления служебных комбинаций и информирует об этом УУ приемника. Одновременно ДК осуществляет декодирование принимаемого сообщения. Информационные элементы по мере заполнения ВхРг переписываются в БН. По завершении приема блока информации УУ приемника на основании данных о правильности приема служебных символов и отсутствии ошибок в принятом блоке, а также при наличии разрешающего сигнала с детектора уровня несущей (поступающего от УПС) разрешает выдачу информации ООД получателю. В блоке БПВИ осуществляются преобразование принятой последовательности в требуемый формат и согласование по уровню с уровнями ООД. Одновременно УУ приемника вырабатывает сигнал «Подтверждение», который формируется ФСОС и поступает в обратный канал связи. Если же при приеме блока информации нарушено хотя бы одно из принятых условий (искажены комбинации начала и конца блока или номер принятого блока не соответствует установленной очередности) либо в процессе декодирования обнаружена ошибка, происходит стирание принятого блока, а в обратный канал связи передается сигнал «Запрос».

Блоки аварийной сигнализации и индикации УЗО обеспечивают сигнализацию предусмотренных аварийных ситуаций и их индикацию, а также индикацию режимов работы УЗО. Выходные сигналы этого блока используются для информирования оператора или ЭВМ.

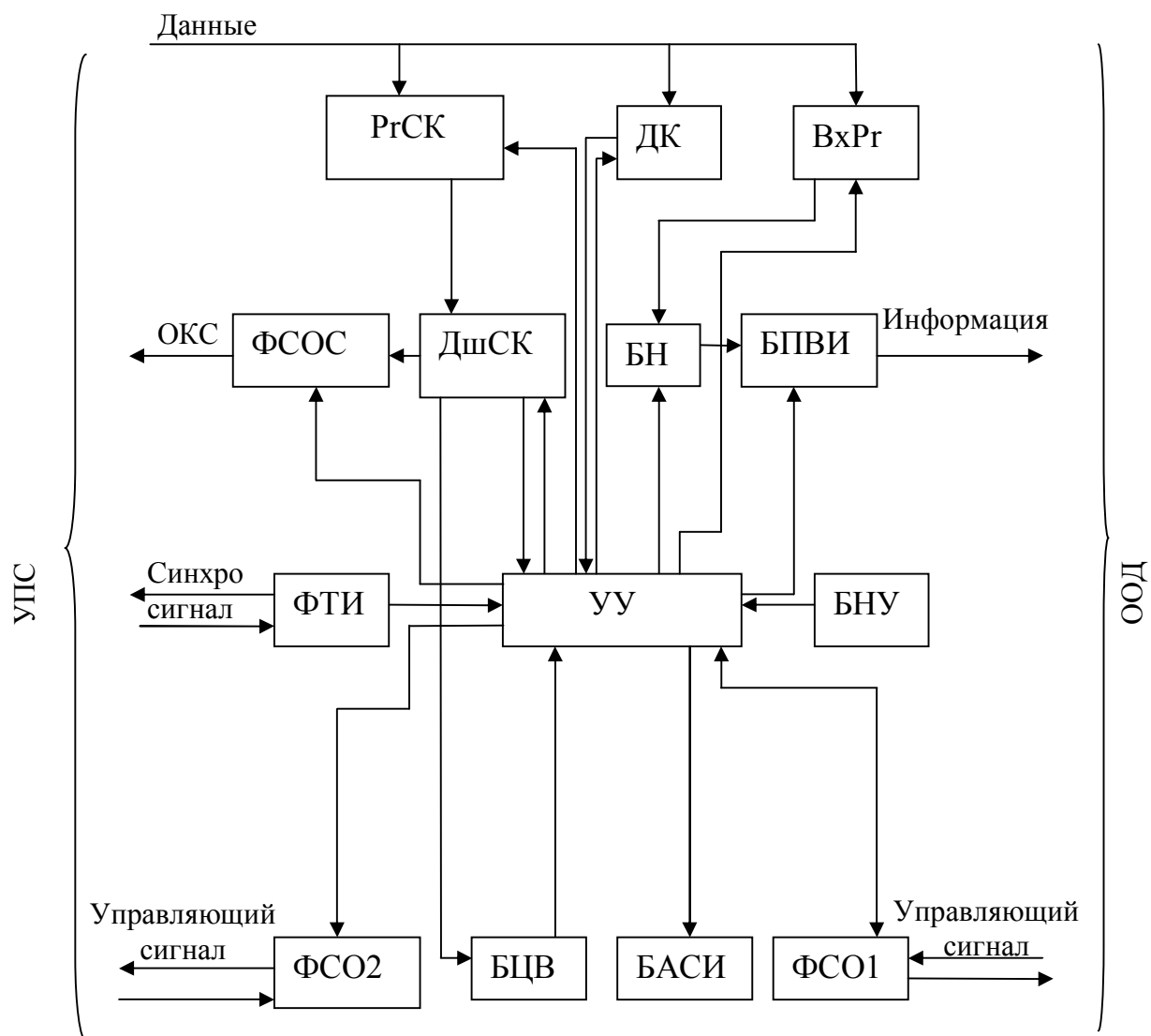


Рис. 7.2. Структурная схема приёмной части устройства защиты от ошибок

8. МОДЕМЫ

8.1. Общие сведения

Термин «модем» образован из двух слов: «модулятор» и «демодулятор». Модем изменяет не сами данные, а форму электромагнитного сигнала, приспособивая его для передачи данных в соответствующей среде.

Строгой классификации модемов не существует и, вероятно, не может существовать по причине большого разнообразия как самих модемов, так и сфер применения и режимов их работы. Тем не менее можно выделить ряд признаков, по которым и провести условную классификацию. К таким признакам или критериям классификации можно отнести следующие:

- область применения;
- функциональное назначение;
- тип используемого канала;
- конструктивное исполнение;
- реализация протоколов модуляции, исправления ошибок и сжатия данных.

Можно выделить еще множество более детальных технических признаков, таких как применяемый способ модуляции, интерфейс сопряжения с оконечным оборудованием данных (ООД) и так далее. На некоторых из них мы остановимся подробнее.

8.1.1. Область применения. Современные модемы можно разделить на несколько групп:

- для коммутируемых телефонных каналов;
- для выделенных (арендуемых) телефонных каналов;
- для физических соединительных абонентских линий (*XDSL*-модемы);
- для цифровых систем передачи;
- для сотовых систем связи;
- для пакетных радиосетей;
- для спутниковых каналов связи;
- для локальных радиосетей;
- для телевизионных кабельных сетей.

Подавляющее большинство выпускаемых модемов предназначено для использования на коммутируемых телефонных каналах. Такие модемы должны уметь работать с автоматическими телефонными станциями, различать их сигналы и передавать свои сигналы набора номера.

Основное отличие модемов да физических линий от других типов модемов состоит в том, что полоса пропускания физических линий не ограничена значением 3,1 кГц, характерным для телефонных каналов. Однако полоса пропускания физической линии также является ограниченной и зависит в основном от типа физической среды (экранированная и неэкранированная витая пара, коаксиальный кабель и др.) и её длины. С точки зрения используемых для подачи сигналов модемы для физических линий могут быть разделены на модемы низкого уровня (линейные драйверы), использующие цифровые сигналы, и моде-

мы «основной полосы», в которых применяются методы модуляции, аналогичные применяемым в модемах для телефонных каналов.

В модемах первой группы обычно используются цифровые методы биимпульсной передачи, позволяющие формировать импульсные сигналы без постоянной составляющей и часто занимающие более узкую полосу частот, чем исходная цифровая последовательность.

В модемах второй группы часто используются различные виды квадратурной амплитудной модуляции, позволяющие радикально сократить требуемую для передачи полосу частот. В результате на одинаковых физических линиях такими модемами может достигаться скорость передачи до 100 кбит/с, в то время как модемы низкого уровня обеспечивают только 19,2 кбит/с.

Модемы для цифровых систем передачи напоминают модемы низкого уровня. Однако в отличие от них обеспечивают подключение к стандартным цифровым каналам и поддерживают функции соответствующих канальных интерфейсов.

Модемы для сотовых систем связи отличаются компактностью исполнения и поддержкой специальных протоколов модуляции и исправления ошибок, позволяющих эффективно передавать данные в условиях сотовых каналов с высоким уровнем помех и постоянно изменяющимися параметрами.

Пакетные радиомодемы предназначены для передачи данных по радиоканалу между мобильными пользователями. При этом несколько радиомодемов используют один и тот же радиоканал в режиме множественного доступа. Радиоканал по своим характеристикам близок к телефонному и организуется с использованием типовых радиостанций, настроенных на одну и ту же частоту в УКВ либо КВ диапазоне. Пакетный радиомодем реализует методы модуляции и множественного доступа.

Локальные радиосети являются быстроразвивающейся перспективой сетевой технологией, дополняющей обыкновенные локальные сети. Ключевым их элементом являются специализированные радиомодемы (адаптеры локальных радиосетей). В отличие от ранее упомянутых пакетных радиомодемов такие модемы обеспечивают передачу данных на небольшие расстояния (до 300 м) с высокой скоростью (2 – 10 Мбит/с), сопоставимой со скоростью передачи в проводных локальных вычислительных сетях (ЛВС). Кроме того, радиомодемы локальных радиосетей работают в определенном диапазоне частот с применением сигналов сложной формы, таких как сигналы с псевдослучайной перестройкой рабочей частоты.

Модемы для телевизионных кабельных сетей получают всё большее распространение благодаря развитию кабельных ТВ сетей и высоким скоростям передачи. Такие модемы используют свободные телевизионные каналы с полосой пропускания в 6 МГц в диапазоне от 50 до 450 МГц для передачи своих сигналов. Большие полосы пропускания ТВ каналов обуславливают и высокие скорости передачи, которые достигают порядка 36 Мбит/с.

8.1.2. Метод передачи. По методу передачи модемы делятся на синхронные и асинхронные (раздел 4.2). Понятие синхронности модема в общем случае

может рассматриваться применительно к режиму передачи как по интерфейсу ООД – АКД (аппаратуре канала данных) (стыку, не зависящему от среды), так и по каналу связи (стыку, зависящему от среды). Коренной критерий синхронности метода передачи заключается в работе тактовых генераторов отправителя и получателя: независимой друг от друга (асинхронной) или согласованной (синхронной). Каждый полученный от ООД бит синхронный модем отправляет с той же скоростью и в том же виде, что и получил. Такой режим широко используется в синхронных сетях (X.25 и др.), а также совместно с различными контроллерами промышленной автоматики и т.п.

Напротив, модем считается асинхронным, если по интерфейсу ООД – АКД он работает в асинхронном режиме. В этом случае, данные формируются в стартстопные символы (несколько информационных бит, обрамленные стартовым и стоповым битами) и затем передаются по интерфейсу.

Как уже было отмечено, понятие синхронности модемов значительно реже относят к режиму передачи по каналу (стыку C1, зависящему от среды передачи). В этом случае «синхронность» модема необходимо рассматривать как «синхронность» конкретного используемого способа модуляции. В простых способах модуляции, таких как амплитудная и частотная, для осуществления процесса демодуляции не обязательно знать опорную или тактовую частоту передачи. Демодуляция практически сводится к селекции по амплитуде или частоте амплитудным или частотным детектором. Поэтому, данные способы модуляции и модемы, их реализующие, иногда называют асинхронными.

Напротив, осуществление демодуляции (амплитудно-) фазомодулированного сигнала просто невозможно без знания опорной частоты с точностью до фазы. Опорная частота в этом случае, как правило, выделяется из самого принимаемого сигнала и далее используется для осуществления процесса демодуляции. Чтобы принимающее устройство могло обеспечить надежное восстановление опорной частоты, передаваемая последовательность битов не должна содержать длинных последовательностей нулей или единиц.

8.1.3. Интеллектуальные возможности. По интеллектуальным возможностям можно выделить модемы:

- без системы управления;
- поддерживающие набор AT-команд;
- с поддержкой команд *V.25bis*;
- с фирменной системой команд;
- поддерживающие протоколы сетевого управления.

8.1.4. Конструктивное исполнение. По конструкции различают модемы:

- внешние;
- внутренние;
- портативные;
- групповые.

Внешние модемы представляют собой автономные устройства, подключаемые к компьютеру или другому ООД посредством одного из стандартных интерфейсов ООД – АКД. Внутренний модем – это плата расширения, встав-

ляемая в соответствующий слот компьютера. Каждый из вариантов конструктивного исполнения имеет свои преимущества и недостатки.

Портативные модемы предназначены для использования пользователями совместно с компьютерами *Notebook*. Они отличаются малыми габаритами и высокой ценой. Их функциональные возможности, как правило, не уступают возможностям полнофункциональных модемов.

Групповыми модемами называют совокупность отдельных модемов, объединенных в общий блок и имеющих общие блок питания, устройства управления и отображения. Отдельный модем группового модема представляет собой плату с разъемом, устанавливаемую в блок, и рассчитан на один или небольшое число каналов.

8.1.5. Поддержка международных и фирменных протоколов. Модемы также можно классифицировать в соответствии с реализованными в них протоколами. Все протоколы, регламентирующие те или иные аспекты функционирования модемов, могут быть отнесены к двум большим группам: международным и фирменным.

Протоколы международного уровня разрабатываются под эгидой *ITU-T* и принимаются им в качестве рекомендаций (ранее *ITU-T* назывался Международным Консультативным Комитетом по Телефонии и Телеграфии – МККТТ, международная аббревиатура – *CCITT*). Все рекомендации *ITU-T* относительно ТфОП модемов относятся к серии *V*. Фирменные протоколы разрабатываются отдельными компаниями – производителями модемов, с целью преуспеть в конкурентной борьбе. Классификация модемных протоколов представлена на рис. 8.1.

8.1.6. Структурная схема. Обобщенный вариант конструкции современного модема представлен в виде, изображенном на рис. 8.2. Модем состоит из адаптеров портов канального и *DTE-DCE* интерфейсов, универсального (*PU*), цифрового сигнального (*DSP*) и модемного процессоров, постоянного (ПЗУ, *ROM*), постоянного энергонезависимого перепрограммируемого (ППЗУ, *EPROM*), оперативного (ОЗУ, *RAM*) запоминающих, устройств и схемы индикаторов состояния модема.

Порт интерфейса *DTE-DCE* обеспечивает взаимодействие с *DTE*. Если модем внутренний, вместо интерфейсов *DTE-DCE* может применяться интерфейс внутренней шины компьютера *ISA* или *PCI*. Порт канального интерфейса обеспечивает согласование электрических параметров с используемым каналом связи. Канал может быть аналоговым или цифровым, с двух- или четырехпроводным окончанием.

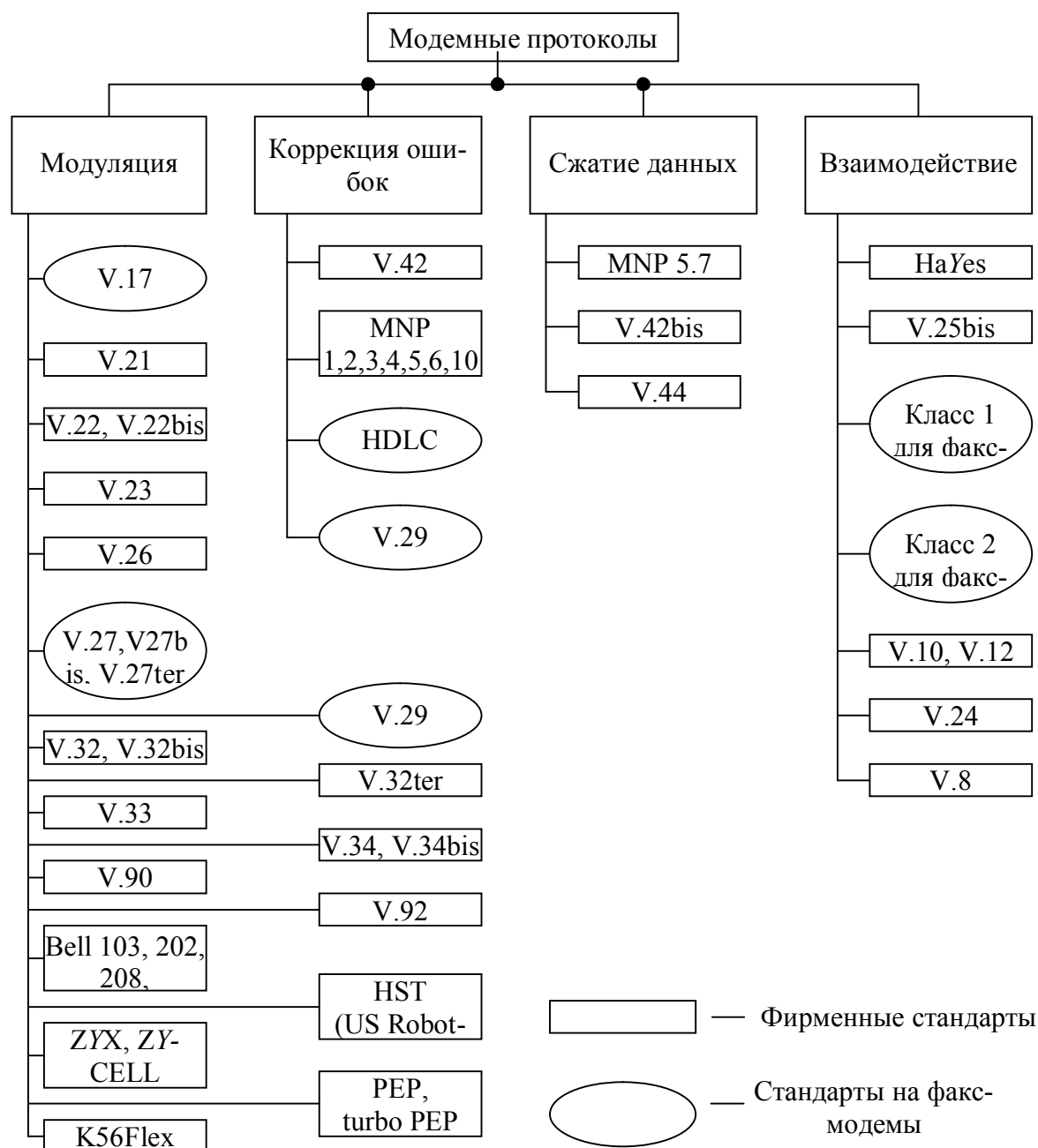


Рис. 8.1. Классификация модемных протоколов

Универсальный процессор *PU* выполняет функции управления взаимодействием с *DTE* и схемами состояния модема. Именно он выполняет посылаемые *DTE AT*-команды и управляет режимами работы остальных составных частей модема. Также универсальный процессор может реализовывать операции компрессии/декомпрессии передаваемых данных.

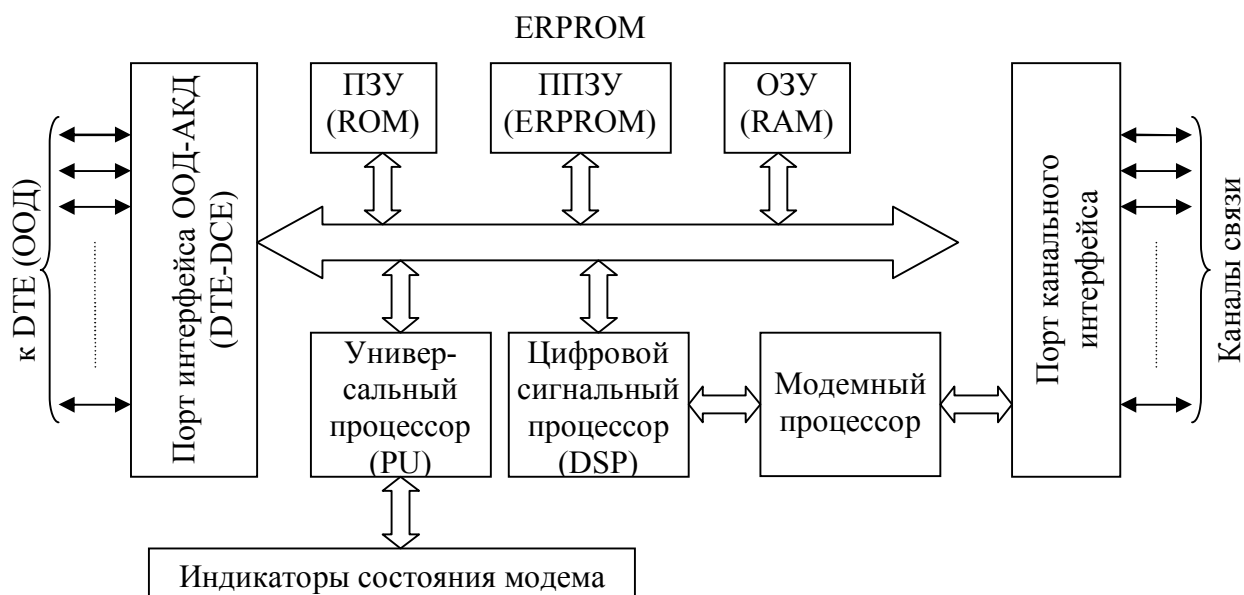


Рис. 8.2. Обобщённая структурная схема современного модема

Интеллектуальные возможности модема определяются в основном типом используемого *PU* и микропрограммой управления модемом, хранящейся в *ROM*. Путем замены или перепрограммирования *ROM* иногда можно достичь существенного улучшения свойств модема, т. е. произвести его модернизацию, или апгрейд (*UpGRaDe*). Такого рода модернизация некоторых моделей модемов может обеспечить поддержку новых протоколов или сервисных функций, таких как автоматическое определение номера (АОН) вызывающего абонента. Для облегчения такой модернизации в последнее время вместо микросхем *ROM* стали широко применяться микросхемы флэш-памяти (*FLaShROM*).

Схема *ERPROМ* позволяет сохранять установки модема в так называемых профайлах или профилях модема на время его выключения. Память *RAM* интенсивно используется для временного хранения данных и выполнения промежуточных вычислений как универсальным, так и цифровым сигнальным процессорами.

На сигнальный процессор, как правило, возлагаются задачи по реализации основных функций протоколов модуляции (кодирование свёрточным кодом, относительное кодирование, скремблирование и т. д.), за исключением разве что собственно операций модуляции/демодуляции. Последние операции обычно выполняются специализированным модемным процессором.

Описанное распределение функций между составными частями модема может быть, и, скорее всего, будет совсем не таким, какое реализовано в конкретном модеме. Однако внутренняя начинка современного модема все эти функции в той или иной мере должна выполнять.

8.2. Элементы синхронного модема для телефонной сети общего пользования (ТфОП)

Несмотря на приведённую на рис. 8.2 почти схемотехническую реализацию современного модема принципы работы его составных частей удобнее рассмотреть, опираясь на их функциональную интерпретацию, не зависящую от конкретной реализации.

С позиции исполняемых, функций, связанных с преобразованием передаваемых сигналов, современный синхронный модем содержит приёмник, передатчик, компенсатор электрического эха, схему управления и источник питания (рис. 8.3). Основные функции приемника и передатчика физически исполняются *DSP*.

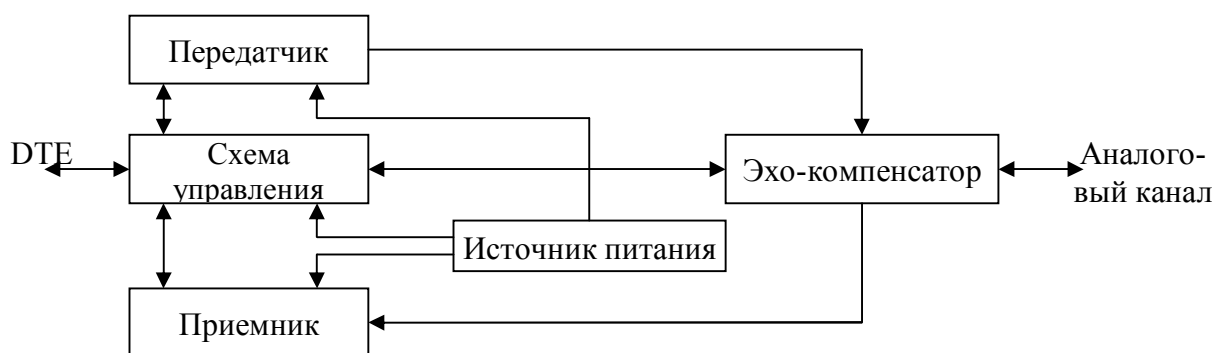


Рис. 8.3. Схема синхронного модема

Передаваемые *DTE* данные поступают в передатчик модема, который выполняет операции скремблирования, относительного кодирования, синхронизации и иногда вносит предискажения, частично компенсирующие нелинейность амплитудной и фазочастотной характеристик (АЧХ и ФЧХ) используемого телефонного канала. Схема передатчика приведена на рис. 8.4.

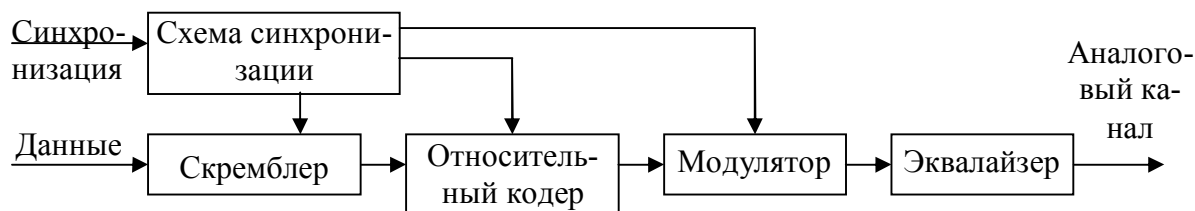


Рис. 8.4. Схема передатчика синхронного модема

Схема синхронизации передатчика получает сигнал, опорной частоты от внутреннего генератора или от *DTE*, например, через 24-й контакт разъема *DB-25* интерфейса *RS-232*. В последнем случае модем обязан поддерживать синхронный режим работы не только по каналу с удаленным модемом, но и по интерфейсу *DTE-DCE*. Скремблер предназначен для придания свойств случай-

ности (рандомизации) передаваемой последовательности, данных для облегчения выделения тактовой частоты приемником удаленного модема. При использовании сигналов ФМ и производных от них, применение относительного кодирования позволяет решить проблему неоднозначности фазы восстановленной на приеме несущей.

Приемник модема в свою очередь содержит адаптивный эквалайзер со схемой управления, модулятор с задающим генератором, демодулятор, относительный декодер, дескремблер, схему синхронизации (рис. 8.5). Модулятор приемника и задающий генератор позволяют перенести спектр принимаемого сигнала (300 – 3400 Гц) в область более высоких частот, для облегчения операций фильтрации и демодуляции. Относительный декодер и дескремблер выполняют операции, обратные выполняемым в передатчике. Схема синхронизации выделяет тактовую частоту из принимаемого сигнала и подает его на другие узлы приемника.

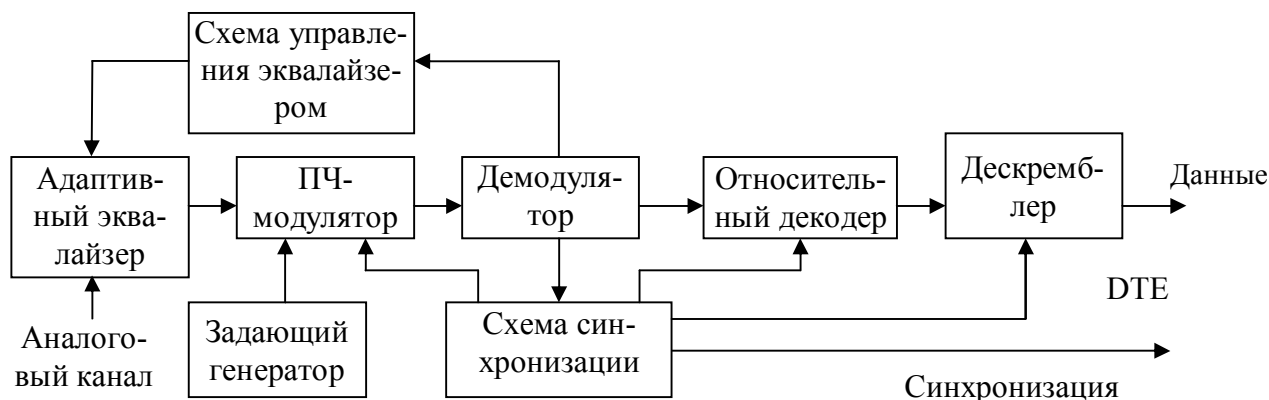


Рис. 8.5. Схема приёмника синхронного модема

Адаптивный эквалайзер приемника, как и эквалайзер передатчика, позволяет компенсировать нелинейные искажения, вносимые каналом передачи. Адаптивность эквалайзера заключается в его способности подстраиваться под изменяющиеся параметры канала в течение сеанса связи. Для этого сигнал ошибки фазы с демодулятора поступает на схему управления, которая вырабатывает управляющие сигналы для эквалайзера. Сам эквалайзер состоит из линии задержки с отводами и набора управляемых усилителей с изменяемым коэффициентом усиления.

В качестве примера синхронного модема рассмотрим модем *WeSteRN ELectRiс 201*, структурная схема которого приведена на рис. 8.6. Он осуществляет передачу данных со скоростью 1200 Бод и частотой несущей 1800 Гц. В нём используется разновидность модуляции со смещением фазы, при которой в единичном сигнале (Бод) кодируются 2 бита – двухбитовая относительная фазовая манипуляция. В модемах типа *WE 201* смещение фазы кратно 45° (табл. 8.1).

Фазовое кодирование, которое выполняется в модемах ИЕ 201

Двухбитовая последовательность	Сдвиг фаз, град
00	45
10	135
11	225
01	315

Давайте рассмотрим подробнее, какие процессы происходят в модеме при передаче и приеме данных. Чтобы облегчить восприятие излагаемого материала, каждый этап работы передатчика и приемника модема обозначен на рис. 8.6 буквой алфавита.

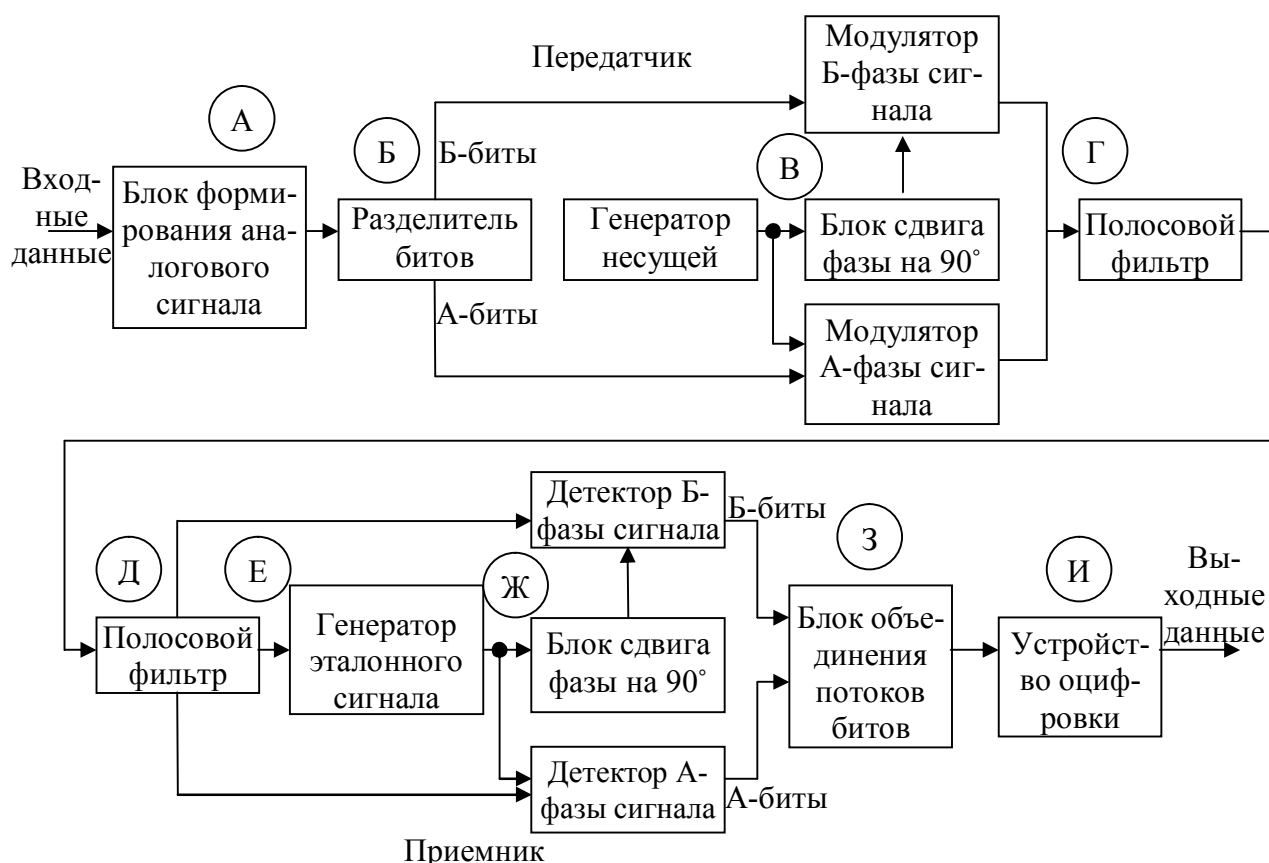


Рис. 8.6. Схема модема типа ИЕ 201

А. В соответствии с полученными на входе приемника данными генерируются импульсы, которые затем фильтруются с целью придания им подходящей для выполнения модуляции формы.

Б. Чередующиеся биты потока становятся А-битами, остальные – Б-битами.

В. Для фазовой модуляции синусоидального сигнала используется два битовых потока. Фаза несущей потока Б-битов смещена по отношению к фазе несущей потока А-битов на 90° .

Г. Далее аналоговые сигналы обоих модуляторов объединяются, а полученный сигнал подается на полосовой фильтр, где устраняются все нежелательные частоты.

Д. На приемнике полученный сигнал пропускается еще через один фильтр, который отсеивает шумы и другие нежелательные сигналы.

Е. Сигнал, поступивший на вход генератора эталонного аналогового сигнала, синхронизируется. (Посредством эталонного сигнала определяется сдвиг фазы поступившего единичного сигнала.)

Ж. На детектор А-фазы принятого сигнала эталонный сигнал подается без изменений, а перед подачей на детектор Б-фазы выполняется сдвиг его фазы на 90° .

З. Потоки битов А и Б, полученные на выходе обоих детекторов, объединяются в один битовый поток.

И. На завершающем этапе преобразования сигнал пропускается через устройство оцифровки, на выходе которого форма электромагнитных импульсов становится прямоугольной. Таким образом, данные приобретают ту же форму, какую имели при поступлении на вход передатчика.

При использовании описанной технологии кодирования синхронизационная информация извлекается непосредственно из самого сигнала. Иными словами, нет необходимости в применении каких-либо дополнительных данных синхронизации, на основе которых можно было бы определять сдвиг фазы принимаемого единичного сигнала. Таким образом, при возникновении одной из наиболее распространенных ошибок, которая состоит в считывании смещения фазы не того единичного сигнала, с ошибкой будет принят только один бит. В модеме ИЕ 201 имеется внутренний генератор частоты с фазовой синхронизацией, при работе которого используются значения сдвигов фаз входящих сигналов. На основе разности между номинальными значениями сдвигов фаз, которые приведены в табл. 5.4, и истинными значениями сдвигов фаз принятых сигналов определяется качество принимаемого сигнала. Поскольку смещения фаз каждого из принятых сигналов различаются, энергии сигнала, содержащейся в потоке, передаваемом со скоростью 1200 бод, вполне достаточно для восстановления тактовой частоты генератора.

