Вероятностная синхронизация в телекоммуникационных системах: ускорение передачи информационных кадров

Предлагается улучшить способ передачи данных, который уже давно применяется в широко известных интерфейсах семейства RS (RS-232, V.35, RS-422, RS-449 и других). В синхронной версии любого из этих интерфейсов предусмотрена пара сигналов, передаваемых по двум линиям в одном и том же направлении: TxD и CLK. Первый сигнал предназначен для последовательной передачи данных, второй – для синхронизации приёмника. Фронты сигнала CLK разграничивают во времени битовые интервалы, а его срезы попадают в центры этих интервалов и используются приёмником для считывания битов данных. Столь простой и выдержавший проверку временем способ передачи данных в последние годы получил развитие в нескольких направлениях. В данной статье кратко прослежена история одного из усовершенствований и предложено новое решение, основанное на идее вероятностной синхронизации [1 – 4].

Введение

Прежде, чем перейти к описанию предлагаемого способа передачи данных между двумя устройствами, рассмотрим некоторые известные решения. В их основе лежит интерфейс RS-232, предложенный ещё в начале 60-х годов и широко применяемый до настоящего времени.

В интерфейсах семейства RS для синхронной передачи данных из устройства 1 в устройство 2 (рис. 1, *а, б*), например, из компьютера в модем, используются две лини: TxD и CLK. Для передачи данных в обратном на правлении применены две аналогичные по функциональному назначению линии RxD и RxC (последние на рисунке не показаны) и те же временные соотношения между сигналами, что и на рис. 1, *б*.

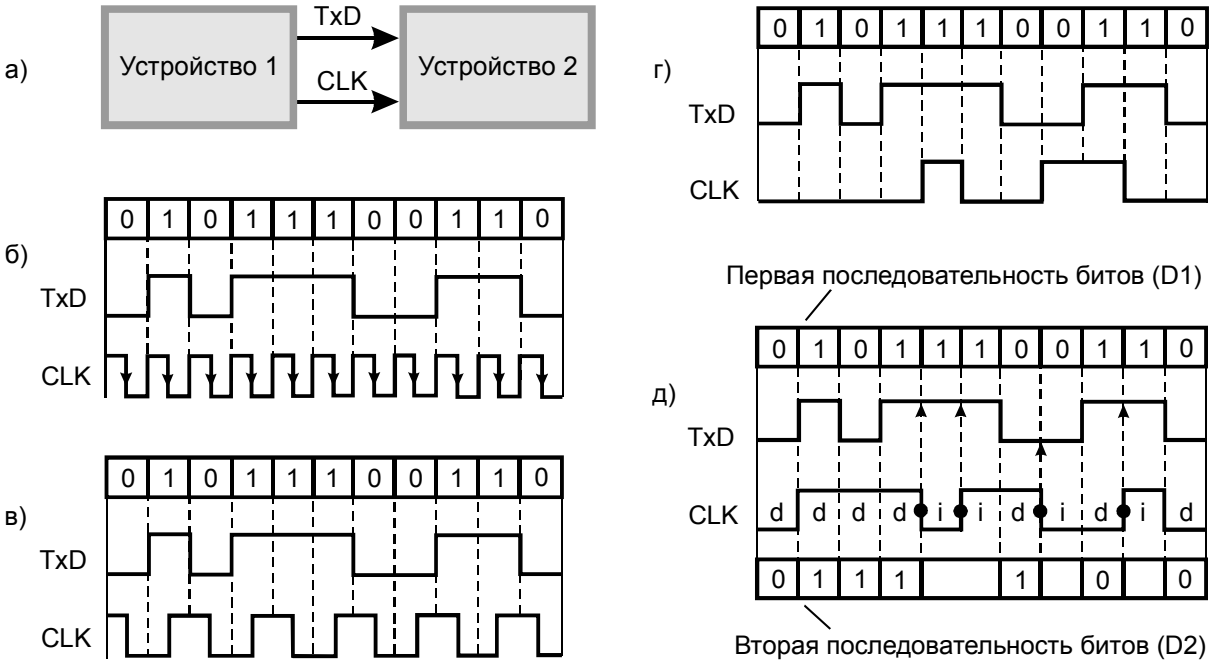

Рис. 1. Попутная передача данных и синхросигнала: *а* – схема; *б* – традиционный вариант временной диаграммы; *в – д* – усовершенствованные варианты

Каждому биту из передаваемой последовательности 01011100110 выделен определённый интервал времени – битовый интервал. Границы между интервалами показаны на рис. 1 штриховыми линиями. Сигналам лог. 1 и лог. 0 соответствуют напряжения высокого и низкого уровня. По фронту синхросигнала CLK устройство 1 начинает выдачу очередного бита данных. По срезу этого сигнала устройство 2 считывает бит с линии TxD.

Недостатками такого способа передачи данных являются низкая скорость и высокий уровень помех, излучаемых на соседние провода многожильного кабеля, если он использован в составе канала связи. Первый недостаток связан с тем, что передача каждого бита данных сопровождается двумя (а не одним) изменениями состояния синхросигнала CLK. Второй недостаток: в спектре синхросигнала доминирует ярко выраженный энергетический максимум на основной частоте. Это приводит к тому, что из-за паразитных (емкостных и индуктивных) связей с соседними проводами многожильного кабеля в них формируется достаточно выраженный сигнал помехи [1].

В [5] для обозначения центров битовых интервалов предложено использовать как фронты, так и срезы синхросигнала CLK (рис. 1, *в*). Фронты и срезы синхросигнала CLK должны соответствовать установившимся значениям сигнала TxD. Точнее, должны гарантироваться достаточные интервалы времени предварительной установки и удержания сигнала TxD соответственно до и после изменения сигнала CLK.

Сравнивая временные диаграммы, приведенные на рис. 1, *в*, и 1, *б*, можно заключить, что при одинаковой скорости передачи данных частота сигнала CLK уменьшена вдвое. Это позволяет увеличить длину линии связи. Можно привести и иную трактовку тех же нововведений: при неизменной частоте сигнала CLK скорость передачи данных удвоена. Интересно отметить, что столь простое и эффективное усовершенствование интерфейса RS-232 предложено лишь через 40 лет после его создания!

Однако только что рассмотренный способ передачи данных (рис. 1, *в*), к сожалению, также обладает отмеченным ранее недостатком – повышенным уровнем излучаемых помех при передаче периодического сигнала CLK.

Этот недостаток устранён при передаче данных в соответствии с диаграммой на рис. 1, *г*. [6]. В данном случае сигнал CLK, как и ранее, выполняет функцию разграничения битовых интервалов, но он изменяется только в тех ситуациях, при которых сигнал данных TxD остается неизменным. Иными словами, на любой границе любого битового интервала изменяется либо сигнал TxD, либо сигнал CLK. Приемник отслеживает изменения этих сигналов и поэтому в полной мере располагает информацией о местоположении границ между битовыми интервалами.

При передаче случайных или псевдослучайных данных TxD вероятности формирования сигналов лог. 0 и 1 в очередном битовом интервале одинаковы и не зависят от предыстории [1]. Производный от сигнала TxD сигнал CLK также можно рассматривать как случайный (псевдослучайный). При этом он не содержит ярко выраженных спектральных составляющих, что способствует снижению уровня помех, излучаемых в соседние провода кабеля. Для преобразования произвольных данных к виду, близкому случайному, используют их аппаратное или программное скремблирование при передаче и дескремблирование при приеме [1].

Рассмотренный способ передачи данных (рис. 1, *г*) усовершенствован в [7] (рис. 1, *д*). Сигнал CLK по-прежнему в необходимых случаях разграничивает битовые интервалы, но одновременно с этим используется для передачи дополнительного потока данных. По линиям TxD и CLK передаются две независимые последовательности данных D1 и D2. Первая последовательность D1 (основная) скремблирована и передается с гарантированной скоростью, определяемой частотой следования битовых интервалов. Вторая последовательность D2 (дополнительная) может не скремблироваться и передается со средней скоростью, равной половине скорости передачи первой последовательности.

Сигнал CLK помимо “обязательных” для разделения битовых интервалов фронтов сигнала (помечены на рисунке точками), содержит семь позиций для передачи дополнительной последовательности битов D2.

Поясним сказанное. В период, соответствующий первым четырем битовым интервалам, сигнал TxD изменяется, поэтому сигнал CLK можно использовать как

“транспортное средство” для передачи первых четырех битов (dddd) = 0111 последовательности D2. Эти биты могут иметь произвольные значения.

После выдачи в линию CLK битов (dddd) = 0111, принадлежащих последовательности D2, на границе между четвертым и пятым битовыми интервалами передатчик создает гарантированный перепад уровней (этот перепад помечен первой точкой), так как сигнал TxD остается неизменным. Это достигается размещением в пятом битовом интервале сигнала CLK инвертированного четвертого бита i (бита синхронизации) из последовательности D2. Далее бит повторно инвертируется и размещается в шестом интервале, так как сигнал TxD по-прежнему неизменен, затем в седьмом интервале размещается очередной бит d, принадлежащий последовательности D2 и т. д.

Устройство 2 принимает сигналы из линий TxC и CLK, запоминает их и “прорисовывает” временные диаграммы, показанные на рис. 1, *д*. Зная правила кодирования, по этим диаграммам устройство определяет, какие перепады сигнала CLK являются обязательными и сопровождают передачу битов i синхронизации, а какие сопровождают передачу “полезных” битов d. Поэтому восстановление данных, принадлежащих последовательностям D1 и D2, не сопряжено с какими-либо трудностями.

При равновероятном появлении сигналов лог. 0 и лог. 1 в битовых интервалах данных D1 (что достигается применением скремблирования на передающей стороне) средняя скорость передачи потока D2 составляет 50% скорости передачи потока D1. Действительно, в этом случае вероятность перепада уровней сигнала TxD на любой границе между битовыми интервалами равна 0,5 и не зависит от предыстории. Но именно эта вероятность определяет возможность размещения очередного бита d в структуре сигнала CLK.

Таким образом, в среднем каждый второй битовый интервал пригоден для передачи бита d данных второго канала D2. В результате суммарная скорость передачи данных между устройствами 1 и 2 (рис. 1, *а*) увеличена в 1,5 раза по сравнению со способами, соответствующими рис. 1, *в* и *г*. При этом сигналы TxD и CLK имеют “размытые” энергетические спектры, что способствует снижению уровня помех, излучаемых на соседние провода кабеля.

Для автоматического установления и поддержания кодовой синхронизации между скремблером устройства 1 и дескремблером устройства 2 можно применить схему с неизолированными генераторами псевдослучайных последовательностей битов [1] или её программную модель.

Идея применения вероятностной синхронизации

Следующий шаг на пути усовершенствования только что рассмотренной системы передачи данных основан на использовании вероятностной синхронизации [1 – 4].

Предположим, что из передатчика в приёмник по линиям TxD и CLK (рис. 1, *а*) пересылаются два потока данных D1 и D2 (рис. 1, *д*). Так как поток данных D1 скремблирован, то в нём с определённой вероятностью можно обнаружить любую заранее заданную кодовую комбинацию произвольной разрядности.

Поясним сказанное. Предположим, что некий внешний наблюдатель следит за состоянием линии TxD и в произвольно выбранных битовых интервалах считывает биты D1. Результат такого слежения должен подтвердить, что вероятности обнаружения сигналов лог. 0 или лог. 1 в любом битовом интервале одинаковы и равны 1/2, так как после скремблирования любые данные можно рассматривать как случайные.

Пусть, далее, в потоке последовательных данных D1 ищется некоторый двухразрядный код, например, 00. Вероятность его обнаружения при произвольной выборке из потока двух соседних битовых интервалов равна 1/4, так как все четыре возможные комбинации битов (00, 01, 10 и 11) равновероятны. Заданный трёхразрядный код будет обнаруживаться при произвольной однократной выборке с вероятностью 1/8 и т. д. В

общем случае при однократной выборке вероятность обнаружения в потоке данных D1 любого заданного N-разрядного кода равна 2^{-N}. Эта вероятность быстро уменьшается с увеличением N, но в любом случае остаётся ненулевой. Так, при однократной выборке 20-разрядного кода из потока данных D1 вероятность его совпадения с заранее заданным кодом равна $2^{-20} \approx 10^{-6}$. Иными словами, при непрерывном слежении за потоком данных D1 совпадения будут наблюдаться со средним периодом повторения, близким 10^6 бит. При скорости передачи данных 100 Мбит/с средняя частота событий обнаружения заданного кода составит примерно 100 Гц.

Предположим теперь, что устройства 1 и 2 содержат одинаковые анализаторы кодов, настроенные на поиск одной или нескольких заранее заданных последовательностей битов. Эти последовательности обнаруживаются обоими анализаторами одновременно с точностью до задержки передачи сигнала по линии связи, что несущественно – важно лишь, чтобы независимо от задержки сигнала структура потока данных на выходах устройства 1 совпадала с его структурой на входах устройства 2. Поэтому линия связи может содержать ретрансляторы, которые вносят неопределённую, но постоянную задержку сигнала.

Применяя далее понятие “одновременности” обнаружения заданных кодов удалёнными друг от друга устройствами 1 и 2, будем принимать во внимание только что сделанное замечание.

Моменты обнаружения заданных кодов могут служить опорными метками для одновременного выполнения устройствами 1 и 2 некоторых заранее оговорённых действий, характер которых зависит от содержания решаемой задачи. В этом, собственно, и заключается идея вероятностной синхронизации – безо всяких затрат на передачу специальной синхронизирующей информации удалённые друг от друга устройства одновременно “ниоткуда” получают метки времени, которые возникают “сами собой” в силу случайного характера передаваемых данных после их скремблирования.

При построении анализатора кодов может использоваться метод “скользящего окна”, сквозь которое просматривается бегущая строка битов потока TxD (рис. 2). Такое окно можно реализовать на сдвиговом регистре, через который проходит анализируемый поток битов. Код с параллельных выходов этого регистра сравнивается с одним или несколькими эталонами. (Известны и иные, более эффективные способы распознавания заданных кодов [7].) Как показано на рисунке, в некоторый момент в окне появляется искомый 20-разрядный код Z = 010…101. Этот код примечателен тем, что соответствующий ему сигнал TxD содержит перепады уровня на всех границах между битовыми интервалами. Таким же свойством обладает и обратный код: 101…010.

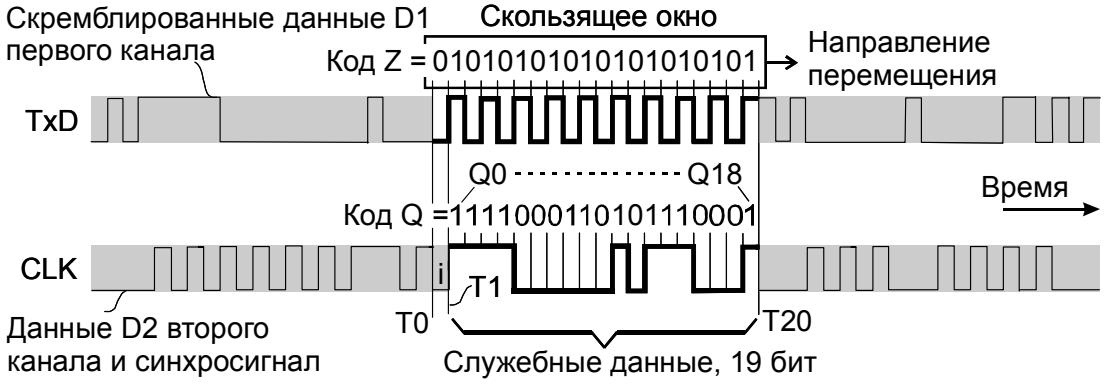

Рис. 2. Диаграммы, поясняющие общую идею вероятностной синхронизации

Так как при обнаружении кода Z границы между битовыми интервалами в пределах окна однозначно заданы фронтами и срезами сигнала TxD, то сигнал CLK на протяжении 19 битовых интервалов (промежуток времени T1 – T20) получает “полную свободу” и может переносить произвольные данные. Интервал T0 – T1 в данной кодовой ситуации используется для передачи бита i синхронизации. Здесь этот интервал не-

пригоден для передачи бита d канала D2, так как “обязательный” перепад уровня сигнала CLK в момент T0 может с равной вероятностью привести к вынужденному переходу сигнала в состояние лог. 0 или лог. 1 в зависимости от предыстории. В иных ситуациях, когда в момент T0 имеется перепад сигнала в линии TxD, интервал T0 – T1 может использоваться для передачи бита d.

После обнаружения в окне заданного кода окно скачком перемещается на 20 позиций вправо и затем вновь “скользит” с шагом, равным одному биту.

В протоколе обмена данными между устройствами 1 и 2 (рис. 1, *а*) можно предусмотреть соглашение, согласно которому в ситуации, показанной на рис. 2, сигнал CLK отображает 19-разрядное слово служебных данных, сформированных устройством 1. Интерпретация этих данных также определяется протоколом обмена и может быть, например, такой, как показано ниже.

- Нулевой бит Q0 19-разрядного слова служебных данных определяет содержание группы последующих битов. Если Q0 = 0, то последующие биты (Q1 – Q18) представляют собой данные d второго канала D2. В противном случае, который соответствует приведенной на рисунке диаграмме, Q0 = 1, и последующие 18 битов рассматриваются как данные управляющего канала D3.
- При Q0 = 1 биты Q1 – Q6 определяют тип данных канала D3, например: адрес источника служебных данных, адрес приёмника служебных данных, ответ, контрольная сумма, информация об ошибках, команда приостановки или возобновления встречного потока данных и т. п.
- При Q0 = 1 биты Q7 – Q18 интерпретируются в зависимости от типа данных, заданного разрядами Q1 – Q6.

Отметим некоторые примечательные свойства рассмотренного способа передачи данных.

1. Возможна одновременная передача данных по каналам D1, D2 и D3.
2. Скорость передачи данных по каналу D1 гарантирована и определяется длительностью битового интервала в канале связи. Служебная информация, сопровождающая работу каналов D1 и D2, передаётся по линии CLK (канал D3) и поэтому не снижает пропускную способность канала D1.
3. Скорость передачи данных по каналу D2 равна половине скорости передачи данных по каналу D1. Вставками служебных данных Q при такой оценке пренебрегаем, так как их доля в общем потоке битов в линии CLK в приведенном примере не превышает 19×10^{-6} – одна вставка 19-разрядного слова приходится в среднем на $2^{20} \approx 10^6$ бит.
4. Скорость передачи данных по служебному каналу может регулироваться выбором разрядности N искомого кода Z. Как было показано, средняя частота следования моментов обнаружения кода Z в данном примере округлённо равна 100 Гц. Это соответствует передаче служебной информации по каналу D3 с максимальной скоростью, равной 18 × 100 = 1,8 кбит/с. Если в течение некоторого промежутка времени нет необходимости передавать служебную информацию, то на её месте размещаются данные d канала D2 (Q0 = 0). С увеличением разрядности N кода Z средняя частота следования моментов его обнаружения снижается; каждый дополнительный бит уменьшает эту частоту в два раза.
5. Чтобы распознать служебные данные, не нужно вводить в потоки D1 или D2 какие-либо специальные признаки (флаги). Последовательность одновременных событий распознавания заданных кодов Z (моментов передачи служебных данных) на противоположных сторонах линии связи формируется “самопроизвольно” в процессе передачи полезных данных и подчиняется законам теории вероятностей. В этом, собственно, и состоит идея вероятностной синхронизации – можно передавать и получать некие метки времени, не прикладывая для этого никаких “усилий”!

Далее предложено использовать эту идею для указания положения границ между информационными кадрами, передаваемыми по линиям TxD и CLK.

Ускорение передачи информационных кадров

Данные обычно передают по каналам связи в виде непрерывной последовательности информационных кадров [7] (рис. 3). Для их разграничения применяют флаги, каждый из которых, как правило, размещается в начале кадра. Флаг может быть представлен некоторой уникальной последовательностью битов, которая не встречается в теле кадра. Чтобы гарантировать такую уникальность, применяют битстаффинг – добавление служебных битов в поток “полезных” данных.

Использование флагов и применение битстаффинга снижает эффективность работы канала связи, так как заметная часть его пропускной способности расходуется на передачу служебной информации в ущерб полезной. В предлагаемой системе передачи данных канал связи используется более эффективно, так как для разграничения кадров вместо флагов используются служебные данные, передаваемые достаточно редко в случайные моменты времени в соответствии с рассмотренной ранее общей идеей (рис. 2). При этом пропускная способность канала D1 не снижается – он используется только для передачи “полезных” битов данных. Пропускная способность канала D2, как было показано, снижается в пренебрежимо малой степени.

Рис. 3. Передача данных в виде последовательности информационных кадров (традиционный способ)

Предлагаемый способ передачи данных поясняется рис. 4.

В соответствии с ранее рассмотренной временной диаграммой (рис. 1, *д*), линия TxD используются для передачи данных канала D1, а линия CLK – для передачи битов d канала D2, битов i синхронизации и, в дополнение к этому, битов канала D3.

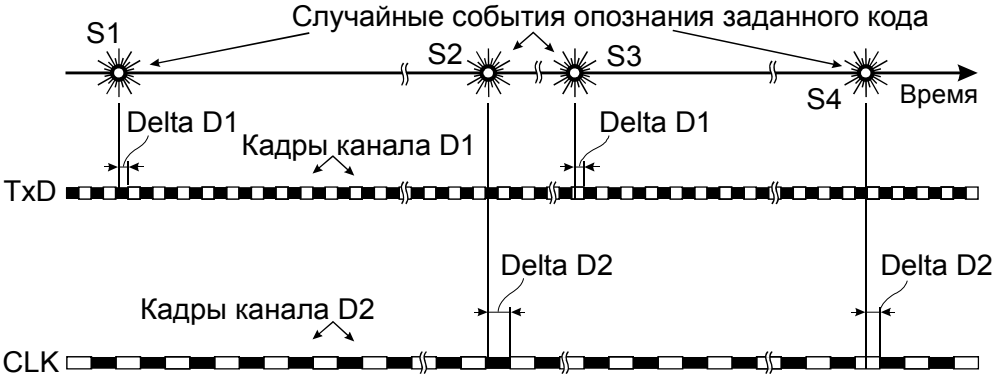

Рис. 4. Передача данных в виде последовательности информационных кадров (предлагаемый способ)

Данные канала D1 (D2) представлены непрерывной последовательностью информационных кадров, показанных на рисунке в виде чёрных и белых прямоугольников. Каждый кадр в данном примере содержит 4×10^3 “полезных” битов. Как уже отмечалось, “полезные” биты d, передаваемые по линии CLK, по мере необходимости перемежаются с битами i синхронизации, а также с кодами Q. Поэтому кадры канала D2 помимо 4×10^3 битов d содержат в среднем столько же битов i плюс некоторое число битов кодов Q (прямоугольники нижней диаграммы растянуты по горизонтали в два раза по сравнению с прямоугольниками верхней диаграммы).

В процессе передачи данных между устройствами 1 и 2 (рис. 1, *а*) ими регистрируются случайные события S_i опознания заданного кода Z = 010…101 на линии TxD (рис. 2). Эти события условно показаны на рис. 4 в виде вспышек света. При каждом опознании кода Z устройство 1 (источник данных) вставляет в передаваемый по линии CLK поток кадров заранее подготовленное 19-разрядное слово Q служебных данных.

Устройство 2 (приёмник данных) одновременно с устройством 1 распознаёт код Z, получает слово Q и после его анализа определяет, где находится начало ближайшего кадра потока D1 или D2. Если кадровая синхронизация ранее уже была достигнута, то полученная устройством 2 информация лишь подтверждает её правильность. Если кадровая синхронизация ещё не была достигнута, то полученная информация позволяет её установить.

Событие S1 (рис. 4) сопровождается передачей 19-разрядного служебного кода Q, в котором содержится число Delta D1, равное расстоянию до начала ближайшего кадра потока D1. Это расстояние измеряется в числе битовых интервалов между моментом обнаружения заданного кода Q = 010…101 в скользящем окне и моментом получения первого бита следующего кадра. Устройство 2 загружает это число в счётчик тактов, работающий в режиме вычитания. К моменту опустошения счётчика в устройство 2 поступает первый бит очередного кадра канала D1.

Аналогично событие S2 сопровождается передачей кода Delta D2, который определяет длину интервала от момента обнаружения заданного кода Q до момента начала ближайшего кадра потока D2. Этот интервал задаётся числом битов d. Служебные биты i и Q_j (если они имеются) при подсчёте не учитываются. (Как уже отмечалось, приёмник “прорисовывает” временные диаграммы, аналогичные приведенным на рис. 1, *д*, и по ним определяет местоположение битов d.) Точно так же события S3 и S4 несут информацию о положении ближайших кадров потоков D1 и D2 и т. д.

В отличие от общепринятого способа указания границ между кадрами (когда каждый из них снабжается флагом), применение вероятностной синхронизации позволяет передавать значительно меньший объём сведений об этих границах без ущерба для функционирования системы передачи данных. В установившемся режиме приёмник ведёт учёт кадров автономно в соответствии с собственной системой их отсчёта и лишь “изредка” получает числа Delta D1 и Delta D2, которые рассматривает как контрольные сведения, подтверждающие правильность своей ориентации во времени.

Напомним, что события S_i могут сопровождать и иную информацию (определяемую разрядами Q0 – Q6 кода Q служебных данных), поэтому для кадровой синхронизации может использоваться лишь незначительная их доля, например, 0,1%.

В табл. 1 показаны возможные кодовые ситуации, соответствующие событиям S1 – S4 (рис. 4).

Таблица 1. Служебные данные, передаваемые в моменты регистрации событий S1 – S4

Обозначение случайного события (рис. 4)	Код служебных данных Q0 – Q18 (рис. 2)		
	Q0	Q1 – Q6	Q7 – Q18
S1	1	111000 (Delta D1)	$110101110001 = 3441_{10}$ = Delta D1
S2	1	111001 (Delta D2)	$011000000000 = 1536_{10}$ = Delta D2
S3	1	111000 (Delta D1)	$111111111010 = 4090_{10}$ = Delta D1
S4	1	111001 (Delta D2)	$100000000000 = 2048_{10}$ = Delta D2

Из таблицы следует, что события S1 – S4 сопровождаются передачей кодов служебных данных, в которых Q0 = 1. Как уже отмечалось, это означает, что в разрядах

Q1 – Q18 этих кодов присутствуют собственно служебные данные, а не биты d. Коды 111000 и 111001 в разрядах Q1 – Q6 задают типы данных, размещённых в разрядах Q7 – Q18. В данном примере код 111000 (111001) показывает, что последующие биты задают расстояние Delta D1 (Delta D2) до начала ближайшего кадра потока D1 (D2). В правом столбце таблицы приведены численные значения расстояний.

Архитектурные очертания предлагаемой системы передачи данных

Покажем принципиальную возможность реализации предлагаемой системы. Напомним, что она содержит соединенные линиями TxD и CLK устройства 1 и 2 – соответственно, источник и приёмник данных (рис. 1, *а*). Кодирование данных устройством 1 и их декодирование устройством 2, как предполагаем, выполняется программно.

На первом этапе подготовки данных к передаче они размещаются в памяти устройства 1 (рис. 5, *а*). Данные представлены нескремблированными последовательностями битов D1’ и D2. Эти последовательности разделены на информационные кадры. Кадры последовательностей D1’ и D2, вообще говоря, могут иметь разную или переменную длину. В последнем случае информация о длине кадра содержится в нём самом.

На втором этапе (рис. 5, *б*) последовательность D1’ программно скремблируется, т. е. преобразуется в последовательность D1, которую можно рассматривать как случайную. С помощью программного “скользящего окна” в этой последовательности выявляются коды Z = 010…101. Напомним, что всякий раз после выявления такого кода окно скачком перемещается на 20 бит вправо, после чего поиск очередного кода Z продолжается с шагом в один бит. Последовательность битов D2 остается неизменной.

На третьем этапе (рис. 5, *в*) последовательность D1 остаётся неизменной, а биты d последовательности D2 “разбавляются” битами i синхронизации, как было показано на рис. 1, *д*. Кроме того, под каждым кодом Z размещается код Q, как было показано на рис. 2 и 4. В процессе кодирования кадры последовательности D2 растягиваются вправо примерно в два раза. В результате исходно одинаковые по длине строки битов (рис. 5, *а*) становятся разными (рис. 5, *в*), поэтому для выравнивания длин требуется дополнительная “подкачка” и преобразование данных D1’, а также необходимы другие (описанные ранее) операции.

Чтобы совместить во времени процессы кодирования и выдачи данных в линии TxD и CLK, удобно применить конвейерный способ их обработки с использованием буферной памяти типа FIFO (First In – First Out). Такую память можно смоделировать программно на основе обычного оперативного запоминающего устройства. Как показано на рис. 6, буферную память можно условно представить в виде пары концентрических колец из битовых ячеек, пронумерованных от 0 до M – 1, где M – объём буфера.

Внутреннее кольцо предназначено для временного хранения фрагмента строки битов D1 (рис. 5, *в*) внешнее – для хранения фрагмента строки D2 + SYN + D3 (SYN – обозначение множества битов i синхронизации). Примерно по диаметру кольца размещены “бегущие по кругу” процессоры считывания и записи данных в буфер.

Как показано на рис. 6, в некоторый момент процессор считывания данных извлекает из пары ячеек памяти с адресом M – 2 два бита и пересылает их в линии TxD и CLK. В следующих тактах очередные пары битов считываются и передаются в канал связи из ячеек с адресами M – 1, 0, 1, 2 и т. д. Процессор шаг за шагом продвигается по дугам, изображённым на рисунке утолщёнными линиями. Эти дуги соответствуют ячейкам памяти, в которых размещены данные, подготовленные к выдаче в линии TxD и CLK. Скорость V продвижения процессора по дугам постоянна и равна 1/T, где T – длительность битового интервала в канале связи.

Процессор записи данных выполняет все описанные ранее операции (рис. 5) по подготовке данных и последовательно записывает результаты в ячейки с адресами M/2 – 2, M/2 – 1, M/2 и т. д. Средняя скорость записи результатов кодирования в бу-

ферную память должна быть в точности равна скорости V для того чтобы эта память не переполнялась и не опустошалась.

Для выполнения этого условия скорость кодирования выбирается несколько более высокой, чем требуемая. В необходимых случаях, когда средняя угловая длина дуг, показанных утолщёнными линиями, заметно превышает 180 град., кодирование замедляется выполнением “пустых” команд, и наоборот, эти команды не выполняются, если средние длины дуг становятся короче половин окружностей. Длины дуг вычисляются сравнением адресов записи и чтения, которыми оперируют процессоры.

Благодаря наличию буферной памяти процессор, выполняющий кодирование, может пополнять её данными с неравномерным темпом в зависимости от возникающих кодовых ситуаций. Чем больше буферная память, тем больше у этого процессора “свободы” в распределении собственного времени.

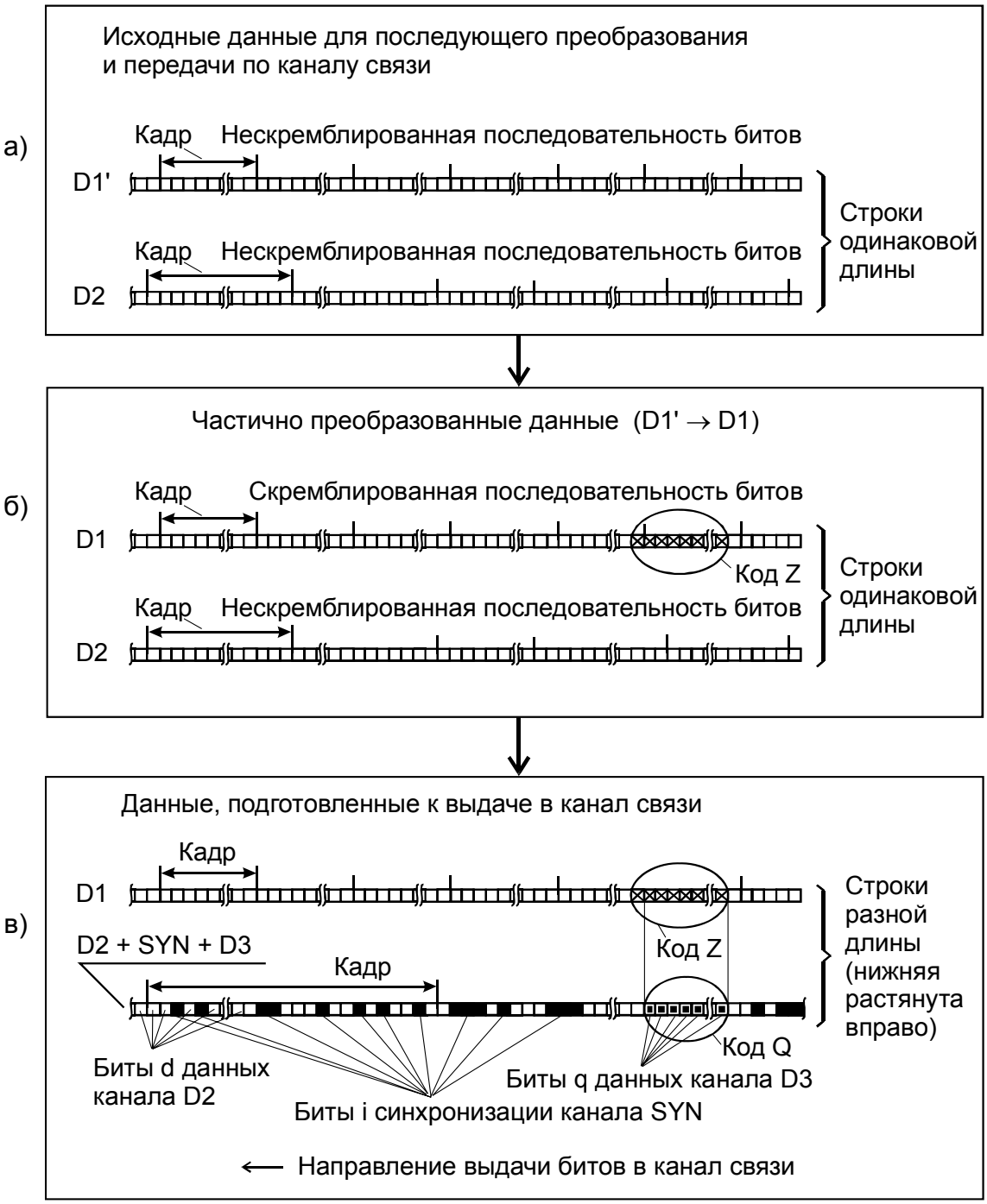

Рис. 5. Диаграммы, поясняющие процесс подготовки исходных данных к выдаче в линию связи

Приёмное устройство 2 (рис. 1, *а*) может содержать такую же буферную память и два “бегущих по кольцу” процессора. Первый процессор принимает пары битов из канала связи и с постоянным темпом записывает их в последовательные ячейки буферной памяти. Второй процессор, расположенный примерно на противоположной стороне диаметра кольца (по аналогии с диаграммой на рис. 6) осуществляет декодирование данных. Средняя скорость декодирования должна в точности совпадать со скоростью V, но работа декодирующего процессора может быть в значительной степени неравномерной с точки зрения темпа считывания данных из буфера.

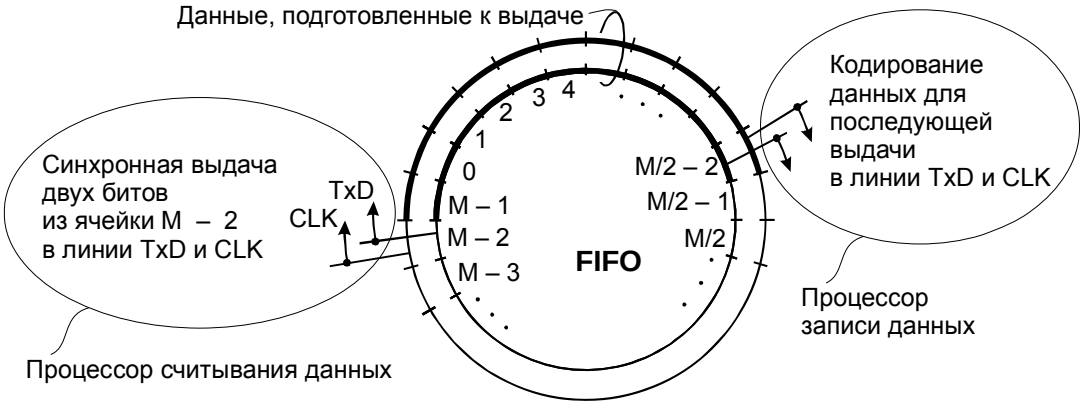

Рис. 6. Совмещение кодирования с выдачей данных в канал связи

Ранее предполагалось, что скремблер и дескремблер построены на основе неизолированных генераторов псевдослучайных последовательностей битов [1] или их программных моделей. Это обеспечивает их кодовую самосинхронизацию, но приводит к размножению ошибок, которые могут возникать в канале связи. Кроме того, такая система неустойчива по отношению к некоторым неблагоприятным кодовым ситуациям.

Для устранения этих недостатков можно применить изолированные генераторы псевдослучайных последовательностей битов (или их программные модели).В этом случае кодовая синхронизация между изолированными генераторами устройств 1 и 2 может осуществляться их одновременной установкой в одно и то же начальное состояние при обнаружении кодов Z.

Заключение

В этой статье предложена система передачи данных с попутной синхронизацией. Её прототипами являются системы с синхронными интерфейсами семейства RS (RS-232, V.35 и другие). Предлагаемая система обладает следующими свойствами.

1. Линия TxD, как и в известных системах, используется для передачи данных с гарантированной скоростью V = 1/T, где T – длительность битового интервала в канале связи. Однако, в отличие от известных систем, при использовании кадровой структуры данных нет необходимости передавать по этой линии служебную информацию. Кадры следуют по ней “встык” без каких-либо разделителей, что повышает эффективную скорость передачи данных.
2. Линия CLK синхронизации используется не только по прямому назначению. По ней передаются данные дополнительного канала связи со скоростью, примерно равной V/2. Помимо этого, по линии CLK в случайные моменты времени передаются данные управляющего канала, если это необходимо. Скорость передачи данных по этому каналу можно регулировать выбором параметров системы.
3. Приёмник синхронизируется в каждом такте, так как на границах между битовыми интервалами, по крайней мере, на одной из линий (TxD или CLK) присутствует перепад уровней напряжения.
4. Алгоритмы кодирования и декодирования данных несложны и могут быть реализованы программно.

ЛИТЕРАТУРА:

1. *Б.В.Шевкопляс. Скремблирование передаваемых данных. — Схемотехника, 2004, №12, с. 24 — 27, 2005, №1, с. 29 — 32, 2005, №2, с. 32 — 35, 2005, №3, с. 30 — 33. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
2. *Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: вставка команд в поток данных без использования избыточных битов. – Схемотехника, 2005, № 5, с. 23 – 25, № 6, с. 23 – 26. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
3. *Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: разграничение байтов в битовом потоке данных. — Схемотехника, 2005, №8, с. 23 — 26, №9, с. 23 — 25. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
4. *Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: разграничение каналов в мультиплексированном потоке данных. — Схемотехника, 2005, №11, с. 17 — 19, №12, с. 18 — 22. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
5. *Пат. США № 6269414 B1 http://www.uspto.gov*
6. *Пат. США № 6044421 http://www.uspto.gov*
7. *С.М.Сухман, А.В.Бернов, Б.В.Шевкопляс. Синхронизация в телекоммуникационных системах. Анализ инженерных решений. – М.: Эко-Трендз, 2003. Электронная версия книги*: *http://lit.lib.ru/s/shewkopljas_b_w/*

Б. Шевкопляс