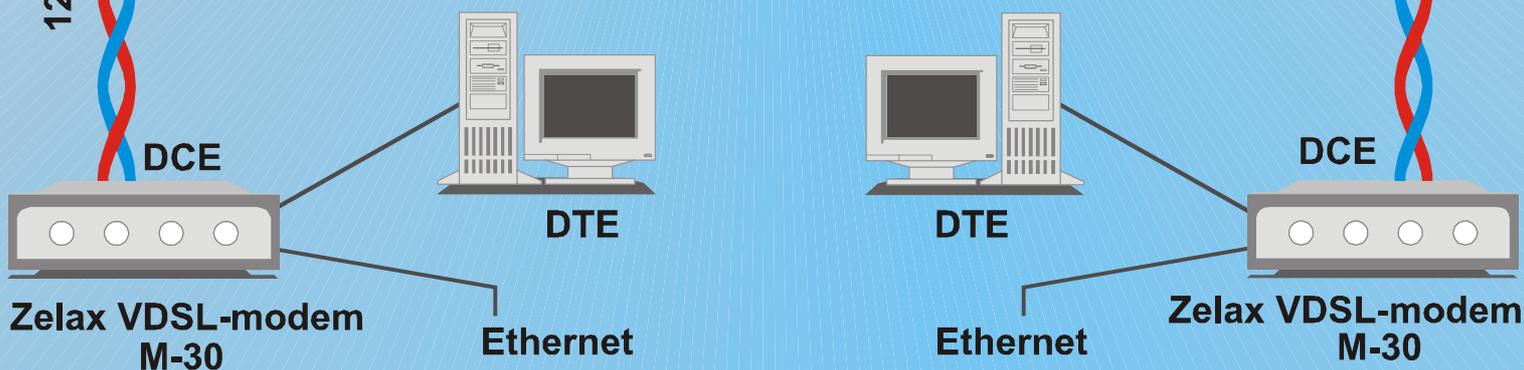


С.М.Сухман, А.В.Бернов, Б.В.Шевкопляс

СИНХРОНИЗАЦИЯ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

АНАЛИЗ ИНЖЕНЕРНЫХ РЕШЕНИЙ

12 Мбит/с, 0,8 км



УДК

Авторы:

Сухман Сергей Маратович

Бернов Аркадий Валентинович

Шевкопляс Борис Владимирович

Рассмотрено более ста структурных и схемных решений, обеспечивающих согласованную работу телекоммуникационных устройств и их составных частей. Описано взаимодействие устройств типа DTE/DCE, приведены способы предотвращения проскальзываний синхронизации и устранения фазовых помех, рассмотрены вопросы построения систем синхронизации в сетях передачи данных, методы и схемы распознавания информационных кадров и ячеек в потоках данных, способы кодирования данных для их передачи по каналу связи, схемы выделения синхросигнала и данных из канала связи и др. Большинство рассмотренных решений представляют собой изобретения, т. е. защищены патентами.

Может быть полезна студентам, аспирантам и техническим специалистам, желающим расширить базовые знания в части освоения практических задач сопряжения телекоммуникационных устройств.

Оглавление

ПРЕДИСЛОВИЕ	5
1. ВЗАИМОДЕЙСТВИЕ МЕЖДУ УСТРОЙСТВАМИ ПО ИНТЕРФЕЙСУ RS-232	7
1.1. Устройства типа DTE и DCE: сложности терминологии	7
1.2. Основные сигналы и режимы работы интерфейса RS-232	10
1.3. Асинхронный обмен данными между одноименными устройствами типа DTE/DCE	17
2. ВЗАИМОДЕЙСТВИЕ УСТРОЙСТВ ТИПА DTE/DCE В СИНХРОННОМ РЕЖИМЕ: ТИПОВЫЕ РЕШЕНИЯ	21
2.1. Системы с внутренней синхронизацией	21
2.2. Системы с внешней синхронизацией	23
2.3. Использование модема как устройства типа DTE	24
2.4. Система с двумя последовательно включенными каналами связи	25
3. ВЗАИМОДЕЙСТВИЕ УСТРОЙСТВ ТИПА DTE/DCE В СИНХРОННОМ РЕЖИМЕ: НЕСТАНДАРТНЫЕ РЕШЕНИЯ	26
3.1. Синхронная передача данных между устройствами типа DTE и DCE с использованием стартовых и стоповых битов	26
3.2. Синхронный обмен данными с передачей кадровых меток	28
3.3. Повышение быстродействия и расширение функциональных возможностей схемы с попутной синхронизацией	30
3.3.1. Удвоение скорости передачи данных с использованием положительного и отрицательного фронтов сигнала CLK для их приема 31	
3.3.2. Удвоение скорости передачи данных заменой сигнала CLK сигналом разграничения одноименных битов	33
3.3.3. Расширение функциональных возможностей системы с разграничением одноименных битов	34
3.4. Аппаратное управление потоком данных с использованием пачек сигналов TxS	47
3.4.1. Согласование средней скорости передачи данных между устройствами DTE и DCE со скоростью передачи данных по линии 47	
3.4.2. Упрощение системы синхронизации формирователя HDSL-кадров	49
3.5. Взаимодействие удаленных устройств с использованием дистанционной цифровой коррекции фазы сигнала от синхрогенератора	52
3.5.1. Выравнивание синхросеток передаваемых и принимаемых данных	52
3.5.2. Передача синхросигнала “против течения” потока данных	54
3.6. Взаимодействие удаленных устройств с непосредственной односторонней передачей синхросигнала по каналу связи	56
3.7. Поиск начала асинхронного сообщения	58
3.8. Обнаружение и исправление ошибок синхронизации при передаче непрерывного асинхронного потока данных	59
3.9. Распознавание межбайтовых границ в непрерывном синхронном потоке данных	61
3.9.1. Схема с дублированием и инвертированием одного бита данных	62
3.9.2. Схема с дублированием и инвертированием байта данных	64
4. ПРОСКАЛЬЗЫВАНИЯ СИНХРОНИЗАЦИИ	66
4.1. Причины и следствия проскальзываний	66
4.2. Каким образом избежать проскальзываний	68
4.3. Частичное восстановление кадра, искаженного в результате битового проскальзывания	73
4.3.1. Механизм битовых проскальзываний	73
4.3.2. Как по возможности сохранить кадр	74
4.4. Обнаружение проскальзываний в многоканальной системе передачи данных	77
4.5. Устранение проскальзываний синхронизации при передаче речевых сигналов	88
4.5.1. Идея использования периодов “тишины”	88
4.5.2. Идея устранения проскальзываний с помощью ЦАП и АЦП	91
4.6. Исключение проскальзываний при сопряжении разноскоростных компонентов синхронных систем без использования буфера типа FIFO	92
4.6.1. Исключение проскальзываний при сопряжении разноскоростных компонентов одноканальной системы	92
4.6.2. Исключение проскальзываний при сопряжении разноскоростных компонентов системы с мультиплексированием каналов 95	
5. ФАЗОВЫЕ ПОМЕХИ	98
5.1. Основные виды фазовых помех	98
5.2. Идея построения адаптивного фильтра для подавления фазовых помех	98
5.3. Вариант схемной реализации аттенюатора джиттера	99
6. МЕТОДЫ СИНХРОНИЗАЦИИ УЗЛОВ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ	102
6.1. Обновление синхросигнала на пути его распространения по сети	103
6.2. Предотвращение заикливания синхросигналов	106
6.3. Синхронизация кольцевых структур	111
6.4. Использование искусственного вандера для прослеживания трасс синхросигналов в сетях со сложной топологией	118
7. СИНХРОНИЗАЦИЯ НА УРОВНЕ РАСПОЗНАВАНИЯ И ОБРАБОТКИ КАДРОВ И ЯЧЕЕК	121
7.1. Распознавание многоразрядного флага в битовом и байтовом потоках данных	122
7.1.1. Распознавание флага в битовых потоках данных	122
7.1.2. Распознавание флага в байтовых потоках данных	133
7.2. Минимизация длины флага начала кадра	136
7.2.1. Использование чередующихся нулевых и единичных битов в качестве флагов начала кадров	137
7.2.2. Использование псевдослучайных битов в качестве флагов начала кадров	140
7.3. Использование раздробленного флага начала кадра	144
7.3.1. Применение неуникального флагового кода	145
7.3.2. Построение кросс-корреляционной матрицы для распознавания раздробленного флага	145
7.4. Распознавание ячеек ATM в битовом и байтовом потоках данных	148
7.4.1. Структура ячейки ATM	148
7.4.2. Использование кода CRC в процессе распознавания границ ячеек	149
7.4.3. Схема “деления” двоичных чисел	150
7.4.4. Формирование заголовка ячейки передатчиком	150
7.4.5. Проверка правильности заголовка ячейки приемником	151
7.4.6. Поиск заголовка в непрерывном битовом потоке данных	154
7.4.7. Поиск заголовка в непрерывном байтовом потоке данных	156
7.5. Распознавание и восстановление искаженных кадров при передаче данных по радиоканалу	160
7.6. Распознавание флагового кода, содержащего ошибки	163

7.7. Поиск флага в потоке данных, передаваемых по волоконно-оптической линии связи	165
7.8. Передача данных вместо избыточных битов синхронизации кадра	167
7.9. Способы размещения низкоскоростного потока данных в высокоскоростном потоке кадров.....	170
7.10. Уменьшение числа операций, выполняемых при распознавании флага начала кадра	174
8. СПОСОБЫ КОДИРОВАНИЯ ДАННЫХ ДЛЯ ИХ ПЕРЕДАЧИ ПО КАНАЛУ СВЯЗИ.....	179
8.1. Основные способы кодирования цифровой информации для ее передачи по последовательным каналам связи.....	179
8.1.1. Структура последовательного канала связи.....	179
8.1.2. Униполярный код NRZ.....	180
8.1.3. Биполярный код NRZ.....	181
8.1.4. Код Манчестер-II.....	181
8.1.5. Код AMI.....	181
8.1.6. Коды BNZS, HDB3	182
8.2. Трехуровневое кодирование сигнала с гарантированным изменением уровней между соседними битовыми интервалами	182
8.3. Способы кодирования сигнала для уменьшения излучаемых помех при его передаче по витой паре проводов	184
8.3.1. Скремблирование полярностей импульсов	184
8.3.2. Двубинарное кодирование	187
8.4. Передача данных с использованием скремблера и дескремблера.....	191
8.4.1. Генераторы псевдослучайных битовых последовательностей	191
8.4.2. Скремблер и дескремблер с неизолрованными генераторами псевдослучайных битовых последовательностей.....	192
8.4.3. Скремблер и дескремблер с изолированными генераторами псевдослучайных битовых последовательностей.....	193
8.4.4. Скремблер и дескремблер с неизолрованными генераторами – улучшенный вариант.....	194
8.4.5. Синхронизация изолированных генераторов скремблера и дескремблера.....	196
8.4.6. Применение системы скремблер – дескремблер для тестирования канала связи	198
8.5. Кодирование данных и синхросигнала для параллельной передачи по группе витых пар проводов	202
8.5.1. Схема кодирования – декодирования	202
8.5.2. Усовершенствованная схема – первый вариант	204
8.5.3. Усовершенствованная схема – второй вариант	211
8.5.4. Усовершенствованная схема – третий вариант	214
8.5.5. Усовершенствованная схема – четвертый вариант	220
9. ВЫДЕЛЕНИЕ СИНХРОСИГНАЛА И ДАННЫХ ИЗ ПРИНИМАЕМОГО СИГНАЛА	226
9.1. Выделение синхросигнала и данных схемами на основе генераторов с фазовой автоподстройкой частоты.....	226
9.2. Выделение синхросигнала и данных схемой на основе сдвиговых регистров	231
9.3. Одноконтурная и двухконтурные схемы выделения синхросигнала.....	234
9.4. Схема выделения синхросигнала с компенсацией нагрузочной емкости.....	237
9.5. Распознавание синхросигнала в искаженном входном сигнале с помощью симметрирующего коррелятора	239
9.6. Шифратор и дешифраторы кода Манчестер-II	243
10. СИНХРОНИЗАЦИЯ ПЕРЕДАТЧИКА И ПРИЕМНИКА ТЕСТОВЫХ СИГНАЛОВ ПРИ ИЗМЕРЕНИИ ДЛИНЫ ЛИНИЙ ПЕРЕДАЧИ ДАННЫХ	250
10.1. Измеритель длины петли передачи данных.....	251
10.2. Способ синхронизации измерителя длины кабельной линии передачи данных	255
ЛИТЕРАТУРА	258

Многие вещи нам непонятны не потому, что наши понятия слабы; но потому, что сии вещи не входят в круг наших понятий.

К. Прутков

Предисловие

Темпы развития телекоммуникаций в последние годы столь велики, что их “описательная” сторона постоянно отстает, если не считать рекламных изданий. Сравнительно полные сведения можно получить по информационным технологиям, но зачастую и они имеют слишком общий характер и допускают неоднозначную трактовку, а стоит только коснуться конкретных деталей – и мы попадаем в информационный вакуум. Большинство частных вопросов остаются без ответа. В этой книге собрана информация, позволяющая в какой-то степени, на примерах, отразить наиболее значимые вопросы обеспечения согласованной (синхронной) работы телекоммуникационных устройств и их составных частей.

Конечно, нельзя объять необъятное, но почему бы в очередной раз не попробовать? Ведь дорогу осилит идущий, и после ознакомления с сотней приведенных здесь решений, кому-то из читателей будет легче воспринимать новые идеи или генерировать свои. Как сказал Исаак Ньютон, “при изучении наук примеры полезнее правил”.

Несколько слов о структуре книги. В первой и второй главах приведены основные сведения о популярных интерфейсах физического уровня из семейства RS. Рассмотрены примеры простейших систем с такими интерфейсами, причем центральное место отведено вопросам синхронизации между взаимодействующими устройствами. В третьей главе рассмотрены нестандартные решения, которые позволяют улучшить характеристики традиционных систем передачи данных. Обратите внимание на то, что можно усовершенствовать даже проверенные десятилетиями элементарные схемы передачи данных (см., например, п. 3.3)! В четвертой и пятой главах рассмотрены методы борьбы с проскальзываниями синхронизации и фазовыми помехами, в шестой – примеры построения систем синхронизации в сетях передачи данных.

Особое внимание уделено методам и схемам распознавания информационных кадров и ячеек в потоках данных (гл. 7). Обычно начало кадра помечается уникальным флажковым кодом, но оказывается, что разрядность этого кода можно сократить до одного бита! И это притом, что приемник способен не только распознавать кадры, но и отслеживать их нумерацию (см. п. 7.2.2)! А как насчет работы с кадрами без флажковых кодов? Есть и такое решение (см. п. 7.4)!

В трех последующих главах рассмотрены способы кодирования данных для их передачи по каналу связи, схемы выделения синхросигнала и данных из принимаемого сигнала, а также решения, связанные с измерением длины линий связи.

Большинство рассмотренных решений представляют собой изобретения, т. е. защищены патентами. С описаниями приведенных в списке литературы и других зарубежных изобретений можно ознакомиться на сайте <http://www.uspto.gov>. Следует отметить, что на этом сайте Вас ждет мощный поисковый сервер, с помощью которого можно найти конкретные решения в интересующей Вас области. В режиме расширенного поиска (Advanced Search) можно использовать ключевые слова, просматривать дерево ссылок на выбранный патент, отслеживать появление новых патентов интересующего Вас автора и т. д. (всего 31 способ поиска, не считая комбинированных). При этом доступ к базам данных бесплатный.

Описания технических решений и рисунки приводятся в максимально упрощенном виде. В книге нет ни одного точного перевода текста первоисточника (или даже его

фрагментов), на который мы ссылаемся. Более того, многие решения доработаны, введены недостающие детали, без которых пришлось бы прибегать к догадкам и предположениям, чтобы понять идею. По ходу изложения предложен ряд усовершенствований некоторых решений. Вообще говоря, едва ли не каждое приведенное в книге решение можно улучшить, поэтому многие из них могут послужить аналогами и прототипами Ваших будущих изобретений.

Представленные в книге материалы обсуждались со студентами Московского государственного института электронной техники (МИЭТ) в ходе лекций и практических занятий, проводимых авторами. В книге учтены замечания, высказанные студентами и преподавателями кафедры телекоммуникационных систем (ТКС). Особую благодарность за всестороннюю помощь и поддержку авторы выражают заведующему кафедрой телекоммуникационных систем, д. т. н., профессору В. В. Баринову и генеральному директору компании “Зелакс” В. А. Степанову.

В заключение хотелось бы пожелать читателям оптимизма и хорошего делового настроения, которое поможет оценить красоту и мощь многих приведенных здесь решений и схем и не обращать особого внимания на недостатки изложения.

1. Взаимодействие между устройствами по интерфейсу RS-232

Прежде чем начать рассмотрение вопросов обеспечения синхронизации на уровне больших систем, например сетей передачи данных, мы сочли необходимым в первых трех главах остановиться на уровне пользовательского интерфейса. Во-первых, уже на этом уровне появляются вполне достойные внимания задачи, такие как задача выбора местоположения источника синхронизации, задача выравнивания фаз передаваемого и принимаемого синхросигналов и т. п. Во-вторых, как показывает опыт общения с потенциальными потребителями телекоммуникационной аппаратуры, на этом уровне есть некоторые “вечные” вопросы (например, “В чем различие между устройствами типа DTE и DCE?”), которые хотелось бы разрешить.

В этой главе кратко рассмотрены основные сигналы интерфейсов семейства RS на примере интерфейса RS-232. Эти интерфейсы описаны в десятках книг и статей, но мы надеемся, что некоторые приведенные далее пояснения окажутся полезными при их изучении.

1.1. Устройства типа DTE и DCE: сложности терминологии

В литературе по телекоммуникационным системам широко применяются аббревиатуры “DTE” и “DCE”. По определению:

DTE – Data Terminal Equipment – оконечное оборудование данных (ООД);

DCE – Data Communication Equipment – аппаратура передачи данных (АПД).

На первый взгляд, все понятно, и никаких сложностей нет. Но это не совсем так. Чтобы пояснить применимость аббревиатур “DTE” и “DCE” к конкретным устройствам, рассмотрим простой пример системы передачи данных, рис. 1.1. В этой системе компьютеры обмениваются данными по каналу связи через модемы. Компьютеры обозначены аббревиатурой “DTE”, а модемы – “DCE”.



Рис. 1.1. Пример системы передачи данных

Как видно из рис. 1.1, всё на своих местах. Действительно, нельзя отрицать, что каждый из компьютеров является *оконечным* (подключенным к концу тракта передачи данных) и представляет собой *оборудование* для обработки данных. Однако в этих рассуждениях ощущается некоторая “натяжка”. В самом деле, компьютер 2, по-видимому, не будет оконечным, если к нему подключить еще один модем для продолжения канала связи, как показано на рис. 1.2.



Рис. 1.2. Расширенная система передачи данных

Неясно, как после такой модернизации системы правильно назвать компьютер 2: устройством DTE или DCE? Может быть, компьютер 2 превратился из устройства DTE в устройство DCE? Нет. Он, как мы предположили, каким был, таким и остался, к нему лишь подключили дополнительный модем. Еще один неприятный факт, не укладывающийся в “теорию”: многие большие компьютеры изначально выполнены как устройства DCE, а не DTE. Применяв такие компьютеры в схеме на рис. 1.1, мы получим цепочку из устройств типа DCE (рис. 1.3).



рис. 1.3. Система, использующая компьютеры, выполненные в виде устройств DCE

Неясно, где здесь окончное оборудование данных? Точнее, ясно, что это компьютеры, но тогда их следовало бы обозначить аббревиатурой DTE?

И это еще не всё. Во всех приведенных примерах модемы выступали в качестве устройств типа DCE. Противоречий нет, так как модем в полной мере представляет собой *аппаратуру передачи данных*. Однако некоторые современные модемы (например, Зелакс М-144 [77]) могут быть перестроены на работу в качестве устройств типа DTE, хотя терминальных функций они явно не выполняют.

Границы применимости аббревиатуры “DCE” также сильно размыты; возможно поэтому рекомендация ITU-T V.24 (ITU-T – International Telecommunication Union, Telecommunication standardization sector – международный телекоммуникационный союз, сектор стандартизации телекоммуникаций) дает более “осторожную” (но не более четкую) трактовку аббревиатуры “DCE”: Data Circuit-Terminating Equipment – аппаратура окончания канала данных.

Чтобы окончательно не запутаться в терминологии, обратимся к рис. 1.4, на котором представлена “каноническая” схема подключения компьютера (DTE) к линии (каналу связи) через модем (DCE). Обозначения “DTE” и “DCE”, как показывает практика, в подавляющем большинстве случаев привязаны именно к этой схеме и поэтому трактуются однозначно.

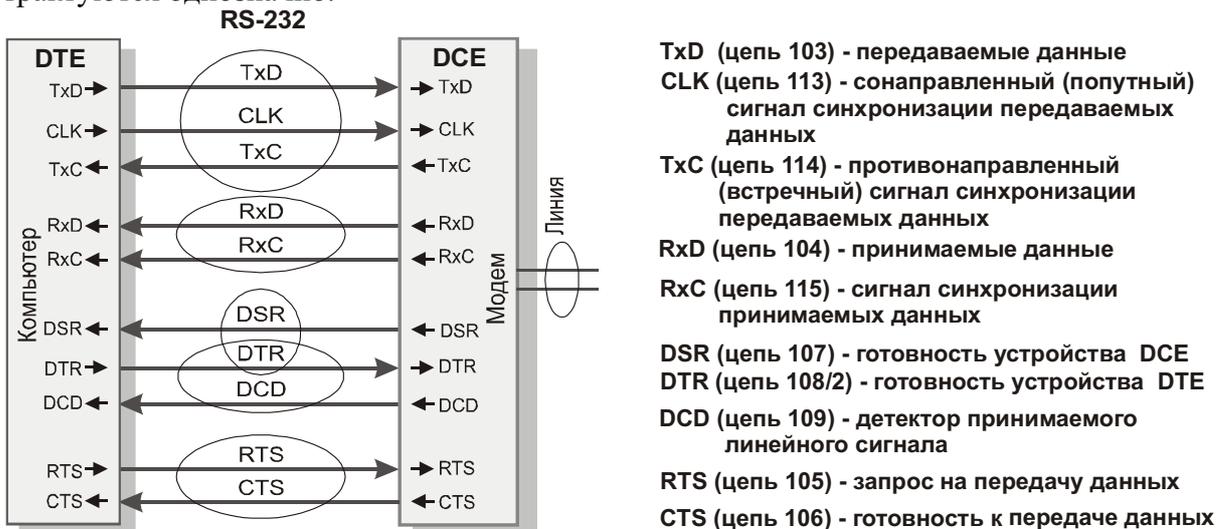


рис. 1.4. Схема соединения устройств типа DTE и DCE

На рисунке приведены основные сигналы интерфейса RS-232. Общая цепь сигнальной земли не показана, родственные группы сигналов выделены овалами. Если отвлечься от функционального назначения компьютера и модема, то можно утверждать,

что устройства типа DTE и DCE различаются направлением передачи сигналов интерфейса RS-232 (или иного интерфейса из семейства RS). Всего лишь этим!

Итак, мы считаем, что:

DTE – устройство, у которого сигнал передаваемых данных TxD является выходным (по ассоциации с фразой “Данные – Тебе”); ☺

DCE – устройство, у которого сигнал передаваемых данных TxD является входным (по ассоциации с фразой “Данные – Себе”). ☺

В этой шутке стопроцентная доля правды. Как видно из рис. 1.4, можно дать до десяти подобных пар определений по числу задействованных сигналов; каждый сигнал – выходной для одного устройства и входной для другого. Может быть, кому-то покажется удобной для запоминания такая пара определений:

DTE – устройство, которое принимает сигнал DCD;

DCE – устройство, которое выдает сигнал DCD.

Здесь рассуждаем так. Если устройство следит за состоянием линии и способно формировать сигнал обнаружения несущей (DCD – Data Carrier Detect), то оно явно *коммуникационное* (DCE); устройство, которое поглощает этот сигнал, не иначе как *терминальное* (DTE). Отметим, что в рекомендации V.24 сигнал DCD трактуется уже несколько иначе по сравнению с приведенной традиционной расшифровкой, хотя смысл остался практически тем же – он именуется как “детектор принимаемого линейного сигнала канала данных” (Data channel received line signal detector).

Таким образом, мы убедились в том, что аббревиатуры “DTE” и “DCE” имеют слабое отношение к функциональному назначению обозначаемых ими устройств. Они всего лишь определяют направления передачи сигналов. Если согласиться с таким подходом к терминологии, то схемы, показанные на рис. 1.2 и рис. 1.3, не будут выглядеть “странными”; таковыми, скорее, покажутся вопросы, поставленные в комментариях к этим рисункам в тщетных поисках здравого смысла.

При описании взаимодействия устройств типа DTE и DCE мы остановились на интерфейсе RS-232 как наиболее простом и распространенном. Этот интерфейс был разработан в 1969 г. и с тех пор не раз модифицировался и “ветвился”. Однако показанные на рис. 1.4 сигналы, передаваемые между устройствами DTE и DCE, можно найти в любой его версии, если иметь в виду логическое представление сигналов (“есть – нет”).

Иногда ошибочно считается, что интерфейс RS-232 работает только в асинхронном режиме (о синхронном и асинхронном режимах – чуть позже). Эта типичная ошибка основана на широко распространенной реализации минимального подмножества сигналов интерфейса RS-232 в персональных компьютерах. Не следует ассоциировать термин “RS-232” с термином “асинхронный режим” работы порта, так как полный вариант интерфейса может работать как в асинхронном, так и в синхронном режимах. Последний характеризуется расширенным набором участвующих в обмене цепей, точнее, дополнительным использованием трех цепей синхронизации: CLK, TxС и RxС (см. рис. 1.4).

На физическом уровне сигналы передаются разными уровнями напряжения. Например, стандарты RS-232C, RS-232D предусматривают уровни передачи сигнала от ± 5 В (min) до ± 15 В (max) относительно общей сигнальной земли; в стандарте RS-422A сигналы передаются в дифференциальном виде ± 2 В (min), в стандарте RS-485 – уровнями напряжения $\pm 1,5$ В (min) и т. д.

К последним модификациям интерфейсов семейства RS-232 можно отнести следующие: ANSI/TIA/EIA-232-F-1999, ANSI/TIA/EIA-574-90(R98), ANSI/TIA/EIA-723-98, ANSI/TIA/EIA-404-B-96. С подробностями можно ознакомиться на сайте ассоциации производителей телекоммуникационного оборудования TIA (The Telecommunication Industry Association) [9].

1.2. Основные сигналы и режимы работы интерфейса RS-232

К основным сигналам интерфейса RS-232 можно отнести следующие: TxD, CLK, TxC, RxD, RxC, DTR, DSR, DCD, RTS, CTS.

Вновь рассмотрим пару “компьютер - модем”. Полный интерфейс RS-232 предусматривает использование 25-контактного соединителя, в котором каждый контакт соответствует определенной цепи, включая цепи защитного и сигнального заземления. Однако на практике обычно используются далеко не все цепи. Например, лишними часто оказываются цепи* 118 – 122 обслуживания дополнительного канала связи, цепи 111, 112, 116 управления скоростью передачи и другие.

Конечно, если нечего обслуживать или нечем управлять, то и соответствующие цепи не нужны. Но что интересно: даже если есть дополнительный канал связи, и скорость передачи должна регулироваться, то и тогда чаще всего перечисленные цепи не используются!

Дело в том, что современные “интеллектуальные” модемы могут получать по цепи TxD не только данные, но и команды, флаги и иные признаки, предусмотренные программным протоколом и характеризующие параметры обмена. Так что для передачи служебной информации из компьютера в модем и обратно не обязательно и даже нежелательно использовать “второстепенные” цепи интерфейса RS-232, из которых часть мы здесь упомянули. Но есть и такие цепи, без которых трудно обойтись; их рассмотрим после небольшого замечания относительно ставшего “второстепенным” сигнала RI.

Он по праву не вошел в десятку основных сигналов, так как с повышением интеллекта модема надобность в нем просто отпала. Напомним, что сигнал RI (Ring Indicator, цепь 125 – индикатор вызова) логически повторяет сигнал вызова абонента со стороны АТС. Точнее, сигнал RI активен (его уровень напряжения составляет +12 В), когда на “нашу” абонентскую линию из АТС подается относительно высокое переменное напряжение (“телефон звонит”); когда переменное напряжение снимается (“звонок умолкает”), то сигнал RI возвращается в пассивное состояние (-12 В).

Чтобы понять, почему столь “важный” сигнал стал ненужным, рассмотрим цепочку: АТС – абонентская линия – модем – компьютер (рисунок не приводим). Предположим, что в исходном состоянии абонентская линия свободна (“трубка повешена”). Если модем не обладает высоким “интеллектом”, то он при вызове со стороны АТС просто формирует и передает в компьютер сигналы RI. Компьютер реагирует на них, например, так: по окончании шестого сигнала передает в модем команду “снять трубку”; после некоторой паузы проверяет наличие сигналов DCD, DSR, CTS и приступает к обмену служебными, а затем и “полезными” данными.

Если модем “интеллектуален”, то он и сам может отсчитать заданное число вызывных сигналов, затем “снять трубку” и выполнить все операции по установлению связи, идентификации инициатора обмена, приему и буферизации полученных от него данных и т. п.; при этом компьютер подключается к работе только когда это “крайне необходимо”. Как видим, в этой ситуации цепь передачи сигнала RI из модема в компьютер действительно не нужна.

* Имеются в виду цепи “серии 100”, описанные в рекомендации V. 24 ITU-T. Большинство цепей имеют сокращенные названия (например TxD), введенные ассоциацией электронной промышленности Electronic Industries Association (EIA). Названия и их расшифровка в ряде случаев со временем изменяются, например, вместо TxD ныне в интерфейсе RS-232F использовано сокращение TD. Но не будем судить об этой ситуации слишком строго.

Впрочем, вернемся к десятке наиболее употребляемых сигналов (см. рис. 1.4). При их описании номера цепей и расшифровка терминов даны в соответствии с рекомендацией V.24. (Как уже отмечалось при упоминании сигнала DCD, эта расшифровка не всегда совпадает с привычной.) Направление передачи сигналов видно из рисунка.

Сигналы TxD, CLK, TxC, RxD, RxC

Сигналы TxD, CLK, TxC, RxD, RxC примечательны тем, что их функциональное назначение не вызывает неоднозначной трактовки. С них и начнем.

TxD (цепь 103) – Transmitted Data – данные, передаваемые из устройства DTE в устройство DCE в асинхронном или синхронном режиме.

В синхронном режиме границы и середины битовых интервалов соответствуют положительным и отрицательным фронтам синхросигнала, передаваемого по отдельной цепи. Этот режим преимущественно используется в высокопроизводительных системах передачи данных.

В асинхронном режиме каждая передаваемая группа битов (обычно – байт) обрамляется двумя служебными битами Старт и Стоп. Приемник данных определяет середины битовых интервалов, начиная отсчет времени от начала бита Старт. Таким образом, цепь передачи синхросигнала не используется. Асинхронный режим применяется очень широко, так как его аппаратная реализация более проста – на рынке имеется ряд дешевых микросхем универсальных асинхронных приемопередатчиков (UART). Но за эту простоту приходится платить снижением эффективности использования цепей передачи данных, как будет показано далее.

Взаимодействие устройств в асинхронном режиме

Напомним, что для конкретности мы рассматриваем интерфейс RS-232, хотя логические соотношения между сигналами такие же, как и в иных интерфейсах семейства RS (RS-422, RS-485 и др.).

В асинхронном режиме (рис. 1.5) передаваемые данные представлены потоком символов, каждый из которых снабжен служебными битами Старт и Стоп.

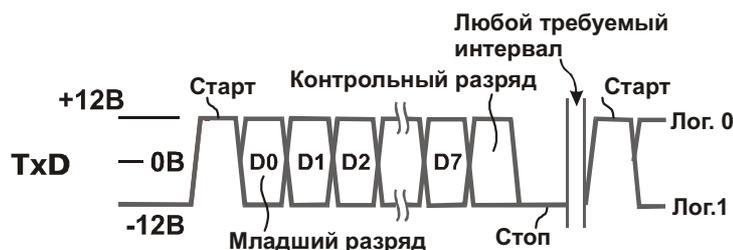


рис. 1.5. Временная диаграмма передачи данных в асинхронном режиме (здесь и далее – для интерфейса RS-232)

Число стоп-битов при настройке устройств обычно выбирают равным 1 или 2, реже 1,5 (имеется в виду интервал времени, в полтора раза превышающий длительность битового интервала). Число битов (5...8) в символе, наличие или отсутствие контрольного разряда (дополняющего число единичных битов символа до четного или нечетного) и скорость передачи также задаются при начальной настройке. Скорость передачи V обычно составляет 50... 460800 бит/с, но может быть и более высокой.

Приемник синхронизируется положительным фронтом (переходом напряжения от -12 В к $+12$ В) сигнала Старт. Зная длительность битового интервала $T = 1/V$ и формат посылки, приемник последовательно считывает передаваемые биты. Асинхронность состоит в том, что приемник не знает, в какой момент поступит очередной символ.

Иными словами, отсутствует единая для всего потока данных синхронизация, т. е. она имеет локальный характер – устанавливается заново всякий раз при обнаружении приемником начала очередного старт-бита.

Взаимодействие устройств в синхронном режиме

Синхронный режим позволяет более эффективно использовать цепи передачи данных за счет исключения старт- и стоп-битов. Поясним это на примере.

Предположим, что при передаче данных в асинхронном режиме символ содержит 8 бит (байт), контроль по четности или нечетности отсутствует, стоп-бит один. Тогда, с учетом стартового бита, для передачи каждого байта нужно 10 битовых интервалов (соответствующих передаче старт-бита, восьми битов данных и стоп-бита). При объявленной скорости передачи данных 115200 бит/с и отсутствии пауз между стоп- и старт-битами скорость передачи байтов равна $115200 / 10 = 11520$ байт/с или $11520 \times 8 = 92160$ бит/с. Таким образом, фактическая скорость передачи данных составляет 0,8 от объявленной.

В синхронном режиме эти скорости одинаковы, так как передаются только полезные биты; их истинность подтверждается синхросигналами в соответствующих цепях интерфейса. Возможны два варианта взаимодействия устройств.

Вариант 1. Направления передачи синхросигналов CLK и передаваемых данных TxD совпадают (codirectional – сонаправленная или попутная синхронизация), т. е. источник данных и синхросигналов один и тот же (рис. 1.6, а).

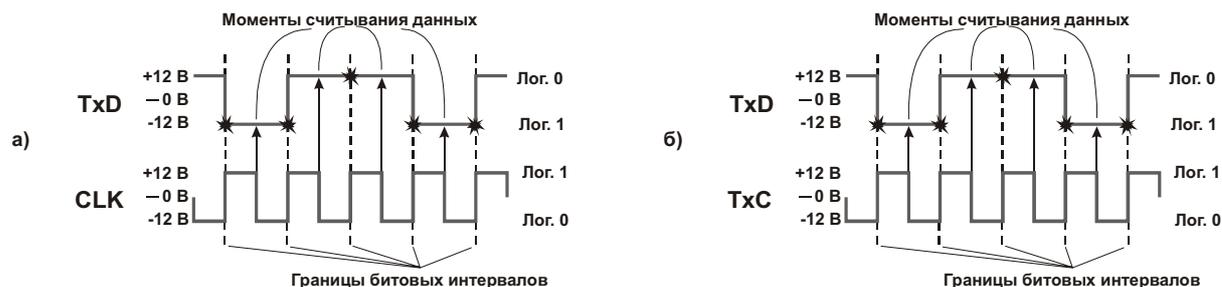


Рис. 1.6. Временные диаграммы передачи данных TxD в синхронном режиме:
а – вариант 1; б – вариант 2

CLK (цепь 113) – Transmitter signal element timing (DTE) – сигнал синхронизации передаваемых данных, формируемый устройством DTE. По отрицательному фронту (перепаду напряжения от +12 В до -12 В) сигнала CLK приемник запоминает бит данных. По положительному фронту этого сигнала в линию TxD поступает следующий бит.

Далее для краткости описания систем, в которых используется синхронный обмен данными, используется термин “синхросетка”. Он определяет систему отсчета битовых интервалов и соответствует показанной на рисунке разметке сигнала данных с помощью группы штриховых вертикальных линий. Выражение “два потока данных размещены в одной синхросетке” означает, что границы битовых интервалов этих потоков совпадают во времени. Напротив, выражение “два потока данных размещены во взаимно смещенных на 180 градусов синхросетках” означает, что границы битовых интервалов этих потоков не совпадают во времени: центр битового интервала одного потока совмещен с границей битового интервала второго потока, и наоборот.

Вариант 2. Направления передачи синхросигналов TxC и передаваемых данных TxD противоположны (contradirectional – противонаправленная или встречная синхронизация), т. е. источник данных размещен в одном устройстве, а источник синхросигналов – в другом (рис. 1.6, б).

TxC (цепь 114) – Transmitter signal element timing (DCE) – сигнал синхронизации передаваемых данных TxD, формируемый устройством DCE. Получив положительный фронт этого сигнала, передатчик выдает очередной бит; затем по отрицательному фронту сигнала приемник фиксирует новый бит и т. д. Сигналы CLK или TxC передаются непрерывно на протяжении всего сеанса связи между устройствами. Сквасженность этих сигналов обычно равна двум.

Сравнение методов попутной и встречной синхронизации

Какая синхронизация лучше: попутная (CLK – TxD) или встречная (TxC – TxD)? Если рассуждать теоретически, то предпочтительна первая, и вот почему. В первом варианте передачи данных задержки распространения сигналов TxD и CLK в кабеле между устройствами DTE и DCE в значительной мере взаимно компенсируются благодаря примерно одинаковым условиям следования этих сигналов “параллельными курсами”.

Второй вариант основан на не совсем оправданной “вере” в то, что запрос (положительный фронт сигнала TxC) на выдачу очередного бита данных будет мгновенно удовлетворен, и отрицательный фронт сигнала TxC попадет в середину битового интервала (см. рис. 1.6, б). Но отклонение составляет как минимум две задержки распространения сигналов по кабелю и четыре задержки интерфейсных схем на пути прохождения запроса (TxC) и ответа (TxD). Действительно, ведь правильно было бы передать запрос, дождаться поступления бита данных, и только тогда начинать отсчет интервала времени для указания момента последующего приема этого бита. Конечно, отмеченные различия между вариантами не существенны при низких скоростях передачи данных.

И все же на практике чаще всего применяется встречная синхронизация. Это связано с тем, что во всех отношениях удобно провести границу между системой транспортирования данных и прочими устройствами. А если это так, то система транспортирования должна быть функционально законченной и, в частности, иметь собственный генератор синхросигналов. Таким образом, принимаемые в систему транспортирования данные неизбежно окажутся объектом встречной синхронизации.

Когда полезно проинвертировать синхросигнал

Реальность часто вносит коррективы даже в простые решения, которые мы сейчас рассматриваем. В силу разных причин, начиная от влияния паразитных емкостей и заканчивая элементарными ошибками при монтаже устройства (когда в труднодоступном месте перепутаны контакты, на которые выведен парафазный синхросигнал), реальная временная диаграмма может сильно отличаться от теоретической. Причем настолько, что имеет смысл проинвертировать синхросигнал, чтобы получить более точное попадание его отрицательного фронта на период стабильности бита данных, как показано на рис. 1.7.

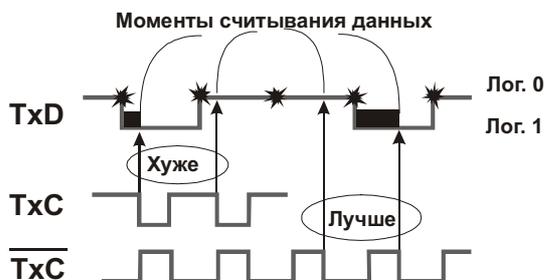


Рис. 1.7. Реальная временная диаграмма. Сквасженность синхросигналов отлична от двух, а данные поступают с задержкой

На рисунке затемненными прямоугольниками показаны запасы времени установления сигнала TxD при его регистрации. Очевидно, что лучше иметь больший запас, чем меньший, хотя здесь нас может подстеречь другая опасность – приближая отрицательный фронт синхросигнала к концу битового интервала (т. е. увеличивая длину затемненного прямоугольника), мы рискуем зарегистрировать нестабильные данные.

Рассмотрим сигналы RxD и RxC.

RxD (цепь 104) – Received data – данные, принимаемые устройством DTE в асинхронном или синхронном режиме. Асинхронный режим передачи сигналов RxD аналогичен рассмотренному ранее (см. рис. 1.5, на котором обозначение TxD теперь следовало бы изменить на RxD). Синхронный режим также аналогичен рассмотренному ранее первому варианту передачи сигналов TxD (рис. 1.8).

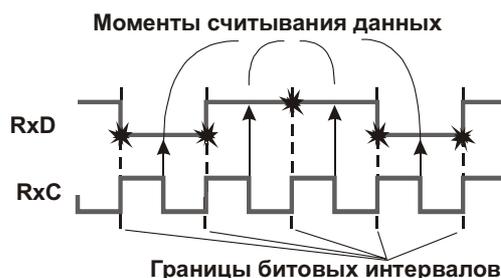


Рис. 1.8. Временная диаграмма передачи данных RxD в синхронном режиме

RxC (цепь 115) – Receiver signal element timing (DCE) – сигнал синхронизации данных RxD, формируемый устройством DCE. Истинность бита данных подтверждается отрицательным фронтом сигнала RxC, как показано на рисунке; смена бита данных возможна по положительному фронту сигнала RxC.

Пары сигналов DTR – DSR и DTR – DCD

Начнем с определений.

DTR (цепь 108/2) – Data terminal ready – готовность устройства DTE к обмену данными.

DSR (цепь 107) – Data set ready – готовность устройства DCE к обмену данными.

DCD (цепь 109) – Data channel received line signal detector – детектор принимаемого линейного сигнала канала данных. Иными словами, это подтверждение наличия в линии сигнала от удаленного абонента, причем гарантируется, что параметры принимаемого сигнала лежат в заранее оговоренных пределах для его уверенного распознавания. Напомним, что традиционная расшифровка сокращения DCD – Data carrier detect – сигнал обнаружения несущей.

Первоначально сигналы DTR и DSR рассматривались как парные, т. е. взаимодополняющие, квитирующие. Это вполне естественно, так как прежде чем начать обмен данными, нужно как минимум иметь сведения о готовности партнера к обмену. Однако сегодня пара DTR – DSR явно устарела, и более информативной выглядит пара DTR – DCD. Чтобы понять, почему это произошло, обратимся к недалекому прошлому.

В ранних разработках телекоммуникационной аппаратуры широко использовался полудуплексный режим обмена данными с удаленным абонентом. В таком режиме данные передаются попеременно то в одном, то в другом направлении. Ясно, что в полудуплексном режиме при передаче данных в линию сигнал DCD не несет информации, так как принимаемого сигнала нет. Поэтому пара сигналов DTR – DSR, как и положено, подтверждает взаимную готовность устройств к работе, а сигнал DCD обретает смысл лишь в периоды приема данных из линии.

С развитием телекоммуникационной аппаратуры основным режимом обмена стал полностью дуплексный, при котором данные передаются одновременно в обе стороны. В таком режиме сигнал DCD сохраняет смысл на протяжении всего времени пребывания на связи пары модемов. Поэтому появилась возможность возложить на сигнал DCD дополнительную смысловую нагрузку, которую ранее нес сигнал DSR (последний и стал лишним). Другими словами, теперь многие устройства типа DCE спроектированы так, что сигнал DCD отвечает не только за обнаружение несущей, но и за общую готовность устройства к работе.

Таким образом, сигнал DSR во многом утратил былое значение, и сейчас в массе производимых отечественных и зарубежных изделий пары взаимодополняющих сигналов DTR – DCD встречаются чаще, чем DTR – DSR. При этом цепь передачи сигнала DSR может отсутствовать; на освободившийся вход DSR устройства DTE обычно подается постоянное напряжение +12 В, имитирующее готовность устройства DCE к работе.

Сигналы RTS и CTS

В общем случае эти сигналы используются для управления потоками данных. Однако их первоначальное назначение (для отображения запроса и готовности передачи данных от устройства DTE к устройству DCE) в настоящее время зачастую игнорируется – эти сигналы могут альтернативно трактоваться как равноправные признаки готовности устройств DTE и DCE к приему данных от устройства – партнера.

Поясним сказанное. Сначала рассмотрим первоначальное назначение сигналов RTS и CTS, определенное в рекомендации V.24.

Первоначальное назначение . . .

RTS (цепь 105) – Request to send – запрос на передачу данных;

CTS (цепь 106) – Ready for sending – готовность к передаче данных. Традиционная расшифровка аббревиатуры CTS – Clear to send – означает “свободен, прозрачен для передачи”.

Во времена преимущественного использования полудуплексного обмена данными с удаленным абонентом сигналы RTS и CTS имели однозначную трактовку (она справедлива и сейчас для полудуплексных систем). Сигнал RTS выражал просьбу, адресованную устройству DCE со стороны устройства DTE, примерно такого содержания: “Прошу при первой возможности переключиться из состояния прослушивания линии связи с удаленным абонентом в режим передачи данных в эту линию.” После выполнения этой просьбы в устройство DTE посылался ответный сигнал CTS подтверждения факта переключения устройства DCE с приема на передачу данных в линию. Отметим, что к этому моменту удаленный абонент также должен был успеть переключиться с передачи на прием (если он передавал данные в ту же линию), чтобы исключить конфликты. И, наконец, после получения сигнала CTS устройство DTE начинало выдачу данных TxD.