8.3. Приёмник асинхронного модема

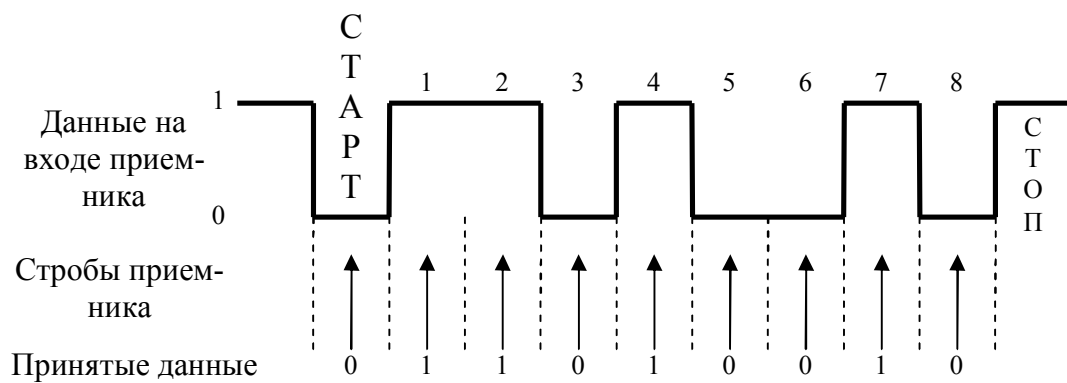
8.3.1. Синхронизация. Выполнение тактовой синхронизации является обязательным условием передачи и успешного приема данных, однако с ней связана одна из фундаментальных проблем, возникающих при передаче данных. Несмотря на то, что передатчик и приемник имеют генераторы тактовых импульсов одинаковой частоты, для декодирования данных приемник должен тем или иным образом определить (выделить) тактовую частоту передатчика

более точно. Рассмотрим последовательность передаваемых по линии битов. В случае, соответствующем рис. 8.7, а, приемник перенастраивает свой тактовый генератор на использование перехода с 1 на 0 для определения стартового бита (то есть настраивается на синхронизацию по заднему фронту), а затем определяет середину интервала времени прохождения бита. Из рис 8.7, б видно, что хотя тактовый генератор приемника имеет немного большую частоту, это не приводит к возникновению ошибки, поскольку строб попадает в интервал времени прохождения бита.

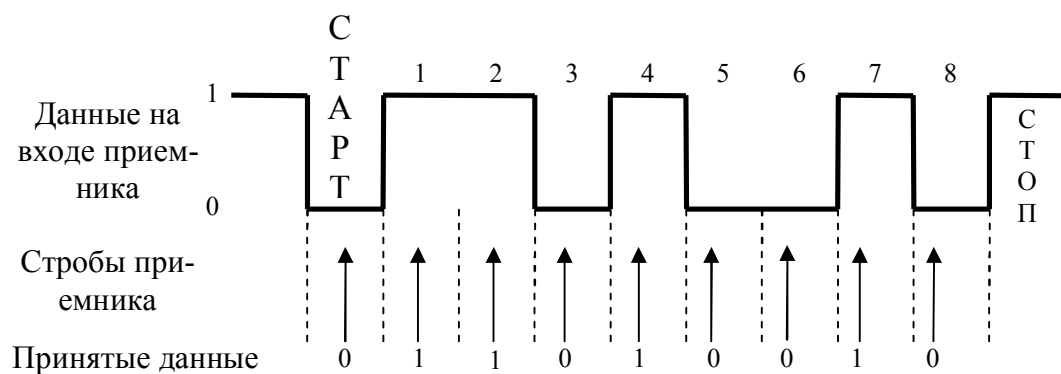
На рис. 8.7, в представлена еще одна ситуация. Здесь частота тактового генератора приемника настолько меньше тактовой частоты передатчика, что во время прохождения четвертого бита строб не генерируется вообще; таким образом, на выходе возникает ошибка. Для предотвращения возникновения ошибки система передачи должна обеспечивать синхронизацию приемника.

Кадровая синхронизация. Следующим этапом после тактовой синхронизации является **кадровая синхронизация**. Она используется при асинхронной передаче. Каждый символ задается битами (флагами) начала и конца передачи. Можно сказать, что на приведенной выше иллюстрации (рис. 8.4) представлен случай символьной синхронизации, так как символы, получаемые приемником, имеют флаги начала и конца (стартовый и стоповый биты).

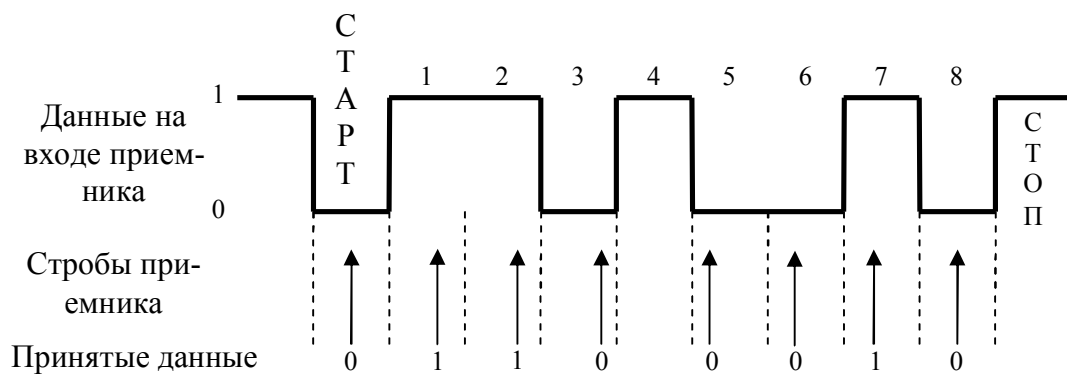
8.3.2. Структурная схема приёмника асинхронного модема. На рис. 8.8 изображена структурная схема приема асинхронного потока данных, закодированных 8-разрядными числами. Здесь применяется тактовый генератор с частотой в 16 раз больше, чем частота передачи символов входящего потока данных. Такая частота используется для как можно более точного определения перехода с 1 на 0 (стартового бита). Схема определения момента перехода с 1 на 0 отпирает схему обнаружения стартового бита. Подсчитываются 8 тактов генератора (половина длительности бита), и затем линия проверяется на присутствие логического нуля. Если проверка дает отрицательный результат, то предполагается, что переход с 1 на 0 произошел из-за помех в линии, схема обнаружения стартового бита запирается, и никаких дальнейших действий не производится.



а



б



в

Рис. 8.7. Асинхронная передача данных:
 а – строб попадает в середину интервала времени прохождения каждого бита;
 б – генератор тактовой частоты приёмника имеет несколько большую частоту;
 в – генератор тактовой частоты приёмника имеет слишком низкую частоту

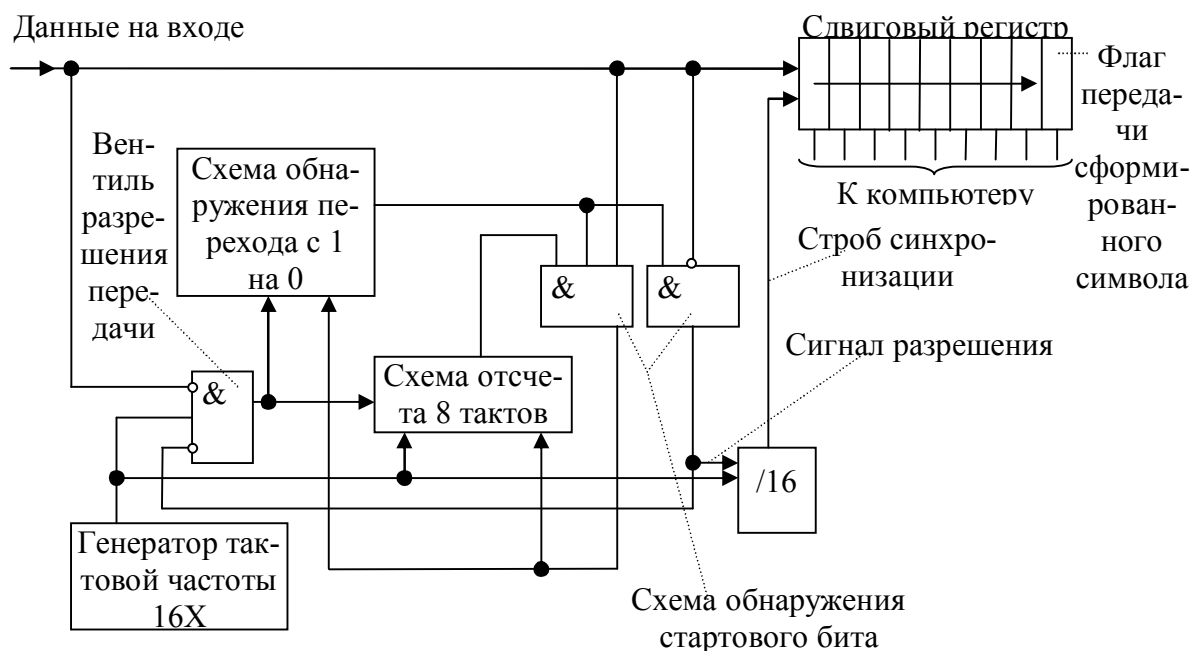


Рис. 8.8. Схема асинхронного приёмника

Если на линии по-прежнему находится напряжение логического нуля, значит, по ней передается правильный стартовый бит. Запускается счетчик, который делит частоту тактового генератора на 16 для генерации синхроимпульсов, производящих смещение содержимого сдвигового регистра на один бит при приеме нового бита. Отсчет проводится строго в середине интервала длительности бита. Смещение синхроимпульсов относительно середины интервала длительности бита можно уменьшить путем увеличения частоты тактового генератора в 32 или даже в 64 раза по сравнению с частотой следования импульсов. Однако при использовании более высоких частот синхронизации требуется соответственно увеличить емкость счетчиков схемы. В сдвиговой регистр для преобразования из последовательного кода в параллельный поступает группа из 8 бит. Затем компьютеру или контроллеру посылается соответствующий флаг, сигнализирующий о том, что символ принят. Получив флаг, компьютер посылает сдвиговому регистру разрешение на пересылку хранящихся в нем 8 бит для обработки.

При использовании только сдвигового регистра возникает следующая проблема. Так как символы поступают в сдвиговой регистр непрерывно, компьютер должен считать полученный символ за очень короткое время – время прихода стоповых битов. В этом случае простым решением является применение дополнительного (буферного) регистра для временного хранения информации. Данные из сдвигового регистра поступают в буферный регистр. Когда набор из 8 бит, составляющих символ, будет сформирован, установится флаг присутствия символа буферного регистра. После этого он будет готов к передаче символа в компьютер, а сдвиговой регистр – к приему нового символа.

На рис. 8.9 изображена схема приемного устройства с двумя регистрами. Существуют ситуации, при которых очередной символ не может быть обработан, поскольку из регистра не считан предыдущий. В результате происходит переполнение буфера. При переполнении большинство приемных устройств перезаписывают новый символ поверх старого (старый символ теряется) и дают компьютеру сигнал о возникновении ошибки.

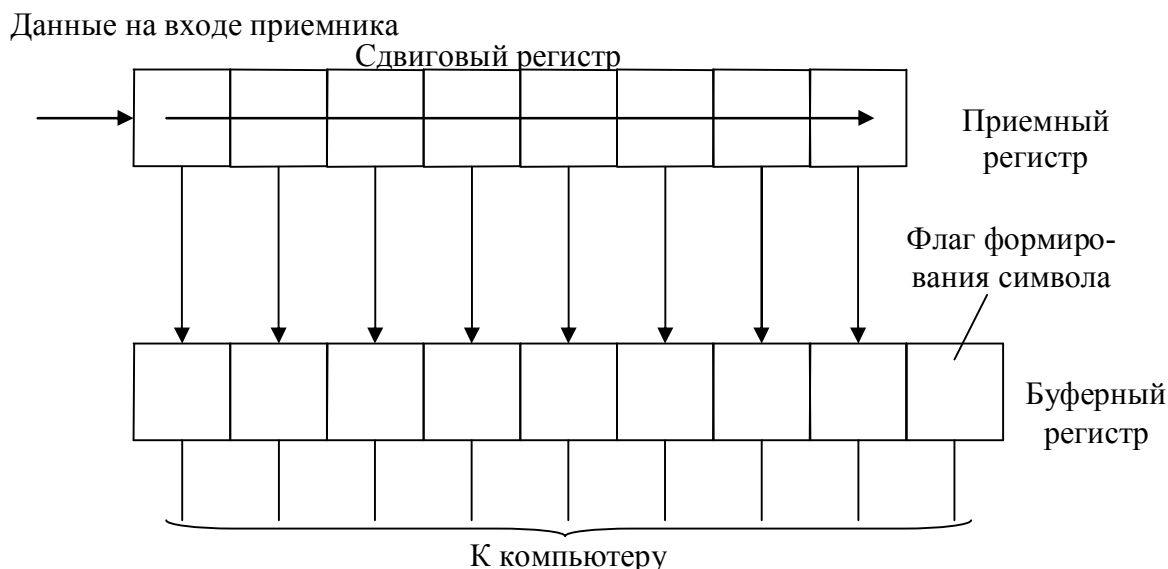


Рис. 8.9. Интерфейс с двумя регистрами

При использовании интерфейса с двумя регистрами интервал времени между поступлениями стоповых битов не больше, чем время считывания компьютером одного символа. Таким образом, для проверки стопового бита (иногда называемого девятым битом) на наличие логической 1 отводится достаточно времени. Стоповый бит может находиться в состоянии логического 0 из-за следующих ситуаций: неисправен канал связи, в приемнике произошел сбой синхронизации, передающая станция посылает специальный сигнал. При возникновении одной из вышеперечисленных ситуаций на асинхронный приемник посылается сигнал появления ошибки.

9. МЕТОДЫ И УСТРОЙСТВА СИНХРОНИЗАЦИИ

Как было показано выше, при обмене данными на физическом уровне единицей информации является бит, поэтому средства физического уровня всегда поддерживают побитовую синхронизацию между приемником и передатчиком. Чтобы приемник мог правильно декодировать получаемый набор битов, он должен знать:

скорость передачи битов, определяемую интервалом времени, выделяемым на один битовый разряд;

начало и конец каждого элемента (символа или байта);

начало и конец полного блока сообщений или кадра.

Эти три фактора называют соответственно побитной или тактовой синхронизацией, побайтной или посимвольной синхронизацией и поблочной или покадровой синхронизацией.

Канальный уровень оперирует кадрами данных и обеспечивает синхронизацию между приемником и передатчиком на уровне кадров. В обязанности приемника входит распознавание начала первого байта кадра, границ полей кадра и признака окончания кадра. Обычно достаточно обеспечить синхронизацию на указанных уровнях - битовом и кадровом, - чтобы передатчик и приемник обеспечили устойчивый обмен информацией. Однако при плохом качестве линии связи (как правило, это относится к телефонным коммутируемым каналам) для удешевления аппаратуры и повышения надежности передачи данных вводят дополнительные средства синхронизации на уровне байт. Такой режим работы называется асинхронным или старт-стопный.

9.1. Тактовая синхронизация

Под тактовой синхронизацией будем понимать процесс принудительного установления соответствия между периодами входящих импульсов и мгновениями их регистрации.

Необходимость тактовой синхронизации можно показать на следующем примере.

При фиксированном коэффициенте нестабильности генератора $k_n = \Delta f / f_n$, где f_n – номинальная частота абсолютно стабильного генератора, стробирующий импульс будет изменять свое местоположение в ту или иную сторону. Следовательно, через время $t = 1/\Delta f = 1/(k_n f_n)$ он может переместиться в середину соседнего импульса, т.е. на целый период регистрации. В телеграфных аппаратах частота f_n генераторов берется равной скорости дискретной модуляции B . Следовательно, можно принять $t = 1/(k_n B)$. С учетом наличия двух генераторов (на передаче и на приеме), в худшем случае имеющими отклонение частот от f_n в разные стороны, получим $t = 1/(2k_n B)$.

Смещение стробирующего импульса от идеального положения снижает исправляющую способность. Значит, это смещение возможно лишь в допустимых пределах ($\varepsilon_{don} = \Delta t_{don} / t_0$) и время, в течение которого строб достигнет границы установленной зоны (время поддержания синхронизма T_{nc}), $T_{nc} = \varepsilon_{don} / (2k_n B)$.

Например, если задаться $\varepsilon_{don} = 0,4$ и значением коэффициента нестабильности генераторов $k_n = 10^{-5}$, то при скорости дискретной модуляции $B = 50$ Бод время поддержания синхронизма составит 400 с (6 мин 40с). Если при тех же начальных условиях скорость дискретной модуляции увеличить до 2 400 Бод,

то время поддержания синхронизма $T_{ПС}$ составит всего 8,33 с. Из этого следует, что меры по поддержанию синхронизма необходимы.

9.1.1. Требования к устройствам синхронизации, их классификация.

К устройствам синхронизации дискретных систем связи предъявляются следующие требования: незначительное время вхождения в синхронизм при первоначальном включении; малая погрешность в работе на каналах с высоким уровнем помех; устойчивая работа при перерывах связи; простота построения, а следовательно, малая стоимость.

Классификацию УС можно проводить по следующим признакам (рис. 9.1) [17].

По методу коррекции рассогласования частот они бывают статические (стартстопные) и динамические (синхронные).

В **статических** УС регистрирующие импульсы выдаются «порциями» по специальным сигналам начала (старт) и конца (стоп). Если в работе распределителей обнаружится асинхронность, то «порция» может окончиться раньше или позже идеального мгновения. Следующая «порция» будет ожидать своего начала большее или меньшее время, т.е. рассогласование фаз устраняется изменением времени остановки приемного распределителя при неизменном периоде работы передающего распределителя.

В **динамических** УС распределитель приема работает непрерывно, и асинхронность устраняется изменением частоты генератора тактовых импульсов. Это накладывает более жесткие требования на стабильность генераторов по сравнению со статическими системами.

По частоте измерения рассогласования периодов УС делятся на системы с **подстройкой по специальным** (коррекционным) **импульсам** и на системы с **подстройкой по рабочим** (кодовым) **импульсам**. Наибольшее распространение получили последние.

По способу формирования регистрирующих импульсов УС делятся на системы с **местным генератором** (замкнутые, системы автоматического управления) и системы резонансные (разомкнутые, с фильтром). У первых есть генераторное оборудование (генератор тактовых импульсов – ГТИ), состоящее из задающего генератора синусоидальных колебаний, формирующего устройства для получения импульсов нужных формы и амплитуды и делителя частоты. Во вторых, последовательность регистрирующих импульсов вырабатывается непосредственно из входящих посылок, выделяя из них основную частоту резонатором (фильтром).

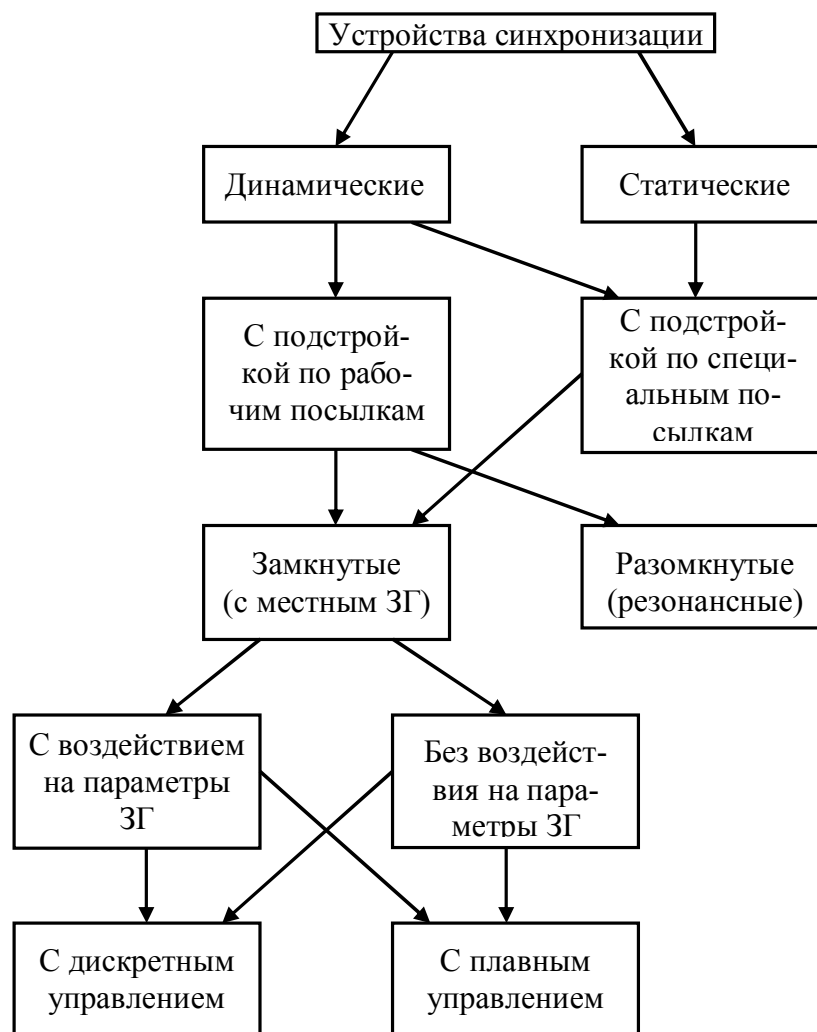


Рис. 9.1. Классификация устройств синхронизации

Резонансные УС проще замкнутых, но из-за низкой помехоустойчивости применяются редко.

Замкнутые УС по способу регулирования частоты ГТИ делятся на системы с непосредственным и системы с косвенным воздействием на параметры генератора. В первом случае (рис. 9.2) расхождение периодов, обнаруженное фазовым дискриминатором ФД через инерционный элемент ИЭ и управляющее устройство УУ, воздействует непосредственно на задающий генератор ЗГ, изменяя его частоту в ту или иную сторону. Инерционный элемент усредняет сигнал на выходе ФД за относительно продолжительное время, так как любое случайное отклонение периодов может вызвать ложное регулирование частоты ЗГ. Во втором случае (рис. 9.3) управляющее устройство действует на промежуточный преобразователь ПП, в качестве которого, как правило, применяют делитель частоты.

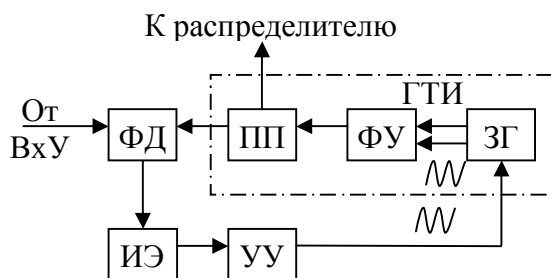


Рис. 9.2. Устройство синхронизации с непосредственным воздействием на параметры генератора

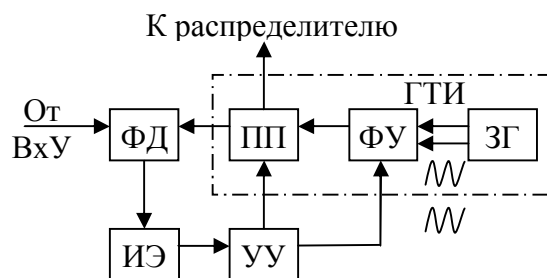


Рис. 9.3. Устройство синхронизации с косвенным воздействием на параметры генератора

По степени воздействия на положение (фазу) регистрирующих импульсов УС делят на устройства с релейным управлением и устройства с плавным управлением.

В первом из них рассогласование фаз, обнаруженное в I -м цикле, устраняется в последующих циклах. Степень воздействия на фазу регистрирующих импульсов постоянна, а число воздействий пропорционально рассогласованию.

Во втором случае рассогласование, обнаруженное в I -м цикле, устраняется в $(I+1)$ -м цикле. Степень воздействия на фазу регистрирующих импульсов пропорциональна измеренному рассогласованию.

9.1.2. Выделение синхросигнала и данных схемами на основе генераторов с фазовой автоподстройкой частоты. Генератор с фазовой автоподстройкой частоты (рис. 9.4) [18] содержит фазовый компаратор RD , инвертор, генераторы одинаковых по величине токов I_1, I_2 , ключи на транзисторах $VT1$ и $VT2$, фильтр низких частот и синхрогенератор VCO , управляемый напряжением. Сигнал DIN в коде NRZ (см. п. 3.1) поступает из линии на вход фазового компаратора. Этот сигнал не должен оставаться в неизменном состоянии (лог. 0 или 1) на протяжении более или менее длительного интервала времени. Например, при проектировании системы передачи данных может быть заранее известно, что максимальная длительность неизменного состояния сигнала не должна превышать двадцати битовых интервалов. (Напомним, что для обеспечения гарантированного изменения состояния сигнала в любом заранее заданном интервале времени применяют скремблирование этого сигнала перед его выдачей в линию, см.п. 3.5.)

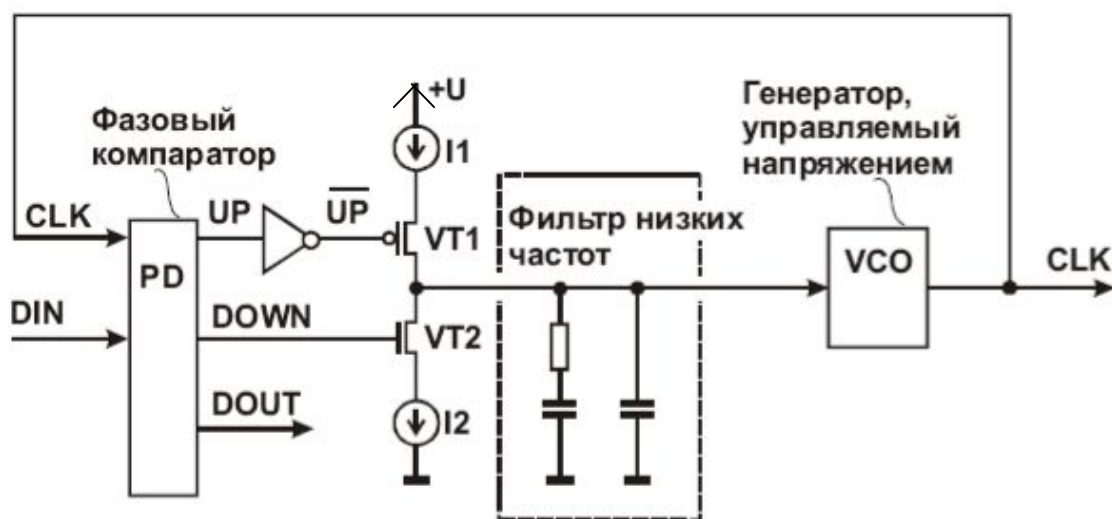


Рис. 9.4. Схема генератора с фазовой автоподстройкой частоты (первый вариант)

Фазовый компаратор сопоставляет сигнал *DIN* с синхросигналом *CLK* от генератора *VCO*. При правильном фазовом соотношении этих сигналов на выходе данных фазового компаратора формируется сигнал *DOUT*, соответствующий данным в линии, при этом границы его битовых интервалов задаются отрицательными фронтами сигнала *CLK*. Чтобы достичь, а затем и поддерживать правильное фазовое соотношение между сигналами *DIN* и *CLK*, фазовый компаратор непрерывно отслеживает ошибку взаимного расположения фронтов этих сигналов и формирует управляющие сигналы ускорения *UP* и замедления *DOWN* темпа работы генератора *VCO*.

Генератор *VCO* при номинальном управляющем напряжении, примерно равном $U/2$, формирует синхросигнал номинальной (ожидаемой) частоты. Повышение и понижение напряжения на управляющем входе генератора приводит к незначительному повышению и снижению частоты. Это позволяет подстраивать фазу синхросигнала в нужном направлении.

Чтобы пояснить, каким образом, варьируя частоту сигнала, можно корректировать его фазу, проведем «мысленный эксперимент», в котором частота сигнала *CLK* на выходе генератора *VCO* в незначительных пределах регулируется переменным резистором *R* (рис. 9.5). При некотором положении движка переменного резистора, близком к среднему, частота сигнала *CLK* совпадает с частотой «идеального» сигнала *CLKI*, к которому необходимо подстроиться. Сигнал *CLKI* поступает на вход синхронизации осциллографа и отображается его верхним лучом. При равенстве частот сигналов *CLK* и *CLKI* нижняя осциллограмма неподвижна относительно верхней, при этом, вероятнее всего, наблюдается некоторый фазовый сдвиг сигнала *CLK* относительно сигнала *CLKI*. На рисунке показана одна из возможных ситуаций, при которой сигнал *CLK* отстает по фазе от сигнала *CLKI* примерно на 110 градусов.

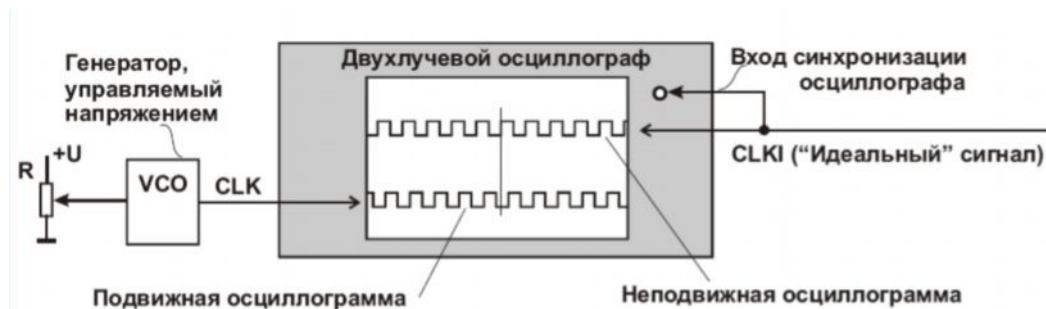


Рис. 9.5. Схема, поясняющая процесс фазовой подстройки частоты

Чтобы скомпенсировать несовпадение фаз, немного увеличим частоту сигнала CLK перемещением движка переменного резистора вверх. Фронты этого сигнала будут формироваться чуть раньше, нижняя осциллограмма начнет медленно перемещаться влево относительно неподвижной верхней, фазовый сдвиг начнет уменьшаться. Остается только приостановить перемещение в момент, когда фазовый сдвиг станет нулевым. Для этого в нужный момент следует вернуть движок переменного резистора в исходное положение, соответствующее равенству частот.

Сигналы UP и $DOWN$ (рис. 9.4) управляют транзисторными ключами, регулирующими поступление токов I_1 и I_2 на вход фильтра низких частот. При $UP = 0$, $DOWN = 0$ транзисторы $VT1$ и $VT2$ выключены, источники тока изолированы от фильтра и генератора VCO , напряжение на входе этого генератора остается постоянным либо незначительно изменяется из-за выравнивания напряжений на конденсаторах фильтра, если исходно эти напряжения не совпадали. При $DOWN = 0$ и поступлении серии импульсов $UP = 1$, транзистор $VT1$ включается по каждому импульсу, транзистор $VT2$ постоянно выключен, импульсы тока I_1 заряжают конденсаторы фильтра, напряжение на входе генератора VCO повышается, частота сигнала CLK незначительно возрастает. При $UP = 0$ и поступлении серии импульсов $DOWN = 1$ ситуация противоположная — транзистор $VT2$ открывается по каждому импульсу, транзистор $VT1$ постоянно закрыт, импульсы тока I_2 разряжают конденсаторы, частота сигнала CLK незначительно снижается. При $UP = 1$, $DOWN = 1$ оба транзистора включены и, поскольку $I_1 \ll I_2$, токи практически компенсируются и напряжение на входе генератора VCO почти не изменяется. В реальных условиях работы генератора (рис. 9.4) импульсы UP и $DOWN$, в основном, чередуются во времени и чаще всего не перекрываются. Благодаря фильтру низких частот, импульсные воздействия на частоту генератора VCO со стороны фазового компаратора усредняются. Заметный вклад в процесс регулирования частоты сигнала CLK может внести только усредненное по времени устойчивое расхождение этих воздействий.

Рассмотрим первый вариант схемы фазового компаратора RD (рис. 9.6). Схема содержит последовательно включенные D -триггеры $D1$, $D3$, инвертор $D2$ и элементы ИСКЛЮЧАЮЩЕЕ ИЛИ $D4$ и $D5$. Входные данные DIN принимаются триггером $D1$ по положительным фронтам сигнала CLK и переписываются в триггер

гер $D3$ по отрицательным фронтам этого сигнала. Элементы $D4$ и $D5$ формируют положительные импульсы UP и $DOWN$ (сигналы лог. 1) в периоды несовпадений сигналов на входах и выходах соответствующих триггеров $D1$ и $D3$. С увеличением этих периодов возрастает длительность соответствующих импульсов (фактически применена широтно-импульсная модуляция).

Таким образом, сигналы UP и $DOWN$ отражают фазовые соотношения между тремя сигналами - DIN и двумя его производными, полученными с использованием положительного и отрицательного фронтов сигнала CLK . При правильном соотношении сигналов DIN и CLK сигналы UP и $DOWN$ вносят одинаковые, но противоположные вклады в процессы ускорения и замедления работы генератора VCO , поэтому частота и фаза синхросигнала остается неизменной. При смещении фазы синхросигнала длительность и число импульсов UP или $DOWN$ возрастает, что приводит к компенсации смещения.

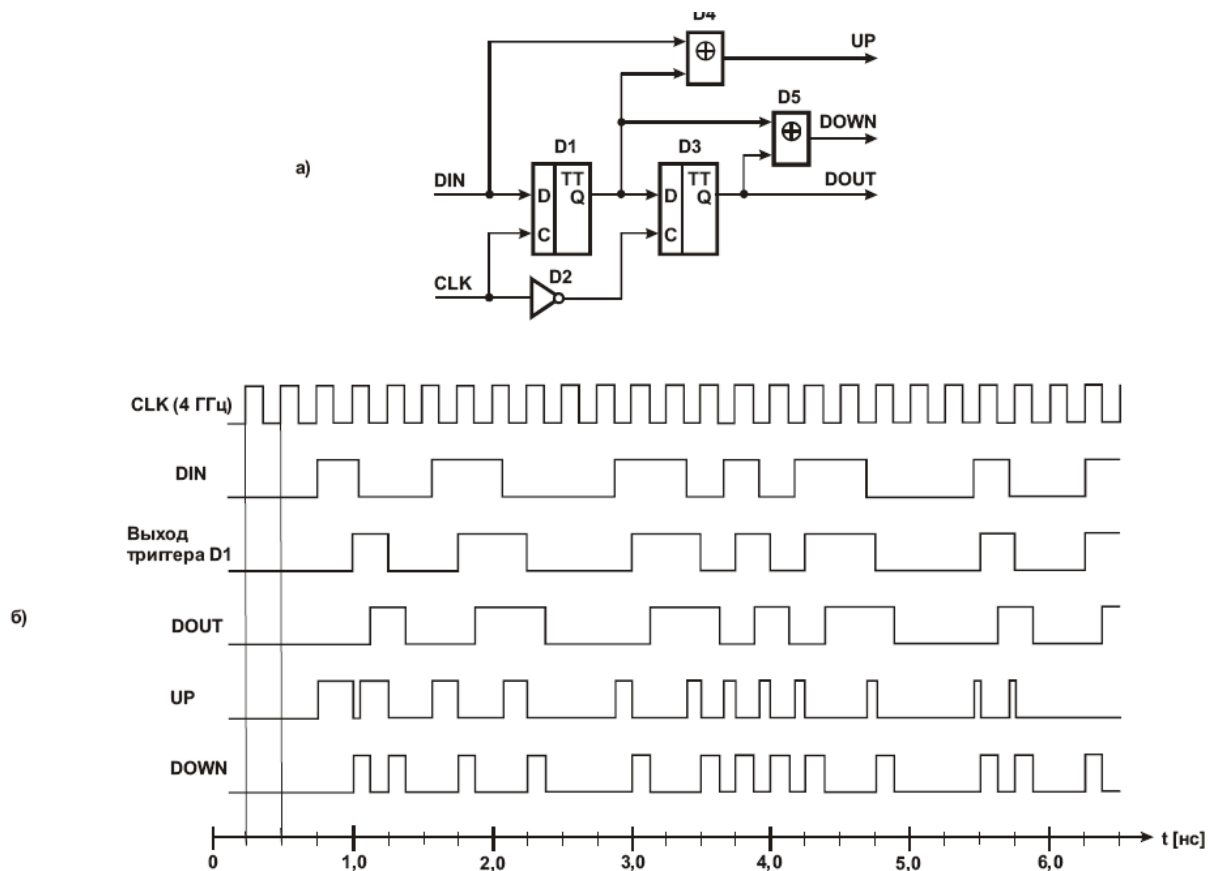


Рис. 9.6. Фазовый компаратор RD :
а - схема (первый вариант); б - временные диаграммы

Недостатком схемы (рис. 9.6) является высокая частота синхронизации. В приведенном примере частота сигнала CLK составляет 4 ГГц (период синхросигнала равен 0,25 нс). Эту частоту можно уменьшить вдвое, если использовать для приема данных как положительные, так и отрицательные фронты синхросигнала [18].

