

## 4. Проскальзывания синхронизации

### 4.1. Причины и следствия проскальзываний

Термин “проскальзывание” (slip) применяют для обозначения весьма характерной ошибки при передаче данных по цепи устройств в отсутствие общего для этой цепи синхросигнала. Поясним сказанное. Система передачи данных, показанная на рис. 4.1, *а*, не подвержена проскальзываниям. Она содержит три телекоммуникационных устройства X, Y и Z, соединенных линиями L1 и L2. Номинальная скорость передачи данных по линиям равна 1 Мбит/с. Фактическая скорость передачи в данном случае задается генератором G1 и немного отличается от номинальной. Частота синхросигнала от этого генератора составляет, например, 1 000 010 Гц, т. е. превышает номинальную на 10 Гц.

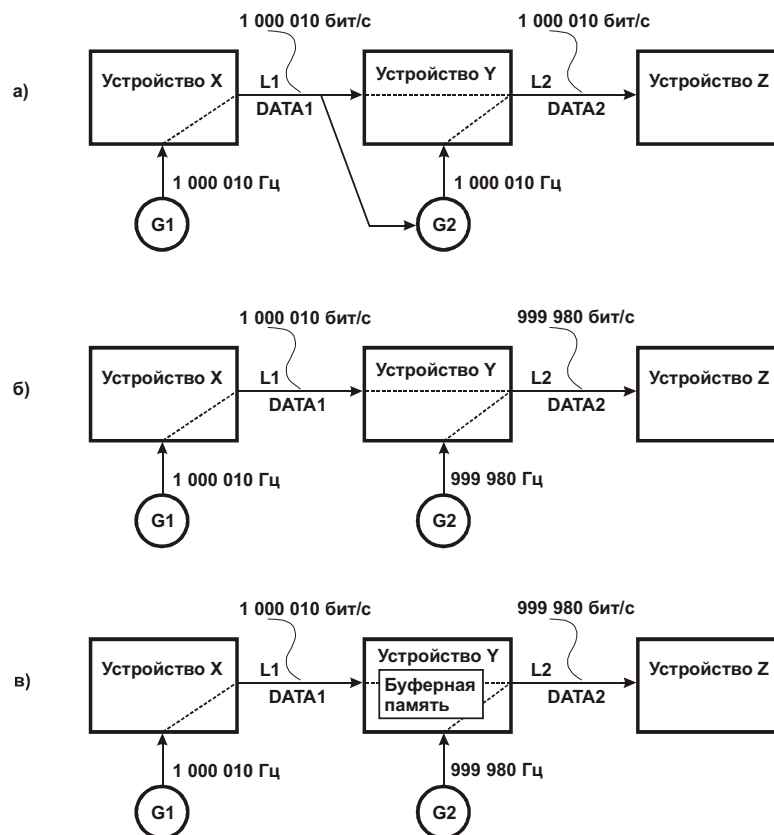


Рис. 4.1. Синхронизация передачи данных: *а* – правильная (без проскальзываний); *б* – неправильная (с проскальзываниями, когда буферной памяти нет); *в* – допустимый вариант (при условии регулирования уровня заполнения буферной памяти)

Передаваемый по линии L1 поток данных DATA1 в неявном виде содержит синхросигнал от генератора G1. Этот синхросигнал восстанавливается генератором G2, выполненным по схеме с фазовой автоподстройкой частоты. Принятые устройством Y данные DATA1 транслируются в линию L2 под управлением синхросигнала от генератора G2. Существенно, что данные DATA1 и DATA2 передаются по линиям L1 и L2 с точно совпадающими скоростями.

Проскальзывания могут наблюдаться в том случае, когда генератор G2 не синхронизирован с генератором G1. В примере, приведенном на рис. 4.1, *б*, частота синхросигнала от генератора G2 ниже номинальной на 10 Гц. Разность частот синхросигналов от генераторов составляет 30 Гц. Этот факт можно интерпретировать следующим

образом. В устройство Y за одну секунду “вытекают” 1 000 010 битов. За это же время из устройства Y “вытекают” 999 980 битов. Куда делись 30 битов? Они либо просто потеряны (это и есть проявление отрицательных битовых проскальзываний), если в устройстве Y нет буферной памяти, как на рис. 4.1, б, либо сохранены в этой памяти, если она имеется и еще не переполнена (рис. 4.1, в).

Положительные битовые проскальзывания наблюдаются при обратном соотношении частот синхросигналов, когда вытекающий поток данных DATA2 более интенсивен, чем втекающий. В отсутствие буферной памяти в выходной поток данных в течение одной секунды будут внедрены 30 лишних битов, отсутствующих во входном потоке DATA1. При наличии буферной памяти, исходно в той или иной мере заполненной данными, выходной поток данных полностью соответствует входному до тех пор, пока уровень заполнения этой памяти снижается, но остается ненулевым. (Положительные и отрицательные битовые проскальзывания более подробно рассмотрены в п. 4.3.1.)

Буферная память способна временно сглаживать разность скоростей втекающего и вытекающего потоков данных. Чем больше ее объем, тем дольше период безошибочной работы. Однако увеличение объема памяти нежелательно, так как при этом возрастает задержка передачи данных через устройство Y. При переполнении или опустошении буферной памяти проскальзывание все же возникает, причем из выходного потока удаляется (или в него внедряется) группа битов, соответствующая полному объему памяти. Иными словами, устройству Y на протяжении некоторого времени удастся сглаживать разность скоростей втекающего и вытекающего потоков, но в какой-то момент дальнейшая отсрочка “решения накопившихся проблем” становится невозможной, в результате происходит потеря или повтор передачи некоторого массива битов. (Более подробно об этом – см. п. 4.4.) Однако из сказанного не следует, что ситуация безнадежна; проскальзывания, как показано далее (п. 4.2), можно предотвратить.

Влияние проскальзываний на качество передачи различного рода данных приведено в Таблица 4.1. табл. 4.1 [2].

Таблица 4.1. табл. 4.1

Проявления проскальзываний синхронизации для разных типов данных или технологий их передачи

Тип данных или технология их передачи	Проявления проскальзываний синхронизации
Звуковые данные (музыка, речь)	Щелчки при прослушивании музыки, речи
Передача факс-сообщений	Неправильный текст
Передача сообщений по электронной почте	Неправильный текст или необходимость повторной передачи
Видео-информация	Искажение изображения (например “замораживание” или потеря картинки)
Технология SONET/SDH	Потеря данных
Технология ATM	Неправильные данные
Технология DSL	Потеря пакетов

Далее рассмотрены методы и схемы предотвращения проскальзываний и устранения их последствий.

Возможность предотвращения проскальзываний путем введения избыточных битов в потоки данных показана в п. 4.2. Для стабилизации уровня заполнения буферной памяти из потока данных исключается или в него вводится нужное число избыточных битов. Это позволяет компенсировать повышение или снижение уровня заполнения бу-

ферной памяти, вызываемое неточным равенством частот синхросигналов от генераторов G1 и G2 (см. рис. 4.1, в).

В п. 4.3 описан метод частичного восстановления кадра с ошибкой, вызванной битовым проскальзыванием. Ошибку в ряде случаев можно локализовать с точностью до одного или нескольких байтов, вместо того чтобы полностью отбрасывать ошибочный кадр. Это может оказаться полезным при передаче речевых данных, когда “ремонт” кадра позволяет сохранить часть передаваемых цифровых отсчетов аналогового аудиосигнала и тем самым уменьшить искажения при его прослушивании.

В п. 4.4 рассмотрены методы и схемы диагностики проскальзываний в многоканальной системе передачи данных. Примечательно, что для обнаружения проскальзывания, затрагивающего все каналы, достаточно иметь доступ только к одному из них.

В п. 4.5 отмечается, что при передаче “оцифрованных” речевых сигналов можно в незначительных пределах (незаметно для слушателя) изменять длительность пауз между словами. Это позволяет избежать проскальзываний благодаря стабилизации уровня заполнения буферной памяти путем исключения или добавления к ее содержимому кодов, соответствующих паузам между словами.

В п. 4.6 приведены схемы сопряжения устройств, синхронизируемых от независимых генераторов синхросигналов. Почти невероятно, но – факт: при определенных условиях можно исключить проскальзывания при взаимодействии не синхронизированных между собой устройств без использования буферной памяти!

## 4.2. Каким образом избежать проскальзываний

Основной способ предотвращения проскальзываний основан на введении избыточных битов в потоки данных. Если буферная память (см. рис. 4.1, в) близка к переполнению, то часть этих битов вычеркивается устройством Y из входного потока L1 и не запоминается. В результате темп поступления данных в буферную память снижается, уровень ее заполнения уменьшается. Аналогично при чрезмерном снижении уровня во входной поток данных при его размещении в памяти вставляются избыточные биты, которые постепенно повышают уровень ее заполнения до номинального (близкого 50%). Поясним сказанное.

Предположим, что по линии могут передаваться кадры трех типов: А, В и С (рис. 4.2). Кадр любого типа содержит флаговый код (флаг), обозначающий его начало, и поле данных фиксированной длины. Кадр А не содержит избыточных битов, в кадры В и С включены соответственно один и два избыточных бита R. Номинальная скорость передачи данных рассчитывается исходя из условия, что используются только кадры В; кадры А и С включаются в поток лишь в связи с необходимостью коррекции уровня заполнения буферной памяти. Из рисунка следует, что скорость передачи данных может отличаться от номинальной в ту или иную сторону, если в последовательность кадров ...ВВВ... внедрить кадры или группы кадров А или (и) С. Приемнику известна длина неизбыточного кадра (А), поэтому он легко отыскивает биты R в последовательности кадров. Эти биты, если они есть, размещены между последним битом поля данных предыдущего кадра и первым битом флага последующего кадра.

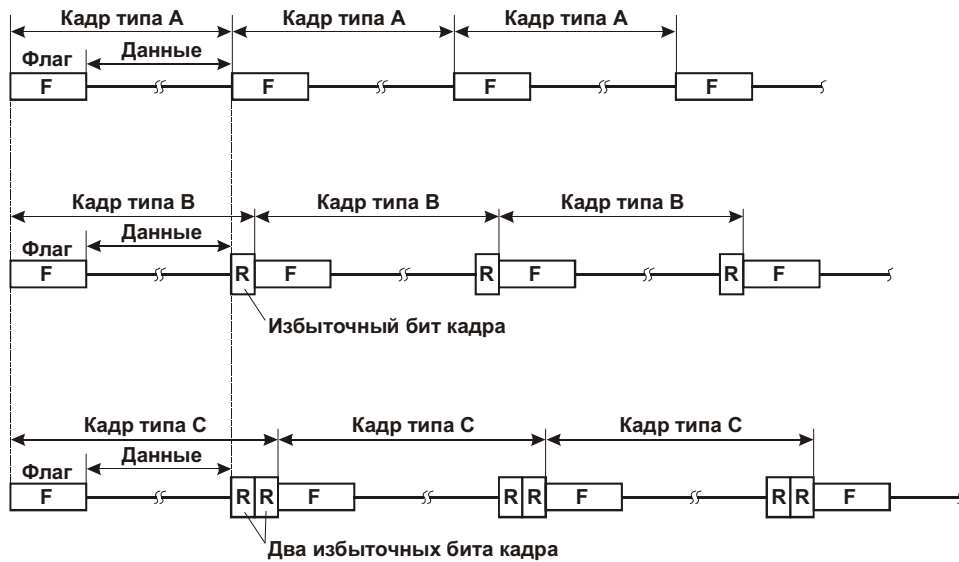


рис. 4.2. Структура кадров, используемых для предотвращения проскальзываний

Прежде чем пояснить, каким образом используются избыточные биты, рассмотрим модель прохождения последовательности кадров через буферную память типа FIFO (рис. 4.3). Память представлена в виде бака с жидкостью. Скорости втекающего и вытекающего потоков жидкости ассоциированы с частотами CLK1 и CLK2 синхронизации входного и выходного потоков битов.

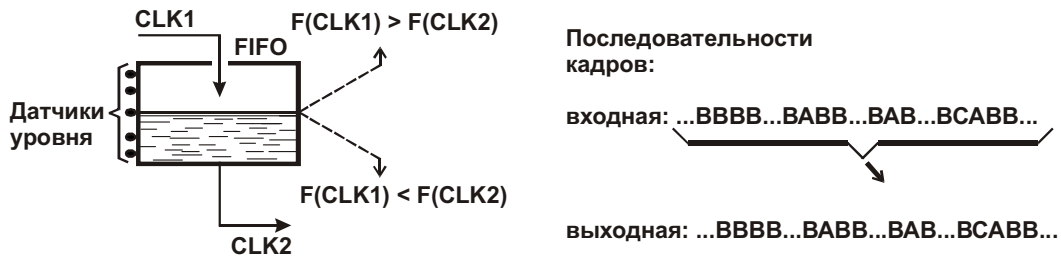


рис. 4.3. Прохождение последовательности кадров через буферную память типа FIFO в отсутствие коррекции уровня ее заполнения

В данном примере входная последовательность кадров передается на выход без преобразования. При этом уровень заполнения буферной памяти не может оставаться стабильным. Если частота  $F(\text{CLK1})$  входного синхросигнала превышает частоту  $F(\text{CLK2})$  выходного, то уровень повышается, и наоборот. Степень заполнения буферной памяти контролируется с помощью датчиков уровня. При заметном отклонении уровня от отметки “50%” следует принять меры по его принудительному возврату в допустимый диапазон.

Рассмотрим один из возможных способов стабилизации уровня заполнения буферной памяти с помощью вычеркивания или добавления избыточных битов (рис. 4.4, 4.5).