. . . и альтернативное

В асинхронном режиме сигналы RTS и CTS обслуживают оба направления передачи данных, что выходит за рамки рекомендации V.24 и отражает некий “стандарт де-факто”. Поэтому наименования сигналов не соответствуют (и даже противоречат) их назначению. Сигнал RTS теперь рассматривается как готовность устройства DTE принять данные RxD от устройства DCE. Аналогично сигнал CTS свидетельствует о готовности устройства DCE принять данные TxD от устройства DTE и способности передать эти данные в линию.

Логика работы такова: передача данных в ту или иную сторону возможна только при условии, что приемник готов эти данные принять. Если обнаружена неготовность приемника, то источник данных приостанавливает работу, ждет появления готовности,

возобновляет передачу и т. д. Это – так называемое аппаратное управление потоком данных (hardware flow control). Рекомендуем также ознакомиться с решениями, описанными в п. 3.4.

Программное управление потоком данных

В отличие от только что рассмотренного аппаратного управления потоком данных, программное управление (software flow control) применяется в асинхронном режиме при использовании кода ASCII (или иного символьного кода, построенного на основе ASCII). Сигналы RTS и CTS не используются, на соответствующие входы устройств DCE и DTE подается напряжение +12 В, имитирующее “аппаратную” готовность устройств к обмену данными. Поясним, как осуществляются процессы программного управления потоками данных между устройствами DTE и DCE.

В устройстве DTE имеется входная буферная память, в которую записывается поток данных, принимаемых по цепи RxD. Аналогично в устройстве DCE содержится буферная память для временного хранения данных, принимаемых по цепи TxD. Задача состоит в том, чтобы предотвратить переполнение буферной памяти в каждом устройстве, если темп поступления данных выше темпа их рассасывания.

Рассмотрим сначала ситуацию, при которой устройство DTE (компьютер, см. рис. 1.4) не справляется с потоком данных RxD, поступающих из линии через устройство DCE (модем). Такая ситуация возможна даже при не очень высокой скорости асинхронного обмена, если, например, компьютер в данный период выполняет более приоритетную задачу, не связанную с текущим обменом.

Когда буфер устройства DTE заполняется до некоторого критического уровня, например до 90% , компьютер переходит к прерывающей программе, которая предписывает выдать в цепь TxD символ Xoff – код $13_{16} = 0001.0011_2$ в ASCII. Получив этот символ, модем приостанавливает выдачу данных RxD. В зависимости от построения программного обеспечения компьютер либо сразу возвращается к прерванной задаче, либо приступает к разгрузке буфера, либо переходит к каким-то иным действиям. В любом случае должен наступить момент, когда буфер окажется достаточно свободным для получения новых данных. Тогда в цепь TxD будет программно выдан символ Xon = 11_{16} , модем возобновит выдачу данных RxD и т. д.

Противоположная ситуация состоит в том, что модем не справляется с потоком данных TxD. В этом случае он посылает в компьютер по цепи RxD символ Xoff, приостанавливая поток. После рассасывания данных через линию модем посылает по цепи RxD символ Xon, передача данных возобновляется и т. д.

Примечательно, что символы Xon и Xoff, посылаемые из компьютера в модем, в линию связи с удаленным абонентом не передаются.

Трехпроводный вариант интерфейса RS-232

В этом упрощенном варианте интерфейса число используемых сигналов сокращено до двух. Кабель, соединяющий два устройства, содержит всего три провода: первый – для объединения цепей сигнальной земли, второй и третий – для передачи сигналов TxD и RxD (см. рис. 1.4, на котором следовало бы исключить все “лишние” связи, а на освободившиеся входы подать напряжение +12 В). Чтобы уменьшить амплитуду перекрестных помех, следует вместо трех проводов использовать две витые пары “земля – сигнал”.

Поскольку цепи синхронизации отключены, обмен данными возможен только в асинхронном режиме. Как уже отмечалось, при достаточном “интеллекте” устройств в поток символов можно вводить “уникальные” коды, которые расцениваются приемни-

ком как команды, так что трехпроводный вариант интерфейса оказывается не столь примитивным, как это может показаться на первый взгляд.

Существует и аппаратная поддержка трехпроводного варианта интерфейса, правда, затрагивающая всего лишь один режим, связанный с окончанием сеанса связи между устройствами. Если одно из устройств желает прекратить взаимодействие с другим устройством, то оно вместо обычного формирует расширенный старт-бит длительностью более 300 мс. Устройство – приемник первоначально расценивает эту ситуацию как ошибочную, но по истечении 300 мс принимает к сведению факт логического разрыва соединения и очищает свою буферную память от ошибочных данных, принятых за это время.

Электрические уровни сигналов интерфейса RS-232

Возвращаясь к рис. 1.4 отметим, что, согласно интерфейсу RS-232, электрические уровни всех входящих в него сигналов отсчитываются от сигнальной земли SG, цепь 102 – signal ground or common return. Управляющие сигналы передаются в прямом коде, т. е. наличию сигнала соответствует типовой уровень напряжения +12 В, а его отсутствию – минус 12 В. Данные TxD и RxD представляются в обратном коде, т. е. сигналу лог. 1 соответствует напряжение минус 12 В, а сигналу лог. 0 – напряжение +12 В (см. рис. 1.5 – рис. 1.7). Старт- и стоп-биты передаются соответственно положительным и отрицательным уровнями напряжения.

Формирователь сигнала должен выдавать напряжение высокого уровня в пределах +5...+15 В или напряжение низкого уровня в пределах минус 5...15 В (типовое значение: ±12 В). Приемник расценивает входное напряжение из диапазона +3...+25 В как напряжение высокого уровня, и из диапазона минус 3...25 В как напряжение низкого уровня.

Если вход не используется, то на него следует подать соответствующее напряжение высокого или низкого уровня. Например, входы DSR, DCD, TxС и CTS (см. рис. 1.4) можно отключить от соответствующих цепей и подать на них напряжение +12 В. При таком подключении предполагается, что модем всегда готов к работе, в линии постоянно присутствует полноценный принимаемый сигнал, передача данных TxD в синхронном режиме происходит только с использованием сигнала CLK, управление потоком данных в асинхронном режиме либо программное (Xon – Xoff), либо аппаратное (hardware flow control), но одностороннее, при котором модем никогда не “возражает” против приема данных TxD, а в синхронном режиме он всегда готов начать или продолжить прием данных TxD и их передачу в линию. Отметим, что поскольку напряжение +12 В в явном виде не выведено на соединитель, входы DSR, DCD, TxС и CTS можно соединить с выходом DTR, на котором при нормальной работе устройства это напряжение присутствует.

Предыдущие описания сигналов и режимов работы интерфейса RS-232 основывались на “классической” схеме соединения устройств типа DTE и DCE (см. рис. 1.4). Однако эта схема не является единственно возможной. Рассмотрим другие схемы соединения взаимодействующих устройств.

1.3. Асинхронный обмен данными между одноименными устройствами типа DTE/DCE

На практике часто необходимо соединить между собой два одноименных устройства: DTE – DTE или DCE – DCE. Сначала рассмотрим взаимодействие этих устройств в асинхронном режиме (при котором цепи синхронизации не используются); взаимодействие в синхронном режиме описано в гл. 2.

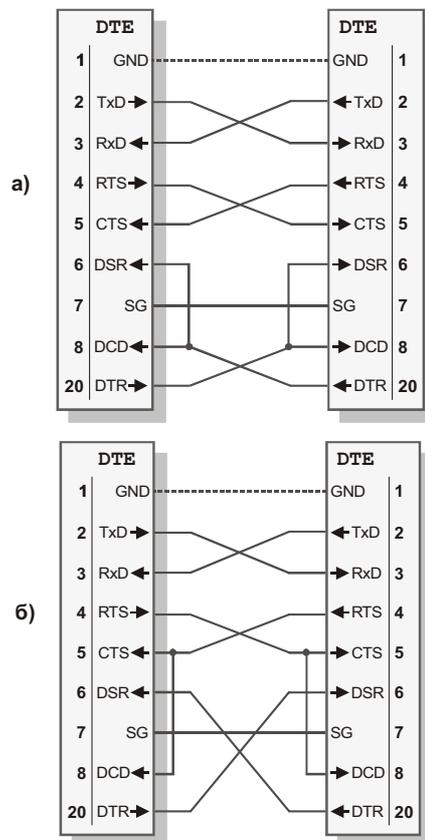
Варианты сопряжения двух устройств типа DTE

Рассмотрим типовые варианты сопряжения, представленные на рис. 1.9. Предположим, что все показанные на нем устройства – компьютеры, и каждый из них уверен, что его партнер – модем (именно для такой пары и был в свое время разработан интерфейс RS-232). Но модема в действительности нет, поэтому в данной ситуации соединительные кабели между устройствами называют нуль-модемными.

На рисунке представлены не все возможные варианты кабелей. К сожалению, если в документации на устройство нет четко сформулированных условий формирования и проверки управляющих интерфейсных сигналов (а так чаще всего и бывает), то “творческий процесс” неизбежен, т. е. придется применять и изобретать разные варианты кабелей.

Цифрами обозначены номера контактов соединителя DB-25 (вилка), штриховой линией – провод, соединяющий цепи GND защитной земли (этих цепей может и не быть); цепи сигнальной земли SG в общем случае изолированы от цепей GND. Во всех вариантах выход TxD соединен с входом RxD устройства – партнера.

Вариант а отличается от варианта б способом формирования входного сигнала DCD: в первом случае он повторяет сигнал DTR, во втором – сигнал RTS устройства – партнера. Вариант а, пожалуй, наиболее логичен. Действительно, здесь мы видим две (и даже три!) классические пары взаимодополняющих сигналов управления: RTS – CTS, DTR – DSR и DTR – DCD. Этот вариант наиболее распространен, и его имеет смысл использовать в тех случаях, когда нет достаточной информации об особенностях реализации интерфейса соединяемых устройств.



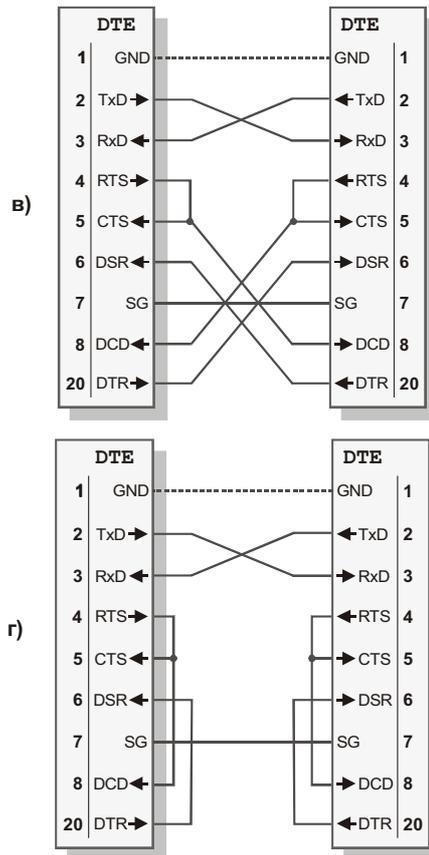


рис. 1.9. Некоторые варианты соединения устройств типа DTE

В варианте б входы CTS и DCD объединены. Это означает, что пассивное состояние сигнала на объединенных входах (напряжение минус 12 В) воспринимается не только как неготовность приемника устройства – партнера, но и как его общая неготовность к работе. Поэтому помимо классического аппаратного управления потоком данных по цепям RTS – CTS (как в варианте а) здесь наблюдается некий побочный эффект. Он заключается в том, что приостановка потока в одном из направлений (снятием сигнала RTS) влечет за собой прекращение приема данных, передаваемых в противоположном направлении, так как при отсутствии сигнала DCD принимать данные нет смысла. Хорошо это или плохо – решает тот, кто точно знает цель построения конкретной системы из двух устройств типа DTE.

Вариант в отличается от варианта б невозможностью работы в режиме аппаратного управления потоком данных, так как между устройствами нет связей по цепям RTS – CTS. Сходство этих вариантов – в полной блокировке обмена данными между устройствами при отсутствии сигнала RTS хотя бы в одном из них.

Вариант г предполагает постоянную готовность устройства – партнера к работе, так как входные управляющие сигналы имитируются, а не отражают истинное положение вещей.

Пример сопряжения двух устройств типа DCE

Схемы сопряжения устройств типа DCE по смыслу близки рассмотренным ранее (см. рис. 1.9). Они содержат общие цепи сигнальной SG и (не всегда) защитной GND земли, а также две перекрестные связи RxD – TxD. Типовой пример такой схемы приведен на рис. 1.10.

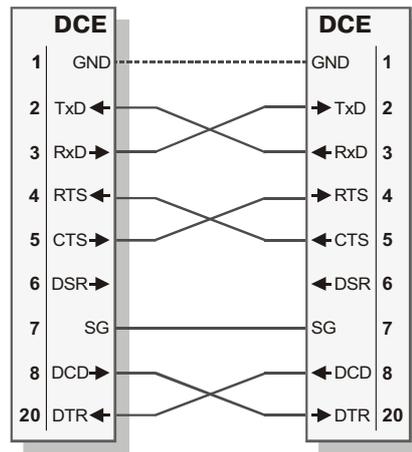


рис. 1.10. Схема сопряжения двух устройств типа DCE

В данном примере задействованы пары цепей CTS – RTS и DCD – DTR. Здесь также можно предложить ряд вариантов соединения. Например, вместо выхода DCD можно использовать выход DSR, выходы CTS можно соединить перемычками с входами RTS своих же устройств и т. п.

Интересно отметить, что соединительные кабели между устройствами DCE часто также называют нуль-модемными, хотя их по аналогии правильнее было бы назвать нуль-компьютерными.

2. Взаимодействие устройств типа DTE/DCE в синхронном режиме: типовые решения

В синхронном режиме, так же как и в асинхронном, возможно взаимодействие устройств типа DTE и DCE в любых сочетаниях.

Напомним, что пара устройств DTE – DCE обычно объединяется прямыми связями между одноименными контактами соединителей (см. рис. 1.4). Отметим также, что три сигнала синхронизирующей группы (CLK, TxС, RxС) не используются одновременно: возможны лишь сочетания CLK – RxС или TxС – RxС. При объединении одноименных пар устройств (DTE – DTE и DCE – DCE) помимо использования цепей, рассмотренных ранее (см. рис. 1.9 и рис. 1.10), необходимо должным образом соединить контакты, отвечающие за синхронизацию.

Можно предложить десятки вариантов схем сопряжения одноименных и разноименных устройств (DTE и DCE). Эти схемы различаются числом и местоположением источников синхросигналов, выбором сочетания цепей CLK – RxС или TxС – RxС, структурными особенностями устройств, схемами кабелей и т. д.

Чтобы не запутаться в этих вариантах и не тратить силы на изучение “полуэкзотических” решений, рассмотрим лишь некоторые показательные схемы сопряжения. Для этого обратимся к наиболее распространенным примерам построения синхронных каналов связи между удаленными устройствами (рис. 2.1 – рис. 2.4).

В схемах, представленных на этих рисунках, данные передаются между оконечными устройствами через устройства типа DCE 1 и 2 (в наших примерах – через модемы) по каналу связи. Простейший канал связи – это одна или две витые пары медных проводов. В более общем случае канал связи может содержать последовательно включенные ретрансляторы, мультиплексоры и иные устройства. В наших примерах существенно только то, что канал связи представляет собой всего лишь некоторую среду передачи данных, смешанных с синхросигналами. При этом передача ведется одновременно в обоих направлениях.

Канал связи с подключенными к его началу и концу устройствами типа DCE удобно рассматривать как элементарную функционально-законченную транспортную систему передачи данных. Эта система может синхронизироваться от собственных или внешних опорных генераторов; тогда ее обычно называют соответственно системой с внутренней или внешней синхронизацией.

2.1. Системы с внутренней синхронизацией

В схеме, приведенной на рис. 2.1, каждое направление передачи данных обслуживается соответствующим генератором G1 и G2 синхросигналов высокой точности и стабильности*. Номинальные частоты сигналов этих генераторов одинаковы, но фактически они, конечно, несколько различны (абсолютного совпадения быть не может). Рассмотрим процесс передачи данных из устройства DTE 1 в устройство DTE 2.

* Термин “точность” определяет степень соответствия номинальной (заявленной) частоты генератора усредненной по времени фактической (измеренной) частоте. Например, генератор с номинальной частотой 10 МГц может иметь фактическую частоту, усредненную за 100 часов, отличающуюся от номинальной на 0,4 Гц. Термин “стабильность” определяет степень соответствия фактической усредненной по времени частоты мгновенной измеренной частоте. Эти частоты могут различаться, например, на 0,7 Гц. Таким образом, генератор может быть: а) точным и стабильным; б) точным, но нестабильным; в) неточным, но стабильным; г) неточным и нестабильным. Для оценки генераторов приняты стандартные уровни качества (см. п. 6.1).

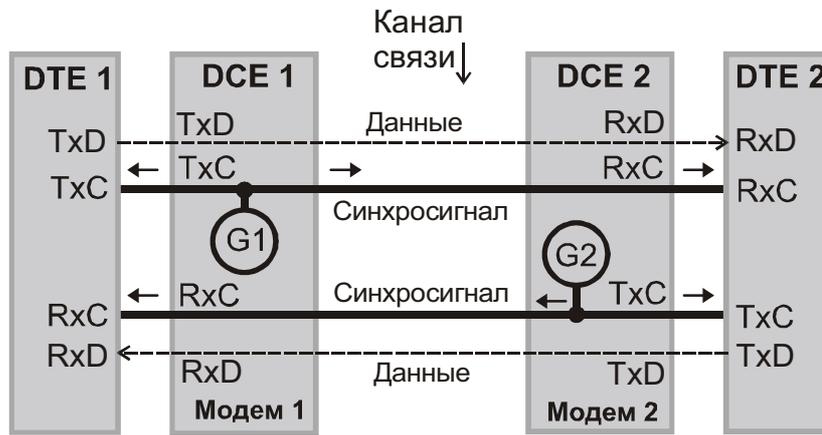


Рис. 2.1. Система с внутренней синхронизацией – первый вариант.

Под действием сигнала TxC от генератора $G1$ устройство $DTE 1$ выдает данные TxD в соответствии с временной диаграммой, приведенной на рис. 1.6, б. Эти данные поступают в модем 1, временно в нем запоминаются и затем под действием синхросигнала от того же генератора $G1$ кодируются и передаются в канал связи. Таким образом, в канал поступает смесь данных с синхросигналом.

Модем 2 выделяет из полученной по каналу смеси синхросигнал и данные. Синхросигнал очищается от помех, данные временно запоминаются. Далее под управлением восстановленного синхросигнала, именуемого теперь сигналом RxC , данные RxD передаются из модема 2 в устройство $DTE 2$ в соответствии с временной диаграммой, приведенной на рис. 1.8. Схема симметрична, поэтому процесс передачи данных в обратном направлении аналогичен описанному.

Схема, показанная на рис. 2.2, отличается от предыдущей тем, что вместо синхросигнала от генератора $G2$ используется синхросигнал, выделенный из канала. В данном случае все процессы протекают под управлением генератора $G1$ ведущего (Master) модема 1. Ведомый (Slave) модем 2, по существу, помимо прочих, выполняет функцию ретранслятора синхросигнала от генератора $G1$.

Передача данных из устройства $DTE 1$ в устройство $DTE 2$ аналогична описанной ранее. При передаче данных в обратном направлении они временно запоминаются в модеме 2 и затем выдаются в канал, но в этом случае процессы синхронизируются выделенным из канала сигналом.

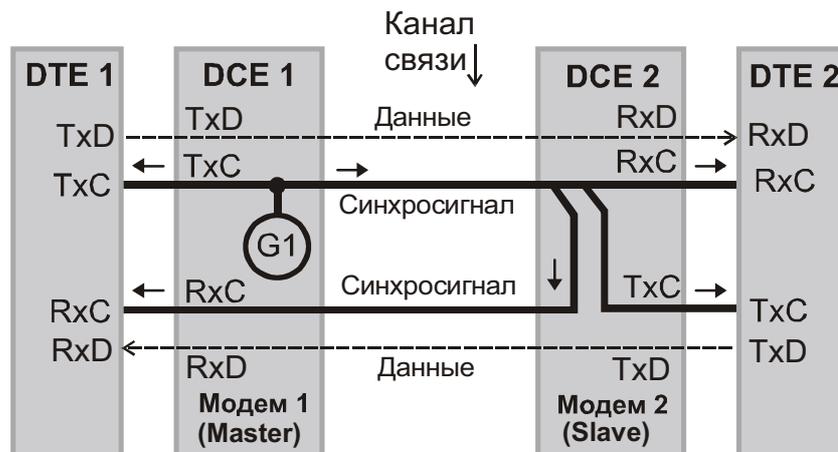


Рис. 2.2. Система с внутренней синхронизацией – второй вариант

Отметим, что сигналы TxC и RxC на входах устройства $DTE 1$ имеют одинаковую частоту, но взаимно сдвинуты по фазе на некоторый заранее неизвестный угол (но это не нарушает работоспособности данной системы). Поэтому передаваемые и прини-

маемые данные принадлежат взаимно сдвинутым по фазе синхросеткам, по которым определяются границы и центры битовых интервалов. Например, положительный фронт сигнала ТхС может быть близок отрицательному фронту сигнала RxС (сдвиг близок 180 град.) и т. п. Сдвиг вызван существенным различием задержек распространения сигнала от одного и того же генератора G1 до входов устройства DTE 1 по короткому и длинному путям. Кроме того, качество сигнала после его прохождения по длинному пути может в той или иной степени снизиться (подробности - см. гл. 5).

Следует отметить, что для нормальной работы некоторых устройств (мультиплексов и т. п.) подобный фазовый сдвиг между синхросетками передаваемых и принимаемых данных недопустим.

2.2. Системы с внешней синхронизацией

В схеме на рис. 2.3 опорный генератор размещен в оконечном устройстве типа DCE. Под действием синхросигнала с этого генератора данные передаются из устройства DCE 1 в модем 1, временно запоминаются в нем и затем в смеси с синхросигналом поступают в канал связи. Модем 2 и оконечное устройство типа DTE работают в тех же режимах, что и в схеме, приведенной на рис. 2.2. Обратная передача данных из модема 1 в устройство DCE 1 сопровождается выделенным из канала сигналом RxС, который при поступлении в это устройство трактуется как сигнал CLK.

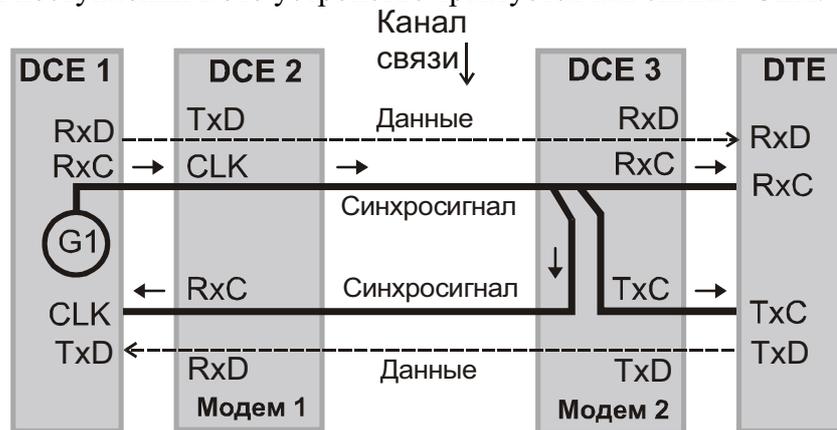


рис. 2.3. Система с внешней синхронизацией – первый вариант

Схема на рис. 2.4 отличается от приведенной на рис. 2.3 двумя существенными особенностями.

Во-первых, генератор G1 формирует синхронные и синфазные сигналы RxС и ТхС. Они, как и положено, соответственно сопровождают “свои” выходные данные и запрашивают “чужие”, поступающие на вход устройства DCE 1. Таким образом, модем 1 при взаимодействии с устройством DCE 1 должен выдавать и принимать данные, размещенные в одной и той же синхросетке.

Во-вторых, модем 1 (типа Зелакс М-64) дополнительно выполняет функцию привязки принимаемых из канала данных к исходному синхросигналу от генератора G1. Это осуществляется следующим образом. Модем 1, как и в предыдущих примерах, выделяет из канального сигнала синхроимпульсы и данные. Под действием этих синхроимпульсов принятые из канала данные временно запоминаются в буферной памяти (на рисунке показана в виде черного квадрата). В отличие от описанного ранее сквозного прохождения синхросигнала через модем 1, дальнейшее распространение выделенных из канала синхроимпульсов прекращается, что условно отражено на схеме “крестиком”.

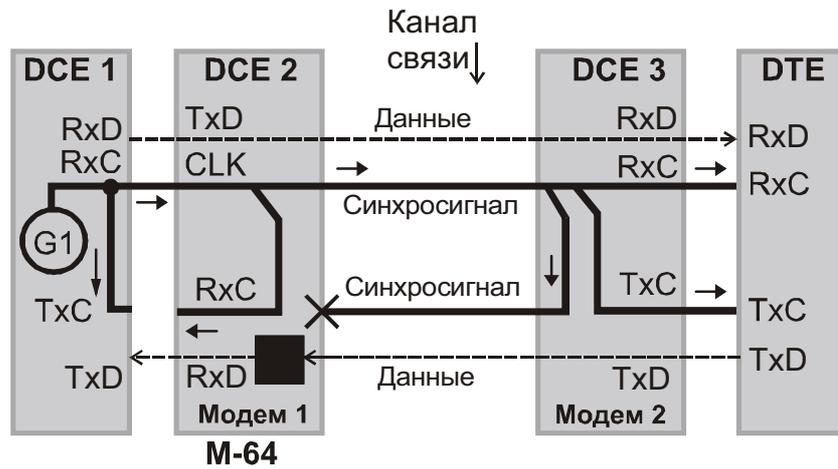


рис. 2.4. Система с внешней синхронизацией – второй вариант

Данные считываются из буферной памяти под управлением сигналов от генератора G1, поступающих по цепи G1 – RxC – CLK. Эти данные сопровождаются “своим” сигналом RxC, который, однако, не передается в устройство DCE 1, так как в кабеле DCE 1 – DCE 2 нет соответствующего провода.

В результате имеем следующий сценарий обмена данными. Устройство DCE 1 сопровождает свои выходные данные сигналом RxC. Модем 1 принимает этот сигнал на вход CLK и под его управлением временно запоминает входные данные и пересылает их в канал. В то же время устройство DCE 1 запрашивает данные от модема 1 сигналом TxС, совпадающим с RxC. Оно уверено в привязке поступающих от модема 1 данных к сигналу TxС. И эта уверенность оправдывается благодаря правильному выбору режима синхронизации буферной памяти.

2.3. Использование модема как устройства типа DTE

Пара модемов Зелакс М-144 может работать с каналом связи в различных режимах. При этом, в частности, один из модемов может рассматриваться как устройство типа DTE. Мы ознакомимся только с одним из таких режимов (рис. 2.5).

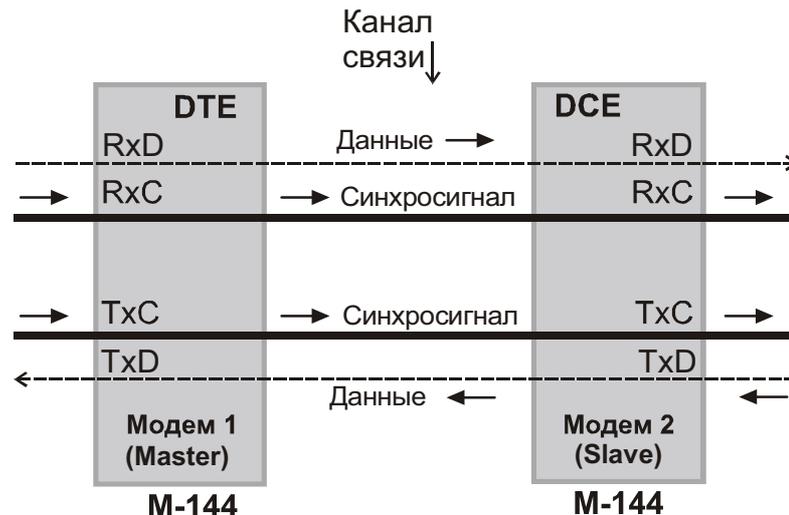


рис. 2.5. Вариант подключения модемов М-144 к каналу связи

Как следует из схемы, модемы 1 и 2 выполняют соответственно функции устройств типа DTE и DCE. При этом обеспечивается параллельное и сквозное прохождение сигналов синхронизации “слева – направо” от двух внешних источников синхронизации, в общем случае независимых. Неизбежно возникает вопрос: может ли синхросиг-

нал ТхС распространяться в канале навстречу “своим” же данным (т. е. как бы “против течения”)? Может, но описание подробностей такого распространения выходит за рамки настоящей главы (см. п. 3.5.2). Примем пока этот факт “на веру”.

2.4. Система с двумя последовательно включенными каналами связи

Чтобы получить более общее представление о синхронизации систем, использующих приведенные решения, рассмотрим пример такой системы (рис. 2.6). В ней применены две рассмотренные ранее схемы (см. рис. 2.1 и рис. 2.5). Последняя использована в качестве удлинителя для передачи сигналов RxC и ТхС к удаленному оконечному устройству DTE 3. Пару модемов 2 и 3, расположенных недалеко друг от друга, можно рассматривать как ретранслятор.

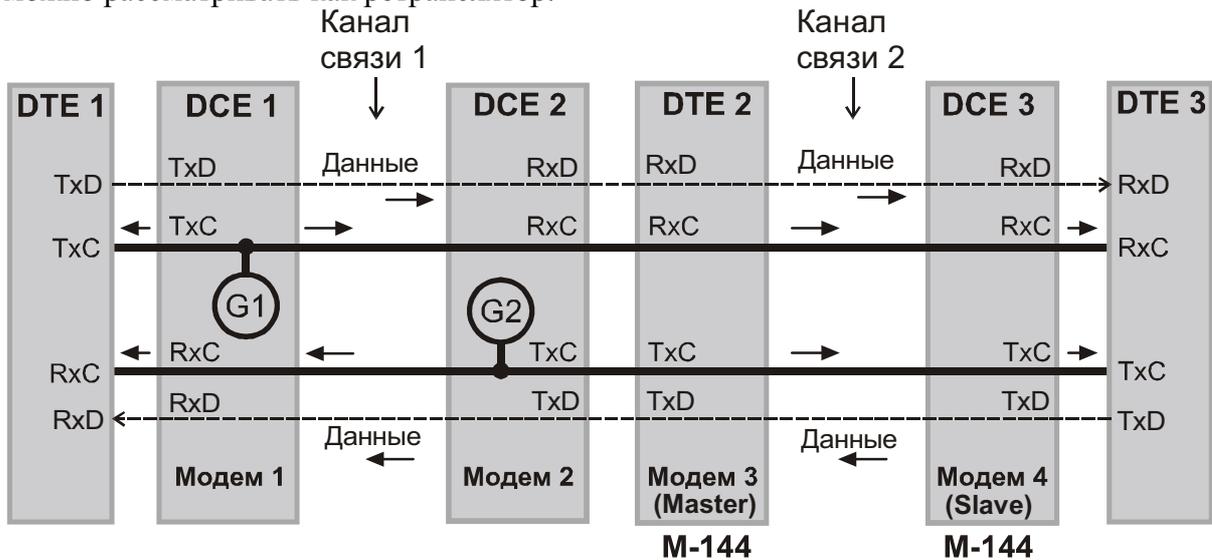


Рис. 2.6. Система с двумя последовательно включенными каналами связи

Отметим, что при проектировании систем с внутренней и внешней синхронизацией необходимо следить за тем, чтобы в них не было замкнутых контуров распространения синхросигнала, не связанных с опорным генератором. В приведенных схемах таких контуров нет. Но в более сложных системах, состоящих из десятков или сотен синхронных устройств, предотвратить возникновение таких контуров бывает непросто (способы предотвращения закливания синхросигналов описаны в гл. 6).

3. Взаимодействие устройств типа DTE/DCE в синхронном режиме: нестандартные решения

Рассмотренные в этой главе нестандартные решения позволяют улучшить характеристики существующих систем передачи данных. Приведем краткий обзор этих решений.

В схеме передачи данных, рассмотренной в п. 3.1, байты или иные группы битов разграничены стартовыми и стоповыми битами, и в то же время применен синхросигнал. В результате получена более надежная временная диаграмма передачи данных, с явным указанием границ и центров битовых интервалов.

В следующей схеме (п. 3.2) одна и та же линия используется как для передачи синхросигнала, так и для передачи временных меток для разграничения информационных кадров. Это позволяет получить простую и надежную систему передачи данных с битовой и кадровой синхронизацией.

Группа решений по п. 3.3 основана на усовершенствовании классического способа передачи данных с использованием сигналов TxD и CLK. Как уже отмечалось, здесь модернизируется простейшая схема, считавшаяся безупречной на протяжении десятилетий! В результате расширены функциональные возможности систем передачи данных и повышена скорость обмена данными.

Классическое аппаратное управление потоком данных между устройствами типа DTE и DCE подразумевает обмен сигналами RTS – CTS (см. п. 1.2). Как показано в п. 3.4, можно исключить эти сигналы и упростить схемы синхронизации, если вместо непрерывного потока синхросигналов использовать пачки импульсов.

Решения, рассмотренные в п. 3.5, позволяют дистанционно регулировать фазу сигнала от удаленного синхрогенератора. Это, в частности, дает возможность передавать синхросигналы “против течения” потока данных, что на первый взгляд представляется невозможным! Показано использование такой передачи при построении удлинителя несинхронных сигналов TxS и RxS.

В п. 3.6 рассмотрена экономичная по аппаратуре система с непосредственной передачей синхросигнала по каналу связи. Она применима, в частности, для дуплексной передачи данных с синхронизацией от общего источника синхросигналов TxS и RxS.

Задача распознавания начала сообщения в асинхронном потоке данных рассмотрена в п. 3.7. Обычно сообщения разделяются паузами, но они могут возникнуть в результате “замираний” сигнала или вследствие действия помех. Чтобы отличить истинную разделительную паузу от ложной, в предлагаемом решении применяется простой способ кодирования символов.

При передаче непрерывного асинхронного потока данных служебные биты Старт и Стоп “прячутся” среди информационных битов. Как в этом случае передатчик может “помочь” приемнику в отыскании служебных битов? Ответ на этот вопрос можно найти в п. 3.8. В заключение главы (п. 3.9) рассмотрены решения задачи распознавания межбайтовых границ в непрерывном синхронном потоке данных.

3.1. Синхронная передача данных между устройствами типа DTE и DCE с использованием стартовых и стоповых битов

В рассмотренном далее решении применена комбинация идей синхронного и асинхронного обмена данными. Далее для упрощения изложения будет рассмотрен только односторонний обмен данными, но все сказанное применимо также и к передаче данных во встречном направлении. Напомним, в чем заключается асинхронный обмен (рис. 3.1).

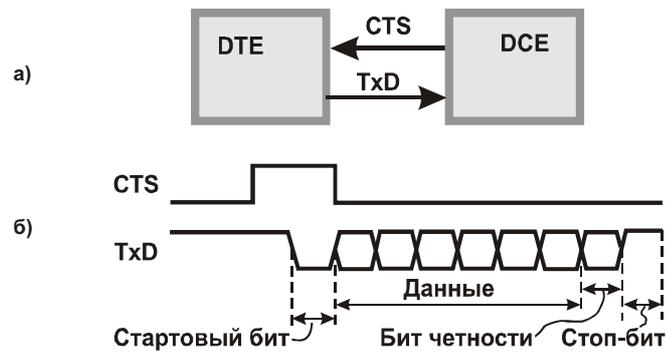


Рис. 3.1. Передача данных в асинхронном режиме:
а – типовая схема; *б* – временная диаграмма

При обнаружении сигнала готовности устройства DCE к приему данных (положительного напряжения в линии CTS) устройство DTE, если это необходимо, начинает передачу асинхронной посылки. Она состоит из стартового бита, семи или восьми битов данных (для упрощения рисунка показаны только шесть битов) и может сопровождаться битом контроля на четность (или нечетность; все параметры посылки, конечно, однозначно задаются до проведения сеанса связи). И, наконец, в завершение передаются один или два стоп-бита, которые служат разделителями посылок, если они следуют непрерывным потоком.

Устройство DCE обнаруживает отрицательный фронт сигнала TxD (начало стартового бита) и начинает отсчет времени с целью определения моментов, соответствующих серединам битовых интервалов. В эти моменты данные записываются в приемный регистр устройства DCE.

Надежность такого способа обмена снижается по мере повышения скорости передачи данных. Импульсы теряют прямоугольную форму из-за того, что битовый интервал становится соизмеримым с длительностью фронта сигнала. Точность построения синхросетки устройством DCE снижается, так как фронт стартового бита становится все более пологим; кроме того, становится ощутимым джиттер (дрожание фазы – см. гл. 5) сигнала TxD.

Синхронный обмен обеспечивает более высокую скорость передачи данных. Это достигается как за счет исключения стартовых и стоп-битов, так и благодаря явному (а не подразумеваемому) подтверждению истинности каждого передаваемого бита данных сигналом CLK на дополнительно введенной линии синхронизации. Однако при синхронном обмене необходима упаковка данных в кадры определенного формата, чтобы устройство DCE могло иметь ориентиры в непрерывном входном потоке битов. Это устройство распознаёт границы и распаковывает кадры. Для одних приложений наиболее целесообразно использовать синхронную передачу данных, для других – асинхронную.

Предлагаемое в [13] решение (рис. 3.2) позволяет повысить надежность асинхронного обмена при высоких скоростях передачи данных. Это достигается благодаря тому, что, в отличие от классического асинхронного способа передачи данных, истинность передаваемых битов, как и при синхронной передаче, подтверждается отрицательными фронтами синхросигнала CLK, а границы битовых интервалов соответствуют положительным фронтам этого сигнала.

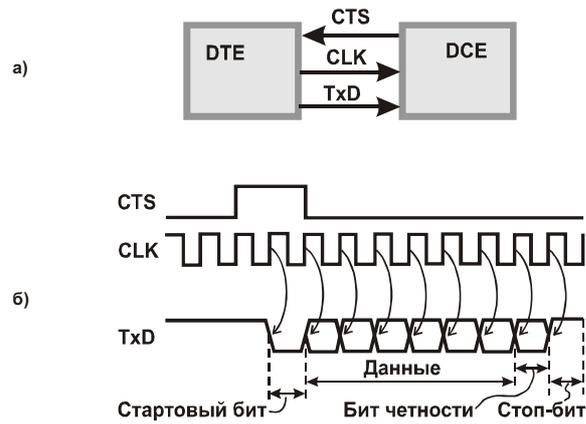


рис. 3.2. Передача данных в синхронном режиме: *а* – типовая схема; *б* – временная диаграмма, сочетающая принципы синхронного и асинхронного обмена

В исходном состоянии на вход устройства DCE поступает поток лог. 1, что соответствует отсутствию полезных данных. При наличии готовности принять данные устройство DCE устанавливает положительное напряжение в линии CTS (в данном примере имеется готовность принять только одну посылку). Устройство DTE, обнаружив готовность, формирует и выдает посылку, размещая ее в синхросетке сигнала CLK.

Устройство DCE расценивает первый принятый нулевой бит как стартовый и снимает сигнал готовности приема (это касается только данного примера – сигнал готовности может сохраняться в течение любого нужного периода). Стартовый и последующие биты принимаются по отрицательным фронтам синхросигнала CLK. Таким образом, сохраняется асинхронный способ передачи данных, при котором посылки передаются с произвольным темпом (вплотную или с любыми интервалами). В то же время внутри каждой посылки обеспечивается жесткая привязка битов к синхросигналу, что повышает надежность приема.

Описанный способ обмена данными применим также к комбинациям устройств DTE – DTE и DCE – DCE.

3.2. Синхронный обмен данными с передачей кадровых меток

Неструктурированный поток данных (без начала, конца и каких-либо промежуточных “опознавательных знаков”) может представлять ограниченный интерес и используется разве что при отладке или поиске неисправностей системы на физическом уровне. При работе транспортных систем передаваемые данные объединяются в логически законченные структурные единицы: кадры, пакеты и проч. Эти единицы (для определенности – кадры) могут значительно различаться по форме и содержанию в зависимости от принятой технологии передачи данных, протоколов обмена и т. д. Однако их объединяет то, что в явном или неявном виде должны быть заданы границы, по которым приемник может выделить кадр из потока.

В простейшем случае задача разграничения кадров решается аппаратно: между передатчиком и приемником вводится специальная линия, по которой передаются импульсы – кадровые метки. В примере, приведенном на рис. 3.3, передатчик взаимодействует с приемником в синхронном режиме и посылает ему кадровые метки FR для обозначения межкадровых границ. Границы битовых интервалов соответствуют положительным фронтам синхросигнала CL. Прием данных происходит по отрицательным фронтам этого синхросигнала.

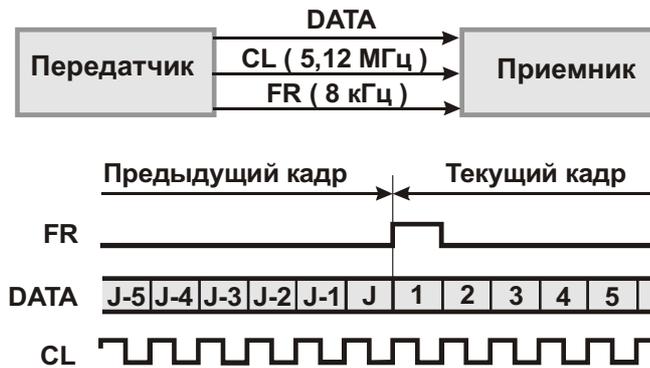


рис. 3.3. Раздельная передача данных DATA, синхросигнала CL и кадровых меток FR

Кадры следуют с частотой 8 кГц. Каждый кадр содержит J битов; в данном примере $J = 640$. Скорость передачи данных составляет $640 \times 8 = 5120$ бит/с. Кадровая метка совпадает во времени с первым битовым интервалом кадра.

Чтобы уменьшить число линий канала связи, сохраняя передачу кадровых меток, применяют разные методы кодирования. Например, используют только одну линию, по которой передается манчестерский код (см. п. 8.1.4). В этом коде сигналы DATA и CL перед выдачей в линию суммируются по модулю два. Кадровые метки отображаются моментами преднамеренного нарушения правил кодирования [61].

В схеме, приведенной на рис. 3.4 [60], число линий канала связи уменьшено до двух.

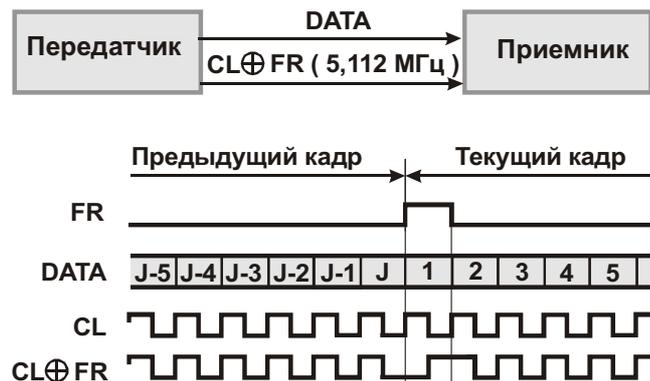


рис. 3.4. Передача данных DATA и суммарного синхросигнала $CL \oplus FR$

Сигнал общей синхронизации $CL \oplus FR$ получен суммированием по модулю 2 битовых CL и кадровых FR синхроимпульсов. Из-за периодического прореживания синхроимпульсов частота сигнала $CL \oplus FR$ на 8 кГц ниже частоты сигнала CL и составляет 5,112 МГц.

Схема восстановления синхросигналов и временные диаграммы приведены на рис. 3.5. Схема содержит генератор синхросигнала частотой 184,32 МГц, в 36 раз превышающей частоту сигнала CL передатчика. Стабильность генератора обеспечивается петлей фазовой автоподстройки частоты. Эта петля предназначена для привязки формируемых сигналов к входному синхросигналу, точнее, к его основной составляющей CL частотой 5,12 МГц. Фазовый компаратор следит за совпадением фронтов входных сигналов. При обнаружении более или менее устойчивого разбаланса фронтов корректируется напряжение U управления частотой генератора в направлении улучшения совпадения. Инерционность петли управления определяется частотой среза фильтра низких частот. Благодаря инерционности, устройство не реагирует на кратковременные помехи и пропадания входного сигнала. Обнаружение начала кадра сопровождается появлением сигнала $FRAME = 1$.