Прежде чем перейти к описанию второго варианта схемы фазового компаратора, рассмотрим его основную часть - формирователь сигналов *UP* и *DOWN* (рис. 9.7), работающий на пониженной частоте (2 ГГц).

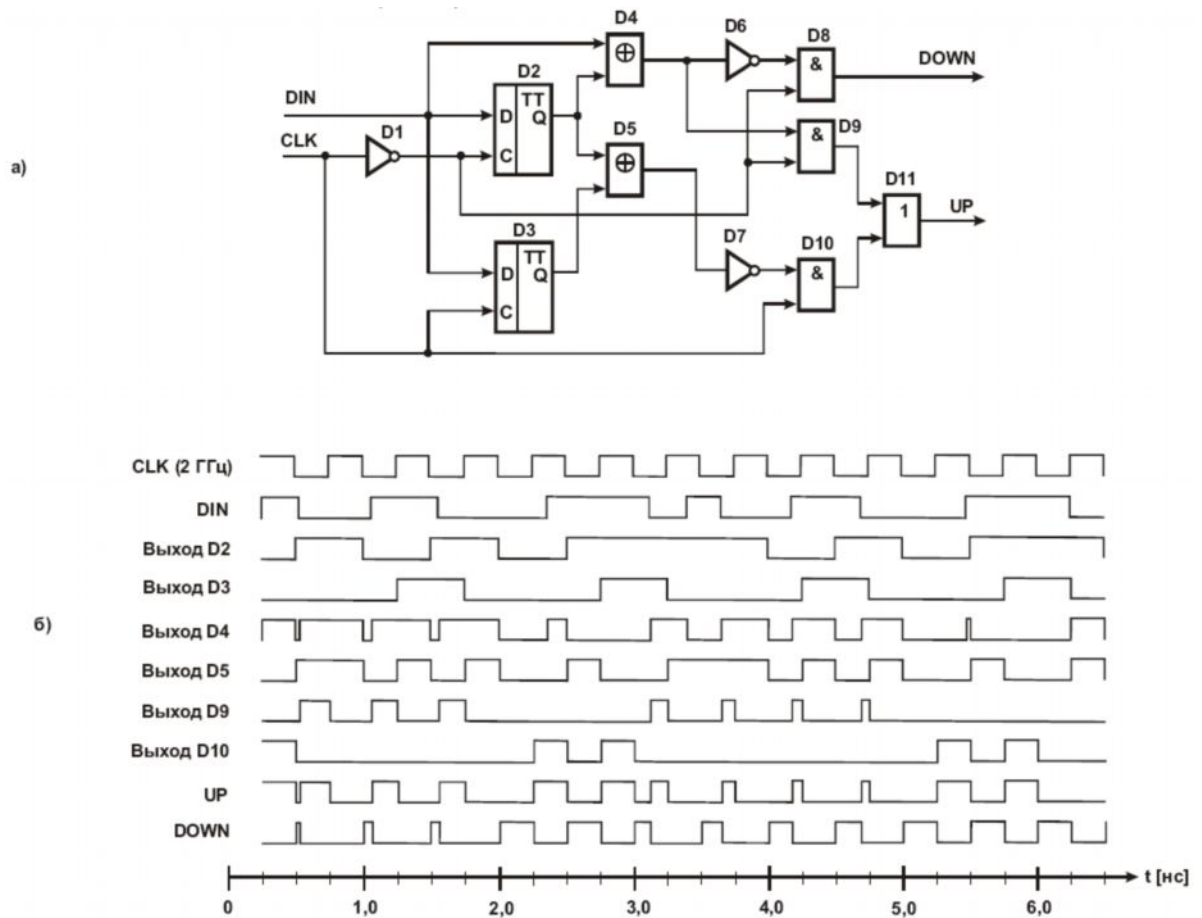


Рис. 9.7. Формирователь сигналов *UP* и *DOWN*, с синхронизацией от сигнала *CLK* пониженной частоты (2ГГц)

Входной сигнал *DIN* по положительному фронту сигнала *CLK* записывается в триггер *D3*, а по отрицательному фронту - в триггер *D2*. Элемент Иключающее ИЛИ *D4* формирует сигнал лог. 1 при несовпадении сигналов на входе и выходе триггера *D2*. Элемент Иключающее ИЛИ *D5* формирует сигнал лог. 1 при несовпадении сигналов на выходах триггеров *D2* и *D3*. Элементы *D6-D11* формируют сигналы *UP* и *DOWN* на основе сигналов с выходов элементов *D4*, *D5*, а также и прямого и инверсного сигналов *CLK*.

Второй вариант схемы фазового компаратора (рис. 9.8) содержит только что рассмотренный формирователь сигналов *UP* и *DOWN* (рис. 9.7), а также *D*-триггер *D12*.

Эта схема обладает следующими полезными особенностями. Во-первых, как уже отмечалось, при той же скорости поступления данных по линии, что и в схеме, показанной на рис. 9.6, она оперирует синхросигналом, частота которого уменьшена в два раза. Во-вторых, выходной поток данных распределен по двум каналам. Биты данных *DIN*, считанные из линии по положительным фронтам

сигнала CLK , временно запоминаются в триггере $D3$ и выдаются из него в виде потока $DOUT1$. Биты данных DIN , считанные из линии по отрицательным фронтам сигнала CLK , последовательно проходят через триггеры $D2$ и $D12$ и образуют поток $DOUT0$.

В чем состоит польза от разделения каналов? Во-первых, благодаря такому разделению, последующая обработка данных из обоих каналов может осуществляться параллельно, но вдвое медленнее по сравнению с обработкой данных из одноканального потока. Например, если выходные данные должны проходить через буферную память типа *FIFO*, то структура этой памяти распараллеливается: вместо одноразрядной быстродействующей памяти можно использовать двухразрядную, но более «медленную», и т. д.

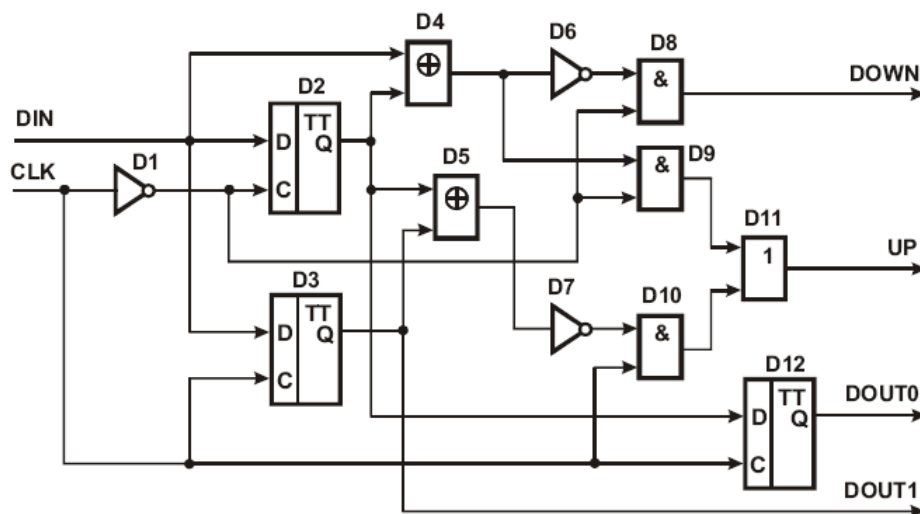


Рис. 9.8. Фазовый компаратор RD (второй вариант)

Во-вторых, разделение данных по каналам можно использовать для их демultipлексирования, если рассматривать входной поток данных DIN как сумму двух потоков от независимых источников: «четные» биты принадлежат первому потоку, «нечетные» - второму. В такой интерпретации рассмотренная схема (рис. 9.8) представляет собой элементарный демultipлексор ($DEMUX$ 1:2), распределяющий входной поток данных по двум направлениям. Из этих элементарных демultipлексоров можно строить древовидные структуры для увеличения числа каналов; одна из таких структур показана на рис. 9.9.

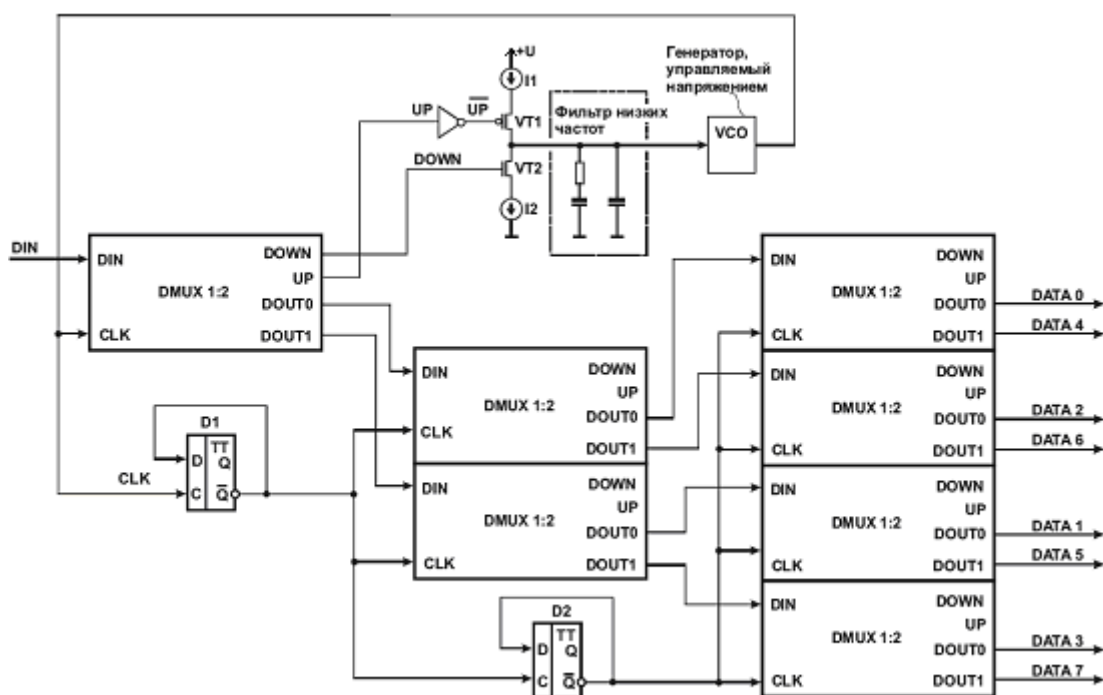


Рис. 9.9. Древоподобная структура из элементарных демультиплексоров (фазовых компараторов, выполненных по схеме на рис. 9.8)

В этой структуре элементарный демультиплексор первого уровня принимает сигнал DIN и управляет генератором VCO , включенным в контур фазовой автоподстройки частоты точно также, как и в рассмотренной ранее схеме (рис. 9.4). Для синхронизации двух демультиплексоров второго уровня и четырех демультиплексоров третьего уровня частота синхросигнала CLK от генератора VCO снижается в два и четыре раза с помощью делителя частоты на D -триггерах $D1$ и $D2$. Так как выходные сигналы UP и $DOWN$ демультиплексоров второго и третьего уровней не используются, формирователи этих сигналов (рис. 9.8, элементы $D6 - D11$) можно из них исключить. Структуру (рис. 9.9) можно также использовать в качестве преобразователя битового потока данных DIN в байтовый $DATA0 DATA7$.

9.1.3. Выделение синхросигнала и данных схемой на основе сдвиговых регистров. Схема, показанная на рис. 9.10 [25], предназначена для выделения синхросигнала $SYNC$ и данных $DOUT$ из сигнала DIN , поступающего через приемник из линии. Для начала предположим, что линейный сигнал представлен кодом NRZ (см. рис. 3.2). При этом допустим, что в коде NRZ максимальное число следующих подряд лог. 0 или лог. 1 не превышает, например, шести.

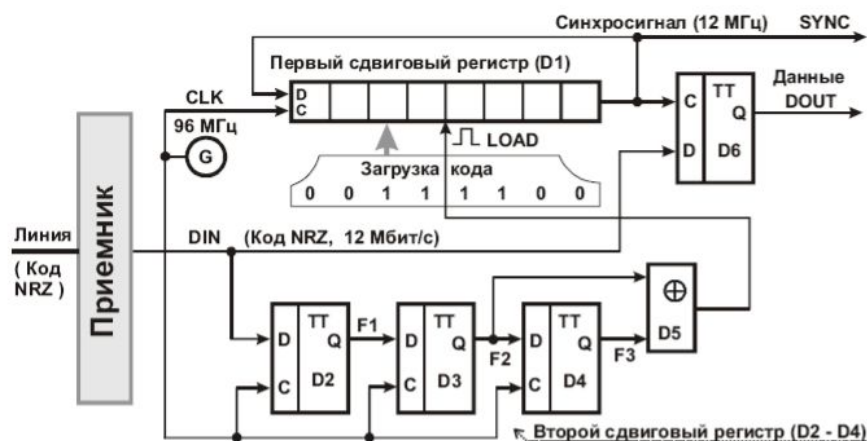


Рис. 9.10. Схема выделения синхросигнала и данных из линии. Входной сигнал представлен кодом *NRZ*

Схема построена на основе двух сдвиговых регистров. Первый регистр *D1* состоит из восьми включенных в «кольцо» *D*-триггеров с общими входами *C* синхронизации и объединенными входами *LOAD* управления асинхронной загрузкой. Циклический сдвиг кода в регистре *D1* выполняется в отсутствие сигнала *LOAD* (*LOAD* = 0) по положительному фронту синхросигнала *CLK* от кварцевого генератора *G*. При наличии сигнала *LOAD* = 1 регистр *D1* устанавливается в состояние 00111100 независимо от состояния сигнала *CLK*.

Частота сигнала *CLK* (96 МГц) в данном примере в восемь раз (по числу разрядов регистра *D1*) превышает скорость передачи данных по линии (12 Мбит/с). Сдвиговый регистр *D1* фактически выполняет функцию делителя частоты на восемь, так как в нем в идеальной ситуации непрерывно циркулирует код 00111100: в течение четырех тактов сдвига *SYNC* = 1, затем в течение следующих четырех тактов *SYNC* = 0 и т. д. При этом фаза сигнала *SYNC* привязана к сигналу *LOAD*, который, в свою очередь, формируется при любом изменении состояния входного сигнала *DIN*.

Второй сдвиговый регистр выполнен на триггерах *D2* - *D4* с объединенными входами *C* синхронизации. Логический элемент Исключающее ИЛИ *D5* формирует сигнал лог. 1 при несовпадении сигналов на его входах. Рассмотрим процесс выделения синхросигналов и данных из сигнала *DIN* (рис. 9.11).

В исходном состоянии, когда нет изменений сигнала *DIN* (или когда эти изменения были слишком давно), сигналы *SYNC* и *DOUT* не определены. Изменение сигнала *DIN* (в данном примере переход из нулевого в единичное состояние) регистрируется по ближайшему положительному фронту сигнала *CLK*. Это проявляется в том, что в триггере *D2* фиксируется сигнал *F1* = 1.

Отметим, что в результате неблагоприятного сочетания сигналов на входах этого триггера, а именно, когда в момент поступления положительного фронта сигнала *CLK* сигнал *DIN* нестабилен, триггер *D2* может сформировать кратковременный ложный сигнал. Иными словами, этот триггер может начать процесс «защелкивания», но не завершить его. Однако кратковременный ложный сигнал не будет распространяться в последующие каскады сдвигового регистра

благодаря общей цепи синхронизации. Полноценное защелкивание сигнала лог. 1 в триггере $D1$ окажется возможным только в следующем такте.

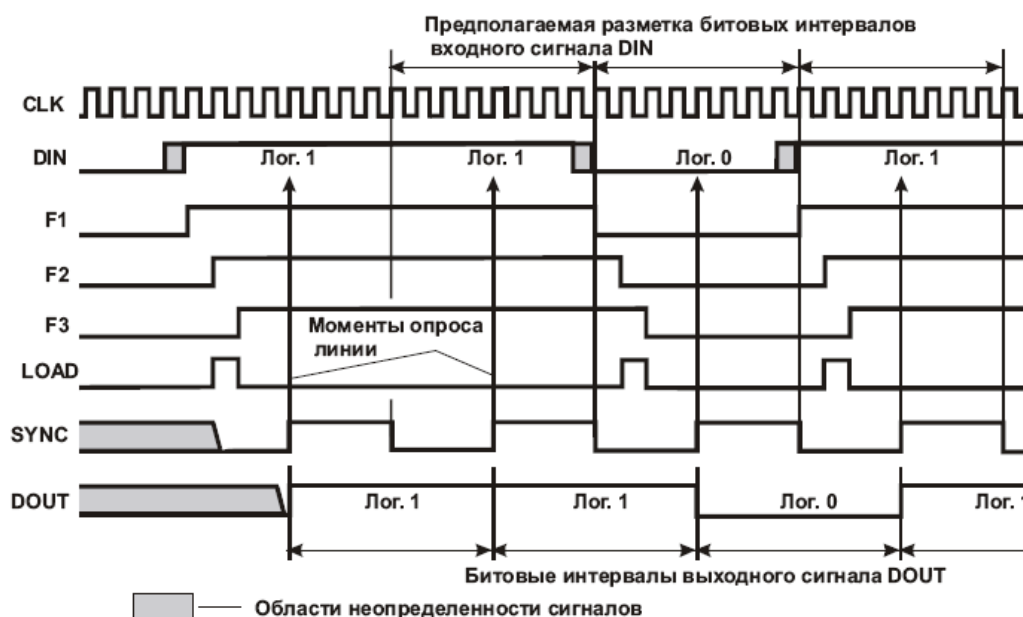


Рис. 9.11. Временные диаграммы выделения синхросигналов и данных из линии

Сигнал $F1 = 1$ в двух последующих тактах продвигается к выходу второго сдвигового регистра. При этом в период неоднозначности сигналов $F2$ и $F3$ формируется сигнал загрузки сдвигового регистра $DL\ LOAD = 1$, синхросигнал $SYNC$ принимает нулевое значение. Через два такта после окончания сигнала $LOAD = 1$ формируется положительный фронт сигнала $SYNC$, гарантирующий правильную регистрацию бита данных $DOUT$ в триггере $D6$. Начиная с этого момента, достигаются нужные соотношения между выходными и входными сигналами. Чтобы убедиться в правильности этих соотношений, рассмотрим временную диаграмму, приведенную на рис. 9.12.

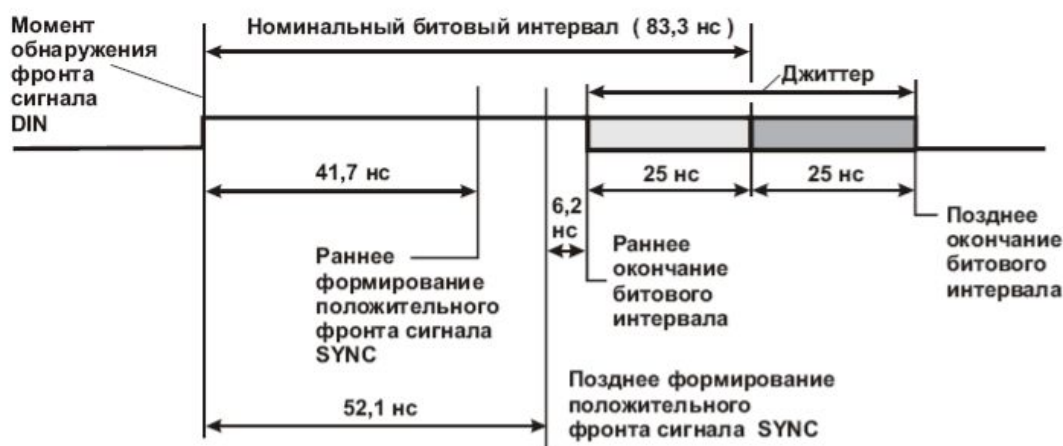


Рис. 9.12. Временная диаграмма выделения бита данных с учетом джиттера и некоторой неопределенности момента формирования положительного фронта сигнала $SYNC$

При скорости передачи данных 12 Мбит/с номинальная длительность битового интервала составляет $1/(12\,000\,000)$ с или 83,3 нс. Если принять максимальную амплитуду джиггера равной 25 нс, то битовый интервал в наихудших ситуациях может быть укорочен или удлинен на 25 нс и равен соответственно $83,3 - 25 = 58,3$ нс или $83,3 + 25 = 108,3$ нс. Допуски длительности битового интервала отмечены на рисунке затененными областями.

Период сигнала CLK равен $1/(96\,000\,000)$ с или 10,417 нс. Из ранее рассмотренной временной диаграммы (см. рис. 9.11) следует, что положительный фронт сигнала $SYNC$ при наилучшем стечении обстоятельств может сформироваться через четыре периода сигнала CLK после изменения сигнала DIN . При наихудшем стечении обстоятельств (когда изменение сигнала DIN чуть опоздало к моменту регистрации) вместо четырех периодов потребуется пять.

Первой и второй ситуации соответствуют отмеченные на рис. 9.12 задержки, равные соответственно $10,417 \times 4 = 41,7$ нс и $10,417 \times 5 = 52,1$ нс. Таким образом, обеспечиваются надежная предустановка сигнала DIN на D -входе триггера $D6$ (минимум 41,7 нс) и достаточное время удержания этого сигнала (минимум 6,2 нс при норме «больше нуля») после формирования сигнала $SYNC$.

Если для загрузки сдвигового регистра $D1$ вместо кода 00111100 применить код 00011110, то моменты раннего и позднего формирования положительного фронта сигнала $SYNC$ сместятся на временной диаграмме влево на один период сигнала CLK . При этом минимальная предустановка уменьшится до 31,2 нс, а время удержания увеличится до 16,6 нс.

В примере, приведенном на рис. 9.13, передаваемые по линии данные представлены кодом $HDB3$ (рис. 9.14). Построение этого кода было рассмотрено в п. 3.1.6. Напомним, что в коде $HDB3$ цепочки нулей заменены определенными «заготовками» (выделены на рис. 9.14 серым фоном), что позволяет исключить продолжительные паузы входного сигнала.

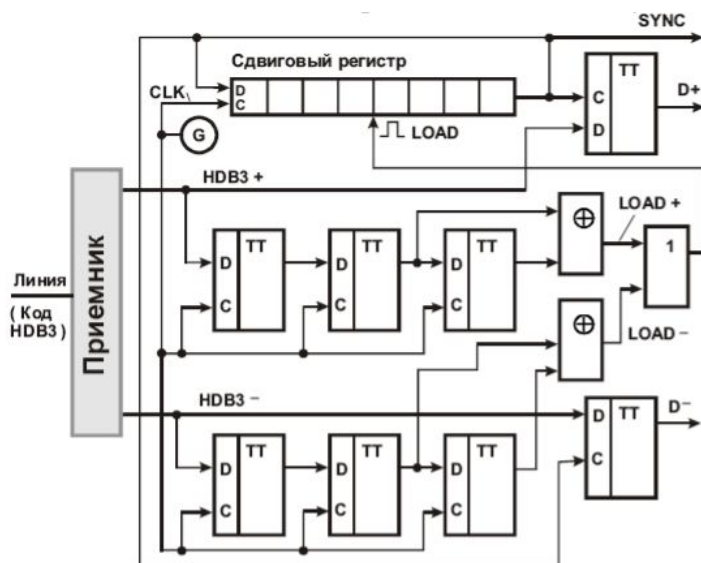


Рис. 9.13. Схема выделения синхросигнала и данных из линии

Приемник формирует две последовательности сигналов: *HDB3+* и *HDB3-*. Первая соответствует положительным импульсам в линии, вторая - отрицательным. Трехразрядные сдвиговые регистры формируют импульсы *LOAD +* и *LOAD -*, которые отражают изменения состояний сигналов *HDB3+* и *HDB3-*.

Эти импульсы суммируются логическим элементом ИЛИ, поэтому сигнал *LOAD* отражает все изменения сигнала в линии. Импульсы *LOAD +* и *LOAD -*, обведенные на рисунке овалами, теоретически формируются одновременно, но реально при определенном стечении обстоятельств могут иметь взаимный сдвиг в ту или иную сторону на один период сигнала *CLK*. При этом длительность суммарного импульса *LOAD* (выделен на рисунке прямоугольником) составит не два, а три периода сигнала *CLK*. Это, однако, учитывается допусками (см. рис. 9.12) и не нарушает работы устройства.

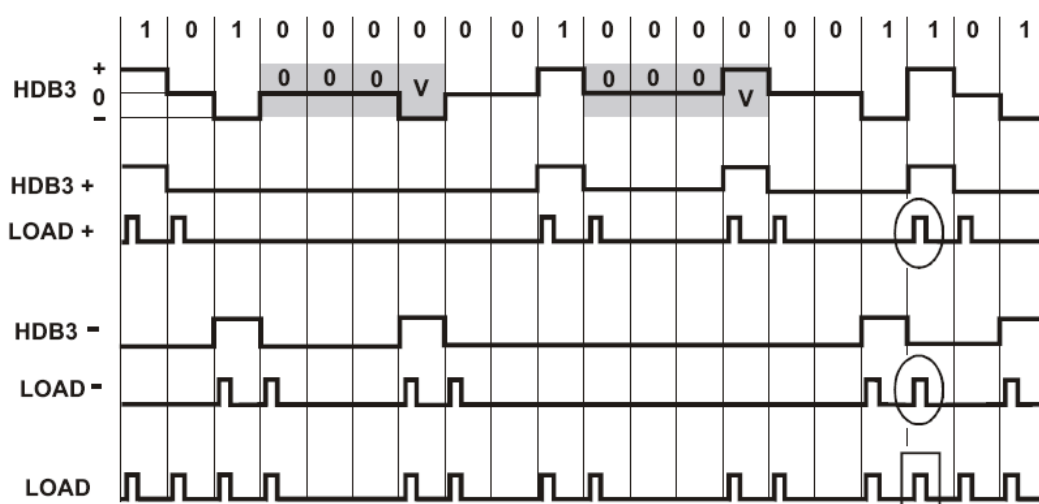


Рис. 9.14. Временные диаграммы формирования сигнала *LOAD*

Выходные сигналы *D+* и *D-* синхронизированы сигналом *SYNC* и соответствуют положительным и отрицательным импульсам на входе устройства. Дальнейшая обработка сигналов *D+* и *D-* заключается в восстановлении исходной битовой последовательности (показанной в верхней части рис. 9.14) и может выполняться микропроцессором. Подстройку фазы тактовых импульсов приемника можно производить путем удлинения или укорачивания регистра сдвига. Схема такого устройства синхронизации приведена на рис. 9.15. Регулирование фазы выполняется без непосредственного воздействия на генератор тактовых импульсов, для чего в состав формирователя тактовых импульсов включается кольцевой десятитактный регистр сдвига, разделяющий тактовый импульс на десять не пересекающихся во времени интервалов. Фаза тактовых импульсов приемника изменяется искусственным укорочением регистра сдвига при отставании тактовых импульсов приемника от поступающих из канала связи импульсов или удлинением регистра сдвига в противном случае.

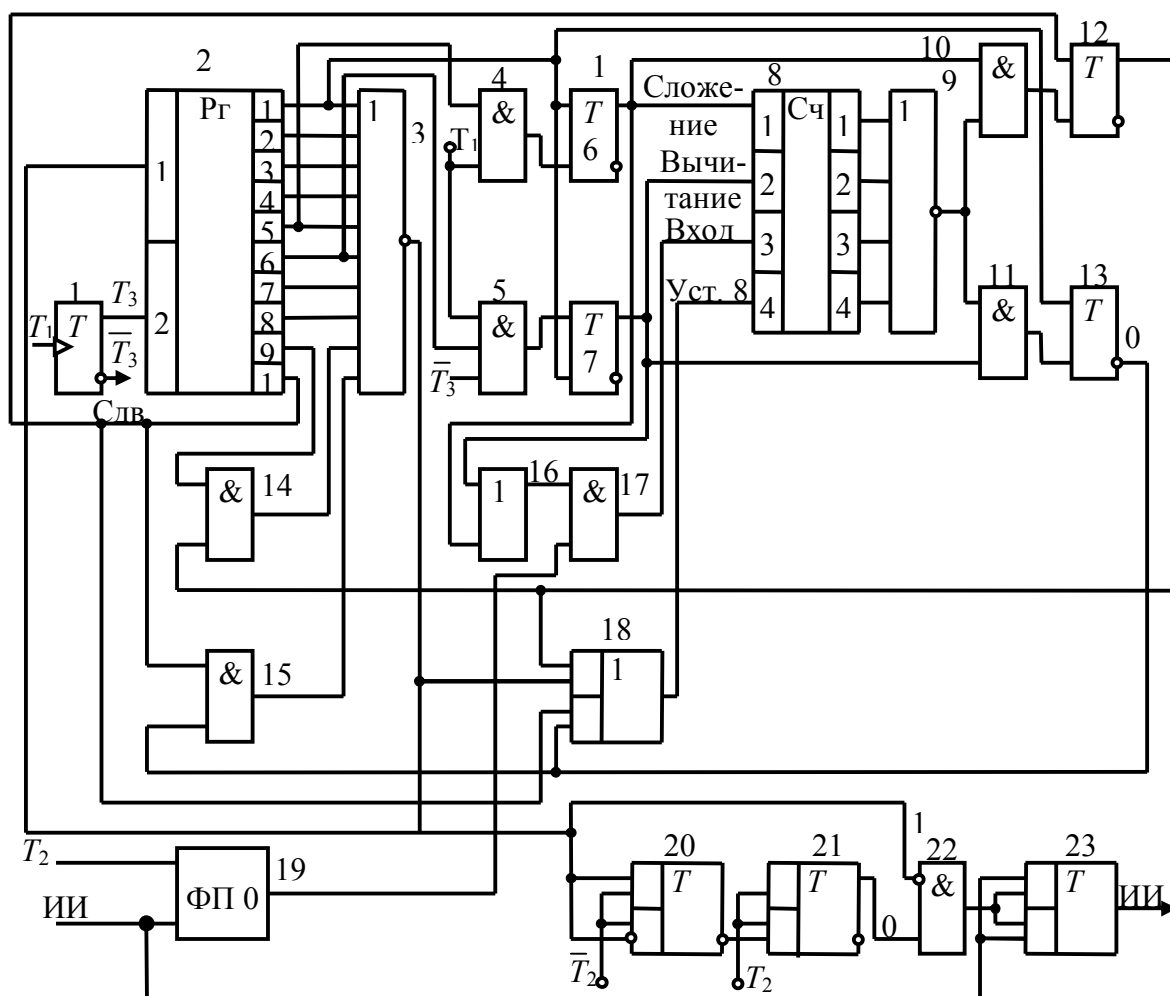


Рис. 9.15. Функциональная схема синхронизации по тактам

В устройстве имеется схема фиксации моментов перехода через нуль сигналов, приходящих из канала связи, – схема выделения фронтов. Фазовый дискриминатор определяет положение принятого импульса по отношению к тактовому импульсу приемника. Для характеристики фазовых соотношений импульсов различают зону отставания, зону опережения и зону синхронного приёма. Последняя расположена посередине тактового импульса приемной части. Коррекция фазы тактовых импульсов приемника необходима, если фронты принимаемых импульсов попадают в зону опережения или отставания. Для обеспечения достаточной помехоустойчивости синхронизации фазовое рассогласование интегрируется реверсивным счетчиком. Коррекция фазы производится только в том случае, если счетчиком зафиксировано определенное число импульсов рассогласования.

Кроме коррекции фазы в устройстве осуществляется поэлементный прием информационных сигналов методом стробирования.

Устройство состоит из следующих элементов: регистра 2, схемы регулировки цикла регистра (триггеры 12, 13; схемы И14, 15); схемы ввода 1 в регистр (схема ИЛИ – НЕЗ), фазового дискриминатора (схема ИЛИ16; схемы

И 4, 5, 17; триггеры 6, 7), интегратора (счетчик 8; схема ИЛИ – НЕ9), схемы фиксации перехода через нуль информационных сигналов (ФПО 19), схемы стробирования импульсов (схемы И22; триггеры 20, 21, 23).

От генератора тактовых сигналов постоянно поступают тактовые импульсы T_1 со скважностью два и частотой $F = 20V$ (где V – частота модуляции). Счетным триггером 1 формируются тактовые сигналы T_3 и $\bar{T}_3$ с частотой $F_1 = 10V$. Сигналы T_3 продвигают 1 в регистре сдвига 2. Запись 1 в первую ячейку регистра сдвига осуществляется от схемы ИЛИ – НЕ 3. Сигнал на выходе этой схемы возникает при отсутствии 1 на всех ее входах. Изменение цикла регистра производится с помощью триггеров 12 и 13 и схем И14 и 15. В исходном состоянии (при отсутствии расхождения фаз) на схему И15 от триггера 13 поступает сигнал 0, исключая влияние десятой ячейки распределителя, на работу схемы ИЛИ – НЕ3, а на схему И14 поступает сигнал 1 от триггера 12, в результате чего выход девятой ячейки регистра оказывается подключенным ко входу схемы ИЛИ – НЕ3.

Таким образом, если коррекция фазы не производится, сигнал на выходе схемы ИЛИ – НЕ 3 появляется одновременно с сигналом от десятой ячейки регистра, а запись 1 в первую ячейку происходит следующим сигналом T_3 . В этом случае цикл регистра равен 10 тактам T_3 . Временная диаграмма для этого случая приведена на рис. 9.16, а.

Из информационных импульсов ИИ с помощью тактовых импульсов высокой частоты (96 кГц) T_2 схема ФП 0 формирует короткие импульсы длительностью от одного до двух периодов сигнала T_2 , совпадающие по фронту с моментом перехода через нуль информационных посылок.

Фазовый дискриминатор постоянно контролирует попадание этих сигналов в одну из трех зон. В зоне отставания триггер 6 находится в состоянии 1. Он взводится по сигналу первой ячейки регистра 2 и сбрасывается от схемы И 4 по совпадению сигнала T_1 и импульса от 5-й ячейки регистра. В зоне опережения триггер 7 находится в состоянии 1. Он взводится от схемы И5 по совпадению тактовых сигналов T_1 , $\bar{T}_3$ и импульса от 6-й ячейки регистра. Таким образом, ширина зоны синхронного приема составляет $1,5T_3$, а зоны отставания и опережения – по $4,25T_3$. Выходные сигналы ИЛИ–НЕ3 по частоте соответствуют частоте модуляции сигналов в линии связи V .

Сигналы от триггеров 6 и 7 через схему ИЛИ16 поступают на схему И17, разрешая прохождение импульсов от ФП 0 на четырехразрядный двоичный счетчик 8. Если 1 поступает от триггера 6, счетчик 8 работает на сложение. Если же 1 поступает от триггера 7, счетчик 8 работает на вычитание, причем за один цикл регистра в счетчик может быть введен только 1 импульс.

Предварительно в счетчик 8 записывается число 8. Схема ИЛИ–НЕ9 срабатывает при нулевом состоянии счетчика, т. е. когда разность числа «отстающих и опережающих» импульсов будет равна 8. Если зафиксировано отставание, то сигнал на выходе схемы ИЛИ–НЕ9 совпадает с сигналом 1 на выходе триггера 6, в результате чего сигналом от схемы И10 сбрасывается триггер 12. Сигнал 0

с этого триггера поступает на схему И14, при этом запрещается поступление сигнала с 9-й ячейки регистра на схему ИЛИ-НЕ3. Схема ИЛИ-НЕ3 срабатывает одновременно с 9-й ячейкой регистра, т. е. цикл регистра уменьшается на один такт T_3 (рис. 9.16, б).

Если зафиксировано опережение, то сигнал на выходе схемы ИЛИ-НЕ9 совпадает с сигналом 1 от триггера 7, и через схему И11 сбрасывается триггер 13. Схемой И15 подключается к входу схемы ИЛИ-НЕ3 выход 10-й ячейки регистра. Цикл регистра удлиняется на один такт T_3 (рис. 9.16, в).

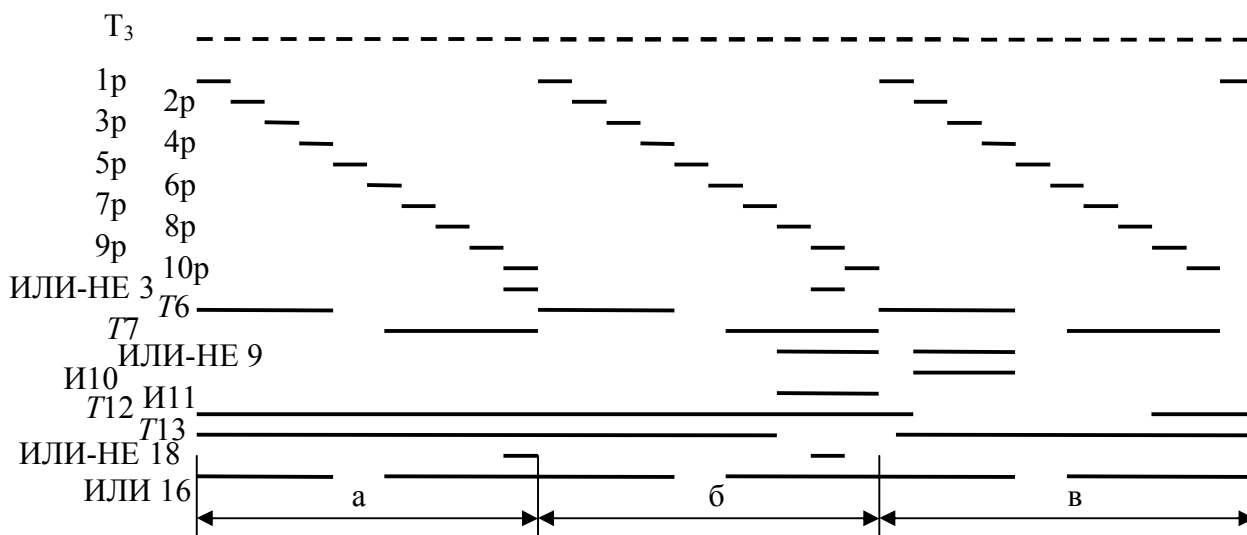


Рис. 9.16. Временная диаграмма работы устройства синхронизации:
а – коррекция отсутствует; б – коррекция при отставании; в – коррекция при опережении

Выходной сигнал схемы ИЛИ-НЕ 3 через схему И-ИЛИ 18 вводит число 8 в реверсивный счетчик. Установка триггера 12 в исходное состояние производится от 10-й ячейки регистра, а триггера 13 – по сигналу 1-й ячейки регистра (рис. 9.15).

Поэлементный прием методом стробирования осуществляется с помощью триггера 23, на вход которого поступают информационные сигналы ИИИ и короткие стробирующие импульсы длительностью полпериода T_2 . Стробирующие импульсы формируются с помощью триггеров 20 и 21 и схемы И22.

На рис. 9.17 приведена схема алгоритма, поясняющая функционирование устройства.

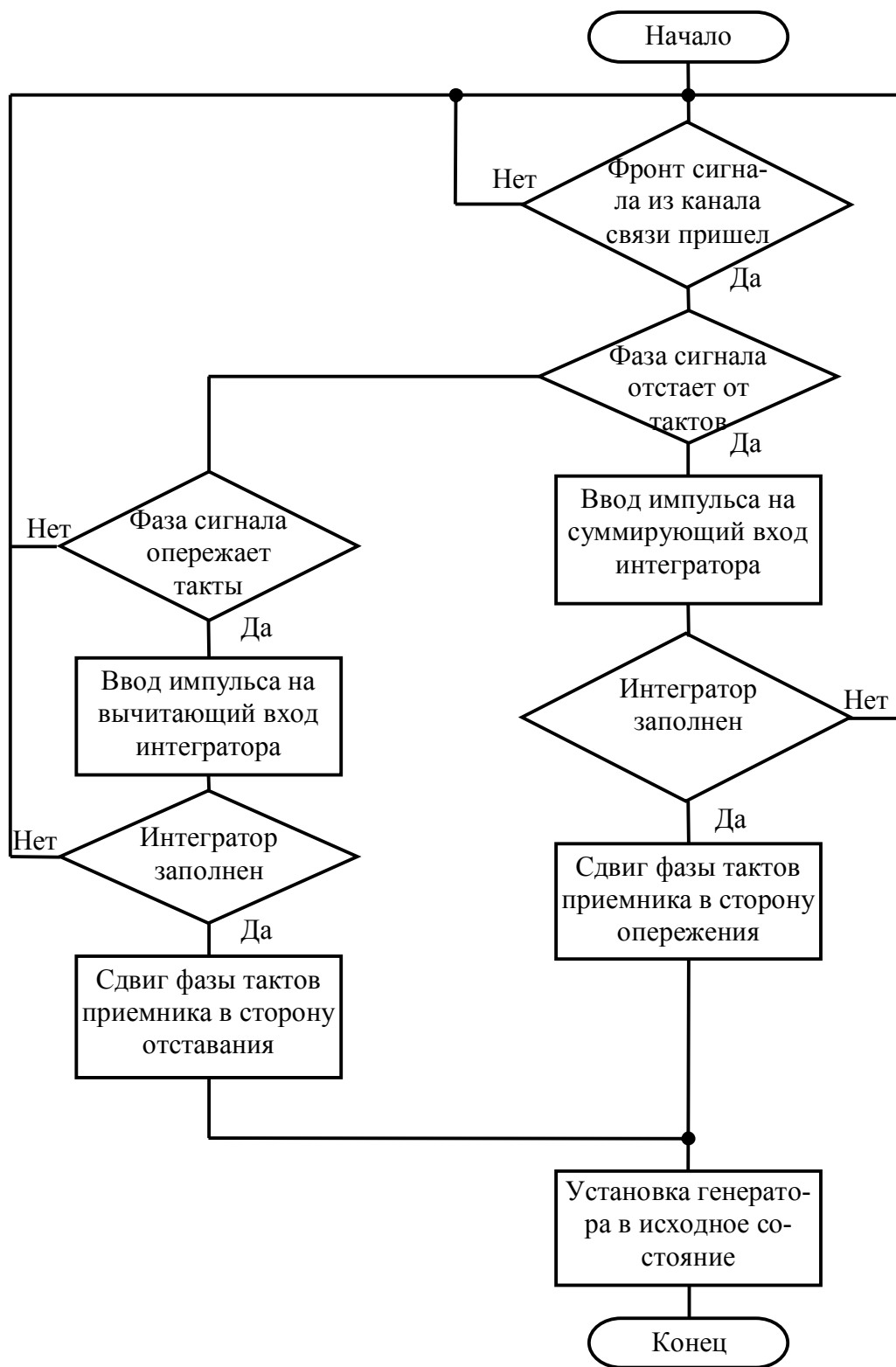


Рис. 9.17. Схема алгоритма функционирования устройства синхронизации

9.1.4. Одноконтурная и двухконтурные схемы выделения синхросигнала. В схеме, приведенной на рис. 9.18, выделение синхросигнала и данных из линии основано на использовании контура фазовой автоподстройки частоты.

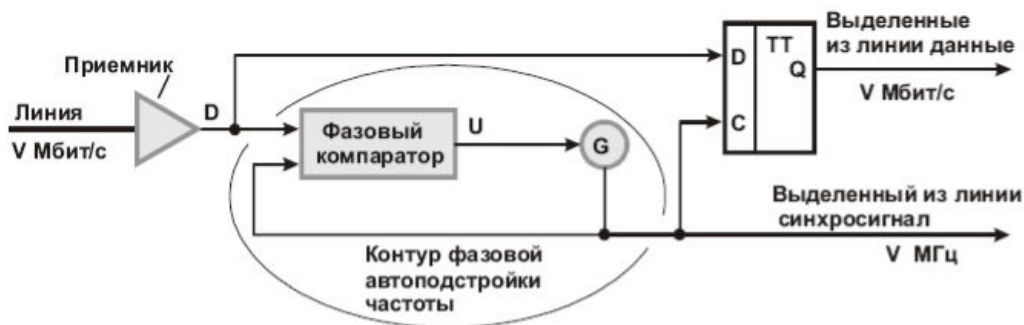


Рис. 9.18. Схема, поясняющая принцип выделения синхросигнала и данных с помощью контура фазовой автоподстройки частоты

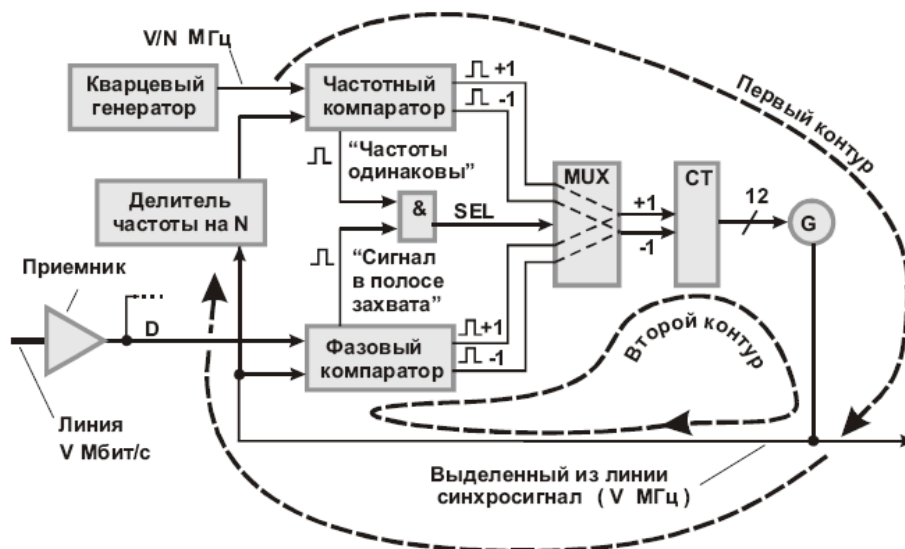
В установившемся режиме генератор G , управляемый напряжением U , формирует непрерывную последовательность синхроимпульсов. Их фронты (например, отрицательные) с высокой точностью совпадают во времени с моментами изменения сигнала D на выходе приемника. Моменты изменений в общем случае нерегулярны и зависят от передаваемой последовательности битов. При правильном выборе способа кодирования линейного сигнала существует гарантия хотя бы одного изменения сигнала D в течение некоторого заданного промежутка времени. Например, при использовании кода $B6ZS$ (см. п. 3.1.6) гарантируется наличие хотя бы одного импульса (двух изменений сигнала) в течение любых пяти последовательных битовых интервалов.

Фазовый компаратор оценивает степень совпадения фронтов сигналов на своих входах и в зависимости от результата формирует некоторое управляющее напряжение U . На пример, при хорошем совпадении фронтов, когда коррекции не требуется, $U = 2,5$ В. При необходимости ускорить или замедлить темп выработки синхроимпульсов напряжение повышается или снижается. Диапазон изменения напряжения может составлять (0,5 - 4,5) В. Фазовый компаратор содержит фильтр низких частот и поэтому обладает достаточной инерционностью, чтобы не реагировать на возможные мешающие факторы: «дрожание» фазы входного сигнала, импульсные помехи в линии и т. п.

В идеальном случае положительные фронты выделенного из линии синхросигнала соответствуют центрам битовых интервалов, поэтому для выделения данных используется D -триггер.

Контур фазовой автоподстройки частоты представляет собой достаточно «тонкий инструмент». Этот контур должен иметь очень узкую полосу захвата – порядка долей герца. Иными словами, для вхождения устройства в синхронизм с линейным сигналом нужно, чтобы исходная частота генератора G почти не отличалась от входной частоты. В противном случае на входах фазового компаратора будут наблюдаться случайные фазовые соотношения между сигналами, что не по-

зволит выбрать нужное направление коррекции частоты. Из этого следует, что генератор G должен быть высокостабилизированным, но это противоречит требованию его хорошей управляемости со стороны фазового компаратора.



Первый контур предназначен для предварительной настройки генератора G на номинальную частоту синхросигнала. После завершения такой настройки вместо первого контура включается второй, аналогичный рассмотренному ранее (см. рис. 9.18).

Кварцевый генератор формирует на первом входе частотного компаратора сигнал частотой V/N , где V - скорость передачи данных по линии, $N > 2$ - целое число. Сигнал той же частоты поступает на второй вход этого компаратора. Снижение сравниваемых частот позволяет упростить схему частотного компаратора и снизить потребляемую им мощность.

на входах, логический элемент И получает от обоих компараторов сигналы лог. 0. Так как $SEL = 0$, мультиплексор MUX транслирует на входы счетчика CT сигналы с выходов частотного компаратора.

Частотный компаратор определяет знак и величину частотной ошибки и формирует последовательность импульсов на одном из выходов. Код в счетчике CT изменяется в направлении уменьшения ошибки. В конечном счете ошибка уменьшается до допустимых пределов, частотный компаратор формирует сигнал «Частоты одинаковы». Одновременно с этим или с небольшой задержкой фазовый компаратор формирует признак «Сигнал в полосе захвата», означающий, что фазовый компаратор выявил закономерность между моментами изменения сигналов на обоих входах и поэтому способен корректировать ее в нужном направлении.

В результате совпадения указанных условий формируется сигнал $SEL = 1$, мультиплексор переключается на трансляцию сигналов с фазового компаратора. Далее осуществляется точная подстройка генератора, при которой выходной сигнал привязывается к входному по фазе. Это позволяет выделять данные с помощью D -триггера, как было показано ранее (см. рис. 9.18).

При нормальной работе устройства, как уже отмечалось, включен второй контур управления генератором G . Но при потере входного сигнала в линии частота нестабилизированного генератора G , оставшегося без управления, начинает непрерывный дрейф в ту или иную сторону. Так как во втором контуре нет опорной частоты, фазовый компаратор не посылает корректирующие импульсы в счетчик CT . Поэтому если, например, в результате «замирания» сигнала в линии потеряна группа из 70 битов, то дрейф приведет к выходу частоты генератора G из полосы захвата фазового компаратора. Тогда после возобновления правильной последовательности сигналов в линии вся описанная ранее процедура грубой и точной настройки генератора G будет повторена, что связано с неоправданно большими затратами времени.

Схема, приведенная на рис. 9.20, обладает лучшей устойчивостью к «замираниям» сигнала в линии. Это связано с тем, что нестабилизированный генератор $G2$ контура фазовой автоподстройки частоты ни при каких обстоятельствах не остается без управления.

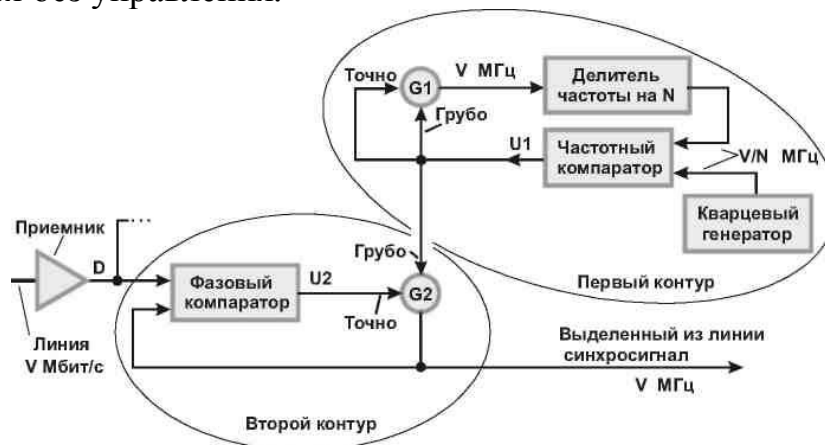


Рис. 9.20. Двухконтурная схема выделения синхросигнала из линии - второй вариант

Схема построена на основе двух нестабилизированных генераторов $G1$ и $G2$. Эти генераторы размещены в одном кристалле интегральной схемы и имеют одинаковую топологию. Поэтому они обладают очень близкими электрическими параметрами, что существенно для данного решения. Генератор GL ($G2$) содержит аналоговые входы грубой и точной подстройки частоты. При нормальной работе устройства регулирующие напряжения $U1$ и $U2$ находятся примерно в середине диапазона регулировки (например близки 2,5 В при диапазоне (0,5 - 4,5) В). Входы грубой и точной подстройки генератора $G1$ объединены.

В первом, автономном контуре управления отслеживается частота кварцевого генератора, так что генератор $G1$ формирует стабилизированную частоту V , близкую скорости передачи данных в линии. Поскольку генераторы $G1$ и $G2$ имеют практически одинаковые электрические параметры, генератор $G2$ также настроен на эту частоту по входу грубой подстройки. В результате независимо от уровня напряжения $U2$ частота сигнала на выходе генератора $G2$ отличается от нужной всего лишь на доли герца и в любой ситуации не выходит за пределы полосы захвата фазового компаратора!

Таким образом, фазовый компаратор всегда готов к работе, даже после длительных «замираний» сигнала в линии. Исключен дрейф частоты в отсутствие сигнала в линии, уменьшено время вхождения в синхронизм после обнаружения этого сигнала.

Генераторы $G1$ и $G2$ выполнены на МДП-транзисторах по схеме симметричного мультивибратора (рис. 9.21).

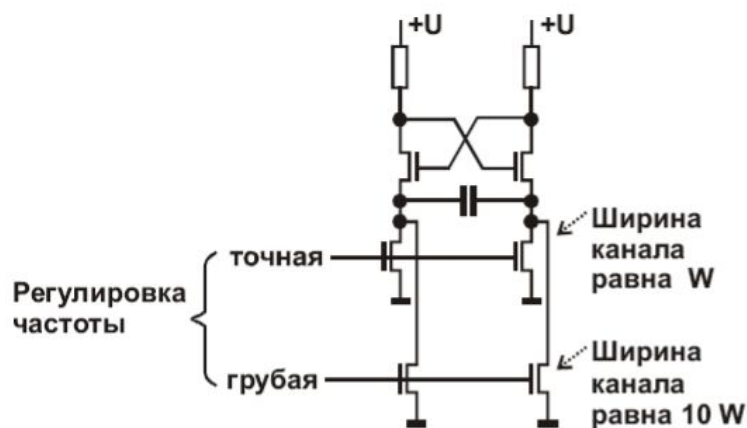


Рис. 9.21. Схема генератора

Четыре нижних транзистора используются в качестве двух сдвоенных переменных резисторов, с помощью которых можно регулировать частоту сигнала. Ширина канала транзисторов грубой регулировки частоты (нижняя пара) в 10 раз превышает ширину канала транзисторов точной регулировки. Поэтому сопротивления каналов и соответствующие диапазоны регулировки частоты отличаются примерно в 10 раз.

9.1.5. Схема выделения синхросигнала с компенсацией нагрузочной емкости. Выделенный из линии синхросигнал может поступать к нескольким

микросхемам, размещенным на одной или нескольких печатных платах устройства, например мультиплексора. При этом паразитная емкость проводника, несущего синхросигнал, и входные емкости элементов – приемников этого сигнала увеличивают длительности его фронтов. Это эквивалентно задержке синхросигнала, нарушающей его точную привязку к данным, поэтому задержка должна быть скомпенсирована. На рис. 9.22 приведен пример схемы компенсации задержки синхросигнала [18].

Схема содержит генератор G с фазовой автоподстройкой частоты, микропроцессор, группу электронных ключей SW , набор компенсирующих конденсаторов с емкостями C , $2C$, $4C$, $8C$, два усилителя сигнала $SYNC$ и два фазовых компаратора $EQ1$ и $EQ2$. Усилители выполнены в одной интегральной схеме и поэтому имеют практически одинаковые электрические характеристики. Каждый ключ управляется соответствующим сигналом из группы сигналов Q , поступающих из микропроцессора. Входные данные $DATA$ поступают из линии и в неявном виде содержат синхросигнал. В цепи передачи выделенного из линии синхросигнала CLK присутствует заранее не известная паразитная емкость CX .

В зависимости от комбинации управляющих сигналов Q к цепи передачи сигнала CL подключается та или иная подгруппа компенсирующих конденсаторов. В установившемся режиме их суммарная емкость с помощью микропроцессора подбирается примерно равной паразитной емкости CX . При $Q = 0000$ все конденсаторы отключены, при $Q = 1111$ все конденсаторы подключены и их суммарная емкость составляет $15C$, где C – емкость, определяющая шаг подбора. Максимальная компенсирующая емкость ($15C$) должна превышать максимальную паразитную емкость CX .

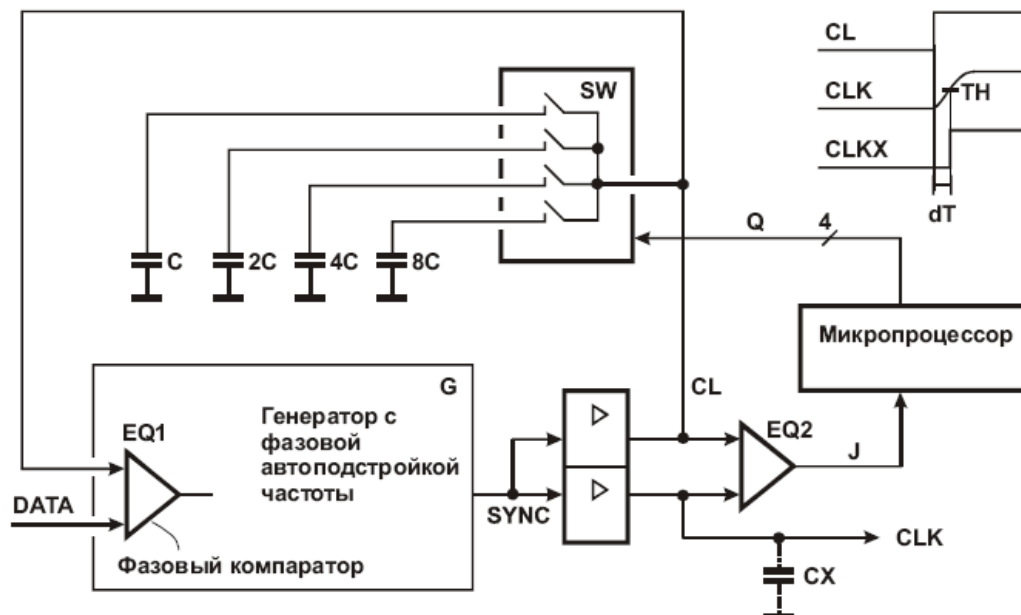


Рис. 9.22. Схема компенсации задержки выделенного синхросигнала

Предположим, что в исходном состоянии $Q = 0$, компенсирующие конденсаторы изолированы от цепи CL передачи сигнала обратной связи генератора с

фазовой автоподстройкой частоты. Предположим также, что паразитная емкость CX оказывает заметное влияние на длительность фронтов сигнала CLK . В этой ситуации сигналы CL и $DATA$ находятся в правильных фазовых соотношениях, которые отслеживаются компаратором $EQ1$ генератора G . Однако сигнал CLK имеет пологий фронт, как показано на рисунке. Порог ТН срабатывания приемников этого сигнала лежит примерно в середине между уровнями лог. 0 и 1. Поэтому момент срабатывания приемников смещен относительно положительного фронта сигнала CL на время DT . Иными словами, сигнал CLK эквивалентен некоторому сигналу $CLKX$, полученному из сигнала CL путем его сдвига на время DT .

Задача заключается в минимизации этого времени. Фазовый компаратор $EQ2$ реагирует на отрицательный сдвиг фазы сигнала CL относительно сигнала CLK выдачей на вход микропроцессора признака $J = 0$. Микропроцессор следит за состоянием этого признака и последовательно наращивает код Q , что приводит к последовательному увеличению компенсирующей емкости. Фронт сигнала CL начинает приближаться по форме к фронту сигнала CLK . В некоторый момент фазовое соотношение между этими сигналами изменяет знак, признак J переходит в состояние лог. 1. Обнаружив этот переход, микропроцессор фиксирует код Q на достигнутом уровне.

В дальнейшем микропроцессор периодически, например один раз в секунду, подбирает величину компенсирующей емкости таким образом, чтобы компаратор $EQ2$ находился на грани срабатывания.

Рассмотренное решение позволяет автоматически отслеживать изменения паразитной емкости CX при изменении температуры, влажности, при отключении части нагрузки и т. п., что обеспечивает надежную привязку выделенного из линии синхросигнала к данным.

9.1.6. Распознавание синхросигнала в искаженном входном сигнале с помощью симметрирующего коррелятора. Корреляторы обычно применяют в системах распознавания образов для сопоставления и оценки сходства эталонного и искомого изображений. Далее рассмотрено применение коррелятора для решения задачи распознавания синхросигнала в искаженном входном сигнале. Точнее, задача заключается в отыскании в проходящем битовом потоке «очертаний» синхросигнала с заданными параметрами и привязки восстановленного синхросигнала к центрам битовых интервалов.

В примере, приведенном на рис. 9.23, коррелятор входит в состав схемы распознавания синхросигнала и содержит 16-разрядные регистры $RG1$, $RG2$, компараторы $EQ0 - EQ15$, $INEQ$ и сумматор ADD . Входные данные $DATA$ «мелкими шагами» (составляющими доли битового интервала) проходят через сдвиговый регистр $RG1$. В регистре $RG2$ зафиксирован эталонный код для сравнения с кодом в регистре $RG1$. Поразрядное сравнение кодов выполняется компараторами $EQ0 - EQ15$. Компаратор формирует сигнал лог. 1 при совпадении битов на его входах. Сумматор ADD подсчитывает число совпадений битов в регистрах и выдает результат SUM на компаратор для сравнения с некоторым порогом TH . Если $SUM > TH$, то $Y = 1$, в противном случае $Y = 0$. Сигнал $Y = 1$

подтверждает близость кодов в регистрах $RG1$ и $RG2$ и представляет собой метку времени для привязки к ней искомого синхросигнала. Сигнал Y с выхода коррелятора поступает на вход генератора PLL с фазовой автоподстройкой частоты, который восстанавливает синхросигнал CL .

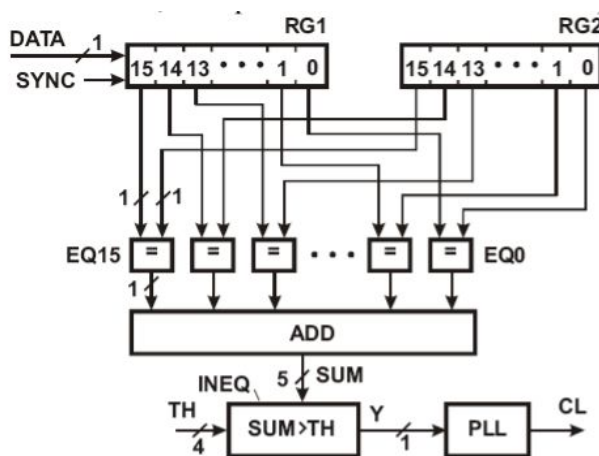


Рис. 9.23. Схема распознавания синхросигнала в потоке данных - первый вариант

Рассмотрим процесс распознавания синхросигнала более детально. На вход коррелятора поступают два сигнала: $DATA$ и $SYNC$. Первый принят из линии; он в неявном виде содержит синхросигнал CL , который следует восстановить. Второй формируется местным автономным генератором; его частота в восемь раз превышает частоту синхросигнала CL . Для большей наглядности описания работы коррелятора предположим, что сигнал $DATA$ представлен кодом NRZ (см. 3.1.2) вида ...010101... . Каждый бит этого кода представлен на рис. 9.24, а восемью одноименными отсчетами (цепочками из лог. 0 или 1).

Сигнал $DATA$ проходит через сдвиговый регистр $RG1$ с шагом, равным $1/8$ части битового интервала кода NRZ . Такое продвижение удобно рассматривать как просмотр сигнала $DATA$ через движущееся окно. Содержимое окна сравнивается с эталоном 0000111111110000 из регистра $RG2$. Этот эталон рассчитан на появление в коде NRZ комбинации битов «010». Задача заключается в фиксации момента наибольшей схожести кодов в регистрах $RG1$ и $RG2$. Как показано в первой строке табл. 9.1, в некотором такте (п-8) сигнала $SYNC$ код в регистре $RG1$ противоположен эталонному, поэтому нет ни одного совпадения битов, следовательно $SUM = 0$.



Рис. 9.24. Временные диаграммы, поясняющие принцип действия коррелятора:
а - сигнал *DATA* без искажений; б - сигнал *DATA* с искажениями;
в - восстановленный синхросигнал

Таблица 9.1

Процесс формирования сигнала *SUM* при обработке сигнала *DATA*, показанного на рис. 9.24, а, схемой, приведенной на рис. 9.23

Номер такта	Код в регистре RG1	Значение сигнала SUM
n-8	1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1	0
n-7	1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1	2
n-6	1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1	4
n-5	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1	6
n-4	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0	8
n-3	0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0	10
n-2	0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0	12
n-1	0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0	14
n	0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0	16
n+1	0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0	14
n+2	0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0	12
n+3	0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0	10
n+4	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1	8

В следующем такте (n-7) совпадения наблюдаются в разрядах 3 и 11, поэтому $SUM = 2$. По мере продвижения кода в регистре *RG1* наблюдается увеличение числа совпадений, и в такте с номером *n* коды полностью совпадают, сигнал *SUM* достигает максимального значения, равного 16. В последующих тактах код в регистре *RG1* все более отличается от эталонного, сигнал *SUM* возвращается к нулевому значению и т. д. В отсутствие искажений сигнала *DATA* порог *TH* можно выбрать равным 15. Тогда положительный фронт сигнала *Y* с хорошей точностью попадает в середину битового интервала, в котором передается единичный сигнал. Снижая порог, можно уменьшить требуемую степень

схожести кодов в регистрах $RG1$ и $RG2$, однако при этом снижается точность попадания сигнала Y в середину битового интервала.

Генератор PLL с фазовой автоподстройкой частоты формирует сигнал CL так, чтобы его положительные фронты совпадали с положительными фронтами сигнала Y или запаздывали на заданный интервал, что необходимо при снижении порога ТН. Сигнал Y в общем случае формируется с нерегулярным периодом, зависящим от характера передаваемых данных. Передающая сторона должна обеспечивать гарантированное появление хотя бы одной эталонной комбинации (сочетания битов «010» в коде NRZ) на протяжении, например, любых двадцати битовых интервалов. Напомним, что в данном примере рассматривается последовательность ...010101..., обеспечивающая формирование сигнала Y в каждом втором битовом интервале.

Выраженный максимум сигнала SUM , однако, быстро размывается при типичном искажении входного сигнала $DATA$, показанном на рис. 9.24, б. Такое искажение связано, например, со смещением порога срабатывания приемника сигнала из линии. Действительно, импульсы в линии имеют не прямоугольную, а трапецеидальную форму. Если порог срабатывания приемника смещен в сторону повышения, то сигналы лог. 1 окажутся более короткими, чем сигналы лог. 0 и наоборот. Как следует из рисунка, сигнал лог. 1 представлен четырьмя (а не восемью) единичными отсчетами в регистре $RG1$, хотя передается тот же самый, что и ранее, код NRZ : ...010101... . Работа коррелятора в этих условиях поясняется табл. 9.2.

Таблица 9.2

Процесс формирования сигнала SUM при обработке сигнала $DATA$, показанного на рис. 9.24,б, схемой, приведенной на рис. 9.23

Номер такта	Код в регистре $RG1$	Значение сигнала SUM
n-8	1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	2
n-7	1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1	2
n-6	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1	2
n-5	0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1	5
n-4	0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0	8
n-3	0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0	10
n-2	0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0	12
n-1	0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0	12
n	0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0	12
n+1	0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0	12
n+2	0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0	12
n+3	0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0	10
n+4	0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0	8

Как следует из таблицы, сигнал SUM имеет размытый максимум, равный 12; он наблюдается на протяжении пяти соседних тактов. Это не позволяет с хорошей точностью выделить момент, соответствующий середине битового ин-

278



В схеме (рис. 9.25), в отличие от предыдущей, отсутствует регистр для хранения эталона (!). Последовательность отсчетов входного сигнала продвигается через регистр RG и анализируется логическим блоком F , компараторами $EQ1 - EQ6$ и сумматором ADD . Принцип действия коррелятора поясняется рис. 9.27.

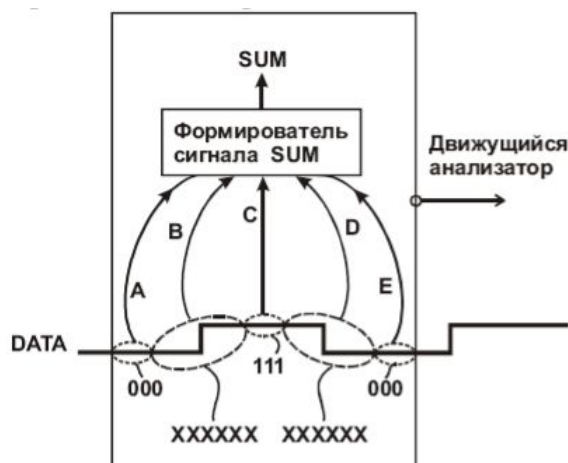


Рис. 9.27. Схема, поясняющая принцип действия коррелятора (см. рис. 9.25)

При формировании сигнала *SUM* анализируются группы отсчетов *A*, *B*, *C*, *D*, *E*. Прежде всего проверяется одновременное выполнение трех условий: $A = 000$, $C = 111$, $E = 000$ или $A = 111$, $B = 000$, $C = 111$. Если условия выполнены, то это означает, что положение движущегося анализатора примерно соответствует показанному на рисунке или аналогичному, смещенному на один битовый интервал сигнала *NRZ*. В этом случае анализ продолжается. Теперь сопоставляются группы отсчетов *B* и *D*, представленные на рисунке знаками «X». Определяется степень симметрии отсчетов в этих группах. Если сигнал *NRZ* не искажен (как показано на рисунке), то $B = 000111$, $D = 111000$. Сравнение этих групп с учетом зеркальной перестановки разрядов в одной из них показывает максимально возможное число совпадений, равное шести. Если хотя бы одно названных условий ($A = 000$, $C = 111$, $E = 000$ или $A = 111$, $B = 000$, $C = 111$) не выполнено, то анализ прекращается и сигнал *SUM* устанавливается в нуль.

Предположим, что длительность сигналов лог. 1 в коде *NRZ* чуть уменьшилась за счет соответствующего увеличения длительности сигналов лог. 0. Тогда получим, например, такое соотношение кодов *B* и *D*: $B = 000011$, $D = 110000$. Эти коды вновь оказались симметричными, несмотря на асимметрию сигнала *DATA*, т. е. разрешающая способность коррелятора не уменьшилась при обработке искаженного сигнала!

Рассмотрим работу коррелятора более подробно. Код *DATA* проходит через сдвиговый регистр *RG* под управлением синхросигнала *SYNC*, частота которого в девять раз превышает скорость передачи данных по линии. Группы разрядов *B* и *D* с помощью компараторов *EQ1* - *EQ6* поразрядно сравниваются в «зеркальном» порядке относительно центрального (десятого) разряда регистра: разряд 8 сравнивается с разрядом 12, разряд 7 - с разрядом 13 и т. д. Группы разрядов *A*, *C* и *E* анализируются логическим блоком *F*. Сигнал $EN = 1$ формируется при $A = E = 000$ и $C = 111$ или при $A = E = 111$ и $C = 000$. Сумматор *ADD* при $EN = 1$ подсчитывает число единичных сигналов на входах (все входы имеют равные единичные веса). Если $EN = 0$, то работа сумматора запрещена, на его выходе постоянно присутствует нулевой код.

Прохождение неискаженного и искаженного сигналов *DATA* через коррелятор поясняется табл. 9.3 и табл. 9.4.

Таблица 9.3

Процесс формирования сигнала *SUM* при обработке неискаженного сигнала *DATA* схемой, приведенной на рис. 9.25

Номер такта	Код в регистре RG	Значение сигнала SUM	Значение сигнала EN
n-3	1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1	0	1
n-2	1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1	2	1
n-1	1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1	4	1
n	1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1	6	1
n+1	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1	4	1
n+2	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1	2	1
n+3	1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1	0	1

Таблица 9.4

Процесс формирования сигнала *SUM* при обработке искаженного сигнала *DATA* схемой, приведенной на рис. 9.25

Номер такта	Код в регистре RG	Значение сигнала SUM	Значение сигнала EN
n-3	1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1	0	0
n-2	1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	0	0
n-1	1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	4	1
n	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1	6	1
n+1	1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1	4	1
n+2	1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1	0	0
n+3	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1	0	0

Из сопоставления таблиц следует, что, несмотря на внесенные искажения, связанные с нарушением симметрии сигналов лог. 0 и 1 в коде *NRZ*, коррелятор по-прежнему формирует хорошо выраженный сигнал опознания центра битового интервала. При этом распознаются две (а не одна) комбинации битов в коде *NRZ*: 010 и 101. Это увеличивает частоту следования импульсов привязки на входе генератора с фазовой автоподстройкой частоты и способствует повышению качества восстановленного синхросигнала.

9.2. Синхронизация на уровне распознавания и обработки кадров и ячеек (кадровая синхронизация)

Как уже отмечалось, передаваемые по транспортным системам данные объединяются в логически законченные структурные единицы: кадры, пакеты, ячейки и т. п. Поэтому приемник данных должен распознавать не только передаваемые биты, но и построенные из них кадры или иные информационные посылки. В этом смысле можно говорить о синхронизации приемника с передатчиком на уровне передачи таких посылок.

Устройства кадровой синхронизации должны удовлетворять ряду требований: быстро входить в синхронизм при первоначальном включении и возобновлении работы после потери синхронизации; иметь высокую помехоустойчивость; незначительно снижать пропускную способность из-за необходимости передавать флаги.

Обычно кадр начинается с флага - уникальной комбинации битов (например 01111110), которая не встречается в его оставшейся части. Чтобы достичь такой уникальности, оставшаяся часть кадра (область данных) перед выдачей в линию анализируется и при необходимости «разбавляется» вставкой служебных нулевых битов в имеющиеся длинные цепочки единиц (эта процедура называется битстаффингом). В данном примере флаг содержит шесть единиц; поэтому после каждых пяти единиц, следующих вплотную друг за другом в области данных, всегда (независимо от значения последующего бита) вставляется нулевой (служебный) бит. Тем самым исключается возможность последующего обнаружения удаленным устройством группы из шести или более следующих друг за другом единиц в области данных, что могло бы привести к ложному опознанию флага. Флаг при передаче не подвергается битстаффингу и поэтому сохраняет уникальность, что позволяет распознать его удаленным приемником.

После получения кадра, переданного по линии связи, и обнаружения начала области данных выполняется операция, обратная битстаффингу. Нулевой бит, следующий за каждой группой из пяти единиц в области данных, справедливо расценивается приемником как служебный и безусловно вычеркивается. Поэтому, в частности, длинные цепочки единиц смыкаются, полученное поле данных кадра обретает первоначальный вид. Из-за применения битстаффинга длина передаваемых по линии кадров непостоянна: например, кадр, содержащий в поле данных «сплошные нули», передается по линии без изменения, а кадр со «сплошными единицами» в поле данных требует значительного разбавления нулями. (При постоянной длине кадра непостоянным окажется число размещенных в нем байтов данных.) Таким образом, избыточность потока передаваемых по линии данных обусловлена как применением многоразрядного флага, так и битстаффингом.

В начале этой главы (п. 9.2.1) рассмотрены схемы распознавания многоразрядных флагов в битовом и байтовом потоках данных. Далее (п. 9.2.2, 9.2.5) приведены схемы распознавания одноразрядных и раздробленных флагов, а также методы определения границ ячеек без использования флаговых битов

(п. 9.2.6). В решениях по п. 9.2.7 - 9.2.9 учитываются возможные искажения данных из-за помех в линии. Способ передачи данных по п. 9.2.10 позволяет повысить эффективность использования избыточных битов синхронизации. В конце главы (п. 9.2.11) приведен способ уменьшения числа операций, связанных с распознаванием флагов начала кадров.

Начнем с относительно простых решений.

9.2.1. Распознавание многоразрядного флага в битовых потоках данных. Схема, приведенная на рис. 9.28, а, предназначена для распознавания девятиразрядного флагового кода (флага) во входном потоке данных. В данном примере флаг выбран равным 000000001. Он предварительно загружается в девятиразрядный регистр *RG2*.

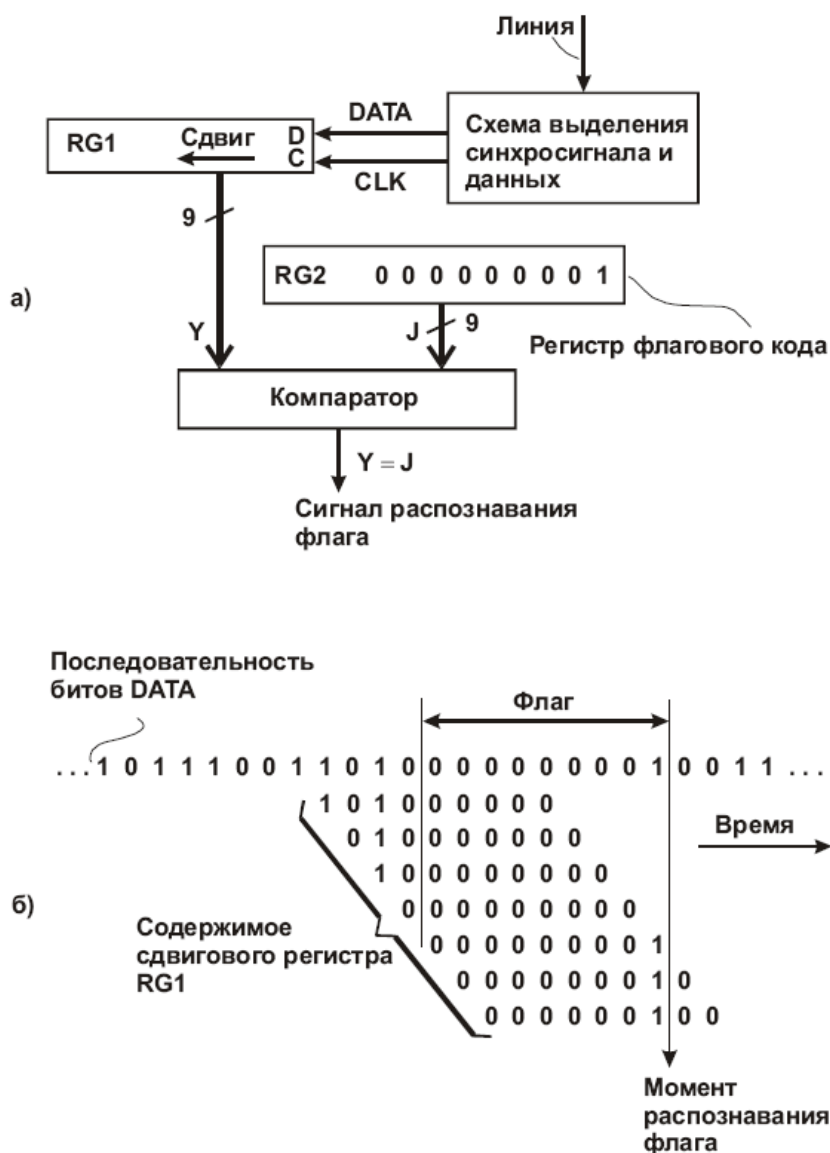


Рис. 9.28. Распознавание флага в битовом потоке данных

а - схема на основе многоразрядного компаратора;

б - прохождение кода через сдвиговый регистр *RG1*

Из поступающего по линии сигнала выделяются синхросигнал *CLK* и данные *DATA* (код *NRZ*). Под действием синхросигнала данные загружаются в

сдвиговый регистр $RG1$ и последовательно продвигаются в нем, как показано на рис. 9.28, б. При совпадении кодов в регистрах $RG1$ и $RG2$ срабатывает компаратор, на его выходе формируется сигнал распознавания флага. Длительность этого сигнала равна одному периоду (такту) синхросигнала CLK . В моменты изменения данных в регистре $RG1$ на выходе компаратора могут наблюдаться ложные срабатывания, поэтому для фиксации истинного значения выходного сигнала ($Y = J$) его следует принять на триггер (триггер на рисунке не показан). Синхронизация этого триггера может осуществляться задержанным (на время, превышающее задержку компаратора) сигналом CLK либо непосредственно этим сигналом. В последнем случае результат сравнения кодов будет запаздывать на один такт.

Несмотря на простоту данного решения, его применимость ограничена сравнительно низкоскоростными системами передачи данных. Это связано с тем, что переходные процессы на выходе компаратора должны закончиться к началу очередного битового интервала (такта), длительность которого при высоких скоростях передачи данных может составлять, например, 3 нс или менее. С увеличением разрядности флагового кода число логических элементов компаратора возрастает, увеличивается его задержка; поэтому, чтобы уложиться в битовый интервал, необходимо применение все более быстродействующей элементной базы.

9.2.2. Распознавание флага в байтовых потоках данных. В высокоскоростных системах передачи данных принятый последовательный код сразу же преобразуется в параллельный, чаще всего восьмиразрядный. Дальнейшая обработка данных проводится на уровне байтов (а не битов), что в восемь раз снижает требуемую тактовую частоту. А это, в свою очередь, позволяет уменьшить потребляемую устройством мощность, снижает требования к элементной базе, размещению проводников на печатной плате и т. д.

В схеме, приведенной на рис. 9.29, также как и в ранее рассмотренной (см. рис. 9.28, а), из поступающего по линии сигнала выделяются синхросигнал CLK и данные $DATA$ [18]. Под действием синхросигнала данные непрерывно загружаются в сдвиговый регистр $RG1$ и последовательно продвигаются в нем в направлении, указанном стрелкой. Выдвинутые за пределы регистра данные теряются.

Сигнал CLK проходит через делитель частоты на восемь (трехразрядный двоичный счетчик) и в виде сигнала $CLK/8$ поступает на входы синхронизации последовательно включенных параллельных регистров $RG2$ и $RG3$. Под действием этого сигнала в каждом восьмом такте сигнала CLK восьмиразрядный код из регистра $RG2$ переписывается в регистр $RG3$; одновременно с этим код из регистра $RG1$ переписывается в регистр $RG2$. Таким образом, битовая последовательность данных $DATA$ преобразуется в поток байтов, который проходит по цепи $RG2 - RG3$. Начальное состояние делителя частоты не определено, поэтому разметка исходной битовой последовательности на байты произвольна – истинные границы байтов (которые подразумевались при выдаче данных удаленным передатчиком) в общем случае не совпадают с границами байтов в регист-

рах $RG2$ и $RG3$. Байты $BYTE\_NEW$ (новый байт) и $BYTE\_OLD$ (старый байт) в регистрах $RG2$ и $RG3$ образуют «наблюдаемое» в данный момент 16-разрядное слово, в котором, возможно, содержится искомый девятиразрядный флаг. В момент очередного продвижения потока байтов (при поступлении сигнала $CLK/8$) наблюдаемое 16-разрядное слово смещается на один байт в сторону новых данных.

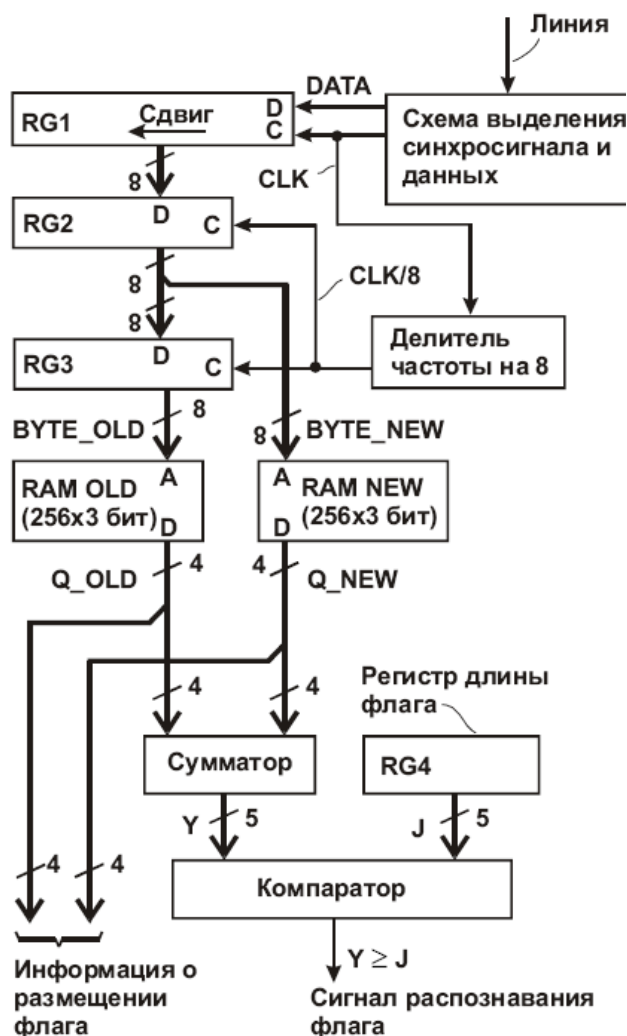


Рис. 9.29. Распознавание флага в байтовом потоке данных

В каждом такте сигнала $CLK/8$ байты $BYTE\_OLD$ и $BYTE\_NEW$ анализируются для обнаружения в них левого и правого фрагментов флага (000000001). Для анализа используются блоки памяти RAM_OLD и RAM_NEW с соответствующими кодировками. Байты $BYTE\_OLD$ и $BYTE\_NEW$ используются для адресации ячеек этих блоков. Число предполагаемых совпадений соответствующих байтов с фрагментами флага отображается четырехразрядными кодами данных Q_OLD и Q_NEW , которые считывают из блоков памяти. Каждый из этих кодов принадлежит диапазону 0 - 8. Эти коды арифметически суммируют-

ся, в результате формируется двоичное число Y , по которому можно судить о наличии (или отсутствии) флага в паре регистров $RG2 - RG3$.

Чтобы получить окончательный результат анализа, число Y сравнивается с числом J (отображающим разрядность флага), предварительно загруженным в регистр длины флага. В данном примере использован девятиразрядный флаг, поэтому $J = 9$. Сигнал распознавания флага формируется при $Y > J$. При этом информация о размещении флага в наблюдаемом 16-разрядном слове ($BYTE\_OLD - BYTE\_NEW$) содержится в коде Q_OLD или (и) Q_NEW (в данном примере» – в последнем).

Далее рассмотрены подробности этого решения.

Как показано в верхней части рис. 9.30, возможны восемь вариантов размещения флага в байтах $BYTE\_OLD - BYTE\_NEW$. Неоднозначность размещения связана с тем, что начальное состояние делителя частоты сигнала CLK не определено, и «нарезка» битовой последовательности на байты может начаться с произвольного места. В нижней части рисунка показаны схемы вычисления переменных Q_OLD и Q_NEW . Эти переменные отражают предполагаемое число битовых совпадений кодов с левым и правым фрагментами флага. Из верхней строки схемы вычисления переменной Q_OLD следует, что нулевой код в байте $BYTE\_OLD$ может (но не обязательно должен) соответствовать обнаружению восьми нулевых битов флага, поэтому $Q\_OLD = 8$. Следующая строка показывает возможность совпадения кода с флагом в семи разрядах и т. д. Аналогичные рассуждения применимы и к схеме вычисления переменной Q_NEW .

Условие распознавания флага, как уже отмечалось, состоит в том, что

$$Y = Q\_OLD + Q\_NEW > 9.$$

Чтобы убедиться в его правильности, рассмотрим все благоприятные (соответствующие обнаружению флага) сочетания фрагментов кодов, приведенных в нижней части рис. 9.30. Эти сочетания обозначим двузначными числами, в которых первое соответствует коду Q_OLD , а второе – коду Q_NEW . Так, например, число 88 соответствует сочетанию фрагментов кодов 0000 0000 ($BYTE\_OLD$) и 0000 0001 ($BYTE\_NEW$). Полный 16-разрядный код равен 0000 0000 0000 0001. В правой части этого кода присутствует искомый девятиразрядный флаг 000000001. Его местоположение однозначно определяется кодом $Q\_NEW = 8$. Еще один пример: числу 37 соответствует 16-разрядный код XXXX 1000 0000 001X, в котором содержится искомый флаг, и т. д. (Символы «X» соответствуют произвольным значениям битов.) Итак, благоприятные сочетания следующие:

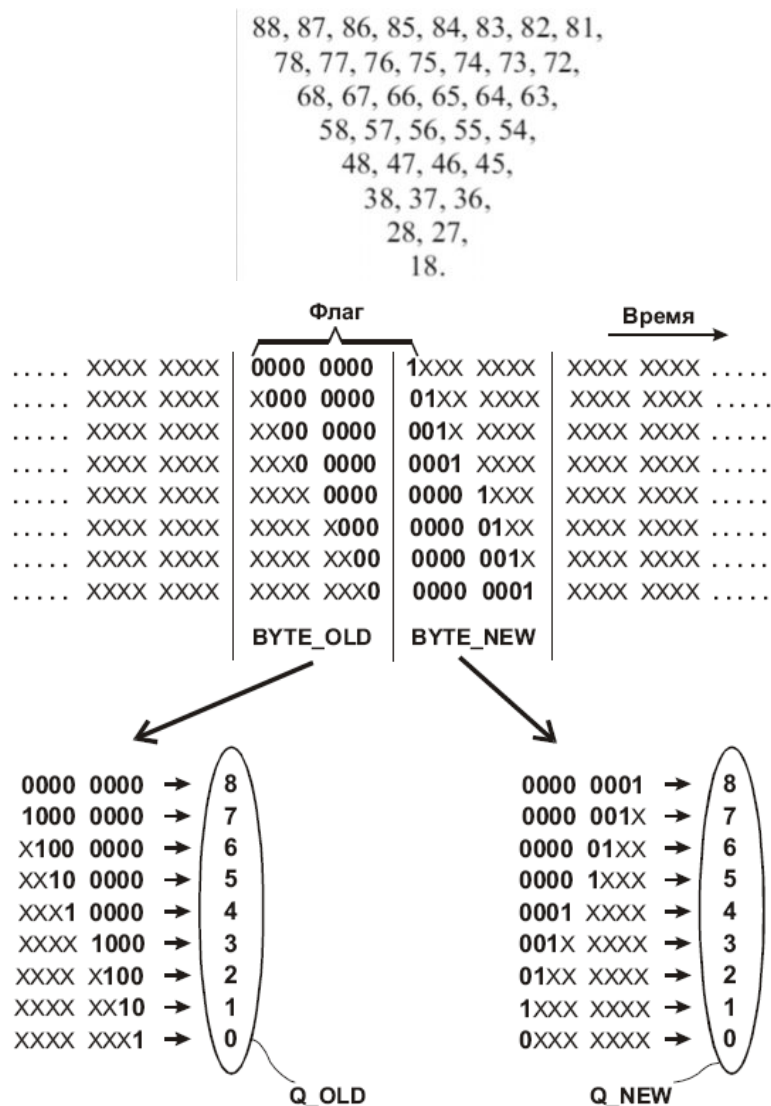


Рис. 9.30. Варианты размещения флага в байтах *BYTE_OLD* – *BYTE_NEW* и схемы вычисления переменных *Q_OLD* и *Q_NEW*. Символы «X» соответствуют произвольным значениям битов

Все эти сочетания характеризуются тем, что в каждом из них сумма первой и второй цифр превышает 8. Оставшиеся (неблагоприятные) сочетания в лучшем случае обеспечивают не более восьми совпадений с девятиразрядным флагом, что принципиально недостаточно для его распознавания.

Кодировка блоков памяти *RAM_OLD* и *RAM_NEW* представлена на рис. 9.31. Каждая ячейка блока памяти *RAM_OLD* (*RAM_NEW*) хранит четырехразрядное число *Q_OLD* (*Q_NEW*). Ячейки представлены на рисунке в виде массива чисел, принадлежащих диапазону от 0 до 8. Массив представлен матрицей из восьми строк и 32 столбцов (всего 256 ячеек). Адресация ячеек ведется слева направо, сверху вниз. Левая верхняя ячейка имеет адрес 00_{16} , правая нижняя – FF_{16} .

Поясним процесс распознавания флага на примере. Предположим, что после преобразования последовательного кода в параллельный в регистрах $RG3$ и $RG2$ сформированы коды $BYTE\_OLD = C0_{16}$ и $BYTE\_NEW = 10_{16}$:

$\dots \text{XXXX XXXX} \quad \underbrace{1100 \ 0000}_{C0_{16}} \quad \underbrace{0001 \ 0000}_{10_{16}} \quad \text{XXXX XXXX} \dots$

Как видим, в 16-разрядном коде присутствует флаг (000000001); пять его нулевых битов размещены в байте $BYTE\_OLD$, а оставшиеся четыре бита (0001) – в байте $BYTE\_NEW$. По адресу $C0_{16}$ из блока памяти RAM_OLD извлекается код 6. Одновременно с этим по адресу 10_{16} из блока памяти RAM_OLD извлекается код 4 (эти коды выделены на рис. 9.31 рамками). Сумма считанных из памяти кодов равна 10 (превышает длину флагового кода), поэтому на выходе компаратора формируется признак опознания флага. Положение правой границы флага однозначно определяется по приведенной в нижней части рис. 9.30 схеме вычисления кода Q_NEW . Действительно, зная результат вычисления

($Q\_NEW = 4$) и продвигаясь от этого результата к породившим его условиям, можно утверждать, что правая граница флага неизбежно совпадает с границей, разделяющей байт $BYTE\_NEW$ на две равные части (так как иное размещение границы привело бы к другому результату вычисления кода $BYTE\_NEW$).

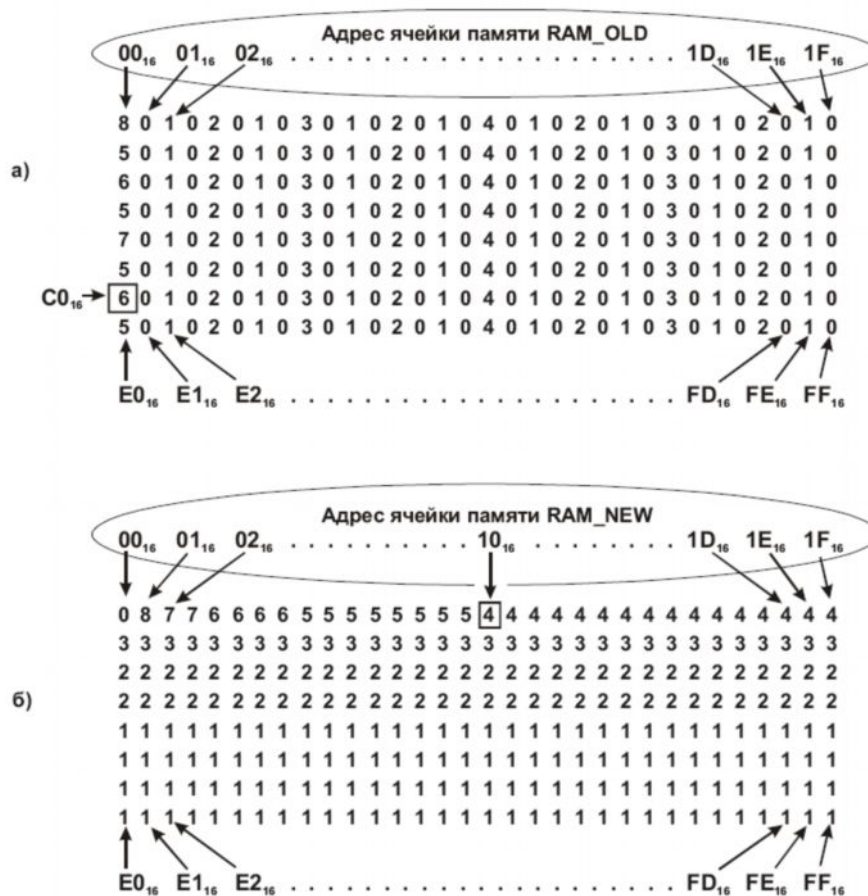


Рис. 9.31. Кодировка блоков памяти $RAMOLD$ (а) и $RAMNEW$ (б)

Отметим, что в данном примере левая граница флага «размыта», т. е. может сливаться с цепочкой соседних нулевых битов, как в только что рассмотренном примере. Поэтому определение положения флага по его левой границе в данном случае невозможно.

С точки зрения пользователя транспортной системы флаг представляется лишним элементом, и поэтому неплохо было бы если не избавиться от него (как ни странно, и это возможно, см. п. 9.2.6), то хотя бы уменьшить его длину, чтобы повысить скорость передачи данных.

9.2.3. Использование чередующихся нулевых и единичных битов в качестве флагов начала кадров. Предлагаемое решение задачи минимизации длины флага основано на использовании чередующихся нулевых и единичных битов в качестве флагов начала кадров [18] и поясняется схемой, приведенной на рис. 9.32. Низкоскоростные потоки данных из каналов 1 - N поступают на входы мультиплексора $MUX 1$ ($MUX 2$). Сумма этих потоков в виде последовательности кадров пересылается с относительно высокой скоростью по линии связи к удаленному мультиплексору $MUX 2$ ($MUX 1$), который распознаёт кадр и распределяет данные по соответствующим каналам.

В предлагаемом решении битстаффинг не нужен, поэтому передаваемый по линии кадр имеет фиксированную длину, флаг представлен одним битом в каждом кадре.

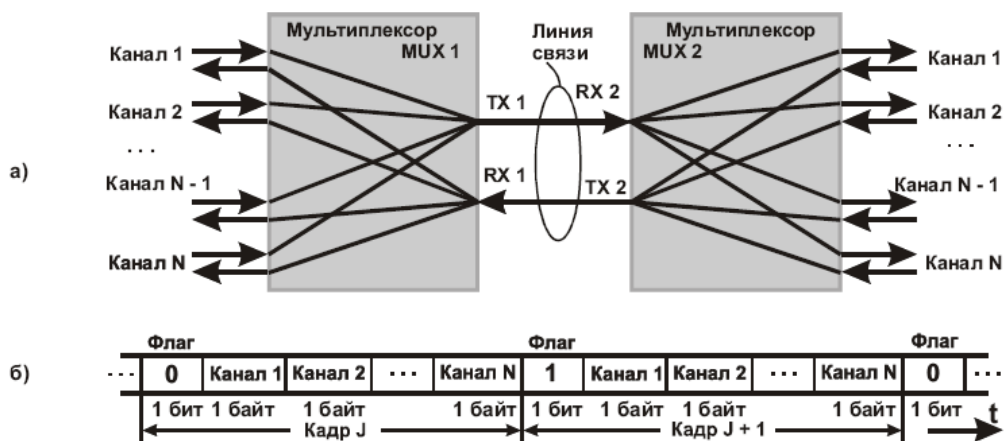


Рис. 9.32. Мультиплексная передача данных с временным разделением каналов:
а-структурная схема; б - структура одного из потоков данных в линии связи
(TX 1 -> RX 2 или TX 2 -> RX 1)

Идея заключается в следующем. В установившемся режиме после вхождения мультиплексоров в логическое соединение, а затем и в кадровый синхронизм каждый мультиплексор помечает передаваемые им в линию последовательные кадры чередующимися одноразрядными флаговыми битами, как показано на рис. 9.32, б. Некоторый кадр с номером J (нумерация условная) содержит флаг, равный лог. 0. Следующий кадр с номером $J + 1$ имеет флаг, равный лог. 1. Последующие кадры содержат флаги, равные соответственно лог. 0, лог.1, лог. 0 и т. д.

Принимая кадр из линии, каждый мультиплексор проверяет соответствие полученного значения флагового бита ожидаемому. Например, после получения кадра с флаговым битом, равным лог. 0, предсказывается поступление кадра с флаговым битом, равным лог. 1. Позиция ожидаемого флагового бита известна, так как кадр имеет фиксированную длину, а предыдущая точка отсчета границы кадра уже отслежена в предыстории.

В отсутствие ошибок наблюдается правильное чередование флаговых битов. Ошибки, например проскальзывания, в конечном счете приведут к нарушению такого чередования и, следовательно, будут обнаружены, но, возможно, не сразу. При этом синхронизация будет временно потеряна и затем вновь восстановлена.

Вхождение мультиплексоров в кадровую синхронизацию. Для начала нужно установить связь между мультиплексорами по линии. Линия может быть выделенной или коммутируемой, содержащей ретрансляторы или иные устройства.

Каждый мультиплексор ($MUX1$, $MUX2$) имеет встроенный модем, подключенный к линии (модемы на рисунках не показаны). Один из модемов является вызывающим, т. е. инициатором установления связи, другой – вызываемым. В результате начального взаимодействия модемов (например, по протоколу *V.42 XID*) и переговоров между ними, в некоторый момент мультиплексоры оказываются на связи и готовы передавать друг другу данные.

Нас сейчас интересует последующая фаза взаимодействия между мультиплексорами - установление кадровой синхронизации.

Последовательность событий такова.

1. Сразу после установления связи между модемами мультиплексоры начинают посылать друг другу непрерывные и неструктурированные потоки G , состоящие из лог. 1 (рис. 9.33, а, рис. 9.34).

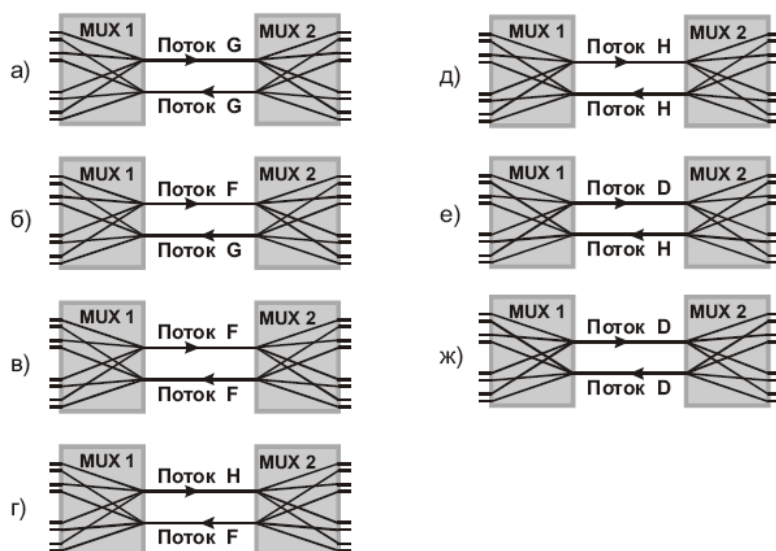


Рис. 9.33. Стадии а-ж установления кадровой синхронизации между мультиплексорами $MUX1$ и $MUX2$. Структура потоков F , G , H и D приведена на рис. 9.34

2. Один из мультиплексоров, например тот, по чьей инициативе организована связь (в данном примере мультиплексор *MUX1*), через некоторое время начинает формировать поток *F* (см. рис. 9.33, б, рис. 9.34). Мультиплексор *MUX2* по-прежнему выдает поток *G*.

3. Мультиплексор *MUX2* обнаруживает в поступающем на его вход потоке *F* периодическое появление комбинаций 1011111101 и расценивает второй лог. 0 каждой такой комбинации как флаг начала смежной пары кадров. Зная длину кадра, мультиплексор *MUX2* расставляет недостающие (единичные) флаги, как это сделано на рис. 9.34. Задача наполовину решена — мультиплексор *MUX2* теперь способен распознавать границы кадров, присылаемых мультиплексором *MUX1*. Факт установления кадровой синхронизации по направлению *MUX1* → *MUX2* внешне проявляется в том, что мультиплексор *MUX2* вместо потока *G* начинает выдавать поток *F* (см. рис. 9.33, в).

4. Мультиплексор *MUX1* обнаруживает в поступающем на его вход потоке *F* периодическое появление комбинаций 1011111101 и расценивает второй лог. 0 каждой такой комбинации как флаг начала смежной пары кадров. Зная длину кадра, мультиплексор *MUX1* расставляет недостающие (единичные) флаги, как это сделано на рис. 9.34. Теперь синхронизация установлена в обоих направлениях, и можно начинать переход к рабочему режиму. Для этого мультиплексор *MUX1* оповещает партнера об успешном установлении двусторонней кадровой синхронизации переходом к выдаче потока *H* с сохранением достигнутой ранее кадровой синхронизации (рис. 9.34). Новое состояние системы показано на рис. 9.33, г.

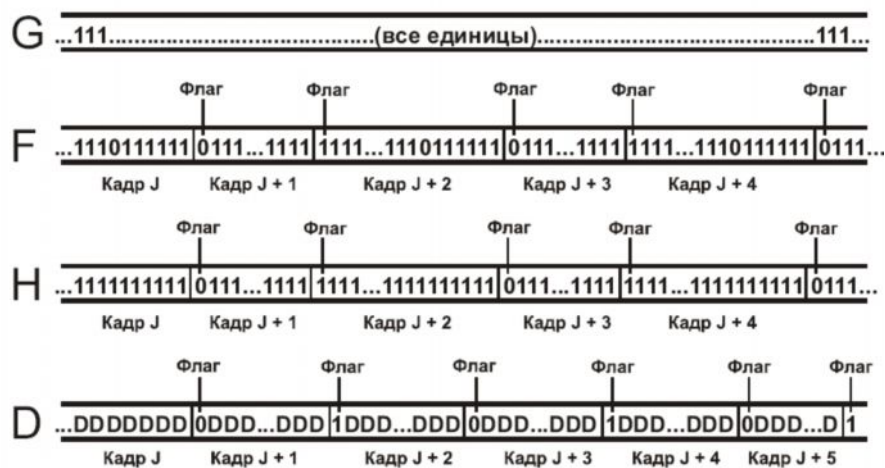


Рис. 9.34. Потоки фиксированных данных (диаграммы *G*, *F*, *H*), используемые на этапе установления синхронизации, и поток полезных данных (диаграмма *D*)

5. Мультиплексор *MUX2* подтверждает готовность перехода к рабочему режиму выдачей партнеру потока *H* с сохранением достигнутой ранее кадровой синхронизации (см. рис. 9.33, а).

6. Мультиплексор *MUX1* обнаруживает на входе поток *H* и переходит к выдаче потока *D* (см. рис. 9.34) с сохранением достигнутой ранее кадровой син-

хронизации. Этот поток соответствует рабочему режиму временного разделения каналов $1 - N$ (см. рис. 9.32). Состояние системы показано на рис. 9.33, е.

7. Мультиплексор *MUX2* через некоторое время после начала выдачи потока *H* переключается на выдачу потока *D* (см. рис. 9.34) с сохранением достигнутой ранее кадровой синхронизации. Этот поток соответствует рабочему режиму временного разделения каналов $1 - N$ (см. рис. 9.32). Состояние системы показано на рис. 9.33, ж. Таким образом, синхронизация достигнута, система обслуживает каналы $1 - N$.

Отметим, что каждый мультиплексор содержит внутренний таймер, который на соответствующих этапах установления синхронизации контролирует время ожидания правильных ответных реакций партнера. При превышении допустимого времени ожидания мультиплексор переходит к выдаче партнеру неструктурированного потока *G* (см. рис. 9.34). Партнер теряет синхронизацию и также переходит к выдаче потока *G*. В результате оба мультиплексора оказываются в рассмотренном ранее исходном состоянии (см. рис. 9.33, а), и попытка вхождения в синхронизм повторяется.

Потеря и восстановление синхронизации. Если в процессе или после установления взаимной синхронизации мультиплексоров хотя бы один из них обнаруживает неправильную последовательность флаговых битов (отличную от последовательности вида ...010101...), то это означает, что синхронизация потеряна.

В качестве примера поведения системы при потере синхронизации рассмотрим последовательность событий, приведенную на рис. 9.35.

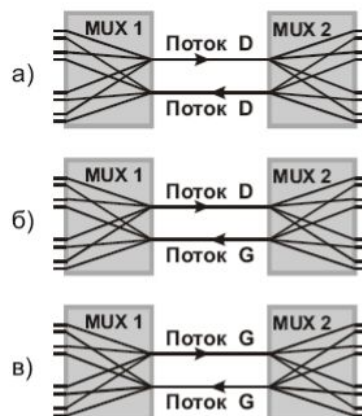


Рис. 9.35. Последовательность состояний системы при потере синхронизации мультиплексором *MUX 2*

1. В исходном состоянии (рис. 9.35, а) оба мультиплексора находятся в рабочем режиме и формируют потоки *D* (см. рис. 9.34 и рис. 9.32).

2. В некоторый момент мультиплексор *MUX2* обнаруживает ошибку в последовательности полученных флаговых битов: вместо ожидаемого лог. 0 принята лог. 1, или наоборот. Так как мультиплексор *MUX2* не может справиться с

проблемой в одиночку (у него нет ориентиров для поиска очередного флагового бита), он обращается за помощью к мультиплексору *MUX1* выдачей ему неструктурированного потока *G* (рис. 9.35,б).

3. Мультиплексор *MUX1* не обнаруживает во входном потоке *G* флаговых битов и теряет синхронизацию с мультиплексором *MUX2* (столь грубая просьба о помощи воспринята). В ответ на потерю синхронизации мультиплексор *MUX1* посылает в мультиплексор *MUX2* поток *G*. Таким образом, ситуация усугубляется – синхронизация отсутствует уже в обоих направлениях (рис. 9.35, в). Мультиплексоры передают друг другу потоки *G*, а это и есть начальное условие для запуска уже рассмотренной процедуры установления синхронизации (см. рис. 9.33, а).

Через определенное время, заданное таймером мультиплексора *MUX1*, система переходит в состояние, показанное на рис. 9.33, б, и т. д. Процесс завершается полным восстановлением синхронизации (см. рис. 9.33, ж).

Отметим, что для установления синхронизации можно воспользоваться методом статистического анализа потока принимаемых данных (см. п. 9.2.5)

9.2.4. Использование псевдослучайных битов в качестве флагов начала кадров. В только что описанной системе передачи данных последовательность передаваемых кадров сопровождается чередующимися одноразрядными флаговыми битами: «...010101...». В установившемся режиме приемник следит за правильностью следования флаговых битов и обрабатывает принимаемые кадры. Однако однородный поток кадров в большинстве случаев необходимо структурировать, т. е. помечать в нем более крупные информационные объекты, например, группы кадров. Начало группы можно помечать, например, установкой в единицу некоторого служебного бита в заголовке кадра, но такое решение неэкономично. Нельзя ли совместить использование одноразрядных флаговых битов с разметкой относительного положения кадров в их непрерывном потоке?

Чтобы решить эту задачу, следует как-то преобразовать однородную последовательность «...010101...». Для введения признака начала группы кадров можно было бы ввести в эту последовательность преднамеренное нарушение, например заменой нуля единицей или наоборот. Однако такая замена не обеспечивает хорошей отличимости внесенного нарушения от искажений сигнала в результате действия помех в линии. Еще одна проблема заключается в том, что аппаратура формирования кадровых битов и их распознавания должна быть достаточно простой, процесс вхождения в синхронизацию должен обладать быстрой сходимостью. Этим требованиям отвечает решение рассмотренное на рис. 9.36.

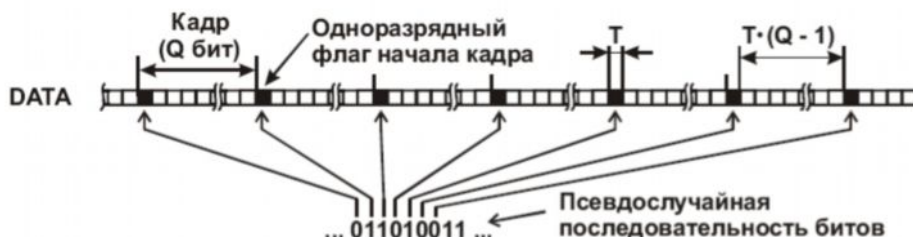


Рис. 9.36. Схема, поясняющая идею использования псевдослучайных битов в качестве флагов начала кадров; *DATA* - поток битов между передатчиком и приемником

Так же, как и в предыдущем решении (см. рис. 9.32) начало кадра обозначается флаговым битом. Но в данном случае цепь из флаговых битов нетривиальна по структуре и формируется с помощью генератора псевдослучайной последовательности битов (рис. 9.37).

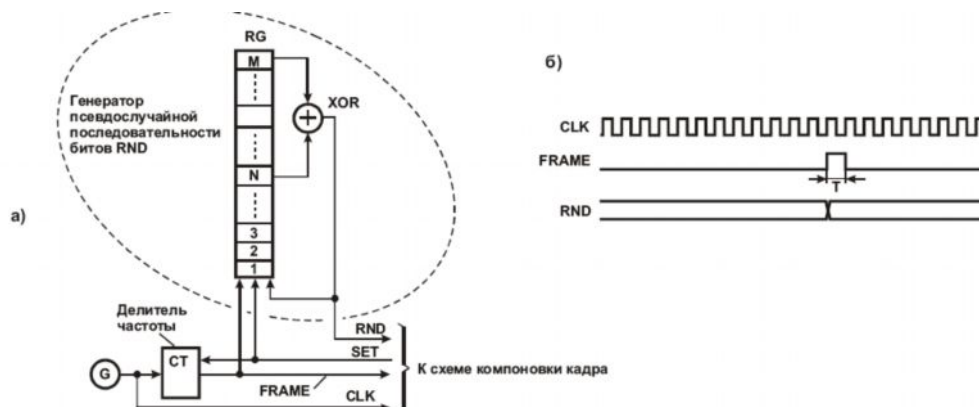


Рис. 9.37. Формирование флаговых битов: а - схема; б- временная диаграмма

Предположим, что число битов в кадре равно Q , частота сигнала CLK от генератора G , задающего скорость передачи данных, равна F Гц. Длительность T битового интервала составляет $1/F$ с. Генератор псевдослучайной последовательности битов RND построен по классической схеме на основе M -разрядного сдвигового регистра RG и элемента Исключающее ИЛИ (XOR), входы которого соединены с выходами разрядов M и N регистра. Выход элемента XOR соединен с входом данных регистра RG . Делитель частоты (счетчик по модулю Q) снижает частоту F сигнала CLK в Q раз и формирует кадровый импульс $FRAME$ длительностью T , который поступает в схему компоновки кадра и одновременно воздействует на вход синхронизации регистра RG генератора псевдослучайной последовательности битов. По положительному фронту сигнала $FRAME$ этот генератор формирует очередной псевдослучайный бит RND . Схема компоновки кадра при обнаружении импульса $FRAME$ считывает бит RND и использует его для вставки очередного флага в поток данных, как было показано на рис. 9.36.

По сигналу SET регистр RG устанавливается в состояние $111...1$, а делитель частоты — в нулевое состояние. Такая установка позволяет задать исходную точку в системе псевдослучайного отсчета передаваемых кадров. В этой системе отсчета последовательность кадров неявно нумеруется кодами в регистре RG . Эти коды формируются в заранее известном и строго определенном порядке, зависящем от разрядности регистра и точки подключения к нему обратной связи. Так, предварительно установленный в состояние $111...1$ регистр RG после воздействия первого импульса $FRAME$ переходит в состояние $011...1$ (пер-

вый разряд – слева), после второго импульса - в состояние 001... 1 и т. д. В полном цикле работы генератора псевдослучайной последовательности битов в регистре однократно формируются все возможные коды, за исключением нулевого. Полученные таким способом M -разрядные номера кадров в явном виде не передаются в удаленный приемник, но могут быть восстановлены им при реставрации псевдослучайной последовательности битов, как будет показано далее.

В протоколе обмена данными номера кадров определяют их статус. Например, при $M = 5$ по линии связи передаются группы из $2^5 - 1 = 31$ кадров. В каждой группе кадр с номером 11111_2 рассматривается как начальный. Приемная аппаратура распознаёт эти кадры по их номерам и трактует их содержимое по-разному, в соответствии с принятым протоколом.

Сформированная последовательность кадров с одноразрядными флаговыми битами (рис. 9.36) передается по линии связи в приемную аппаратуру. Для распознавания границ и номеров кадров применяется схема (рис. 9.38) на основе такого же генератора псевдослучайной последовательности, как и в схеме формирования флаговых битов.

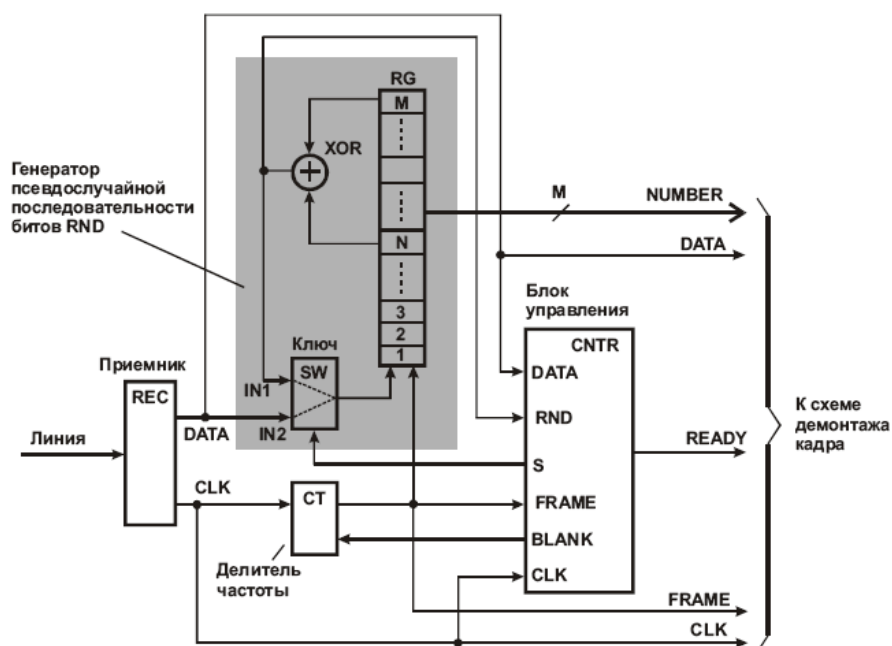


Рис. 9.38. Схема распознавания межкадровых границ

Приемник *REC* усиливает принятый сигнал и выделяет из него синхросигнал *CLK* и биты *DATA* – собственно данные и вставленные в нужных местах псевдослучайные биты флагов начала кадров (см. рис. 9.36). Делитель частоты выполнен в виде счетчика по модулю Q (напомним, что Q – число битов в кадре). После установления правильной синхронизации восстановленные сигналы *CLK*, *FRAME* (признак наличия в потоке флагового бита) и *RND* (восстановленный псевдослучайный бит) находятся в тех же временных соотношениях, что и

соответствующие сигналы в схеме формирования флаговых битов (см. временные диаграммы на рис. 9.37).

Сигнал *BLANK* длительностью в один период T сигнала *CLK* в необходимых случаях формируется блоком управления *CNTR* и притормаживает работу делителя частоты на один такт. Это позволяет смещать импульс *FRAME* на один такт вправо относительно «неподвижной» последовательности битов *DATA* при поиске истинного положения флагового бита. Блок *CNTR* формирует также сигнал S управления электронным ключом *SW*. На этапе установления синхронизации ключ транслирует на выход сигнал с входа *IN2*. Когда синхронизация установлена, ключ передает на выход сигнал с входа *IN1*, т. е. переводит генератор псевдослучайной последовательности битов в автономный режим, при котором он нечувствителен к сигналам *DATA*. Параллельный M -разрядный код *NUMBER* на выходах регистра *RG* при правильной синхронизации отображает псевдослучайный номер кадра и с некоторой задержкой повторяет код в аналогичном регистре схемы формирования флаговых битов (см. рис. 9.37). Пока синхронизация не установлена, сигнал $READY = 0$; после установления синхронизации $READY = 1$.

Процесс установления синхронизации сопровождается серией довольно длительных экспериментов, которые проводятся приемной аппаратурой (эта аппаратура, разумеется, должна обладать достаточным «интеллектом»). Перед началом каждого эксперимента делается предположение о том, что некоторая периодически повторяющаяся позиция в потоке битов соответствует флаговой. Конечно, вероятность попадания в правильную сетку размещения флаговых битов с одной попытки мала и составляет $1/Q$. Но если «повезет», то первый же эксперимент позволит войти в синхронизацию, т. е. правильно ориентироваться в потоке кадров, отслеживая их границы и номера. Если первый эксперимент оказался неудачным, то предполагаемая сетка размещения флаговых битов смещается блоком управления на один бит вправо. Проводится второй эксперимент и т. д. В худшем случае правильная синхронизация окажется возможной только после проведения последнего эксперимента с номером Q .

Предположим, что в начале эксперимента синхронизации нет, сигнал $READY = 0$, регистр *RG* и делитель частоты находятся в произвольных состояниях. Блок управления *CNTR* переводит ключ *SW* в режим передачи сигналов с входа *IN2*. При прохождении определенной позиции каждого кадра (эта позиция «подозревается» в качестве носителя флагового бита) на выходе делителя частоты формируется импульс *FRAME*. По его положительному фронту в первый разряд регистра *RG* принимается бит - кандидат на звание флагового. Одновременно с этим остальные биты этого регистра сдвигаются вверх на один разряд, бит из разряда M теряется.

По прошествии M кадров регистр *RG* заполняется принятыми битами, «подозреваемыми» в выполнении роли флаговых. Теперь возможны две ситуации.

1. *Регистр действительно заполнен флаговыми битами группы из M соседних кадров.* В этой ситуации содержимое регистра *RG* с некоторой задержкой повторяет код в аналогичном регистре схемы формирования флаговых би-

тов (см. рис. 9.37). Существенно, что всякий раз в момент записи очередного бита в первый разряд регистра *RG* сигналы *RND* и *DATA* на входах *IN1* и *IN2* ключа *SW* совпадают. Это происходит благодаря тому, что в передающей и приемной аппаратуре применены одинаковые генераторы псевдослучайных последовательностей битов. Поэтому сигнал на входе *IN1* формируется по тем же правилам, которым следует сигнал на входе *IN2*. Иными словами, сигнал *RND* можно рассматривать как результат предсказания значения очередного флагового бита, поступившего из линии.

Блок управления следит за совпадением предсказанных и фактически принятых флаговых битов. Если предсказания постоянно оправдываются при прохождении, например, 100 кадров, то предполагается, что синхронизация достигнута, сигнал *READY* устанавливается в состояние лог. 1. Блок управления переводит ключ *SW* в режим передачи сигнала с верхнего входа (*IN1*), так что генератор псевдослучайной последовательности битов переходит к автономной работе и не реагирует на ошибки, которые могут появляться в сигнале *DATA*. Это создает некоторую разумную инерционность системы синхронизации и предотвращает выход и повторное вхождение в синхронизм при допустимом уровне ошибок в линии.

Теперь для дальнейшей обработки кадра имеется вся необходимая информация. Обработка, прежде всего, заключается в демонтаже кадра: вычеркивании флагового бита, определении номера кадра в группе, проверке и вычеркивании контрольных сумм и т. д.

При правильной синхронизации блок управления *CNTR* продолжает сопоставлять предсказанные и фактически принятые флаговые биты. Интенсивность несовпадений битов в этой ситуации не должна превышать некоторую заданную величину. Например, допустимым может считаться одно несовпадение на 100 тыс. кадров, в противном случае качество системы передачи данных расценивается как низкое. При резком повышении интенсивности несовпадений битов синхронизация считается потерянной, сигнал *READY* переводится в состояние лог. 0 и блок управления начинает новую серию экспериментов по восстановлению синхронизации.

2. *Регистр заполнен не флаговыми битами.* Эта ситуация приводит к множественным несовпадениям предсказанных и фактически полученных битов, причем вероятность формирования непрерывной последовательности ложных случайных совпадений быстро уменьшается со временем и с ней можно не считаться. Сигнал *READY* остается в нулевом состоянии, эксперимент признается неудачным. Блок управления *CNTR* формирует сигнал *BLANK*, который приостанавливает работу делителя частоты на один такт. После этого начинается новый эксперимент, при котором флаговые биты ищутся в соседних (сдвинутых вправо на один такт) позициях кадров и т. д.

Подводя итоги материалам, рассмотренным в п. 9.2.1...9.2.4, отметим следующее.

1. Длину флага можно сократить до одного бита, но для его распознавания нужен достаточно высокий «интеллект» приемной аппаратуры.

2. Использование псевдослучайных последовательностей флаговых битов позволяет не только распознавать границы кадров, но и вести их относительную нумерацию в пределах выделенных групп. Нумерация позволяет приемнику классифицировать принимаемые кадры по их номерам в соответствии с принятым протоколом обмена. Например, приемник может отличать «обычные» кадры от служебных, а среди них различать кадры различных типов. В многоканальной системе передачи данных (с мультиплексированием каналов) номер кадра позволяет приемнику определить, к какому каналу относятся содержащиеся в кадре данные, и т. д.

9.2.5. Использование раздробленного флага начала кадра. Вновь вернемся к вопросу синхронизации приемника с передатчиком. Напомним, что поток передаваемых по линии связи битов состоит из структурных единиц (например, кадров). Идея построения «обычного» кадра поясняется рис. 9.39. Кадр представляет собой группу битов с границами p и q . Флаг размещен в начале кадра, представлен уникальным кодом и имеет фиксированную длину; его границы - a и b . За флагом следуют слова $b - c$, $c - d$, $d - e$ и т. д.

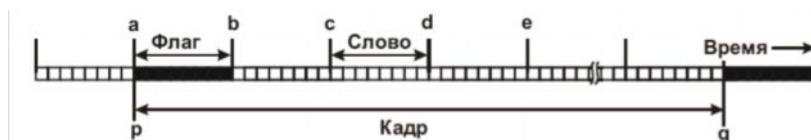


Рис. 9.39. Пример структуры кадра

Как следует из рассмотренных выше примеров, длину флага можно сократить до одного бита, при этом битстаффинг не нужен. Здесь мы приведем родственное, но более общее решение, в котором многоразрядный флаг существует, но в виде разобренных битов. Такое решение позволяет быстро устанавливать синхронизацию между устройствами и сохранять ориентиры (биты флага) при частичном повреждении кадра из-за помех в линии.

Рассмотрим предлагаемую структуру кадра (рис. 9.40 и рис. 9.41).

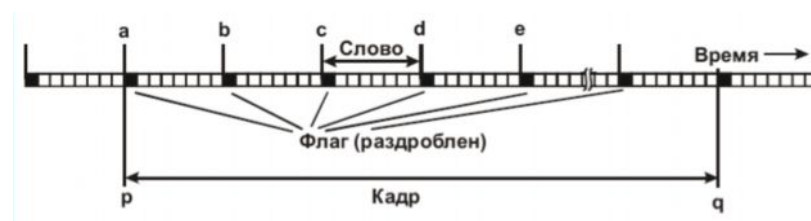


Рис. 9.40. Структура кадра с раздробленным на отдельные биты флагом

В отличие от традиционной структуры кадра (рис. 9.39), в данном случае флаг фиксированной длины раздроблен на отдельные биты. Эти биты дополнительно служат метками начала слов кадра. В то же время совокупность флаговых битов позволяет достаточно надежно распознать кадр как целое. Длина слова фиксирована, битстаффинг не нужен.

Приемник распознаёт флаг с использованием вероятностных оценок. При анализе входного потока битов он выявляет в нем устойчивые закономерности. Проще говоря, приемник как бы просматривает поток, показанный на рис. 9.41, сквозь непрозрачную маску, в которой вырезаны отверстия, соответствующие серым квадратикам на рисунке. Маска может накрывать, например, 10 кадров, так что при четырехразрядном флаге в прорези маски попадает цепочка из $10 \times 4 = 40$ битов.

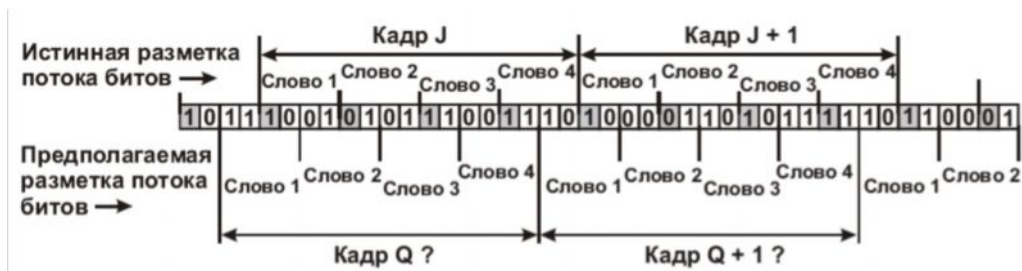


Рис. 9.41. Упрощенный пример разметки потока битов на кадры и слова. Флаговые биты (рассредоточенные коды 1011_2) выделены серым фоном

Шаг за шагом перемещая маску вдоль исследуемого потока, не позже чем через три шага увидим сквозь ее отверстия правильную флаговую комбинацию битов, повторенную 10 раз, что с высокой вероятностью подтверждает факт обнаружения исходной разметки кадров. (Как видим, число шагов поиска невелико, что способствует быстрому установлению синхронизации приемника с передатчиком.) В отсутствие ошибок передачи вероятность ложного обнаружения искомой цепочки из 40 битов в случайном потоке данных составляет и уменьшается по мере дальнейшего перемещения маски.

9.2.6. Распознавание ячеек ATM в битовом и байтовом потоках данных. Технология *ATM* (*ASYNChRoNoUS TRaNSFeR MoDe* – асинхронный режим передачи) предусматривает размещение данных внутри 53-байтовых ячеек (рис. 9.42). Как показано на рисунке, ячейки следуют по линии непрерывным потоком, хотя возможны и иные варианты их транспортирования.

Ячейка состоит из заголовка и поля данных (соответственно 5 и 48 байтов). Заголовок содержит четыре байта для размещения управляющей информации (28 битов – номер логического канала плюс четыре флаговых бита) и один контрольный байт *CRC* (*CYcLiCaL ReDUNDaNcY Check* – циклический избыточный код).



Рис. 9.42. Непосредственная передача ячеек ATM по линии связи

Байт *CRC* вычисляется передатчиком и представляет собой контрольную сумму четырех предыдущих байтов. Приемник после получения ячейки проводит аналогичные вычисления и сопоставляет свой результат с имеющимся в ячейке кодом *CRC*. В отсутствии ошибок коды совпадают. Но как приемник определяет границы ячейки, ведь она не содержит флагов начала или конца?

Использование кода *CRC* в процессе распознавания границ ячеек. В общем виде методика поиска границ ячеек состоит в следующем. На начальном этапе поиска приемник просматривает поступающий поток ячеек (см. рис. 9.42) сквозь «непрозрачную маску с вырезанной в ней щелью», сквозь которую видны 40 битов данных. Другими словами, перед приемником проходит «бегущая строка» длиной 5 байтов. В каждом такте строка смещается на один бит.

Приемник в каждом такте проверяет содержимое бегущей строки на предмет обнаружения в ней заголовка ячейки. Если соответствующий байт (самый новый в бегущей строке) представляет собой контрольную сумму (*CRC*) полученных ранее четырех байтов, то делается предположение о том, что заголовок найден. Если это предположение оправдано, то в дальнейшем моменты обнаружения заголовка будут следовать с шагом, равным длине ячейки. Если совпадение было случайным (его вероятность равна $1/256$), то оно не найдет подтверждения при последующей проверке событий на периодичность.

Как следует из рис. 9.42, истинный заголовок в наихудшем случае попадает в поле зрения приемника через $53 \times 8 - 1 = 423$ такта после начала поиска. Это соответствует ситуации, при которой поиск начат как раз в тот момент, когда предыдущий заголовок только что сместился на один бит и уже не полностью попадает в область бегущей строки.

После выделения цепи правильных совпадений на фоне случайных приемник точно знает положение границ ячеек. Синхронизация с передатчиком достигнута. Но описанный анализ потока битов продолжается, и каждый новый заголовок проверяется на правильность. Если в заголовке обнаружена корректирующая ошибка, то она исправляется (код *CRC* позволяет вычислить положение искаженного бита). Если ошибка некорректируемая, то ячейка уничтожается. Последовательность заголовков с некорректируемыми ошибками соответствует потере синхронизации, поэтому приемник возвращается к начальному этапу поиска границ ячеек.

Поиск заголовка в непрерывном байтовом потоке данных. Ячейки АТМ могут передаваться либо непосредственно по линии связи, как было показано на рис. 9.42, либо в составе более крупных структурных единиц, например кадров *SDH* (рис. 9.43). (*SDH* – *SYNChRoNoUS DIGITaL HieRaRchY* – синхронная цифровая иерархия, европейский стандарт на средства передачи данных.)

Здесь существенно то, что аппаратура обработки кадра распознаёт его начало и преобразует поток битов в поток байтов. Это позволяет снизить необходимую скорость обработки данных в восемь раз. Кадр построен так, что заранее неизвестно, где размещен блок ячеек АТМ. Поэтому задача отыскания границ ячеек остается актуальной и при байтовом потоке данных.

Более подробно ознакомиться со схемой распознавания заголовка ячейки в байтовом потоке данных можно в [18].

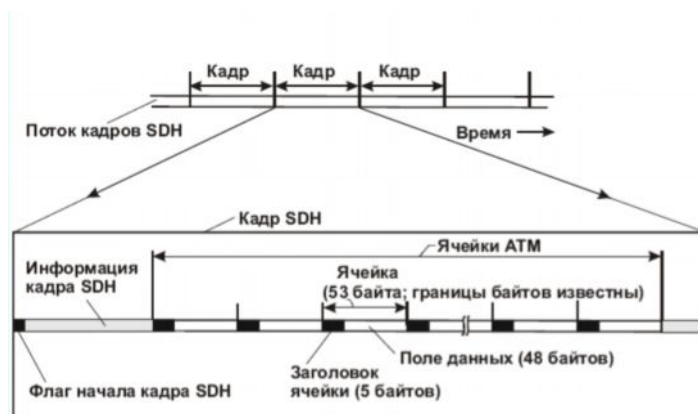


Рис. 9.43. Передача ячеек ATM по линии связи в упакованном виде. Ячейки помещены в кадры SDH

9.2.7. Распознавание и восстановление искаженных кадров при передаче данных по радиоканалу. В системе мобильной телефонной связи (рис. 9.44) предусмотрена одновременная передача «оцифрованных» речевых и иных данных по радиоканалу. Данные, не относящиеся к передаче речи, пересылаются в паузах между словами и фразами. Дальнейшее описание относится именно к таким данным.



Рис. 9.44. Фрагмент системы мобильной телефонной связи

Потоки данных состоят из последовательностей кадров. Кадр первоначально формируется в памяти передающего устройства и представляет собой группу байтов, в которой первый и два последних служебные, а остальные содержат полезные данные, объединенные в сообщение (рис. 9.45).



Рис. 9.45. Структура кадра до его выдачи в радиоканал

Биты Q первого байта задают порядковый номер кадра, биты L определяют длину сообщения. Если все биты L нулевые, то сообщение не содержит ни одного байта. Такой «ACK-кадр» (от *AckNoWLeDGe* - ответ) рассматривается как

подтверждение успешного приема кадра, содержащего сообщение. (Ответный кадр пересылается в противоположном направлении по отношению к кадру, несущему сообщение.) Два последних байта представляют собой контрольную сумму (16-разрядный циклический код *CRC*) всех предшествующих байтов кадра.

Дальнейшая подготовка кадра к выдаче в радиоканал состоит в его дроблении на отдельные асинхронные послылки. При этом каждый байт преобразуется в две послылки, как показано на рис. 9.46.

Преобразование состоит в следующем. Сначала байт $D1 \dots D8$ делится на два полубайта: $D1 \dots D4$ и $D5 \dots D8$. Им присваиваются четырехразрядные порядковые номера: $N1 \dots N4$ и $M1 \dots M4$. Эти номера «склеиваются» с соответствующими полубайтами, в результате образуются два байта:

$$N1 \dots N4 D1 \dots D4 \text{ и } M1 \dots M4 D5 \dots D8.$$

На заключительной стадии преобразования к обоим байтам добавляются стартовые биты *ST*, биты *P* контроля по четности (нечетности) и стоповые биты *ST*.

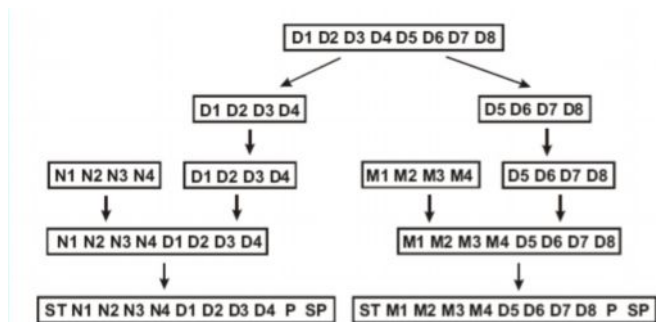


Рис. 9.46. Преобразование байта в две старт-стоповые послылки

В результате преобразования исходный кадр дробится на множество посылок (их число равно удвоенному числу байтов кадра). Они выдаются в радиоканал в паузах между «всплесками» передачи речевой информации. Поэтому поток данных неравномерен и может содержать как следующие друг за другом вплотную, так и разрозненные группы посылок или отдельные послылки. В процессе передачи по радиоканалу часть посылок может быть потеряна из-за возможных замираний сигнала или действия помех. Приемник может также получать ложные послылки, которые вклинились в неравномерный поток правильных. Чтобы обеспечить надежный обмен данными в условиях замираний сигнала и действия помех, предлагаются три простые, но примечательные идеи.

Идея 1. Проверка правильности порядка следования посылок слежением за динамикой изменения номеров полубайтов. Как было показано, каждая посылка содержит полубайт данных и предшествующий ему четырехразрядный номер этого полубайта. В отсутствие ошибок приемник должен регистрировать некоторую заранее заданную последовательность номеров полубайтов, например такую: $\dots 0, 1, 2, 3, \dots, F, 0, \dots$ (использована шестнадцатиричная форма записи чисел). Если получена последовательность $\dots 0, 1, 2, 7, 3, 4, 5, 6, \dots$, то

можно сделать вывод о том, что посылка с порядковым номером 7 нарушает динамику изменения номеров и явно лишняя. Поэтому приемник просто отбрасывает ее. Аналогично в последовательности ... 0, 1, 2, 4, 5, 6, ... недостает цифры 3. Это означает, что соответствующая посылка потеряна по пути к приемнику. Поэтому передача ошибочного кадра повторяется в связи с неполучением подтверждения правильного приема данных.

Идея 2. Расширение диапазона нумерации посылок при ограниченной разрядности номеров. С помощью четырехразрядного двоичного кода можно отображать числа в диапазоне 0 - F_{16} . Поэтому на первый взгляд представляется естественной простая последовательная нумерация посылок: ... 0, 1, 2, ..., 9, A, B, C, D, E, F, 0, 1, 2, ... Однако такая нумерация повторяется с периодом, равным 16, а кадр может быть представлен десятками и сотнями посылок. Действительно, можно построить достаточно большой кадр, если принять условие, что длина сообщения (см. код *LLLLL* на рис. 7.33) выражается не в отдельных байтах, а в более крупных единицах, например, в группах из четырех байтов.

Можно ли с помощью четырехразрядных кодов пронумеровать сотню или более посылок так, чтобы приемник мог уверенно определять положение любой из них относительно начала кадра? На первый взгляд, ответ на этот вопрос может быть только отрицательным. Но решение есть, и в этом мы сейчас убедимся.

Пример. Предположим, что составляющие кадр посылки пронумерованы с помощью такой последовательности номеров:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 0, 3, 6, 9, C, F, 2, 5, 8, B, E, 1, 4, 7, A, D, 0, 5, A, F, 4, 9, ..., 0, 7, E, 5, C, ...

Первые 16 посылок пронумерованы в обычном порядке, начиная с нулевого номера. Следующие 16 посылок также нумеруются, начиная с нуля, но номер наращивается с шагом, равным трем. Следующие 16 посылок нумеруются с нуля с шагом 5, следующие - с шагом 7, затем с шагами 9, 11, 13 и 15. В этом примере в качестве шагов выбраны восемь последовательно возрастающих чисел, не превышающих 15 и не имеющих общего делителя с числом 16, т. е. нечетные числа. Таким образом, получена *уникальная* последовательность из $16 \times 8 = 128$ *неуникальных* номеров. Эту последовательность можно продолжить, применив какие-либо заранее учтенные в протоколе обмена дополнительные правила ее формирования. Например, можно повторить последовательность, поменяв местами номера, размещенные в первых и последних пяти позициях каждой группы из 16 посылок и т. д. Иными словами, возможности построения такого рода уникальных последовательностей номеров практически безграничны.

Приемник просматривает полученную последовательность и проверяет ее правильность. Как уже было показано, зная вид последовательности, можно выявить и отбросить лишние номера и соответствующие посылки, если они есть, и зафиксировать пропущенные номера посылок. Так как последовательность уникальна, то можно отыскать ее начало и конец, т. е. идентифицировать первую и последнюю посылки кадра.

Отметим, что рассмотренная идея применима также и к нумерации кадров – наличие трех битов *QQQ* для указания номера кадра вовсе не означает, что соответствующий диапазон составляет только восемь номеров.

Идея 3. Усовершенствованное мажоритарное восстановление ошибочных байтов кадра. Если приемник получил все посылки, относящиеся к кадру, то он реконструирует байты в последовательности, обратной той, которая была приведена на рис. 9.44. Для этого он отбрасывает служебные биты *ST*, *P* и *SP* выбранной пары посылок, затем уничтожает их номера *N1* - *N4* и *ML* - *M4* и склеивает полубайты. В результате формируется первоначальный байт *DL* - *D8*.

После формирования всех байтов кадра приемник проверяет его контрольную сумму (см. рис. 9.45). Если контрольная сумма правильная, то приемник посылает уведомление об этом источнику кадра. (Напомним, что в качестве уведомления используется кадр с нулевым кодом *LLLLL* длины сообщения.)

Если контрольная сумма неправильная, или в кадре отсутствуют некоторые байты, то приемник не посылает уведомление, сохраняет имеющуюся информацию о кадре и ожидает поступления копии ошибочного кадра. При поступлении копии проверяется ее контрольная сумма. Если она правильная, то посылается уведомление об этом, в противном случае приемник сопоставляет две ошибочные версии кадра и пытается собрать из них один полноценный кадр. Если это удалось сделать и контрольная сумма правильная, то посылается уведомление, иначе приемник ждет поступление новой копии и т. д.

При наличии нескольких копий кадра применяется обычный метод мажоритарного выбора, при котором используется «голосование по большинству». Например, если два одноименных байта из трех совпадают, то их значение принимается в качестве истинного. Если все три байта разные, то голосование переносится на уровень битов. Это позволяет в некоторых ситуациях, когда одиночные ошибки распределены по разным битам, восстановить правильный байт. Но если, например, из пяти одноименных байтов два совпадают между собой, но отличаются от трех других, которые, в свою очередь, не совпадают друг с другом в любых сочетаниях, то эти совпадающие байты принимаются в качестве истинных, хотя, строго говоря, они - в меньшинстве. Конечно, окончательное решение об отсутствии ошибок в кадре принимается после проверки его контрольной суммы.

9.2.8. Распознавание флагового кода, содержащего ошибки. В некоторых системах, например в системе мобильной связи, передача кадров часто происходит на фоне помех, которые искажают отдельные биты или группы битов. В частности, не исключена возможность появления искажений во флаговых кодах, обозначающих начало кадра. С этими искажениями нельзя не считаться, так как если ориентироваться только на получение безошибочных флаговых кодов, то из-за их нерегулярного поступления трудно установить или поддерживать ранее установленную кадровую синхронизацию. Задача распознавания флаговых кодов, в том числе, содержащих ошибки, может быть решена с помощью схемы, приведенной на рис. 9.47, а [18].

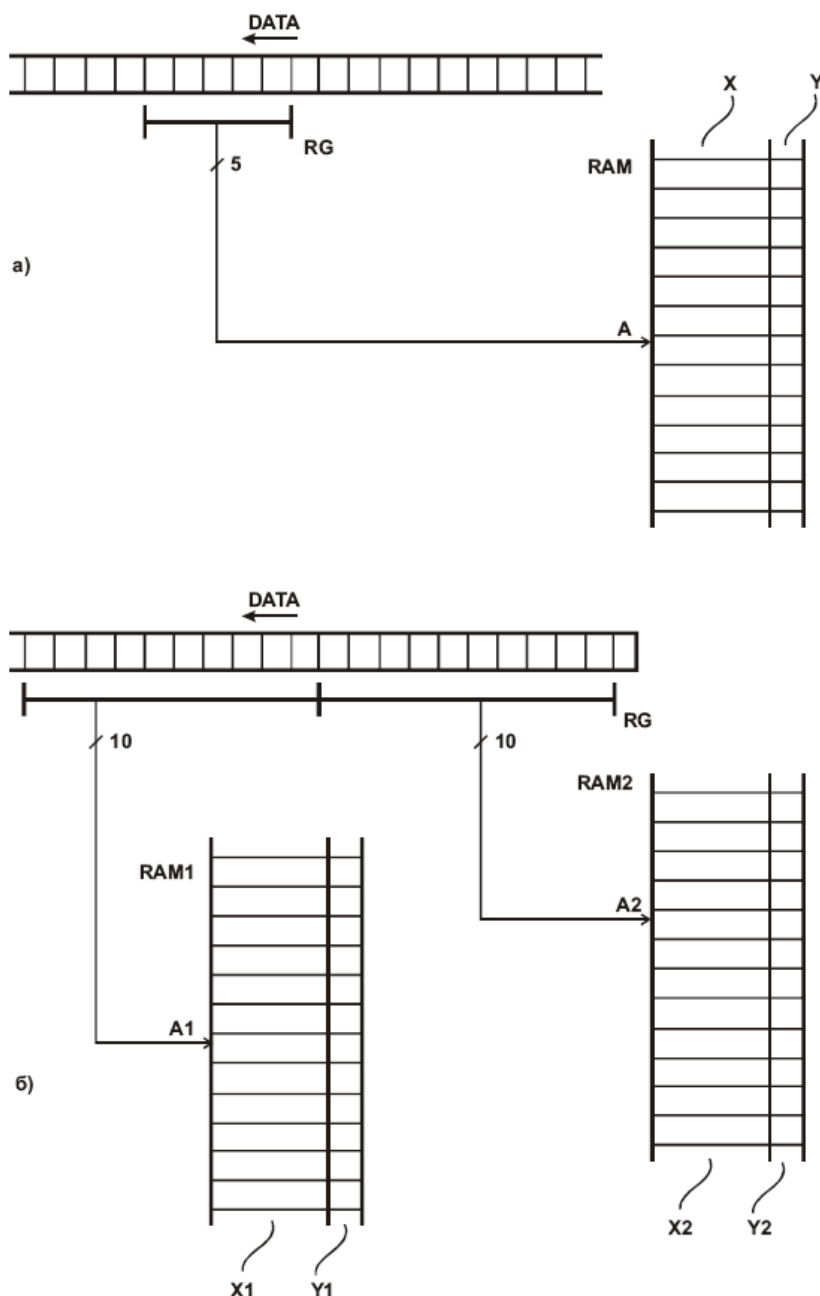


Рис. 9.47. Схема поиска флагового кода:

а - с использованием одного блока памяти; б - с использованием двух блоков памяти

Входной поток данных *DATA* проходит через сдвиговый регистр *RG*. Этот процесс можно интерпретировать как просмотр движущейся последовательности битов через неподвижное пятиразрядное окно (направление движения данных показано стрелкой). Параллельный код из регистра *RG* поступает на адресные входы блока памяти *RAM*.

Предположим, что в искомом флаговом коде 10101 (малая разрядность выбрана для удобства иллюстрации идеи) допустима одиночная ошибка. Тогда помимо этого кода схема поиска должна распознавать еще пять кодов, отличающихся от искомого в одном из разрядов: 00101, 11101, 10001, 10111 и

10100. Для распознавания правильного и пяти близких к нему ошибочных кодов в память предварительно записывается информация, представленная в табл. 9.5.

Таблица 9.5.

Таблица кодировки блока памяти (см. рис. 9.47, а)

Адрес	Эталонный код (X)	Число ошибок (Y)
00000	-	-
00001	-	-
00010	-	-
00011	-	-
00100	-	-
00101	10101	1
00110	-	-
00111	-	-
01000	-	-
01001	-	-
01010	-	-
01011	-	-
01100	-	-
01101	-	-
01110	-	-
01111	-	-
10000	-	-
10001	10101	1
10010	-	-
10011	-	-
10100	10101	1
10101	10101	0
10110	-	-
10111	10101	1
11000	-	-
11001	-	-
11010	-	-
11011	-	-
11100	-	-
11101	10101	1
11110	-	-
11111	-	-
<u>Примечание.</u> Символ “—” означает отсутствие данных и может кодироваться отдельным битом		

Из таблицы следует, что точное или приближенное (с ошибкой в одном разряде) совпадение входного кода с заданным флаговым кодом сопровождается выдачей из памяти эталонного кода X и числа ошибок Y . В данном примере эталонный код уникален (10101), поэтому из таблицы можно было бы исклю-

чить второй столбец. Однако эталонных кодов может быть несколько, например, когда каждый из них соответствует кадру определенного иерархического уровня. В этом случае по считанным из памяти кодам X и Y определяется ближайший эталонный код и степень близости к нему входного кода. На основе считанных из памяти данных накапливается статистическая информация, по которой можно провести правильное разбиение потока данных на кадры.

С увеличением длины флагового кода растет требуемый объем блока памяти. Так, для анализа 20-разрядного кода необходим блок памяти объемом 2^{20} ячеек. Чтобы уменьшить этот объем, можно вместо одного применить несколько параллельно включенных блоков памяти. Как показано на рис. 9.46, б, 20-разрядный код из регистра RG разделен на две группы по десять разрядов. Каждая группа используется в качестве адреса для выбора одной из $2^{10} = 1024$ ячеек в соответствующих блоках памяти $RAM1$ и $RAM2$. Эти блоки осуществляют сравнение фрагментов входного и эталонных кодов.

Такое решение, однако, предполагает наличие дополнительных схем (на рисунке не показаны) для согласованной оценки результатов сравнения фрагментов кодов. Так, условие $Y1 = Y2 = 0$ соответствует точному совпадению входного кода с одним из эталонных только в том случае, когда 20-разрядный код $X1 X2$ также совпадает с этим же эталонным кодом. (Иными словами, нужно исключить из рассмотрения совпадения фрагментов входного кода с соответствующими фрагментами, принадлежащими разным эталонным кодам). Ошибки, обнаруженные во фрагментах входного кода (коды $Y1$ и $Y2$), должны суммироваться для последующего сравнения с максимально допустимым числом.

Таким образом, в условиях повышенного уровня помех приемник проявляет определенную «терпимость» к ошибкам во флаговых кодах. Он пытается поддерживать достигнутую ранее или возобновить утраченную синхронизацию с передатчиком. Принятие решения о наличии или потере синхронизации основывается на основе анализа статистических данных об ошибках.

9.2.9. Поиск флага в потоке данных, передаваемых по волоконно-оптической линии связи. Задача состоит в обнаружении флаговой комбинации битов во входном потоке данных. Но специфика ее решения обусловлена очень высокой скоростью передачи данных, при которой длительность битового интервала составляет доли нано секунды.

Устройство для распознавания флаговой комбинации битов (рис. 9.48) содержит оптический разветвитель (рис. 9.49), двенадцать оптических линий задержки, два фотоприемника, дифференциальный усилитель и компаратор. В данном примере флаг представлен 13-разрядным кодом 1111100110101. При обнаружении флага формируется сигнал $S = L$.

Данные D передаются по волоконно-оптической линии связи в виде световых сигналов (рис. 9.50). Наличие света соответствует лог. 1, отсутствие света - лог. 0. Каждый бит размещен в отведенном ему временном интервале.

Оптический разветвитель разделяет входной световой поток на 13 равных по интенсивности частей. (На входе устройства можно установить усилитель для повышения мощности потока.) Каждая часть разделенного светового пото-

ка попадает в соответствующее выходное оптоволокно (линию). Верхняя (по схеме) линия (см. рис. 9.48) предназначена для непосредственной передачи сигнала на один из входов фотоприемника $F2$. Задержка передачи сигнала по этой линии условно принимается равной нулю («ноль, умноженный на T » или $0T$; T – длительность битового интервала).

Задержка передачи сигнала по второй – тринадцатой линиям составляет соответственно $1T, 2T, 3T, \dots, 12T$. Задержка сигнала в каждой линии определяется числом последовательно включенных петель из оптоволокна. Каждая петля задерживает сигнал на время, равное длительности одного битового интервала T . Такая структура позволяет развернуть последовательный код в параллельный.

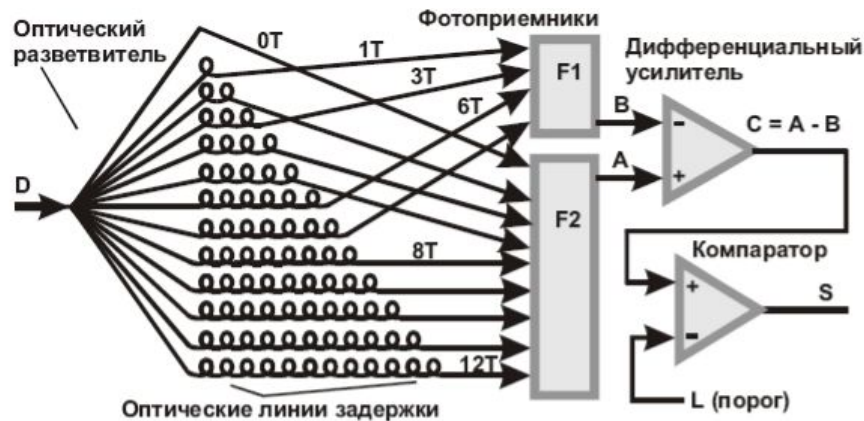


Рис. 9.48. Схема устройства для распознавания флаговой комбинации битов во входном потоке данных D , поступающем по волоконно-оптической линии связи

Действительно, одновременно с поступлением новейшего бита на входы фотоприемников поступают: новый, задержанный на один такт ($1T$); предшествующий ему и задержанный на два такта ($2T$) и т. д. и, наконец, наистарейший бит, задержанный на 12 тактов ($12T$). Таким образом, на входах фотоприемников образуется 13-разрядное окно, через которое можно просматривать последовательность входных данных D . При этом наистарейший бит соответствует принятому на вход устройства в интервале t_0 (см. рис. 9.50), а новейший бит – принятому в интервале t_{12} .

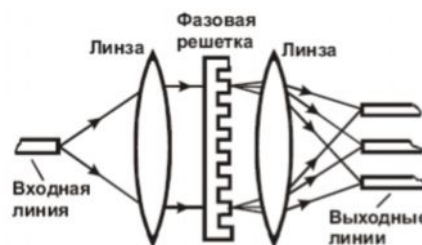


Рис. 9.49. Конструкция оптического разветвителя

Выбранный в данном примере вариант подключения фотоприемников к предыдущим каскадам устройства определяется искомым флаговым кодом

(1111100110101). Девять единичных битов этого кода собираются фотоприемником $F2$, а четыре нулевых – фотоприемником $F1$. Электрический сигнал на выходе фотоприемника пропорционален суммарной мощности световых потоков на его входах.

При поступлении искомого флагового кода фотоприемник $F1$ не получает световой энергии ни по одному из входов. Напротив, фотоприемник $F2$ получает световые потоки от всех входных линий. Такая комбинация сигналов уникальна и сопровождается максимально возможной разностью напряжений между точками A и B на входах дифференциального усилителя (см. рис. 9.50). Остается только зарегистрировать максимальную разность, и задача решена.

Для этого разностный сигнал $C = A - B$ подается на компаратор и сравнивается с заданным порогом L . Сигнал опознавания $S = 1$ формируется при превышении порога сигналом C . Порог выбирается настолько высоким, что сигнал $S = 1$ формируется только при точном опознании флагового кода.

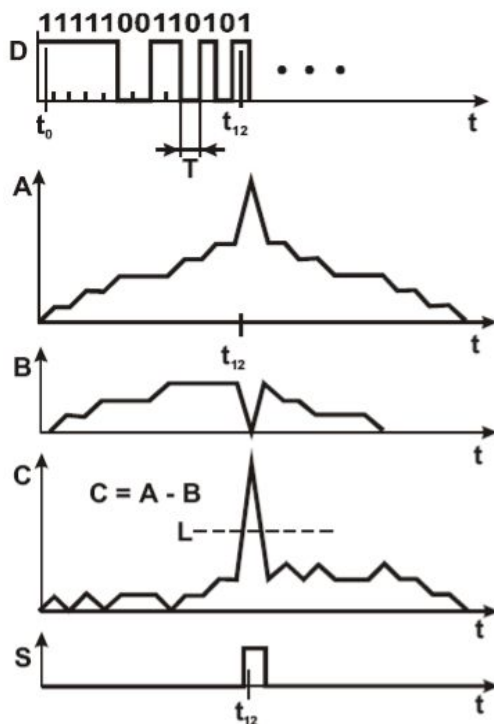


Рис. 9.50. Временные диаграммы сигналов при распознавании флага

Снижение порога позволяет вести не точный, а приближенный поиск нужной последовательности битов во входном потоке. Например, сигнал $S = 1$ при определенном снижении порога может формироваться не только в результате поступления флагового кода (1111100110101), но и любого близкого к нему, содержащего одиночную ошибку (например, 1111100110100, ошибка в младшем разряде). Это может оказаться полезным при повышенном уровне ошибок в канале связи. Полученные результаты приближенного поиска сортируются, и, с учетом предыстории, среди них выявляются наиболее правдоподобные, периодически повторяющиеся, которые интерпретируются как флаги.

При оптимальном выборе флаговых кодов можно получить максимально контрастные картины опознания, при которых пик разностного сигнала C выражен наиболее ярко.

9.2.10. Передача данных вместо избыточных битов синхронизации кадра. Как уже отмечалось, данные, передаваемые по транспортной сети, группируются в структурные единицы, которые в зависимости от используемой технологии называют контейнерами, ячейками, пакетами, кадрами и т. п. В любом случае удаленный узел – приемник данных должен распознать начало передаваемой по сети структурной единицы.

Рассмотрим одну из наиболее простых структурных единиц – кадр, предусмотренный рекомендацией *ITU-T V.110* (рис. 9.51, а).

Кадр построен в виде матрицы из десяти строк и восьми столбцов. В пересечении каждой строки и столбца размещен один бит, так что кадр содержит 80 бит. Первым передается крайний левый бит верхней строки (лог. 0), последним – крайний правый бит B_{63} нижней строки. Биты $B_1 - B_{63}$ представляют собой передаваемые данные; $S_1 - S_8$ – так называемые биты синхронизации.

В начале кадра всегда содержится флаг – кодовая комбинация из восьми лог. 0 и последующей лог. 1 (000000001). Такая комбинация не должна встречаться в любом другом месте кадра; это гарантируется введением битов $S_1 - S_8$ синхронизации, каждый из которых представлен лог. 1. Из приведенного на рис. 9.51, б примера видно, что в каждой строке матрицы, начиная со второй, содержится по крайней мере одна лог. 1, так что флаговая комбинация просто не умещается где-либо в ненадлежащем месте. Как видим, все сделано правильно; но «работа по правилам» не всегда дает наилучший результат. Действительно, левый столбец матрицы является избыточной константой, рассчитанной на наихудший случай, когда все биты $B_1 - B_{63}$ представлены лог. 0 либо среди них имеется незначительное число лог. 1. Нельзя ли уменьшить избыточность левого столбца?

Обратимся к рис. 9.51, в и проанализируем значимость битов синхронизации, начиная с первого S_1 (крайнего левого бита третьей сверху строки матрицы). Он, как и предусмотрено рекомендацией *V.110*, равен лог. 1. Но, просматривая его ближайшее окружение, видим, что ложного опознания флага не произойдет, если биту S_1 присвоить значение, равное лог. 0.

Действительно, в третьей сверху строке матрицы при $S_1 = 0$ появится непрерывная последовательность из четырех лог. 0, но это по-прежнему гарантирует удаленный приемник кадра от ложного опознания флага (не только на своем законном месте). Отметим факт свободы выбора значения бита S_1 , выделив его на рисунке кружочком.

Подобные рассуждения применимы также к битам S_2, S_3, S_5 и S_8 , выделенным на рисунке кружочками. Эти биты могли бы принимать не только единичные, но и нулевые значения, не мешая удаленному приемнику распознать флаг только на своем законном месте – в начале кадра. В то же время биты S_4, S_6 и S_7 (выделены квадратиками) должны безусловно принимать значения лог. 1. В противном случае сформируются достаточно длинные последовательности

из лог. 0 (выделены тремя затененными областями), которые, например, при $S4 = S6 = S7 = 0$ приведут к троекратному ложному опознанию начала кадра в его теле.

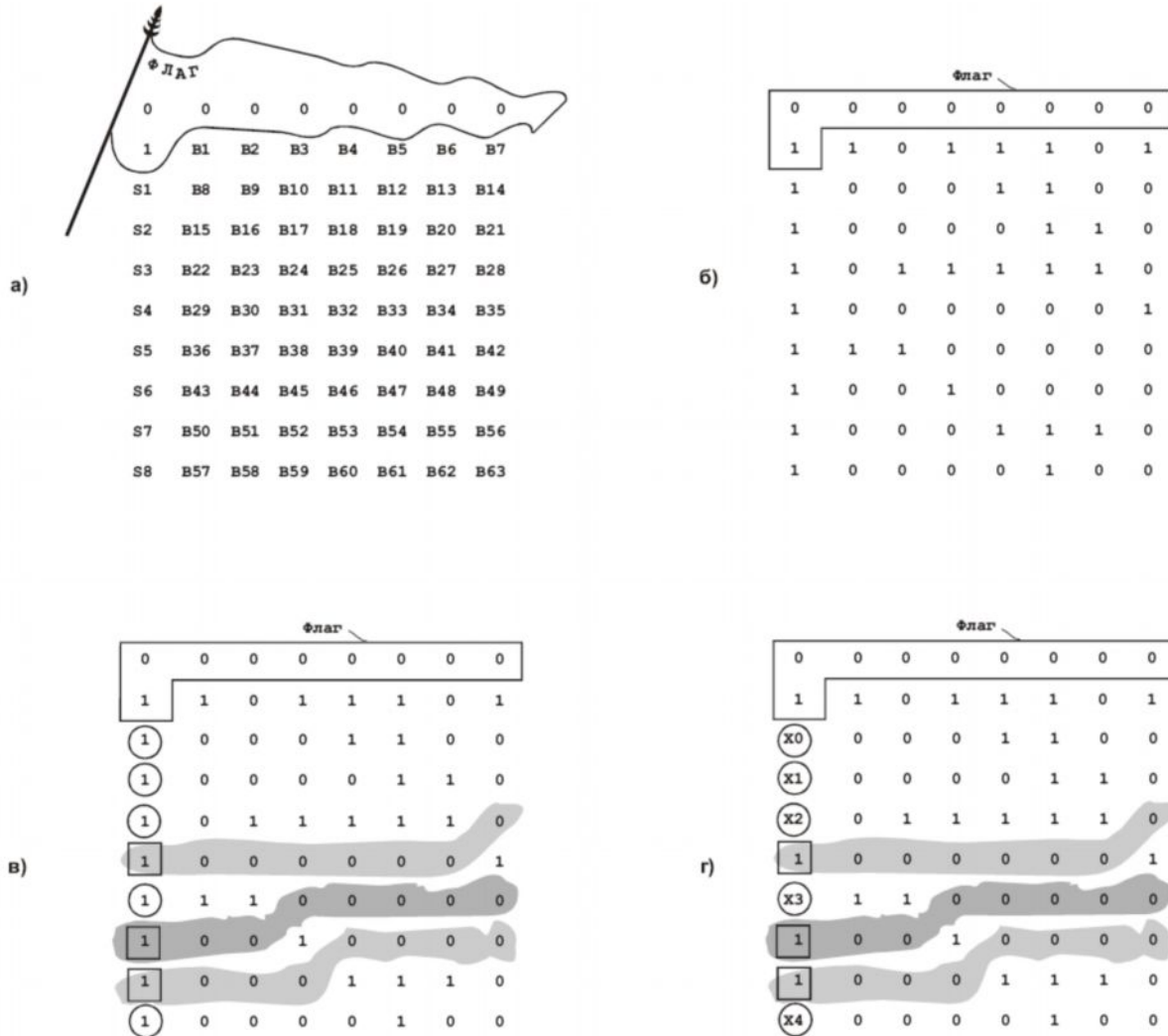


Рис. 9.51. Передача данных вместо избыточных битов синхронизации кадра:
а - общая структура кадра в соответствии с рекомендацией ITU-T V.110; б - кодовый пример кадра; в - поиск битов синхронизации, которые должны безусловно принимать значения лог. 1 (выделены квадратиками); г - определение позиций для передачи данных (X0 - X4) вместо битов синхронизации (выделены кружочками)

Таким образом, становится понятной идея, которую можно сформулировать так: в передаваемом кадре V.110 обычно можно отыскать вакантные места для размещения от одного до восьми битов – своего рода «неучтенных пассажиров» транспортной системы. В нашем примере имеем пять таких битов X0 - X4 (рис. 9.51, г).

Теперь возникает вопрос: как удаленный приемник сможет понять, какие биты полученного кадра обведены кружочками, а какие - квадратиками? Ведь информация о форме обвода в явном виде не передается по линии! Попробуем с этим разобраться.

Алгоритм работы приемника таков.

1. Приемник, постоянно прослушивая линию, обнаруживает поступление флага (кода 000000001), после чего принимает кадр (см. рис. 9.51, а) и записывает его в первой области памяти, выделенной для временного хранения кадров.

2. Приемник запоминает биты $S1 - S8$ во второй области памяти, выделенной для их временного хранения, а на места этих битов в матрице помещает лог. 1. Таким образом, приемник «видит» матрицу, показанную на рис. 9.51, б.

3. Приемник проводит описанный ранее анализ необходимости существования каждого бита синхронизации и, если не было ошибок передачи данных $B1 - B63$, приходит к тем же результатам, которые были приведены на рис. 9.51, в. В нашем примере приемник приходит к выводу, что биты $S4$, $S6$ и $S7$ должны быть представлены лог. 1.

Он проверяет, так ли это, извлекая из второй области памяти соответствующие биты. (Любое несовпадение расценивается как ошибка передачи, и кадр отбрасывается.) Попутно выявляются позиции битов, обведенных кружочками. Теперь остается только извлечь из второй области памяти соответствующие биты ($S1 - S3$, $S5$, $S8$) и скомпоновать из них искомый код $X0 \dots X4$ (см. рис. 9.51, г). Таким образом, «неучтенные пассажиры» благополучно прибыли на место.

Можно предложить много вариантов использования кодов $X0 \dots XL$. Например:

1) коды $X0 \dots XI$ переменной длины, передаваемые в последовательности кадров, можно рассматривать как фрагменты некоторого, в общем случае независимого, массива данных. Иными словами, можно организовать второй, параллельный канал передачи данных, который работает не в ущерб первому, основному;

2) каждый код $X0 \dots XI$ можно рассматривать как некое контролирующее приложение к данным $B1 - B63$, которые передаются в этом же кадре. Например, разряд $X0$ может представлять собой бит контроля по четности всей группы $B1 - B63$. Разряды $X1$ и $X2$ могут соответственно служить битами контроля по четности подгрупп $B1 - B31$ и $B32 - B63$ и т. д. с уменьшением длины контролируемой области.

Иное возможное распределение функций разрядов: бит $X1$ контролирует по четности данные, размещенные в своей и предыдущей строках матрицы. Такое решение позволяет обнаруживать ошибки «на лету», не дожидаясь окончания приема полного кадра.

9.2.11. Уменьшение числа операций, выполняемых при распознавании флага начала кадра. Распознавание многоразрядных флагов начала кадра в потоке данных обычно реализуется с помощью микропроцессорного устройства. Это устройство выполняет некоторую циклическую программу, предусматривающую ряд операций извлечения данных из буферной памяти и их сопоставления с заранее известным флаговым кодом. Уменьшение общего числа таких операций позволяет снизить тактовую частоту микропроцессора и уменьшить интенсивность считывания данных из буферной памяти, что, в свою очередь, снижает потребляемую устройством мощность.

Рассмотрим один из способов уменьшения числа операций, выполняемых при распознавании флага начала кадра. Чтобы показать преимущество этого способа перед традиционным, предположим, что тот и другой реализуются с использованием одной и той же аппаратуры обработки сигнала, поступающего из линии (рис. 9.52, а). Прежде всего, из этого сигнала выделяется битовый поток данных *DATA* и сопровождающий его синхросигнал *CLK*. Далее поток битов преобразуется в поток байтов *D*, сопровождаемый синхросигналом *CLK/8* с выхода делителя частоты на восемь. Поток байтов поступает в буферную память и считывается из нее для дальнейшей обработки (см. штриховую линию на рисунке). Устройство распознавания флага выполнено на основе микропроцессора. Оно анализирует проходящий через буферную память поток данных и при обнаружении флага начала кадра формирует сигнал *FOUND*.

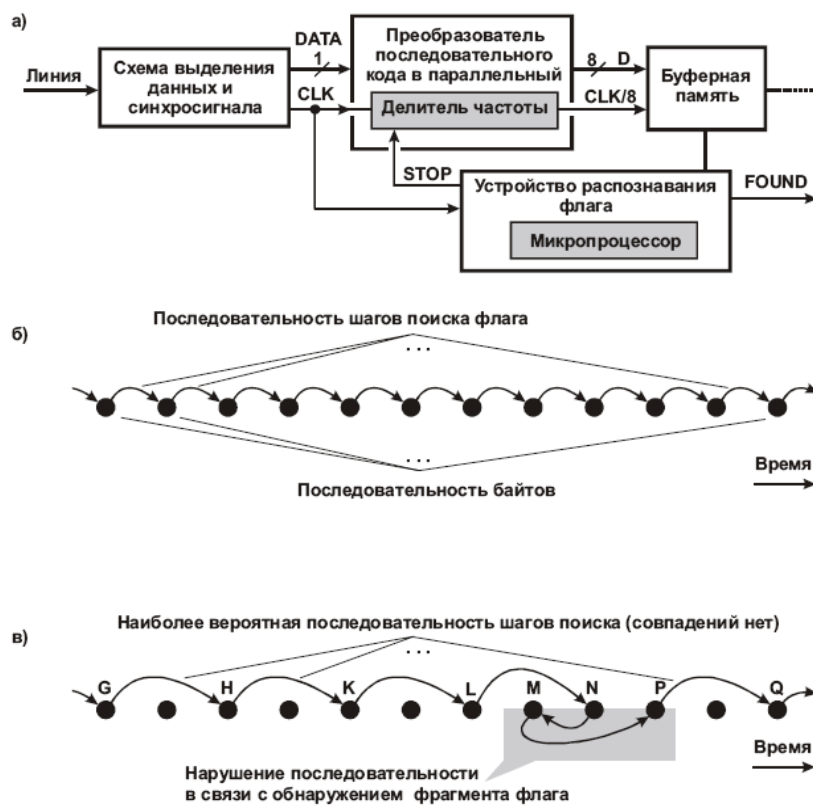


Рис. 9.52. Распознавание флага начала кадра: а - структурная схема аппаратуры обработки сигнала из линии; б, в - диаграммы, соответствующие традиционному и предлагаемому способам поиска

Исходное состояние делителя частоты может быть произвольным, поэтому первоначальное разделение битового потока данных *DATA* на байты *D*, вероятнее всего, неверно, т. е. не совпадает с разделением, выполненным удаленным передатчиком данных в линию. Следствием этого является невозможность регулярного обнаружения флагов начала кадров в потоке байтов *D*, что достижимо только при правильном определении границ байтов.

Чтобы найти истинные границы байтов, устройство распознавания флага проводит серию экспериментов по поиску регулярной последовательности флагов. В каждом эксперименте границы байтов смещаются на один бит, поэтому число экспериментов не превышает семи, а при удачном стечении обстоятельств сразу же выясняется, что предполагаемые границы байтов совпадают с истинными границами. Для того чтобы сместить границы байтов на один бит, устройство распознавания флага формирует сигнал *STOP*, который приостанавливает работу делителя частоты на один такт сигнала *CLK*.

Предположим, что границы байтов в битовом потоке определены правильно, а флаг представлен 32-разрядным кодом

$$0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 1011\ 0011_2 = 00\ 00\ 01\ B3_{16}.$$

Традиционный способ поиска флага предписывает микропроцессору последовательно просматривать байты, размещенные в буферной памяти (рис. 9.52, б) и сравнивать их с байтами кода 00 00 01 B3₁₆, начиная с первого (00₁₆). Успешное распознавание флага сопровождается последовательным совпадением четырех байтов данных с байтами этого кода.

Предлагаемый способ поиска предусматривает проведение быстрого оценочного просмотра последовательности байтов. Просмотр становится более детальным только в окрестностях тех точек (байтов), которые подозреваются на причастность к переносу флага (рис. 9.52, в). В результате число выполняемых операций, условно показанных на рисунке стрелками, уменьшается, что и требовалось. Рассмотрим оба способа поиска подробнее.

Традиционный способ поиска флага описывается диаграммой состояний, приведенной на рис. 9.53.

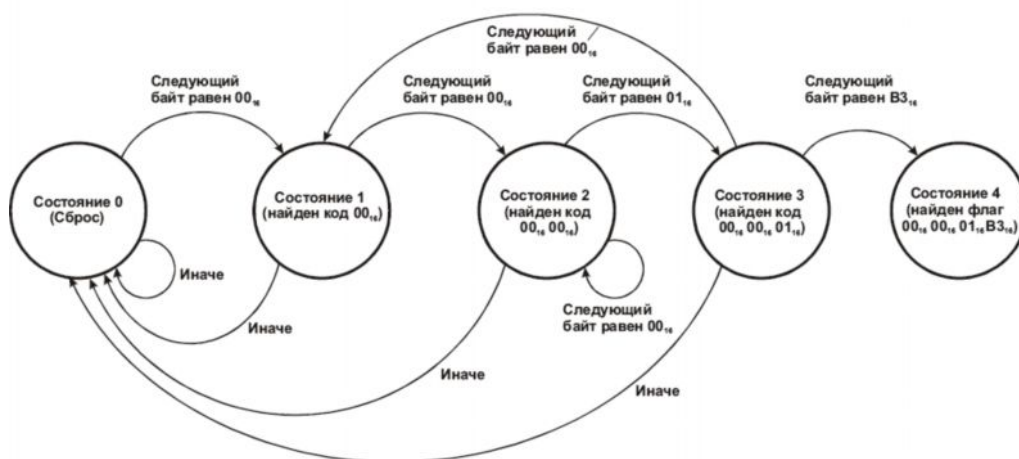


Рис. 9.53 Диаграмма состояний устройства распознавания флага (традиционный способ поиска)

Стрелками помечены разрешенные переходы между состояниями. Схема алгоритма поиска показана на рис. 9.54. Предположим, что в начале поиска устройство находится в состоянии 0. Если очередной байт равен 00₁₆, то осуществляется переход в состояние 1, в противном случае сохраняется состояние 0. При

повторном обнаружении байта 00_{16} происходит переход в состояние 2, в противном случае - в состояние 0 и т. д. Эти же условия соответствуют переходам между блоками 1 - 4 на схеме алгоритма. Обнаружение флага соответствует переходу в состояние 4 или, что тоже самое, выходу из блока 8 по ветви «Да».

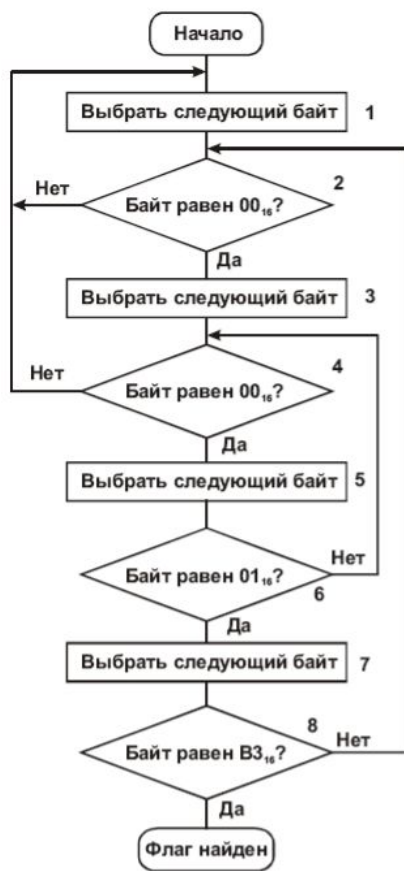


Рис. 9.54. Схема алгоритма поиска флага (традиционный вариант)

Диаграмма состояний и блок-схема предлагаемого алгоритма поиска флага приведены на рис. 9.55 и рис. 9.56.

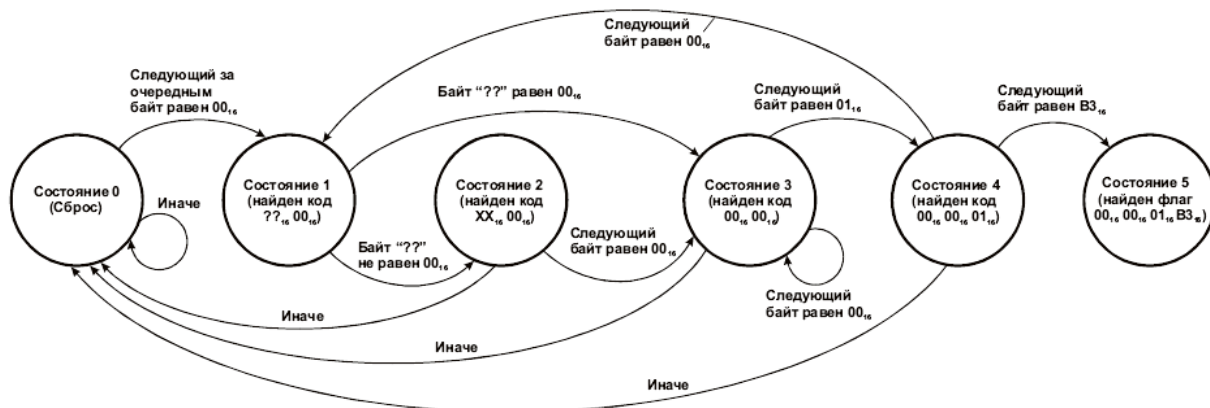


Рис. 9.55. Диаграмма состояний устройства распознавания флага (предлагаемый способ поиска)

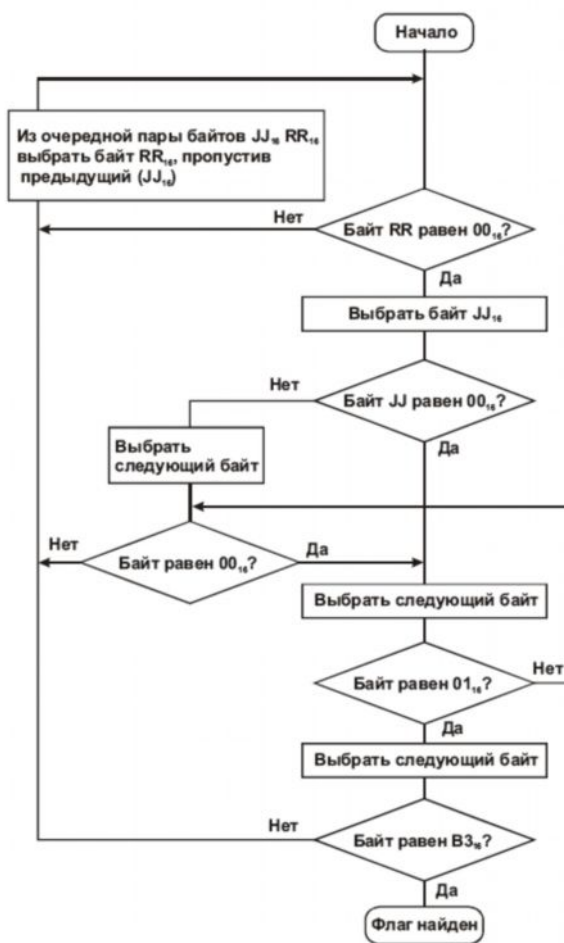


Рис. 9.56.Схема алгоритма поиска флага (предлагаемый вариант)

Отличие предлагаемого способа поиска от традиционного, как уже отмечалось, состоит в том, что вместо полного контроля последовательности байтов проводится ее грубый просмотр, а подробности анализируются лишь по мере необходимости уточнения ситуации. Такой способ поиска правомерен в силу следующих причин.

1. Флаги встречаются в потоке данных сравнительно редко, особенно если они представлены уникальными кодами (в случае применения битстаффинга), а кадры содержат большое число байтов. Поэтому столь же редко возникает необходимость полного анализа четырехбайтового кода в связи с обнаружением флага.

2. Необходимым (но не достаточным) условием обнаружения флага является регистрация двух размещенных рядом нулевых байтов. Поэтому для проверки этого условия можно проверять не все байты подряд, а продвигаться по их последовательности с шагом, равным двум байтам. Данные в кадрах можно рассматривать как случайные – это безусловно справедливо, если применено их скремблирование. Тогда вероятность обнаружения нулевого байта равна $1/256$. Иными словами, ускоренное продвижение по последовательности байтов - скорее норма, чем исключение.

Для пояснения сказанного вернемся к диаграмме, показанной на рис. 9.52, в. В процессе поиска флага последовательно анализируются ненулевые байты G , H , K и L , расположенные в буферной памяти с интервалом в две ячейки. На диаграмме состояний (рис. 9.55) этот этап поиска соответствует неудачным попыткам выхода устройства из состояния 0. После байта L выбирается нулевой байт N , что соответствует переходу устройства в состояние 1. При этом возникает необходимость проверить предыдущий байт M , так как появилась надежда обнаружить флаг. Однако этот байт при проверке оказывается ненулевым, устройство переходит в состояние 2. Далее выбирается байт P , но он также оказывается ненулевым, поэтому устройство возвращается в состояние 0. После этого ускоренный анализ продолжается - выбирается байт Q и т. д.

Таким образом, предлагаемый способ значительно сокращает число обращений к памяти и уменьшает объем вычислительных операций, выполняемых микропроцессором при поиске флага.

10. УСТРОЙСТВА СОПРЯЖЕНИЯ С ЛИНИЯМИ И КАНАЛАМИ

Для гальванической развязки линейных и местных цепей применяют оптроны. Рассмотрим принципы работы входных и выходных устройств.

10.1. Выходное устройство

Упрощенная схема выходного устройства, работающего в двух- и однополюсном режимах, приведена на рис. 10.1. При работе в двухполюсном режиме (перемычки 1 – 2, 5 – 6 и 7 – 8 узла коммутации $УК$) используются две линейные батареи $ЛБ1$ и $ЛБ2$. Сигналы от входного триггера поступают поочередно (в противофазе) на $VX1$ и $VX2$. При подаче сигнала логической 1 на $VX1$ открывается транзистор $VT1$ и через светодиод оптронной пары $ОП1$ течет ток. Включается фотодиод $ОП1$ и открывается составной транзисторный ключ $VT3-VT4$. В результате в линейной цепи протекает ток от $+ЛБ1$, через устройство защиты от короткого замыкания в линии $УЗ$, диод $VD1$, транзистор $VT4$, диод $VD2$, переключку 1 – 2, линию – $ЛБ1$. Аналогично работает нижнее плечо выходного устройства при подаче сигнала логической 1 на $VX2$. В линии течет ток от батареи $ЛБ2$ через транзисторный ключ $VT6$.

При работе в однополюсном режиме (перемычка 3 – 4 на $УК$) сигнал логической 1 поступает одновременно на VXL и $VX2$. В линии течет ток только от $ЛБ1$. При передаче бестоковой посылки ключи $VT4$ и $VT6$ закрыты.

Стабистор $VD5$ служит для предохранения выходных транзисторных ключей от пробоя при работе на линию с большой индуктивностью.

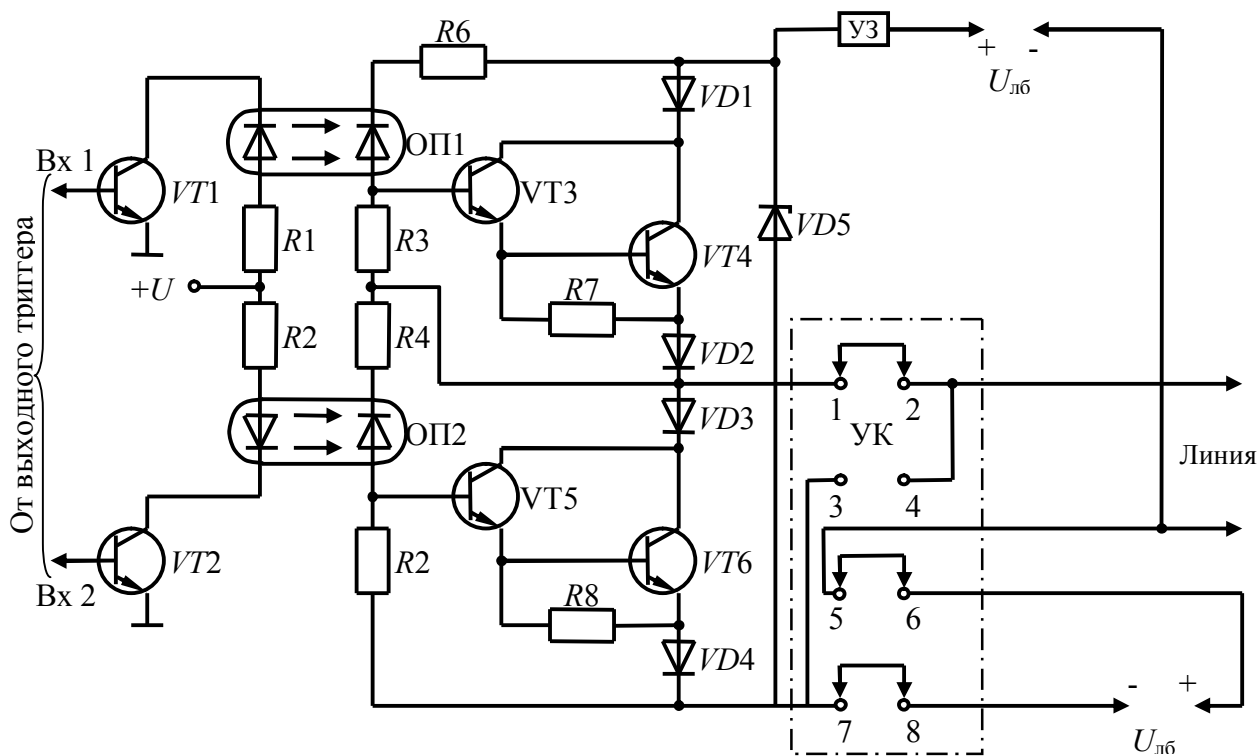


Рис. 10.1. Схема выходного устройства

10.2. Входное устройство

Входное устройство, как и выходное, входит в состав устройства сопряжения с линией (УСЛ) и обеспечивает согласование электрических характеристик канала связи и электронного телеграфного аппарата [17]. Упрощенная схема входного устройства, работающего в однополюсном режиме, приведена на рис. 10.2. Оно содержит следующие основные элементы: линейный фильтр (дроссели $L1$ и $L2$, конденсаторы $C1$ и $C2$), необходимые для ограничения спектра принимаемого сигнала; стабилитроны $VD1$, $VD2$ для защиты от токовых перегрузок; диодный мост $VD3$ - $VD6$, предназначенный для приема токового сигнала любой полярности; оптрон $ОП1$, обеспечивающий гальваническую развязку; оптрон $ОП2$ для установки уровня регистрации; ключевой $VT1$ и выходной $VT2$ транзисторы.

При отсутствии тока на входе по цепи: $+27$ В, резисторы $R8$, $R3$, $R4$, светодиод $ОП2$, резистор $R5$, провод ОБЩ течет постоянный ток, который задает уровень регистрации. Этот уровень можно менять резистором $R4$. Через сопротивление фотодиода оптрона $ОП2$ на базу транзистора $VT2$ подается запирающий потенциал. На вход инвертора $HE1$ поступает потенциал (логическая 1), а на выходе электронного устройства появляется сигнал логического 0.



Упрощенная схема входного устройства, работающего в двухполюсном режиме, приведена на рис. 10.3. При появлении в линии положительного импульса начинает протекать ток по цепи: $Bx1$, дроссель $L1$, $R1$, $R2$, светодиоды оптронных пар $ОП1$ и $ОП2$, диод $VD4$, дроссель $L2$, $Bx2$. Изменяются сопротивления фотодиодов $ОП1$ и $ОП2$, в результате чего транзистор $VT2$ закрывается, а транзистор $VT1$ открывается. На входе инвертора HE появляется сигнал логического 0. В результате на первые входы схем $И1$, $И2$ поступают сигналы логической 1. На второй вход $И1$ подается также сигнал логической 1 с коллектора $VT2$. В результате на выходе $И1$ сигналом логического 0 выходной триггер $ВыхТ$ устанавливается в положение 0 и с его инверсного выхода снимается сигнал логической 1.

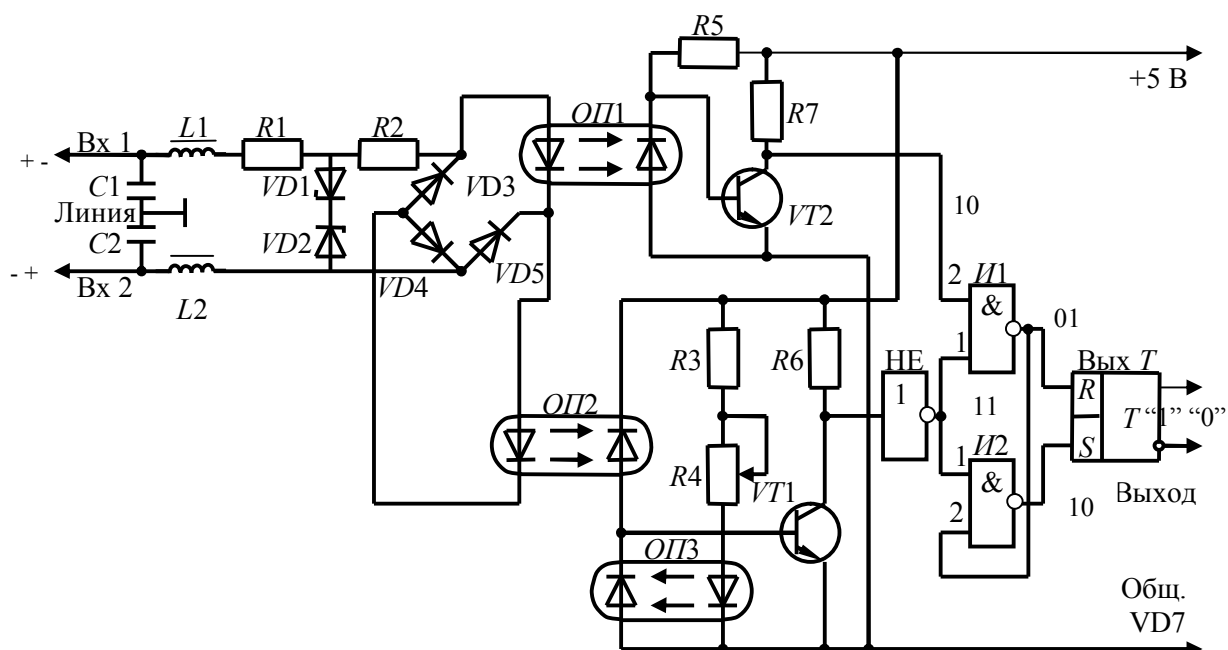


Рис. 10.3. Схема входного устройства, работающего в двухполюсном режиме

При появлении в линии отрицательного импульса начинает протекать ток по цепи: Вх2, дроссель $L2$, диод $VD5$, светодиод оптрона $ОП2$, диод $VD3$, резисторы $R2$, $R1$, дроссель $L1$, Вх1. Так как ток через светодиод оптронной пары $ОП1$ не протекает, то транзистор $VT2$ оказывается открытым и на второй вход схемы $И1$ поступает сигнал логического 0. На выходе этой схемы появляется сигнал логической 1 и устанавливается выходной триггер ВыхТ в положение 1.

С инверсного выхода триггера снимается сигнал логического 0. Резистором $R4$ определяется оптимальный уровень фиксации двухполюсных сигналов.

11. СЕТИ ПЕРЕДАЧИ ДИСКРЕТНЫХ СООБЩЕНИЙ

11.1. Общие сведения

Сетью передачи дискретных сообщений называется вторичная сеть связи, обеспечивающая обмен телеграфными сообщениями, передачу данных, обмен факсимильной информацией и другими видами дискретных сообщений (телеизмерения, телеуправления и др.) между пользователями. Главной задачей сети передачи дискретных сообщений (ПДС) является доставка сообщений по требуемому адресу, в указанные сроки, при заданной вероятности передачи.

В состав технических средств сети ПДС входят центры коммутации различного класса (узлы коммутации УК, концентраторы $K_{Ц}$, мультиплексоры M), каналы связи (соединительные линии, тракты передачи) и разнообразные око-

нечные абонентские пункты АП (аппаратура передачи данных АПД, персональные ЭВМ и др.).

Построение сетей является объёмной самостоятельной задачей, а поэтому в данном параграфе мы остановимся лишь на необходимости введения в сети ассоциативных (объединяющих) систем. Это связано с тем, что время задержки передачи, начиная от некоторого значения коэффициента использования сети, начинает резко расти: 40 – 50% – для сети *EtheRNet*, 60% – для технологии *Token RING* и 70% – до *FDDI*. Та же причина накладывает жёсткие ограничения на максимальную длину сети. Поэтому при увеличении числа пользователей локальной сети и её размеров целесообразно разделить сеть на несколько частей (сегментов) и соединить их повторителями, маршрутизаторами, мостами, коммутаторами (рис. 11.1).

Повторитель (*Repeater*) используется для соединения отдельных сегментов с целью увеличения общей длины сети. По физической сути это обычный регенератор, восстанавливающий сигналы по амплитуде и длительности. Логически повторитель работает на физическом уровне, пропуская сквозь себя сигналы без обработки. Повторитель на несколько выходов (портов) часто называют концентратором (*concentrator*), или хабом (*hub*).

Вторым связующим сетевым элементом является мост (*bridge*), который обеспечивает передачу информации из одного сегмента в другой только в случае необходимости. Логически он работает на втором уровне эталонной модели взаимодействия открытых систем (ЭМ ВОС). Те же функции выполняет коммутатор (*switch*), отличаясь от моста тем, что каждый его порт оснащен специализированным процессором, что позволяет повысить его производительность.

Еще более мощным средством разделения (объединения) сегментов являются маршрутизаторы (*router*). Они образуют логические сегменты, работают на третьем уровне ЭМ ВОС, используя составные числовые адреса; позволяют связывать в единую сеть подсети с разными сетевыми технологиями, например *EtheRNet* и *X.25*.

И, наконец, сети могут объединяться шлюзами (*gateway*). Они применяются в том случае, когда нужно объединить сети с разным типом системного и прикладного программного обеспечения. Логически шлюз работает на уровне У7 ЭМ ВОС, преобразуя протоколы одного класса в другой.

На сегодня крупные сети не строятся без логической структуризации, применяя в разных сочетаниях повторители, мосты, маршрутизаторы, шлюзы.

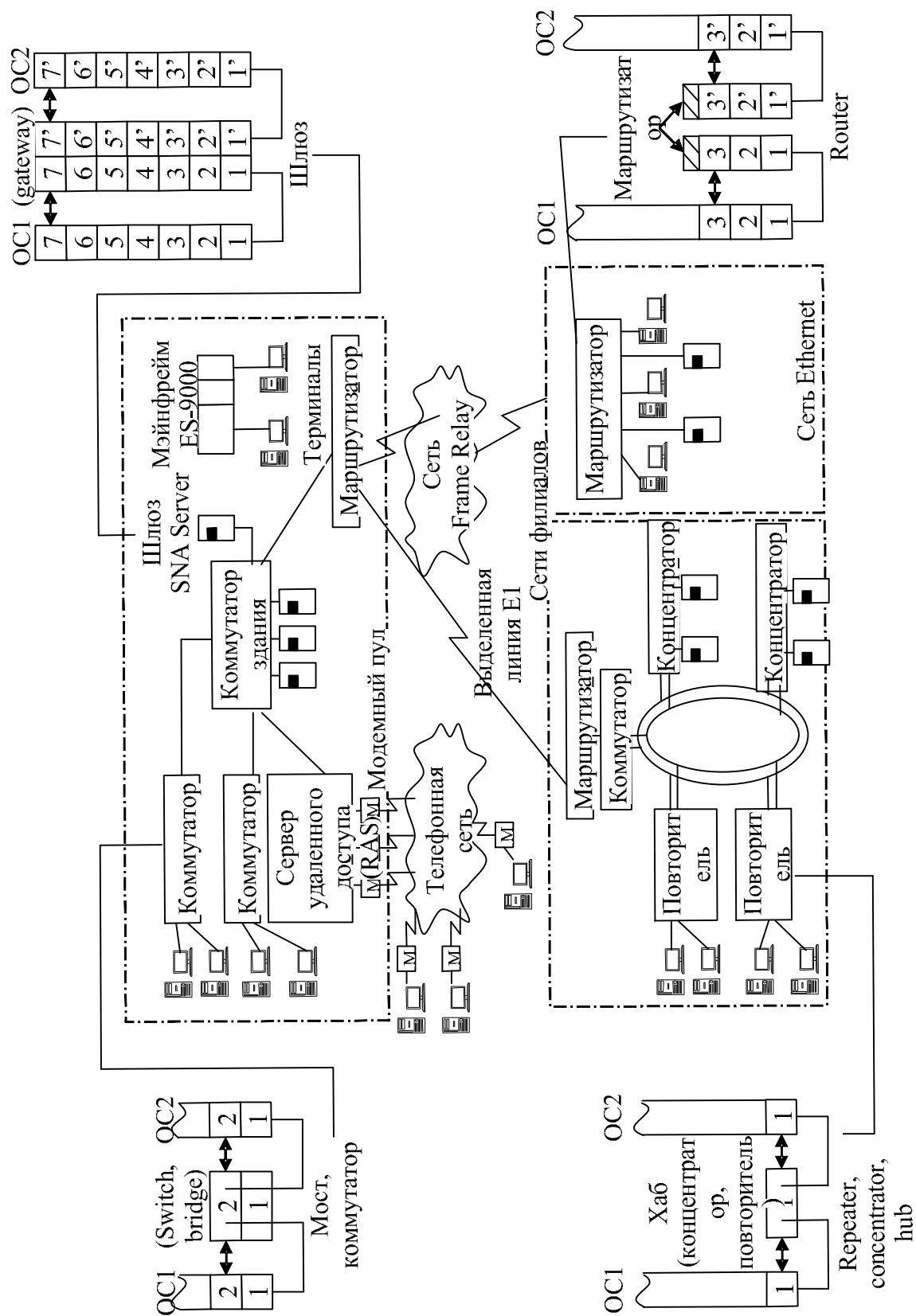


Рис. 11.1.1. Пример разделения сети на сегменты

11.2. Сетевая модель *OSI*

В 1978 году Международный комитет по стандартизации (*ISO*) разработал стандарт архитектуры *ISO 7498*, для объединения различных сетей. В разработке участвовало 7 комитетов, каждому из них был отведен свой уровень. В 1980 году *IEEE* опубликовал спецификацию 802, детально описавшую механизмы взаимодействия физических устройств на канальном и физическом уровнях модели *OSI*. В 1984 году спецификация модели *OSI* была пересмотрена и принята как международный стандарт для сетевых коммуникаций.

Сетевая модель *OSI* (англ. *Open SYstemS INteRcoNnectIoN ReFeReNce MoDeL* – модель взаимодействия открытых систем) – абстрактная модель для сетевых коммуникаций и разработки сетевых протоколов. Представляет уровневый подход к сети. Каждый уровень обслуживает свою часть процесса взаимодействия. Благодаря такой структуре совместная работа сетевого оборудования и программного обеспечения становится гораздо проще и понятнее.

Модель состоит из 7-ми уровней (рис. 11.2), расположенных вертикально друг над другом. Каждый уровень может взаимодействовать только со своими соседями и выполнять отведенные только ему функции.

Прикладной уровень (*AppLicatIoN LaYeR*). Верхний (7-й) уровень модели, обеспечивает взаимодействие сети и пользователя. Уровень разрешает приложениям пользователя доступ к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к **уровню представления**.

Уровень представления (*PReSeNtatIoN LaYeR*). Этот уровень отвечает за преобразование протоколов и кодирования/декодирования данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/раскодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

Сеансовый уровень (*SeSSIoN LaYeR*). Отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия.

Транспортный уровень (*TRaNSpoRt LaYeR*). 4-й уровень модели, предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. При этом неважно, какие данные передаются, откуда и куда, то есть он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, длинные разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка.

Данные	Уровень	Протоколы
Данные	Прикладной доступ к сетевым службам	<i>HTTP, Gopher, TelNet, DNS, SMTP, SNMP, CMIP, FTP, TFTP, SSH, IRC, AIM, NFS, NNTP, NTP, SNTP, XMPP, FTAM, APPC, X.400, X.500, AFP, LDAP, SIP, ITMS, MoDBUS TCP, BACNet IP, IMAP, POP3, SMB, MFTP, BITToRReNt, ED2k, PROFIBUS.</i> Это всего лишь несколько самых распространенных протоколов.
Данные	Представления представление и кодирование данных	<i>HTTP/HTML, ASN.1, XML, TDI, XDR, SNMP, FTP, TelNet, SMTP, NCP, AFP.</i>
Данные	Сеансовый Управление сеансом связи	<i>ASP, ADSP, DLC, NameD PipeS, NBT, NetBIOS, NWLINK, PRINteR AcceSS PRotocol, ZoNe INFoRmatIoN PRotocol, SSL, TLS, SOCKS.</i>
Блоки	Транспортный безопасное и надежное соединение точка-точка	<i>TCP, UDP, NetBEUI, AEP, ATP, IL, NBP, RTMP, SMB, SPX, SCTP, DCCP, RTP, STP, TFTP.</i>
Пакеты	Сетевой Определение пути и IP (логическая адресация)	<i>IP, IPV6, ICMP, IGMP, IPX, NWLINK, NetBEUI, DDP, IPSec, ARP, RARP, DHCP, BootP, SKIP.</i>
Кадры	Канальный MAC и LLC (Физическая адресация)	<i>ARCNet, ATM, DTM, SLIP, SMDS, EtherNet, FDDI, FRAME ReLaY, LocalTaLk, Token RING, StaRLaN, WIFI, L2F, L2TP, PPTP, PPP, PPPoE, PROFIBUS.</i>
Биты	Физический кабель, сигналы, бинарная передача	<i>RS-232, RS-422, RS-423, RS-449, RS-485, ITU-T, XDSL, ISDN, T-caRRieR (T1, E1), модификация стандарта EtherNet: 10BASE-T, 10BASE2, 10BASE5, 100BASE-TX, 100BASE-FX, 100BASE-T, 1000BASE-T, 1000BASE-TX.</i>

Рис. 11.2. Модель OSI

Сетевой уровень (*NetWoRk LaYeR*). 3-й уровень сетевой модели *OSI*, предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имен в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию пакетов, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

Канальный уровень (*DATA LiNk LaYeR*). Этот уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками, которые могут возникать. Полученные с физического уровня данные он упаковывает в кадры данных, проверяет на целостность, если нужно исправляет ошибки и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием. Спецификация *IEEE 802* разделяет этот уровень на 2 подуровня – *MAC (MeDia AcceSS CoNtRoL)* регулирует доступ к разделяемой физической среде, *LLC (LoGicaL LiNk CoNtRoL)* обеспечивает обслуживание сетевого уровня. На этом уровне работают коммутаторы, мосты и сетевые адаптеры.

В программировании этот уровень представляет драйвер сетевой платы, в операционных системах имеется программный интерфейс взаимодействия канального и сетевого уровня между собой, это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: *ODI*, *NDIS*.

Физический уровень (*PhYSicaL LaYeR*). Самый нижний уровень модели, предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель и соответственно их прием и преобразование в биты данных в соответствии с методами кодирования цифровых сигналов. Другими словами, осуществляет интерфейс между сетевым носителем и сетевым устройством. На этом уровне работают концентраторы и повторители (ретрансляторы) сигнала.

Взаимодействие уровней. Уровни взаимодействуют сверху вниз и снизу вверх посредством интерфейсов и могут еще взаимодействовать с таким же уровнем другой системы с помощью протоколов. Подробнее можно посмотреть на рисунке.

Модель *OSI* и реальные протоколы. Семиуровневая модель *OSI* является теоретической, и содержит ряд недоработок. Реальные сетевые протоколы вынуждены отклоняться от нее, обеспечивая непредусмотренные возможности, поэтому привязка некоторых из них к уровням *OSI* является несколько условной.

Основная недоработка *OSI* – непродуманный транспортный уровень. На нем *OSI* позволяет обмен данными между приложениями (вводя понятие *порта* – идентификатора приложения), однако, возможность обмена простыми датаграммами в *OSI* непредусмотрена – транспортный уровень должен образовывать соединения, обеспечивать доставку, упорядочивать поток и т.п. Реальные же протоколы реализуют такую возможность.

Семейство *TCP/IP*. Семейство *TCP/IP* имеет два транспортных протокола: *TCP*, полностью соответствующий *OSI*, и *UDP*, отвечающий транспортному

уровню только наличием порта, обеспечивающий обмен датаграммами между приложениями.

Семейство *IPX/SPX*. В семействе *IPX/SPX*, порты (называемые «сокет» или «гнезда») появляются в протоколе сетевого уровня *IPX*, обеспечивая обмен датаграммами между приложениями (операционная система резервирует часть сокетов для себя). Протокол *SPX*, в свою очередь, дополняет *IPX* всеми остальными возможностями транспортного уровня в полном соответствии с *OSI*.

Кроме того, *IPX* не имеет адреса для хоста, полагаясь на адресацию канального уровня (например, MAC-адреса для *EtherNet*).

11.3. Маршрутизатор

Маршрутизатор или **роутер** (от англ. *RoUteR*) сетевое устройство, используемое в компьютерных сетях передачи данных, которое, на основании информации о топологии сети (таблицы маршрутизации) и определенных правил, принимает решения о пересылке пакетов сетевого уровня модели *OSI* их получателю. Обычно применяется для связи нескольких сегментов сети.

Принцип работы. Традиционно, маршрутизатор использует таблицу маршрутизации и адрес получателя, который находится в пакетах данных для дальнейшей передачи данных. Выделяя эту информацию, он определяет по таблице маршрутизации путь, по которой следует передать данные и направляет пакет по этому маршруту. Если в таблице маршрутизации для адреса нет описанного маршрута, пакет отбрасывается.

Существуют другие способы определения маршрута пересылки пакетов, когда, например, используется адрес отправителя, используемые протоколы верхних уровней и другая информация, содержащаяся в заголовках пакетов сетевого уровня. Нередко маршрутизаторы могут осуществлять трансляцию адресов отправителя и получателя (англ. *NAT, NetWoRk ADDReSS TRaNSLatIoN*), фильтрацию транзитного потока данных на основе определенных правил с целью ограничения доступа, шифрование/дешифрование переданных данных и т.д.

Таблица маршрутизирования. Таблица маршрутизирования содержит информацию, на основе которой маршрутизатор принимает решение о дальнейшей пересылке пакетов. Таблица состоит из некоторого числа записей - маршрутов, в каждой из которых содержится адрес сети получателя, адрес следующего узла, которому следует передать пакеты и некоторый вес записи - метрика.

Метрики записей в таблице играют роль в вычислении кратчайших маршрутов к различным получателям. В зависимости от модели маршрутизатора и используемых протоколов маршрутизации, в таблице может содержаться некоторая дополнительная информация. Например:

192.168.64.0/30 [110/49] *V*_{1a} 192.168.1.2, 00:34:34, *FaStEtheRNet*0/0.1

где 192.168.64.0/30 - сеть назначения,

110/49 - метрика маршрута,

192.168.1.2 - адрес следующего маршрутизатора, которому следует передать пакеты для сети 192.168.64.0/30,

00:34:34 - время, в течение которого был известен этот маршрут,

*FaStEtheRNet*0/0.1 - интерфейс маршрутизатора, через который можно достичь «соседа» 192.168.1.2.

Таблица маршрутизации может составляться двумя способами:

- **статическая маршрутизация** – когда записи в таблице вводятся и изменяются вручную. Такой способ требует вмешательства администратора каждый раз, когда происходят изменения в топологии сети. С другой стороны, он является наиболее стабильным и требующим минимум аппаратных ресурсов маршрутизатора для обслуживания таблицы.

- **динамическая маршрутизация** – когда записи в таблице обновляются автоматически при помощи одного или нескольких протоколов маршрутизации – *RIP*, *OSPF*, *EIGRP*, *IS-IS*, *BGP*, *HSRP* и др. В этом случае любые изменения в топологии сети автоматически отображаются в таблице маршрутизации. Кроме того, маршрутизатор строит таблицу оптимальных путей к сетям назначения на основе различных критериев – количества промежуточных узлов, пропускной способности каналов, задержки передачи данных т. п. Критерии вычисления оптимальных маршрутов чаще всего зависят от протокола маршрутизации, а также задаются конфигурацией маршрутизатора. Такой способ построения таблицы позволяет автоматически держать таблицу маршрутизации в актуальном состоянии и вычислять оптимальные маршруты на основе текущей топологии сети. Однако, динамическая маршрутизация оказывает дополнительную нагрузку на устройства, а высокая нестабильность сети может приводить к ситуациям, когда маршрутизаторы не успевают синхронизировать свои таблицы, что приводит к противоречивым сведениям о топологии сети в различных ее частях и потере передаваемых данных.

Применение. Маршрутизаторы помогают уменьшить загрузку сети, благодаря ее разделению на домены коллизий и широковещательные домены, а также фильтрации пакетов. В основном их применяют для объединения сетей разных типов, зачастую несовместимых по архитектуре и протоколам, например для объединения локальных сетей *EtheRNet* и *WAN*-соединений, использующих протоколы *DSL*, *PPP*, *ATM*, *FRAME* *ReLaY* и т. д. Нередко маршрутизатор используется для обеспечения доступа из локальной сети в глобальную сеть Интернет, осуществляя функции трансляции адресов и межсетевого экрана.

В качестве маршрутизатора может выступать как специализированное устройство, так и РС компьютер, выполняющий функции простейшего роутера.

ЛИТЕРАТУРА

1. Маковеева М.М., Шинаков Ю.С. Системы и средства связи с подвижными объектами. - М.: Радио и связь, 2002.
2. Сорока Н.И., Кривинченко Г.А. Телемеханика: Конспект лекций для студентов специальности Т.11.01.00 "Автоматическое управление в технических системах". Ч.1: Сообщения и сигналы. – Мн.: БГУИР, 2000. – 128 с.
3. Гаранин М.В. , В.Н. Журавлев, С.В. Кунегин. Системы и сети передачи информации. – М.: Радио и связь, 2001. – 336 с.
4. Журавлев В.Н. Поиск и синхронизация в широкополосных системах. – М.: Радио и связь, 1986. – 240 с.
5. Бэрлекамп Э. Алгебраическая теория кодирования. Пер. с англ. / Под ред. С.Д. Бирмана. – М.: Мир, 1971. – 477с.
6. Варакин Л.Е. Системы связи с шумоподобными сигналами. – М.: Радио и связь, 1985. – 348с.
7. Кислов В.Я. и др. Корреляционные свойства шумоподобных сигналов, генерируемых системами с динамическим хаосом // Радиотехника и электроника, 1997. Том 42, № 11. С. 1341 – 1349.
- 8.Смирнов Н.И., Горгадзе С.Ф. Синхронное кодовое разделение абонентских станций: перспективное поколение персональных систем связи // Технологии и средства связи. 1998. № 4. С. 58 – 62.
9. Горностаев Ю.М. Мобильные системы третьего поколения. Москва, МЦНТИ. 1998.
10. Пеннин П.Н. Системы передачи цифровой информации. Учебное пособие для вузов. М.: Сов. радио, 1976. – 368 с.
11. Сорока Н.И., Кривинченко Г.А. Телемеханика: Конспект лекций для студентов специальностей 53 01 03 "Автоматическое управление в технических системах" и 53 01 07 "Информационные технологии и управление в технических системах". Ч.3: Линии связи и помехоустойчивость информации. – Мн.: БГУИР, 2004.
12. Хелд Г. Технологии передачи данных. 7-е изд. – Питер, 2003. – 720 с.
13. Петраков А.В., Лагужин В.С. Телеохрана. – М.: Энергоиздат. 1998.
14. Сорока Н.И., Кривинченко Г.А. Телемеханика: Конспект лекций для студентов специальности "Автоматическое управление в технических системах". Ч.2: Коды и кодирование. – Мн.: БГУИР, 2001. – 168 с.
15. Сорока Н.И., Кривинченко Г.А. Телемеханика: Конспект лекций для студентов специальностей 53 01 03 "Автоматическое управление в технических системах" и 53 01 07 "Информационные технологии и управление в технических системах". Ч.4: Системы телемеханики. – Мн.: БГУИР, 2005. – с.
16. Сорока Н.И., Кривинченко Г.А. Теория передачи информации: Конспект лекций для студентов специальности Т.11.01.00 "Автоматическое управление в технических системах". – Мн.: БГУИР, 1998. – 88 с.
17. Кудрашов В.А., Глушко В.П. Системы передачи дискретной информации – М.: УМК МПС, 2002. – 384 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. ЦИФРОВЫЕ МЕТОДЫ МОДУЛЯЦИИ	4
1.1. Общие сведения.....	4
1.2. М-ичная фазовая модуляция	7
1.3. М-ичная частотная модуляция.....	21
1.4. М-ичная амплитудная модуляция.....	33
1.5. Комбинированные методы модуляции	33
1.6. Треллис-модуляция	44
1.7. Принцип работы треллис-кодера	48
1.8. Декодер Витерби.....	52
1.9. Треллис-кодирование 4D/PAM-5	59
2. ШИРОКОПОЛОСНЫЕ СИГНАЛЫ	62
2.1. Основные термины и определения	62
2.2. Методы расширения спектра частот передаваемых сообщений.....	65
2.3. Псевдослучайные бинарные последовательности (ПСП).....	80
2.4. Методы приёма сложных сигналов	95
2.5. Примеры применения сложных сигналов в цифровых системах передачи данных.....	99
3. ФОРМИРОВАНИЕ ТРЕБУЕМОГО ЭНЕРГЕТИЧЕСКОГО СПЕКТРА ДАННЫХ	111
3.1. Линейные коды для передачи информации по последовательным каналам связи	111
3.2. Трехуровневое кодирование сигнала с гарантированным изменением уровней между соседними битовыми интервалами.....	119
3.3. Кодирование PAM-5	120
4. КАНАЛЫ, РЕЖИМЫ И ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ	144
4.1. Классификация каналов	144
4.2. Виды связи и режимы передачи данных	155
4.3. Технологии DSL.....	168
4.4. Технологии мультиплексирования	172
4.5. Асинхронный режим передачи	195
4.6. Соотношение между скоростью передачи и шириной полосы.....	201
5. МЕТОДЫ ПОВЫШЕНИЯ КАЧЕСТВА РАБОТЫ СИСТЕМ ПЕРЕДАЧИ ДИСКРЕТНОЙ ИНФОРМАЦИИ	205
5.1. Общие сведения.....	205
5.2. Методы перемежения	206
5.3. Автоматический запрос повторной передачи	210
6. СЖАТИЕ СООБЩЕНИЙ.....	214
6.1. Типы систем сжатия	214
6.2. Основные алгоритмы сжатия без потерь информации.....	218
7. УСТРОЙСТВО ЗАЩИТЫ ОТ ОШИБОК (УЗО)	230
7.1. Основные типы УЗО.....	230
7.2. Алгоритм функционирования УЗО.....	231

7.3. Структурная схема передающей части УЗО	232
7.4. Структурная схема приёмной части УЗО	235
8. МОДЕМЫ	237
8.1. Общие сведения	237
8.2. Элементы синхронного модема для телефонной сети общего пользования (ТфОП)	243
8.3. Приёмник асинхронного модема	246
9. МЕТОДЫ И УСТРОЙСТВА СИНХРОНИЗАЦИИ	250
9.1. Тактовая синхронизация	251
9.2. Синхронизация на уровне распознавания и обработки кадров и ячеек (кадровая синхронизация)	281
10. УСТРОЙСТВА СОПРЯЖЕНИЯ С ЛИНИЯМИ И КАНАЛАМИ	316
10.1. Выходное устройство	316
10.2. Входное устройство	317
11. СЕТИ ПЕРЕДАЧИ ДИСКРЕТНЫХ СООБЩЕНИЙ	319
11.1. Общие сведения	319
11.2. Сетевая модель <i>OSI</i>	322
ЛИТЕРАТУРА	338