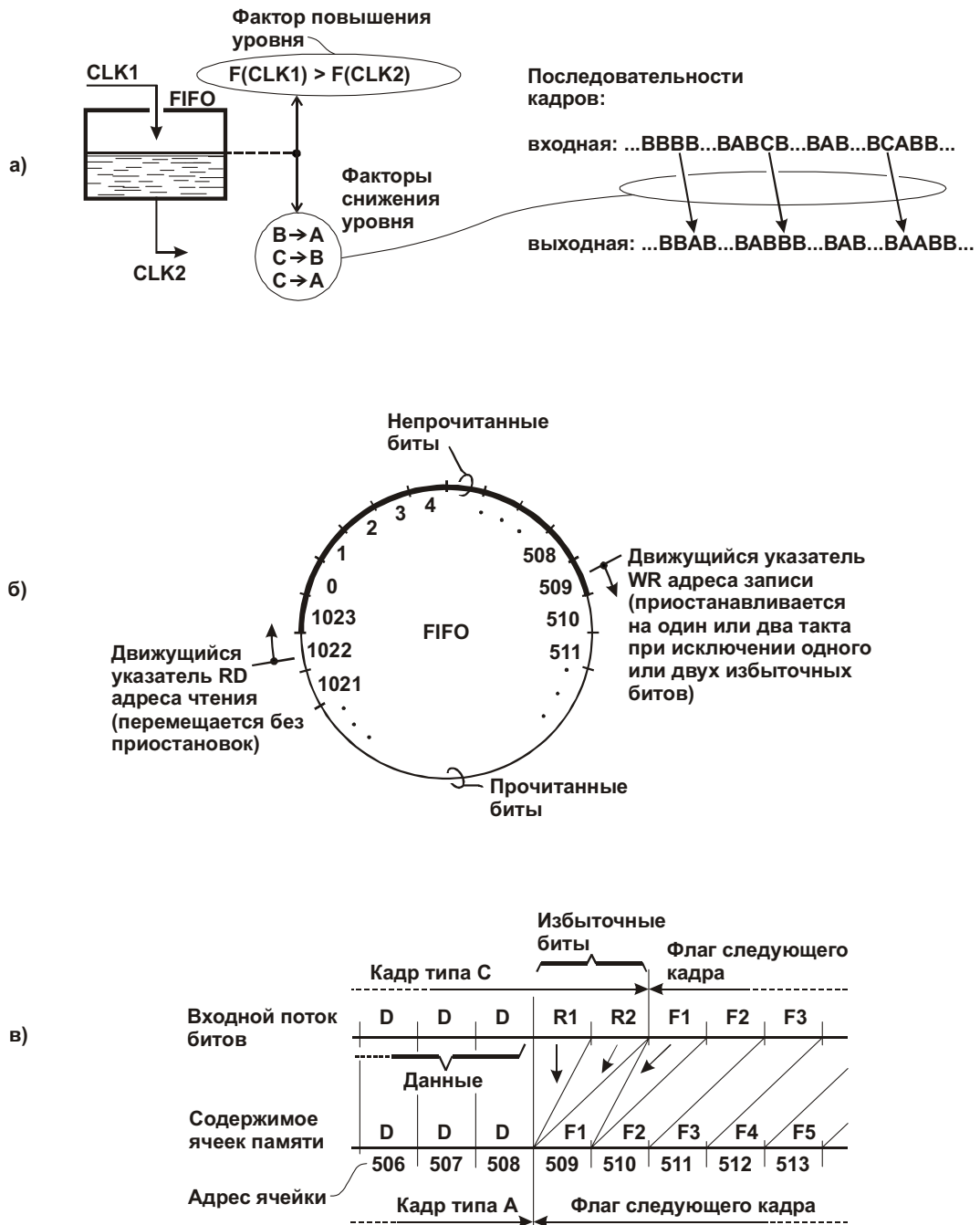


Рис. 4.4. Стабилизация уровня заполнения буферной памяти при положительной разности скоростей входного и выходного битовых потоков ( $F(\text{CLK1}) > F(\text{CLK2})$ ): а – общая модель; б – модель динамики заполнения буферной памяти; в – модель процесса исключения двух избыточных битов

Предположим, что частота входного синхросигнала превышает частоту выходного ( $F(\text{CLK1}) > F(\text{CLK2})$ ), например, на 30 Гц. Как показано на рис. 4.4, а, такое соотношение частот стремится вызвать повышение уровня заполнения буферной памяти. В отсутствие противодействующих факторов за одну секунду уровень заполнения буферной памяти мог бы повыситься на 30 бит. В качестве факторов снижения уровня выступают преобразования некоторых кадров, при которых из них исключаются один или два избыточных бита: кадр В преобразуется в кадр А, кадр С – в кадры В или А.

Для стабилизации уровня заполнения буферной памяти необходимо за одну секунду преобразовать ряд передаваемых кадров так, чтобы исключить из битового потока в

общей сложности 30 избыточных битов (или несколько больше, тогда уровень начнет понижаться). Задача разрешимая, если учесть, что за одну секунду через буферную память проходят несколько тысяч кадров, и подавляющее большинство из них – кадры В, содержащие по одному избыточному биту. Преобразование кадра поясняется рис. 4.4, б. На этом рисунке приведена уточненная модель буферной памяти, позволяющая более детально описать протекающие в ней процессы.

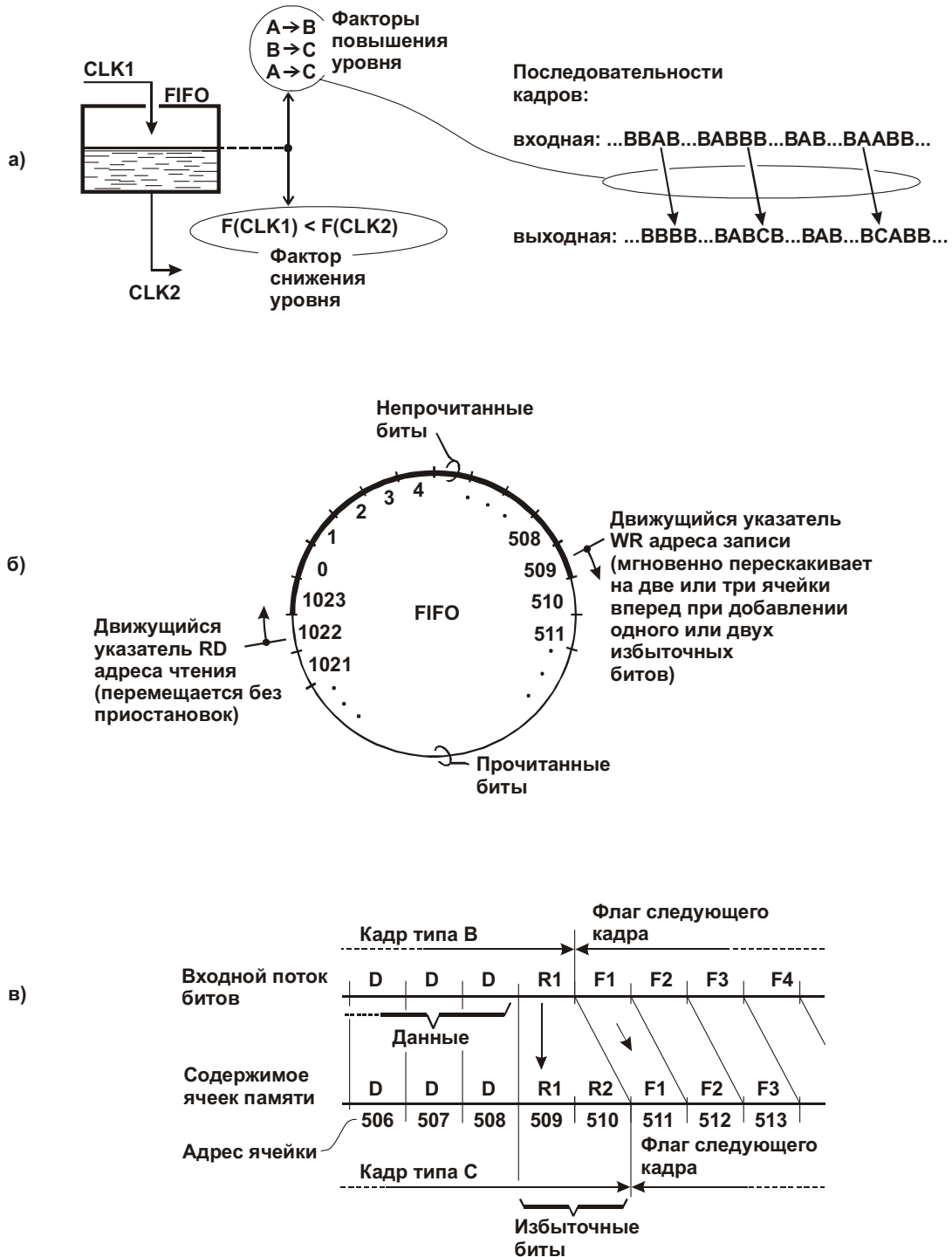


рис. 4.5. Стабилизация уровня заполнения буферной памяти при отрицательной разности скоростей входного и выходного битовых потоков ( $F(\text{CLK1}) < F(\text{CLK2})$ ): а – общая модель; б – модель динамики заполнения буферной памяти; в – модель процесса вставки одного избыточного бита

Память представлена кольцом из 1024 однобитовых ячеек. Запись и считывание битов втекающего и вытекающего потоков происходит по соответствующим адресам, которые задаются указателями WR и RD. Эти указатели могут быть выполнены в виде двух независимых десятиразрядных регистров-счетчиков. Как показано на рисунке утолщенной дугой, “расстояние” между этими указателями примерно равно половине объема буферной памяти, что соответствует уровню ее заполнения близкому 50%. Указатели перемещаются по часовой стрелке с почти одинаковыми скоростями и за одну секунду совершают несколько тысяч оборотов. В отсутствие преобразования кадров за это время из-за разности частот синхросигналов CLK1 и CLK2 длина дуги, соответствующей непрочитанным битам, увеличилась бы на 30 ячеек. Преобразование кадра осуществляется так.

Предположим, что в буферную память вводится кадр С (рис. 4.4, в). Последние три бита поля данных этого кадра в трех последовательных тактах размещаются в ячейках с адресами 506 – 508. В следующем такте в ячейку с адресом 509 записывается первый избыточный бит R1. В следующих двух тактах в эту же ячейку сначала записывается бит R2, а затем первый бит F1 флага очередного кадра. Этот бит замещает ранее записанные биты, так как указатель WR адреса записи в течение трех тактов оставался неподвижным. В последующих тактах биты очередного кадра (F2, F3 и т. д.) последовательно записываются в ячейки с последовательными адресами (510, 511 и т. д.). В результате выполнения описанных действий, из кадра С при его размещении в буферной памяти исключены избыточные биты R1 и R2, т. е. кадр С преобразован в кадр А. За время искусственной приостановки перемещения указателя WR адреса записи указатель RD адреса чтения в естественном порядке продвинулся на две ячейки, поэтому число непрочитанных битов сократилось на два. Иными словами, уровень заполнения буферной памяти снизился на два бита.

Рассмотрим противоположную ситуацию, при которой скорость передачи битов в выходном потоке данных немного превышает скорость передачи битов во входном потоке (рис. 4.5, а – в). Для противодействия снижению уровня заполнения буферной памяти в проходящий через нее поток битов вводятся дополнительные избыточные биты (в нашем примере – тридцать или более битов в течение одной секунды). Это достигается преобразованием кадров А в кадры В или С, кадров В – в кадры С.

Последние биты кадра В в трех последовательных тактах размещаются в ячейках с адресами 506 – 508. В следующем такте избыточный бит R1 кадра В запоминается в ячейке с адресом 509. В следующем такте первый бит F1 флага следующего кадра помещается в ячейку с адресом 511. При этом бит в ячейке с адресом 510 остается неизменным и рассматривается как второй избыточный бит формируемого кадра С. Значение этого бита может быть произвольным. Скачок указателя WR адреса записи через адрес 510 приводит к мгновенному увеличению числа непрочитанных битов на единицу. Это означает, что уровень заполнения буферной памяти повысился на один бит. Далее указатели WR и RD в течение некоторого времени перемещаются почти синхронно, пока не проявится очередной фактор повышения или снижения уровня заполнения буферной памяти.

Чтобы повысить эффективность использования избыточных битов, можно возложить на них функцию контрольных разрядов для проверки кадра на четность (нечетность) подобно тому, как это делается при передаче старт-стоповых посылок (см. п. 1.2). Можно не использовать кадры С, если рассчитывать номинальную скорость передачи битов, исходя из предположения, что кадры А и В следуют с равной вероятностью. При этом средняя вносимая в кадр избыточность составит 0,5 бита, а не один бит, как в рассмотренном примере.

Таким образом, для предотвращения проскальзываний при передаче непрерывного синхронного потока данных этот поток должен содержать избыточные служебные биты, число которых может варьироваться. Чтобы незначительно замедлить или ускорить

передачу данных, число избыточных битов увеличивают или уменьшают. Это позволяет стабилизировать уровень заполнения буферной памяти приемника, т. е. предотвратить ее переполнение или опустошение.

### 4.3. Частичное восстановление кадра, искаженного в результате битового проскальзывания

#### 4.3.1. Механизм битовых проскальзываний

Чтобы более наглядно продемонстрировать характер искажений, связанных с битовыми проскальзываниями, рассмотрим модель передачи данных между устройствами 1 и 2 в отсутствие буфера (рис. 4.6, б – г).

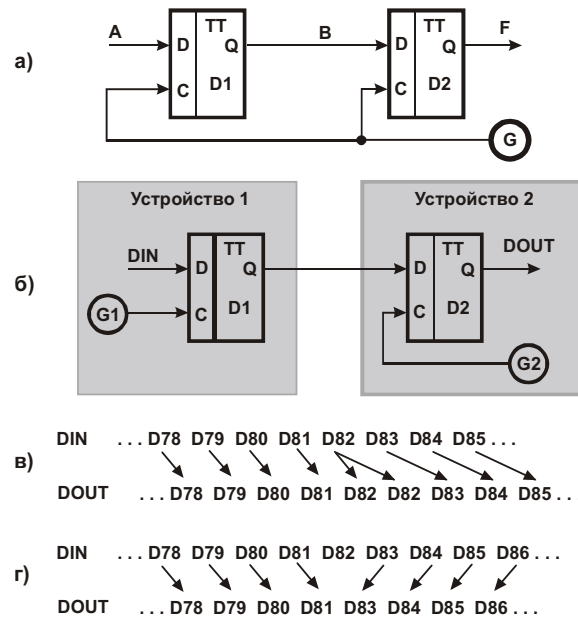


Рис. 4.6. Двухразрядный сдвиговый регистр (а) и неудачные попытки его использования для передачи данных между устройствами 1 и 2 (б – г)

Модель построена на основе сдвигового регистра. Классический сдвиговый регистр (рис. 4.6, а) содержит ряд последовательно включенных D-триггеров (в нашем примере – два триггера) с общей цепью синхронизации от генератора импульсов G. По положительному фронту синхроимпульса сигнал с выхода B триггера D1 переписывается на выход F триггера D2. Одновременно в триггер D1 принимается бит данных с входа A. Отрицательный фронт синхроимпульса не изменяет состояния сигналов в точках B и F. При поступлении следующего положительного фронта импульса процесс повторяется и т. д.

Совершенно иная картина наблюдается в том случае, когда триггеры D1 и D2 синхронизируются от независимых генераторов G1 и G2 с одинаковыми номинальными частотами (рис. 4.6, б). Сколь бы близкими ни были эти частоты, всегда существует дрейф взаимного фазового сдвига между синхросигналами. В некоторые моменты, повторяющиеся, например, с периодом 10 с, фазовый сдвиг равен нулю; в промежутках между ними он увеличивается в ту или иную сторону в зависимости от соотношения частот генераторов. В критические моменты, которые повторяются с тем же периодом, происходит дублирование либо потеря бита, передаваемого по цепочке триггеров.

Предположим, что частота генератора G2 чуть превышает частоту генератора G1. Тогда выходной поток данных DOUT будет чуть более интенсивным, чем входной поток DIN. В нашем примере через каждые 10 с к выходному потоку будет добавляться





“нарезается” на множество фрагментов, каждый из которых размещается в поле данных D кадра, приведенного на рис. 4.7.

Кадры последовательно передаются в канал связи. При этом скорость передачи данных в канале должна быть достаточно высокой, так как к потоку полезных данных D добавляется поток служебных.

Получатель кадра распознаёт флаг начала и приступает к приему данных. Если ошибок нет, то полезные данные извлекаются из кадра, преобразуются в равномерный поток отсчетов, совпадающий с исходным, и поступают на вход цифро-аналогового преобразователя. И, наконец, аналоговый сигнал усиливается и передается на динамик.

Как было показано, в результате проскальзывания может появиться лишний бит или исчезнуть один из имевшихся битов, что искажает кадр не только по содержанию, но и по форме. Большинство подобных ошибок проявляется в том, что по крайней мере один из битов синхронизации приобретает неправильное значение (в этом мы сможем убедиться на последующих примерах). Обычно кадры с искаженными битами синхронизации отбрасываются. При этом теряется целая группа отсчетов аналогового сигнала, по крайней мере та, которая следует после обнаружения неправильного бита синхронизации.

Правомерен вопрос: нельзя ли восстановить искаженные данные, хотя бы не во всех случаях и не полностью? Ведь чем меньше потерянных отсчетов аналогового сигнала, тем выше качество получаемого абонентом речевого сигнала.

Предлагаемая идея коррекции кадра [10] основана на таком “наблюдении”. При внедрении в кадр лишнего бита все последующие смещаются по цепочке вправо и вниз. Но это не приводит к хаосу в поврежденной части кадра; в нем возникает новый порядок. Как следует из рис. 4.8, оставшаяся невредимой часть битов синхронизации просто переместилась из первого столбца матрицы во второй. Поэтому, зная новое положение группы битов синхронизации, можно восстановить истинное положение оставшейся невредимой группы битов данных.

Аналогичная ситуация складывается и при потере бита (рис. 4.9); отличие в том, что биты синхронизации перемещаются из первого столбца матрицы в шестнадцатый.

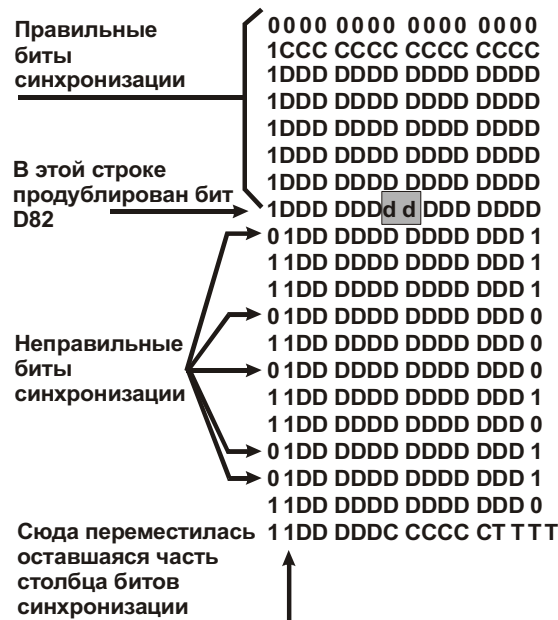


рис. 4.8. Кадр с ошибкой, вызванной проскальзыванием. Бит D82 превратился в два бита dd, последний бит T вытеснен за границу кадра

Предположим, что абонент принял кадр, показанный на рис. 4.8. “На всякий случай” принимается один дополнительный бит после окончания кадра, что, как видим, в

данной ситуации оказывается уместным. После анализа кадра выясняются следующие обстоятельства.

1. В первом столбце матрицы, где должны размещаться биты синхронизации, имеется пять ошибок. Маловероятно, что эти ошибки независимые, поэтому имеет смысл проверить, не переместился ли остаток группы битов синхронизации во второй или шестнадцатый столбец матрицы. Если это так, то появится шанс спасти искаженную часть кадра.

2. Во втором столбце матрицы действительно содержится сплошной массив из лог. 1, в шестнадцатом – такого массива нет (единицы чередуются с нулями). Поэтому есть основания полагать, что столбец битов синхронизации сместился вправо в результате внедрения в матрицу лишнего бита.

В подавляющем большинстве ситуаций положение этого бита можно определить только с точностью до строки. В нашем примере размножился бит D82, но процессор, анализирующий кадр, конечно, об этом не знает. Выявить точное местоположение “бита-дублера” можно лишь в частном случае, например, когда в строке содержится код 1010 1010 0101 0101. Здесь единицы чередуются с нулями везде, кроме искомой (выделенной особым шрифтом) пары позиций (dd = **00**). Зная, что эта строка содержит ошибку и что ошибка проявляется в виде повторения бита, можем уверенно определить ее положение.

Ошибочная строка помечается в памяти процессора, и в дальнейшем при формировании звукового сигнала соответствующие отсчеты, попавшие в эту строку, подменяются рассчитанными на основании наиболее вероятного прогноза изменения аналогового сигнала.

3. Зная новое положение остатка столбца битов синхронизации, можно реконструировать кадр. Для этого из ошибочной строки вычеркивается любой бит данных, а образовавшаяся вакансия заполняется сдвигом битов влево и вверх. При реконструкции учитывается дополнительный бит, ранее принятый в качестве возможного расширения кадра. В данном случае он соответствует четвертому биту T. В отсутствие ошибок дополнительный бит может принадлежать флаговой области следующего кадра.

Анализ кадра, приведенного на рис. 4.9, проводится аналогично. При этом позицию пропущенного бита D82 можно определить только с точностью до строки. В данном примере эта позиция помечена восклицательным знаком и занята соседним битом, который в отсутствие ошибок имел бы номер 83 (нумерация массива данных D: слева – направо, сверху – вниз, считая от единицы).

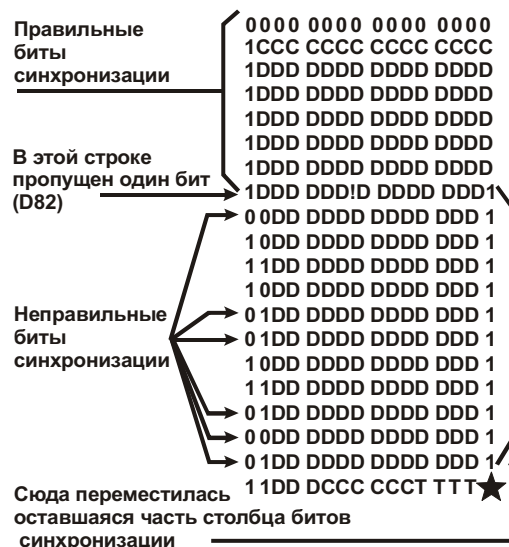


Рис. 4.9. Еще один пример кадра с ошибкой, вызванной проскальзыванием. Бит D82 исчез, “ряды сомкнулись”, в кадре появилась вакансия (помечена звездочкой)

При анализе кадра также учитывается, что вакантный бит, помеченный на рисунке звездочкой, может представлять собой первый бит (лог. 0) флага начала следующего кадра.

Для реконструкции кадра в ошибочную строку дополнительно помещается произвольный бит (лог. 0 или лог. 1), что сопровождается вытеснением имеющихся битов по цепочке вправо и вниз.

Прежде чем перейти к описанной ранее процедуре исправления кадра, процессор оценивает шансы на успех и, возможно, вообще отказывается выполнять процедуру. Действительно, могут наблюдаться разные проявления “игры в рулетку”, при которых попытка коррекции только усугубляет ошибочную ситуацию.

Например, может быть обнаружена не связанная с проскальзыванием простая одиночная ошибка в исходном (первом) столбце битов синхронизации, и в то же время в шестнадцатом столбце по “иронии судьбы” может присутствовать “нормальная” группа лог. 1. Если в данной ситуации применить рассмотренную ранее процедуру, то одиночная ошибка превратится в групповую, что недопустимо.

Вероятность правильного применения процедуры коррекции возрастает с увеличением числа обнаруженных ошибок в столбце битов синхронизации. Поэтому для оценки применимости процедуры выбирается некоторое пороговое значение числа таких ошибок. Например, порог может быть выбран равным трем, и тогда процедура не применяется, если обнаружены только одна или две ошибки в столбце битов синхронизации.

Вероятность правильного применения процедуры коррекции возрастает также с уменьшением номера строки, в которой обнаружен первый ошибочный бит синхронизации. Это связано с более уверенным распознаванием перемещения группы битов синхронизации во второй или шестнадцатый столбец матрицы. Поэтому процедура не применяется в случае, когда первый ошибочный бит синхронизации обнаружен в одной из нескольких нижних строк матрицы.

#### **4.4. Обнаружение проскальзываний в многоканальной системе передачи данных**

В многоканальных системах передачи данных низкоскоростные информационные потоки объединяются в более высокоскоростные, которые, в свою очередь, также могут объединяться, и т. д. Полученные в результате такого объединения высокоскоростные потоки данных передаются по каналам связи и затем разделяются на соответствующие низкоскоростные потоки. При передаче высокоскоростных потоков данных по каналам связи возможны проскальзывания, например из-за нарушения синхронизации между соседними узлами транспортной сети. Задача обнаружения проскальзываний может решаться на уровне анализа объединенных (высокоскоростных) потоков данных; однако это связано с некоторыми трудностями, поскольку необходим доступ к этим потокам, тестирующая аппаратура должна быть высокочастотной и достаточно “интеллектуальной” для анализа передаваемых кадров и т. п.

Рассмотренные далее три варианта построения тестовой аппаратуры позволяют обнаруживать проскальзывания в высокоскоростных потоках при наличии доступа только к одному низкоскоростному потоку, по которому передаются тестовые данные. Это позволяет упростить аппаратуру тестирования и выполнять проверку многоканальной системы передачи данных при работе только с одним каналом, не нарушая работу остальных.

##### *Первый вариант*

Многоканальная система передачи данных (рис. 4.10) содержит мультиплексоры MUX1, MUX2 и коммутатор SWITCH. Данные передаются по 24 дуплексным кана-

лам, скорость передачи данных в каждом канале по каждому направлению равна 64 кбит/с. Мультиплексоры MUX1 и MUX2 содержат по 24 порта RS-232. Каждая пара одноименных портов мультиплексоров обслуживает один канал. Скорость передачи данных по линиям MUX1 – SWITCH и SWITCH – MUX2 равна 1,544 Мбит/с.

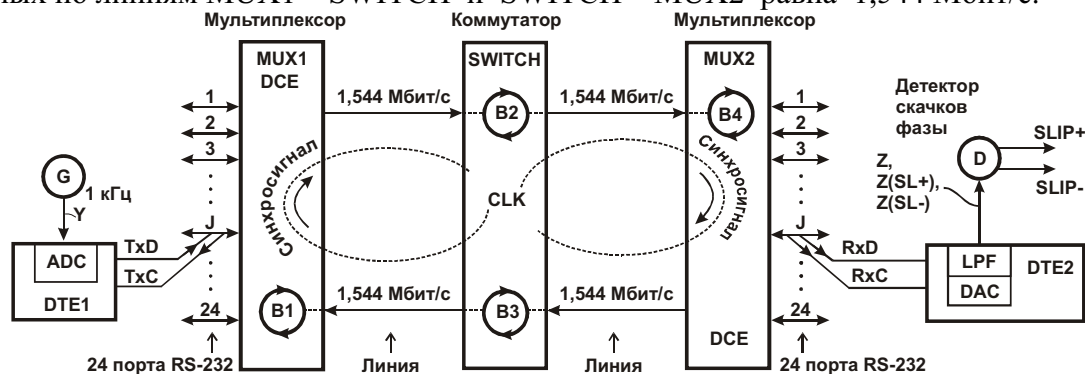


Рис. 4.10. Схема обнаружения проскальзываний в многоканальной системе передачи данных – первый вариант

Мультиплексоры и коммутатор содержат блоки В1 – В4 буферной памяти типа FIFO. Эти блоки предназначены для временного хранения текущего фрагмента последовательности битов, поступающих из линии. Объем блока буферной памяти равен числу битов в передаваемом кадре.

Передача данных, например слева направо, происходит так. Мультиплексор MUX1 циклически опрашивает каналы 1 – 24. К моменту опроса некоторого канала мультиплексор обнаруживает в соответствующем входном восьмиразрядном буфере (эти буферы на рисунке не показаны) очередную группу из восьми битов, предназначенных для выдачи в линию. Эту группу можно назвать байтом, но следует учесть, что истинные границы байта, установленные первоисточником данных, мультиплексору не известны. В буфере могут храниться, например, два бита одного и шесть битов соседнего байтов из данного канала. Тем не менее, можно условно считать, что мультиплексор MUX1 последовательно считывает байты данных из входных буферов каналов.

В каждом цикле опроса мультиплексор MUX1 компонуется кадр, содержащий 24 байта или  $24 \times 8 = 192$  бита данных и один служебный бит, всего 193 бита. Этот кадр передается в линию. Кадры следуют друг за другом без пауз – вслед за последним битом предыдущего кадра сразу же передается первый бит последующего. Частота следования кадров равна 8 кГц, время передачи кадра равно  $1/8000$  с = 125 мкс. Скорость потока данных в линии составляет  $193 \times 8000 = 1\,544\,000$  бит/с. Поток проходит через коммутатор в мультиплексор MUX2, который осуществляет обратное преобразование. Данные из расформированного кадра в виде непрерывных равномерных потоков со скоростями 64 кбит/с распределяются по соответствующим каналам 1 – 24. Одновременно с рассмотренными, аналогичные процессы протекают при передаче данных в обратном направлении.

Для исключения проскальзываний мультиплексоры и коммутатор должны синхронизироваться от одного и того же источника синхросигнала. В данном примере источник синхросигнала CLK размещен в коммутаторе. Мультиплексоры выделяют этот синхросигнал из входных потоков данных, поступающих по линии, и используют его для формирования всех остальных потоков. Таким образом, скорости всех потоков данных строго согласованы.

Однако по разным причинам, например, в результате неисправности или неправильных действий человека – оператора, одна или обе петли распространения общего сигнала CLK могут разомкнуться, т. е. целостность системы синхронизации может временно нарушиться. Тогда один или оба мультиплексора переходят к синхронизации от внутренних генераторов. При этом неизбежно возникают проскальзывания, поскольку

ку частоты не синхронизированных между собой генераторов не могут идеально совпадать. Если система используется для передачи речи, то проскальзывания проявляются в виде щелчков или иных кратковременных помех, что снижает качество связи. Однако при передаче цифровых данных проскальзывания приводят к необходимости повторных пересылок искаженных кадров под управлением протоколов высокого уровня. Если проскальзывания достаточно часты, то передача данных оказывается невозможной.

Чтобы зарегистрировать возможные проскальзывания, предлагается применить аппаратуру тестирования, состоящую из двух частей – передающей и приемной. Передающая аппаратура подключается к некоторому каналу  $J$  мультиплексора MUX1. К этому же каналу мультиплексора MUX2 подключается приемная аппаратура.

Передающая аппаратура [35] содержит генератор тестового синусоидального сигнала  $Y$  частотой 1 кГц и устройство DTE1, в состав которого, в частности, входит шестиразрядный аналого-цифровой преобразователь ADC. Преобразователь опрашивает сигнал  $Y$  с частотой 8 кГц, полученной в результате деления частоты синхросигнала  $TxC$  (64 кГц) на восемь. Каждый шестиразрядный цифровой отсчет аналогового сигнала  $Y$  дополняется стартовым и стоповым битами для последующего распознавания границ отсчета приемной аппаратурой тестера (см. п. 3.1). Полученные байты последовательным кодом передаются в мультиплексор MUX1 в виде непрерывного последовательного потока данных  $TxD$  со скоростью 64 кбит/с.

Под управлением синхросигнала  $RxC$  непрерывный поток данных  $RxD$  из канала  $J$  мультиплексора MUX2 принимается устройством DTE2. Из непрерывного потока данных выделяются байты, сформированные на передающей стороне. Стартовые и стоповые биты отбрасываются, шестиразрядные отсчеты сигнала  $Y$  подаются на вход цифро-аналогового преобразователя с частотой 8 кГц, полученной в результате деления частоты сигнала  $RxC$  (64 кГц) на восемь. Ступенчатый аналоговый сигнал сглаживается фильтром LPF низких частот, в результате на выходе устройства DTE формируется сигнал  $Z$ , по форме близкий сигналу  $Y$ .

В отсутствие проскальзываний сигнал  $Z$  повторяет сигнал  $Y$  с некоторым постоянным фазовым сдвигом относительно последнего. При положительных  $SL+$  и отрицательных  $SL-$  проскальзываниях сигналы  $Z(SL+)$  и  $Z(SL-)$  на выходе устройства DTE скачкообразно растягиваются и сжимаются на 45 угловых градусов (рис. 4.11). В этих случаях детектор  $D$  формирует соответствующие сигналы  $SLIP+$  и  $SLIP-$ .

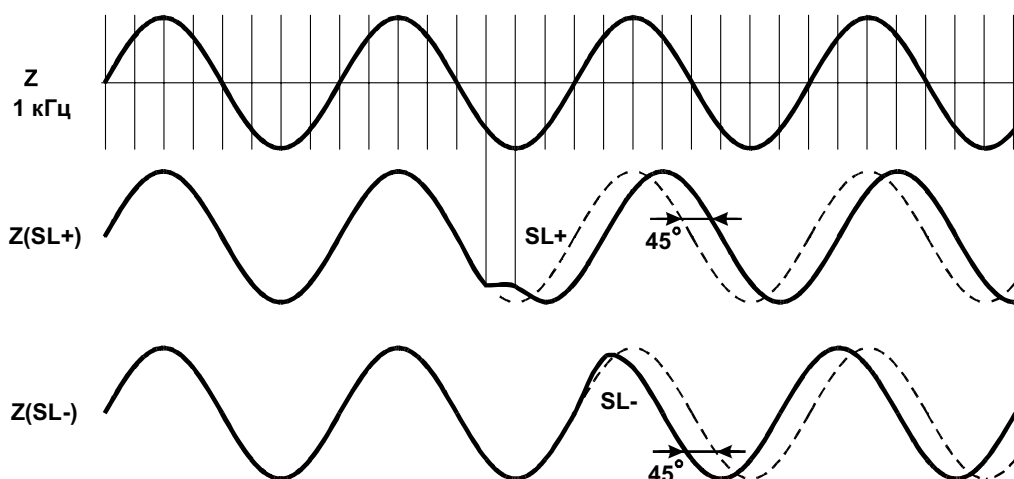


Рис. 4.11. Временные диаграммы тестового сигнала при отсутствии проскальзываний (верхняя диаграмма) и при их обнаружении (нижние диаграммы)

Чтобы убедиться в том, что проскальзывания проявляются именно так, необходимо рассмотреть динамику работы блоков  $B2$  и  $B4$  буферной памяти (см. рис. 4.10.). Эти

блоки расположены на трассе распространения тестовых данных и в равной мере участвуют в их передаче. Для определенности рассмотрим работу блока В4.

Как уже отмечалось, объем блока буферной памяти равен числу битов в передаваемом кадре. Буферную память удобно представить в виде кольцевой структуры из 193 одноразрядных ячеек (триггеров), как показано на рис. 4.12, а. Ячейкам присвоены постоянные адреса из диапазона 1 – 193. Вдоль кольца из ячеек дискретно перемещаются указатели адресов записи и чтения. По адресу записи в ячейку памяти заносится очередной бит данных, принятый из линии. По адресу чтения из ячейки извлекается бит, который используется для дальнейшей обработки.

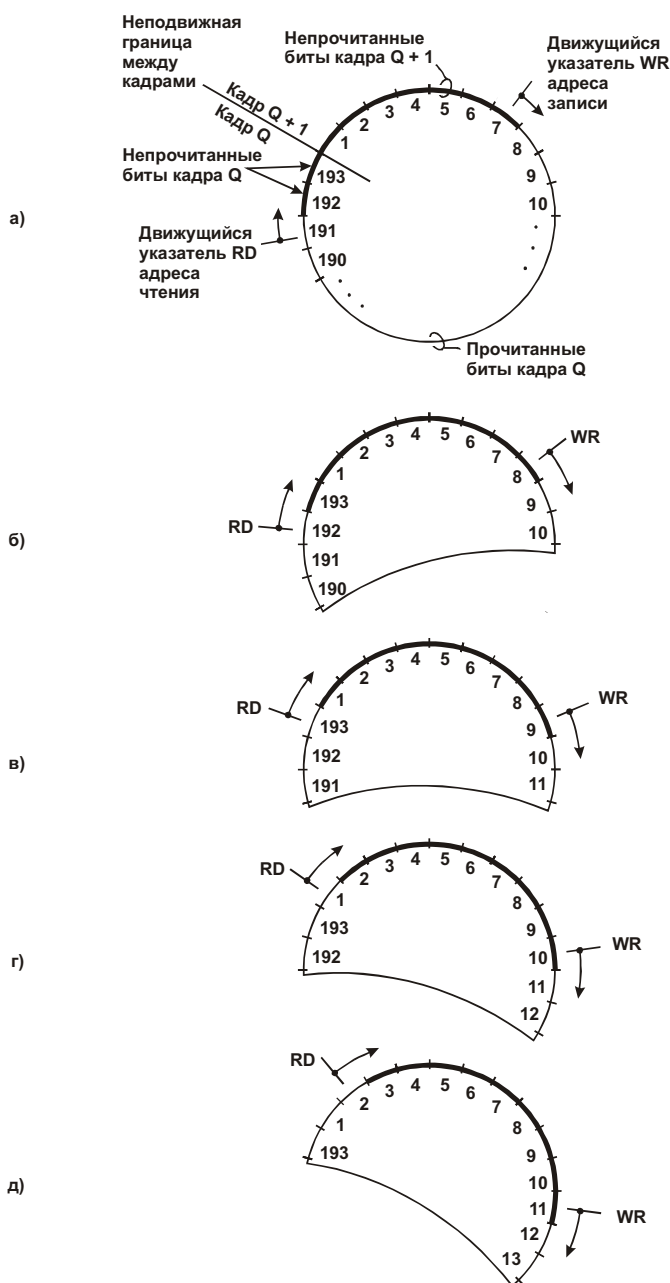


рис. 4.12. Прохождение данных через блок памяти В4 в отсутствие проскальзываний

При правильной работе системы передачи данных средние скорости перемещения указателей записи и чтения одинаковы. Расстояние между этими указателями (разность адресов соответствующих ячеек) должно быть ненулевым и предпочтительно равным

половине диапазона адресов. Проскальзывание возникает в том случае, когда расстояние между указателями сокращается до нуля при обгоне одного указателя другим.

На диаграмме, представленной на рис. 4.12, *а*, буфер в основном пуст, в том смысле, что ячейки с адресами 8 – 191 уже прочитаны аппаратурой мультиплексора MUX2 и их содержимое теперь не актуально. Для мультиплексора представляют интерес ячейки с адресами 192 – 7, которые содержат новые данные, только что записанные, но еще не считанные. Одна часть новых данных (содержимое ячеек 192 и 193) относится к некоторому почти прочитанному кадру  $Q$ , другая (содержимое ячеек 1 – 7) – к очередному кадру  $Q + 1$ , который лишь в течение ближайших семи тактов начал продвижение в буферную память. В следующих тактах оба указателя адресов последовательно и синхронно смещаются на одну позицию по часовой стрелке, как показано на рис. 4.12, *б* – *д*.

При правильной работе системы такая “гонка за лидером” может продолжаться сколь угодно долго. Расстояние между указателями может периодически изменяться из-за незначительных колебаний частоты выделенного из линии синхросигнала в результате действия фазовых помех (джиттера и вандера, см. гл. 5). Этот синхросигнал привязан к данным, поступающим из линии в буфер, поэтому скорость продвижения указателя  $WR$  колеблется в некоторых пределах. Скорость продвижения указателя  $RD$  более стабильна, так как она определяется тем же синхросигналом, но после устранения (или значительного подавления) содержащихся в нем фазовых помех. В идеальном случае средние положения указателей должны располагаться по диаметру кольца из ячеек. Тогда допустимая амплитуда колебаний скорости перемещения указателя  $WR$  максимальна, т. е. мультиплексор способен противостоять значительным фазовым помехам.

Предположим, что в результате неисправности или неправильных действий оператора мультиплексор MUX2 переключился в режим синхронизации от внутреннего автономного генератора синхросигналов. Его номинальная частота равна 1,544 МГц и совпадает с номинальной скоростью передачи данных по линии SWITCH – MUX2. Однако фактические значения частоты и скорости не могут идеально совпадать и различаются, например, на 30 Гц.

Положительное проскальзывание обусловлено тем, что скорость перемещения указателя  $RD$  оказывается более высокой, чем скорость перемещения указателя  $WR$ ; это приводит к “запрещенному обгону” одного указателя другим. Иными словами, скорость поступления данных из линии в буферную память меньше скорости их считывания из памяти. Предположим, что в исходном состоянии расстояние между указателями равно трем (рис. 4.13, *а*). Через некоторый промежуток времени  $T$ , соответствующий, например, двум – трем сотням пробегов указателей по кругу, дистанция между ними скачкообразно сокращается до двух (рис. 4.13, *б*). По истечении следующего промежутка времени  $T$  дистанция вновь сокращается и становится минимально допустимой (рис. 4.13, *в*). Наконец, через время  $T$  дистанция сокращается до нуля (рис. 4.13, *г*). Это означает, что импульс считывания бита из ячейки (выполненной, например в виде триггера) с каждым тактом приближается к импульсу записи в эту ячейку и в некоторый момент совпадает с ним. В этом случае результат считывания не определен, что отражено на рис. 4.13, *г* ромбом.

Подобная неопределенность продолжается в течение, например, нескольких тактов, но в дальнейшем указатели расходятся во времени на такой интервал, при котором чтение бита из триггера уверенно опережает запись (рис. 4.13, *д*). При этом, однако, считываются те же биты, которые были ранее считаны, т. е. происходит скачок на 193 бита назад. Это означает, что мультиплексор проводит повторный цикл опроса каналов, начиная с того, на котором произошел “обгон”. Из этого канала и из нескольких соседних может считываться комбинация старых и новых данных (ошибочная информация), из остальных каналов – правильные старые данные, которые уже были считаны. С учетом



этих замечаний для упрощения рассуждений можно полагать, что при положительном проскальзывании в последовательность кадров вводится дубликат одного из них.

После межкадрового скачка минимально допустимая дистанция между указателями в течение некоторого времени сохраняется. На рис. 4.13, *e – з* показано начало процесса заполнения буферной памяти новыми данными. В дальнейшем после ряда изменений дистанции между указателями они вновь сближаются (рис. 4.13, *и*). После очередного скачка дистанции буферная память переходит в состояние, показанное на рис. 4.13, *a*, и т. д.

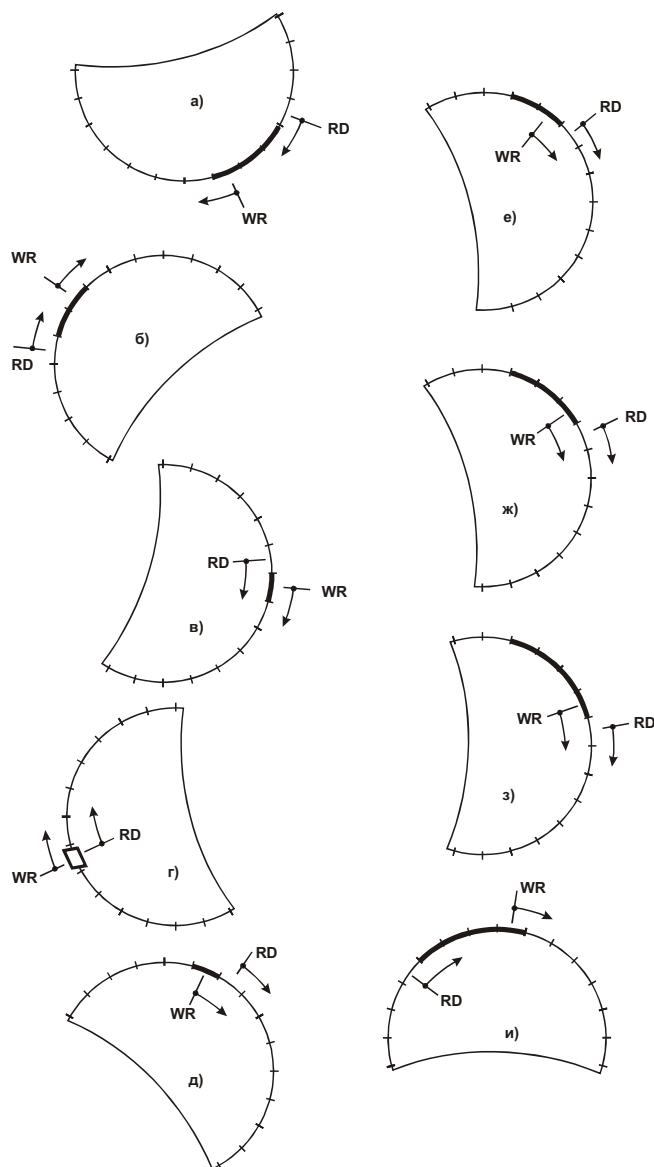


рис. 4.13. Прохождение данных через блок памяти В4 при положительном проскальзывании

Отрицательное проскальзывание обусловлено противоположным соотношением скоростей продвижения указателей адресов, при котором указатель WR постоянно выигрывает гонку. Это соответствует незначительному превышению скорости данных, поступающих в буферную память из линии, над скоростью их считывания из памяти. Пусть в исходном состоянии взаимное расположение указателей соответствует приведенному на рис. 4.14, *a*. Буферная память почти полностью занята новыми битами данных из линии; свободны (уже прочитаны) только три ячейки.

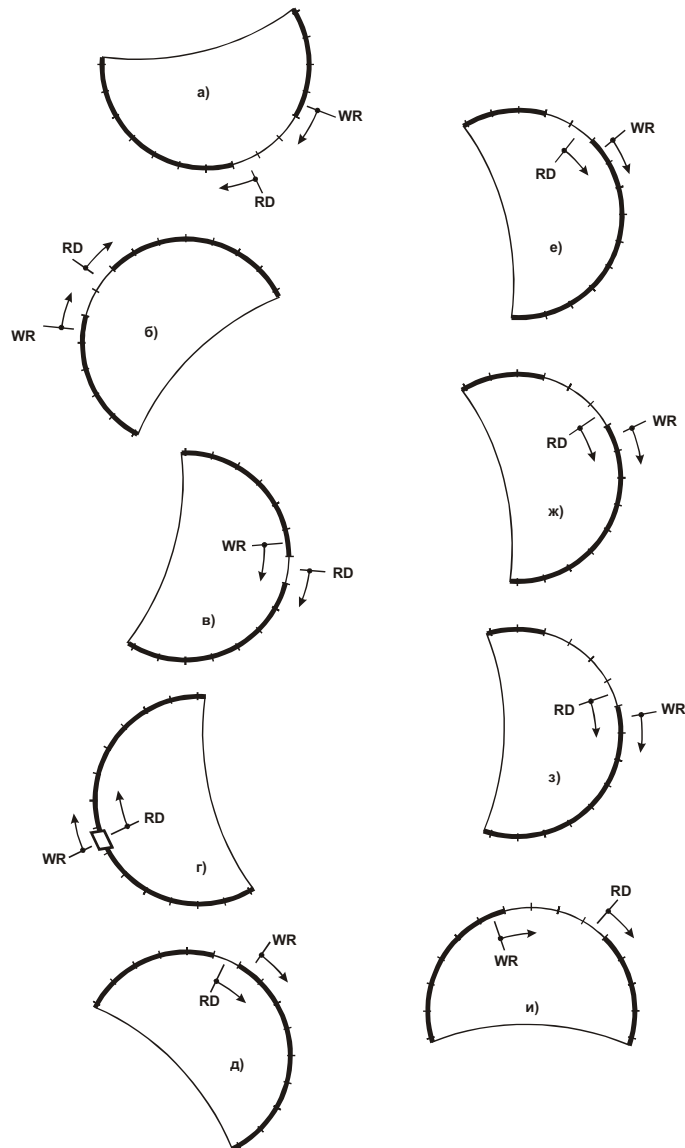


рис. 4.14. Прохождение данных через блок памяти В4 при отрицательном проскальзывании

Через равные, сравнительно большие интервалы времени  $T$  (например  $1/30$  с) дистанция между указателями уменьшается, затем их положения совпадают (рис. 4.14, б – г). Через несколько тактов после этого указатель  $WR$  занимает лидирующее положение, но запись ведется в те ячейки, которые не были прочитаны (см. состояния буферной памяти в четырех тактах после обгона, рис. 4.14, д – з). Вследствие этого теряется группа битов длиной в один кадр. После серии изменений дистанции между указателями они вновь сближаются (рис. 4.14, и). После очередного скачка дистанции буферная память переходит в состояние, показанное на рис. 4.14, а, и т. д. С учетом замечаний, сделанных при описании положительного проскальзывания, для упрощения рассуждений можно полагать, что при отрицательном проскальзывании из последовательности кадров вычеркивается один из них.

Для оценки периода проскальзываний предположим, что частота сигнала  $CLK$  равна  $1\,544\,010$  Гц, т. е. на  $10$  Гц превышает теоретическую. При этом частота синхросигнала внутреннего генератора мультимплексора  $MUX2$  равна  $1\,543\,980$  Гц, т. е. меньше теоретической на  $20$  Гц. Разность частот составляет  $30$  Гц. Это означает, что за одну секунду в буферной памяти В4 совершается  $30$  скачкообразных изменений дис-

танции между указателями WR и RD. Полный цикл обгона завершается после 193 скачков изменения дистанции. Период проскальзываний составляет  $193/30 = 6,4$  с.

Каждый кадр переносит один цифровой отсчет сигнала Y. Полный период этого сигнала представлен восемью отсчетами. Угловое расстояние между отсчетами составляет  $360/8 = 45$  град. Положительное проскальзывание, как было показано, приводит к повторению считывания кадра. Поэтому один из отсчетов повторяется дважды, что приводит к растяжению синусоидального сигнала и положительному скачку его фазы на 45 градусов. Каждое проскальзывание приводит к набегу фазы сигнала, как показано на рис. 4.15.

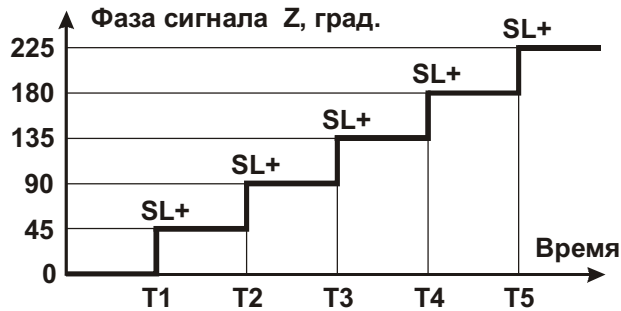


рис. 4.15. Набег фазы тестового сигнала при регистрации положительных проскальзываний SL+

При отрицательном проскальзывании теряется один кадр и, следовательно, последовательность отсчетов сжимается. Из синусоидального сигнала вырезается фрагмент угловой длительностью 45 градусов. Каждое проскальзывание приводит к отрицательному набегу фазы сигнала. Соответствующий график отрицательного набега фазы симметричен приведенному на рис. 4.15 относительно оси времени.

В качестве детектора скачков фазы в простейшем случае может использоваться запоминающий осциллограф. В приведенном ранее примере проскальзывания возникают каждые 6,4 с. Поэтому оператор может снять осциллограмму сигнала за период, равный, например, 10 с, визуально найти в ней неоднородности и сопоставить их характер с временными диаграммами, приведенными на рис. 4.11.

В устройстве DTE1 можно использовать восьмиразрядный аналого-цифровой преобразователь. Стартовый и стоповый биты при этом исключаются. Выдаваемый устройством байт в этом случае содержит только результат преобразования. Так как тестовый сигнал периодический, приемное устройство может распознать старший бит по его периодическому изменению с частотой 1 кГц.

#### Второй вариант

В схеме, приведенной на рис. 4.16, для обнаружения проскальзываний использована аппаратура тестирования, не содержащая аналоговых устройств.

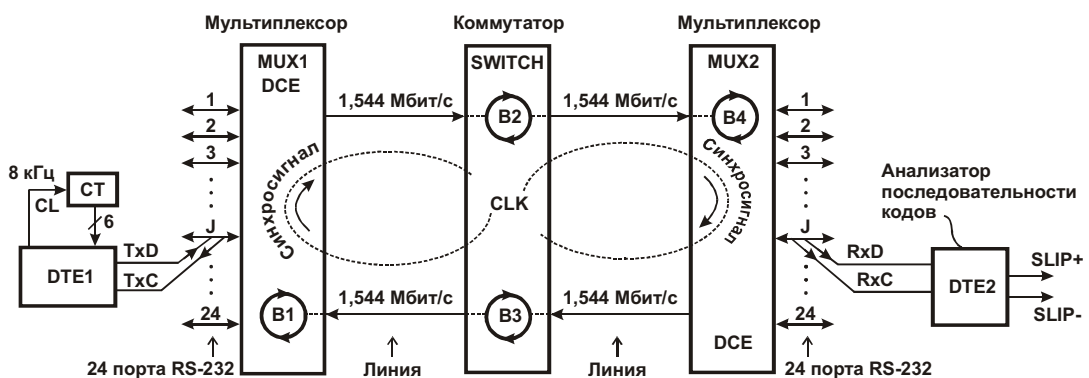


рис. 4.16. Схема обнаружения проскальзываний в многоканальной системе передачи данных – второй вариант

Передающая часть аппаратуры содержит шестизрядный двоичный счетчик СТ и устройство DTE1. Сигнал TxС синхронизации частотой 64 кГц после деления на восемь поступает вход синхронизации счетчика. Код с выхода счетчика обрамляется стартовым и стоповым битами, полученный байт последовательно пересылается в мультиплексор MUX1. Вслед за стоповым битом текущего байта передается стартовый бит следующего и т. д. Таким образом, передающая часть аппаратуры тестирования посылает в канал J системы передачи данных непрерывную последовательность кодов ... 0, 1, 2, ..., 62, 63, 0, 1, 2, и т. д.

Приемная часть аппаратуры тестирования (устройство DTE2) построена на основе микрокомпьютера. Из потока данных RxD выделяются байты, стартовые и стоповые биты отбрасываются, полученная последовательность кодов анализируется. В отсутствие проскальзываний эта последовательность в полной мере соответствует исходной. Возможные искажения отдельных кодов или их групп, не нарушающие общей закономерности формирования последовательности игнорируются как не имеющие отношения к проскальзываниям.

Как было показано, положительное проскальзывание проявляется во внедрении в правильную последовательность дубликата одного из кодов. Например, может быть зарегистрирована такая последовательность: 35, 36, 37, 38, 38, 39, 40, 41, ... или ...12, 13, 14, X, 15, 16, ..., где символ "X" обозначает произвольное значение кода. В этих последовательностях прослеживается приостановка и последующее возобновление счета, связанные с внедрением лишних кодов.

Аналогично регистрируется отрицательное проскальзывание, при котором из правильной последовательности вычеркивается один код.

### Третий вариант

В схеме, приведенной на рис. 4.17 [50], передающая аппаратура тестера представлена девятиразрядным генератором псевдослучайной последовательности битов. Этот генератор выполнен по стандартной схеме на основе сдвигового регистра с элементом Исключающее ИЛИ (XOR) в цепи обратной связи. (Краткое описание таких генераторов приведено в п. 8.4.1.)

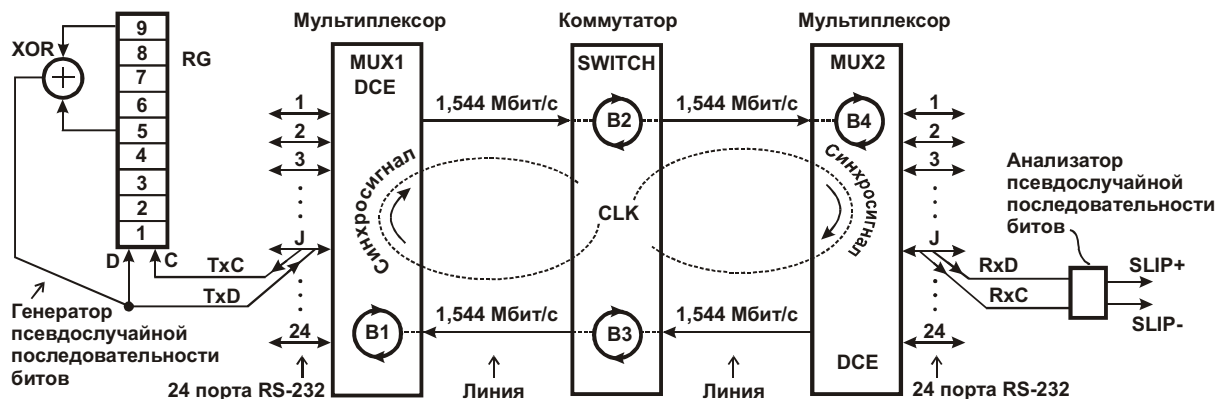


рис. 4.17. Схема обнаружения проскальзываний в многоканальной системе передачи данных – третий вариант

Регистр RG изначально устанавливается в произвольное ненулевое состояние (цепь начальной установки на рисунке не показана). Далее под действием положительных фронтов сигнала TxС частотой 64 кГц содержимое регистра смещается вверх, а в освободившийся (первый) разряд принимается выходной бит TxD, полученный суммированием по модулю два пятого и девятого битов предыдущего кода в регистре. Такой процесс позволяет сформировать псевдослучайную последовательность битов с периодом повторения, равным  $2^9 - 1 = 511$ .

Приемная аппаратура тестера выполняет функции анализатора псевдослучайной последовательности битов RxD. Анализатор следит за правильностью получаемой последовательности сравнением с имеющимся эталоном. Если в последовательность внедрился лишний бит и последовательность “растянулась” или исчез нужный бит и последовательность “сжалась”, то регистрируются соответственно положительное или отрицательное проскальзывания.

Анализатор (рис. 4.18) содержит генератор псевдослучайной последовательности битов на основе сдвигового регистра RG1 с элементом Иключающее ИЛИ (XOR1) в цепи обратной связи. Структура генератора подобна рассмотренной ранее с учетом двух особенностей. Во-первых, цепь обратной связи генератора может размыкаться, когда электронный ключ SW переводится из положения 2 в положение 1. При этом регистр RG1 не участвует в формировании псевдослучайной последовательности битов, а лишь принимает ее в виде потока входных данных RxD. Состояние ключа определяется сигналом W от микрокомпьютера MC. Во-вторых, регистр RG1 содержит два дополнительных разряда: 10 и 11. Эти разряды не участвуют в формировании сигнала обратной связи и, следовательно, не влияют на работу генератора псевдослучайной последовательности битов. Введение дополнительных разрядов позволяет отслеживать предысторию заполнения разрядов 1 – 9 регистра RG1 в двух ближайших предыдущих тактах.

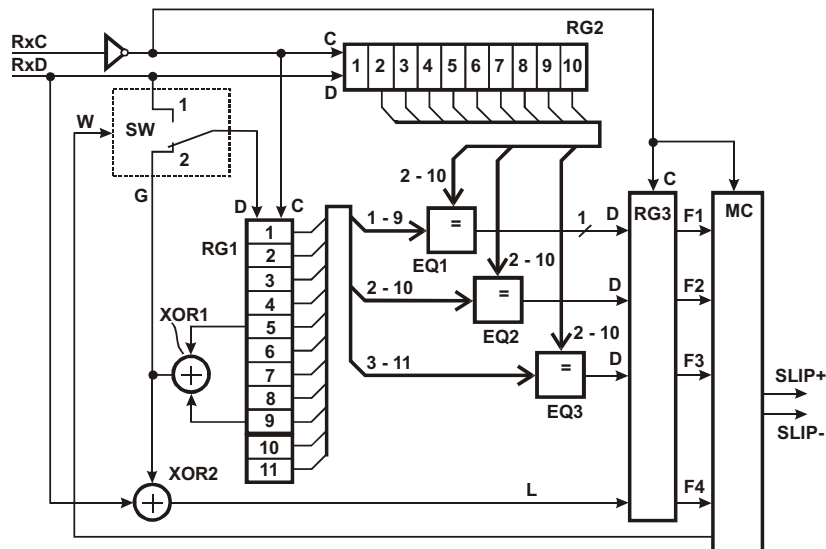


Рис. 4.18. Схема анализатора псевдослучайной последовательности битов

Поток данных RxD проходит также через сдвиговый регистр RG2. Компараторы EQ1 – EQ3 сравнивают группы битов из регистров RG1 и RG2, элемент XOR2 сравнивает предсказанный псевдослучайный бит G с фактически принятым битом RxD. Результаты сравнения фиксируются в четырехразрядном параллельном регистре RG3. Микрокомпьютер следит за динамикой и анализирует сигналы F1 – F4 с выходов этого регистра. В необходимых случаях микрокомпьютер изменяет состояние сигнала W управления ключом SW, а также формирует сигналы SLIP+ или SLIP- обнаружения положительного или отрицательного проскальзываний.

В исходном состоянии ключ SW находится в положении 1. Под управлением инвертированного сигнала RxC входной поток псевдослучайных битов данных RxD одновременно проходит через регистры RG1 и RG2. Поэтому значения битов в одноименных разрядах этих регистров совпадают. Так как структура обратных связей регистра RG1 такая же, как и в генераторе на передающей стороне системы передачи данных, сигналы G и RxD совпадают, с учетом того, что сигнал G опережает сигнал RxD на половину такта. Такое поведение сигнала G можно рассматривать как предсказание очередного ожидаемого псевдослучайного бита RxD. По положительному

фронту сигнала  $RxC$  результат  $L$  сравнения предсказанного и фактически поступившего битов запоминается в регистре  $RG3$ . Если эти биты совпадают, то сигнал  $F4$  равен 0.

Уверенное обнаружение серии совпадений (когда  $F = 0$ , например, в 20 смежных тактах) означает, что код в разрядах 1 – 9 регистра  $RG1$  с некоторой задержкой повторяет код в соответствующем регистре генератора, размещенного на передающей стороне системы передачи данных, т. е. достигнута синхронная работа этих регистров. В этом случае микрокомпьютер переводит ключ  $SW$  в положение 2. Теперь цепь обратной связи замкнута, на вход регистра  $RG1$  поступают предсказанные биты  $G$ . Таким образом, в анализаторе начал функционировать автономный эталонный генератор псевдослучайной последовательности битов. Точнее, в действие вступили три генератора на основе регистра  $RG1$ . Первый генератор формирует номинальный эталонный девятиразрядный код на выходах 2 – 10, второй – задержанный код на выходах 3 – 11, третий – опережающий код на выходах 1 – 9. Эти коды используются для обнаружения и определения направления проскальзывания.

При правильной работе системы передачи данных коды в разрядах 2 – 10 регистров  $RG1$  и  $RG2$  постоянно (в любом такте) совпадают, что регистрируется компаратором  $EQ2$ , сигнал  $F2$  постоянно равен единице. Микрокомпьютер следит за этим сигналом и поддерживает выходные сигналы  $SLIP+$  и  $SLIP-$  в нулевом состоянии.

Положительное проскальзывание сопровождается внедрением во входную последовательность одного лишнего бита. Лишний бит проходит через регистр  $RG2$ , вслед за ним в этот регистр поступает правильная псевдослучайная последовательность битов. Лишний бит теряется после выхода из разряда 10. Из-за внедрения лишнего бита правильная последовательность задержана относительно номинальной эталонной на один такт. Теперь разряды 2 – 10 регистра  $RG2$  постоянно (в каждом такте) совпадают с разрядами 3 – 11 регистра  $RG1$ , т. е. создаются условия для срабатывания компаратора  $EQ3$ . Микрокомпьютер обнаруживает устойчивое единичное состояние сигнала  $F3$  и принимает к сведению появление положительного проскальзывания. Он формирует сигнал  $SLIP+ = 1$  и переводит ключ  $SW$  в положение 1, начиная вхождение в синхронизацию с новой (задержанной на один такт) псевдослучайной последовательностью битов. Далее описанные процессы повторяются.

Отрицательное проскальзывание удаляет один бит из входной последовательности. После выхода границы между старой и новой последовательностями битов за пределы регистра  $RG2$  через этот регистр проходит правильная последовательность, опережающая номинальную эталонную на один такт. Это приводит к устойчивому срабатыванию компаратора  $EQ1$  и формированию сигнала  $F1 = 1$ , который расценивается микрокомпьютером как отрицательное проскальзывание. Микрокомпьютер формирует сигнал  $SLIP- = 1$ , подстраивается к новой последовательности и т. д.

Ошибки, не связанные с проскальзываниями, вызывают искажения сигналов  $F1$  –  $F4$ . Однако их легко отличить от проскальзываний, так как они не могут привести к устойчивому переходу от номинальной к задержанной или опережающей последовательностям битов.

Рассмотренные варианты построения аппаратуры обнаружения проскальзываний имеют много общего и различаются способами генерации и анализа последовательностей тестовых сигналов. Первый вариант оперирует как цифровыми, так и аналоговыми сигналами, что позволяет использовать простые средства регистрации ошибок, например, запоминающий осциллограф. Второй и третий варианты примерно равноценны и ориентированы на использование в автоматизированных системах контроля, так как результаты тестирования можно получить обработкой цифровых сигналов с выходов приемной аппаратуры.

## 4.5. Устранение проскальзываний синхронизации при передаче речевых сигналов

Как уже отмечалось в п. 4.1, проскальзывания синхронизации при передаче музыки, речи или иной аудиоинформации приводят к прослушиванию щелчков. Чтобы исключить проскальзывания, можно воспользоваться, например, введением избыточных битов в поток данных, передаваемых по каналу связи, как было показано в п. 4.2. Такое решение, однако, может оказаться слишком громоздким. Более простые способы исключения проскальзываний (п. 4.5.1, 4.5.2) основаны на том, что при передаче аудиоинформации допустимы искажения, которые преднамеренно вносятся в информационные потоки. Эти искажения настолько невелики, что не воспринимаются слушателем. Но для передачи цифровых данных такой подход неприменим – данные “общего назначения” не могут содержать “незначительные” искажения, так как все биты одинаково важны. Поэтому рассмотренные в п. 4.5 решения применимы только для передачи аудиоинформации.

### 4.5.1. Идея использования периодов “тишины”

Рассмотрим передачу “оцифрованных” речевых сигналов между удаленными друг от друга устройствами А и В, например, радиотелефонами (рис. 4.19). Эти устройства синхронизируются от не синхронизированных между собой внутренних генераторов G1 и G2. Номинальные частоты генераторов одинаковы, но фактически, конечно, они немного различаются. Для компенсации разности частот используется буферная память типа FIFO. Объем памяти ограничен, поэтому, если не принять мер по стабилизации уровня ее заполнения, при передаче непрерывного потока данных могут наблюдаться проскальзывания. Они происходят в моменты переполнения или опустошения буферной памяти. Это приводит к искажению передаваемых данных, и, как следствие, – к прослушиванию щелчков и снижению разборчивости речи.

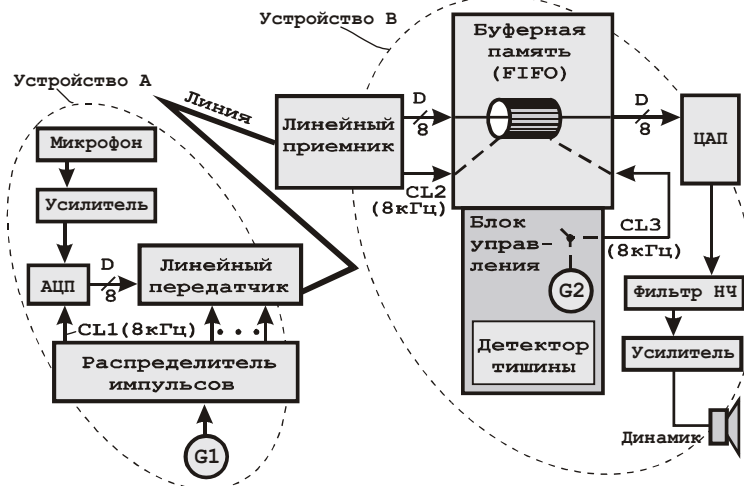


рис. 4.19. Вариант реализации идеи устранения проскальзываний путем варьирования длительностей пауз между словами или фразами

Идея [4] устранения проскальзываний основана на том, что при передаче речи слушатель не может заметить редких (с периодом порядка минуты или более) и незначительных (порядка долей процента) изменений длительности пауз между словами или фразами. Поэтому, наблюдая за текущим содержимым буферной памяти, можно преднамеренно вычеркивать или дописывать в нее коды “тишины”, примыкающие к областям отображения пауз. Это дает возможность регулировать уровень заполнения буфера, т. е. предотвратить его переполнение или опустошение.

Теперь – всё по порядку.

## Прохождение сигнала по тракту “микрофон – динамик”

Как показано на рис. 4.19, сигнал с микрофона усиливается и с помощью аналого-цифрового преобразователя АЦП превращается в равномерную непрерывную последовательность цифровых отсчетов, представленных восьмиразрядным параллельным кодом  $D$ . Темп выдачи отсчетов определяется частотой сигнала  $CL1$  от распределителя импульсов и в данном примере составляет 8 кГц. Источником синхросигналов на входе распределителя импульсов является кварцевый генератор относительно высокой частоты (например порядка сотен килогерц).

Линейный передатчик преобразует входной поток данных (8 кбайт/с = 64 кбит/с) в последовательность кадров. Этот процесс не связан с предлагаемой идеей и не представляет для нас интереса. Отметим только, что в битовый поток данных добавляется служебная информация (флаги начала кадров и проч.), поэтому суммарная скорость передачи смеси речевой и служебной информации по линии превышает полезную скорость, равную, как отмечалось, 64 кбит/с.

Линейный приемник выделяет из получаемого сигнала данные и синхроимпульсы, вычеркивает служебную информацию и преобразует последовательный код в параллельный восьмиразрядный. В результате такого вычеркивания на его выходе формируется неравномерный (пульсирующий) поток полезных байтов, истинность которых подтверждается соответствующим неравномерным сигналом  $CL2$ . Средняя скорость этого потока составляет 8 кбайт/с и в точности равна скорости первоначального потока на выходе АЦП, так как синхросигналы  $CL1$  и  $CL2$  порождаются общим источником – кварцевым генератором  $G1$ .

Поток данных с выхода приемника вводится в буферную память типа FIFO (своего рода трубопровод) с темпом, определяемым сигналом  $CLK2$ . Неравномерностью темпа пренебрегаем, так как она приводит к незначительным периодическим колебаниям уровня заполнения буфера около текущего среднего значения.

Темп считывания данных из буфера немного отличен от темпа записи, так как управляющий сигнал  $CLK3$  чтения порожден кварцевым генератором  $G2$  (путем деления его частоты). Иными словами, ключевая проблема в том, что нет точного баланса между “приходом” и “расходом” данных в буфере из-за незначительного несовпадения частот “одинаковых” кварцевых генераторов  $G1$  и  $G2$ . Но прежде чем рассмотреть решение этой проблемы, завершим описание пути распространения сигналов.

Равномерный поток данных с выхода буферной памяти проходит через цифро-аналоговый преобразователь (ЦАП) и фильтр низких частот (НЧ) на вход усилителя. Фильтр сглаживает ступенчатую форму аналогового сигнала и устраняет его возможные кратковременные выбросы на границах между интервалами квантования. Наконец, сигнал усиливается и воспроизводится динамиком.

## Детектор тишины

Блок управления буферной памятью обладает завидным “интеллектом” и выполнен в виде микропроцессорного устройства. Его основная функция – следить за уровнем заполнения буферной памяти и не допускать ее переполнения и опустошения. Как уже отмечалось, регулировка уровня заполнения проводится добавлением или исключением “микроскопических доз тишины” из текущих “естественных” пауз в звуковом сигнале.

Детектор тишины (см. рис. 4.19) реализован программно. Его работа поясняется рис. 4.20, на котором речевой сигнал для наглядности представлен в аналоговом виде, хотя его обработка детектором ведется в цифровой форме. Алгоритм оценки речевого сигнала таков: детектор срабатывает ( $Y = 1$ ) только в том случае, когда уровень сигнала не превышает некоторого порогового значения  $F$ .



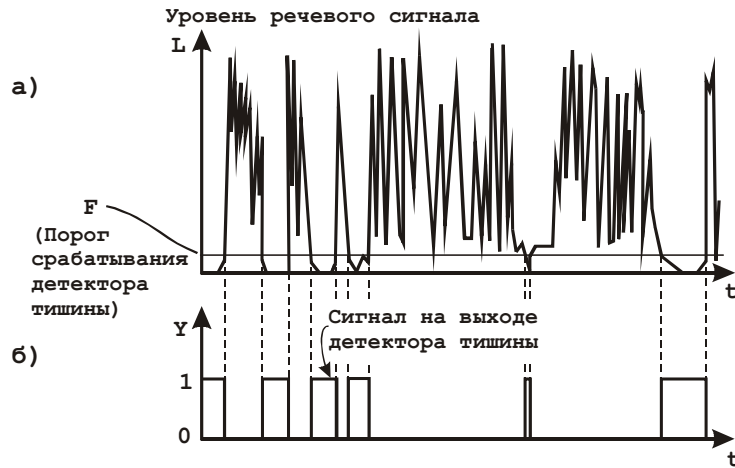


рис. 4.20. Типичный пример временной диаграммы речевого сигнала (а) и результат его оценки на предмет выявления относительной тишины (б)

При записи очередного кода  $D$  в буферную память к нему “прикладывается” дополнительный девятый бит  $Y$  – результат детектирования тишины. Полученный 9-разрядный код под управлением сигнала  $CLK2$  последовательно продвигается по “трубопроводу” и при благоприятном стечении обстоятельств достигает его выхода (есть опасность, что код будет вычеркнут, и “ряды сомкнутся”). Микропроцессор блока управления при необходимости может провести обзор содержимого буферной памяти и отыскать в ней группы “тихий” кодов.

Нормальный уровень заполнения буферной памяти – 50% (рис. 4.21, а). Если скорость считывания данных ниже скорости записи, то уровень заполнения неуклонно растет. Блок управления отслеживает уровень и, обнаружив достаточно большое отклонение от метки “50%”, предпринимает попытку снижения уровня вычеркиванием некоторого числа кодов, помеченных признаком  $Y = 1$ . Если таких кодов в данный момент нет, то микропроцессор дожидается их появления. Процесс периодической коррекции уровня заполнения буферной памяти (в направлении снижения уровня) поясняется рис. 4.21, б.

В противоположной ситуации, когда уровень заполнения буферной памяти неуклонно снижается, блок управления пополняет ее кодами “тишины” (рис. 4.21, в). Желательно, чтобы эти коды отображали некоторый слабый естественный шум вместо абсолютной тишины, чтобы у слушателя не создавалось ощущения “обрыва провода”.

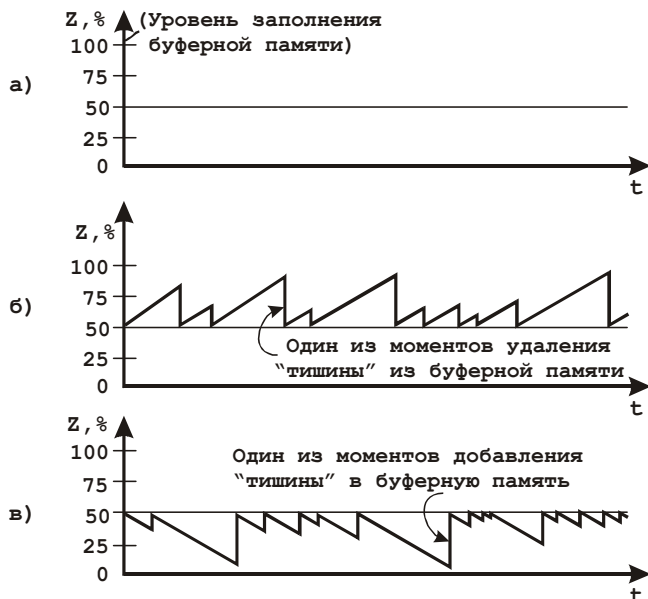


рис. 4.21. Временные диаграммы заполнения буферной памяти: а – идеальная ситуация; б – темп пополнения буфера выше темпа его рассасывания; в – темп пополнения буфера ниже темпа его рассасывания

## Поведение системы в критических ситуациях

Могут возникнуть ситуации, при которых уровень заполнения буферной памяти неуклонно растет или снижается, а ожидаемых пауз нет. Это может произойти, например, при воздействии на микрофон уличного шума. Чтобы справиться с такими ситуациями, предлагается изменять порог срабатывания детектора тишины в зависимости от степени заполнения буферной памяти (рис. 4.22).

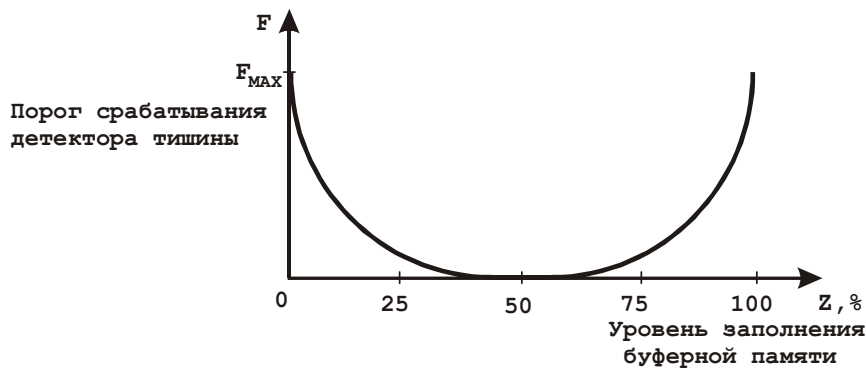


рис. 4.22. Зависимость порога срабатывания детектора тишины от уровня заполнения буферной памяти

Если уровень заполнения буферной памяти незначительно отклоняется от метки “50%”, то порог срабатывания выбирается достаточно низким. С увеличением отклонения порог повышается. Это означает, что блок управления все более настойчиво проявляет стремление исправить ситуацию. При приближении к границам буферной памяти поведение блока управления становится агрессивным – он решает поставленную задачу “во что бы то ни стало”, повышая порог  $F$  до максимально разрешенного уровня  $F_{MAX}$ .

Таким образом, проскальзывания устраняются по мере возможности незначительного расширения или сужения интервалов “молчания”.

### 4.5.2. Идея устранения проскальзываний с помощью ЦАП и АЦП

Принцип действия ранее рассмотренных в этой главе устройств для устранения проскальзываний основан на стабилизации уровня заполнения буферной памяти. Может сложиться впечатление, что буферная память типа FIFO – неотъемлемая часть таких устройств. Однако это не так – можно работать и без использования памяти типа FIFO! Рассмотрим одно из таких решений (с другими можно ознакомиться в п. 4.6).

В системе передачи данных, показанной на рис. 4.23, “оцифрованный” аналоговый сигнал передается от источника по линии 1, проходит через промежуточное устройство (например коммутатор) и по линии 2 поступает потребителю. Составные части этой системы синхронизируются от двух кварцевых генераторов одинаковой номинальной частоты. Первый формирует синхросигнал CLK1, второй – CLK2. Эти частоты различаются, например, на доли герца.

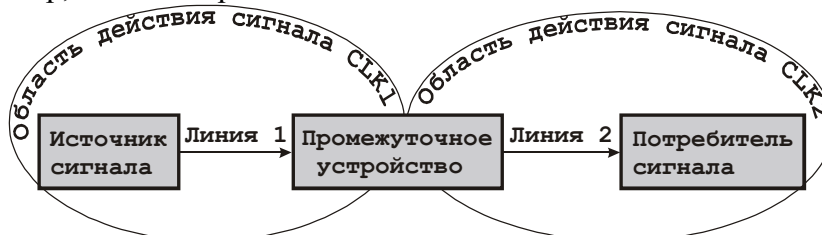


рис. 4.23. Система передачи данных

Чтобы предотвратить искажение аудиоданных в результате проскальзываний синхронизации, можно применить схему промежуточного устройства, показанную на рис. 4.24 [43].

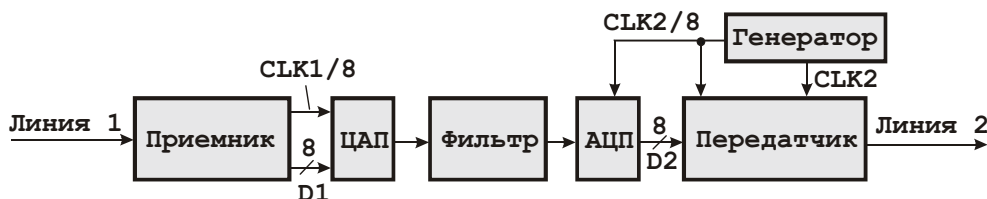


рис. 4.24. Схема промежуточного устройства

Кодированный цифровой сигнал из линии 1 декодируется приемником и преобразуется в равномерный поток байтов  $D1$ . Истинность байта подтверждается фронтом синхросигнала  $CLK1/8$ . Далее сигнал преобразуется в аналоговую форму и фильтруется для устранения следов дискретизации. После этого под управлением второго генератора сигнал вновь преобразуется в цифровой поток байтов  $D2$  и после кодирования поступает в линию 2. Как видим, проскальзываний нет!

Напомним, что рассмотренный способ устранения проскальзываний применим только к передаче “оцифрованной” аналоговой информации. Действительно, цифровые сигналы на входе и выходе цепи “ЦАП – фильтр – АЦП” совпадают лишь с точностью до ошибки, присущей этой цепи, поэтому рассчитывать на точное совпадение этих сигналов не приходится.

## 4.6. Исключение проскальзываний при сопряжении разноскоростных компонентов синхронных систем без использования буфера типа FIFO

Предыдущее решение (п. 4.5.2) подтверждает возможность предотвращения проскальзываний без использования буфера типа FIFO. Но оно применимо только к передаче аудиоданных, причем каждое промежуточное устройство вносит в сигнал искажения, хотя и незначительные. При прохождении сигнала по цепи промежуточных устройств искажения накапливаются и могут превысить допустимый уровень.

Далее на двух примерах (п. 4.6.1, 4.6.2) показана возможность передачи “обычных” цифровых данных между разноскоростными компонентами синхронных систем без использования буфера типа FIFO.

### 4.6.1. Исключение проскальзываний при сопряжении разноскоростных компонентов одноканальной системы

В системе передачи данных, показанной на рис. 4.25 [66], входной и выходной потоки данных (DATA 1 и DATA 2) имеют скорость  $X$  бит/с, в то время как скорость передачи данных по линии составляет  $Y$  бит/с, причем  $Y > X$ .

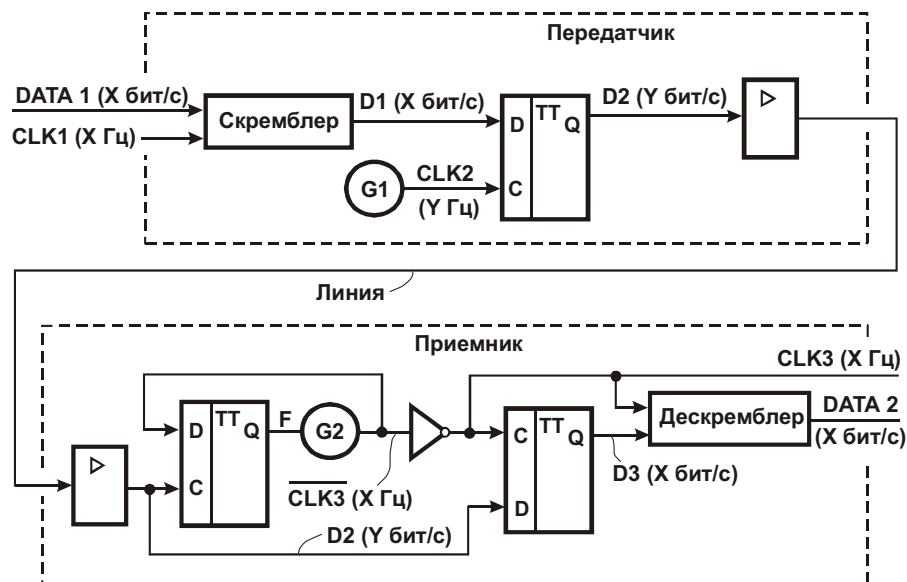


рис. 4.25. Синхронная система передачи данных. Скорости входного и выходного потоков данных (DATA 1 и DATA 2) не совпадают со скоростью потока (D2), передаваемого по линии

Входные данные DATA 1 преобразуются скремблером в псевдослучайную последовательность битов D1 (рис. 4.26). Эта последовательность синхронизирована сигналом CLK1 с частотой X Гц. Положительные фронты синхросигнала CLK1 задают границы битовых интервалов. Кварцевый генератор G1 формирует непрерывную последовательность импульсов CLK2 частотой Y Гц. Сигналы CLK1 и CLK2 не синхронизированы между собой, поэтому временные соотношения между фронтами соответствующих импульсов могут быть произвольными и постоянно изменяются. По положительным фронтам импульсов CLK2 данные D1 записываются в D-триггер передатчика. В результате на выходе триггера формируется поток данных D2, имеющий скорость Y бит/с.

Сравнивая временные диаграммы сигналов D2 и D1, можно заметить, что они схожи, но фронты сигналов не совпадают. На временной диаграмме сигнала D1 затемненными прямоугольниками отмечены периоды, в течение которых соответствующие фронты этого сигнала могут быть восприняты D-триггером в зависимости от возможного взаимного сдвига близлежащих импульсов CLK1 и CLK2. Ширина затемненных прямоугольников равна периоду сигнала CLK2. В лучшем случае фронты могут почти совпадать, в худшем – фронт сигнала D2 может запаздывать относительно соответствующего фронта сигнала D1 на время, практически равное периоду сигнала CLK2.

Эти же затемненные прямоугольники “спроецированы” на временную диаграмму сигнала D2. Здесь они отображают джиттер (дрожание фронтов) этого сигнала (см. гл. 5). Иными словами, сигнал D2 можно рассматривать как сигнал D1, который подвержен джиттеру, обозначенному затемненными прямоугольниками.

Сигнал D2 через выходной усилитель передатчика выдается в линию, затем проходит через входной усилитель приемника и поступает на вход C синхронизации первого триггера, а также на вход D данных второго триггера. Оба триггера срабатывают по положительному фронту сигнала на входе C. Замкнутые в кольцо первый триггер и генератор G2 образуют “петлю фазовой автоподстройки частоты” (или, что то же самое, “генератор с фазовой автоподстройкой частоты”). Выходной сигнал F первого триггера воздействует на генератор G2. Номинальная частота сигнала на выходе этого генератора равна частоте X сигнала CLK1 на входе передатчика. При наличии постоянного сигнала F = 0 частота плавно и очень незначительно повышается, а при F

$= 1$  – снижается. Пределы изменения частоты очень невелики. Генератор G2 обладает достаточной инерционностью по отношению к сигналу F. Это означает, что он реагирует на усредненное по времени значение этого сигнала: преимущественное пребывание первого триггера в состоянии  $F = 0$  вызывает повышение частоты, и наоборот.

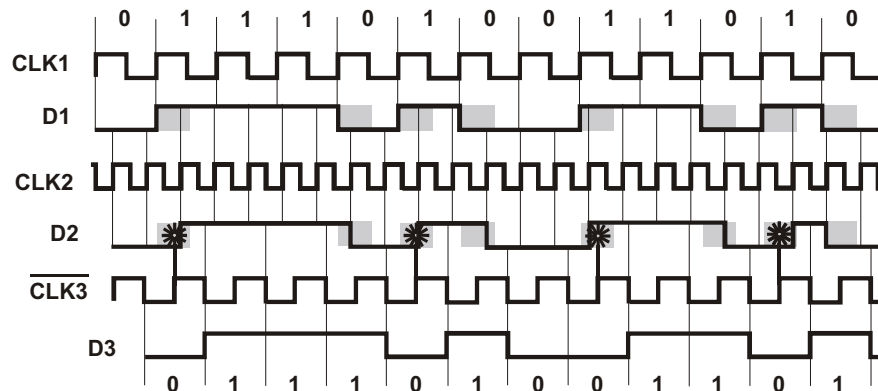


рис. 4.26. Временные диаграммы передачи данных в системе, показанной на рис. 4.25

Покажем, что, благодаря обратной связи с выхода генератора G2 на вход первого триггера, положительные фронты сигнала на выходе генератора привязаны к центрам затемненных прямоугольников на временной диаграмме сигнала D2, что отмечено “звездочками” на рисунке. Для этого рассмотрим две ситуации, которые поочередно возникают при нормальной работе системы.

1. Предположим, что положительные фронты сигнала D2, равномерно распределенные в обозначенных на рисунке зонах джиттера, в большинстве своем совпадают с нулевым состоянием синхросигнала NOT(CLK3). Иными словами, проекции “звездочек” (усредненных положений фронтов) на диаграмму синхросигнала будут чуть опережать его положительные фронты. Это, в свою очередь, означает, что следует слегка увеличить частоту синхросигнала, что приведет к его незначительному фазовому смещению влево. Но как раз это и достигается благодаря тому, что в данной ситуации сигнал управления F будет преимущественно нулевым.

2. В противоположной ситуации временная диаграмма синхросигнала NOT(CLK3) исходно чуть смещена влево относительно показанной на рисунке. Тогда проекции “звездочек” будут попадать на единичные состояния синхросигнала. Это означает, что синхросигнал вырабатывается с опережением, и его следует задержать. Средством задержки служит незначительное снижение его частоты. Оно достигается благодаря тому, что в данной ситуации управляющий сигнал F преимущественно равен единице.

В результате постоянных незначительных колебаний около равновесного состояния осуществляется показанная на рисунке точная привязка синхросигнала NOT(CLK3) к сигналу D2 с учетом искусственно внесенного в него джиттера. Такая привязка обеспечивает стабильность данных на D-входе второго триггера в момент формирования положительного фронта сигнала CLK3.

На выходе второго D-триггера формируется синхронный поток данных D3, повторяющий поток D1. Как следует из временных диаграмм, задержка между одноименными битами этих потоков не превышает одного периода сигнала CLK1 (CLK3). Далее данные D3 дескремблируются, и на выход приемника выдается синхронный поток данных DATA 2, эквивалентный потоку DATA 1.

С уменьшением соотношения Y/X сужается “окно детектирования” сигнала вторым триггером приемника. Окно детектирования представляет собой интервал времени, в течение которого на входе D данных второго триггера присутствует достоверная информация. Положительный фронт сигнала на входе C синхронизации этого триггера попадает в середину окна, но сокращаются периоды предустановки и удержания

(для конкретного типа триггера существуют минимально допустимые значения этих параметров). Уменьшение окна детектирования снижает также допустимый уровень “обычного” джиттера в системе. При соотношении частот  $Z = Y/X = 1,25$  окно детектирования составляет 10% максимального. Приемлемым на практике может считаться значение  $Z$ , превышающее 1,5. В [66] предложены модификации рассмотренного решения, повышающие надежность передачи данных.

Рассмотренное решение позволяет создавать, на первый взгляд, “невозможные” каналы передачи данных; один из таких каналов показан на рис. рис. 4.27. Входные данные проходят по цепи “передатчик – ретранслятор 1 – ретранслятор 2 – приемник”. Передатчики и приемники выполнены по схемам, приведенным на рис. 4.25, но скремблер и дескремблер подключены только на входе и выходе канала.

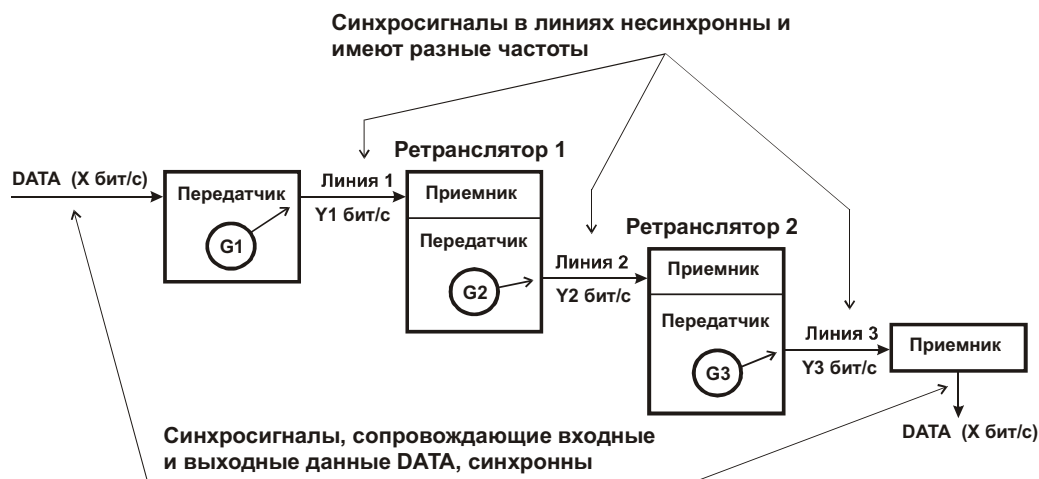


рис. 4.27. Пример сопряжения разноскоростных компонентов канала связи без использования буферов типа FIFO

Необычность этого решения состоит в том, что выходной синхросигнал привязан к входному несмотря на то что синхросигналы в трех линиях связи не синхронны и задаются не связанными между собой генераторами  $G1 - G3$ .

#### 4.6.2. Исключение проскальзываний при сопряжении разноскоростных компонентов системы с мультиплексированием каналов

Рассмотренное далее решение (рис. 4.28) по смыслу близко предыдущему (см. рис. 4.25), но обеспечивает передачу данных по нескольким каналам с их мультиплексированием.

Данные  $D1 - D5$  представлены кодом NRZ (см. п. 8.1) и предварительно скрембированы. Скорости передачи данных в каждом канале индивидуальны. Они задаются пятью генераторами, не синхронизированными между собой (генераторы на рисунке не показаны). Синхросигналы от этих генераторов на мультиплексор не поступают.

Под управлением внутреннего генератора  $G$  мультиплексора (этот генератор формирует сигнал относительно высокой частоты и также не синхронизирован с каким-либо другим) входные данные периодически опрашиваются и komponуются в кадры, которые передаются по линии и преобразуются демультимплексором в выходные данные  $DX1 - DX5$ . Скорости передачи данных  $DX1 - DX5$  совпадают с соответствующими скоростями передачи данных  $D1 - D5$ . Как и в предыдущем решении, основная идея состоит в том, что переход от одной частоты синхронизации к другой не связан с использованием буферной памяти типа FIFO. Перейдем к подробностям данного решения.

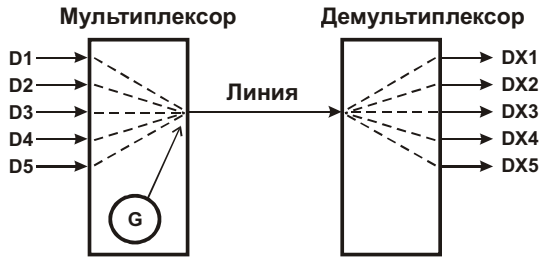


рис. 4.28. Функциональная схема системы с мультиплексированием каналов

Мультиплексор, показанный на рис. 4.29 [67], позволяет объединить входные потоки данных в общий выходной поток. Мультиплексор содержит “собственно мультиплексор” MUX, четырехразрядный двоичный счетчик CT2 и логические элементы. Напомним, что входные синхронные потоки данных D1 – D5 могут иметь разные скорости. Частота сигнала CLK от внутреннего генератора G определяет скорость выходного потока данных DATA. Эта частота должна быть достаточно высокой. Как и в предыдущем примере, каждый входной сигнал должен опрашиваться с частотой Y, не меньшей 1,5 X, где X – скорость соответствующего входного потока данных. Это необходимо для уверенного восстановления потоков демultipлексором.

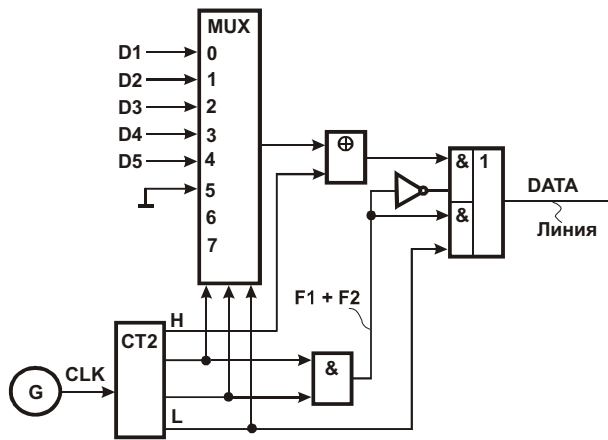


рис. 4.29. Схема мультиплексора

Передаваемые данные группируются в кадры (рис. 4.30). Три младших разряда кода с выхода счетчика управляют мультиплексором MUX. В момент начала передачи кадра J в счетчике сформирован нулевой код. В шести начальных тактах кадра (с нулевого по пятый) на выход мультиплексора MUX последовательно передаются текущие состояния сигналов D1 – D5, а также нулевой сигнал с входа 5 (T0 = 0).

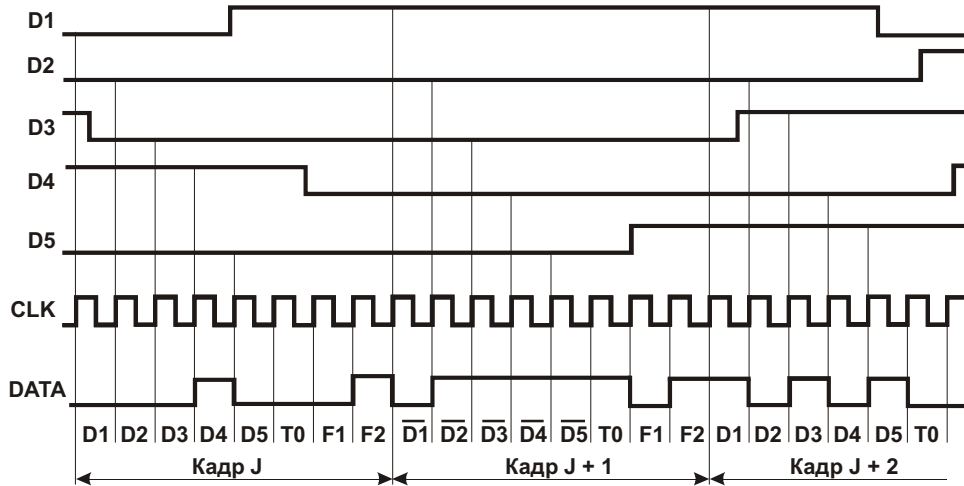


рис. 4.30. Временные диаграммы мультиплексирования сигналов

Так как в течение первых восьми тактов на выходе старшего разряда H счетчика присутствует лог. 0, логический элемент Иключающее ИЛИ транслирует сигналы с выхода мультиплексора MUX без инвертирования. Логический элемент 2И – ИЛИ последовательно передает эти сигналы на выход мультиплексора. В шестом и седьмом тактах срабатывает элемент И, элемент 2И – ИЛИ транслирует на выход сигнал L с

выхода младшего разряда счетчика. В шестом такте этот сигнал равен нулю (F1), в седьмом – единице (F2).

В восьмом – пятнадцатом тактах процессы повторяются, но при этом в старшем разряде N счетчика присутствует сигнал лог. 1. Это приводит к тому, что передаваемые с входов мультиплексора MUX сигналы инвертируются логическим элементом Иключающее ИЛИ. В частности, нулевой сигнал с входа 5 преобразуется в выходной сигнал  $T0 = 1$ . Биты F1 и F2 формируются так же, как и ранее.

Таким образом, последовательность кадров характеризуется следующими признаками.

1. Каждый кадр несет информацию о состоянии группы входных сигналов на момент опроса каждого из них. Эта информация при  $T0 = 0$  представлена в прямом коде, а при  $T0 = 1$  – в обратном. Такой способ кодирования выравнивает спектр выходного сигнала DATA при его выдаче в линию, а также исключает из спектра постоянную составляющую. Приемник при  $T0 = 1$  инвертирует принятые данные.

2. Каждый кадр содержит противофазные флаговые биты F1 и F2. Гарантированное изменение уровня сигнала на границе между этими битами позволяет поддерживать надежную синхронизацию приемника с передатчиком. Флаговые биты неизменны во всех кадрах, поэтому они служат метками, по которым приемник распознаёт взаимное расположение кадров.

Восстановление исходной группы сигналов DX1 – DX5 (эквивалентной группе D1 – D5) осуществляется приемником с помощью демультимплексора, рис. 4.31.

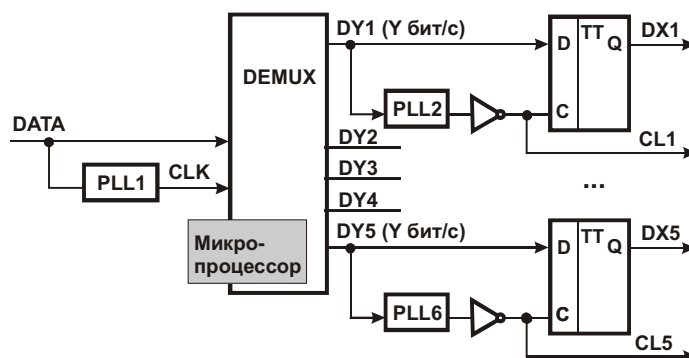


рис. 4.31. Схема демультимплексора

На выходе генератора PLL1 с фазовой автоподстройкой частоты формируется синхросигнал CLK (см. две нижние временные диаграммы на рис. 4.30). Пример схемы такого генератора был рассмотрен ранее (рис. 4.25); другие примеры приведены в гл. 9. С помощью микропроцессора устанавливается статистически достоверное местоположение флаговых битов F1 и F2. Зная размещение межкадровых границ, “собственно демультимплексор” DEMUX распределяет входной поток данных по пяти каналам. При этом учитывается, что данные попеременно передаются в прямом и обратном кодах. Как и в предыдущем решении (см. п. 4.6.1), в каждом канале создается сигнал “DY”, отличающийся от истинного наличием искусственно привнесенного джиттера.

Далее, как и в предыдущем решении, приемник восстанавливает исходные синхросигналы и данные. Для детектирования пяти независимых синхросигналов CL1 – CL5 в каждом канале используется соответствующий генератор PLL2 – PLL6 с фазовой автоподстройкой частоты. Каждый генератор настроен на ожидаемую частоту синхросигнала в соответствующем канале.

Приведенные решения подтверждают возможность предотвращения проскальзываний без использования буферной памяти типа FIFO. Напомним, что эти решения не универсальны и применимы только при выполнении определенных ограничений на соотношение частот синхронизации сопрягаемых компонентов системы передачи данных.