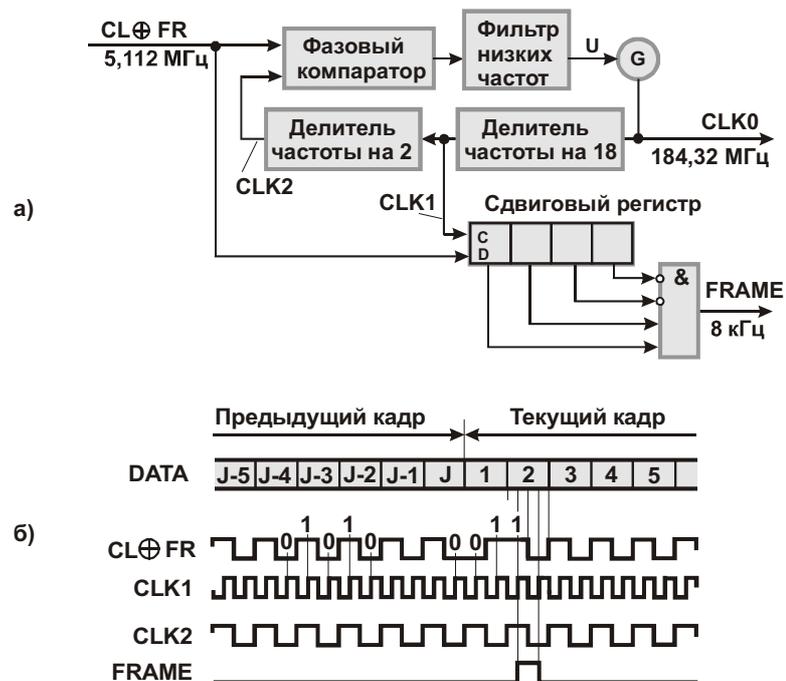


рис. 3.5. Восстановление синхросигналов в приемнике: а – схемное решение; б – временные диаграммы

Сигнал $CL \oplus FR$ непрерывно анализируется схемой на основе сдвигового регистра и элемента И. Сдвиговый регистр продвигает информацию слева направо по положительным фронтам сигнала CLK1. Элемент И содержит по два неинвертирующих и инвертирующих входа. Как следует из временных диаграмм, при передаче битов, далеких от границ кадров, в сдвиговом регистре присутствует код 0101 или 1010. При этом на выходе элемента И сформирован сигнал $FRAME = 0$. (Возможными кратковременными всплесками сигнала FRAME в моменты переходных процессов в сдвиговом регистре пренебрегаем, так как опрос этого сигнала производится последующими схемами, например по отрицательному фронту сигнала CLK1, когда переходные процессы отсутствуют.)

При прохождении межкадровой границы в сдвиговом регистре присутствует код 1100, на выходе элемента И формируется сигнал $FRAME = 1$. Длительность этого сигнала равна половине битового интервала, но его положение, как следует из рис. 3.5, б, не вполне соответствует межкадровой границе. Учесть соответствующую поправку несложно, тем более, что в нашем распоряжении имеется ряд опорных частот, вплоть до частоты, в 36 раз превышающей скорость передачи данных. Наличие столь мелкой временной сетки позволяет вести подробную “прорисовку” сигналов, адаптироваться к их искажениям, выделять помехи, выбирать наиболее надежные точки опроса данных и т. п.

3.3. Повышение быстродействия и расширение функциональных возможностей схемы с попутной синхронизацией

В гл. 1 были рассмотрены два способа синхронного обмена данными между устройствами. Рассмотренные далее усовершенствования касаются первого способа, который предусматривает попутную передачу двух сигналов TxD и CLK (соответственно сигнала данных и синхросигнала), что поясняется рис. 3.6.

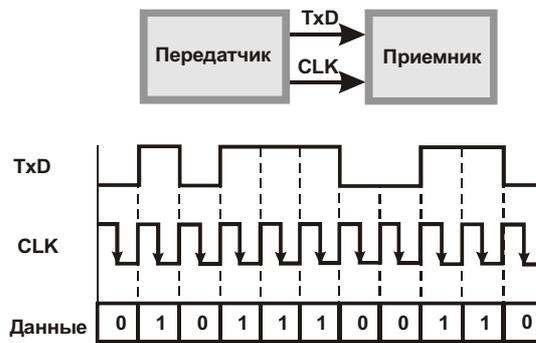


Рис. 3.6. Классическая временная диаграмма передачи данных в синхронном режиме

Положительные фронты синхросигнала CLK задают границы между битовыми интервалами; эти интервалы обозначены на рисунке штриховыми линиями. Данные принимаются по отрицательным фронтам синхросигнала.

Эта схема успешно применяется не один десяток лет, и лишь сравнительно недавно (в 2000 и 2001 годах) предложены рассмотренные далее варианты ее кардинального улучшения [11], [63].

3.3.1. Удвоение скорости передачи данных с использованием положительного и отрицательного фронтов сигнала CLK для их приема

В первом варианте усовершенствованной схемы для приема данных используются как положительные, так и отрицательные фронты синхросигнала CLK (рис. 3.7) [63]. Сравнивая временные диаграммы, приведенные на рис. 3.6 и рис. 3.7, можно отметить, что при одинаковой скорости передачи данных частота сигнала CLK в усовершенствованной схеме уменьшается вдвое. Можно дать и иную трактовку тех же нововведений: при неизменной частоте сигнала CLK скорость передачи данных удваивается.

Разумеется, оба фронта сигнала CLK должны соответствовать установленным значениям сигнала TxD, точнее, должны обеспечиваться достаточные интервалы предустановки и удержания сигнала TxD соответственно до и после прохождения фронта сигнала CLK. Эти параметры определяются техническими данными применяемой элементной базы и параметрами линии связи.

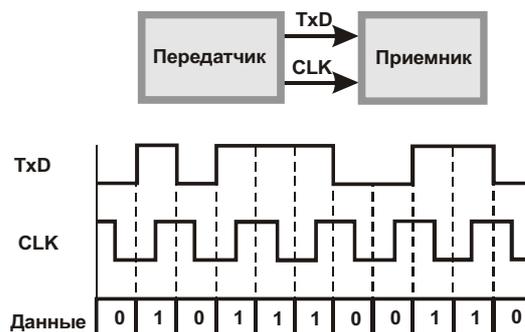


Рис. 3.7. Усовершенствованная временная диаграмма передачи данных в синхронном режиме – первый вариант

Примечательно, что усовершенствованная схема совместима с традиционным оборудованием приема данных, работающим в соответствии с предыдущей временной диаграммой. Действительно, теперь поток передаваемых данных содержит две составляющие, одна из которых удовлетворяет общепринятой временной диаграмме (см. рис. 3.6), а вторая представляет собой некое дополнение. Это дополнение просто не воспринимается традиционным оборудованием. Поэтому традиционная и новая аппаратура

приема данных могут включаться параллельно. Однако передатчик должен знать, с каким приемником ему предстоит работать в данном сеансе связи.

Рассмотрим пример воплощения рассмотренной идеи. В схеме передачи данных, показанной на рис. 3.8, параллельные коды DATA 1 и DATA 2 могут относиться к одному или двум независимым информационным потокам. Эти потоки поступают в передатчик от внешнего источника (на рисунке не показан) и суммируются в линии TxD. Первый воспринимается приемником по отрицательным фронтам сигнала CLK, второй – по его положительным фронтам.

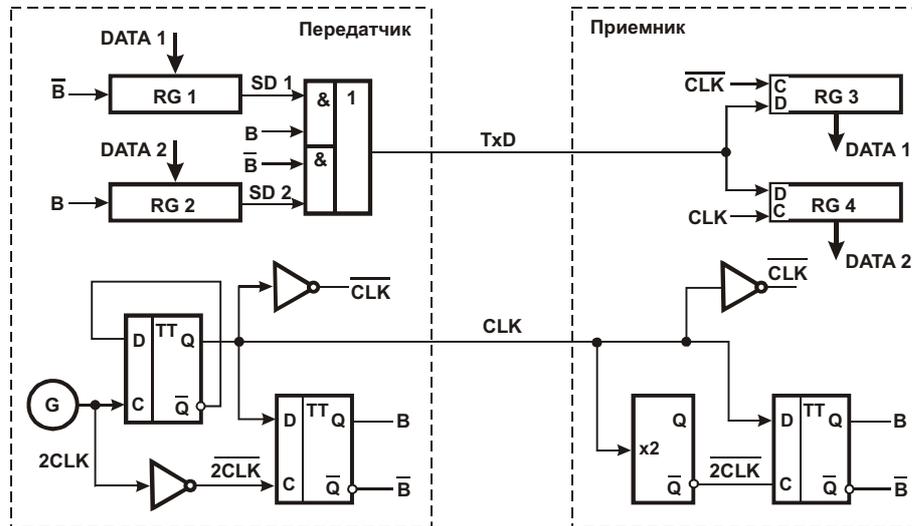


рис. 3.8. Схема передачи данных. Приемник получает биты данных TxD как по положительным, так и по отрицательным фронтам сигнала CLK

Передатчик содержит генератор G, сдвиговые регистры RG 1 и RG 2, элемент 2И-ИЛИ, два D-триггера и два инвертора. Приемник содержит сдвиговые регистры RG 3 и RG 4, D-триггер, инвертор и удвоитель "x2" частоты с инверсным выходом.

Генератор G формирует непрерывную последовательность синхроимпульсов 2CLK с частотой, равной скорости передачи данных TxD по линии (рис. 3.9).

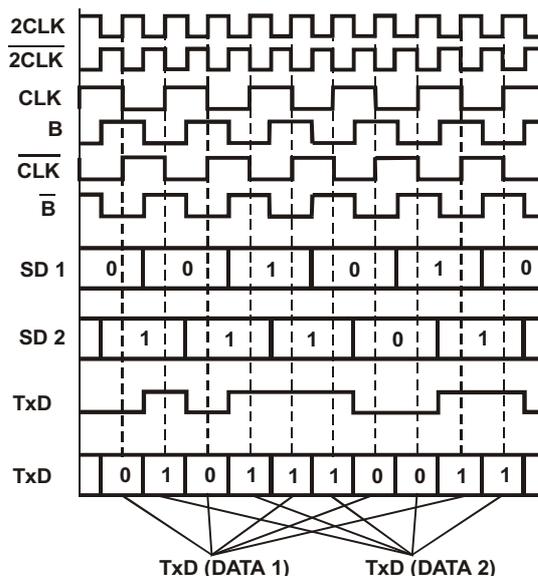


рис. 3.9. Временные диаграммы передачи данных в схеме, показанной на рис. 3.8

Сигнал с генератора G преобразуется в четыре синхросигнала с последовательным фазовым смещением на четверть периода. Данные DATA 1 и DATA 2 периодически заносятся в регистры RG 1, RG 2 и затем сдвигаются в них по положительным фронтам соответствующих управляющих сигналов (цепи управления параллельной записью данных в регистры RG 1 и RG 2 на рисунке не показаны).

Последовательные данные SD 1 и SD 2, выдвигаемые из регистров RG 1 и RG 2, поочередно выдаются в линию через логический элемент 2И-ИЛИ. Поэтому передаваемые данные TxD содержат биты TxD (DATA 1) и TxD (DATA 2), принадлежащие двум разным или одному общему информационным потокам.

Приемник последовательно загружает принимаемые данные в сдвиговые регистры RG 3 и RG 4 с противофазной синхронизацией. При этом информация распределяется по двум первоначальным направлениям. На выходах регистров в соответствующие моменты формируются параллельные данные DATA 1 и DATA 2. В эти моменты они считываются последующими устройствами (эти устройства и схемы управления считыванием на рисунке не показаны).

Удвоитель "x2" частоты сигнала CLK и D-триггер позволяют получить в приемнике полный комплект аналогов синхросигналов передатчика, если в этом есть необходимость.

3.3.2. Удвоение скорости передачи данных заменой сигнала CLK сигналом разграничения одноименных битов

Рассмотрим второй вариант усовершенствования классической схемы передачи данных. Как показано на рис. 3.10, вместо периодического сигнала CLK (см. рис. 3.6) применен сигнал разграничения DELIM, который изменяется в тех случаях, когда передаваемые данные TxD неизменны и остается неизменным, когда данные изменяются [11]. Граница очередного битового интервала определяется приемником по факту изменения состояния пары принимаемых сигналов (TxD и DELIM).

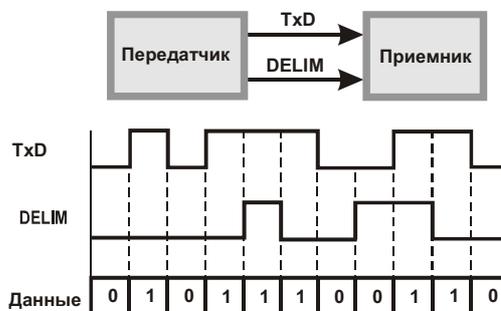


Рис. 3.10. Усовершенствованная временная диаграмма передачи данных – второй вариант

Сравнивая временные диаграммы, приведенные на рис. 3.6 и рис. 3.10, можно отметить следующее.

Во-первых, передаваемые данные формируются с одинаковым разбиением сигнала TxD на битовые интервалы. Во-вторых, сигнал DELIM в усовершенствованной схеме претерпевает всего лишь четыре изменения, в то время как сигнал CLK изменяется 21 раз. В течение одного битового интервала сигнал CLK изменяется дважды, а сигнал DELIM – не более одного раза.

Таким образом, при сохранении физических параметров системы (длины кабеля, элементной базы) можно удвоить скорость передачи данных или при той же скорости передачи, что и в традиционной схеме, вдвое уменьшить максимально возможную частоту сигнала, сопровождающего данные. Кроме того, при передаче случайных данных сигнал DELIM, в отличие от сигнала CLK, не содержит ярко выраженных спектраль-

ных составляющих, что способствует снижению уровня излучаемых в линию помех. (Для преобразования данных к виду, близкому случайной последовательности битов, используют скремблеры, см. п. 8.4.)

3.3.3. Расширение функциональных возможностей системы с разграничением одноименных битов

Идея построения дополнительного канала связи

В схеме, показанной на рис. 3.10, данные транслируются по одной линии, в то время как вторая используется только для восполнения “пробелов” синхронизации. Нельзя ли и по ней передавать данные? Возможность такой передачи подтверждается рис. 3.11.

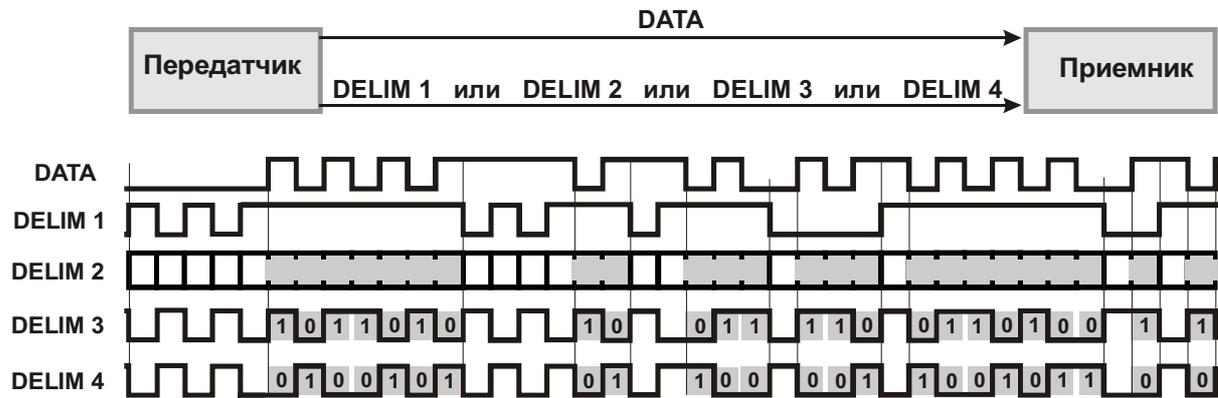


рис. 3.11. Временные диаграммы передачи данных с использованием сигналов DATA и разных вариантов построения сигнала DELIM: DELIM 1 и DELIM 2 – классический и обобщенный варианты; DELIM 3 и DELIM 4 – варианты передачи взаимно-обратных кодов

Сигнал DELIM 1 сформирован по рассмотренным ранее правилам, но можно их нарушить, увеличив информационную нагрузку на этот сигнал. Как показано в общем виде на диаграмме сигнала DELIM 2, существуют области, отмеченные серым фоном, в которых возможна любая последовательность сигналов лог. 0 и лог.1. Обязательные фронты сигнала DELIM 2 показаны в явном виде, необязательные – в форме зубцов. Отметим, что каждая затененная область начинается необязательным и заканчивается обязательным фронтом. Затененным областям соответствуют периоды “интенсивных” изменений сигнала DATA, поэтому эти изменения гарантируют распознавание битовых интервалов приемником независимо от состояния сигнала разграничения.

Временные диаграммы сигналов DELIM 3 и DELIM 4 отображают передачу кода 10110...1 и обратного ему 01001...0. В данном примере начальные участки диаграмм одинаковы, но в дальнейшем сигналы становятся противофазными.

Как следует из диаграмм, сигналы DELIM 1 – DELIM 4 в течение 15 битовых интервалов выполняют чисто административные функции, а в течение 24 битовых интервалов (отмеченных серым фоном) могут нести “полезную” информацию. По существу, сформирован дополнительный канал связи. Данные представлены двумя параллельными потоками – основным и дополнительным.

В приведенном примере данные имеют случайный характер; но если предположить, что сигнал DATA не изменяется, то сигналы DELIM 1 – DELIM 4 будут вынуждены изменяться в каждом такте, и дополнительный канал связи исчезнет.

Чтобы этого не произошло, поток данных DATA перед выдачей в линию скремблируется, т. е. определенным образом шифруется (см. п. 8.4). Шифрация заключается, например, в прогоне данных через сдвиговый регистр с элементами Иключающее ИЛИ в цепях обратной связи. При этом преобразование оказывается настолько запутанным, что любой входной поток данных становится на выходе псевдослучайным. Дешифрация (дескремблирование) потока выполняется приемником с помощью подобного сдвигового регистра с логическими элементами Иключающее ИЛИ. Далее предполагаем, что данные в линии связи скремблированы.

При равновероятном появлении сигналов лог. 0 и лог. 1 в потоке данных DATA пропускная способность дополнительного канала передачи данных составляет 50% пропускной способности основного канала. Действительно, вероятность перепада уровней сигнала DATA на границе битовых интервалов i и j равна 0,5 и не зависит от предыстории. Но именно эта вероятность определяет событие “вставки” очередного бита данных в поток сигналов разграничения битов. Таким образом, в среднем каждый второй битовый интервал пригоден для передачи дополнительного бита данных. В результате общая пропускная способность (или скорость передачи данных) системы по сравнению с прототипом (см. рис. 3.10) увеличена в 1,5 раза.

Рассмотренный способ создания дополнительного канала применим и к параллельной синхронной передаче данных по нескольким линиям. В схеме, показанной на рис. 3.12, данные передаются по трем линиям с общим сигналом DELIM разграничения битовых интервалов.

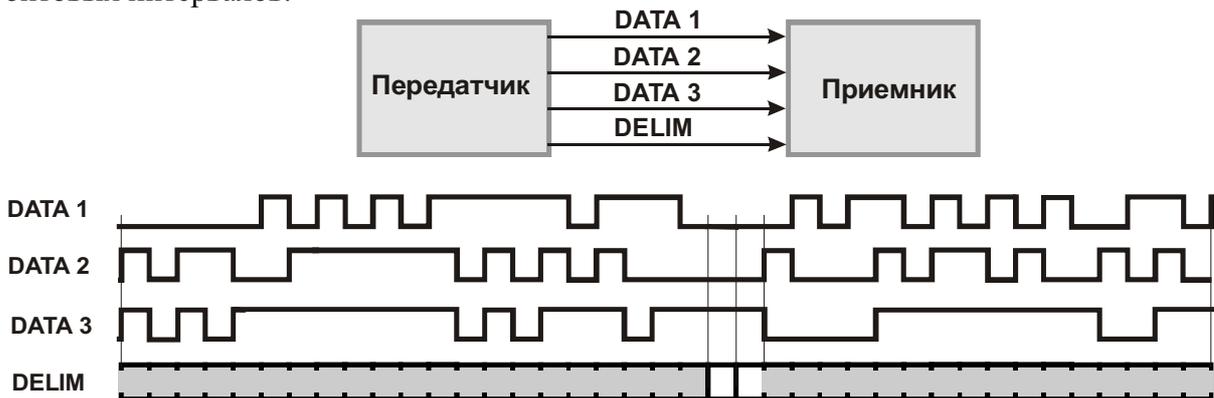


Рис. 3.12. Временные диаграммы передачи данных с использованием сигналов DATA 1 – DATA 3 и сигнала DELIM, показанного в обобщенном виде

Перепад сигнала в любой из четырех линий воспринимается приемником как момент разграничения битовых интервалов. Сигнал DELIM используется по прямому назначению только при одновременном отсутствии динамики сигналов DATA 1 – DATA 3. Вероятность таких событий невысока, поэтому затененная на рисунке область произвольных изменений сигнала DELIM более обширна по сравнению с аналогичной областью из предыдущего примера. Пропускная способность дополнительного канала связи приблизилась к пропускной способности каждого из трех основных каналов.

Для помехоустойчивого восстановления синхросигнала в приемнике может применяться генератор с фазовой автоподстройкой частоты. Примеры построения таких генераторов рассмотрены в гл. 9.

Принятые из четырех линий потоки данных записываются в память приемника и анализируются микропроцессором для разделения служебных и полезных битов сигнала DELIM. Такое разделение проводится по тому же алгоритму, который использовался при разметке приведенной на рисунке диаграммы этого сигнала. (При определенных комбинациях битов данных фрагменты диаграммы помечаются затененными областями и т. д.)

Пример схемной реализации системы с основным и дополнительными каналами связи

Рассмотрим пример воплощения описанной идеи построения дополнительного канала связи. Система передачи данных, показанная на рис. 3.13, в неявном виде содержит два канала передачи данных: основной DTE1 → DTE3 и дополнительный DTE2 → DTE4. Для передачи данных в обратном направлении необходима вторая пара устройств передатчик – приемник, включенная встречно.

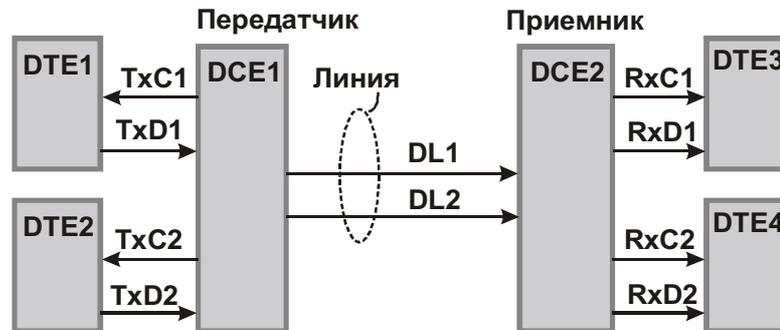


рис. 3.13. Система передачи данных с основным и дополнительными каналами связи

Пара сигналов TxС1 – TxD1 (TxС2 – TxD2) обеспечивает синхронную передачу данных из устройства DTE1 (DTE2) в передатчик DCE1. Положительные фронты сигнала TxС1 (TxС2) задают границы битовых интервалов. По отрицательным фронтам сигнала TxС1 (TxС2) соответствующий бит данных TxD1 (TxD2) временно запоминается в передатчике.

Пара сигналов RxC1 – RxD1 (RxC2 – RxD2) обеспечивает синхронную передачу данных из приемника DCE2 в устройство DTE3 (DTE4). Положительные фронты сигнала RxC1 (RxC2) задают моменты начала битовых интервалов. По отрицательным фронтам сигнала RxC1 (RxC2) соответствующие биты данных RxD1 (RxD2) запоминаются в устройстве DTE3 (DTE4).

По линии передаются сигналы DL1 и DL2 – аналоги сигналов DATA и DELIM 2, показанных на рис. 3.11. Линия может быть выполнена в виде двух витых пар проводов, двух оптоволоконных световодов или иных средств передачи сигналов, в том числе, содержащих ретрансляторы.

Все процессы, протекающие в системе, синхронизированы сигналами от генератора G, размещенного в передатчике (рис. 3.14).

Передатчик работает следующим образом. Сигнал CL1 (TxС1) с выхода генератора G передается в устройство DTE1 (рис. 3.15). В ответ на этот сигнал из устройства DTE1 выдаются данные TxD1 и поступают на вход двухразрядного сдвигового регистра RG.

Под действием положительных фронтов сигнала CL2 данные последовательно продвигаются через этот регистр, выдаются через усилитель в линию и одновременно с этим анализируются элементом Иключающее ИЛИ (сумматором по модулю два). До тех пор, пока биты в регистре RG чередуются (...010101...), формируется сигнал $V = 1$, который открывает элемент И по нижнему входу. На его выходе формируется синхросигнал TxС2, в ответ на который из устройства DTE2 поступают биты данных TxD2 дополнительного канала передачи. Эти биты проходят через мультиплексор MUX, триггер, усилитель и поступают в линию.

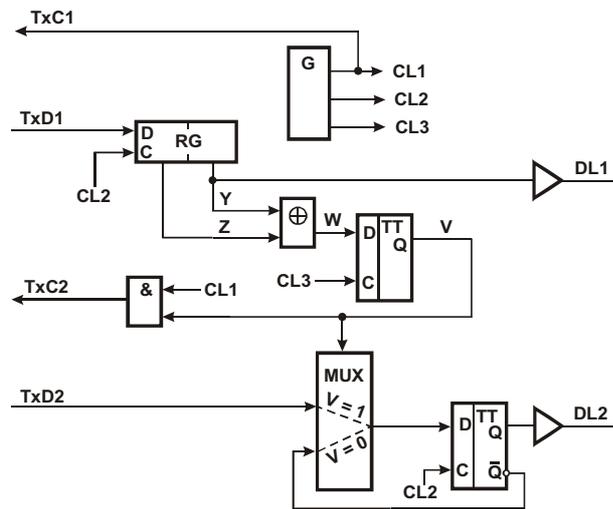


рис. 3.14. Схема передатчика

Обнаружение элементом Исключающее ИЛИ одинаковых битов в регистре RG означает невозможность передачи очередного “полезного” бита по дополнительному каналу (из-за отсутствия перепада уровней сигнала в основном канале). Поэтому работа с устройством DTE2 приостанавливается сигналом $V = 0$. Этот сигнал закрывает элемент И по нижнему входу и переключает мультиплексор MUX в состояние, при котором нижний (по схеме) триггер работает в счетном режиме, т. е. в каждом такте инвертирует хранящийся в нем бит. Таким образом, сигнал DL2 восполняет “отсутствующие” фронты сигнала DL1 (см. вертикальные стрелки на диаграмме).

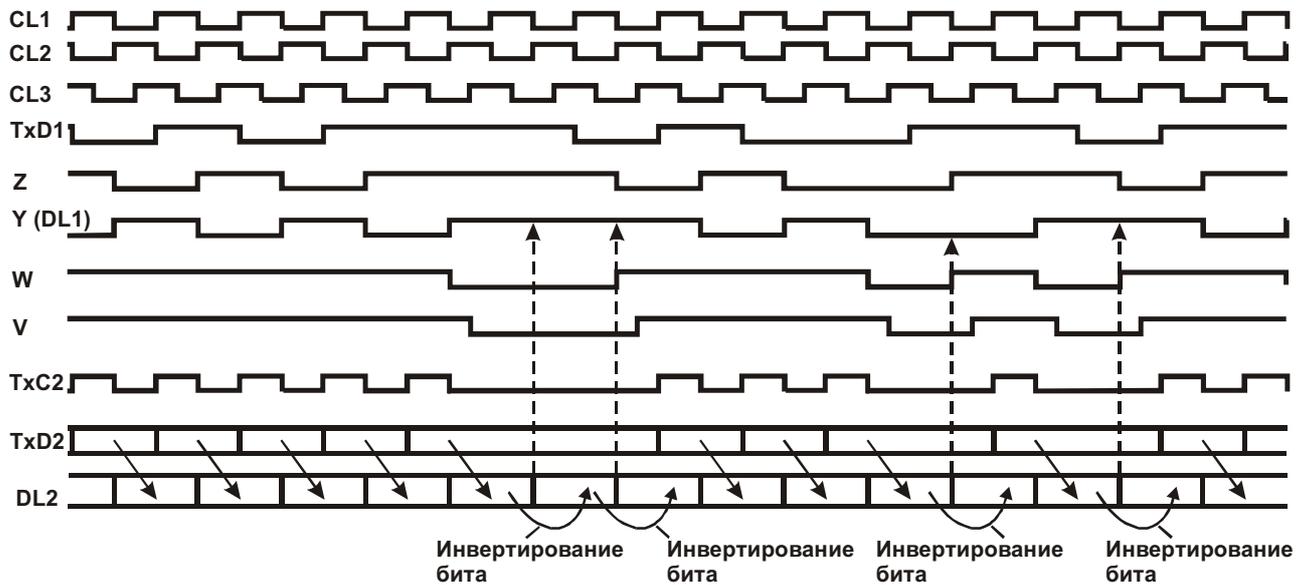


рис. 3.15. Временные диаграммы сигналов передатчика

Приемник (рис. 3.16, рис. 3.17) работает следующим образом.

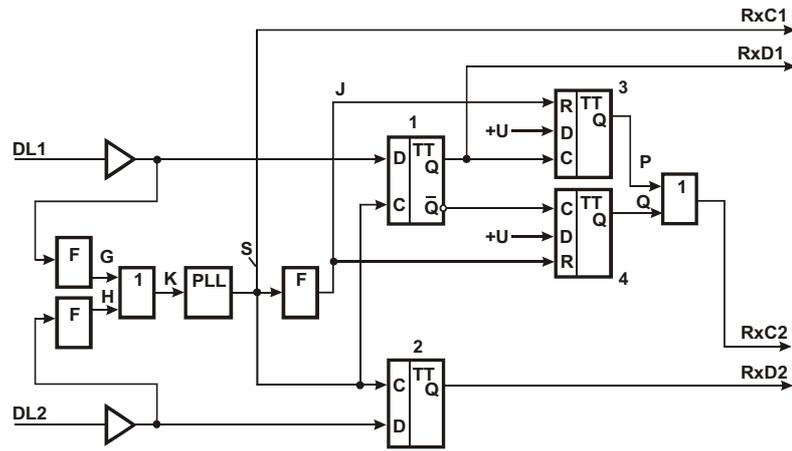


рис. 3.16. Схема приемника

Сигналы DL1 и DL2 проходят через входные усилители и формирователи F импульсов. Сигнал G сопровождает любое изменение сигнала DL1. В данном примере в сигнале G отсутствуют импульсы на позициях, соответствующих вертикальным стрелкам между двумя верхними диаграммами. Сигнал H сопровождает любое изменение сигнала DL2. Так как сигнал DL2 показан на диаграмме условно, без указания конкретного кода, то заранее не известно, будут ли сформированы импульсы, помеченные на диаграмме сигнала H крестиками. Однако можно утверждать, что остальные импульсы на этой диаграмме будут обязательно сформированы, так как они соответствуют гарантированным изменениям сигнала DL2 при его инвертировании. Таким образом, с помощью элемента ИЛИ “недостающие” импульсы восполняются, и сигнал K представляет собой периодическую последовательность импульсов без “пробелов”. Этот сигнал поступает на вход синхронизации блока PLL фазовой автоподстройки частоты.

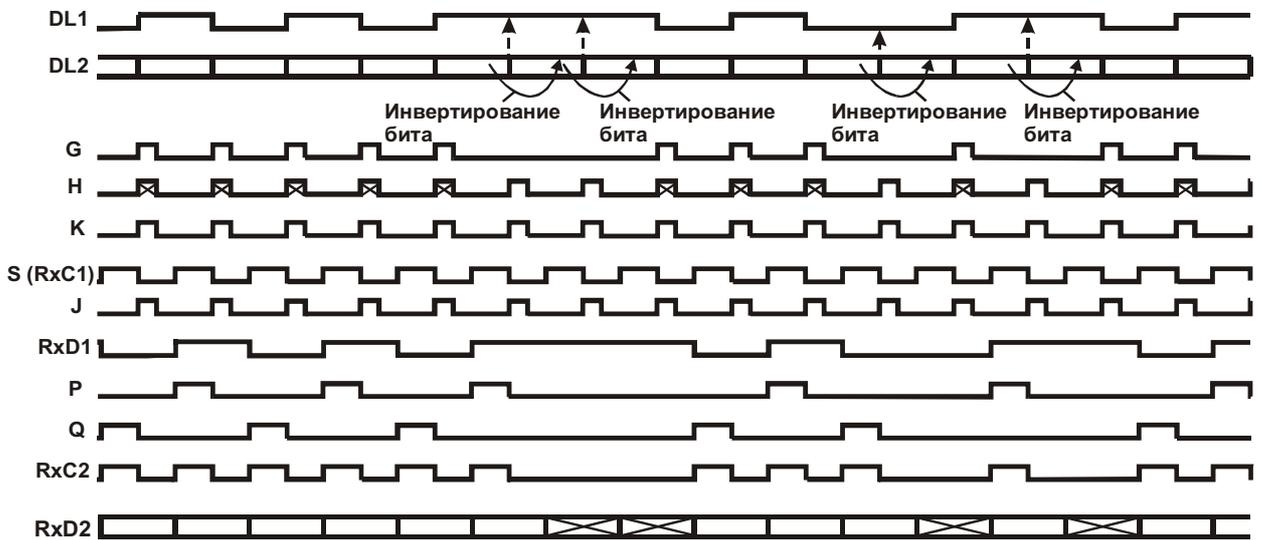


рис. 3.17. Временные диаграммы сигналов приемника

Блок фазовой автоподстройки частоты предназначен для формирования высокостабильного синхросигнала S (RxC1) на основе непрерывного слежения за входным сигналом K. В данном примере отрицательный фронт сигнала S привязан к положительному фронту сигнала K. Благодаря достаточной “инерционности” блока сигнал S практически нечувствителен к дрожанию фазы сигнала K и иным его кратковременным искажениям, вызванным помехами в линии.

По положительному фронту сигнала S принимаемые из линии данные записываются в триггеры 1 и 2. Сигналы с выходов этих триггеров поступают в устройства DTE3 и DTE4. Как отмечалось, сигнал $RxD2$ содержит не только полезные, но и служебные биты, полученные инвертированием предыдущих.

Служебные биты сигнала $RxD2$ (помеченные крестиками на нижней диаграмме), в отличие от полезных, не сопровождаются отрицательными фронтами сигнала $RxC2$ и поэтому не воспринимаются устройством DTE4. Для “прореживания” импульсов $RxC2$ используются триггеры 3, 4 и элемент ИЛИ. Триггер 3 устанавливается в единичное состояние по положительному фронту сигнала $RxD1$, а триггер 4 – по отрицательному фронту этого сигнала. Возврат этих триггеров в нулевое состояние происходит под действием импульсов J , которые поступают с формирователя импульсов и соответствуют отрицательным фронтам сигнала S . Сигналы P и Q с выходов триггеров 3 и 4 суммируются элементом ИЛИ. Поэтому в периоды стабильности сигнала $RxD1$ синхросигнал $RxC2$ не вырабатывается, что и требуется для фильтрации служебных битов.

Идея использования сигнала разграничения битов для передачи кадровых меток

Предлагаемое решение, так же как и предыдущее, расширяет функциональные возможности схемы передачи данных, использующей сигнал разграничения битов (см. рис. 3.10). Точнее, простыми средствами создается некий дополнительный канал связи, по которому передаются признаки, определяющие положение межкадровых границ (рис. 3.18).

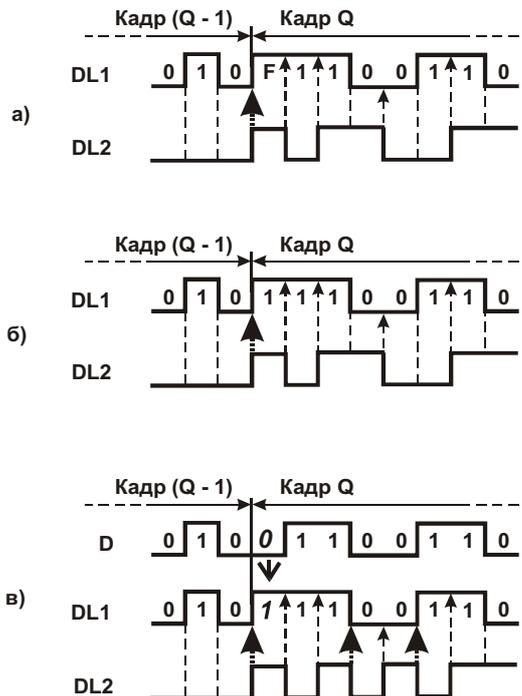


рис. 3.18. Временные диаграммы, поясняющие идею передачи кадровых меток:
а – с применением однобитового флага F ; *б*, *в* – без применения флага

В предлагаемом решении кадры могут передаваться двумя основными способами. Первый способ (рис. 3.18, *а*) подразумевает наличие флагового (служебного) бита F в начале каждого кадра. Второй способ (рис. 3.18, *б*, *в*) применим к передаче кадров “стык в стык”, без каких-либо служебных битов вблизи межкадровых границ. Второй способ предпочтителен, так как данные передаются с большей скоростью благодаря исключению из потока лишних (с точки зрения потребителя данных) битов F . Но этот

способ, в отличие от первого, требует некоторых “интеллектуальных затрат” со стороны потребителя данных, так как в среднем в половине ситуаций нужно не только вычислить положение межкадровой границы, но и проинвертировать первый бит кадра, как будет показано далее. Рассмотрим оба способа подробнее.

Первый способ позволяет обозначить межкадровую границу выполнением двух правил.

Правило 1.1. Флаговый бит F кадра Q (см. рис. 3.18, *a*) выбирается противоположным последнему биту предыдущего кадра $Q - 1$. В данном примере эти биты равны соответственно 1 и 0. Это обеспечивает гарантированный перепад уровней сигнала $DL1$ на межкадровой границе.

Правило 1.2. На межкадровой границе сигнал $DL2$ изменяет состояние. В данном примере этот сигнал на межкадровой границе переходит из 0 в 1.

В результате выполнения обоих правил создается уникальная кодовая ситуация, при которой изменению сигнала $DL2$ соответствует изменение (а не статическое состояние) сигнала $DL1$. Эта ситуация обозначена утолщенной стрелкой и может рассматриваться как допустимое нарушение правил кодирования, принятых в прототипе (см. рис. 3.10). Допустимость такого нарушения подтверждается тем, что межкадровая граница задается одновременно изменяющимися сигналами $DL1$ и $DL2$, что не может ухудшить качество синхронизации между приемником и передатчиком.

Второй способ позволяет обозначить межкадровую границу выполнением двух правил.

Правило 2.1. Если по обеим сторонам межкадровой границы присутствуют разноименные биты (01 или 10), то сигнал $DL2$ на этой границе изменяет состояние, (см. рис. 3.18, *b*). Эта ситуация, по существу, аналогична рассмотренной ранее (см. рис. 3.18, *a*) и отличается от нее только тем, что вместо служебного бита F передается полезный бит, который “случайно” оказался противоположным по отношению к предыдущему.

Правило 2.2. Если по обеим сторонам межкадровой границы первоначально присутствуют одноименные биты (00 или 11), то перепад уровней сигнала $DL1$ на этой границе создается искусственно, инвертированием первого бита кадра Q , как показано на рис. 3.18, *в*. В приведенном примере исходные данные D содержат нулевые биты по обеим сторонам межкадровой границы, но, благодаря инвертированию первого бита кадра Q , сигнал данных $DL1$ с искусственно внесенной ошибкой содержит перепад уровней на этой границе. Чтобы отличить эту ситуацию от предыдущей (см. правило 2.1), в сигнал $DL2$ вводится не одно, а ряд описанных ранее нарушений правил кодирования. Первое из них соответствует межкадровой границе. В данном примере используются три следующих подряд нарушения правил, хотя возможны и иные “договоренности” между передатчиком и приемником. Приемник обнаруживает обусловленную “договором” последовательность нарушений правил, по первому из них определяет межкадровую границу и инвертирует первый бит кадра Q , восстанавливая исходную последовательность битов (см. диаграмму сигнала D).

Чтобы защититься от тривиальных кадров, например типа “все нули” или “все единицы”, в которых вообще отсутствуют перепады сигнала данных, применяется скремблирование. В скремблированном потоке данных в среднем каждый второй битовый интервал содержит перепад уровней сигнала.

Пример схемной реализации системы с разграничением кадров

Рассмотрим пример реализации только что описанной идеи. Система передачи данных, показанная на рис. 3.19, предназначена для пересылки кадров из устройства DTE1 в устройство DTE2.



Рис. 3.19. Система передачи данных, использующая дополнительный канал связи для передачи кадровых меток

Пара сигналов $TxC - TxD$ обеспечивает синхронную передачу данных из устройства $DTE1$ в передатчик. Положительные фронты сигнала TxC задают границы битовых интервалов. По отрицательным фронтам сигнала TxC соответствующие биты данных TxD временно запоминаются в передатчике. Признак FR начала кадра формируется устройством $DTE1$ в виде сигнала лог. 1 и передается на вход передатчика одновременно с первым битом этого кадра.

Пара сигналов $RxC - RxD$ на выходах приемника обеспечивает синхронную передачу данных в устройство $DTE2$. Положительные фронты сигнала RxC задают границы битовых интервалов. По отрицательным фронтам сигнала RxC соответствующие биты данных RxD запоминаются в устройстве $DTE2$. Признак $FRAME$ начала кадра формируется приемником в виде сигнала лог. 1 и передается на вход устройства $DTE2$ одновременно с первым битом этого кадра.

Далее рассмотрены три режима работы системы передачи данных. В первом режиме данные передаются без кадровых синхроимпульсов FR ($FRAME$). Во втором режиме передаваемые кадры помечены однобитовыми флагами F . В третьем режиме кадры не содержат флаговых битов.

Первый режим

Все процессы, протекающие при передаче данных, синхронизированы сигналами от генератора G (рис. 3.20). Сигнал $CL1$ (TxC) с выхода генератора передается в устройство $DTE1$. В ответ на этот сигнал из устройства $DTE1$ выдаются данные TxD . Они поступают на вход данных двухразрядного сдвигового регистра RG . Под действием положительных фронтов сигнала $CL2$ на входе синхронизации этого регистра данные сдвигаются в нем, выдаются через усилитель в линию и одновременно с этим анализируются элементом Искключающее ИЛИ 1. До тех пор, пока биты в регистре RG чередуются (...010101...), формируется сигнал $W = 1$, который поддерживает элемент 2 в режиме инвертирования сигнала, поступающего с нулевого выхода триггера 4.

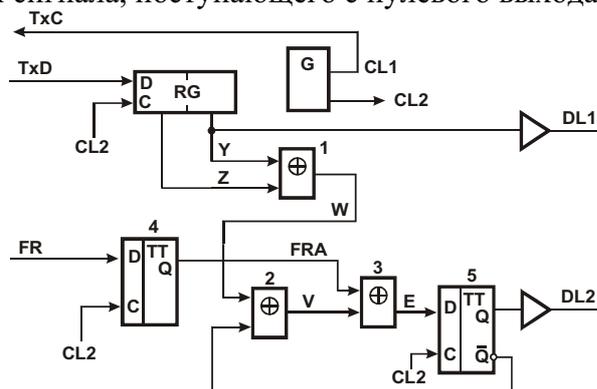


Рис. 3.20. Схема передатчика

Так как в данном режиме сигнал FR кадровой синхронизации отсутствует (постоянно равен лог. 0), триггер 4 постоянно находится в нулевом состоянии ($FRA = 0$). При этом элемент 3 работает в режиме повторения сигнала V ($E = V$). Сигнал с нулевого плеча триггера 5 возвращается на его вход данных после инвертирования элементом 2, поэтому в каждом такте текущее состояние этого триггера подтверждается. Сигнал $DL2$ в линии остается неизменным (0 или 1).

Обнаружение элементом 1 одинаковых битов в регистре RG означает, что необходимо создать перепад уровней сигнала DL2. В этой ситуации сигнал $W = 0$ переводит элемент 2 в режим повторения сигнала с нулевого выхода триггера, поэтому триггер включается в счетный режим, при котором его состояние под действием тактового импульса CL2 изменяется на противоположное.

Сигналы DL1 и DL2 поступают в приемник (рис. 3.21 - рис. 3.23), проходят через усилители и формирователи F импульсов. Сигнал G сопровождает любое изменение сигнала DL1. В данном примере в сигнале G импульсы отсутствуют на позициях, соответствующих вертикальным стрелкам на диаграммах. Сигнал H сопровождает любое изменение сигнала DL2. С помощью элемента ИЛИ “недостающие” импульсы сигнала G восполняются, и сигнал K представляет собой периодическую последовательность импульсов без “пробелов”. Этот сигнал поступает на вход блока PLL фазовой автоподстройки частоты.

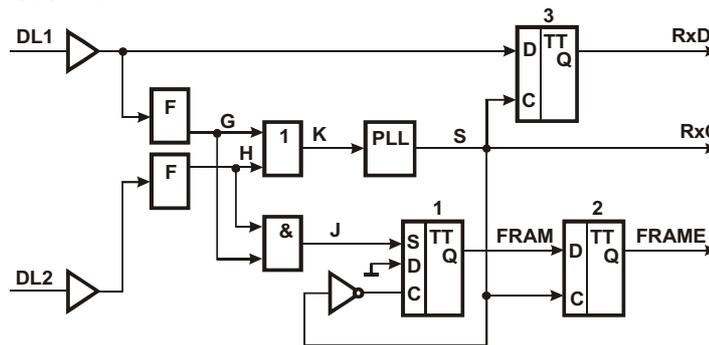


рис. 3.21. Схема приемника

Блок фазовой автоподстройки частоты предназначен для формирования высокостабильного синхросигнала S (RxC) на основе непрерывного “инерционного” слежения за входным сигналом K. Так как в данном режиме импульсы G и H никогда не формируются одновременно, сигнал J на выходе элемента И постоянно равен 0, триггеры 1 и 2 находятся в нулевом состоянии. Принятые из канала связи данные проходят через триггер 3 и передаются в устройство DTE2.

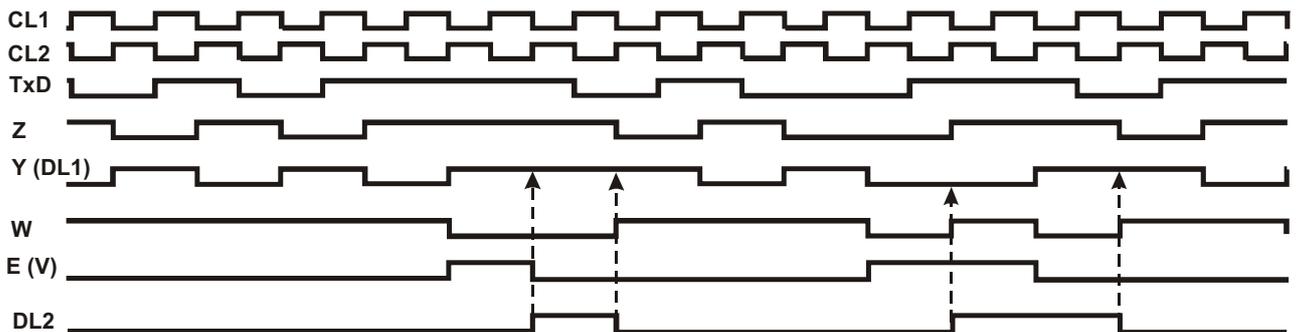


рис. 3.22. Временные диаграммы сигналов передатчика в отсутствие кадровых синхроимпульсов

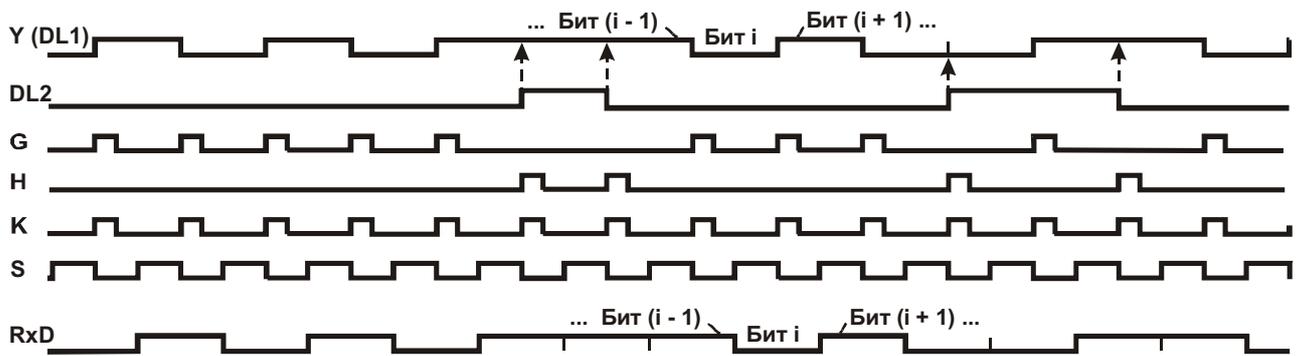


рис. 3.23. Временные диаграммы сигналов приемника в отсутствие кадровых синхросигналов

Второй режим

Этот режим отличается от предыдущего тем, что из устройства DTE1 поступают импульсы FR, соответствующие флаговому биту передаваемых кадров, (рис. 3.24, рис. 3.25). Это приводит к одновременному формированию фронтов сигналов DL1 и DL2. В момент обнаружения межкадровой границы формируется сигнал $J = 1$, который устанавливает триггер 1 (см. рис. 3.21) в единичное состояние. Сигнал с выхода этого триггера проходит через триггер 2 и поступает в устройство DTE2.

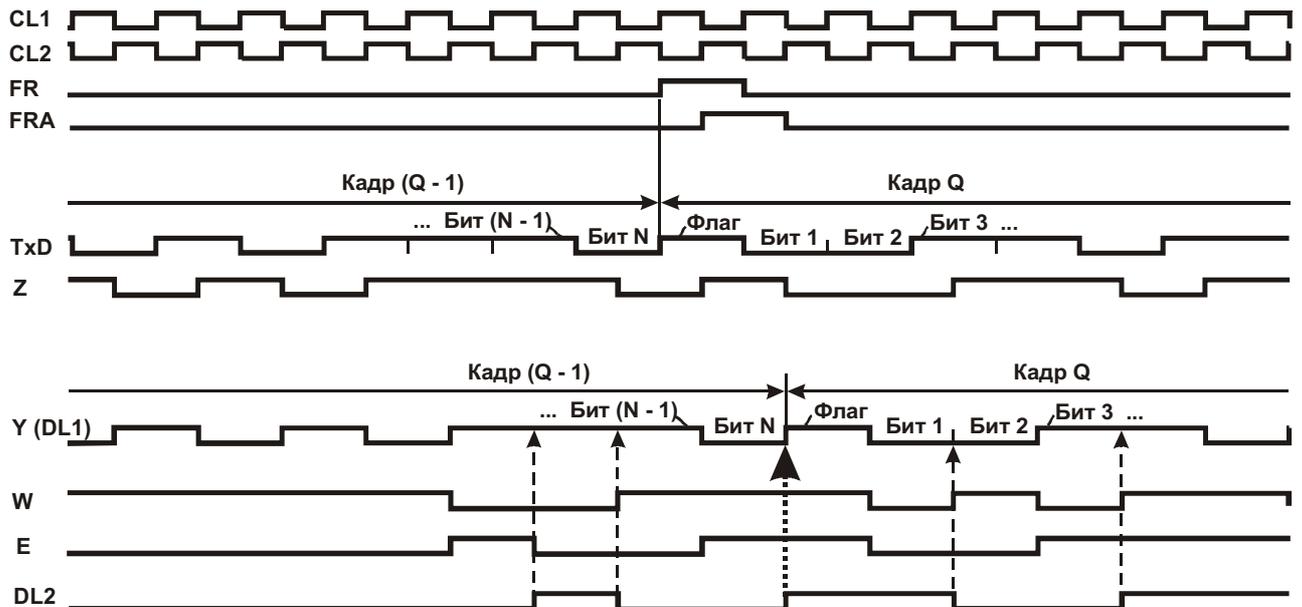


рис. 3.24. Временные диаграммы сигналов передатчика при пересылке кадров с однобитовыми флагами

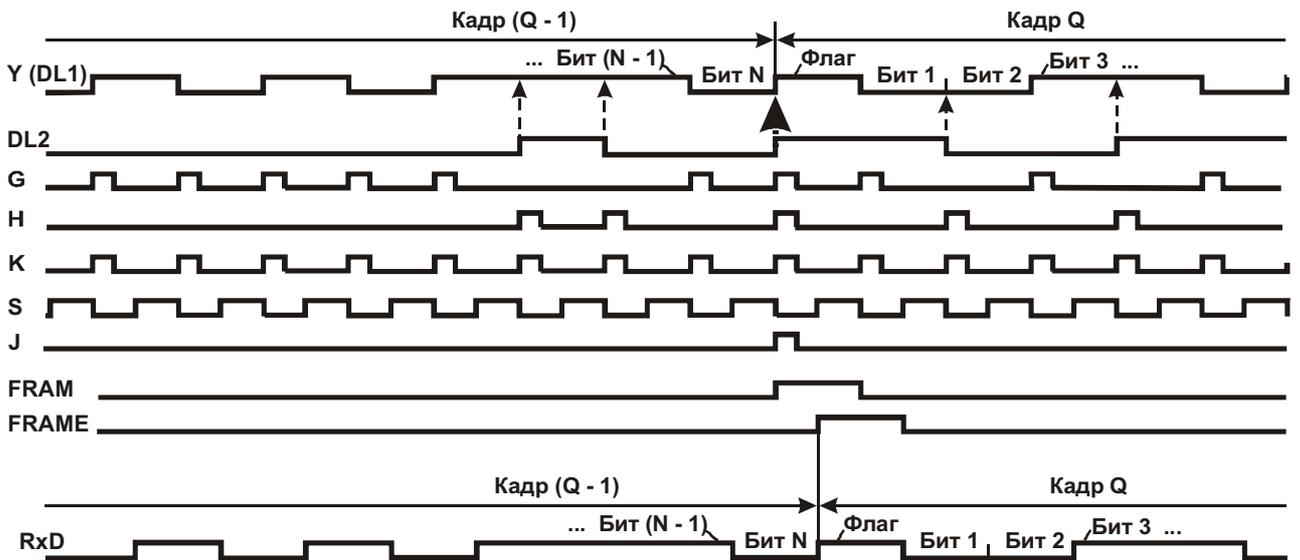


рис. 3.25. Временные диаграммы сигналов приемника при пересылке кадров с однобитовыми флагами

Третий режим

В этом режиме возможны две ситуации. Первая характеризуется наличием противоположных битов по обеим сторонам межкадровой границы. В этом случае временные диаграммы, приведенные на рис. 3.26 и рис. 3.27, отличаются от диаграмм на рис. 3.24 и рис. 3.25 только тем, что вместо флагового бита передается первый “полезный” бит кадра.

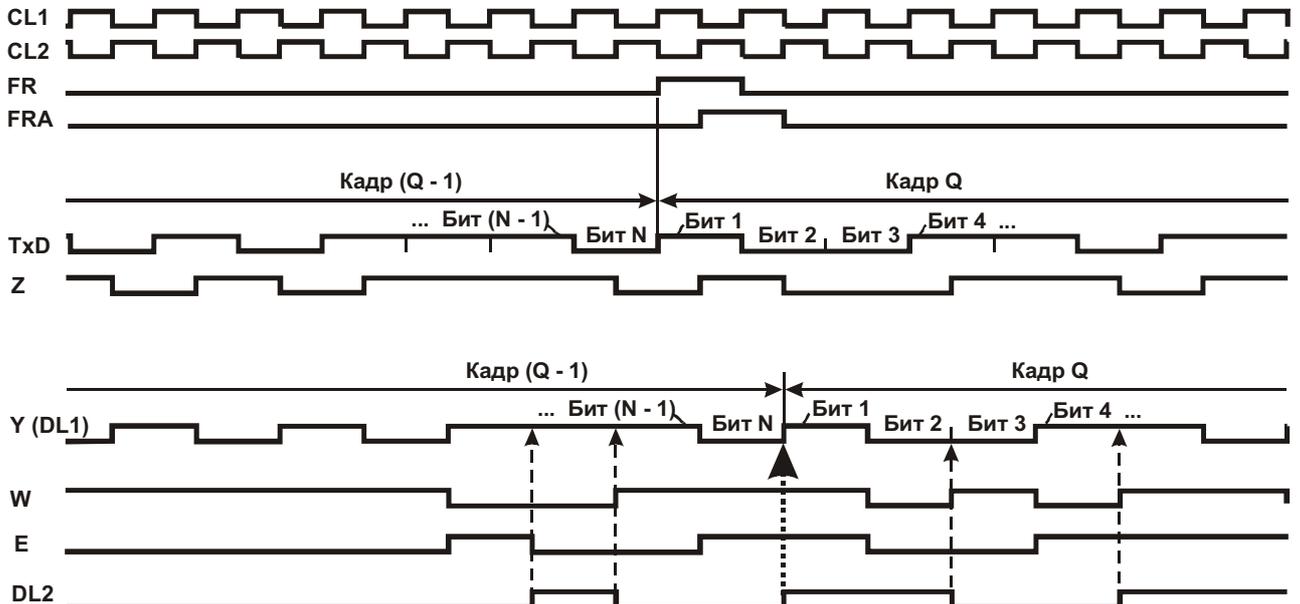


рис. 3.26. Временные диаграммы сигналов передатчика при пересылке бесфлаговых кадров с разноименными битами по обеим сторонам межкадровой границы

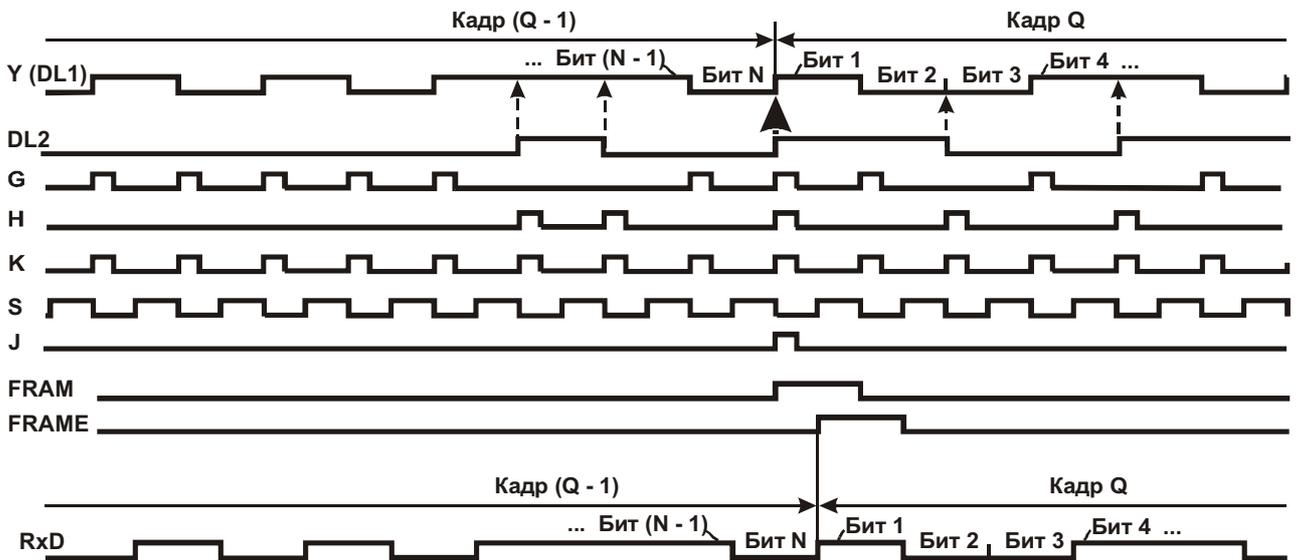


Рис. 3.27. Временные диаграммы сигналов приемника при пересылке бесфлаговых кадров с разноименными битами по обеим сторонам межкадровой границы

Вторая ситуация (рис. 3.28, рис. 3.29) также характеризуется наличием противоположных битов по обеим сторонам межкадровой границы, но, как уже отмечалось, первый бит кадра преднамеренно искажен для создания гарантированного перепада уровней сигнала DL1 на межкадровой границе. Чтобы отличить вторую ситуацию от первой, вместо одного передаются три кадровых импульса.

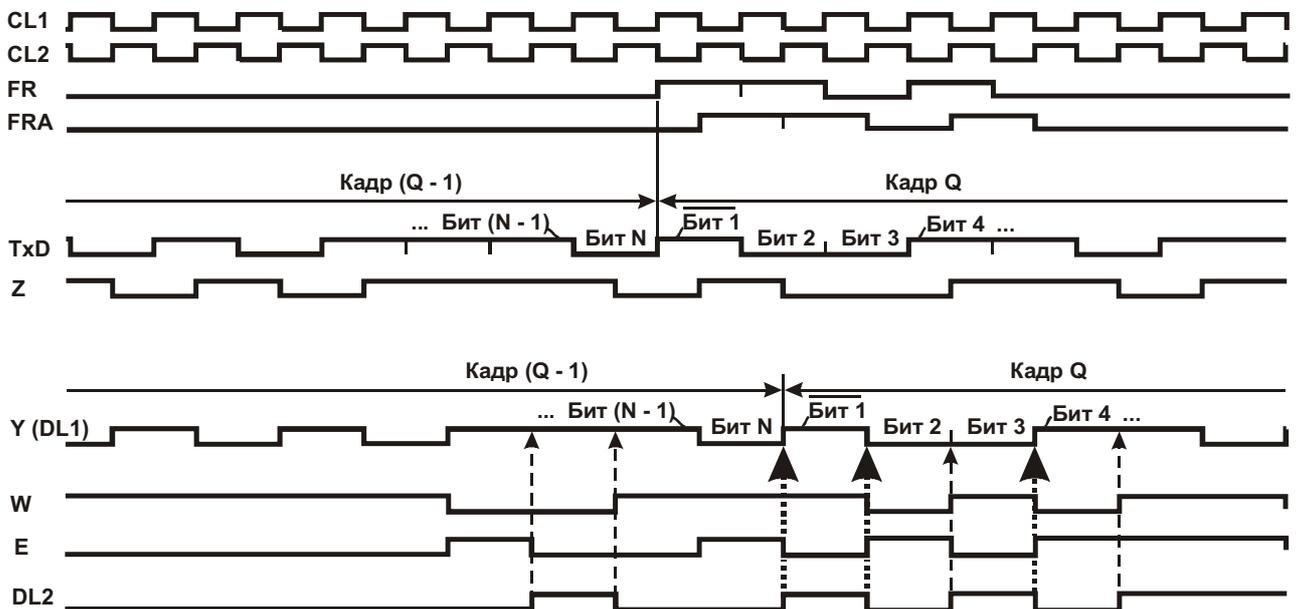


Рис. 3.28. Временные диаграммы сигналов передатчика при пересылке бесфлаговых кадров с одноименными битами по обеим сторонам межкадровой границы

Из диаграмм, приведенных на рис. 3.28, видно, что из устройства DTE1 периодически поступают триады кадровых импульсов. Это приводит к появлению трех событий одновременного формирования фронтов сигналов DL1 и DL2. Сигнал $J = 1$ формируется в приемнике троекратно и воздействует на вход установки триггера 1 в единицу. Отметим, что в D-триггере сигнал S установки единицы имеет более высокий приоритет, чем сигнал на динамическом входе С. Сигнал с выхода триггера 1 проходит

через триггер 2 и поступает на вход устройства DTE2. Как уже отмечалось, получение устройством DTE2 серии близлежащих кадровых импульсов свидетельствует о необходимости инвертирования первого бита кадра.

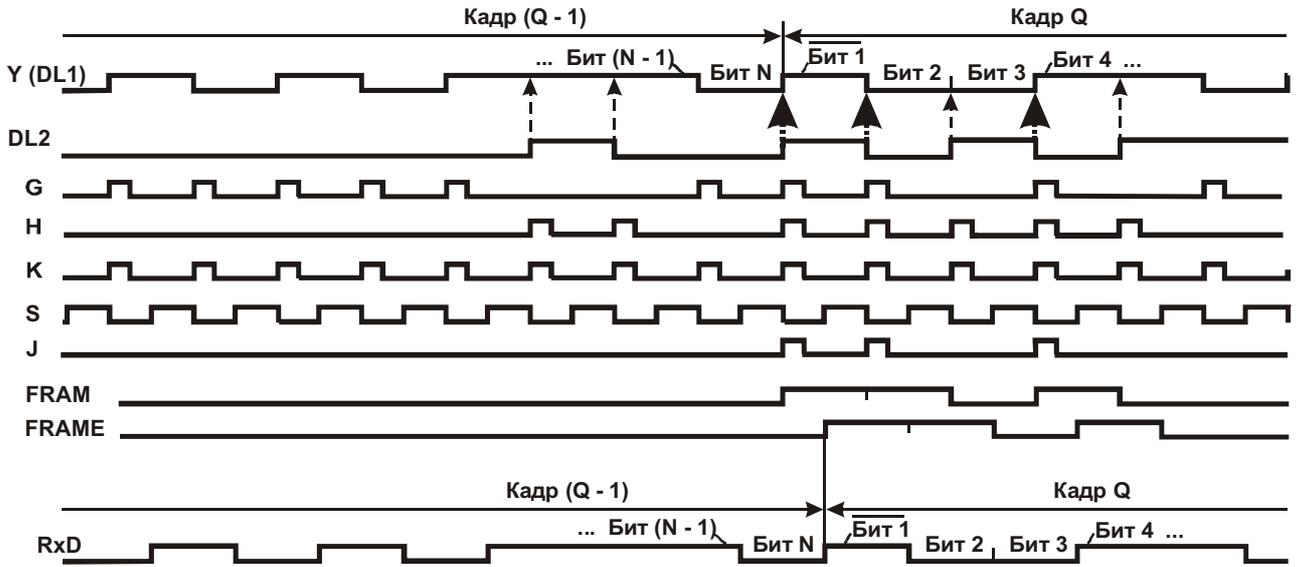


Рис. 3.29. Временные диаграммы сигналов приемника при пересылке бесфлаговых кадров с одноименными битами по обеим сторонам межкадровой границы

Возможны также иные способы передачи данных с помощью предлагаемого устройства. Например, для повышения помехоустойчивости можно оперировать не одним и тремя кадровыми импульсами, а группами из M и $2M$ импульсов, где $M > 10$. Канал кадровых импульсов можно одновременно использовать для передачи дополнительного потока данных и т. п.

Выводы.

Рассмотренные в п. 3.3 решения позволяют сделать следующие выводы.

1. Классический способ передачи данных между устройствами (см. рис. 3.6), при котором истинность битов TxD подтверждается отрицательными фронтами синхросигнала CLK, не обеспечивает высокой скорости обмена. Рассмотрены два направления улучшения классического способа передачи данных.

2. Первое направление основано на использовании как отрицательных, так и положительных фронтов синхросигнала CLK для подтверждения истинности битов данных TxD (см. рис. 3.7). Скорость передачи данных при неизменной частоте синхросигнала CLK повышается в два раза.

3. Второе направление подразумевает изменение функций синхросигнала. Этот сигнал (именуемый сигналом разграничения DELIM) формируется таким образом, что его форма зависит от передаваемых данных (см. рис. 3.10). Точнее, сигнал остается неизменным, если изменяются данные, и, наоборот, сигнал изменяется, если данные постоянны. Такое решение, так же как и предыдущее (см. рис. 3.7), повышает скорость передачи данных в два раза и, кроме того, разравнивает спектр синхросигнала (DELIM), что способствует снижению уровня излучаемых в линию помех.

4. Второе направление не исчерпывается только что упомянутым способом формирования сигнала DELIM. При синтезе этого сигнала открываются достаточно широкие возможности замены его безразличных состояний битами данных. Основное следствие этого состоит в том, что сигнал DELIM может переносить данные параллельно с их переносом по линии данных (см. рис. 3.11). При этом с увеличением числа линий данных эффективность такого решения возрастает (см. рис. 3.12).

5. Способность переноса данных сигналом DELIM наряду с их переносом по штатной линии данных использована в двух примерах построения телекоммуникационных систем. Первый пример иллюстрирует возможность построения двухканальной системы (см. рис. 3.13), второй – возможность синхронизации приемника с передатчиком на уровне информационных кадров, причём флаговые биты, соответствующие началу кадров, могут отсутствовать (см. рис. 3.19).

3.4. Аппаратное управление потоком данных с использованием пачек сигналов TxC

Рассмотренные далее решения (п. 3.4.1, 3.4.2) объединяет способ управления потоком данных TxD между устройствами DTE и DCE. Этот способ заключается в том, что вместо непрерывного синхросигнала использованы пачки сигналов TxC. В зависимости от интенсивности формирования импульсов TxC изменяется интенсивность передачи битов данных TxD. Это позволяет упростить схемы согласования скоростей передачи данных при работе с линией, исключив линии RTS и CTS, а также улучшить структуру системы синхронизации формирователя информационных кадров.

3.4.1. Согласование средней скорости передачи данных между устройствами DTE и DCE со скоростью передачи данных по линии

В отличие от традиционного решения задачи аппаратного управления потоком данных (hardware flow control, см. п. 1.2) сигналами RTS и CTS, в [6] предлагается возложить функцию регулирования скорости одного из потоков на синхросигнал TxC.

В примере построения схемы, реализующей эту идею (рис. 3.30), компьютер (устройство типа DTE, см. гл. 1) обменивается данными с удаленным абонентом (на рисунке не показан) через модем (устройство типа DCE). Компьютер взаимодействует с модемом по интерфейсу RS-232 или иному, принадлежащему этому же семейству. Скорость передачи данных по линии в каждом направлении равна 160 кбит/с. Скорость приема данных RxD в компьютер также равна 160 кбит/с.

Скорость передачи данных TxD из компьютера в модем выбирается большей, чем скорость передачи данных по линии, и в нашем примере составляет 230 кбит/с. Чтобы этот факт не вызывал недоразумений, сразу отметим, что данные TxD передаются из компьютера в модем не непрерывно, а пачками, по мере необходимости пополнения буферной памяти модема. Так что средняя скорость передачи данных из компьютера в модем равна скорости их выдачи в линию и составляет 160 кбит/с, в отличие от мгновенной скорости (230 кбит/с), на которой передаются данные в пачке.

Напомним, что синхросигнал RxS сопровождает сигнал данных RxD. По получении сигнала TxC компьютер формирует очередной бит TxD, который поступает в модем и принимается им. Соответствующая временная диаграмма рассматривалась ранее (см. рис. 1.6, б).

Модем содержит передатчик, приемник и буферную память с блоком управления (другие компоненты сейчас не представляют для нас интереса). Передатчик непрерывно считывает биты из буферной памяти, смешивает их с синхросигналом C и передает полученную смесь в линию. Приемник выделяет из поступающего от удаленного абонента линейного сигнала данные RxD и синхросигнал C, который поступает на вход RxS компьютера. Этот же синхросигнал C, как отмечалось, управляет считыванием данных из буферной памяти и их выдачей в линию.

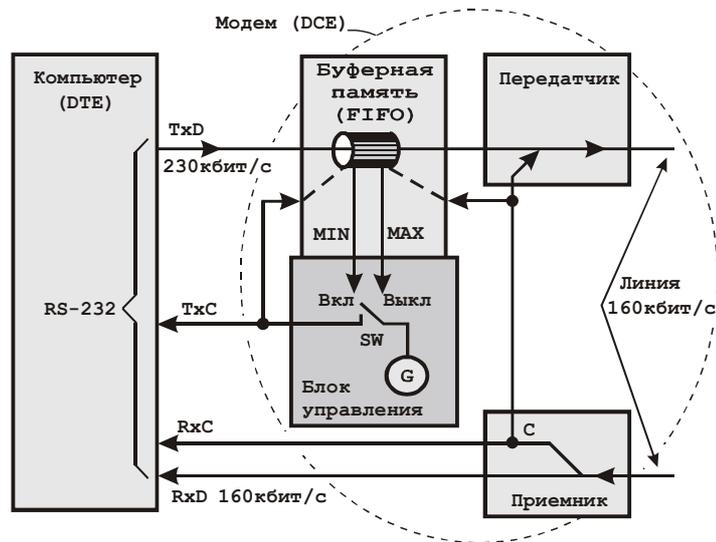


рис. 3.30. Пример схемы с аппаратным управлением потоком данных TxD

Буферная память условно показана в виде трубопровода. Она работает по принципу “первым прибыл, первым обслужен” (First Input – First Output или FIFO) [72]*. С такой памятью могут бесконфликтно работать местные источник и приемник данных. Конфликты, которые могли бы возникнуть при одновременных обращениях к памяти по записи и чтению, устраняются аппаратно. Буферная память содержит индикаторы уровня заполнения. Если уровень велик и превышает некоторый порог, например 90%, то срабатывает индикатор максимума MAX, ключ (логический элемент) SW в блоке управления размыкается. Если уровень мал, например менее 10%, то срабатывает индикатор минимума MIN, ключ SW в блоке управления замыкается. Если уровень лежит в пределах от MIN до MAX, то предыдущее состояние ключа сохраняется. Генератор G вырабатывает непрерывную последовательность синхроимпульсов частотой 230 кГц.

В исходном состоянии компьютер непрерывно принимает поступающие из линии данные RxD под управлением сигнала RxС. В то же время под управлением сигнала TxС компьютер передает данные в модем. При этом уровень заполнения буферной памяти равномерно пульсирует в диапазоне от MIN до MAX.

Действительно, предположим, что ключ SW замкнут. Тогда поток данных TxD поступает в буферную память со скоростью 230 кбит/с. Уровень заполнения буфера неуклонно растет, так как скорость рассасывания данных меньше скорости их поступления. В некоторый момент уровень достигает критической отметки (MAX), ключ SW размыкается, компьютер прекращает выдачу данных TxD, но процесс их рассасывания продолжается. Уровень заполнения монотонно снижается, и когда он пересекает границу MIN, ключ SW замыкается и т. д.

Приведенный пример не следует рассматривать только как конкретную схему управления потоком данных – это всего лишь наиболее простая иллюстрация идеи сокращения числа управляющих линий и повышения эффективности использования процессорного времени компьютера и модема. Последнее достигается за счет концентрации данных в компактные пакеты, разделенные интервалами. Эти интервалы можно использовать для тех или иных вычислений.

* В дальнейшем использованы также две альтернативные модели такой памяти – бак с жидкостью и кольцо из ячеек памяти (см., например, п. 4.2). Модель в виде кольца из ячеек памяти позволяет более детально проследить за процессами, протекающими в буферной памяти, но обладает меньшей наглядностью по сравнению с двумя другими.

3.4.2. Упрощение системы синхронизации формирователя HDSL-кадров

Рассмотрим процесс формирования одной из разновидностей так называемых HDSL-кадров*, уделяя основное внимание вопросам синхронизации взаимодействующих устройств.

В системе, показанной на рис. 3.31, данные передаются из устройства 1 в устройство 2 через модемы 1 и 2 в синхронном режиме. Эти модемы наделены способностью формировать кадры при передаче данных в линию и расформировывать кадры, принимаемые из линии. Здесь и далее показана односторонняя передача данных, но, предполагая, что схема симметрична, будем иметь в виду, что все описанные процессы протекают и при транспортировании данных в обратном направлении.

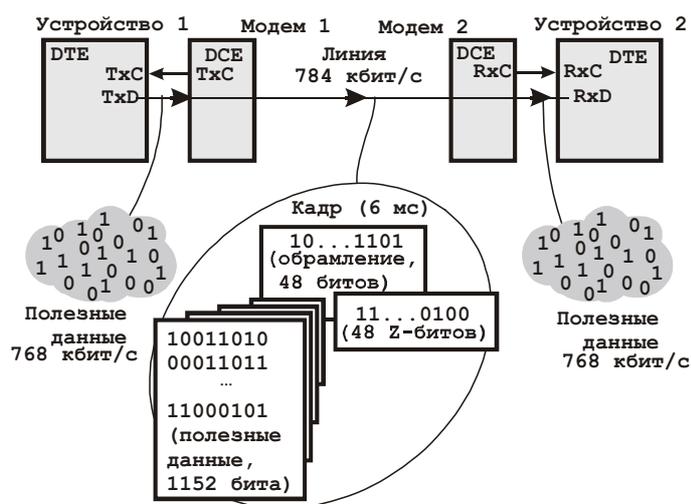


рис. 3.31. Система передачи данных между устройствами 1 и 2

Под управлением синхросигналов ТхС устройство 1 передает в модем 1 поток данных со скоростью 768 кбит/с. Модем 1 упаковывает данные в кадры определенного формата. Номинальная длительность кадра – 6 мс (возможны небольшие отклонения в сторону уменьшения или увеличения, но ими пренебрегаем). Как условно показано на рисунке, каждый кадр содержит четыре поля данных по 1152 бита в каждом, а также два служебных поля.

Первое служебное поле определяет некое обрамление кадра – стартовый флаг, контрольную сумму и иные признаки. Его размер – 48 битов. Второе служебное поле такого же размера (48 так называемых Z-битов) представляет собой группу вакантных мест для передачи сообщений между устройствами, способными анализировать структуру кадра, и, в частности, выделять из него Z-биты. Такими устройствами в нашем примере являются модемы 1 и 2. С помощью Z-битов создается невидимый канал связи между модемами, способный передавать информацию с темпом 48 бит за 6 мс или, что то же самое, 8 кбит/с. (Первые восемь из 48 Z-битов зарезервированы “на будущее” – они пока не используются и постоянно установлены в состояние лог. 1; но в данный момент это нас не интересует.)

Такой же вклад (8 кбит/с) в общий поток данных по линии вносит и первое служебное поле. Действительно, в течение каждого 6 мс передаются 48 битов этого поля, а это соответствует тому, что за одну секунду передаются 8 кбит служебных данных. В результате суммарная скорость передачи данных по линии складывается из скоростей трех потоков (полезного и двух служебных) и составляет $768 + 8 + 8 = 784$ кбит/с.

* HDSL – High-bit-rate Digital Subscriber Line – технология высокоскоростной передачи данных по кабелям на основе витых пар медных проводов.

Иными словами, к первоначальному потоку полезных данных из устройства 1 (768 кбит/с) добавляется бесполезный (с точки зрения гипотетического потребителя данных) поток сравнительно малой интенсивности – 16 кбит/с. Накладные расходы составляют $16/768$ или около 2,1%.

Более строгое отображение структуры кадра приведено на рис. 3.32. Служебные биты помечены затененными областями. Из рисунка следует, что поле обрамления из 48 бит представлено разрозненными фрагментами ($16 + 10 + 10 + 10 + 2 = 48$). То же относится и к Z-битам – они перемежаются с 96-битовыми блоками данных. Граница окончания кадра может перемещаться влево или вправо на два бита (см. обозначения “-” и “+”) при соответствующем изменении длины первого служебного поля. В наших расчетах принимаем номинальную длительность кадра – 6 мс.

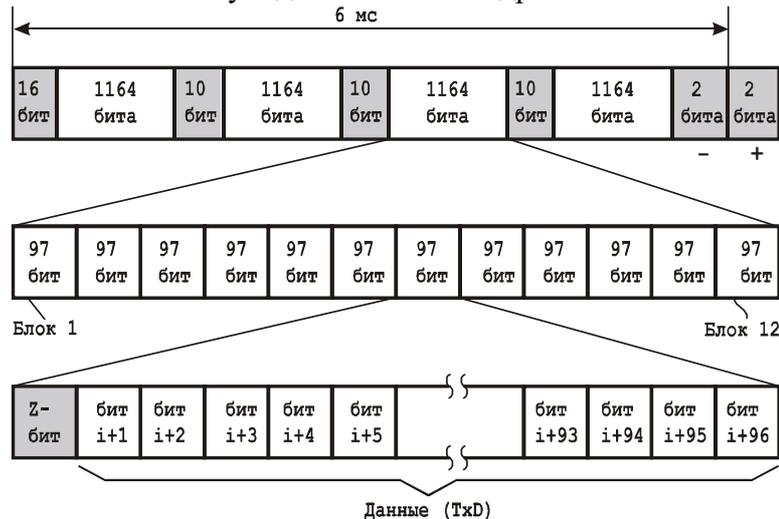


рис. 3.32. Структура одного из вариантов HDSL-кадра.

Проблема заключается в том, что модемы 1 и 2 (см. рис. 3.31) оперируют двумя частотами: 768 и 784 кбит/с. Их можно получить от двух генераторов или от общего синтезатора частот. Однако такие решения слишком громоздки (и это, как будет показано далее, не единственный недостаток). В [7] предлагается в явном виде не использовать частоту 768 кГц и оперировать только частотой 784 кГц, сохраняя структуру потоков в соответствии с рис. 3.31! (При этом “недостающая” частота в неявном виде формируется без дополнительных затрат аппаратуры.)

Идея поясняется схемой и временной диаграммой (рис. 3.33).

В данном примере компьютер и модем выполнены соответственно в виде устройств типа DTE и DCE (см. гл. 1). Компьютер взаимодействует с модемом по одному из интерфейсов семейства RS-232. Передача данных из компьютера в модем выполняется в синхронном режиме, в соответствии с временной диаграммой, которая была показана на рис. 1.6, б.

Скорость передачи данных TxD равна 784 кбит/с при условии, что электронный ключ SW замкнут управляющим сигналом $L = 1$. Если $L = 0$, то скорость передачи равна нулю. При этом, как мы сможем убедиться, средняя скорость равна 768 кбит/с, что и требуется!

Передаваемые из компьютера данные временно накапливаются в буферной памяти типа FIFO. Формирователь кадров добавляет к данным необходимую служебную информацию. Под действием синхросигналов F сформированные кадры непрерывно и равномерно выдвигаются с выхода памяти на вход передатчика. Передатчик кодирует данные и выдает их в линию.

Формирователь кадров выполнен на основе высокоскоростного микропроцессора и обладает достаточно высоким интеллектом. Программа сборки кадра, в частности, знает, что после размещения на входе буферной памяти очередного Z-бита нужно принять

на вход серию из 96 битов данных TxD (см. нижнюю строку диаграммы, представленной ранее на рис. 3.32). Чтобы это осуществить, формирователь кадров выдает сигнал загрузки L = 1, ключ SW замыкается, начинается передача данных TxD.

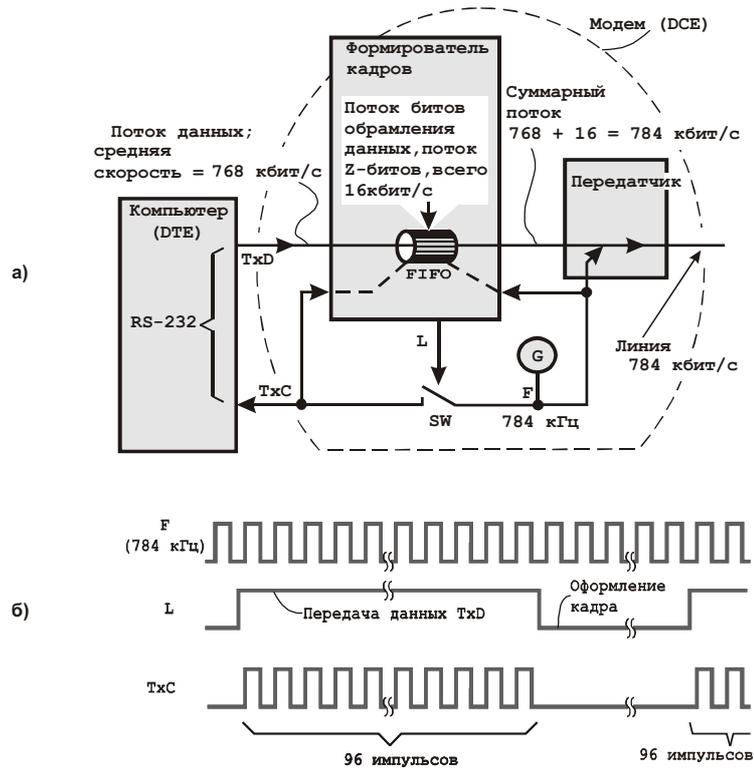


рис. 3.33. Передача данных из компьютера в линию через модем: а – схема синхронизации; б – временные диаграммы

Формирователь отсчитывает 96 импульсов и размыкает ключ SW сигналом $L = 0$. Поток входных данных прекращается. После этого в буферную память вводятся нужные служебные биты, и когда возникает потребность в приеме очередной группы из 96 битов данных, ключ SW вновь замыкается и т. д.

Средняя скорость передачи данных TxD составляет 768 кбит/с. Действительно, с учетом структуры кадра, приведенной на рис. 3.32, за время 6 мс ключ SW замыкается 48 раз, и при каждом замыкании передаются 96 битов данных, всего пересылаются $48 \times 96 = 4608$ битов. Этот темп соответствует передаче $(4608 \times 1000) / 6 = 768\,000$ битов за 1 с, что и требовалось доказать.

Мы уже отмечали одно из преимуществ такого решения – использование только одной частоты синхронизации (другая является как бы виртуальной). Второе преимущество связано с тем, что в процессе оформления кадра нет поступления внешних данных. Это позволяет уменьшить требуемый объем буферной памяти. Кроме того, не нужен аппаратный арбитр, так как принципиально исключены конфликты, которые при иных решениях могли бы возникать при одновременном обращении к буферной памяти со стороны двух источников информации.

Подобное построение системы синхронизации применимо и при расформировании кадра модемом 2 (см. рис. 3.31). В этом случае частота 784 кГц выделяется из линейного сигнала. Данные RxD передаются в устройство 2 аритмично, пачками, со средней скоростью 768 кбит/с.

3.5. Взаимодействие удаленных устройств с использованием дистанционной цифровой коррекции фазы сигнала от синхрогенератора

Дистанционная коррекция фазы сигнала от удаленного синхрогенератора позволяет проектировать системы передачи данных, отвечающие некоторым специфическим требованиям. Одно из таких требований заключается в обеспечении совпадения синхросеток передаваемых и принимаемых данных (п. 3.5.1). Другое требование возникает при проектировании удлинителя сигналов TxС и RxС (в общем случае – несинхронных и несинфазных), когда их необходимо передать на расстояние порядка нескольких километров. Это требование состоит в том, что сигнал TxС должен распространяться “против течения” потока данных (п. 3.5.2).

3.5.1. Выравнивание синхросеток передаваемых и принимаемых данных

В одном из рассмотренных ранее вариантов взаимодействия устройств типа DTE и DCE (см. рис. 2.4) решалась задача получения синхронных и синфазных потоков данных на стыке между устройствами DCE 1 и DCE 2. (Напомним, что эта задача – одна из часто встречающихся на практике при построении синхронных каналов связи между удаленными устройствами.) На рис. 3.34, а представлено решение той же задачи с использованием идеи, предложенной в [8].

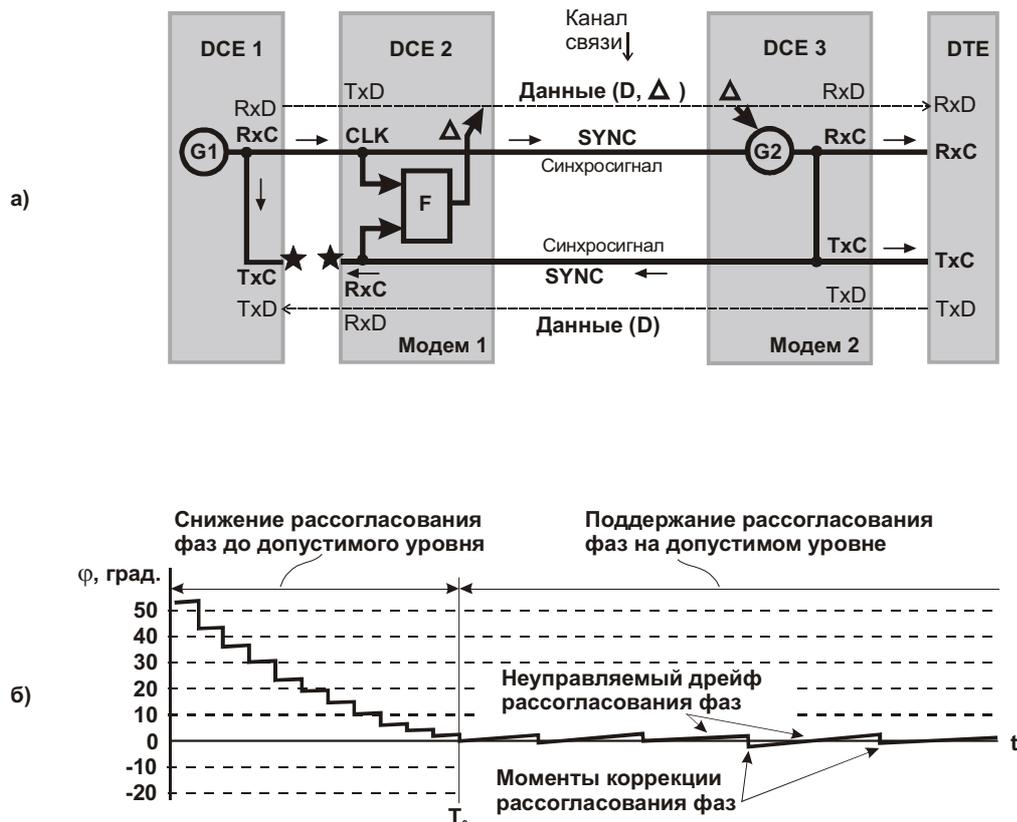


Рис. 3.34. Вариант системы синхронизации двусторонней передачи данных между устройствами DCE 1 и DTE: а – схема; б – временная диаграмма

Рассмотрим задачу более подробно. На практике часто встречаются устройства типа DCE, которые генерируют совпадающие синхросигналы RxС и TxС. В нашем примере устройство DCE 1 (мультиплексор) содержит внутренний генератор G1. Сиг-

нал RxC с выхода генератора $G1$ сопровождает данные, передаваемые в устройство DCE 2, а сигнал TxC с этого же выхода синхронизирует выдачу данных из устройства DCE 2. Таким образом, мультиплексор DCE 1 требует, чтобы его входной поток данных был размещен в той же синхросетке, что и выходной.

Иными словами, сигналы, помеченные на рис. 3.34 звездочками, должны быть синхронными и синфазными (т. е. должны иметь одинаковые частоты и практически совпадающие во времени фронты), несмотря на то, что один из них, распространяясь по каналу связи туда и обратно, возможно, преодолевает до места встречи со своим прототипом многокилометровую петлю и явно опаздывает. Это означает, что сигнал, прошедший длинный путь, должен быть задержан на целое число периодов по отношению к своему прототипу.

Сигнал RxC , помеченный звездочкой, сопровождает данные, передаваемые из устройства DCE 2 в устройство DCE 1. Этот сигнал в явном виде не используется, но его совпадение с сигналом TxC (также помеченным звездочкой) гарантирует правильную передачу данных в устройство DCE 1.

Сигнал RxC с выхода генератора $G1$ поступает на вход CLK устройства DCE 2 (модем 1). Модем 1 временно накапливает данные, поступающие от устройства DCE 1, формирует из них кадры и после кодирования передает их в канал связи. Сигнал в канале связи представляет собой смесь данных D с синхроимпульсами SYNC. Помимо полезных данных в кадре передаются служебные, такие как заголовок, контрольная сумма и т. д. Среди служебных данных содержится некоторый код Δ . Число разрядов этого кода может быть равно, например, трем.

Модем 2 принимает сигнал из канала связи, выделяет из него синхроимпульсы и данные, отделяет полезные данные от служебных и передает полезные данные в устройство DTE. Истинность этих данных подтверждается синхросигналом RxC , который поступает на одноименный вход устройства DTE. В свою очередь, устройство DTE в такт с запросным сигналом TxC выдает данные в модем 2. Эти данные после соответствующих преобразований передаются по каналу связи в модем 1 и далее в устройство DCE 1.

Выделение синхроимпульсов из канального сигнала модемом 2 осуществляется с помощью ведомого генератора $G2$, выполненного по схеме с фазовой автоподстройкой частоты. Особенность этого генератора в том, что под действием управляющего кода Δ он может вносить фазовое смещение выходного сигнала в ту или иную сторону на большую или меньшую величину. Например, при $\Delta = 000_2$ генератор $G2$ не изменяет имеющийся фазовый сдвиг выходного сигнала; при $\Delta = 001_2$ или $\Delta = 010_2$ выходной сигнал дополнительно смещается вправо или влево относительно предыдущего положения на один градус и т. д.

Фазовый компаратор F сравнивает фазы сигналов на его входах (фактически это сигналы, помеченные звездочками). Первый сигнал поступает от генератора $G1$ по короткому пути, второй формируется из канального сигнала с помощью обычного генератора с фазовой автоподстройкой частоты (этот генератор на рисунке не показан). Если фронты сигналов практически совпадают (а это и есть условие правильной работы системы), то $\Delta = 000_2$, генератор $G2$ сохраняет “статус-кво”.

При обнаружении заметного и устойчивого фазового расхождения однократно формируется ненулевой код Δ , который вводится в ближайший кадр и пересылается в модем 2. Генератор $G2$ реагирует на получение этого кода так, что фазовое расхождение сигналов на входах компаратора уменьшается. Если это расхождение снизилось до допустимых пределов, то состояние системы не изменяется; в противном случае проводится новый сеанс коррекции фазы и т. д. Диапазон регулирования фазы сигнала RxC относительно сигнала TxC (тот и другой помечены звездочками) равен ± 180 град.

На рис. 3.34, б показана временная диаграмма регулирования рассогласования фаз ϕ сигналов RxС и TxС. В исходном состоянии (до момента T_0) рассогласование недопустимо велико, поэтому устройство DCE 1 принимает от устройства DCE 2 неправильные данные. Возможность уменьшения рассогласования фаз сохраняется благодаря тому, что канал связи способен передавать данные от устройства DCE 1 к устройству DTE. После ряда шагов коррекции рассогласования фаз (к моменту T_0) передаваемые и принимаемые данные на стыке между устройствами DCE 1 и DCE 2 оказываются размещенными практически в одной и той же синхросетке, что и требовалось. Устройство DCE 1 начинает принимать правильные данные. В дальнейшем (после момента T_0) рассогласование фаз поддерживается в допустимых пределах, как уже было описано.

3.5.2. Передача синхросигнала “против течения” потока данных

Идею коррекции фазы сигнала от удаленного синхрогенератора, описанную в п. 3.5.1, можно применить к решению задачи параллельного распространения по каналу связи двух синхросигналов, один из которых как бы движется против течения. Эта задача уже рассматривалась (см. п. 2.3, рис. 2.5). Для удобства изложения в этот рисунок внесены некоторые изменения и дополнения (рис. 3.35).

Напомним содержание задачи. Предположим, что в нашем распоряжении имеются два устройства: мультиплексор DCE 1 и компьютер DTE 2 (рис. 3.35, а). (Нумерация устройств “привязана” к последующим рисункам.) На выходе мультиплексора можно наблюдать два синхросигнала от двух в общем случае независимых генераторов G_1 и G_2 . Сигнал RxС сопровождает передаваемые данные RxD; сигнал TxС служит запросом на получение данных TxD.

Если мультиплексор и компьютер расположены рядом, например, на расстоянии нескольких метров друг от друга, то их можно соединить кабелем по классической схеме, согласно которой объединяются одноименные контакты соединителей обоих устройств (см. гл. 1). Если же расстояние между мультиплексором и компьютером велико (например, составляет 5 км), то уместно применить для передачи данных удлинитель в виде пары модемов, как показано на рис. 3.35, б. На первый взгляд, все верно, но при более внимательном рассмотрении этой схемы возникают сомнения в ее работоспособности.

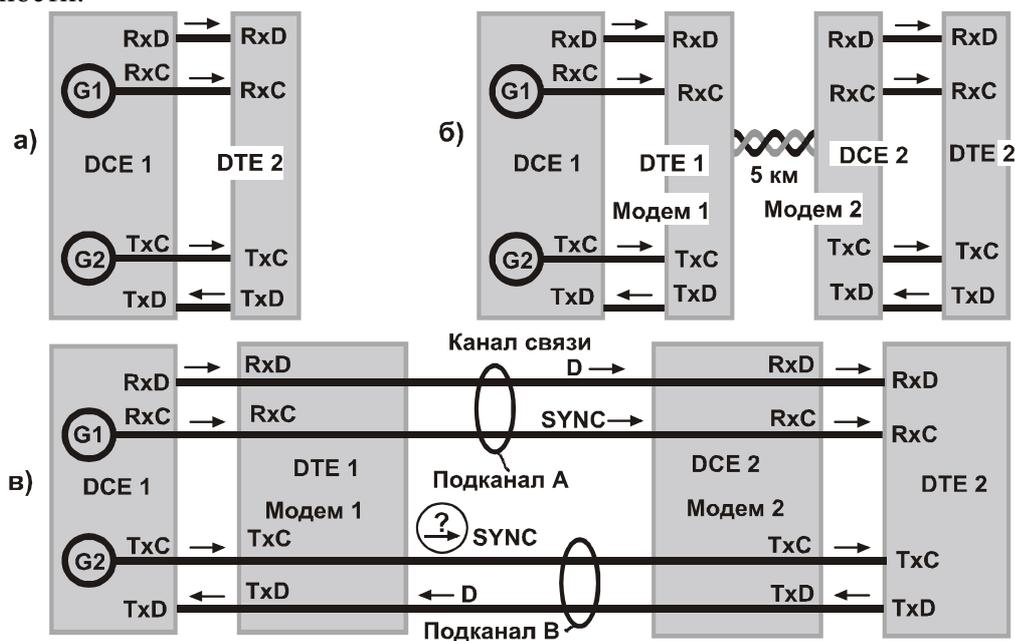


рис. 3.35. Схема взаимодействия устройств DCE 1 и DTE: *а* – при непосредственном соединении; *б, в* – при соединении через удлинитель, образованный цепью “модем 1 – модем 2”

Поясним сказанное. Абстрагируясь от деталей, можно перейти от схемы рис. 3.35, *б* к эквивалентной схеме рис. 3.35, *в*. На ней укрупненно показаны четыре трассы распространения данных и синхросигналов. Канал связи (физически это витая пара проводов) логически представлен двумя подканалами А и В. Подканал А передает данные D и синхросигналы SYNC слева направо, и это не вызывает вопросов, поскольку мы знаем, что линейный передатчик модема 1 (2), как обычно, формирует смесь данных и синхроимпульсов и транслирует ее в подканал.

Однако подканал В должен работать довольно своеобразно. Данные, как и положено, передаются справа налево, но синхросигналы, согласно нашему замыслу (построению удлинителя), должны передаваться встречно! Этот факт отмечен на рисунке вопросительным знаком. В общем, “этого не может быть никогда”.

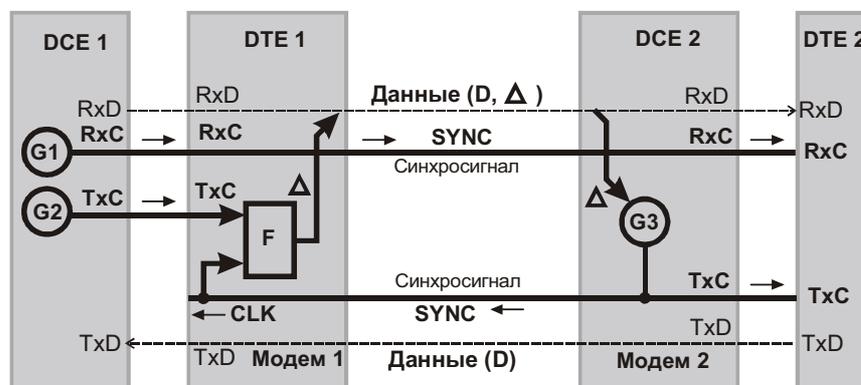


рис. 3.36. Детализация схемы, приведенной на рис. 3.35, *в*.

Решение задачи приведено на рис. 3.36. Как и в предыдущем примере (см. п . 3.5.1), применена канальная петля обратной связи, управляемая фазовым компаратором F; модемы 1 и 2 формируют кадры, содержащие как полезные, так и служебные данные. Код Δ разбаланса фаз передается по каналу связи и воздействует на фазу выходного сигнала высокостабильного генератора G3, номинальная частота которого равна частоте генератора G2 (реально эти частоты незначительно различаются).

В результате такого воздействия разбаланс фаз поддерживается на допустимо низком уровне, сигналы на входах компаратора совпадают с достаточной точностью, генератор G3 “привязан” по частоте к генератору G2, что, в сущности, и требовалось. Иными словами, можно полагать, что сигнал с генератора G2 неким образом проходит через удлинитель и с некоторой задержкой (что в любом случае неизбежно, но неважно) достигает входа TxС компьютера DTE 2. Так что вопросительный знак на рис. 3.35, *в* следовало бы заменить восклицательным!

Отметим, что схемы, приведенные на рис. 2.4 и рис. 3.34, также можно рассматривать как частные случаи удлинителей линий передачи сигналов TxС, TxD, RxC, RxD, в которых сигнал TxС распространяется “против течения” потока данных. Особенность этих схем в том, что сигналы TxС и RxC на стыках между размещенными рядом устройствами синхронны и синфазны.

3.6. Взаимодействие удаленных устройств с непосредственной односторонней передачей синхросигнала по каналу связи

Ранее (см. рис. 2.4 и рис. 3.34) были рассмотрены две схемы получения синхронных и синфазных потоков данных. Первая схема построена на основе буферной памяти, вторая использует фазовый компаратор и канальную петлю обратной связи. Здесь представлено еще одно, менее универсальное, но, возможно, более экономичное решение той же задачи (рис. 3.37) [59]. Его основное отличие от предыдущих состоит в применении трех витых пар проводов в канале связи. Одна пара используется для передачи синхросигналов, по двум другим транслируются данные.

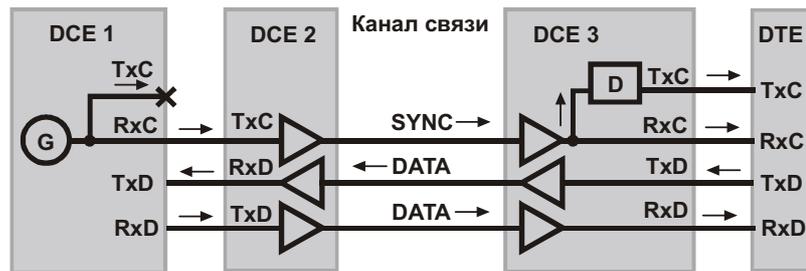


рис. 3.37. Система с непосредственной передачей синхросигнала по каналу связи: D – элемент задержки; выход TxC устройства DCE 1 не используется

Напомним, в чем заключается задача. На практике встречается ситуация, при которой устройство типа DCE (в нашем примере – устройство DCE 1) генерирует совпадающие синхросигналы RxC и TxC. Сигнал RxC сопровождает данные, выдаваемые из устройства DCE 1. Сигнал TxC задает временную сетку для принимаемых данных. Источником этих данных является устройство DTE, которое удалено от устройства DCE 1 на значительное расстояние. Поэтому нужно каким-то образом “заставить” устройство DTE подстроиться так, чтобы передаваемые им данные, подойдя к устройству DCE 1, попали в нужную временную сетку. В данном решении это достигается введением в устройство DCE 3 элемента D регулируемой фазовой задержки синхросигнала. Рассмотрим работу системы (рис. 3.37).

При передаче данных “слева – направо” они проходят из устройства DCE 1 в устройство DTE по нижней трассе. Синхросигнал от генератора G передается по верхней трассе с такими же физическими параметрами, поэтому он задерживается в той же мере, что и данные. Так что на входах RxD и RxC устройства DTE сигналы находятся в правильных фазовых соотношениях (см. рис. 1.8). (В действительности некоторый перекося сигналов, конечно, существует, но при достаточно низкой скорости передачи данных им можно пренебречь.)

Данные выдаются из устройства DTE под управлением сигнала на его входе TxC (см. рис. 1.6, б). При этом не имеет значения, в каком фазовом соотношении находятся сигналы на его входах TxC и RxC. Изменяя это соотношение, можно перемещать данные на входе устройства DCE 1 относительно синхросетки сигнала с генератора G. В частности, при соответствующей настройке элемента D можно достичь правильного фазового соотношения между сигналами на входе TxD и выходе TxC устройства DCE 1.

Элемент D регулируемой фазовой задержки синхросигнала должен обеспечивать плавный или ступенчатый сдвиг фазы выходного сигнала относительно входного в диапазоне от нуля до 360 градусов. Требуемая фазовая задержка зависит от длин и типов кабелей между устройствами и элементной базы устройств DCE 2 и DCE 3.

Предположим, что некая фирма-производитель в массовом порядке поставляет потребителям системы, показанные на рис. 3.37, причем они могут отличаться только длиной кабеля, образующего канал связи. Тогда требуемая фазовая задержка элемента D является функцией одной переменной – длины этого кабеля. Вид функции нетрудно получить расчетным путем или моделированием и подтвердить экспериментально. Если эти работы проведены, то фирма-производитель при изготовлении кабеля определенной длины может закодировать необходимую величину фазовой задержки, например в виде паяных перемычек на соединителе, как показано на рис. 3.38.

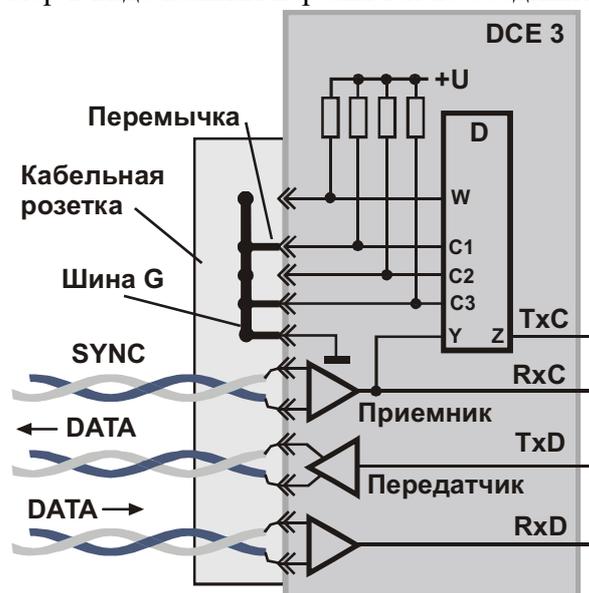


рис. 3.38. Схема сопряжения устройства DCE 3 с каналом связи

В данном примере соединитель содержит шесть пар контактов для подключения трех витых пар проводов кабеля, по которому передаются синхросигналы и данные. Кроме того, предусмотрены пять пар контактов для задания четырехразрядного кода, определяющего необходимую фазовую задержку синхросигнала. Один из контактов предназначен для подключения шины G кабельной розетки к шине нулевого потенциала устройства DCE 3. Остальные контакты розетки могут быть либо соединены, либо не соединены с шиной G с помощью индивидуальных паяных перемычек.

Если перемычка установлена, то на соответствующем управляющем входе элемента задержки присутствует сигнал лог. 0, в противном случае – сигнал лог. 1. Структура элемента задержки представлена на рис. 3.39, временные диаграммы преобразования сигналов – на рис. 3.40.

Если перемычка установлена, то на соответствующем управляющем входе элемента задержки присутствует сигнал лог. 0, в противном случае – сигнал лог. 1. Структура элемента задержки представлена на рис. 3.39, временные диаграммы преобразования сигналов – на рис. 3.40.

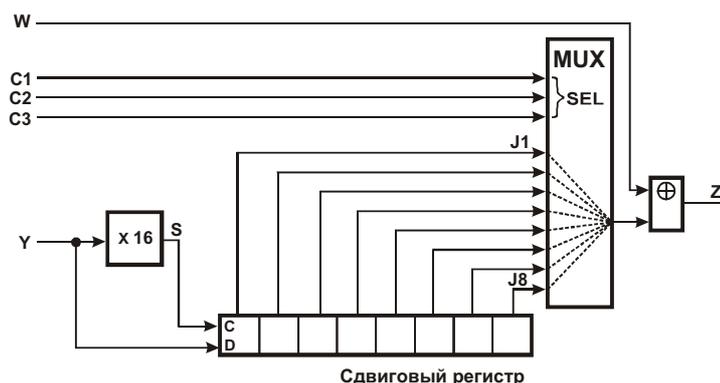


рис. 3.39. Структура элемента задержки D

Элемент задержки содержит умножитель частоты, сдвиговой регистр, мультиплексор и логический элемент Исключающее ИЛИ. Частота входного синхросигнала Y умножается на 16 и используется для продвижения информации в сдвиговом регистре.

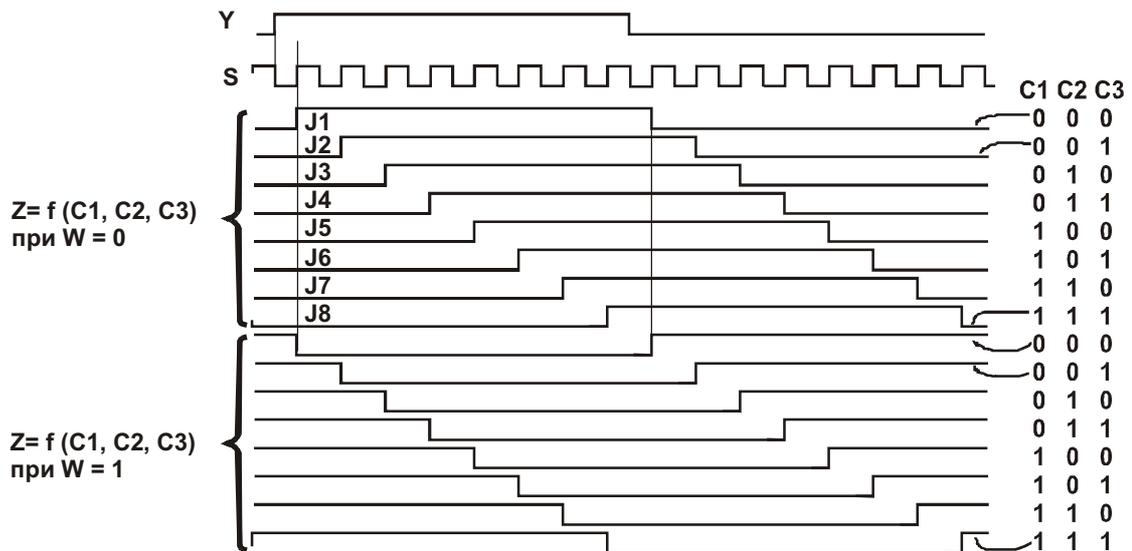


рис. 3.40. Временные диаграммы работы элемента задержки D

В зависимости от кода $C1 C2 C3$ один из выходных сигналов сдвигового регистра транслируется через мультиплексор и логический элемент Иключающее ИЛИ на выход Z . При этом сигнал инвертируется, если $W = 1$. Таким образом, шаг фазовой задержки составляет $1/16$ часть периода входного синхросигнала или $360/16 = 22,5$ градуса.

Умножитель частоты может быть выполнен по схеме, показанной на рис. 3.41.



рис. 3.41. Схема умножителя частоты

Генератор G формирует сигнал S частотой $16F$, где F – частота входного сигнала Y . Фазовый компаратор следит за совпадением фронтов входных сигналов. При обнаружении более или менее устойчивого разбаланса фронтов он корректирует напряжение U управления частотой генератора в направлении улучшения совпадения. Поэтому сигнал S привязан по фазе к сигналу Y . В данном примере предполагается, что положительный фронт сигнала Y совпадает с отрицательным фронтом сигнала S (см. рис. 3.40).

Таким образом, применение отдельной витой пары проводов для передачи синхросигнала позволяет упростить схемные решения по сравнению с приведенными ранее (см. рис. 2.4 и рис. 3.34).

3.7. Поиск начала асинхронного сообщения

Ранее (см. рис. 1.5) мы рассмотрели асинхронный способ передачи данных между устройствами. При такой передаче данные представлены потоком символов, каждый из которых снабжен служебными битами Старт и Стоп.

Напомним, что фронт старт-бита представляет собой метку времени, от которой приемник отсчитывает длительности битовых интервалов данного символа. Стоп-бит разграничивает конец текущего и начало следующего символа, но между ними могут быть и достаточно длинные паузы. Число битов данных (5...8), наличие или отсутствие контрольного разряда (дополняющего число единичных битов символа до четного или нечетного) и скорость передачи задаются при начальной настройке взаимодействующих устройств.

Асинхронность состоит в том, что приемник не знает, в какой момент поступит очередной символ. Иными словами, отсутствует единая для всего потока данных синхронизация, т. е. она имеет локальный характер – устанавливается заново всякий раз при обнаружении приемником начала очередного старт-бита.

Здесь мы рассмотрим процесс установления синхронизации приемника с передатчиком на уровне передачи сообщений, т. е. групп символов [24]. Такие группы могут иметь переменную длину, например в диапазоне от 1 до 106 символов.

Как показано на рис. 3.42, сообщения разделены паузами (масштаб отображения нарушен для повышения наглядности рисунка; в действительности паузы могут быть не столь большими). В данном примере символы, составляющие сообщение, имеют следующую структуру: старт-бит, байт (D0 – D7) данных, стоп-бит и предшествующий ему признак первого (или не первого) символа в группе. Этот признак определен следующим образом: первому символу сообщения соответствует напряжение низкого уровня; последующим – напряжение высокого уровня.

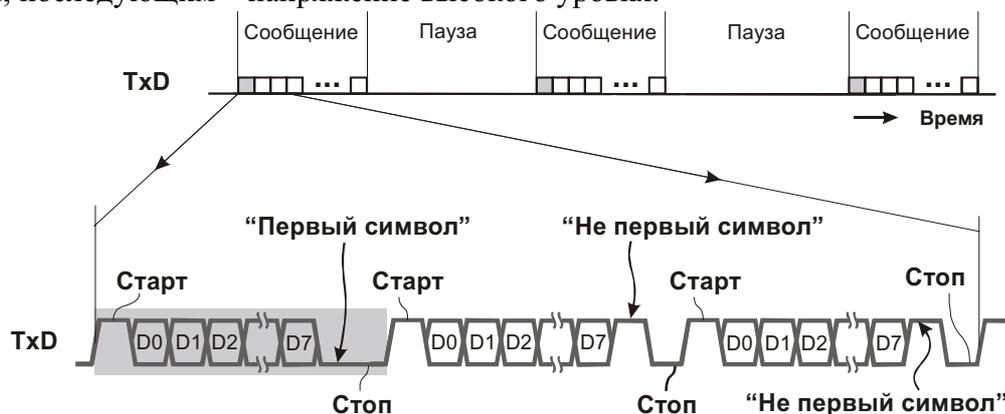


рис. 3.42. Передача асинхронных сообщений по последовательному каналу связи

Чтобы распознать начало сообщения, приемник анализирует состояние линии TxD. Пауза между сообщениями должна превышать некоторый заранее заданный интервал, например длительностью 10 мкс, поэтому на первом этапе поиска приемник дожидается ее появления (при этом паузы с меньшей длительностью не принимаются во внимание).

На втором этапе приемник анализирует полученный после паузы символ и убеждается в том, что он помечен признаком “Первый символ”. Если это не так, то констатируется ошибка. На последующих этапах проверяется длительность возможных пауз между символами (она не должна превышать, например, 8 мкс) и наличие признаков “Не первый символ”. Невыполнение этих условий рассматривается как ошибка передачи.

Такая многоэтапная проверка позволяет повысить надежность распознавания границ сообщений и обеспечить обнаружение ошибок.

3.8. Обнаружение и исправление ошибок синхронизации при передаче непрерывного асинхронного потока данных

При асинхронной передаче данных между устройствами используются биты Старт и Стоп, обозначающие начало и конец информационной посылки. Если такие посылки следуют вплотную друг к другу, то биты Старт и Стоп “прячутся” среди информационных битов, т. е. становятся внешне неотличимыми от них. Может случиться так, что приемник по каким-либо причинам потерял синхронизацию с передатчиком либо включился в работу в то время как плотный поток асинхронных посылок уже присутствует на входе устройства.

Чтобы отыскать положение стартовых битов в потоке данных, можно накопить и обработать статистику перепадов сигналов между битовыми интервалами. На границах между стоп- и старт-битами всегда должен наблюдаться отрицательный фронт сигнала. Но для статистического анализа нужен достаточный “интеллект” приемника и, возможно, большое время наблюдения, так как массив передаваемых данных может быть представлен повторяющимися посылками, например вида 1010101. В данном случае будут обнаружены регулярно повторяющиеся информационные перепады сигнала, неотличимые от стартовых. Нельзя ли упростить решение задачи вхождения приемника в синхронизм с передатчиком?

Сначала небольшое отступление. Как показано в примере на рис. 3.43, при неправильной синхронизации приемник расценивает бит 2 кода 1000110 как стартовый (ST). Бит 3 передаваемых данных воспринимается как бит 1, бит 4 – как бит 2 и т. д.



Рис. 3.43. Восстановление правильной синхронизации без обнаружения ошибки. ST и SP – старт- и стоп-биты информационных посылок

Однако благодаря передаче кода 1111111 правильная синхронизация восстанавливается. Действительно, с точки зрения приемника этот код неотличим от обычной паузы между посылками. Последующий отрицательный фронт сигнала расценивается (на этот раз справедливо) как начало стартового бита очередной информационной посылки. Можно убедиться в том, что передача кода 1111111 гарантирует восстановление синхронизации при любом возможном смещении посылки из-за несвоевременного старта.

Отметим, что приемник может “не знать”, что он работал неправильно, если не зафиксированы ошибочные (нулевые) значения битов Стоп (SP). В процессе перехода к правильной синхронизации один из информационных битов кода 1111111 рассматривается приемником как стоп-бит, имеющий правильное логическое состояние. Поэтому момент перехода к правильной синхронизации не регистрируется, что усложняет последующую локализацию ошибок протоколами более высоких уровней.

В рассмотренном далее решении [62] для восстановления правильной синхронизации и обнаружения этого события используется периодическая передача служебного кода 0000000 (рис. 3.44). Этот код передается, например, в каждой тысячной посылке. При этом соблюдаются определенные правила, позволяющие отличить служебный код от обычного нулевого кода данных.

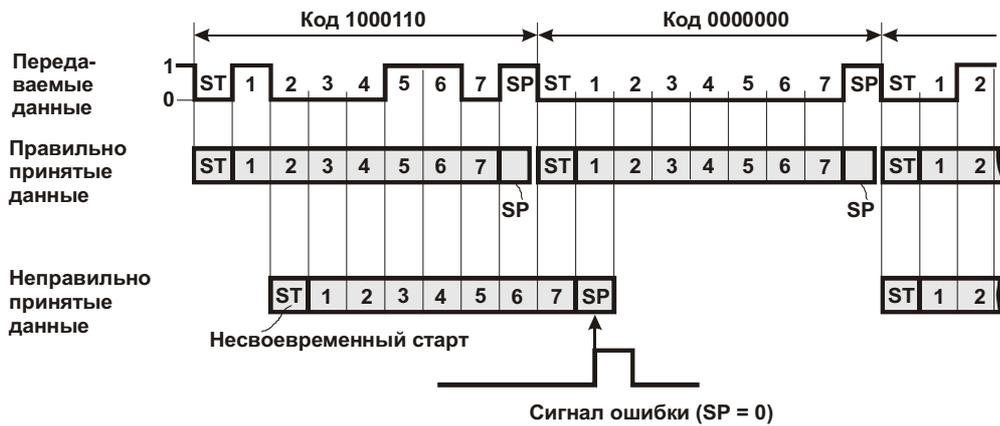


рис. 3.44. Восстановление правильной синхронизации с обнаружением ошибки

Из рисунка видно, что код 0000000, так же как и код 1111111, применим для восстановления правильной синхронизации. Однако концовка неправильно принятой посылки формируется при взаимодействии с кодом 0000000 и поэтому будет обязательно содержать ошибочный стоп-бит ($SP = 0$). Это позволяет сформировать сигнал ошибки или его программный аналог. Таким образом, передатчик может периодически вводить в передаваемый поток данных служебные нулевые коды, предназначенные для обнаружения и коррекции возможных ошибок синхронизации приемника.

Чтобы отличить служебный нулевой код от нулевых данных, можно воспользоваться одним из следующих правил.

1. Нулевые данные (в отличие от служебного кода) дублируются. Так, вместо кода данных 0000000 передается пара кодов 0000000, 0000000. Приемник выделяет парные нулевые коды, отбрасывает дубли и рассматривает оставшиеся коды как данные. Непарные нулевые коды отбрасываются как служебные.

2. В каждую посылку, как обычно, вводится разряд контроля по четности (нечетности). Нулевые данные не дублируются. Они сопровождаются правильными контрольными разрядами. В отличие от нулевых данных, служебные нулевые коды преднамеренно снабжаются неправильными контрольными разрядами. Приемник игнорирует эти коды как не содержащие полезной информации.

Можно предложить и иные способы разделения служебных и полезных данных для более надежной синхронизации в условиях повышенного уровня помех.

3.9. Распознавание межбайтовых границ в непрерывном синхронном потоке данных

В непрерывном синхронном потоке данных должны быть выделены какие-либо ориентиры для распознавания структурных единиц: кадров, ячеек, пакетов и т. п. Так, для распознавания межкадровых границ в синхронный поток данных обычно вводят группы служебных битов – флаговые коды (см гл. 7). Кадр может содержать сотни и тысячи битов, а его флаговый код может быть представлен двумя байтами. При таком соотношении числа служебных (флаговых) и полезных битов данных накладные расходы невелики, но с уменьшением длины кадра они растут.

В рассмотренных далее решениях каждый кадр переносит небольшое число битов полезных данных. Разграничение байтов с помощью многобитных флаговых кодов в данном случае нецелесообразно из-за плохого соотношения числа служебных и полезных битов кадра. Предлагаются два более эффективных способа разграничения байтов. Первый способ (п. 3.9.1) состоит в том, что в качестве флагового используется инвертированный бит данных. Второй способ (п. 3.9.2) основан на преобразовании байта в

20-разрядный кадр. Это позволяет ускорить процесс установления синхронизации приемника с передатчиком и обнаруживать ошибки при передаче данных.

3.9.1. Схема с дублированием и инвертированием одного бита данных

В системе передачи данных [27] передатчик (рис. 3.45) преобразует непрерывный поток байтов в непрерывный поток битов. Байт содержит биты DATA IN 1 – DATA IN 9, например, восемь информационных и один контрольный. Очередной байт параллельно загружается в сдвиговый регистр RG и затем последовательно выдвигается из него в линию связи через усилитель (усилитель на рисунке не показан). При этом для разграничения байтов выходного потока данных SDATA применен метод, во многом схожий с известным методом формирования асинхронных информационных посылок, когда они обрамляются служебными битами Старт и Стоп (см. п.1.2).

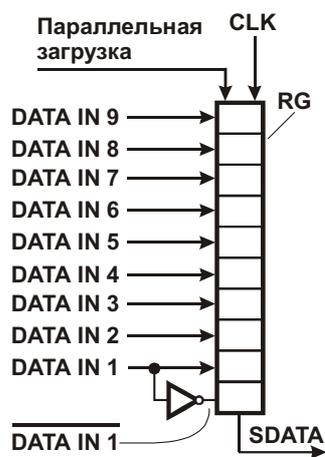


рис. 3.45. Схема формирования потока данных SDATA передатчиком

Данные из сдвигового регистра выдвигаются в такой последовательности. Сначала в линию поступает инвертированный бит $\text{NOT}(\text{DATA IN } 1)$, затем этот же бит DATA IN 1 без инверсии. Вслед за ним выдвигаются остальные биты DATA IN 2 – DATA IN 9. Таким образом, начало передачи байта сопровождается гарантированным изменением уровня сигнала. Подобно переднему фронту бита Старт в асинхронной посылке это изменение уровня сигнала используется для подстройки синхронизации между приемником и передатчиком.

Отметим, что здесь подстройка синхронизации ведется приемником по переднему фронту первого информационного бита, а не по переднему фронту бита Старт. Иными словами, момент подстройки вплотную приближен к первому полезному биту данных, а не удален от него на один бит, как в асинхронной посылке. Это повышает точность построения синхросетки приемником. Еще одно преимущество: отсутствует бит Стоп. (Этот бит был введен в асинхронную посылку в качестве антипода бита Старт для создания гарантированного флагового перепада уровней сигнала в линии.)

Схема преобразования последовательного потока данных в параллельный (рис. 3.46), размещенная в приемнике, содержит регистры RG 1 и RG 2, а также генератор и формирователь синхросигналов. Регистр RG 1 состоит из десяти D-триггеров с индивидуальными входами C синхронизации. Бит данных с входа D заносится в триггер по положительному фронту сигнала на входе C. Регистр RG 2 принимает девятиразрядный код по положительному фронту на общем входе C синхронизации.

В установленном режиме (когда достигнута устойчивая синхронизация приемника с передатчиком) данные SDATA последовательно записываются в триггеры регистра RG 1. После заполнения этого регистра данные переписываются в регистр RG 2, на выходах которого формируется поток байтов “DATA OUT”, эквивалентный исходному потоку “DATA IN”. Особая роль в поддержании синхронизации между передатчиком и приемником отведена выделенному на рисунке серым фоном балансному триггеру регистра RG 1. Он обеспечивает привязку фронтов синхросигнала от внутреннего генератора к центрам битовых интервалов входного сигнала SDATA. Процесс приема и преобразования данных поясняется временными диаграммами, приведенными на рис. 3.47.

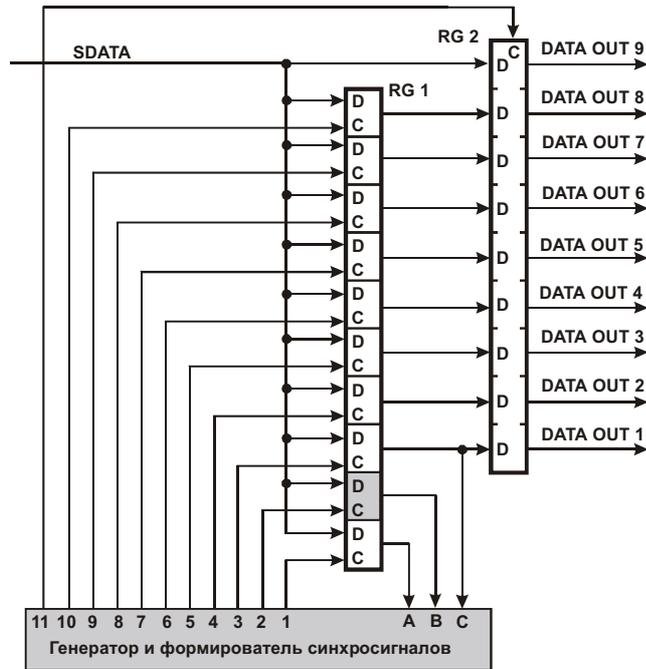


рис. 3.46. Схема преобразования приемником последовательного потока данных в параллельный

Внутренний генератор приемника формирует непрерывный периодический сигнал G . В данном примере его частота в два раза превышает частоту сигнала CLK и может регулироваться в небольших пределах. Сигналы 1 – 11 формируются из сигнала G и поэтому привязаны к нему. Положительные фронты сигналов 1, 3 – 11 соответствуют центрам битовых интервалов принимаемых данных. Положительный фронт сигнала 2 в идеальном случае совпадает с положительным или отрицательным фронтом сигнала переключения между инвертированным и прямым битами $SDATA$ 1. При этом в балансном триггере с равной вероятностью может быть зафиксирован лог. 0 или лог. 1.

Если синхронизация уже достигнута, то она поддерживается следующим образом. В каждом цикле приема байта сразу после обновления сигналов A , B и C они анализируются. Основное условие правильной синхронизации – противоположность логических значений сигналов A и C . При его выполнении проверяется своевременность формирования сигнала записи бита в балансный триггер. Если $B = A$, т. е. если запись бита данных в балансный триггер произошла слишком рано, то частота сигнала G чуть уменьшается. При $B = C$, т. е. в ситуации, когда запись бита данных в балансный триггер произошла слишком поздно, частота этого сигнала в той же мере очень незначительно увеличивается.

В результате такой регулировки положительный фронт сигнала 2 на входе синхронизации балансного триггера постоянно и в небольших пределах колеблется около теоретического положения, соответствующего началу битового интервала $SDATA$ 1. При этом положительные фронты сигналов 1, 3 – 11 в среднем во времени точно совпадают с центрами битовых интервалов.

Если в исходном состоянии синхронизация не достигнута, то частота сигнала G не корректируется. В отсутствие коррекции эта частота не может точно соответствовать удвоенной скорости передачи данных по линии и отличается от нее, например, на несколько герц. Разность частот приводит к медленному дрейфу группы сигналов G , 1 – 11 относительно «неподвижного» сигнала $SDATA$.

По мере дрейфа синхросетки положительный фронт сигнала 2 последовательно и достаточно медленно проходит через границы между битами. Одна из этих границ –

искомая. Она соответствует началу байта. Остальные границы разделяют информационные биты внутри байта.

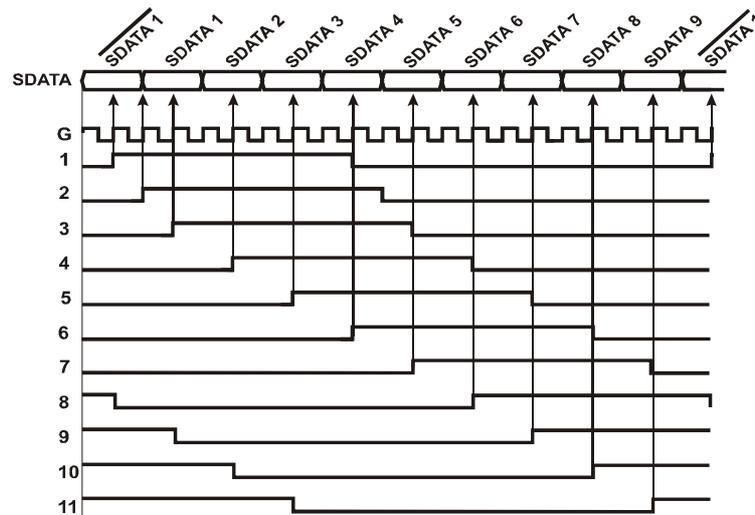


Рис. 3.47. Временные диаграммы преобразования последовательного потока данных в параллельный

Чтобы отличить искомую границу от остальных, необходим некоторый “интеллект” приемника. Прохождение искомой границы характеризуется следующими признаками.

1. При подходе к границе сигналы А и С устойчиво (статистически достоверно) противофазны. Сигнал В устойчиво совпадает с сигналом А (или С в зависимости от направления дрейфа синхросетки).

2. После прохождения границы сигналы А и С остаются устойчиво противофазными, но сигнал В теперь устойчиво совпадает с сигналом С (или А в зависимости от направления дрейфа синхросетки).

После обнаружения первого, а затем и второго условия приемник приостанавливает дрейф синхросетки путем коррекции частоты сигнала G в нужном направлении, как было описано ранее. Временная диаграмма приходит в соответствие показанной на рис. 3.47.

Можно несколько усложнить схему и применить более интеллектуальные способы установления и поддержания синхронизации. Например, вместо одного балансного триггера можно ввести группу триггеров для более точной “ориентации во времени” и использовать более мелкую временную сетку. При этом частоту (фазу) сигнала G можно не корректировать, если она находится в допустимых пределах. Возможны также разные виды реакции на обнаружение ошибок.

Конечно, не исключено ложное опознание границ байтов, если передаваемая информация регулярна и содержит постоянный перепад сигналов по крайней мере в одной паре соседних информационных битов. Но, вероятнее всего, передача такого специфического информационного блока не может быть “бесконечной”. Рано или поздно ошибка определения межбайтовых границ обнаружится с последующим установлением правильной синхронизации. (Подобное справедливо и по отношению к непрерывному потоку асинхронных посылок с битами Старт и Стоп.) Напомним, что для разрушения закономерностей в передаваемой информации применяют скремблирование (см. п. 8.4).

3.9.2. Схема с дублированием и инвертированием байта данных

Для распознавания межбайтовых границ в непрерывном синхронном потоке данных можно применить схему кодирования, приведенную на рис. 3.48 [68].

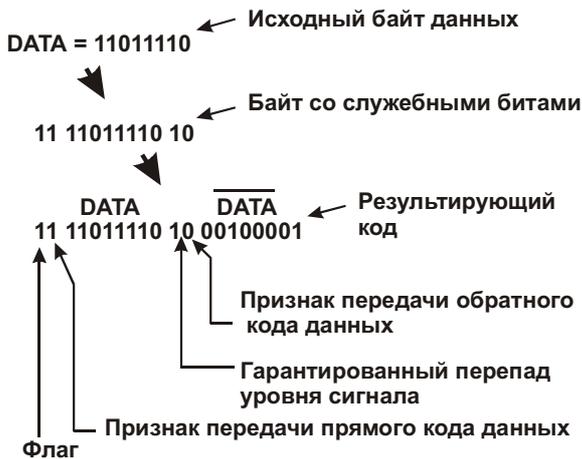


Рис. 3.48. Процесс преобразования байта данных для его передачи по линии

К исходному байту данных, в данном примере равному 11011110, добавляются две пары служебных битов: 11 и 10. Затем к полученному 12-разрядному коду добавляется инвертированный байт данных. Полученный 18-разрядный код передается в линию.

Приемник непрерывно просматривает поступающий поток данных. Схема или программа анализа данных настроена на

выявление закономерности, принятой при кодировании. Рано или поздно приемник обнаруживает, что биты, разделенные девятью тактами, совпадают с точностью до “наоборот”, но один из них (флаговый) остается неизменным. С учетом того, что вслед за флаговым битом должна следовать лог. 1 (признак передачи прямого кода данных), приемник окончательно распознаёт положение межбайтовых границ.

Комбинация служебных битов “10₂” обеспечивает гарантированный перепад уровней сигнала, который необходим для синхронизации генератора приемника. Дублирование данных позволяет обнаруживать ошибки передачи при сравнении прямого и обратного кодов. Если ввести дополнительный бит контроля по четности (нечетности), то можно корректировать одиночные ошибки передачи.

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.

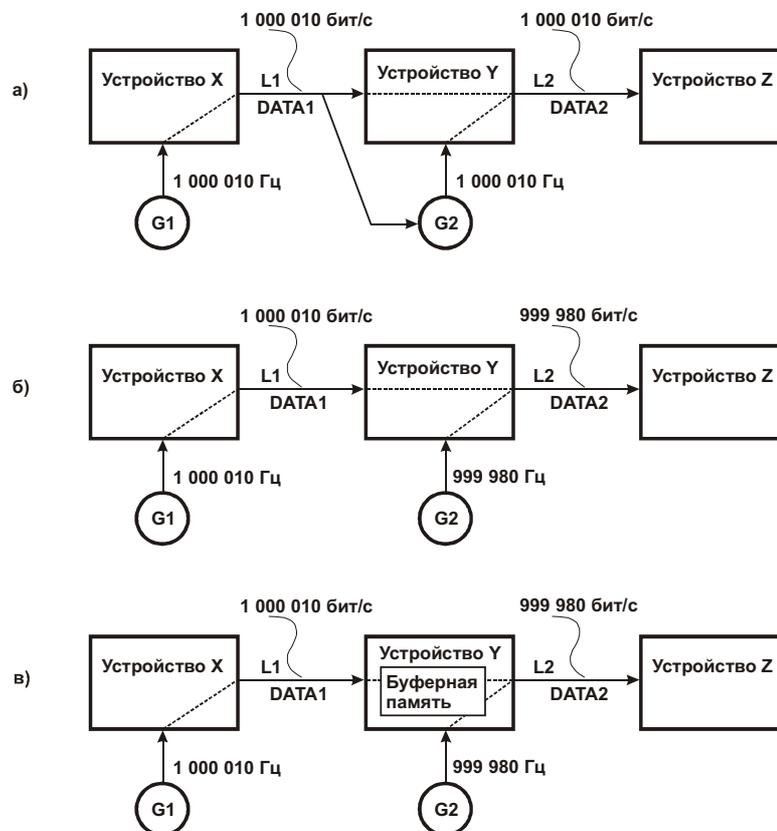


Рис. 4.1. Синхронизация передачи данных: *а* – правильная (без проскальзываний); *б* – неправильная (с проскальзываниями, когда буферной памяти нет); *в* – допустимый вариант (при условии регулирования уровня заполнения буферной памяти)

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.

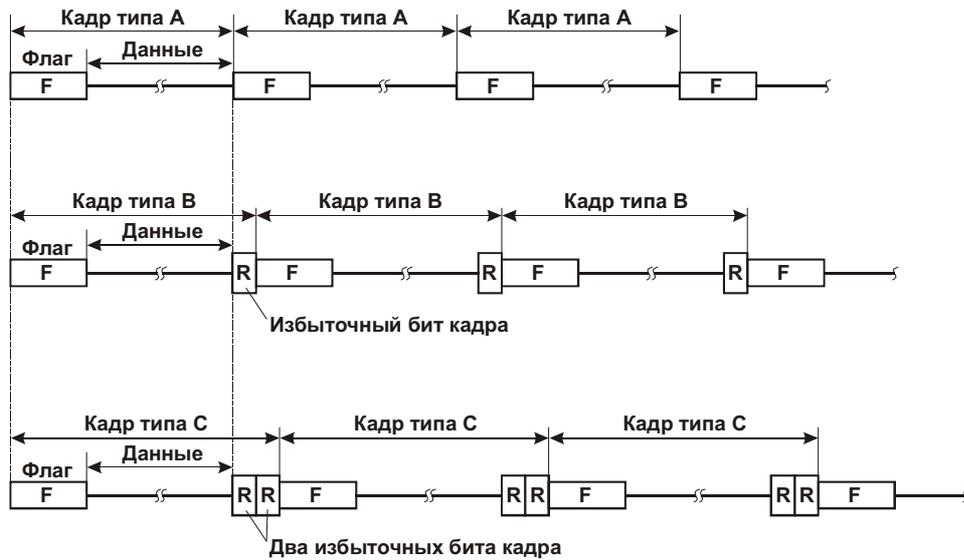


рис. 4.2. Структура кадров, используемых для предотвращения проскальзываний

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.

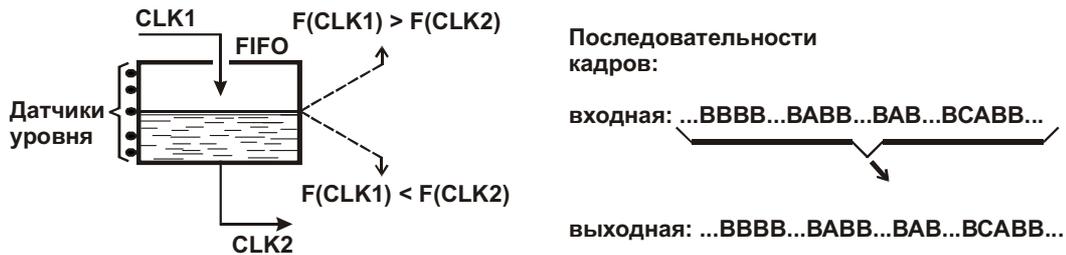


рис. 4.3. Прохождение последовательности кадров через буферную память типа FIFO в отсутствие коррекции уровня ее заполнения

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).

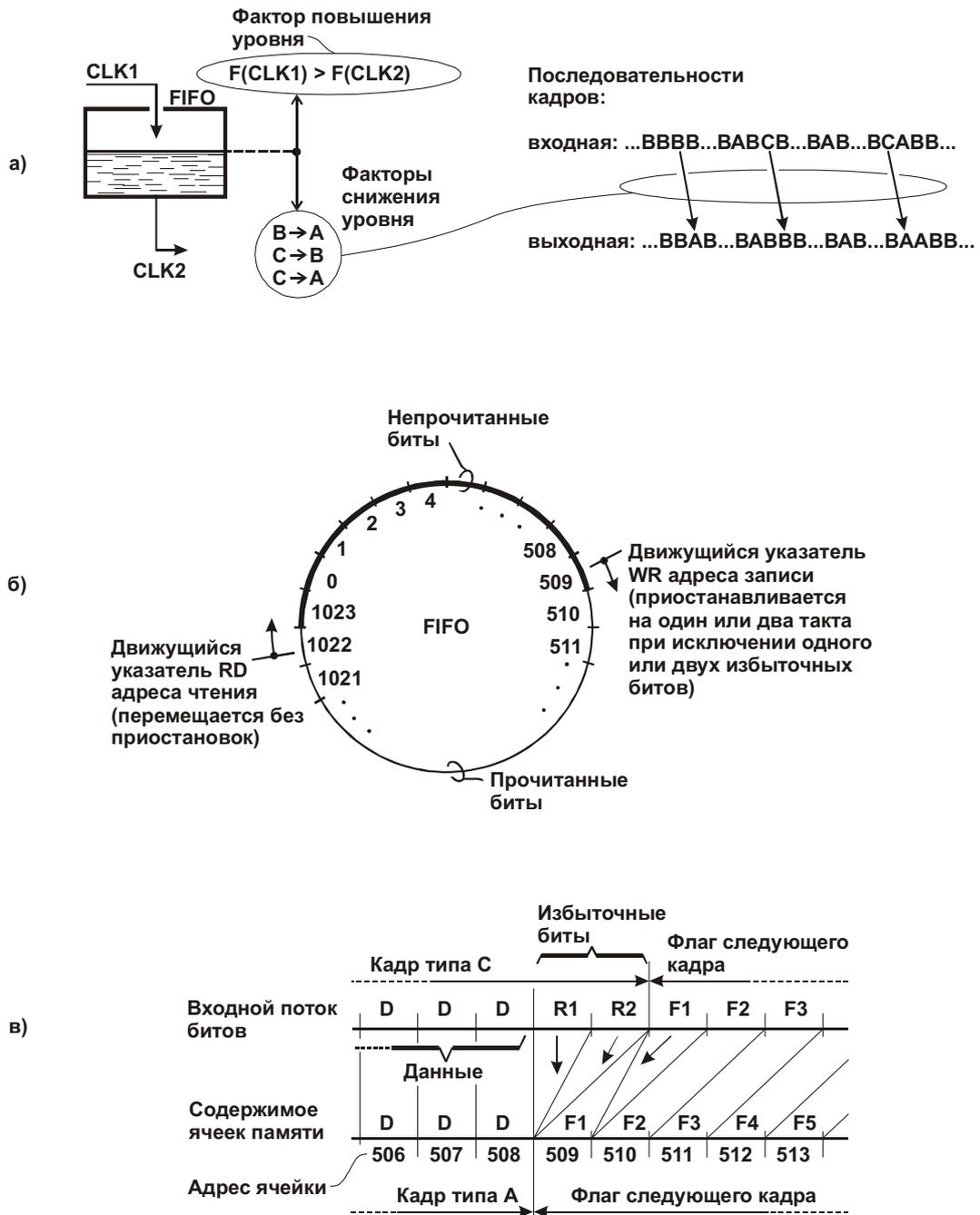


Рис. 4.4. Стабилизация уровня заполнения буферной памяти при положительной разности скоростей входного и выходного битовых потоков ($F(\text{CLK1}) > F(\text{CLK2})$): а – общая модель; б – модель динамики заполнения буферной памяти; в – модель процесса исключения двух избыточных битов

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.

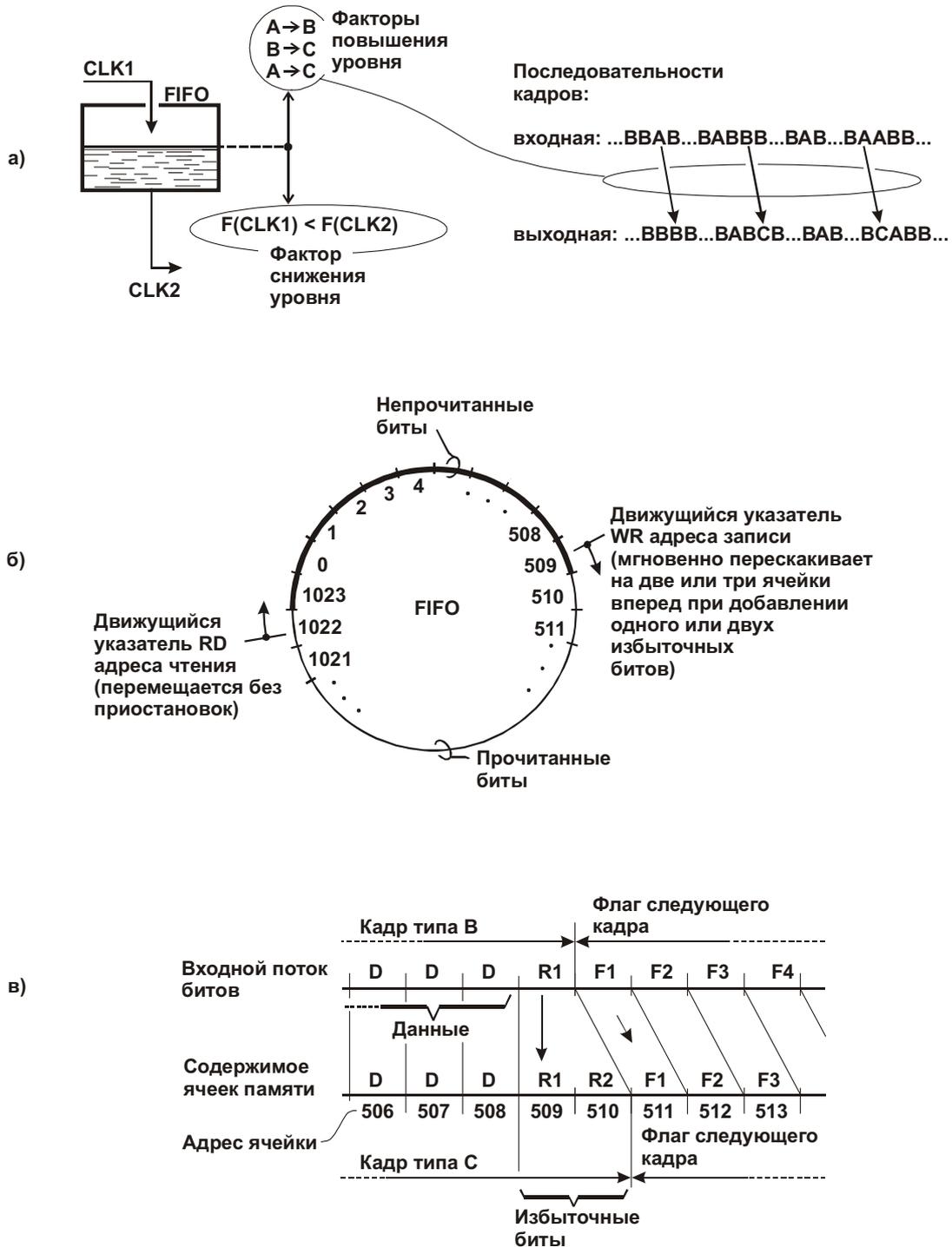


Рис. 4.5. Стабилизация уровня заполнения буферной памяти при отрицательной разности скоростей входного и выходного битовых потоков ($F(\text{CLK1}) < F(\text{CLK2})$): а – общая модель; б – модель динамики заполнения буферной памяти; в – модель процесса вставки одного избыточного бита

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).

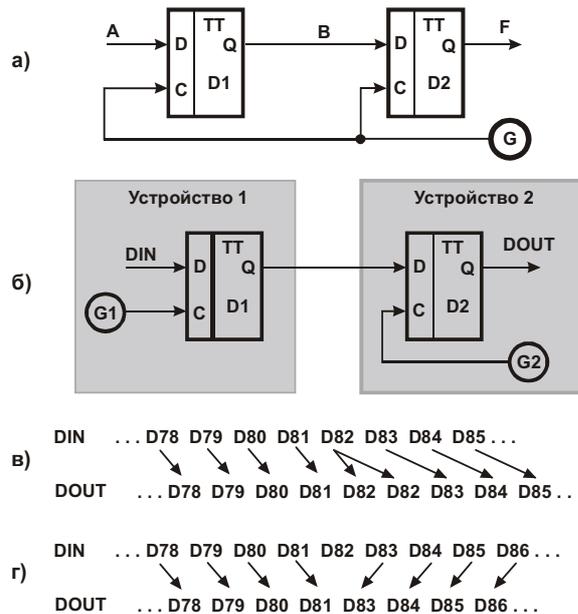


Рис. 4.6. Двухразрядный сдвиговый регистр (а) и неудачные попытки его использования для передачи данных между устройствами 1 и 2 (б – г)

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 В 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.

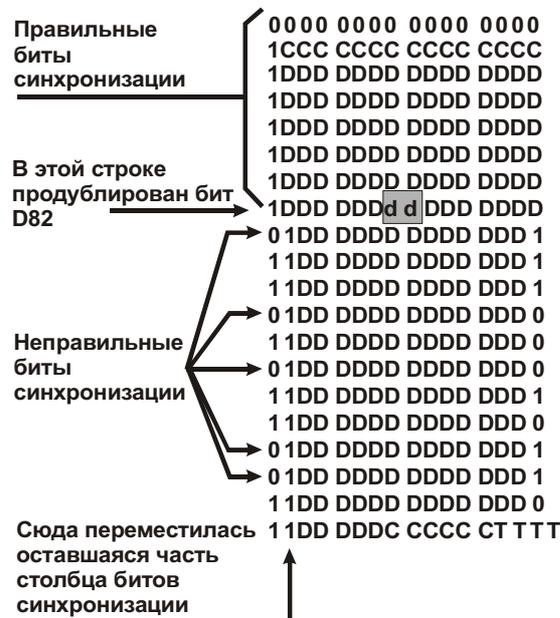


рис. 4.8. Кадр с ошибкой, вызванной проскальзыванием. Бит D82 превратился в два бита dd, последний бит T вытеснен за границу кадра

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).

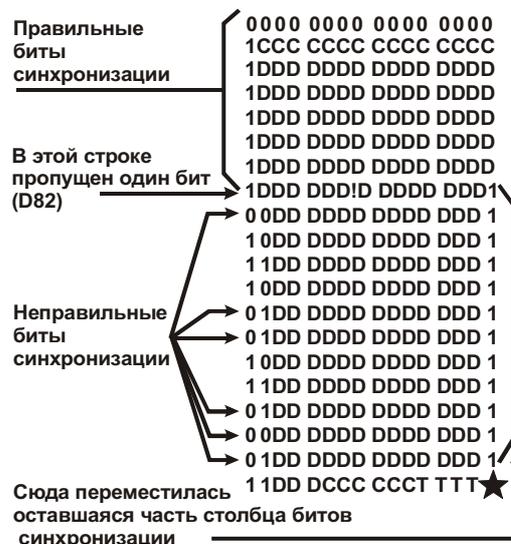


Рис. 4.9. Еще один пример кадра с ошибкой, вызванной проскальзыванием. Бит D82 исчез, “ряды сомкнулись”, в кадре появилась вакансия (помечена звездочкой)

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.

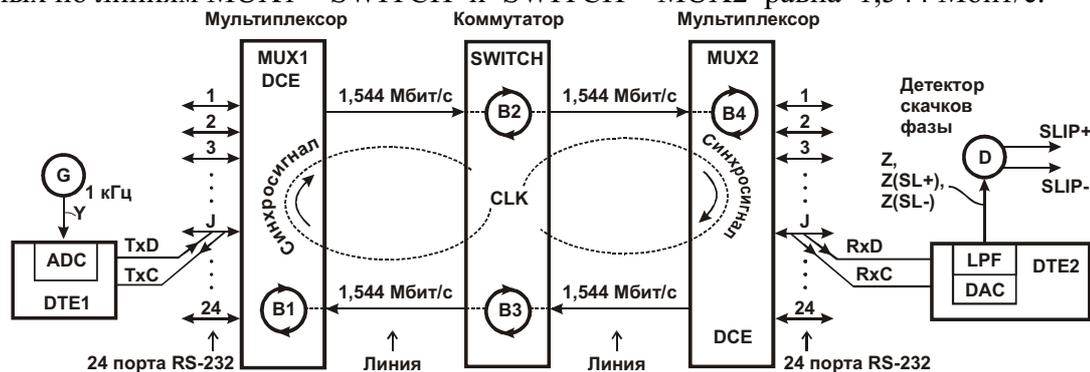


Рис. 4.10. Схема обнаружения проскальзываний в многоканальной системе передачи данных – первый вариант

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 компонует 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 $SLIP-$.

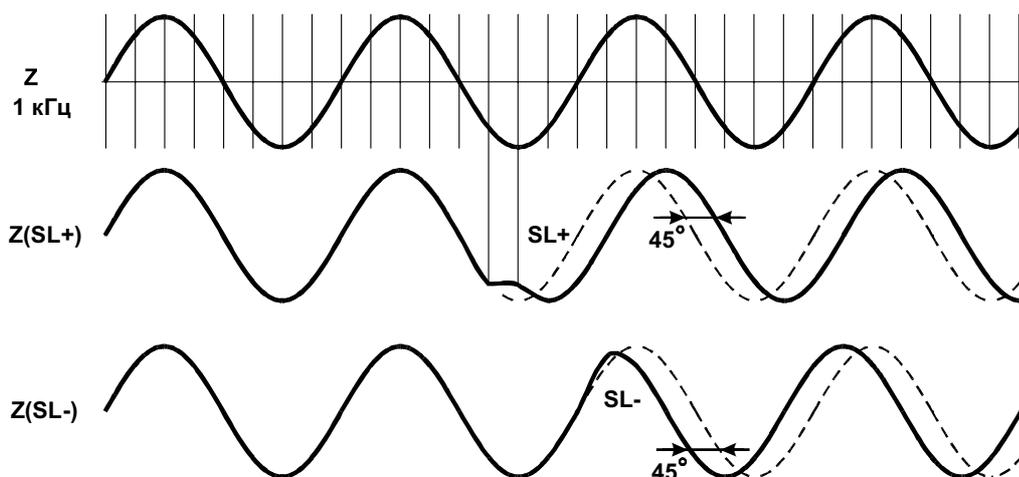


Рис. 4.11. Временные диаграммы тестового сигнала при отсутствии проскальзываний (верхняя диаграмма) и при их обнаружении (нижние диаграммы)

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.

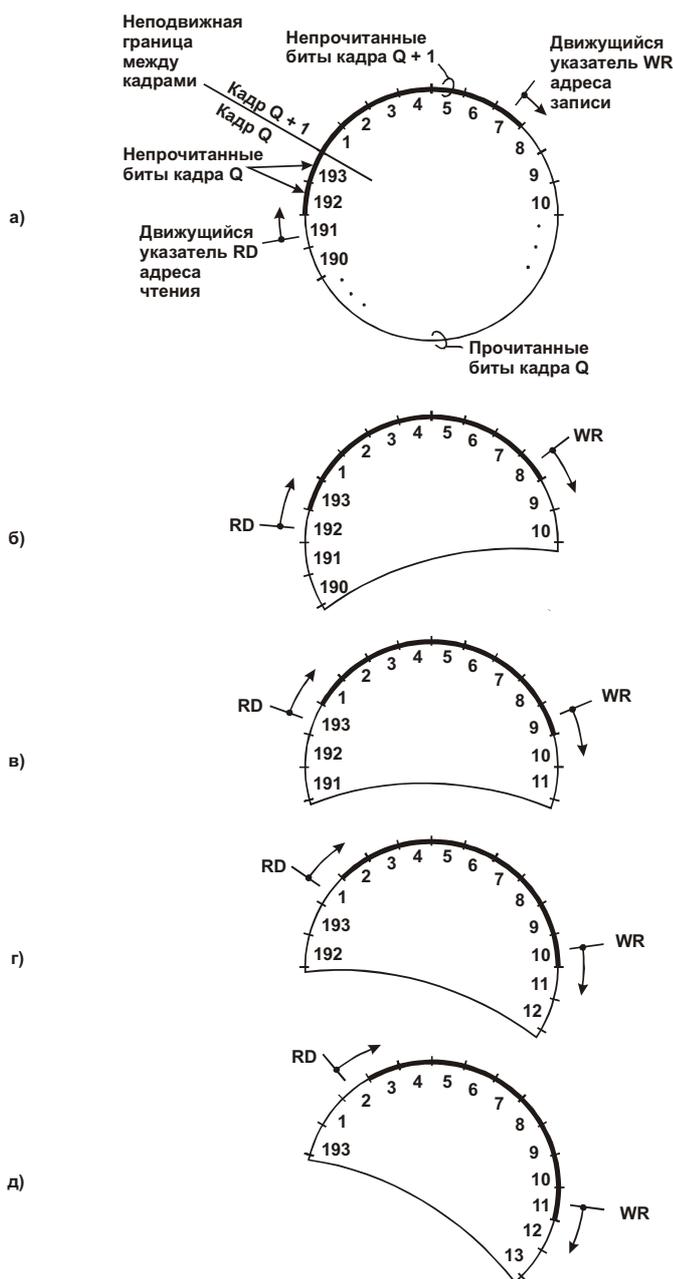


рис. 4.12. Прохождение данных через блок памяти В4 в отсутствие проскальзываний

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.

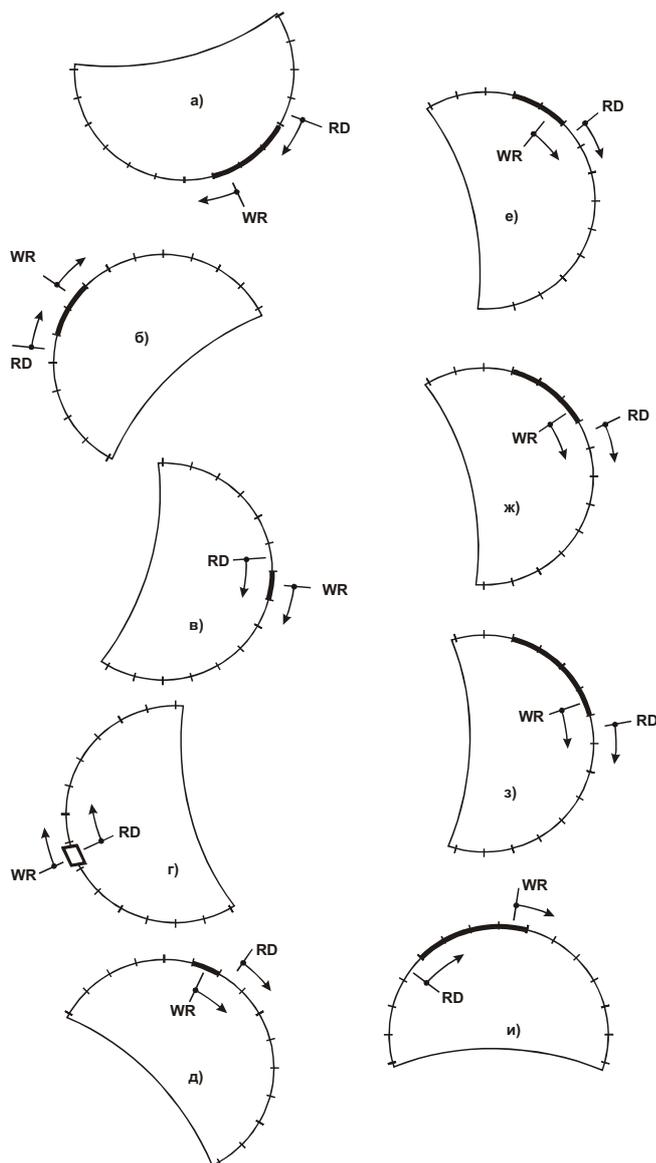


рис. 4.13. Прохождение данных через блок памяти В4 при положительном проскальзывании

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.

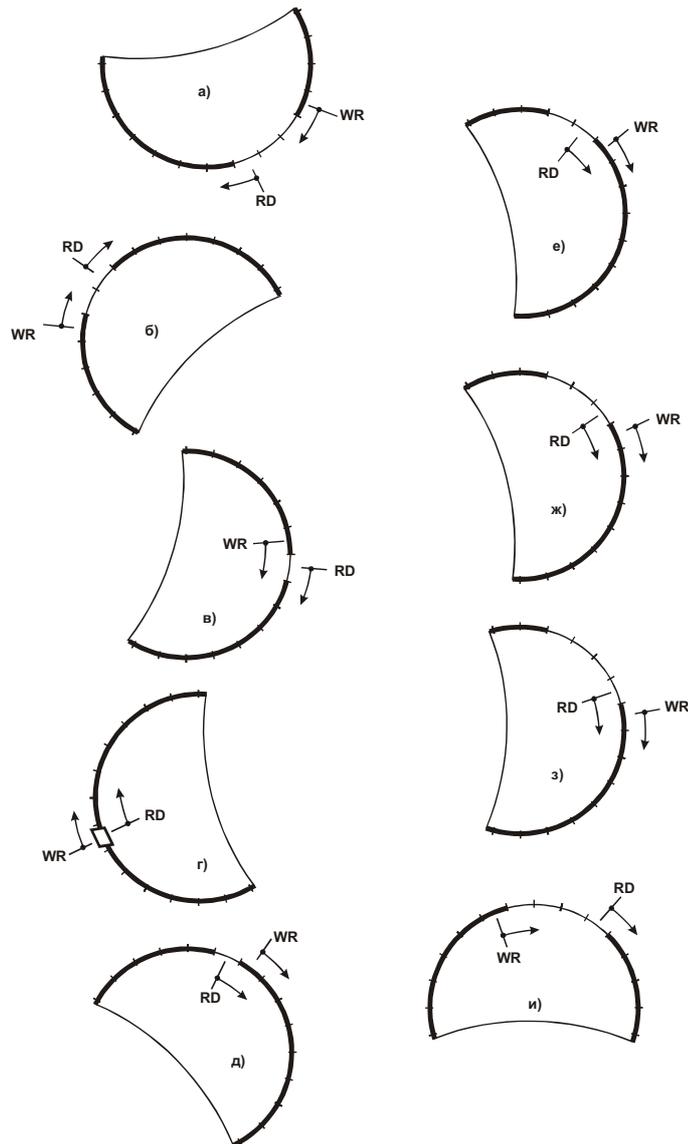


рис. 4.14. Прохождение данных через блок памяти В4 при отрицательном проскальзывании

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 мультиплексора $MUX2$ 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.

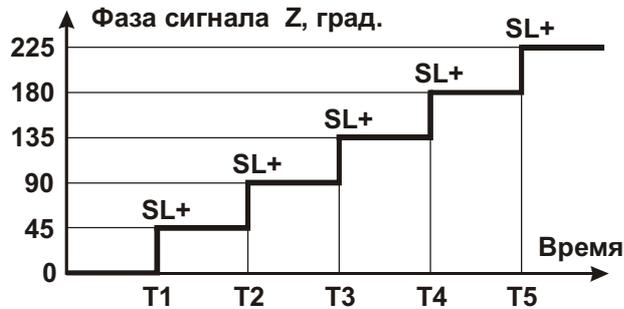


рис. 4.15. Набег фазы тестового сигнала при регистрации положительных проскальзываний SL+

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.

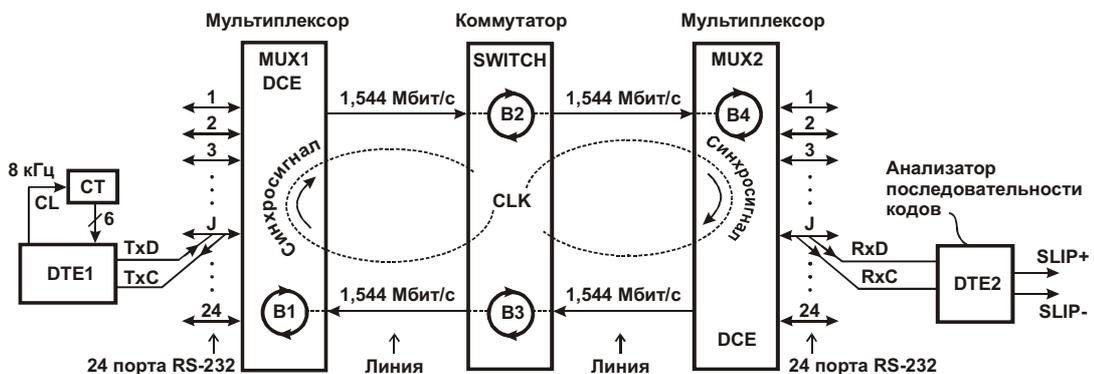


рис. 4.16. Схема обнаружения проскальзываний в многоканальной системе передачи данных – второй вариант

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.)

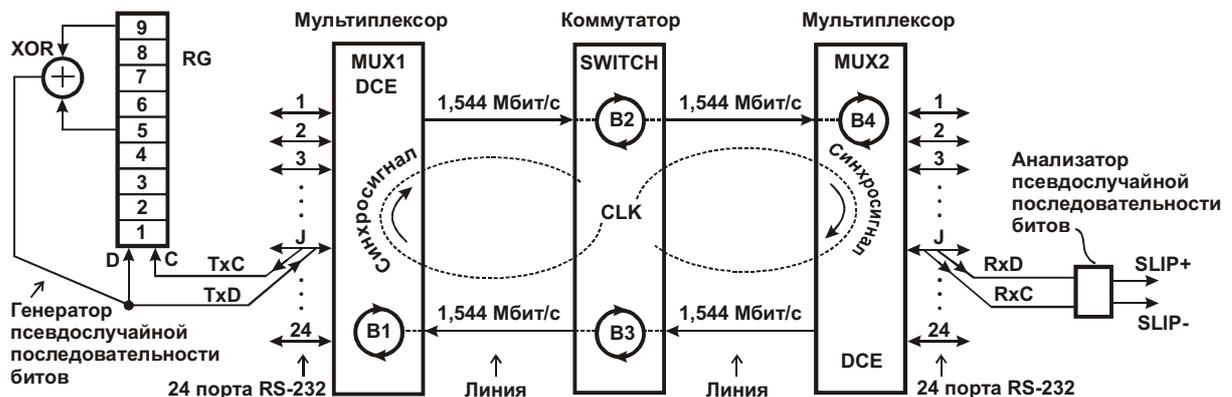


рис. 4.17. Схема обнаружения проскальзываний в многоканальной системе передачи данных – третий вариант

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.

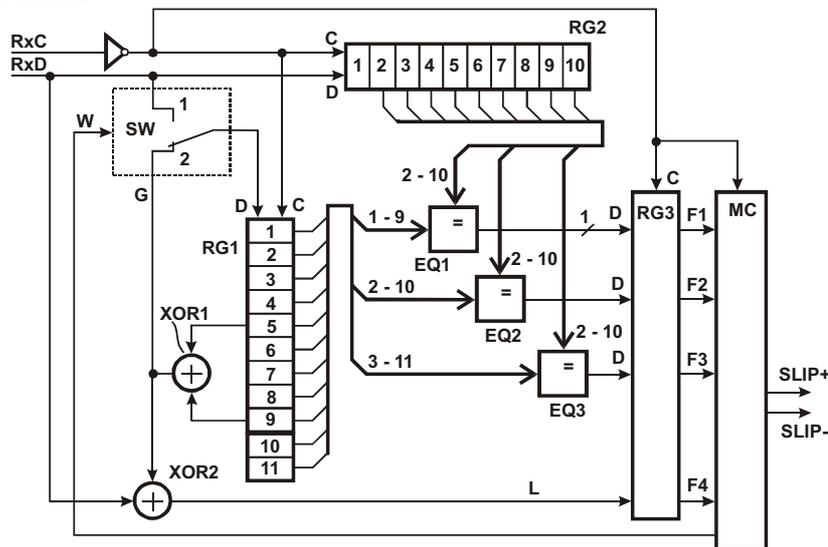


Рис. 4.18. Схема анализатора псевдослучайной последовательности битов

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.

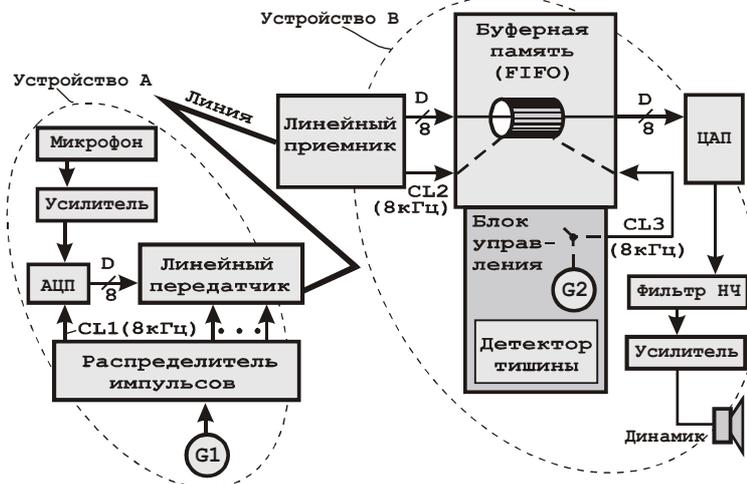


рис. 4.19. Вариант реализации идеи устранения проскальзываний путем варьирования длительностей пауз между словами или фразами

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 .

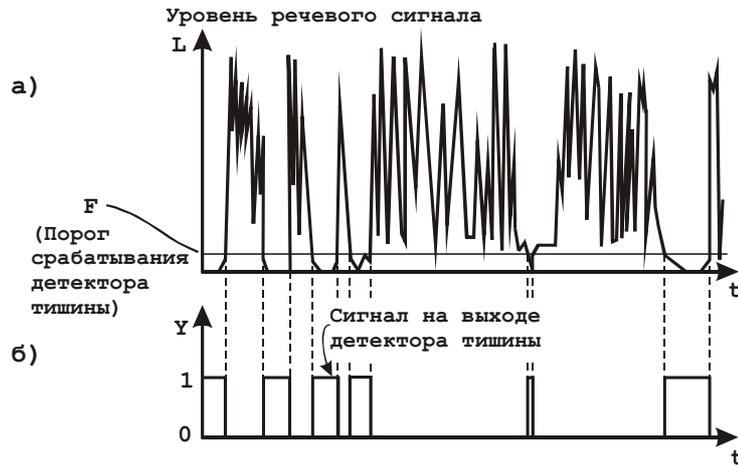


рис. 4.20. Типичный пример временной диаграммы речевого сигнала (а) и результат его оценки на предмет выявления относительной тишины (б)

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”.

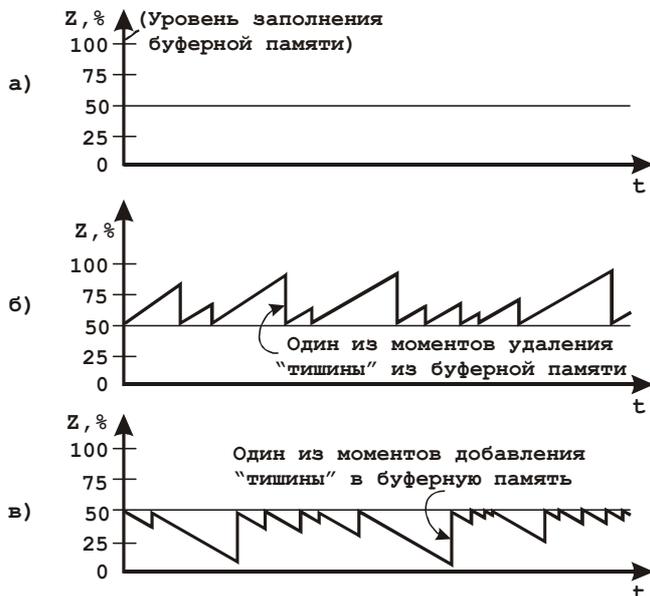


рис. 4.21. Временные диаграммы заполнения буферной памяти: а – идеальная ситуация; б – темп пополнения буфера выше темпа его рассасывания; в – темп пополнения буфера ниже темпа его рассасывания

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).

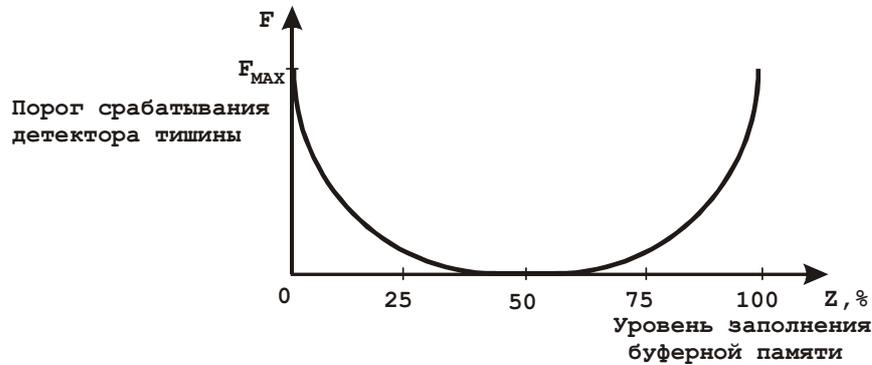


рис. 4.22. Зависимость порога срабатывания детектора тишины от уровня заполнения буферной памяти

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].

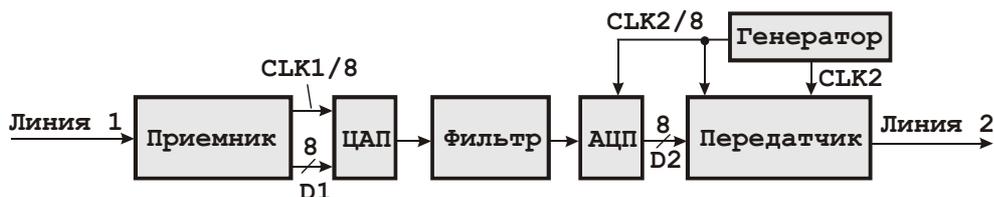


рис. 4.24. Схема промежуточного устройства

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 $Y > X$.

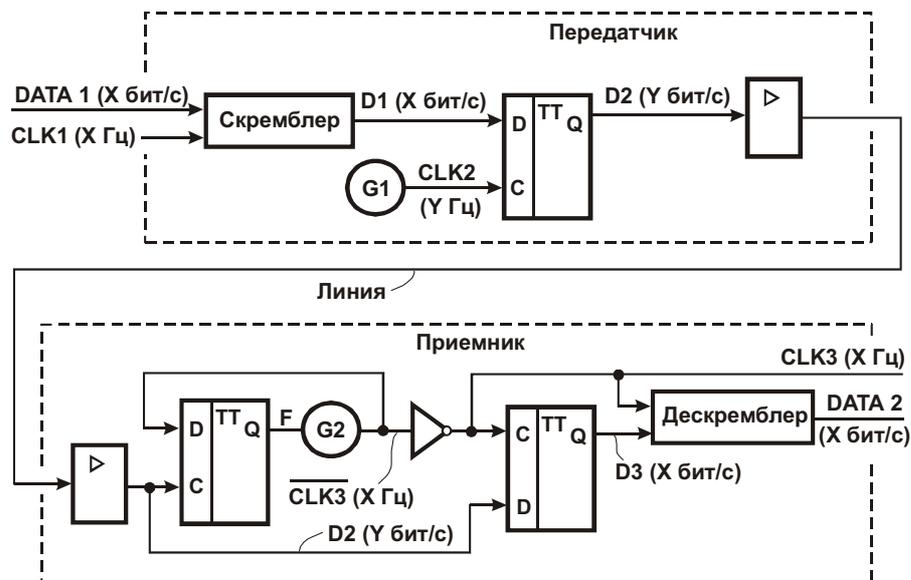


рис. 4.25. Синхронная система передачи данных. Скорости входного и выходного потоков данных (DATA 1 и DATA 2) не совпадают со скоростью потока (D2), передаваемого по линии

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.

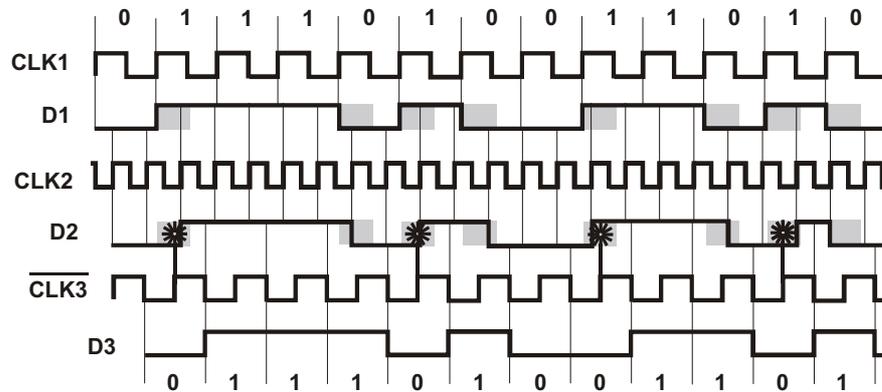


рис. 4.26. Временные диаграммы передачи данных в системе, показанной на рис. 4.25

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.

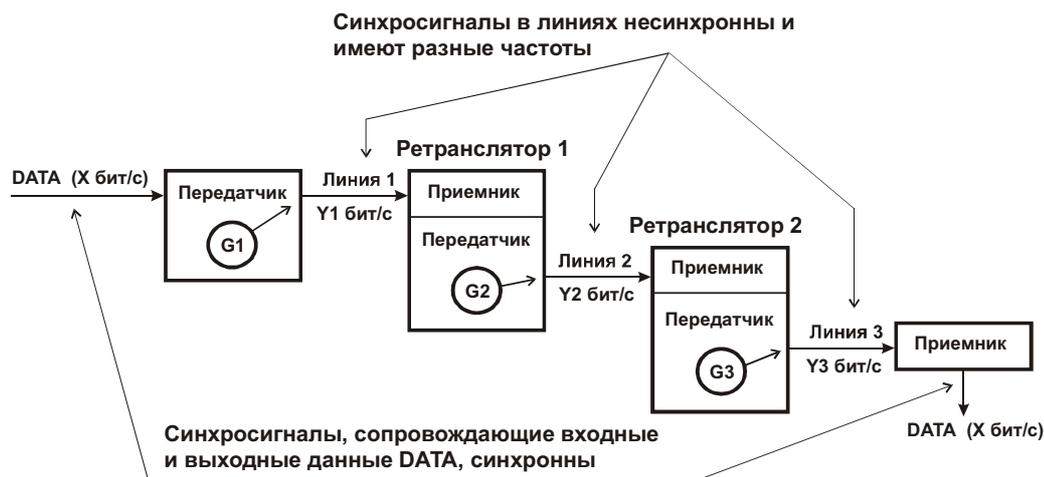


рис. 4.27. Пример сопряжения разноскоростных компонентов канала связи без использования буферов типа FIFO

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 скремброваны. 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 демуплексором 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.

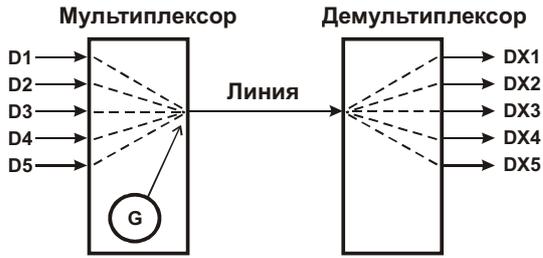


рис. 4.28. Функциональная схема системы с мультиплексированием каналов

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 демультиплексором.

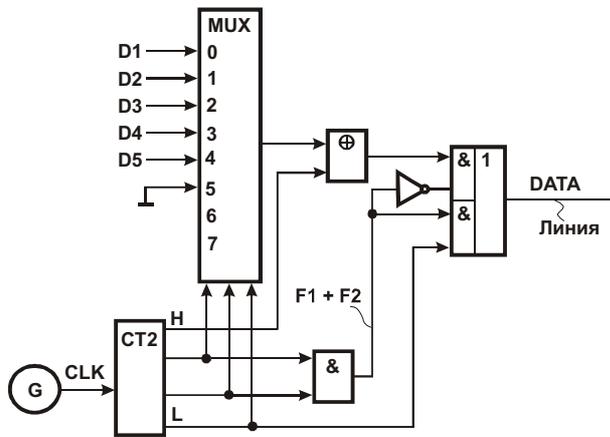


рис. 4.29. Схема мультиплексора

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).

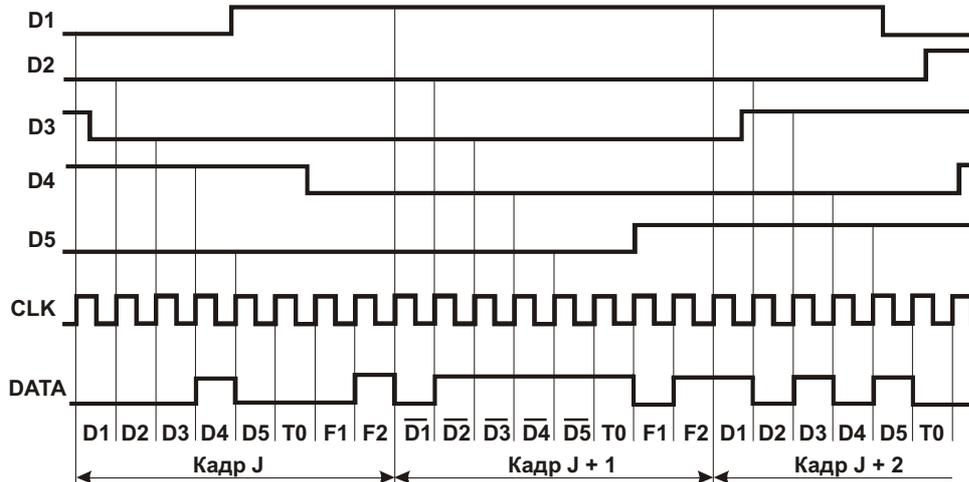


рис. 4.30. Временные диаграммы мультиплексирования сигналов

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.

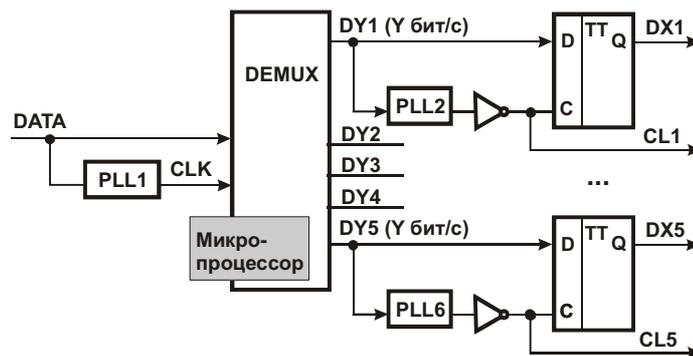


рис. 4.31. Схема демультимплексора

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.

5. Фазовые помехи

5.1. Основные виды фазовых помех

К фазовым помехам относятся упоминавшиеся джиттер (jitter – дрожание) и вандер (wander – странствие) [2], причем вандер – “враг номер один”, так как он неуклонно стремится вызвать проскальзывание, в то время как джиттер легче гасится буферной памятью типа FIFO. Тот и другой имеют одинаковую сущность, которая состоит в паразитной частотной модуляции синхросигнала сигналами высокой (джиттер) или (и) низкой (вандер) частоты. (Своего рода иллюстрацией вандера может служить звучание аккордеона, особенно если пытаться играть медленно и негромко.) Частотная граница между джиттером и вандером обычно принимается равной 10 Гц. Характер искажений поясняется рис. 5.1.

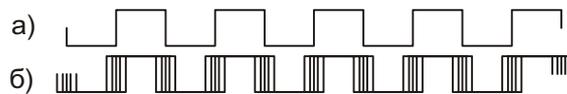


Рис. 5.1. Проявления джиттера и вандера: а – осциллограмма сигнала без помех; б – осциллограмма сигнала при воздействии джиттера или (и) вандера

Происхождение таких искажений может быть вызвано многими факторами. Некоторые из них: перекрестные помехи в линии; пульсации напряжения питания источника и приемника сигналов; неблагоприятные кодовые комбинации при формировании выходного сигнала линейным передатчиком; механическая вибрация (при которой некоторые радиоэлектронные элементы могут работать как преобразователи механической энергии в электрическую); дневные – ночные перепады температуры (сверхнизкочастотный вандер).

5.2. Идея построения адаптивного фильтра для подавления фазовых помех

Рассмотренная далее идея, в сущности, проста. Входной поток данных, подверженный фазовым помехам, можно сопоставить с непрерывным, но пульсирующим потоком S1 воды, который втекает в бак (буферную память типа FIFO), предварительно заполненный примерно до середины (рис. 5.2). При равенстве средних скоростей втекающего S1 и вытекающего S2 потоков воды уровень заполнения бака стабилен, но скорость вытекающего потока в меньшей степени подвержена пульсациям, чем скорость втекающего.

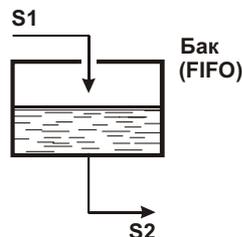


рис. 5.2. Идея подавления пульсаций скорости потока в общем виде

Теперь – то же самое, но чуть более подробно. Схема, показанная на рис. 5.3 [2], позволяет уменьшить уровень искажений входного сигнала, вносимых джиттером и вандером.

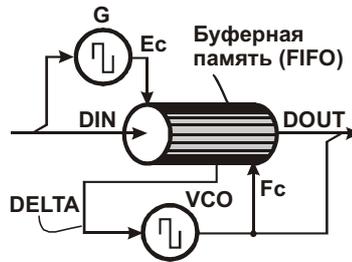


рис. 5.3. Схема фильтра для подавления паразитной частотной модуляции синхросигнала

В схеме используется буферная память типа FIFO с индикатором уровня ее заполнения. Этот индикатор формирует постоянное напряжение DELTA, которое по знаку и величине пропорционально отклонению уровня заполнения памяти от отметки “50%”.

Генератор VCO, управляемый напряжением, способен под действием сигнала DELTA изменять частоту в небольших пределах. При $DELTA = 0$ частота выходного сигнала F_c близка номинальной. При $DELTA > 0$ частота повышается, при $DELTA < 0$ – снижается.

Предположим, что в исходном состоянии буферная память заполнена на 50%, генератор G с автоподстройкой частоты выделяет из входного сигнала синхросигнал E_c , сепаратор данных (на рисунке не показан) восстанавливает данные DIN и передает их на вход памяти. Считывание данных из памяти происходит под управлением сигнала F_c .

При наличии джиттера в сигнале E_c уровень заполнения памяти совершает относительно быстрые колебания, что отражается в виде соответствующих колебаний напряжения на управляющем входе генератора VCO. Благодаря достаточной инерционности генератора и слабой крутизне его характеристики напряжение – частота, сигнал F_c на его выходе “дрожит” в меньшей степени, чем сигнал E_c , причем высокочастотные компоненты джиттера подавляются лучше, чем низкочастотные.

При наличии вандера в сигнале E_c колебания уровня заполнения памяти имеют низкую частоту, поэтому инерционность генератора VCO уже не может служить средством его подавления. Из-за большого периода колебаний возрастает опасность переполнения или опустошения памяти. (Слишком сильно увеличивать объем памяти нельзя, так как при этом неоправданно увеличивается задержка передачи данных.) В данном случае генератор VCO до некоторой степени предохраняет память от переполнений и опустошений путем соответствующего слабого повышения или снижения частоты выходного сигнала F_c . Но низкочастотные компоненты вандера, к сожалению, проходят через такой фильтр.

Интересно отметить, что огромная проникающая способность низкочастотных компонентов вандера при его распространении по телекоммуникационной сети может быть обращена во благо. Для прослеживания трасс распространения синхросигнала от некоторого источника в этот синхросигнал вводится искусственный вандер с заранее заданными параметрами. Обнаружение такого вандера в некотором удаленном узле сети свидетельствует о том, что трасса распространения синхросигнала пролегает через этот узел (см. п. 6.4).

5.3. Вариант схемной реализации аттенюатора джиттера

Напомним содержание задачи. Передатчик и приемник соединены линией связи, которая может содержать ретрансляторы или иные устройства – мультиплексоры, коммутаторы и т. п. (рис. 5.4). В приемнике имеется сепаратор, который выделяет из ли-

нейного сигнала данные DIN и сопровождающий их синхросигнал CIN. В силу ряда причин выделенные из линии сигналы DIN и CIN отличаются от идеальных. Это проявляется прежде всего в наличии джиттера. Для его подавления применен аттенюатор, структура которого приведена на (рис. 5.5) [12].

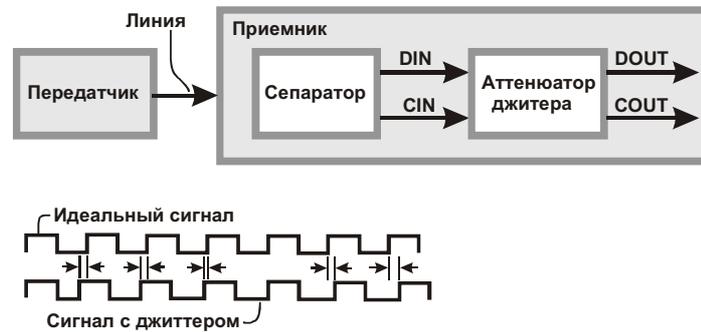


рис. 5.4. Система передачи данных, структура приемника и временные диаграммы, поясняющие проявление джиттера

В стандартах на телекоммуникационную аппаратуру введены жесткие ограничения на допустимый джиттер. Так, согласно стандарту 62411 фирмы AT&T, джиттер синхросигнала не должен превышать 2% длительности периода. Например, в стандарте T1 для скорости передачи 1,544 Мбит/с синхросигнал имеет частоту 1,544 МГц; длительность периода синхросигнала составляет примерно 648 нс, допустимый джиттер не должен превышать 12,96 нс.

Предположим, что выделенные из линии сигналы DIN и CIN не удовлетворяют указанным требованиям. Тогда задача состоит в построении аттенюатора, который должен уменьшить джиттер до допустимых пределов. Рассмотрим структуру аттенюатора.

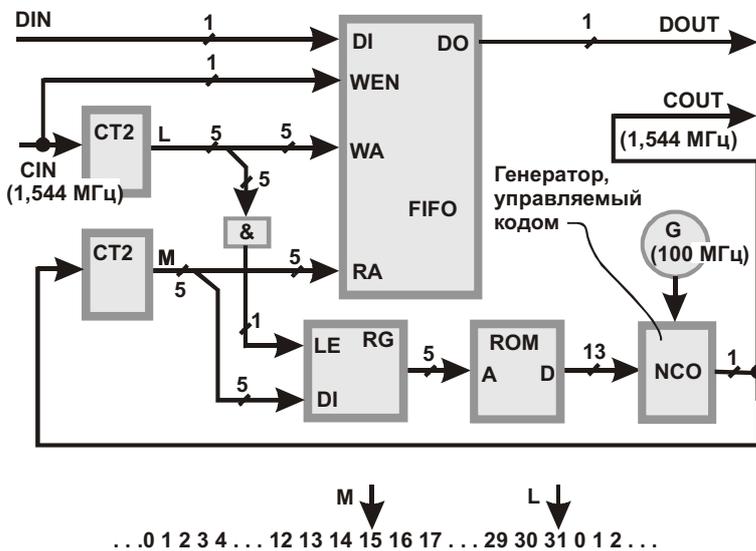


рис. 5.5. Структура аттенюатора джиттера и последовательность состояний пятиразрядных двоичных счетчиков CT2. Цифрами на схеме обозначено число проводов в соответствующих соединительных линиях

Аттенюатор содержит буферную память типа FIFO, два пятиразрядных двоичных счетчика CT2, пятиходовый логический элемент И, пятиразрядный регистр-защелку RG, ПЗУ (ROM), кварцевый генератор G и генератор, управляемый кодом (NCO). На входы аттенюатора поступают сигналы данных и синхронизации DIN и CIN с недопустимо высоким уровнем джиттера. На выходах аттенюатора формируются соответствующие сигналы DOUT и COUT с приемлемым уровнем джиттера.

Буферная память содержит 32 одноразрядные ячейки. Данные для записи в память поступают на вход DI. Адрес ячейки, в которую записываются данные, задается пятиразрядным кодом на входах WA памяти. Момент записи определяется фронтом сигнала на управляющем входе WEN. Адрес ячейки, из которой считываются данные, задается пятиразрядным кодом на входах RA. Считывание выполняется асинхронно, т. е.

не сопровождается каким-либо внешним управляющим сигналом на входе блока памяти.

Считанный бит данных поступает на выход DO. Операции записи и чтения выполняются независимо одна от другой, при этом не исключено, что адреса записи и чтения могут совпасть. Но такие ситуации возможны только при начальном вхождении устройства в синхронизм. В установившемся режиме адрес L записи “опережает” адрес M считывания примерно на половину адресного диапазона, как показано в нижней части рис. 5.5. При этом перебор адресов записи и чтения напоминает “гонку за лидером”, когда указатель M движется по “числовой оси”, соблюдая некоторую среднюю дистанцию по отношению к указателю L.

Двоичные пятиразрядные счетчики непрерывно подсчитывают число входных CIN и выходных COUT синхроимпульсов. Счет ведется по модулю 32, т. е. после заполнения счетчика кодом 11111_2 новый цикл начинается с нулевого кода. Пятиходовый логический элемент И формирует на выходе сигнал лог. 1 при $L = 11111_2$. Этот сигнал поступает на вход LE разрешения записи регистра-защелки RG и фиксирует в этом регистре код M. Таким образом, в регистре фиксируется пятиразрядный код, который, как отмечалось, в установившемся режиме соответствует примерно середине диапазона адресов 0 – 31.

Этот код поступает на адресные входы ПЗУ, в котором хранятся 32 13-разрядные константы управления частотой генератора NCO. Столь высокая разрядность констант выбрана для того, чтобы осуществлять очень “тонкую” подстройку частоты. Генератор NCO формирует выходной синхросигнал COUT путем деления входной частоты 100 МГц на 64 или 65, причем выбор того или иного коэффициента деления осуществляется динамически, на основе информации, содержащейся в 13-разрядной константе.

Сигнал от генератора G частотой 100 МГц задает дискретность моментов формирования фронтов выходного синхросигнала COUT, равную $1/100 \text{ МГц} = 10 \text{ нс}$. При “перескоке” с одного коэффициента деления частоты на другой и обратно фронты выходного сигнала будут “дрожать” с амплитудой 10 нс, но это уже допустимый джиттер. (Напомним, что норма стандарта 62411 фирмы AT&T, принятого в данном примере за основу, допускает амплитуду джиттера 12,96 нс.)

Аттенюатор функционирует следующим образом. Входные данные DIN записываются в буферную память со средней скоростью 1,544 Мбит/с. Мгновенная скорость может отличаться от средней из-за влияния джиттера, поэтому уровень заполнения (разность L – M) буферной памяти колеблется в такт с колебаниями скорости входных сигналов.

Периодически, через каждые 32 такта входного сигнала CIN, срабатывает логический элемент И, разность хода счетчиков запоминается в регистре RG. Если эта разность отклоняется от номинальной (равной 14 или 15), то константа, считываемая из ПЗУ, воздействует на генератор NCO так, чтобы, чуть увеличив или уменьшив частоту, компенсировать имеющееся отклонение.

Благодаря инерционности предлагаемой системы автоматического регулирования, выходной синхросигнал COUT отражает усредненную частоту входного сигнала CIN и подвержен джиттеру лишь в той степени, которая связана с необходимостью “тонкой” автоподстройки генератора NCO.

Схему генератора NCO и примеры построения множеств управляющих кодов можно найти в [12].

Рассмотренные решения позволяют ослабить фазовые помехи, причем со снижением их частоты степень ослабления уменьшается.

6. Методы синхронизации узлов сетей передачи данных

Эта глава посвящена методам и схемным решениям, обеспечивающим синхронную работу узлов сетей передачи данных. В общем виде задача заключается в передаче потока данных между источником и приемником через транспортную сеть, содержащую множество узлов (рис. 6.1). Источник данных, например мобильный телефон, передает в сеть кодированный сигнал, в котором данные (“оцифрованная” речь) смешаны с синхроимпульсами Sc . Мобильный телефон – приемник выделяет из полученного сигнала данные и синхроимпульсы Rc и восстанавливает речевой сигнал. Те же процессы протекают и при передаче потока данных в обратном направлении.

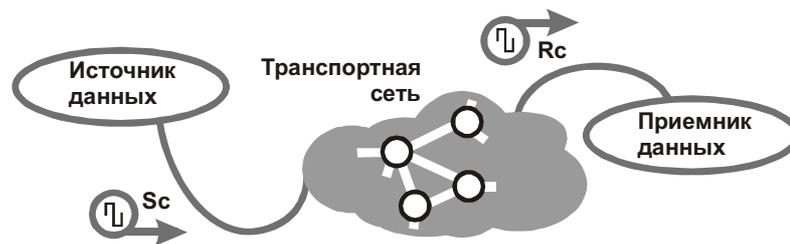


Рис. 6.1. Синхронная передача данных через транспортную сеть

Синхронизация состоит в обеспечении согласованной во времени работы источника и приемника данных. Одно из препятствий, которое нужно при этом преодолеть, связано с тем, что исходный синхросигнал Sc не может быть непосредственно передан приемнику (тогда транспортная сеть была бы просто не нужна). В общем случае этот сигнал в составе потока данных от источника вливается в более мощные потоки. В конечном счете сигнал Rc извлекается из соответствующего потока и передается в приемник. Для правильного взаимодействия узлов сети по трассе распространения потока данных необходимо выполнить некоторые условия.

1. По мере распространения сопровождающего данные синхросигнала в нем накапливаются фазовые помехи (джиттер и вандер). Поэтому в некоторых узлах сети нужно заменять этот сигнал более качественным, в котором фазовых помех нет, или, по крайней мере, их уровень незначителен. Где взять такой синхросигнал в узле сети? Его можно либо выделить из наименее искаженного входного сигнала, либо получить от автономного генератора (п. 6.1).

2. Система синхронизации сети по возможности должна использовать общий источник синхросигналов, чтобы уменьшить вероятность проскальзываний. При этом важно не допустить формирования замкнутых траекторий распространения синхросигналов между узлами сети в результате возникновения неисправностей линии или неправильных действий оператора (п. 6.2).

Системы синхронизации сетей с кольцевой структурой имеют некоторые особенности, рассмотренные в п. 6.3. Такие сети обладают повышенной надежностью, так как при отказе одной линии все узлы остаются доступными для обмена данными. Система синхронизации адаптируется к отказу линии, а после его устранения автоматически возвращается в исходное состояние.

В конце главы (п. 6.4) приведен метод зондирования сетей для выявления трасс распространения синхросигналов. В качестве зонда используется синхросигнал с искусственно введенным в него низкочастотным вандером. Этот сигнал регистрируется в узлах сети, соответствующих трассам его распространения. В результате анализа этих трасс можно проверить правильность построения системы синхронизации и, в частности, обнаружить недопустимые заикливания цепей передачи синхросигнала.

6.1. Обновление синхросигнала на пути его распространения по сети

Для более детального ознакомления с функционированием системы синхронизации сети, показанной на рис. 6.1, рассмотрим ее упрощенную модель (рис. 6.2) [2].

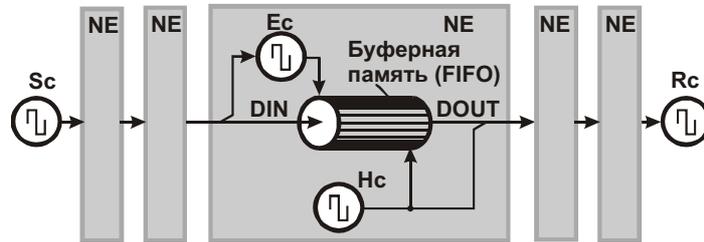


Рис. 6.2. Упрощенная модель сети, показанной на рис. 6.1

В этой модели трасса передачи данных от источника к приемнику проходит через последовательность удаленных друг от друга устройств NE (Network Equipment – оборудование сети). В каждом устройстве NE имеется буферная память типа FIFO.

Каждый узел выделяет из входного сигнала синхроимпульсы E_c и данные DIN . Синхроимпульсы E_c формируются с помощью местного генератора с фазовой автоподстройкой частоты. Данные DIN под управлением синхроимпульсов E_c заносятся в буферную память типа FIFO и последовательно продвигаются к ее выходу.

Данные $DOUT$ считываются из буферной памяти под управлением сигнала H_c . Номинальные частоты сигналов H_c и E_c совпадают, но сигнал H_c , как предполагаем, более близок к идеальному, чем сигнал E_c . Такое предположение оправдано тем, что линейный сигнал, из которого получен сигнал E_c , возможно, недостаточно стабилен, так как он претерпевает воздействия шумов в линии связи и иных факторов. (О вариантах происхождения сигнала H_c – чуть позже.)

Считанные данные $DOUT$ смешиваются с синхросигналом H_c в соответствии с принятым методом кодирования и поступают в линию для передачи в соседнее устройство NE, где процессы повторяются. Таким образом, достигается обновление синхросигнала по мере его распространения по сети.

Как отмечалось, частота синхронизации на разных участках сети может быть различной. Описанный принцип замены “второсортных” синхросигналов высококачественными остается применимым для каждого участка.

Чтобы предотвратить проскальзывания, на начальном этапе передачи данных их считывание из буферной памяти имеет смысл начинать только после того, как память заполнится до некоторого минимально допустимого уровня, иначе будет велик риск ее полного опустошения, например, из-за временного незначительного снижения частоты сигнала E_c . После начального заполнения буферной памяти текущий уровень регулируется так, чтобы он находился в допустимых пределах, например от 10 до 90 % ее полного объема (см. гл. 4).

В установившемся режиме цепочка из устройств NE работает как единое целое, напоминающее спортивную эстафету. Качество синхросигналов ухудшается на дистанциях между устройствами, в основном из-за помех в линиях связи, но в достаточной мере восстанавливается каждым последующим устройством. Если в каждом узле в необходимых случаях происходит описанное ранее обновление синхросигнала, то можно ожидать, что при передаче потока данных в каждом устройстве NE уровень заполнения буферной памяти колеблется в допустимых пределах, проскальзываний не возникает или, по крайней мере, они происходят достаточно редко.

В схеме на рис. 6.2 использовался источник H_c синхроимпульсов высокой точности и стабильности. Как получить такие синхроимпульсы? Есть два варианта: 1) установить

местный генератор; 2) “извлечь” синхроимпульсы из сигнала, принимаемого по одному из имеющихся каналов, выбрав сигнал с наилучшим качеством. Оба варианта представлены на рис. 6.3.

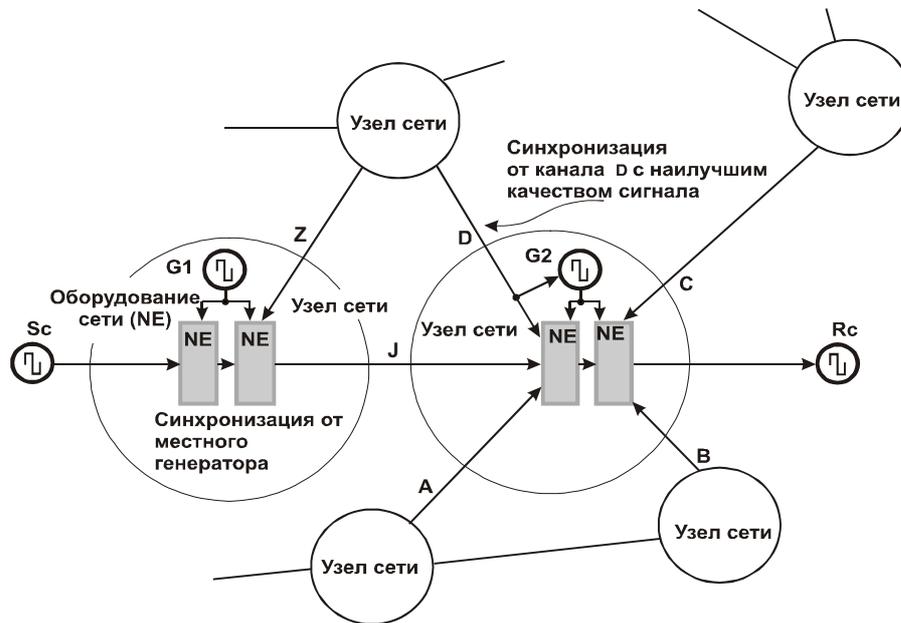


рис. 6.3. Варианты обновления синхросигнала при его распространении по сети

В данном примере на пути передачи исходного синхросигнала S_c (смешанного с данными) имеются четыре устройства NE. Первое и второе устройства размещены в первом здании и синхронизируются от местного генератора G_1 , имеющего высокий уровень точности и стабильности. Качество сигнала по каналу Z невысокое, так как этот канал, так же как и канал J , сильно зашумлен. Третье и четвертое устройства размещены во втором здании, удаленном от первого. Эти устройства работают и с другими каналами (A , B , C , D). В канале D , как предполагаем, присутствует сигнал относительно высокого уровня точности и стабильности. Он и выбирается в качестве опорного синхросигнала S_c (см. рис. 6.2) для генератора G_2 с фазовой автоподстройкой частоты.

Оба варианта обновления синхросигнала не лишены недостатков. Второй вариант подразумевает хорошее очищение восстановленного сигнала от паразитной частотной модуляции, проявляющейся в виде джиттера и вандера; но такое очищение достижимо далеко не всегда (см. гл. 5). Первый вариант может оказаться дорогим или громоздким, так как предполагает использование генератора высокой (по современным меркам) точности и стабильности. Далее кратко рассмотрена классификация таких генераторов и приведен пример построения самообучающегося генератора синхросигналов.

Генераторы сигналов высокой точности и стабильности

При построении сетей передачи данных используются генераторы трех уровней точности и стабильности: Stratum 1, Stratum 2, Stratum 2E, Stratum 3, Stratum 3E (Stratum – слой) [2].

Уровень Stratum 1 – самый высокий. Гарантированная точность – не ниже $\pm 1 \times 10^{-11}$. Этот уровень уже надежно преодолен. Так, источник синхронизации STSC2010 на основе водородного мазера обеспечивает максимальную стабильность частоты, достижимую современной технологией, порядка 10^{-14} (см. каталог фирмы Datum (США) на сайте [2]). Эта же фирма выпускает цезиевый источник синхросигна-

лов FTS4065B с кратковременной стабильностью 10^{-12} и долговременной стабильностью 10^{-14} .

Уровень Stratum 2 – более низкий, он гарантирует точность не ниже $\pm 1,6 \times 10^{-8}$. Генератор FTS1130 этого уровня выполнен на основе сверхстабильного кварцевого резонатора и обеспечивает стабильность уровня 5×10^{-11} в течение дня. Генераторы второго и третьего уровней выполняются также на основе рубидиевых источников синхросигналов.

Уровень Stratum 3 (Stratum 3E) гарантирует точность не ниже $\pm 4,6 \times 10^{-6}$.

В ряде решений применяется приемник спутниковых сигналов глобальной системы позиционирования GPS, что позволяет сравнительно просто сформировать сигнал уровня Stratum 1 и предоставляет возможность отслеживать абсолютное время. (Подробности – на сайте фирмы Datum [2].)

Самообучающийся генератор синхросигналов

В сотовых системах передачи данных по радиоканалам имеется ряд базовых станций, распределенных таким образом, что в поле зрения движущегося объекта (автомобиля, самолета, и т. п.) постоянно присутствует как минимум одна станция. Движущийся объект по мере перемещения обслуживается разными базовыми станциями. Для гарантии качества связи базовые станции должны вести учет абсолютного времени с точностью ± 3 мкс. Такой уровень точности достигается при использовании сигналов от спутниковой глобальной системы позиционирования GPS.

При потере сигнала синхронизации от глобальной системы позиционирования станция должна переключиться на сигнал от внутреннего (автономного) высокостабильного и точного генератора. По параметрам приемлемы рубидиевые генераторы, но их стоимость высока. Можно построить наземную станцию для передачи запасных сигналов синхронизации базовым станциям, но такое решение также дорого.

В [5] предлагается использовать в качестве резервного кварцевый генератор с системой обучения (рис. 6.4).

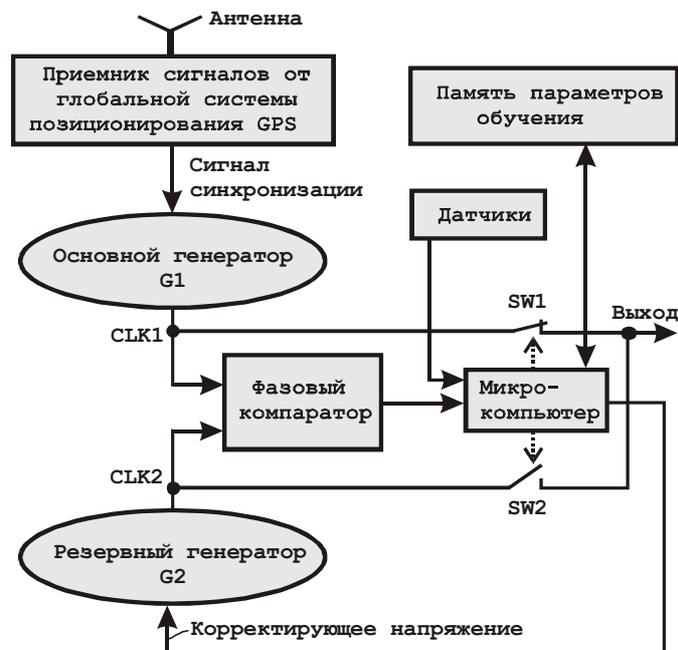


рис. 6.4. Резервированный генератор синхросигналов

В исходном состоянии приемник сигналов от глобальной системы позиционирования формирует импульсы синхронизации, которые воздействуют на основной генера-

тор G1. Электронный ключ SW1 замкнут, ключ SW2 разомкнут, на выход поступают сигналы CLK1 с выхода основного генератора.

Сигнал CLK2 с выхода резервного генератора сравнивается с сигналом CLK1. Если имеется несовпадение фаз, то микрокомпьютер стремится его скомпенсировать выдачей соответствующего корректирующего напряжения на управляющий вход резервного генератора G2 (кварцевый генератор допускает подстройку частоты в очень небольших пределах). Таким образом, сигналы CLK1 и CLK2 совпадают по частоте и имеют близкий нулю взаимный фазовый сдвиг.

Благодаря воздействию корректирующего напряжения компенсируются такие дестабилизирующие факторы, как старение кварцевого кристалла, колебания температуры, отклонение питающего напряжения от номинального и т. п.

При этом микропроцессорная система коррекции обучается вести себя так, чтобы в наилучшей степени приспособиться к изменяющимся внешним условиям в случае потери сигнала синхронизации от глобальной системы позиционирования или отказа основного канала.

Для этого в память параметров обучения записываются все существенные сведения, связанные с процессом управления генератором G2. Так, запоминаются уровни управляющих напряжений и соответствующие реакции со стороны генератора, вычисляются, проверяются и запоминаются оптимальные параметры управления и условия, при которых они достигнуты, обновляются многомерные таблицы, проводится интерполяция и т. д. Для регистрации условий проведения экспериментов микрокомпьютер опрашивает датчики температуры, влажности и проч. и сохраняет нужную информацию в памяти.

При потере сигнала синхронизации от глобальной системы позиционирования или в результате отказа основного канала (например из-за неисправности основного генератора) микропроцессор размыкает ключ SW1 и замыкает ключ SW2. В качестве выходного используется сигнал CLK2 от резервного генератора. Микропроцессор на основе накопленного опыта и получаемой с датчиков информации продолжает управлять резервным генератором так, как будто бы “ничего не произошло”.

После восстановления работоспособности основного канала устройство переходит в описанное ранее исходное состояние, а процесс обучения возобновляется.

Для повышения надежности устройства можно ввести второй, третий и т. д. резервные генераторы с индивидуальным обучением.

6.2. Предотвращение заикливания синхросигналов

Как уже отмечалось, система синхронизации сети по возможности должна использовать общий источник синхросигналов, чтобы уменьшить вероятность проскальзываний. При этом важно не допустить формирования замкнутых траекторий распространения синхросигналов между узлами сети в результате возникновения неисправностей линии или неправильных действий оператора. Прежде чем привести решения, исключаящие заикливание синхросигналов, рассмотрим типовую схему синхронизации участка транспортной сети при дуплексной передаче данных (рис. 6.5, а) [2].

В этой схеме начальный и конечный узлы участка транспортной сети синхронизируются от генераторов G наивысшего уровня точности и стабильности (STRATUM 1). Данные, смешанные с синхросигналами, передаются между устройствами NE (например мультиплексорами) одновременно в обе стороны, т. е. в дуплексном режиме.

Среда передачи данных между соседними устройствами NE – одна или две витые пары проводов, оптоволоконная линия и т. п. Скорости передачи данных в обоих направлениях (обозначенных стрелками) одинаковы. Приведенная на рис. 6.5, а схема достаточно распространена; она, например, может являться фрагментом замкнутого кольца из устройств NE в сети SONET/SDH.

Каждое промежуточное устройство NE (рис. 6.5, б) принимает линейные сигналы с обоих направлений, выделяет из них данные и синхроимпульсы. Для восстановления правильной синхронизации и фильтрации джиттера и вандера, как было показано ранее, используются блоки буферной памяти типа FIFO. Здесь и далее эти блоки не показаны для упрощения рисунков. Поэтому линии, проведенные на рисунках внутри устройств NE, нельзя рассматривать как прямые проводные связи между соответствующими точками. В данном случае штриховые линии отражают возможные пути передачи синхросигналов.

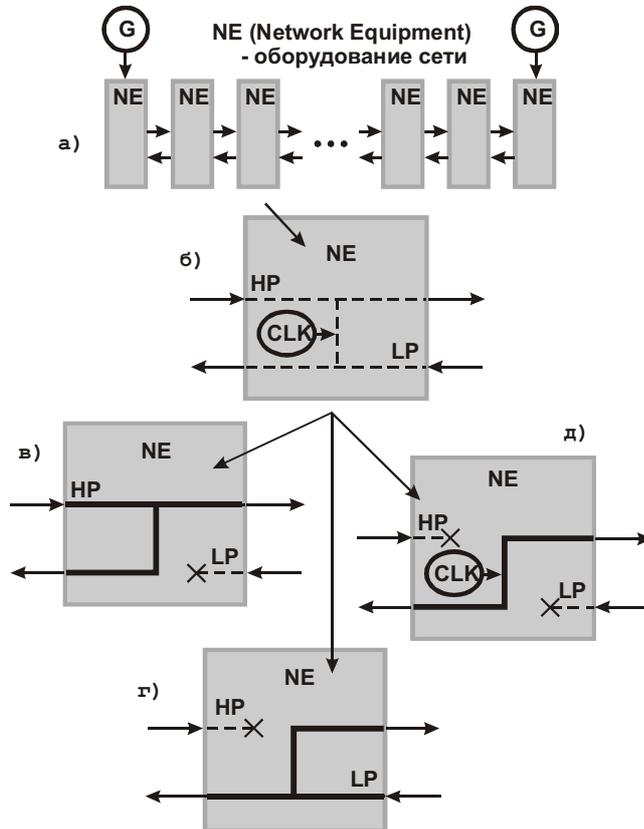


Рис. 6.5. Схема синхронизации участка транспортной сети при дуплексной передаче данных: а – общая структура цепи из устройств NE; б – структура одного из устройств NE; в, г, д – варианты настройки устройства NE

Промежуточное устройство NE содержит местный генератор CLK синхросигнала относительно невысокой точности и стабильности. (Было бы расточительным устанавливать генераторы уровня STRATUM 1 в каждом устройстве NE.) Этот генератор, как будет показано, используется кратковременно и только в критических ситуациях, в процессе оперативной автоматической замены источника синхронизации. Точность и стабильность генератора CLK, однако, должны быть достаточными для функционирования участка сети, пусть даже с повышенным уровнем ошибок. При нормальной работе этот генератор выключен, и для синхронизации выходных сигналов применяется синхросигнал, выделенный из входного, поступающего справа или слева, в соответствии с присвоенными им приоритетами.

Приоритеты отражают целесообразность использования выделенных из входного сигнала синхроимпульсов в качестве выходных. В данном примере (см. рис. 6.5, б) высокий приоритет HP присвоен левому, а низкий LP – правому входам устройства NE. Это означает, что при нормальной работе данные, выдаваемые устройством NE в обе стороны, синхронизируются сигналом, выделенным из левого (а не из правого) канала.

Промежуточное устройство NE можно настроить на работу в одном из трех режимов (рис. 6.5, в, з, д).

В первом режиме (рис. 6.5, в) для общей синхронизации используется сигнал с высокоприоритетного входа HP. Синхроимпульсы, выделенные из сигнала с низкоприоритетного входа LP, используются только для занесения соответствующих данных в буферную память и далее не распространяются, что условно отражено на рисунке “крестиком”. Утолщенными линиями показана трасса распространения основного синхросигнала.

Во втором и третьем режимах (рис. 6.5, з, д) для общей синхронизации используются соответственно сигналы с низкоприоритетного входа LP и от генератора CLK.

Зацикливание синхросигналов

При построении сетей разработчики стремятся к тому, чтобы каждый узел синхронизировался сигналом уровня STRATUM 1. Но, как уже отмечалось, экономически (и, может быть, не только) нецелесообразно устанавливать автономный генератор этого уровня в каждом сетевом устройстве (например, мультиплексоре) или даже в группе устройств. Проще получить высокоточный и высокостабильный сигнал из канала связи после устранения в нем джиттера и вандера (или, по крайней мере, после их значительного подавления). Но при этом необходимо абсолютно точное знание “родословной” всех сигналов – кандидатов на использование в качестве опорных. Иначе могут появиться топологические “привидения” – циклы, не содержащие материального источника опорной частоты. Процесс возникновения такого цикла поясняется схемой, приведенной на рис. 6.6 [2].

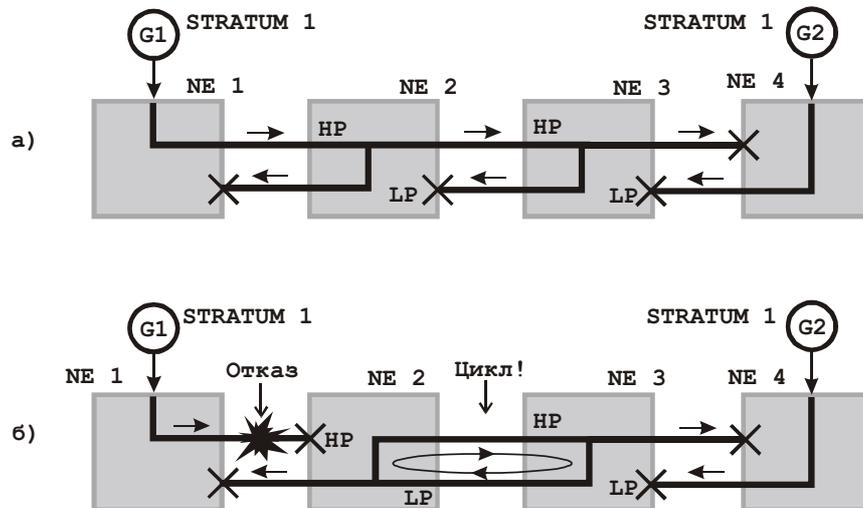


Рис. 6.6. Схема, поясняющая процесс возникновения ошибок синхронизации – циклов, лишенных опорного источника сигналов: а – исходное (работоспособное) состояние цепи из устройств NE; б – состояние цепи после отказа канала связи NE 1 → NE 2

В данном примере участок сети состоит из четырех устройств NE 1 – NE 4. Устройства NE 1 и NE 4 синхронизируются непосредственно от высоконадежных автономных генераторов G1 и G2 уровня STRATUM 1. Поэтому входные синхросигналы из каналов связи в этих устройствах используются только для ввода данных с соответствующего направления в буферную память.

В устройствах NE 2 и NE 3 (рис. 6.6, а) для синхронизации выходных сигналов использованы импульсы, выделенные с высокоприоритетных входов HP. Синхросиг-

налы с низкоприоритетных входов LP используются только для ввода данных в буферную память и прекращают дальнейшее распространение.

Таким образом, все устройства участка сети, показанного на рис. 6.6, а, синхронизируются сигналами уровня STRATUM 1: устройства NE 1, NE 2 и NE 3 – от генератора G1, а устройство NE 4 – от генератора G2. Как видим, всё сделано правильно.

Теперь предположим, что канал связи между устройствами NE 1 и NE 2 перестал работать в одном направлении, например в результате ухудшения изоляции в линейном кабеле (рис. 6.6, б). Этот факт прежде всего будет аппаратно зарегистрирован в устройстве NE 2 в результате полного отсутствия или поступления искаженного и ослабленного сигнала со стороны устройства NE 1. Устройство NE 3 не замечает случившегося и по-прежнему получает от устройства NE 2 сигнал синхронизации, правда, ухудшенного качества, так как генератор с автоподстройкой частоты устройства NE 2 “по инерции” продолжает работать на частоте, близкой к номинальной, но уже без коррекции со стороны генератора G1.

В условиях отсутствия сведений о происхождении сигналов на входах устройства NE 2 человек (администратор) или некий автоматический переключатель режимов синхронизации может принять простое, но непродуманное решение: в ответ на пропадание высокоприоритетного сигнала с входа HP переключиться на низкоприоритетный сигнал с входа LP. Это и делается в данном примере; но после такого переключения, как видим из схемы, образуется цикл, не содержащий связи с опорными генераторами G1 или G2.

Этот цикл формируется двумя генераторами с автоподстройкой частоты, размещенными в узлах NE2 и NE3; каждый генератор стремится подстроиться по частоте к другому, рассматривая его в качестве опорного. Поэтому создается положительная обратная связь, которая приводит к выходу обоих генераторов на граничные допустимые отклонения от центральной частоты или к колебаниям частоты в некоторых малых пределах.

Тем не менее, участок сети может оставаться в работоспособном состоянии (если не считать отказа канала NE 1 → NE 2), так как работа пары “безопорных” источников синхросигналов может быть удовлетворительной в том смысле, что данные, хотя и с повышенной вероятностью ошибки, могут передаваться по сети.

В этом кроется трудность обнаружения таких ситуаций. Поэтому лучшее средство борьбы с ними – профилактическое обследование и анализ правильности трасс распространения синхросигналов. Кроме того, нужно исключить возможность принятия ошибочных решений, подобных описанному.

Далее рассмотрены метод и схема, исключаяющие формирование циклических трасс распространения синхросигналов при отказе канала связи.

Автоматическое предотвращение заикливания синхросигналов

Мы только что рассмотрели сравнительно простую ситуацию, когда неправильное решение, приведшее к заикливанию синхросигнала (см. рис. 6.6, б), было принято из-за недостатка информации о происхождении входных синхросигналов. Но чем сложнее сеть, тем труднее получить о ней полную информацию. Выручает то, что разработаны протоколы передачи сведений о статусе синхросигналов и алгоритмы выбора оптимальных трасс распространения синхросигналов от опорных источников. Чтобы пояснить смысл автоматической прокладки новой трассы после изменившихся условий работы сети, рассмотрим простой пример (рис. 6.7) [2].

Исходные условия работы участка сети совпадают с приведенными в предыдущем примере (сравните рис. 6.6, а и рис. 6.7, а), поэтому повторять описание не будем. И неисправность будет точно такой же, но реакция на нее на этот раз – правильная.

Сравнивая рисунки, видим, что появились новые обозначения: ST 1, DUS и др. Это мнемокоды, отражающие статус передаваемого синхросигнала. Но – всё по порядку.

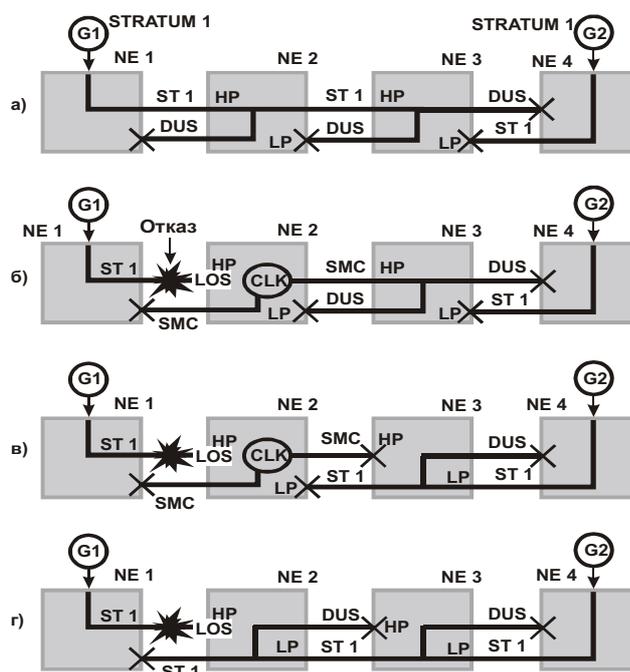


рис. 6.7. Схема, поясняющая процесс адаптации цепи из устройств NE к отказу канала связи NE 1 → NE 2: а – исходное (работоспособное) состояние цепи; б, в – состояния в процессе адаптации; г – конечное состояние

Данные между устройствами NE 1 – NE 4 передаются кадрами. Кадр содержит как полезную (с точки зрения пользователя), так и служебную информацию, о существовании которой пользователь не знает. В частности, в кадре имеется четырехбитное поле статуса синхросигнала. В нем можно закодировать до 2^4 различных признаков, характеризующих синхросигнал, переносящий данный кадр от одного устройства NE_i к другому, соседнему. На разных перегонах от устройства к устройству статус синхросигнала может изменяться; но он может изменяться также и в зависимости от ситуации, в чем мы далее убедимся.

В примере использованы следующие признаки, характеризующие синхросигнал:

ST 1 – признак, означающий, что синхросигнал произошел от генератора уровня STRATUM 1;

DUS – (Do not Use for Sync) – “Не использовать для синхронизации”;

LOS – (Loss – потеря) – признак, формируемый устройством, потерявшим соответствующий входной сигнал;

SMC – (SONET Minimum Clock) – признак, означающий, что синхросигнал имеет относительно низкое качество, т. е. обладает минимально допустимыми параметрами (точностью, стабильностью и др.), приемлемыми в стандартах сетей SONET.

Начальное распределение признаков (см. рис. 6.7, а) не допускает проведения в жизнь ошибочного решения, принятого в предыдущем примере (см. рис. 6.6). Действительно, устройство NE 2 теперь не имеет права рассматривать синхросигнал с низкоприоритетного входа LP в качестве эталонного – об этом предупреждает периодически поступающий на этот вход признак DUS.

При возникновении отказа (см. рис. 6.7, б) разворачивается такая последовательность событий.

Обнаружив отказ, устройство NE 2 формирует “для себя” признак LOS и переключается на синхронизацию от внутреннего генератора CLK. Чтобы оповестить со-

седние узлы о случившемся, устройство NE 2 теперь сопровождает все исходящие кадры признаком SMC. Устройство NE 1 никак не реагирует на изменение статуса синхросигнала из линии, так как оно синхронизируется непосредственно от генератора G1 уровня STRATUM 1, и вполне этим удовлетворено.

Но устройство NE 3 “не желает мириться” со снижением качества эталонного синхросигнала. Оно проверяет возможность получения эталонного сигнала с другого, низкоприоритетного входа LP. Такая возможность существует, так как сигнал на низкоприоритетном входе LP устройства NE 3 имеет статус ST 1 (STRATUM 1). Поэтому устройство NE 3 переключается на новый источник синхронизации и назначает исходящему влево синхросигналу статус ST 1, поощряя его распространение в глубь структуры (см. рис. 6.7, в).

Устройство NE 2 также не желает мириться с низким качеством своего источника синхронизации (генератора CLK) и ждет появления синхросигнала лучшего качества на единственном исправном входе LP. Ожидание оправдывается с поступлением на этот вход кадров, содержащих признак ST 1. Теперь устройство NE 2 переключается на новый источник синхросигнала и назначает статус выходных сигналов в соответствии с рис. 6.7, г. Задача решена: синхронизация восстановлена, каждое устройство получает синхросигналы уровня STRATUM 1.

Среди прочих четырехразрядных статусных признаков, характеризующих синхросигнал, следует особо выделить признак STU (STRATUM Traceability Unknown – “STRATUM-трассируемость неизвестна”). Этот признак (его код 0000_2 , что существенно) несет информацию о том, что происхождение синхросигнала, сопровождающего данный кадр, неизвестно.

В действительности, всегда можно найти первоисточник сигнала. Но дело в том, что протокол обмена статусными признаками разработан сравнительно недавно, и не все сетевые устройства его понимают. Иными словами, ретранслируя кадры, эти устройства вообще неспособны заполнить поле статусных признаков каким-либо кодом. Поэтому поле остается пустым (код 0000_2), а нулевой код во избежание недоразумений воспринимается соседями по сети как признак STU.

Таким образом, новые устройства могут безболезненно внедряться в старые сети передачи данных. По мере увеличения числа новых устройств в сети образуются островки, а затем целые архипелаги и континенты из устройств, автоматически и коллективно решающих проблемы общей синхронизации.

6.3. Синхронизация кольцевых структур

Мы уже не раз говорили о том, что при построении систем синхронизации недопустимо закливание трасс распространения синхросигналов. Но как поступить, если сама сеть передачи данных имеет кольцевую структуру?

Никакого противоречия нет – идеи построения систем синхронизации по сути те же, что и только что рассмотренные при описании схемы, показанной на рис. 6.7. Далее представлены три варианта схем с кольцевой структурой. Эти схемы различаются деталями, но имеют общий признак – они обладают устойчивостью к отказам аппаратуры, в частности, к одиночным отказам каналов связи.

Первый вариант

Рассмотрим схему, приведенную на рис. 6.8 [2]. В этой схеме приняты введенные ранее условные графические обозначения и термины. Для определенности уточним две детали.

1. Генератор G формирует синхросигнал уровня STRATUM 1 при наличии внешней синхронизации от спутниковой системы глобального позиционирования GPS.

Если поступление внешнего синхросигнала прекращается (например при появлении неисправности приемной антенны генератора), то генератор автоматически переключается в автономный режим и формирует синхросигнал уровня STRATUM 2.

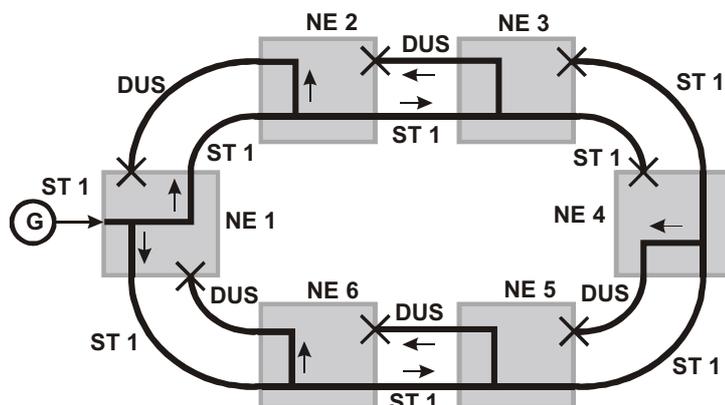


Рис. 6.8. Пример схемы распространения синхросигналов в кольцевой структуре (первый вариант)

2. Каждое устройство NE 1 – NE 6 имеет внутренний генератор синхросигнала уровня STRATUM 3.

При нормальной работе структуры каждое устройство NE_i пользуется синхросигналом уровня STRATUM 1. Устройство NE 1 получает этот сигнал непосредственно от генератора G, а остальные устройства – косвенно, выделяя его из передаваемой по линии смеси данных и синхроимпульсов. Как следует из схемы, паразитных контуров распространения синхросигналов нет, их возникновение профилактически предотвращается приданием статуса DUS (“Не использовать для синхронизации”) синхросигналам, возвращающимся в сторону генератора G.

Рассмотрим поведение системы синхронизации в трех нештатных ситуациях (A, B, C).

Ситуация А. Предположим, что в генераторе G отказал приемник сигналов от спутниковой глобальной системы позиционирования GPS. Генератор продолжает работать в автономном режиме и начинает формировать синхросигнал уровня STRATUM 2, о чем сообщает устройству NE 1. Это устройство, в свою очередь, уведомляет ближайших соседей о снижении качества синхросигнала заменой признака ST 1 признаком ST 2 в передаваемых кадрах (эти изменения на рисунке не отражены).

Информация о снижении качества синхросигнала распространяется к остальным устройствам. Таким образом, все устройства NE_i принимают ее к сведению, но за неимением лучшего синхросигнала работают в тех же режимах, что и ранее. Кольцевая структура остается работоспособной.

Ситуация В. Предположим, что отказал генератор G. Обнаружив отсутствие сигналов от генератора G, устройство NE 1 начинает использовать собственный внутренний генератор синхросигналов уровня STRATUM 3. Так же как и в предыдущей ситуации, все устройства NE_i принимают к сведению изменение статуса получаемого из линии синхросигнала (ST 1 → ST 3).

Имеет ли смысл пользоваться выделенным из линии синхросигналом уровня STRATUM 3 (в качестве опорного), если есть сигнал от собственного генератора того же уровня точности и стабильности? Ответ на этот вопрос зависит от условий работы сети. С одной стороны, желательно синхронизировать все устройства сети от общего генератора – тогда, вроде бы, автоматически решается задача точного согласования скоростей передачи данных между любыми устройствами. Но, с другой стороны, качество линии может быть настолько низким, что прибывающий на вход устройства син-

хросигнал может содержать значительные фазовые помехи – джиттер и вандер. Как уже отмечалось, полностью очистить синхросигнал от низкочастотных фазовых помех практически невозможно. Поэтому в данной ситуации целесообразно использовать в каждом устройстве сети автономный генератор синхросигналов. Тогда придется смириться с тем, что скорости передачи данных между устройствами сети будут незначительно различаться, а это может привести к проскальзываниям, если не принять мер по предотвращению переполнения буферной памяти устройств. В данном примере уровень точности и стабильности внутренних генераторов устройств весьма высок, что облегчает задачу предотвращения проскальзываний. Поэтому предполагаем, что все устройства NE_i переходят к работе от внутренних генераторов. Структура и в этой ситуации остается работоспособной.

Ситуация С. Предположим, что отказало устройство $NE 1$. Связи с соседними узлами (устройствами $NE 2$ и $NE 6$) полностью разорваны. Этим узлам не остается ничего другого, как перейти к работе от внутренних генераторов, поскольку альтернативные источники синхросигналов имеют статус DUS (“Не использовать для синхронизации”). Устройства $NE 3 – NE 5$ также начинают синхронизироваться от внутренних генераторов, так как выделяемые из линии синхросигналы ничуть не лучше своих (а скорее – хуже, как уже отмечалось). Так что оставшаяся исправной часть структуры и в этой ситуации остается работоспособной.

Второй вариант

В сети, показанной на рис. 6.9, *а*, [2] узел 1 содержит генератор $G1$ синхросигналов уровня STRATUM 1. Узлы 2 – 6 содержат генераторы значительно более низких уровней точности и стабильности. При нормальной работе сети эти генераторы отключены. Чтобы обеспечить высокий уровень качества синхронизации сети, узел 1 назначен ведущим, а узлы 2 – 6 – ведомыми. Последние используют для синхронизации сигнал, выделенный из линии и порождаемый генератором $G1$.

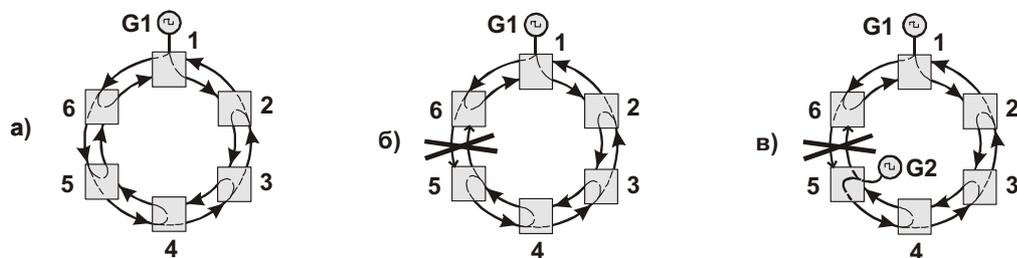


Рис. 6.9. Распространение синхросигналов в сети с кольцевой топологией (второй вариант): *а* – в отсутствие неисправностей; *б* – при неправильной реакции на неисправность; *в* – при использовании двух генераторов синхросигналов ($G1$ и $G2$) для компенсации неисправности

Предположим, что в сети произошел обрыв кабеля между узлами 5 и 6. Узел 5 обнаруживает потерю источника синхросигнала и оценивает ситуацию. Имеются два альтернативных источника: сигнал от узла 4 и от внутреннего генератора.

Если попытаться использовать в качестве опорного сигнал от узла 4, то это приведет к заикливанию синхросигнала между узлами 4 и 5, что недопустимо (рис. 6.9, *б*). Нестабильный сигнал, полученный в результате такого заикливания, будет распространяться также в узлы 3 и 2 в качестве опорного. Это приведет к резкому повышению интенсивности ошибок при передаче данных между узлами 1 – 5, и, вероятнее всего, к полной потере работоспособности фрагмента сети.

Если воспользоваться внутренним генератором $G2$ узла 5 (рис. 6.9, *в*), то получим работоспособную сеть. Но качество синхронизации уступает первоначальному, так как

наряду с генератором G1 уровня STRATUM 1 применен генератор G2 со значительно меньшим уровнем точности и стабильности. В этой ситуации генератор G1 обслуживает узлы 1 и 6, а генератор G2 – узлы 5 – 2.

Существует более приемлемое решение. Действительно, даже с учетом неисправности кольцевая структура сети позволяет восстановить синхронизацию от единственного источника – генератора G1, в чем мы сейчас убедимся.

Рассмотрим алгоритм перехода к такому решению (рис. 6.10) [57].

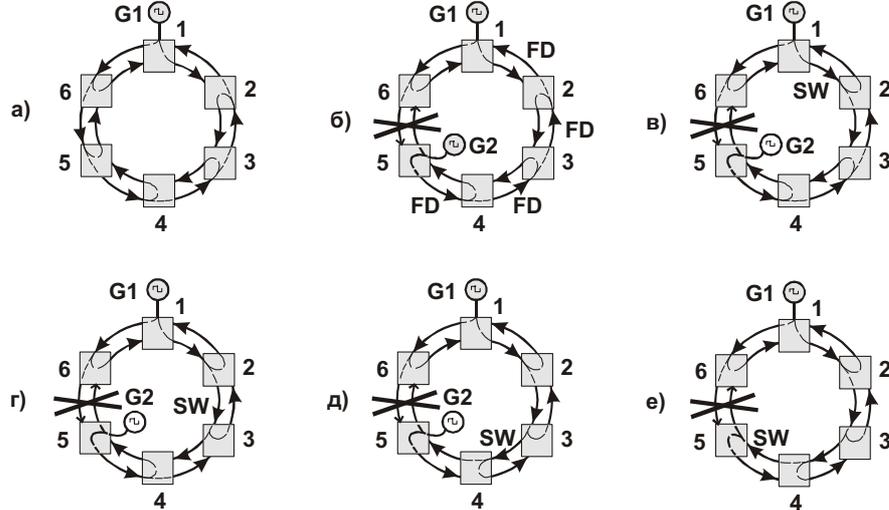


Рис. 6.10. Распространение синхросигналов в сети с кольцевой топологией (второй вариант): *а* – в отсутствие неисправностей; *б* – *д* – в процессе адаптации к неисправности с временным использованием двух генераторов синхросигналов (G1 и G2); *е* – после адаптации

В исходном состоянии неисправностей нет, сеть синхронизируется от генератора G1 (рис. 6.10, *а*). При обрыве кабеля между узлами 5 и 6 (рис. 6.10, *б*) в работу включается генератор G2, как уже было описано. Затем узел 5 посылает по кольцу в сторону от неисправности код FD (Fault Detection – обнаружение ошибки). Этот код распространяется через узлы 4, 3 и 2 и поглощается ведущим узлом 1.

Узел 1 принимает к сведению факт обрыва связи (местоположение обрыва его не интересует) и посылает в ответ по соответствующему направлению код SW (Switch – переключение), как показано на рис. 6.10, *в*. Этот код последовательно “прокладывает путь” для сигнала от генератора G1. Если часть пути уже имеется и поэтому прокладывать путь не надо, то соответствующие узлы структуры остаются в прежнем состоянии и транслируют код дальше по цепи. Но в нашем примере готового участка пути нет, и его создание начинается на первом же шаге передачи кода SW. Получив этот код, узел 2 переключается на синхронизацию от сигнала, выделяемого из линии 1 – 2, и посылает код SW в узел 3 (рис. 6.10, *г*). Узел 3 перестраивается на синхронизацию с направления 2 – 3 и т. д. После адаптации узла 5 код SW далее не распространяется и система переходит в новое устойчивое состояние, при котором все узлы сети синхронизируются от генератора G1, что и требовалось.

После устранения неисправности система синхронизации автоматически поэтапно возвращается в первоначальное состояние (рис. 6.11).

Как было показано, в результате обнаружения неисправности и обмена кодами FD и SW сеть пребывает в состоянии, приведенном на рис. 6.11, *а*. После устранения неисправности узел 5 получает сигнал по цепи 6 – 5 и переходит к синхронизации от этого сигнала. Затем узел 5 посылает в узел 4 код RR (Return Request – запрос возврата), как показано на рис. 6.11, *б*. Получив код RR, узел 4 выбирает в качестве источника синхронизации противоположный линейный вход и посылает код RR в узел 3 (рис.

6.11, в). Этот процесс продолжается, в конечном счете система синхронизации возвращается в исходное состояние, соответствующее отсутствию неисправностей.

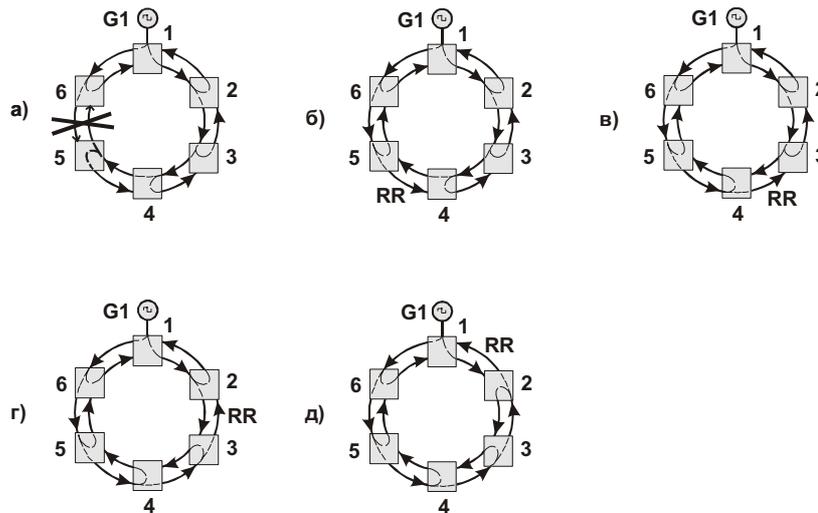


Рис. 6.11. Распространение синхросигналов в сети с кольцевой топологией (второй вариант): *а* – в результате предшествующей адаптации к неисправности; *б* – *в* – в процессе возврата к исходному состоянию после устранения неисправности; *г* – после возврата в исходное состояние

Третий вариант

Напомним, что во втором варианте построения системы синхронизации информация о неисправности распространяется по сегменту кольцевой структуры и достигает ведущего узла. Этот узел анализирует ситуацию и принимает решение о коррекции траектории распространения основного синхросигнала, затем посылает в нужном направлении соответствующую команду и т. д. Как видим, во втором варианте особая роль при принятии решения отводится ведущему узлу сети, т. е. наблюдается некоторая централизация управления, что нежелательно, так как увеличивается время реакции на ошибку и уменьшается надежность системы. Рассмотренный далее третий вариант [49] близок ко второму, но в нем любой узел кольцевой структуры при определенных условиях способен изменить траекторию распространения основного синхросигнала (рис. 6.12).

Как и в предыдущем примере, сеть состоит из шести узлов 1 – 6 и синхронизируется от генератора G1 сигналов высокой точности и стабильности. Узел 1 – ведущий, узлы 2 – 6 – ведомые. Каждый узел содержит внутренний генератор синхросигналов сравнительно низкого качества, но эти генераторы при нормальной работе системы выключены и применяются лишь при устранении нештатных ситуаций. В кольцевой структуре циркулируют два потока информации: один распространяется по часовой стрелке, другой – против. Данные упакованы в кадры. Кадр помимо данных содержит служебную информацию, в частности, характеристику синхросигнала, несущего данный кадр. Эта характеристика представлена двухбитовым кодом ML (см. коды, показанные вблизи соответствующих стрелок на рисунке). Бит M содержит информацию о качестве сигнала: если $M = 0$, то сигнал произошел от генератора G1; если $M = 1$, то от внутреннего генератора одного из узлов сети. Бит L характеризует топологическое происхождение сигнала: при $L = 0$ сигнал распространяется в сторону от породившего его источника (генератора), при $L = 1$ – в обратном направлении, т. е. к своему источнику.

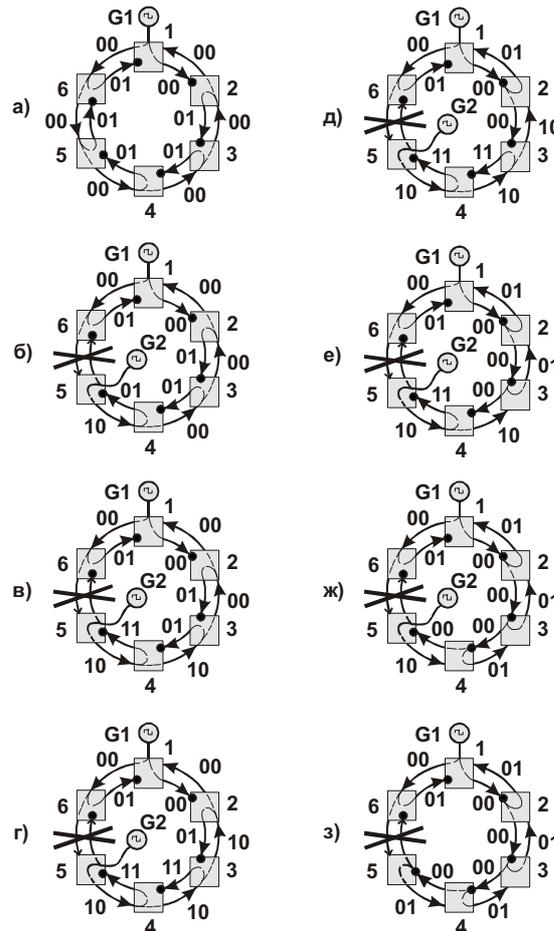


рис. 6.12. Распространение синхросигналов в сети с кольцевой топологией (третий вариант): *а* – в отсутствие неисправностей; *б* – *ж* – в процессе адаптации к неисправности; *з* – после адаптации к неисправности

В исходном состоянии (рис. 6.12, *а*) неисправностей нет, сеть синхронизируется от генератора $G1$. Из рисунка следует, что синхросигнал от генератора $G1$ сопровождает передачу данных в сети как против, так и по часовой стрелке (по внешнему и внутреннему контурам). Однако топология трасс распространения синхросигнала в обоих направлениях различна. Это связано с тем, что имеется система приоритетов при выборе источника синхросигнала узлом сети. Каждый узел может синхронизироваться от одного из трех источников синхросигнала: от внутреннего генератора (в нештатных ситуациях), от сигнала с низкоприоритетного входа (эти входы помечены на рисунке точками) или от сигнала с высокоприоритетного входа (соответствующего внешнему контуру передачи данных).

В данном случае каждый ведомый узел пользуется синхросигналом с высокоприоритетного входа, поэтому синхросигнал внутреннего контура передается по разомкнутым петлям обратной связи (исключение представляет пара узлов 1 – 2). Коды ML на всех дистанциях внешнего контура одинаковы и равны 00 . Это означает, что, во-первых, родословная любого синхросигнала этого контура ведется от генератора $G1$, и, во-вторых, направление передачи прямое. Почти на всех дистанциях внутреннего контура $ML = 01$. Это подтверждает хорошую родословную сигнала, но предостерегает о том, что его “родителем” является сам узел – получатель.

После обнаружения неисправности (обрыва кабеля между узлами 5 и 6) узел 5 переходит в режим синхронизации от внутреннего генератора $G2$ (рис. 6.12, *б*). Чтобы сообщить о смене режима синхронизации соседу, узел 5 сопровождает передаваемые по внешнему контуру кадры признаком $ML = 10$.

Реакция узла 4 на получение кода $ML = 10$ отражена на рис. 6.12, *в*. Этот узел вынужден смириться со снижением качества получаемого синхросигнала. Действительно, сигнал от своего внутреннего генератора по качеству такой же, но его использование привело бы к проскальзываниям, так как частота своего генератора может заметно отличаться от частоты внутреннего генератора узла 5 (например, на 100 Гц или более). Сигнал на низкоприоритетном входе узла 4 не может использоваться в качестве опорного, так как он помечен признаком $L = 1$, т. е. получен по петле обратной связи, исходящей из своего же узла. В результате видимые изменения состояния сети выражаются в том, что узел 4 выдает во внешний контур код $ML = 10$, а во внутренний – код $ML = 11$.

Реакция узла 3 на получение кода $ML = 10$ отражена на рис. 6.12, *г* и полностью повторяет реакцию узла 4 на этот же код. Узел 3 выдает во внешний контур код $ML = 10$, а во внутренний – код $ML = 11$.

Узел 2 реагирует на получение кода $ML = 10$ по-иному (рис. 6.12, *д*). Этот узел имеет возможность переключиться на высококачественный синхросигнал, поступающий на его низкоприоритетный вход и помеченный кодом $ML = 00$. Такое переключение выполняется и сопровождается выдачей кодов 00 и 01 во внутренний и внешний контуры передачи данных. Таким образом, ключевое событие, связанное с началом изменения траектории распространения основного синхросигнала, совершилось без участия ведущего узла (1) сети.

На следующих трех шагах (рис. 6.12, *е, ж, з*) узлы 3, 4 и 5, так же как и узел 2, переключаются на синхронизацию от высококачественного сигнала, поступающего с низкоприоритетного входа. Таким образом, в результате адаптации к ошибке все узлы сети синхронизируются от генератора $G1$.

После устранения неисправности сеть возвращается в первоначальное состояние (рис. 6.13). Во всех рассмотренных ситуациях длительность перехода каждого узла сети из одного состояния в другое может составлять 200 мс.

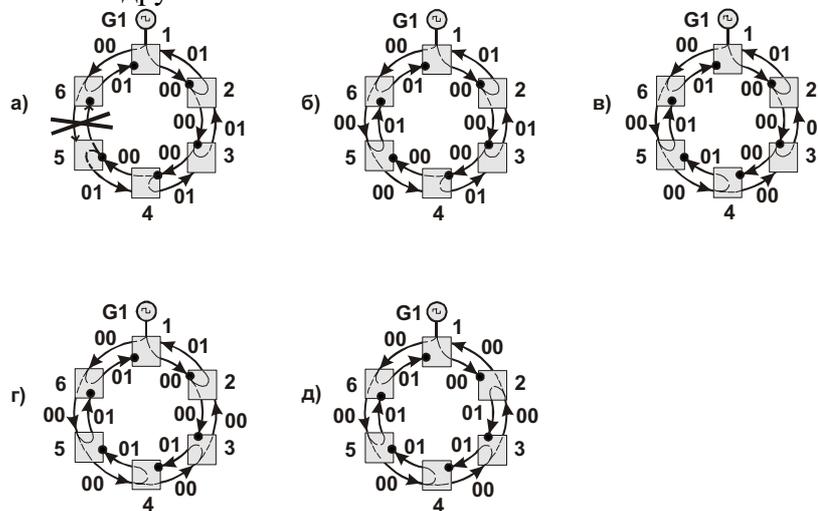


рис. 6.13. Распространение синхросигналов в сети с кольцевой топологией (третий вариант): *а* – в результате предшествующей адаптации к неисправности; *б* – *г* – в процессе возврата к исходному состоянию после устранения неисправности; *д* – после возврата в исходное состояние

6.4. Использование искусственного вандера для отслеживания трасс синхросигналов в сетях со сложной топологией

Как уже отмечалось, вандер практически “неистребим”, если его частота достаточно низка и составляет, например, 0,001 Гц. Это означает, что вандер проникает сквозь сколь угодно длинные цепи из узлов компьютерной сети вместе с несущим его синхросигналом. В [53] предлагается использовать практически неограниченную проникающую способность вандера для зондирования компьютерных сетей со сложной топологией с целью выявления трасс распространения синхросигналов. Воплощение этой идеи поясняется схемами распространения синхросигналов в компьютерной сети (рис. 6.14).

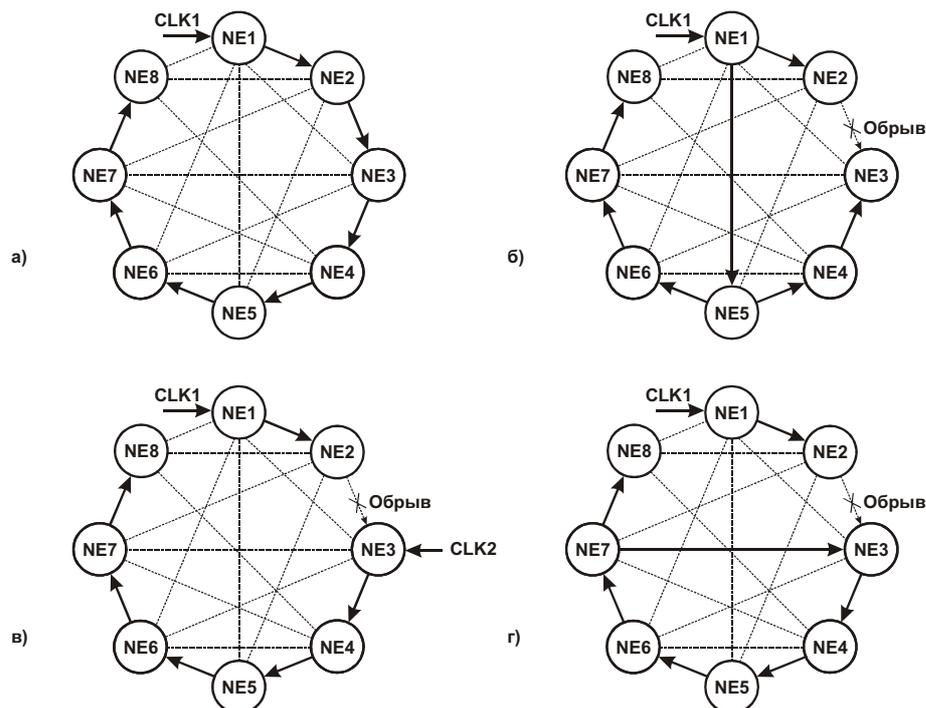


Рис. 6.14. Схемы распространения синхросигнала в сети: а – в – допустимые варианты; г – ошибочный вариант

Сеть содержит узлы NE1 – NE8 (NE – Network Element – элемент сети). Каждый узел соединен с другими узлами, как показано сплошными и штриховыми линиями. Стрелками показаны направления распространения синхросигналов между узлами.

Предположим, что в исходном состоянии (рис. 6.14, а) сеть синхронизируется сигналом CLK1, поступающим в узел NE1. Этот сигнал распространяется по цепи из узлов NE1 – NE8 в смеси с данными по соответствующим направлениям. Узлы NE2 – NE8 выделяют из полученной смеси данные и синхросигнал, который используют для внутренних нужд и для синхронизации исходящих данных во всех направлениях. Данная схема распространения синхросигнала выбрана в качестве примера и, конечно, может быть построена по-другому.

Как показано на рис. 6.14, б, при обрыве связи между узлами NE2 и NE3 синхросигнал CLK1 доставляется в узлы NE3, NE4, NE6 – NE8 через узлы NE1 и NE5. Узел NE2 получает синхросигнал из узла NE1. Это одно из возможных правильных решений задачи восстановления синхронизации сети после обнаружения неисправности. Сейчас нас не интересует, почему сеть “выбрала” именно это решение. Еще одно из возможных правильных решений приведено на рис. 6.14, в. Здесь сигнал CLK1 синхронизирует только узлы NE1 и NE2. Узел NE3, обнаружив потерю основного

синхросигнала от узла NE2, переключился на резервный внешний синхросигнал CLK2.

В схеме, показанной на рис. 6.14, *з*, в результате ошибочной реакции на обрыв связи между узлами NE2 и NE3 правильная синхронизация сохранилась только в узлах NE1 и NE2. В остальных узлах опорный источник синхросигнала утрачен, при этом образовался замкнутый путь распространения “безопорного” синхросигнала (что недопустимо): NE3 – NE7, NE3 ... и т. д.

Задача состоит в выявлении трасс распространения синхросигналов в сети в условиях отсутствия информации о неисправностях. Для ее решения в каждый узел вводится аппаратура модуляции и демодуляции синхросигналов. Проводится ряд экспериментов, в каждом из которых модулятор включен только в одном из узлов, а демодулятор – в одном или одновременно в нескольких или во всех узлах сети. Модулятор вносит незначительные (допустимые) фазовые искажения в синхросигнал, выдаваемый из узла сети по всем направлениям. Точнее, при модуляции синхросигнала его частота качается в допустимом для сети диапазоне, например в пределах $\pm 7,5 \cdot 10^{-6}$ от номинальной частоты. Период качания может выбираться, например, из диапазона 1 ... 1000 с. Таким образом, в синхросигнал искусственно вводится вандер. Частота вандера (в нашем примере 1 ... 0,001 Гц) должна быть достаточно низкой для его беспрепятственного прохождения через узлы сети.

Каждый узел сети в зависимости от сложившейся ситуации использует для синхронизации один из альтернативных источников. Синхросигнал от этого источника демодулируется. Демодулятор выделяет из синхросигнала вандер и сравнивает его частоту и амплитуду с принятыми при модуляции параметрами. В качестве простейшего демодулятора может использоваться частотомер или запоминающий осциллограф, а оценку параметров вандера может проводить человек – оператор.

Вернемся к схеме, показанной на рис. 6.14, *а*. Чтобы убедиться в том, что трасса распространения синхросигнала CLK1 образована незамкнутым кольцом из узлов NE1 – NE8, необходимо провести серию экспериментов, например такую. Во всех узлах сети включаются демодуляторы. В первом эксперименте модулятор включается в узле NE8. Если схема соответствует приведенной на рисунке, то демодулированный сигнал не должен обнаруживаться ни в одном из узлов, включая узел NE8 (напомним, что модулируются выходные, а демодулируется входной сигнал, выбранный в качестве опорного синхросигнала узла). Во втором эксперименте модулятор включается в узле NE7. Демодулированный сигнал с заданными параметрами должен обнаруживаться только в узле NE8. В третьем эксперименте модулятор включается в узле NE6. Правильный демодулированный сигнал должен обнаруживаться только в узлах NE7 и NE8. В четвертом эксперименте модулятор включается в узле NE5 и т. д. Серия экспериментов завершается включением модулятора в узле NE1 и проверкой наличия правильных демодулированных сигналов в узлах NE2 – NE8.

Если в результате обрыва связи между узлами NE2 и NE3 схема распространения синхросигнала стала соответствовать приведенной на рис. 6.14, *б*, то результаты проведения описанных ранее экспериментов, начиная с четвертого, окажутся иными. Действительно, при включении модулятора в узле NE5 правильные демодулированные сигналы будут дополнительно обнаружены в узлах NE4 и NE3. Общая картина распространения синхросигналов окончательно проясняется после завершения последовательного перемещения модулятора по остальным узлам сети с проверкой ответных реакций узлов.

Аналогичная серия экспериментов позволяет выявить трассы распространения синхросигналов, показанные на рис. 6.14, *в*. После перемещения модулятора из узла NE3 в узел NE2 обнаруживается отсутствие правильных демодулированных сигналов во всех узлах. Это свидетельствует о том, что ни один из узлов не использует выходной

сигнал узла NE2 в качестве опорного. Трасса NE1 – NE2 прослеживается после перемещения модулятора в узел NE1.

Проведение аналогичной серии экспериментов в условиях, когда сеть перешла в состояние, соответствующее рис. 6.14, з, обнаруживает ошибочную ситуацию. В сети имеется “предоставленная самой себе” замкнутая петля из генераторов, каждый из которых подстраивается по частоте к соседнему. Это приводит к установлению в петле некоторой неточной и нестабильной частоты и резкому повышению вероятности появления ошибок передачи данных. Включение модулятора в узле NE7 при проведении второго эксперимента создает дополнительное (запланированное) изменение частоты синхросигнала в замкнутой петле. При этом демодулятор узла NE7 обнаруживает вернувшийся по петле искусственно введенный в выходные сигналы вандер, что свидетельствует о заикливании синхросигнала между узлами сети.

Для лучшего распознавания искусственного вандера на фоне естественных фазовых шумов целесообразно вместо аналоговой частотной модуляции применять цифровую, с выраженными скачками фазы синхросигнала. Фактически это соответствует построению широкополосного служебного канала связи с использованием исходящих из узла синхросигналов для переноса данных со сверхнизкой скоростью.

Подводя итоги сказанному в гл. 6, отметим следующее.

1. По мере распространения сигнала по каналу связи в нем накапливаются фазовые помехи. При высоком уровне помех выделенный приемником синхросигнал не обеспечивает надежного приема данных. Для поддержания высокого качества передачи данных по сети необходима их ретрансляция, сопровождаемая обновлением синхросигнала. В качестве источника обновленного синхросигнала может использоваться автономный генератор либо генератор с фазовой автоподстройкой частоты, синхронизированный от сигнала, поступающего от одного из соседних узлов сети.

2. При построении системы синхронизации сети следует исключить возможность заикливания синхросигналов. В противном случае участок сети, на котором произошло заикливание, окажется неработоспособным или, в лучшем случае, будет работать с недопустимо высоким уровнем ошибок. Для автоматического исключения заикливания синхросигналов применяют методы, основанные на присвоении синхросигналам определенных статусных кодов и выполнении ряда правил работы с такими сигналами.

3. Сети с кольцевой топологией обладают высокой устойчивостью к отказам. Синхронизацию такой сети предпочтительно осуществлять от общего генератора. Приведенные примеры построения систем синхронизации показывают, что при выполнении определенных правил работы с синхросигналами можно гарантировать правильную адаптацию сети к отказам и автоматический возврат в исходное состояние после устранения неисправностей.

7. Синхронизация на уровне распознавания и обработки кадров и ячеек

Как уже отмечалось, передаваемые по транспортным системам данные объединяются в логически законченные структурные единицы: кадры, пакеты, ячейки и т. п. Поэтому приемник данных должен распознавать не только передаваемые биты, но и построенные из них кадры или иные информационные посылки. В этом смысле можно говорить о синхронизации приемника с передатчиком на уровне передачи таких посылок.

Обычно кадр начинается с флага – уникальной комбинации битов (например 01111110), которая не встречается в его оставшейся части. Чтобы достичь такой уникальности, оставшаяся часть кадра (область данных) перед выдачей в линию анализируется и при необходимости “разбавляется” вставкой служебных нулевых битов в имеющиеся длинные цепочки единиц (эта процедура называется битстаффингом). В данном примере флаг содержит шесть единиц; поэтому после каждых пяти единиц, следующих вплотную друг за другом в области данных, всегда (независимо от значения последующего бита) вставляется нулевой (служебный) бит. Тем самым исключается возможность последующего обнаружения удаленным устройством группы из шести или более следующих друг за другом единиц в области данных, что могло бы привести к ложному опознанию флага. Флаг при передаче не подвергается битстаффингу и поэтому сохраняет уникальность, что позволяет распознать его удаленным приемником.

После получения кадра, переданного по линии связи, и обнаружения начала области данных выполняется операция, обратная битстаффингу. Нулевой бит, следующий за каждой группой из пяти единиц в области данных, справедливо расценивается приемником как служебный и безусловно вычеркивается. Поэтому, в частности, длинные цепочки единиц смыкаются, полученное поле данных кадра обретает первоначальный вид. Из-за применения битстаффинга длина передаваемых по линии кадров непостоянна: например, кадр, содержащий в поле данных “сплошные нули”, передается по линии без изменения, а кадр со “сплошными единицами” в поле данных требует значительно разбавления нулями. (При постоянной длине кадра непостоянным окажется число размещенных в нем байтов данных.) Таким образом, избыточность потока передаваемых по линии данных обусловлена как применением многозарядного флага, так и битстаффингом.

В начале этой главы (п. 7.1) рассмотрены схемы распознавания многозарядных флагов в битовом и байтовом потоках данных. Далее (п. 7.2, 7.3) приведены схемы распознавания однозарядных и раздробленных флагов, а также методы определения границ ячеек без использования флаговых битов (п. 7.4). В решениях по п. 7.5 – 7.7 учитываются возможные искажения данных из-за помех в линии. Способ передачи данных по п. 7.8 позволяет повысить эффективность использования избыточных битов синхронизации. В п. 7.9 рассмотрены решения задачи экономичного размещения низкоскоростного потока данных в высокоскоростном потоке кадров. В конце главы (п. 7.10) приведен способ уменьшения числа операций, связанных с распознаванием флагов начала кадров.

Начнем с относительно простых решений.

7.1. Распознавание многоразрядного флага в битовом и байтовом потоках данных

7.1.1. Распознавание флага в битовых потоках данных

Схема, приведенная на рис. 7.1, а, предназначена для распознавания девятиразрядного флагового кода (флага) во входном потоке данных. В данном примере флаг выбран равным 000000001. Он предварительно загружается в девятиразрядный регистр RG2.

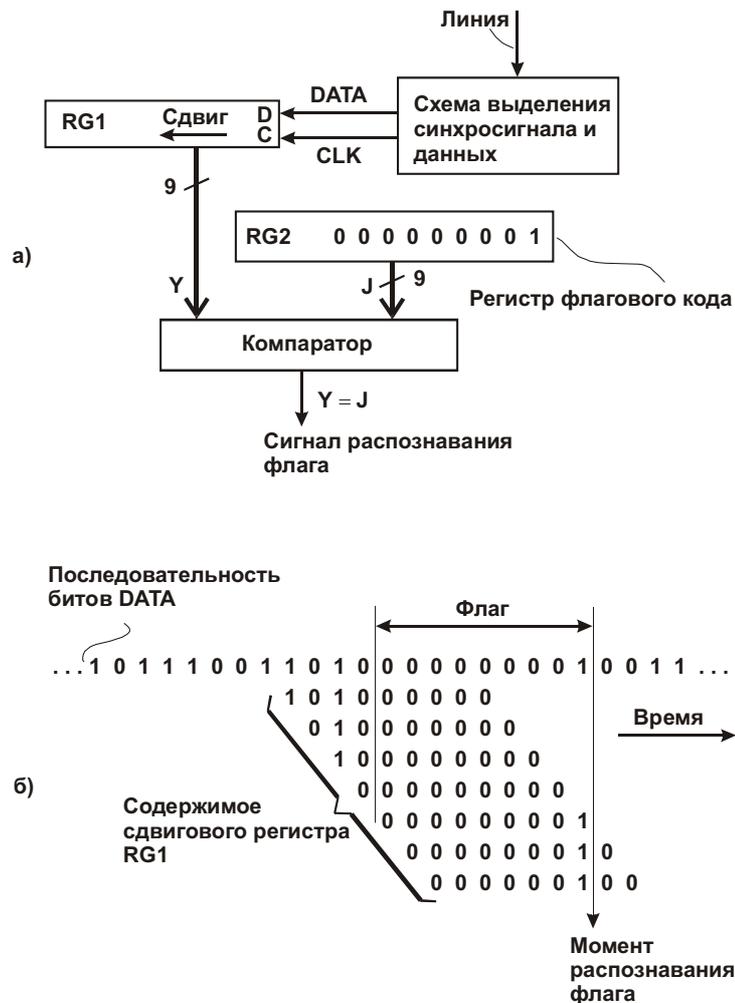


Рис. 7.1. Распознавание флага в битовом потоке данных: а – схема на основе многоразрядного компаратора; б – прохождение кода через сдвиговый регистр RG1

Из поступающего по линии сигнала выделяются синхросигнал CLK и данные DATA (код NRZ). Под действием синхросигнала данные загружаются в сдвиговый регистр RG1 и последовательно продвигаются в нем, как показано на рис. 7.1, б. При совпадении кодов в регистрах RG1 и RG2 срабатывает компаратор, на его выходе формируется сигнал распознавания флага. Длительность этого сигнала равна одному периоду (такту) синхросигнала CLK. В моменты изменения данных в регистре RG1 на выходе компаратора могут наблюдаться ложные срабатывания, поэтому для фиксации истинного значения выходного сигнала ($Y = J$) его следует принять на триггер (триггер на рисунке не показан). Синхронизация этого триггера может осуществляться задержанным (на время, превышающее задержку компаратора) сигналом CLK либо

непосредственно этим сигналом. В последнем случае результат сравнения кодов будет запаздывать на один такт.

Несмотря на простоту данного решения, его применимость ограничена сравнительно низкоскоростными системами передачи данных. Это связано с тем, что переходные процессы на выходе компаратора должны закончиться к началу очередного битового интервала (такта), длительность которого при высоких скоростях передачи данных может составлять, например, 3 нс или менее. С увеличением разрядности флагового кода число логических элементов компаратора возрастает, увеличивается его задержка; поэтому, чтобы уложиться в битовый интервал, необходимо применение все более быстродействующей элементной базы. А это нежелательно, так как повышается потребляемая мощность, уменьшается степень интеграции и т. д.

Чтобы снизить требования к быстродействию элементной базы, следует уменьшить число элементарных задержек на пути распространения сигнала в течение одного такта. Одним из широко применяемых методов уменьшения задержек (или, что то же самое – повышения максимальной тактовой частоты) является конвейерный метод обработки сигналов. Поясним его сущность на примере преобразования структуры рассмотренного ранее (рис. 7.1, а) компаратора к конвейерному виду. Предположим, что компаратор первоначально выполнен в виде пирамидальной комбинационной (не содержащей элементов памяти и цепей обратной связи) схемы, показанной на рис. 7.2, а. Суммарная задержка такого компаратора равна пяти задержкам логических элементов и должна быть меньше длительности такта.

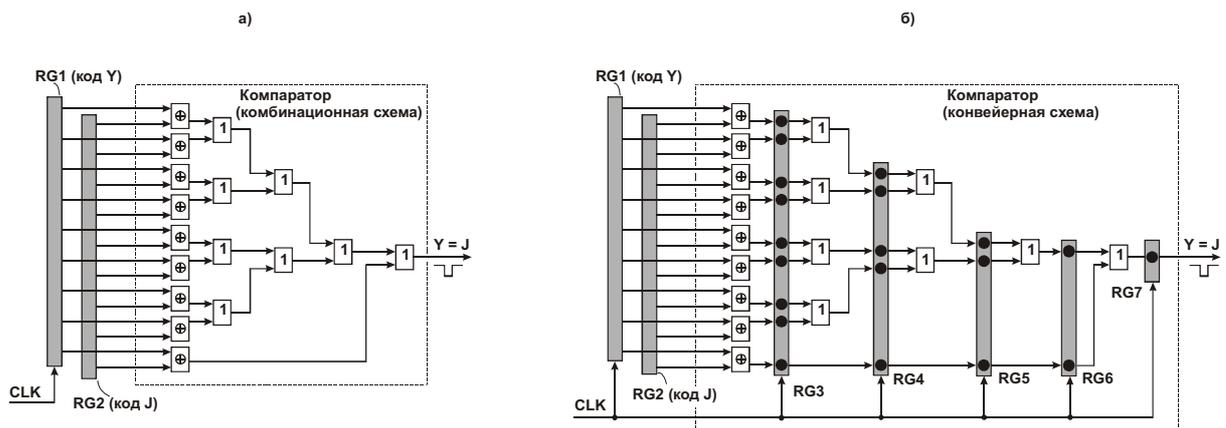


рис. 7.2. Преобразование комбинационной схемы компаратора в конвейерную для повышения максимальной тактовой частоты сравнения кодов Y и J

Группа элементов Иключающее ИЛИ выполняет поразрядное сравнение кода Y в сдвиговом регистре RG1 с флаговым кодом в регистре RG2. При совпадении кодов Y и J на выходах этих элементов формируются сигналы лог. 0, которые проходят через пирамидальную структуру из элементов на выход компаратора. Если коды Y и J не совпадают, на выходе компаратора формируется сигнал лог. 1.

Для преобразования комбинационной схемы компаратора в конвейерную она делится на ярусы, разделенные регистрами (RG3 – RG7), в которых фиксируются промежуточные и окончательный результаты вычислений (рис. 7.2, б). В данном примере использовано максимально возможное расслоение комбинационной схемы, при котором каждый ярус имеет задержку, не превышающую задержку одного логического элемента. Регистры RG3 – RG7 содержат соответственно девять, пять, три, два и один D-триггер, обозначенные на рисунке точками. