

## 8. Способы кодирования данных для их передачи по каналу связи

В начале этой главы (п. 8.1) рассмотрены некоторые традиционные методы и схемы кодирования данных для их передачи по каналу связи. Эти методы хорошо известны и приводятся лишь для пояснения содержания решаемых задач.

Одной из таких задач является обеспечение надежного восстановления синхросигнала приемником. Для этого нужно так закодировать данные, чтобы сигнал в линии изменялся как можно чаще, в идеальном случае – в каждом битовом интервале. В одном из возможных решений (п. 8.2) применено трехуровневое кодирование сигнала. Данное решение интересно тем, что при сравнительно низкой частоте следования импульсов в линии, созданы гарантии изменения уровня сигнала при переходе от одного битового интервала к другому независимо от вида передаваемой последовательности битов.

В п. 8.3 и 8.4 рассмотрены методы и схемы кодирования сигнала с использованием скремблирования, что позволяет разровнять его спектр и тем самым снизить уровень излучаемых помех, а также сократить возможные периоды отсутствия изменений сигнала в линии, что важно для повышения надежности синхронизации.

В п. 8.5 рассмотрены известные и предлагаемые методы и схемы передачи данных по группе витых пар проводов.

### 8.1. Основные способы кодирования цифровой информации для ее передачи по последовательным каналам связи

#### 8.1.1. 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. 8.1) [21, 71].

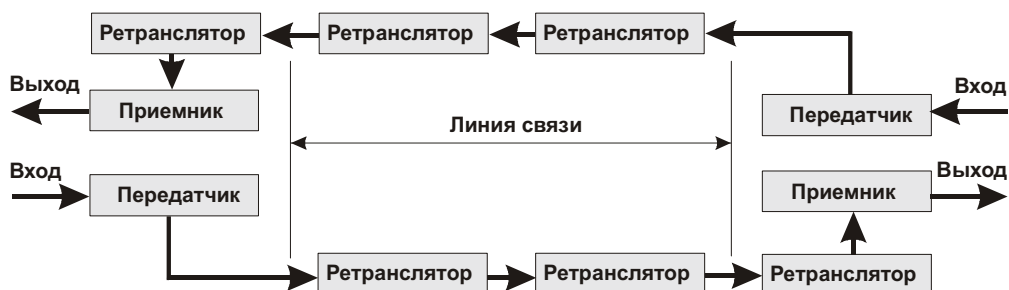


рис. 8.1. Структура типового последовательного канала связи

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

### 8.1.2. Униполярный код NRZ

Простейшим линейным кодом является униполярный код типа NRZ (Non Return to Zero), показанный на рис. 8.2, 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. 8.2, б) при той же амплитуде сигнала, равной  $A$ !

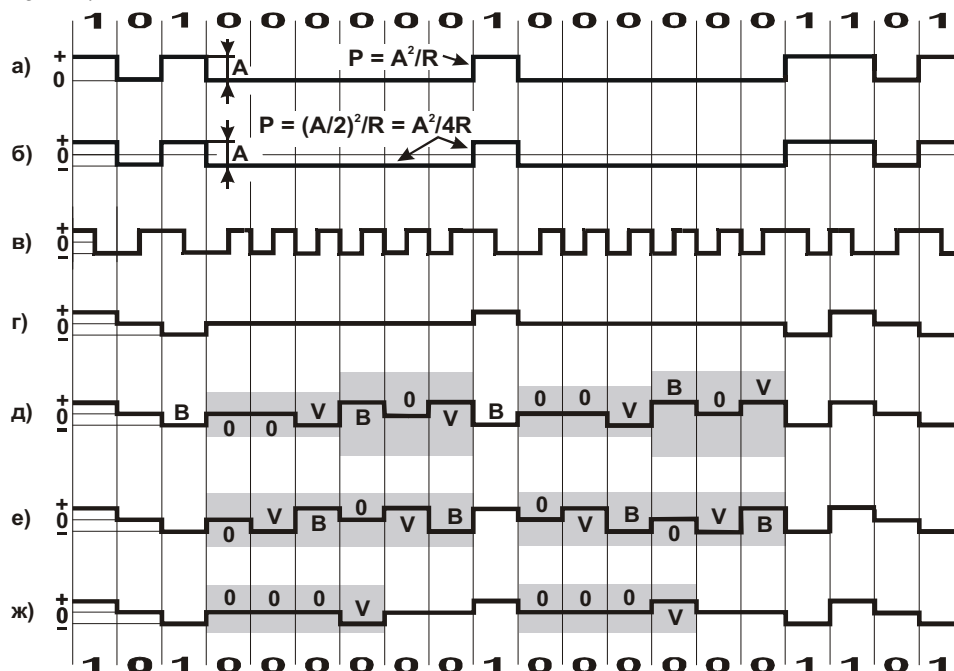


рис. 8.2. Наиболее распространенные линейные коды: а – униполярный код NRZ; б – биполярный код NRZ; в – код Манчестер-II; г – код AMI; д – код B3ZS; е – код B6ZS; ж – код HDB3; затемненными прямоугольниками выделены “заготовки”

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.

### 8.1.3. Биполярный код NRZ

Биполярный сигнал NRZ (рис. 8.2, 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.

### 8.1.4. Код Манчестер-II

Примером кода с избыточностью, введенной согласно только что упомянутому первому способу, является код Манчестер-II. Форма биполярного сигнала при передаче кода Манчестер-II показана на рис. 8.2, 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-II 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.

### 8.1.5. Код AMI

Второй способ введения избыточности связан с добавлением дополнительных электрических уровней, в простейшем случае – третьего, “нулевого”, уровня.

На рис. 8.2, г представлена форма сигнала с попеременной инверсией знака, так называемого AMI сигнала (Alternative Mark Inversion). Нули кодируются отсутствием импульсов, а единицы – попеременно положительными и отрицательными импульсами. Постоянная составляющая сигнала AMI 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.

### 8.1.6. 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 AMI, в которых цепочка из  $N$  нулей заменяется определенной подстановкой, называются BNZS-кодами (Bipolar with  $N$  Zeroes Substitution).

В коде B3ZS (рис. 8.2, *д*) каждые три последовательных нуля подменяются либо комбинацией B0V, либо 00V. Символ B обозначает импульс, который отвечает правилам кодирования AMI, символ V - импульс, который нарушает правила кодирования AMI (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. 8.2, *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 V00V. 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. 8.2, *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.

## 8.2. 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 проводниками 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. 8.3.

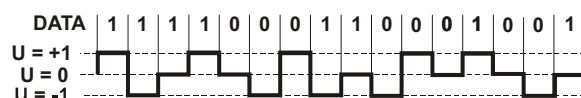


рис. 8.3. Временная диаграмма сигнала в линии

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. Но вскоре мы убедимся, что правила кодирования и декодирования очень просты.

Как следует из рис. 8.4, передатчик содержит двухразрядный регистр RG1, логическую схему L1 и формирователь S трехуровневого сигнала. Приемник содержит преобразователь R трехуровневого сигнала в двухуровневые (лог. 0, лог. 1), двухразрядный регистр RG2 и логическую схему L2.

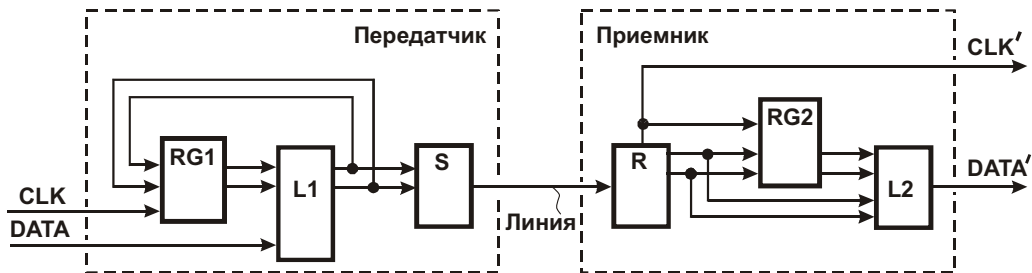


рис. 8.4. Система передачи данных

В начале очередного битового интервала по фронту синхросигнала CLK в регистре RG1 фиксируется двухразрядный код, сформированный логической схемой L1 в предыдущем битовом интервале. С незначительной задержкой, достаточной для надежной фиксации кода в регистре RG1, на вход передатчика подается очередной бит данных DATA. В дальнейшем на протяжении битового интервала на входах логической схемы L1 присутствует результат обработки предыдущего бита (код, отображающий предыдущее состояние передатчика) и очередной бит данных. Логическая схема L1 на основе анализа входной комбинации сигналов формирует двухразрядный код, который определяет новое состояние передатчика. В зависимости от сочетания сигналов на выходе логической схемы L1 формирователь S трехуровневого сигнала выдает в провода линии нулевое, положительное или отрицательное напряжение.

Переходы передатчика между тремя возможными состояниями можно проследить по диаграмме, приведенной на рис. 8.5.

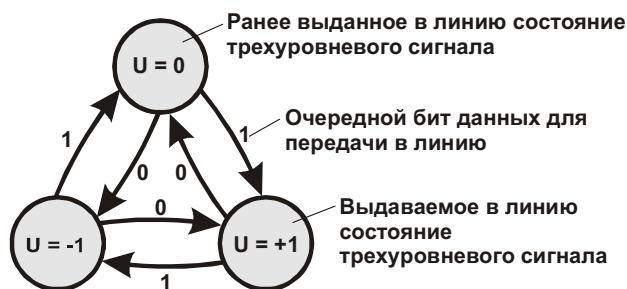


рис. 8.5. Диаграмма состояний передатчика

Передатчик может находиться в трех состояниях, выделенных кружками. Эти состояния обозначены в соответствии с принятыми ранее сокращениями (см. рис. 8.3).

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.

Преобразователь  $R$  трехуровневого сигнала в двухуровневый (см. рис. 8.4) формирует двухразрядный код текущего состояния сигнала в линии и выделяет синхросигнал на основе регистрации фронтов импульсов. В начале очередного битового интервала в регистре  $RG2$  фиксируется предыдущее состояние линии, так что логическая схема  $L2$  оперирует предыдущим и текущим состояниями трехуровневого сигнала. В зависимости от их комбинации можно сделать однозначный вывод о том, какой бит (лог. 0 или лог. 1) поступил на вход приемника.

Декодирование сигналов в приемнике поясняется той же диаграммой, что и предыдущая, но с несколько иной интерпретацией событий (рис. 8.6).

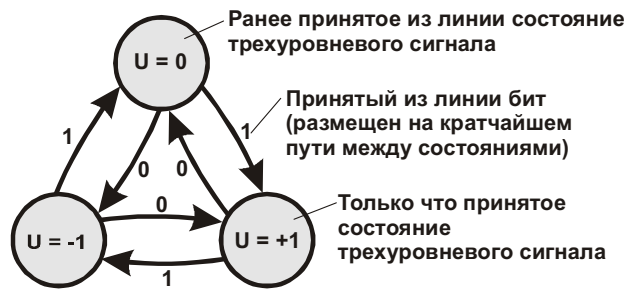


рис. 8.6. Диаграмма состояний приемника

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 (см. п. 8.4).

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

#### 8.3.1. 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 АМІ, Т1 или МLТ-3 при передаче длинной последовательности лог. 1, как показано затененными областями на рис. 8.7.

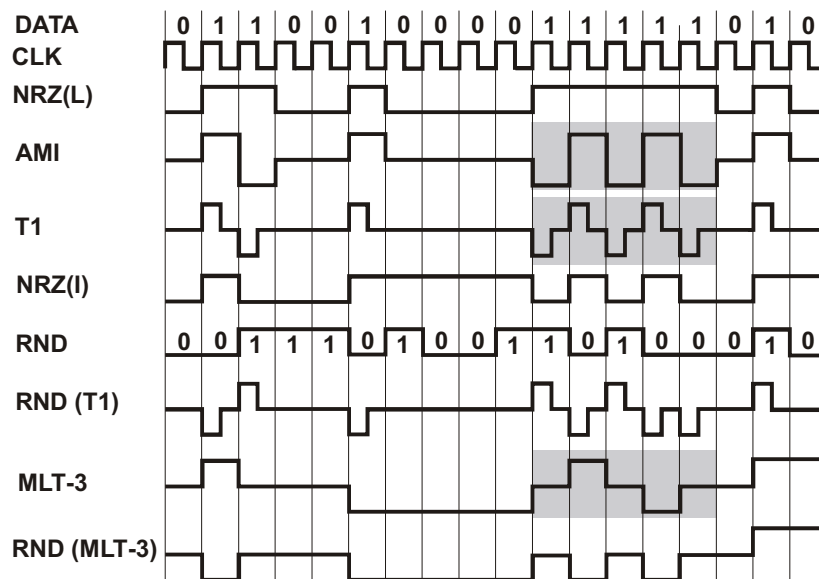


Рис. 8.7. Временные диаграммы передачи данных DATA с использованием различных кодов; RND – сигнал на выходе генератора псевдослучайной последовательности битов

В этих областях невооруженным глазом просматриваются прообразы синусоидальных сигналов, несущих основную энергию. Периоды сигналов АМІ и Т1 при передаче длинной последовательности лог. 1 равны двум битовым интервалам. Период сигнала МLТ-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. 8.4). Для восстановления исходных данных приемник должен выполнить обратную операцию (дескремблирование). При этом необходима синхронная работа шифратора и дешифратора, что несколько усложняет задачу.

Предлагаемое в [26] 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 дескремблирования” (что на первый взгляд представляется лишенным

смысла ☺). В итоге упрощается аппаратура, предназначенная для уменьшения излучаемых помех.

Чтобы перейти к существу вопроса, рассмотрим временные диаграммы, приведенные на рис. 8.7, более подробно.

Как уже отмечалось в п. 8.1, 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 примыкающий 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. 8.8, 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 “соствязаний” 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. 8.8, 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 Искключающее ИЛИ в цепях обратных связей [65]. 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(T1), показанная на рис. 8.8, в, построена аналогично и отличается наличием дополнительного логического элемента И, предназначенного для укорочения положительных импульсов кода NRZ(I).

Схема, представленная на рис. 8.8, 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 (см. гл. 9).

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 пода-



вить 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.

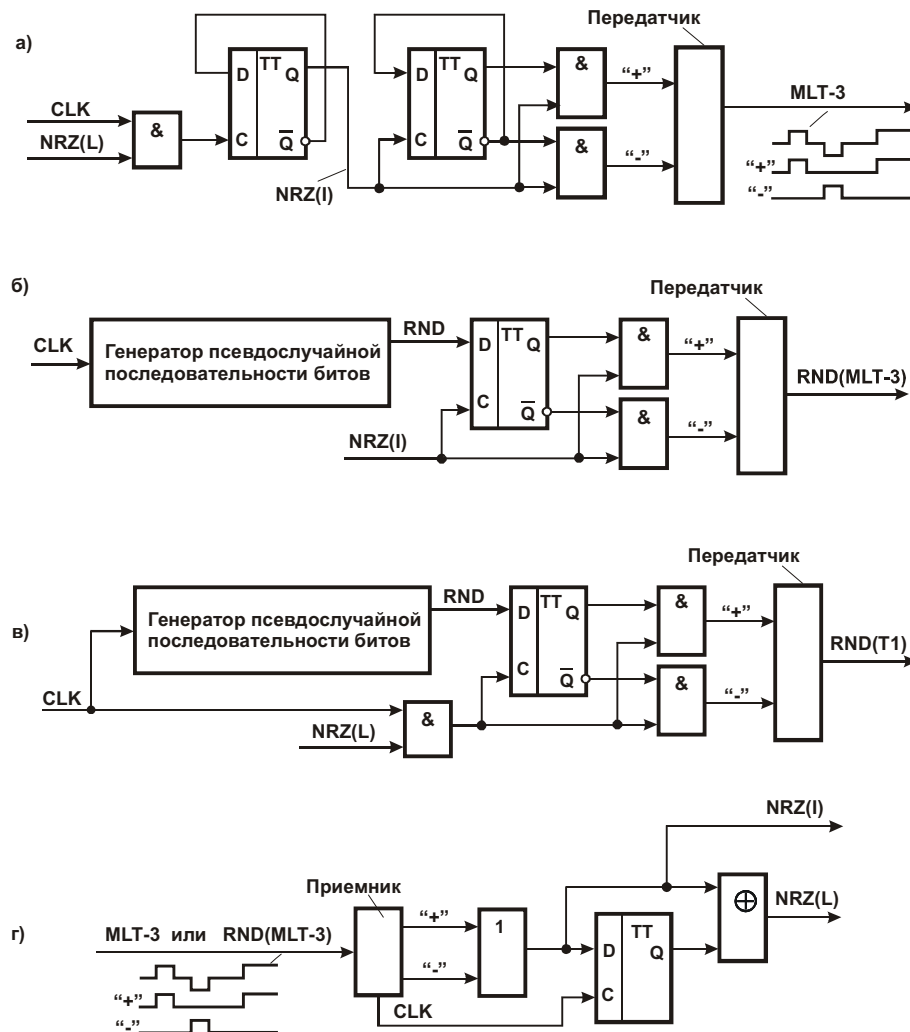


Рис. 8.8. Упрощенные схемные решения: *а* – формирователь кодов NRZ(I), MLT-3; *б* – формирователь кода RND(MLT-3) с псевдослучайным чередованием полярностей импульсов; *в* – формирователь кода RND(T1) с псевдослучайным чередованием полярностей импульсов; *г* – дешифратор кода MLT-3 или RND(MLT-3)

Таким образом, закон, по которому данные скремблировались передатчиком, остается неизвестным приемнику!

Предлагаемый метод применим и к другим трехуровневым кодам, таким как B3ZS, B6ZS, HDB3 (см. п. 8.1).

Рассмотренные схемные решения позволяют простыми средствами уменьшить уровень помех, излучаемых на соседние витые пары проводов кабеля.

### 8.3.2. Двубинарное кодирование

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

В схеме, показанной на рис. 8.9, 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. 8.7).

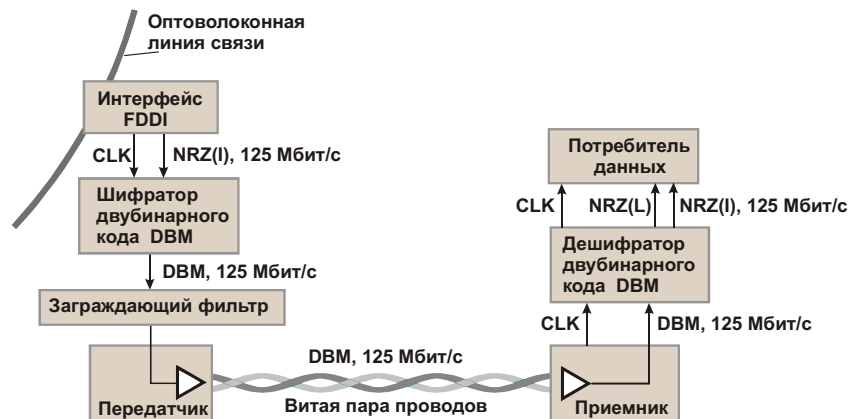


рис. 8.9. Схема высокоскоростной передачи данных в двоичном коде с использованием витой пары проводов

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 двоичное 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. 8.3.1 кодами MLT-3 и RND(MLT-3).

Как показано на рис. 8.9, 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 двоичного кода (рис. 8.10) [1] содержит инвертор, логический элемент Исключающее ИЛИ (XOR), тактируемый элемент T задержки, дешифратор DC со структурой 2×4, элемент ИЛИ, электронные ключи SW1 – SW3 и два источника U1 и U2 постоянного напряжения. Временные диаграммы формирования кода DBM показаны на рис. 8.11.

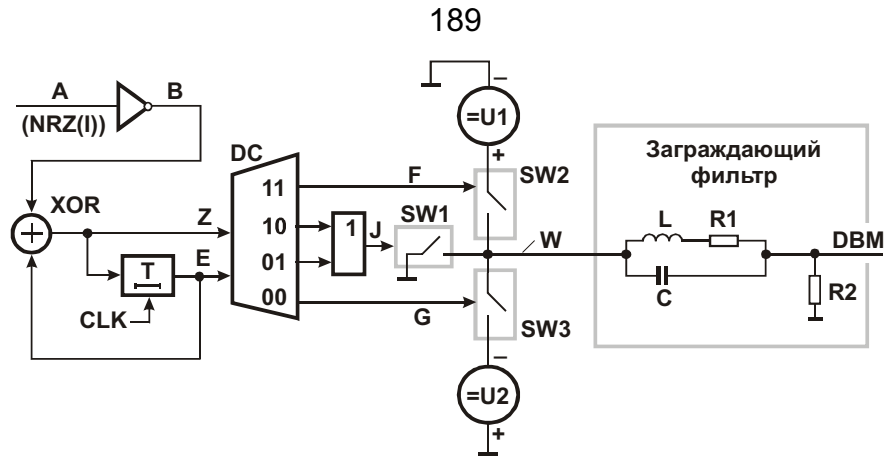


рис. 8.10. Схема шифратора двубинарного кода DBM и структура заграждающего фильтра

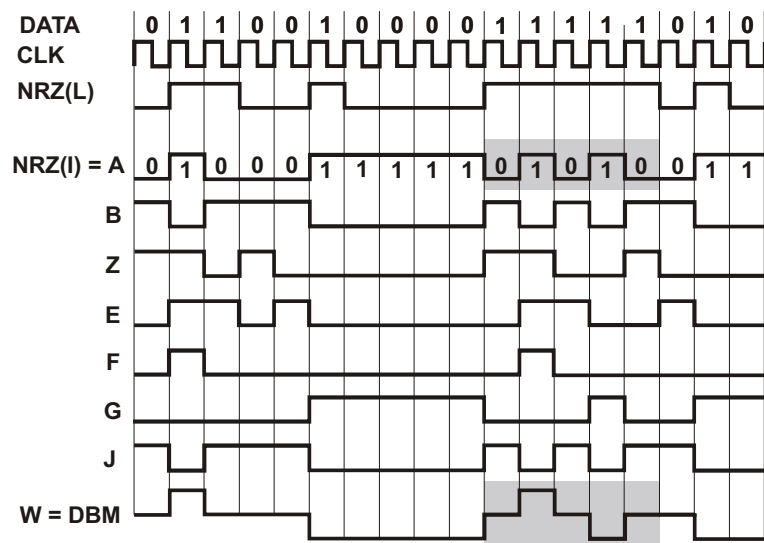


рис. 8.11. Временные диаграммы формирования двубинарного кода DBM

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  $SW3$ , поэтому на выход  $W$  шифратора поступает отрицательное напряжение от источника  $U2$ . При  $Z \neq E$  сигнал  $J = 1$  замыкает ключ  $SW1$ , на выход шифратора поступает нулевое напряжение. При  $Z = E = 1$  сигнал  $F = 1$  замыкает ключ  $SW2$ , на выход шифратора поступает положительное напряжение от источника  $U1$ .

Процесс шифрации удобно проследить с помощью диаграммы состояний, приведенной на рис. 8.12.

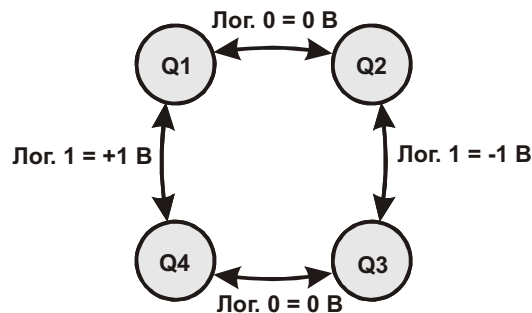


рис. 8.12. Диаграмма состояний шифратора двубинарного кода DBM

Шифратор может находиться в одном из четырех состояний  $Q1 - Q4$ . Если, например, шифратор пребывает в состоянии  $Q1$ , 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  $Q1$  и  $Q4$ . В этой ситуации шифратор переходит в состояние  $Q4$ .

Если шифратор находится в состоянии  $Q1$ , 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  $Q1$  и  $Q2$ . В данной ситуации шифратор переходит в состояние  $Q2$ . Переходы между состояниями  $Q2$  и  $Q3$  возможны при поступлении на вход  $A$  сигналов лог. 1, но эти переходы сопровождаются выдачей отрицательного напряжения ( $-1$  В) на выход  $W$ . Переходы между состояниями  $Q3$  и  $Q4$  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  $Q1$  в состояние  $Q2$  и обратно либо из состояния  $Q3$  в состояние  $Q4$  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  $Q1$  в состояние  $Q4$  и обратно либо из состояния  $Q2$  в состояние  $Q3$  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. 8.11. 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  $(Z_F + R2)/R2$  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 ре-

шениями. 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. 8.9) можно выполнить по схеме, приведенной на рис. 8.8, з. Эта схема нечувствительна к полярности импульсов и в равной мере применима для дешифрации кодов MLT-3, RND(MLT-3) и DBM.

## 8.4. 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. лишены какой-либо видимой закономерности.

### 8.4.1. Генераторы псевдослучайных битовых последовательностей

Скремблеры и дескремблеры обычно построены на основе генераторов псевдослучайных битовых последовательностей. Пример такого генератора приведен на рис. 8.13 [65]. Генератор выполнен на основе кольцевого сдвигового регистра RG с логическим элементом Исключающее ИЛИ (XOR) в цепи обратной связи. Если в исходном состоянии в регистре присутствует любой ненулевой код, то под действием синхросигнала CLK этот код будет непрерывно циркулировать в регистре и одновременно видоизменяться. В качестве выхода генератора можно также использовать выход любого разряда регистра.

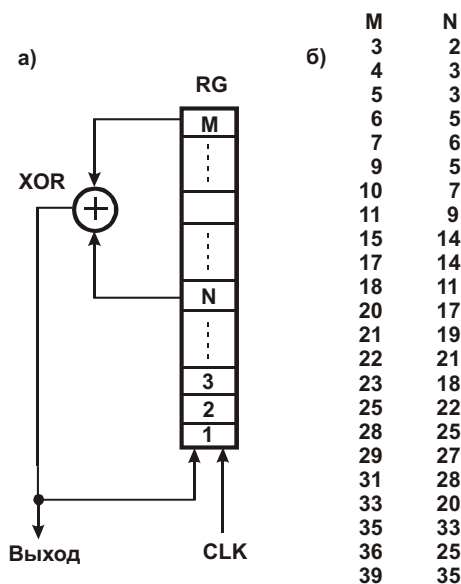


Рис. 8.13. Генератор псевдослучайной битовой последовательности максимальной длины: а – схема; б – таблица для выбора промежуточной точки подключения обратной связи

В общем случае в  $M$ -разрядном регистре обратная связь подключается к разрядам с номерами  $M$  и  $N$  ( $M > N$ ). Выбор оптимального значения  $N$  для заданного  $M$  – непростая задача. К счастью, она уже решена. Вариант таблицы выбора  $N$  приведен на рис. 8.13. Таблица описывает ряд генераторов различной разрядности. Каждый генератор формирует последовательность битов с максимальным периодом повторения, равным  $2^M - 1$ . В такой последовательности встречаются все  $M$ -разрядные коды, за исключением нулевого. Этот код представляет собой своеобразную “ловушку” для данной схемы: если бы нулевой код появился в регистре, дальнейшая последовательность битов была бы также нулевой. Но при нормальной работе генератора попадания в ловушку не происходит. (Усовершенствованные генераторы, не имеющие запрещенных состояний, рассмотрены в [72].)

Последовательность максимальной длины обладает следующими свойствами.

1. В полном цикле ( $2^M - 1$  тактов) число лог. 1 на единицу больше, чем число лог. 0. Добавочная лог. 1 появляется за счет исключения состояния, при котором в регистре присутствовал бы нулевой код. Это можно интерпретировать так, что вероятности появления на выходе регистра лог. 0 и лог. 1 практически одинаковы.

2. В полном цикле ( $2^M - 1$  тактов) половина серий из последовательных лог. 1 имеет длину 1, одна четвертая серий – длину 2, одна восьмая – длину 3 и т. д. Такими же свойствами обладают и серии из лог. 0 с учетом пропущенного лог. 0. Это говорит о том, что вероятности появления “орлов” и “решек” не зависят от исходов предыдущих “подбрасываний”. Поэтому вероятность того, что серия из последовательных лог. 1 или лог. 0 закончится при следующем подбрасывании, равна  $1/2$  вопреки обывательскому пониманию “закона о среднем”.

3. Если последовательность полного цикла ( $2^M - 1$  тактов) сравнивать с этой же последовательностью, но циклически сдвинутой на любое число тактов  $W$  ( $W$  не является нулем или числом, кратным  $2^M - 1$ ), то число несовпадений будет на единицу больше, чем число совпадений.

Наиболее распространены две основные схемы построения пар “скремблер – дескремблер”: с неизолрованными и изолированными генераторами псевдослучайных битовых последовательностей. Рассмотрим эти схемы и их модификации.

#### 8.4.2. Скремблер и дескремблер с неизолрованными генераторами псевдослучайных битовых последовательностей

В схеме, приведенной на рис. 8.14 [70], скремблер и дескремблер выполнены на основе рассмотренных генераторов псевдослучайных битовых последовательностей. Оба генератора имеют одинаковую разрядность и однотипную структуру обратных связей. Все процессы, протекающие в системе передачи данных, синхронизируются от тактового генератора (на рисунке не показан). Этот генератор размещен на передающей стороне системы и может принадлежать источнику данных либо скремблеру. В каждом такте на вход скремблера подается очередной бит передаваемых данных  $SD$ , а в сдвиговом регистре  $RG1$  накопленный код продвигается на один разряд вправо.

Если предположить, что источник данных посылает в скремблер длинную последовательность лог. 0, то элемент  $XOR1$  можно рассматривать как повторитель сигнала  $Y1$  с выхода элемента  $XOR2$ . В этой ситуации регистр  $RG1$  замкнут в кольцо и генерирует точно такую же псевдослучайную последовательность битов, как и в рассмотренной ранее схеме (см. рис. 8.13). Если от источника данных поступает произвольная битовая последовательность, то она взаимодействует с последовательностью битов с выхода элемента  $XOR2$ . В результате формируется новая (скремблированная) последовательность битов  $SCRD$ , по структуре близкая случайной. Эта последовательность, в свою очередь, продвигается по регистру  $RG1$ , формирует поток битов на выходе элемента  $XOR2$  и т. д.

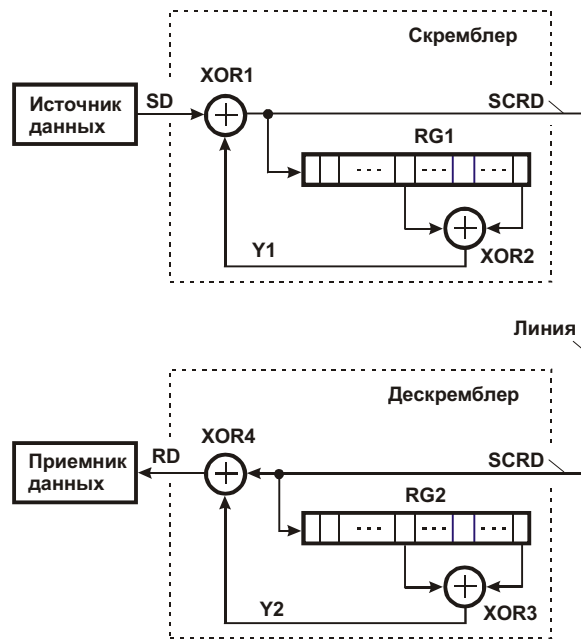


рис. 8.14. Система передачи данных, в которой скремблер и дескремблер содержат неизолированные генераторы псевдослучайных битовых последовательностей

Скремблированная последовательность битов  $SCRД$  передается по линии и поступает в дескремблер. С помощью генератора с фазовой автоподстройкой частоты (этот генератор на рисунке не показан) из входного сигнала выделяется тактовый сигнал. Под управлением тактового сигнала биты  $SCRД$  продвигаются в регистре  $RG2$ , а в приемник данных поступают дескремблированные данные  $RD$ .

Потоки данных  $RD$  и  $SD$  совпадают с точностью до задержки передачи по линии. Действительно, в установившемся режиме в сдвиговых регистрах  $RG1$  и  $RG2$  присутствуют одинаковые коды, так как на входы этих регистров поданы одни и те же данные  $SCRД$ , а тактовая частота, по сути, общая. Поэтому  $Y2 = Y1$ , и, с учетом этого,  $RD = SCRД \oplus Y2 = SD \oplus Y1 \oplus Y2 = SD \oplus Y1 \oplus Y1 = SD \oplus 0 = SD$ .

Рассмотренная система передачи данных не требует применения какой-либо специальной процедуры начальной синхронизации. После заполнения сдвигового регистра  $RG2$ , как было показано, генераторы псевдослучайных битовых последовательностей работают синхронно (их состояния всегда одинаковы). При появлении одиночной ошибки в линии синхронизация временно нарушается, но затем автоматически восстанавливается, как только правильные данные вновь заполнят регистр  $RG2$ . Однако в процессе продвижения ошибочного бита по сдвиговому регистру  $RG2$ , а именно, в периоды его попадания сначала на первый, а затем на второй вход элемента  $XOR3$  сигнал  $Y2$  дважды принимает неправильное значение. Это приводит к размножению одиночной ошибки – она впервые появляется в сигнале  $RD$  в момент поступления из линии и затем возникает еще два раза при последующем двукратном искажении сигнала  $Y$ . Еще один недостаток рассмотренной системы передачи данных связан с тем, что существуют некоторые неблагоприятные кодовые ситуации, с которыми скремблер “не справляется” (подробнее об этом – см. п. 8.4.4).

### 8.4.3. Скремблер и дескремблер с изолированными генераторами псевдослучайных битовых последовательностей

В схеме, приведенной на рис. 8.15, генераторы псевдослучайных битовых последовательностей включены так, что они изолированы от каких-либо нежелательных внеш-

них воздействий. Генераторы, как и в предыдущей схеме, работают синхронно, поэтому скремблирующий  $Z1$  и дескремблирующий  $Z2$  сигналы одинаковы. Ошибка в линии не размножается дескремблером, так как она не попадает в сдвиговый регистр  $RG2$ . Недостаток этой схемы – отсутствие самосинхронизации генератора псевдослучайной битовой последовательности дескремблера (напомним, что в предыдущей схеме такая синхронизация имеется). Процедура синхронизации такой системы описана в п. 8.4.5.

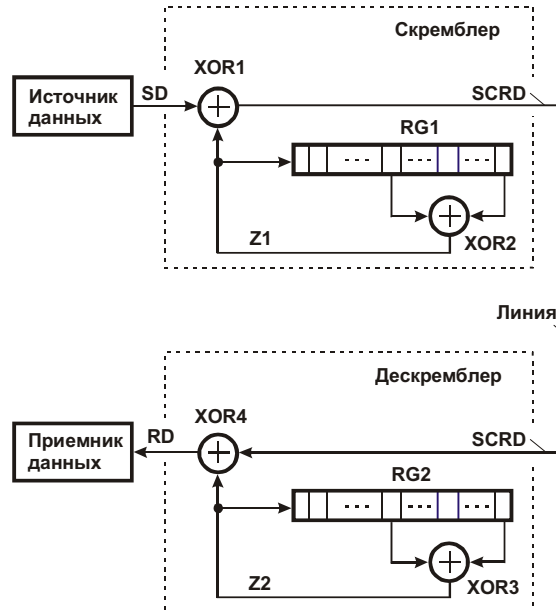


Рис. 8.15. Система передачи данных, в которой скремблер и дескремблер содержат изолированные генераторы псевдослучайных битовых последовательностей

#### 8.4.4. Скремблер и дескремблер с неизолированными генераторами – улучшенный вариант

Рассмотрим улучшенный вариант скремблера – дескремблера, построенного на основе двух одинаковых генераторов псевдослучайных последовательностей битов, рис. 8.16 [69]. Улучшение состоит в устранении упоминавшихся в п. 8.4.2 неблагоприятных кодовых ситуаций. В отличие от схемы, приведенной на рис. 8.14, применены средства коррекции состояний генераторов для устранения нежелательных последовательностей битов.

Скремблер содержит сдвиговый регистр  $RG1$  с логическими элементами Иключающее ИЛИ ( $XOR1$  и  $XOR2$ ) 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 груп-



пы 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).

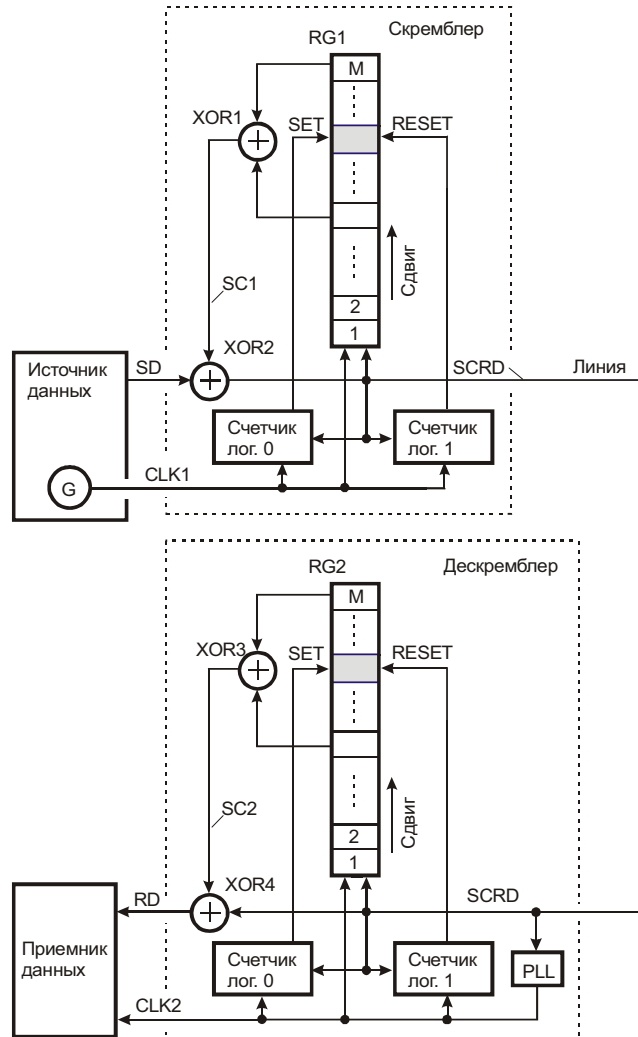


рис. 8.16. Система передачи данных, в которой скремблер и дескремблер содержат неизолированные генераторы псевдослучайных битовых последовательностей (улучшенный вариант)

Система передачи данных функционирует следующим образом. Источник данных формирует синхронный битовый поток SD и соответствующий синхросигнал CLK1. Этот поток проходит через логический элемент XOR2. 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

$$RD = SCRD \oplus SC2 = SCRD \oplus SC1 = SD \oplus SC1 \oplus SC1 = SD.$$

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 XOR2 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 нулевыми битами. Если после этого источник сигнала начнет передавать длинную последовательность лог. 0, то на обоих входах логического элемента XOR2 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 XOR2 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.

#### **8.4.5. Синхронизация изолированных генераторов скремблера и дескремблера**

В системе передачи данных, показанной на рис. 8.17, 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. 8.15).

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 де-

скремблера 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  $SD$ .

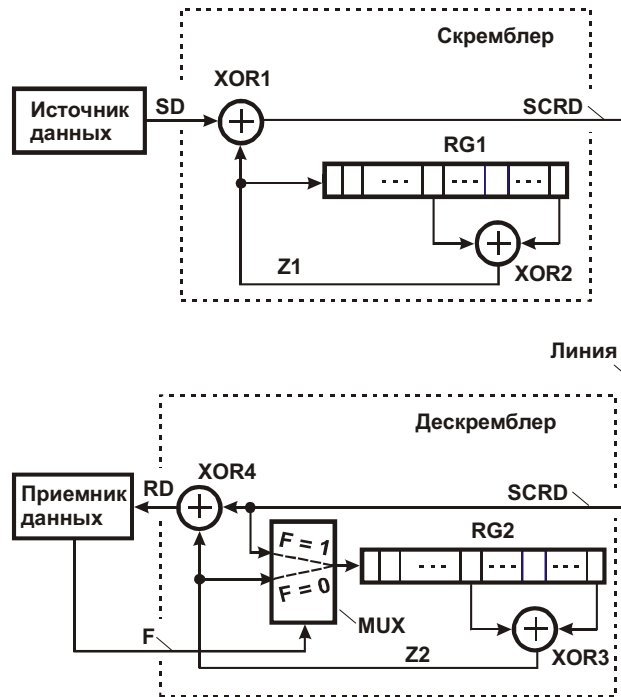


Рис. 8.17. Система передачи данных, в которой скремблер и дескремблер содержат изолированные генераторы псевдослучайных битовых последовательностей. Введены средства синхронизации этих генераторов

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  $SCRД$ , как в схеме на рис. 8.14.

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  $XOR1$  выполняет функцию повторителя сигнала  $Z1$ . Поэтому в данном случае скремблированный сигнал  $SCRД$  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.

### 8.4.6. Применение системы скремблер – дескремблер для тестирования канала связи

Как показано на рис. 8.18, для тестирования цепей RxC и RxD использованы генератор и анализатор псевдослучайных последовательностей битов. По существу, генератор и анализатор представляют собой рассмотренные ранее скремблер и дескремблер, причем скремблируется сигнал “Постоянный нуль”, т. е. последовательность нулевых битов ( $SD = 0$ , см. рис. 8.14). В отсутствие ошибок передачи сигналов RxC и RxD сигнал на выходе триггера ТТ также должен быть нулевым.

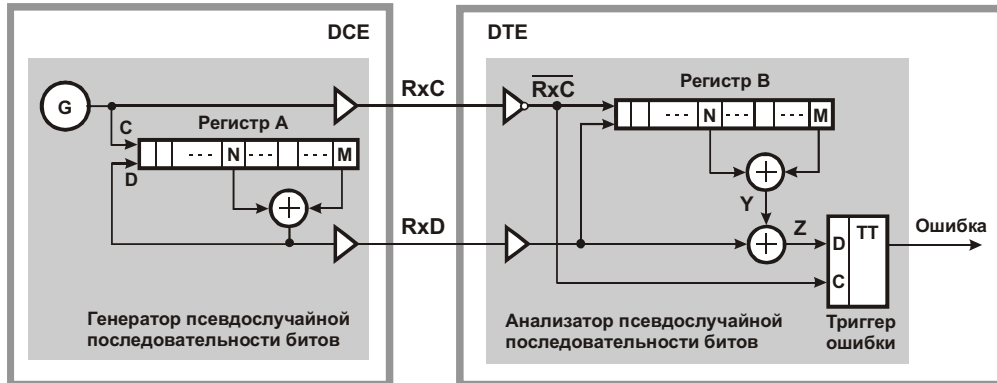


рис. 8.18. Система контроля передачи данных и синхросигнала между устройствами типа DCE и DTE

Под управлением сигнала RxC с генератора G в линию RxD из регистра A поступает псевдослучайная последовательность битов (см. рис. 8.19). Положительные фронты сигнала RxC задают границы битовых интервалов сигнала RxD.

Положительные фронты инвертированного сигнала RxC задают смещенную на половину такта синхросетку приема данных в регистр B и в D-триггер ТТ. Такты работы генератора псевдослучайной последовательности битов обозначены символами T1 с соответствующими индексами в скобках (например T1(L + 4)). Аналогично такты работы анализатора обозначены символами T2.

В тактах T1(J) ... T1(J + 3) состояние регистра A изменяется в такой последовательности: S(J), S(J + 1), S(J + 2), S(J + 3). Соответствующие биты выходных данных: D(J), D(J + 1), D(J + 2), D(J + 3). Как отмечалось при описании системы “скремблер – дескремблер”, в установившемся режиме при отсутствии ошибок содержимое передающего и приемного регистров (в данном случае, регистров A и B) одинаково. Поэтому, как показано на временных диаграммах, коды в регистре B совпадают с кодами в регистре A с учетом взаимного смещения синхросеток на половину такта.

Интересно отметить, что код в приемном регистре (B) формируется на половину такта раньше, чем тот же код в передающем регистре (A)! Такое поведение анализатора можно рассматривать как предсказание очередного правильного бита (0 или 1) в ожидании его поступления по линии RxD. Как следует из временных диаграмм, в отсутствие ошибок предсказания полностью оправдываются. Это проявляется в том, что сигнал Z на входе данных D-триггера принимает устойчивое нулевое значение в моменты записи, поэтому триггер остается в состоянии лог. 0.

Предположим, что в такте T1(J + 4) в результате воздействия на линию RxD импульса помехи передаваемый бит исказился: вместо истинного лог. 0 передается ложная лог. 1 или наоборот. В этой ситуации во второй половине такта T2(J + 4) обнаруживается несоответствие предсказанного и фактически принятого битов (сигнал Z принимает стабильное значение, равное лог. 1). Поэтому в следующем такте T2(J + 5) триггер переходит в состояние лог. 1. Таким образом, первое проявление ошибки зафиксировано с задержкой в половину такта после ее возникновения в линии.

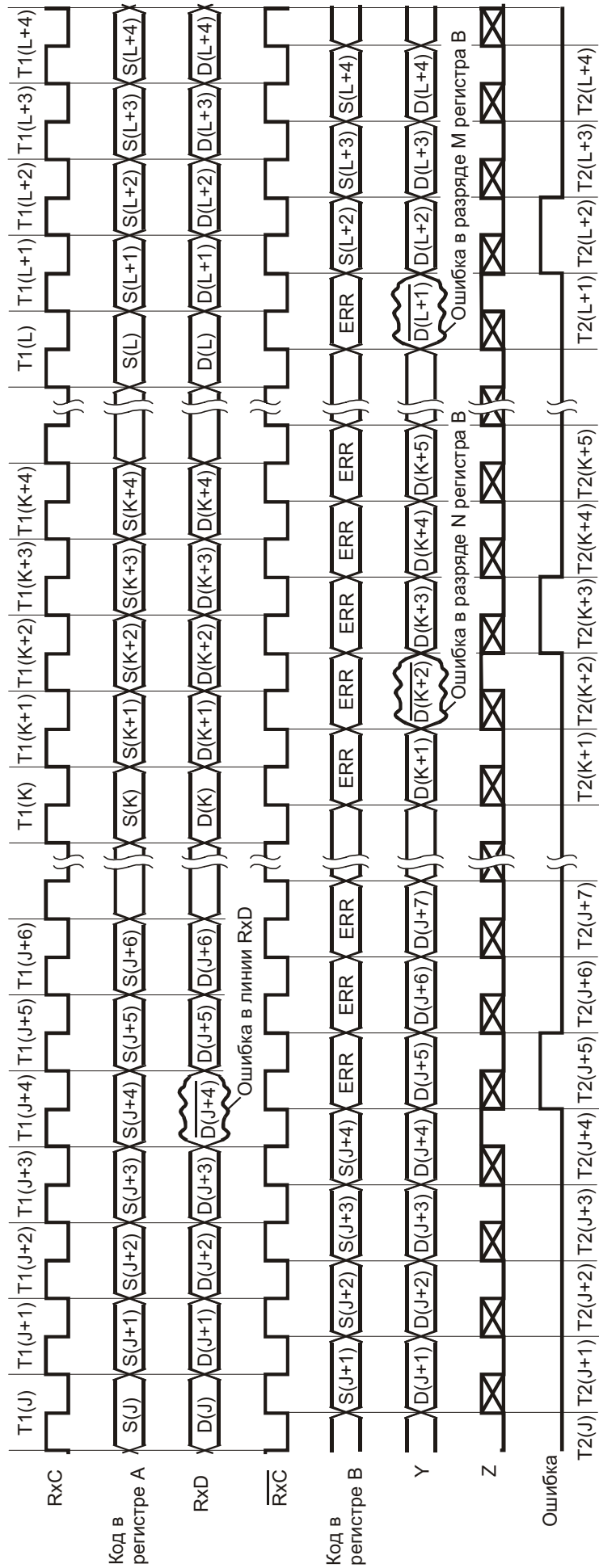


рис. 8.19. Временные диаграммы передачи и приема тестовых битовых последовательностей

Начиная с такта  $T1(J + 5)$  по линии RxD вновь передаются правильные биты. Сравнение предсказанных и фактически принятых битов вновь дает положительные результаты, но ранее принятый в регистр В ошибочный бит начинает продвижение к разряду N. Код в регистре В искажен (что отражено на диаграмме символами “ERR”), но искажения пока внешне не проявляются.

В такте  $T2(K + 2)$  ошибочный бит попадает в разряд N. Вследствие этого происходит неправильное предсказание ожидаемого бита, т. е. во второй половине такта предсказанный бит Y противоположен правильному биту  $D(K + 2)$ , полученному по линии RxD. Поэтому триггер повторно регистрирует ошибку. После этого ошибочный бит продолжает продвижение по регистру В в направлении разряда M. В такте  $T2(L + 1)$  ошибочный бит достигает разряда M. Неправильное предсказание повторяется, триггер в третий раз регистрирует ошибку. После этого ошибочный бит выталкивается из сдвигового регистра В и, следовательно, более не влияет на работу системы контроля. Таким образом, одиночная ошибка в линии приводит к формированию пачки из трех импульсов на выходе триггера.

Рассмотренную систему, состоящую из генератора и анализатора псевдослучайной последовательности битов, обычно называют BER-тестером (Bit-Error-Rate – интенсивность поступления ошибочных битов от объекта проверки; определение дано в международном стандарте ITU-T O.153). Для более краткого обозначения генератора и анализатора псевдослучайной последовательности битов используют термины “BER-генератор” и “BER-анализатор”. Рассмотрим некоторые возможные варианты применения BER-тестеров для проверки работоспособности фрагментов телекоммуникационных систем. В приведенных далее примерах такие системы (точнее, их модели) построены на основе модемов Зелакс М-144.

В системе, показанной на рис. 8.20, использованы четыре модема. Проверяется работоспособность фрагмента, включающего первый – третий модемы, интерфейс типа V.35 между первым и вторым модемами и линию связи, выполненную в виде витой пары проводов. Первый и четвертый модемы имитируют оконечные устройства типа DTE, второй и третий используются по прямому назначению – для передачи данных через протяженную линию связи. Обмен данными по линии связи дуплексный, т. е. предусматривает передачу данных по линии одновременно в обоих направлениях.

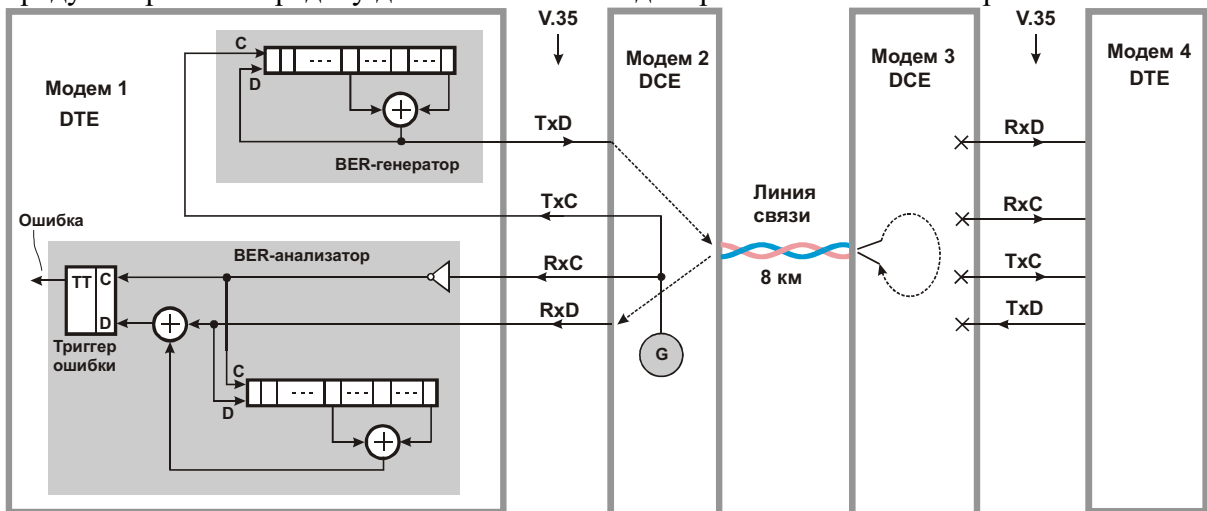


рис. 8.20. Включение BER-тестера в телекоммуникационную систему

В первом модеме включен режим BER-тестера. Это означает, что по положительным фронтам синхросигнала TxС в линию передаваемых данных с выхода BER-

генератора выдаются псевдослучайные биты данных TxD. Параллельно с этим BER-анализатор этого же модема проверяет правильность последовательности битов RxD, сопровождаемых синхросигналом TxC. Все процессы, протекающие в системе, синхронизируются от генератора G, размещенного во втором модеме.

В третьем модеме включен режим возврата данных, поступающих из линии связи. С помощью генератора с фазовой автоподстройкой частоты (этот генератор на рисунке не показан) из принятого линейного сигнала выделяются синхросигнал и данные. Затем данные кодируются и выдаются в линию в направлении второго модема. Во втором модеме из принятого линейного сигнала также выделяются синхросигнал и данные. После этого осуществляется привязка принятых данных к синхросигналу RxC и их выдача в первый модем. В данном примере третий модем логически разрывает все связи с четвертым модемом, так что последний оказывается изолированным.

В отсутствие ошибок данные проходят от первого модема к третьему и обратно без искажений. Сигнал на выходе триггера ошибки постоянно равен нулю. Любое нарушение правильной последовательности данных или (и) существенное искажение синхросигнала на входах BER-анализатора регистрируется триггером ошибки в виде одного или нескольких импульсов Ошибка. Отметим, что, в отличие от рассмотренного ранее упрощенного примера (см. рис. 8.18), поток битов на входе BER-анализатора отстает от потока битов на выходе BER-генератора на несколько тактов из-за задержек, вносимых линией связи и буферной памятью второго и третьего модемов. Это, однако, не вызывает каких-либо затруднений при оценке принятых данных, так как фактически анализируется *закономерность построения* поступающей на вход последовательности, а не простое совпадение переданных и принятых битов.

В схеме, показанной на рис. 8.21, назначение модемов такое же, как и в предыдущем примере, четвертый модем логически изолирован от третьего на время проведения тестирования. Отметим, что первый и второй модемы выполняют функции устройств типа DCE и DTE. В первом и третьем модемах включены BER-тестеры.

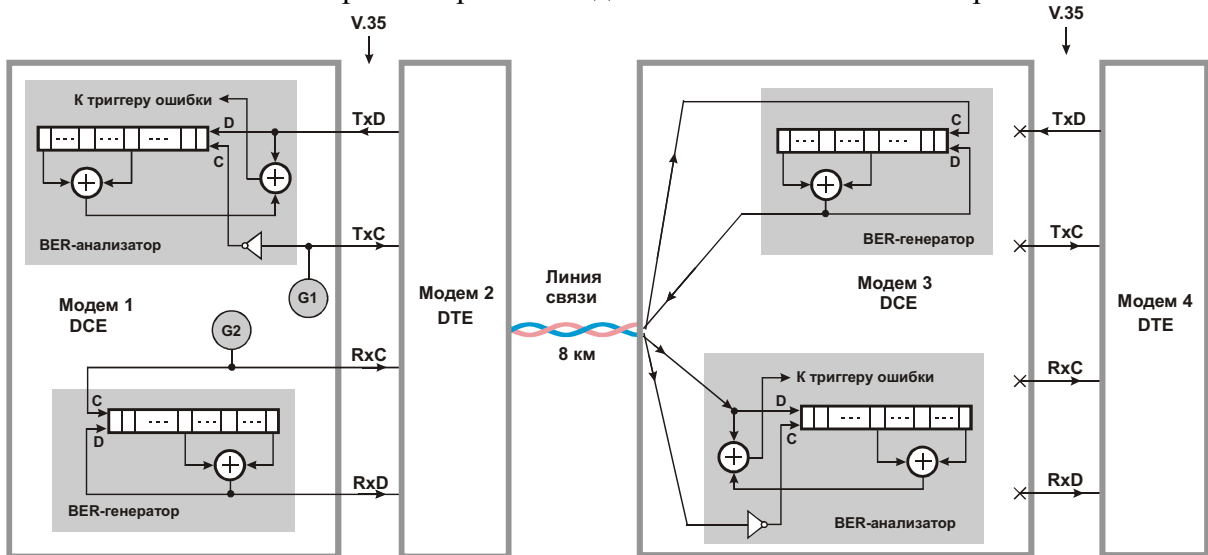


рис. 8.21. Включение двух BER-тестеров в телекоммуникационную систему

Генератор G1 задает скорость передачи данных по “верхнему” каналу: от BER-генератора третьего модема к BER-анализатору первого модема. Независимо от этого генератор G2 задает скорость передачи данных по “нижнему” каналу: от BER-генератора первого модема к BER-анализатору третьего модема. Эти же независимые каналы позволяют при выключенных BER-тестерах одновременно передавать данные в разных направлениях между первым и четвертым модемами с одинаковыми или разными скоростями.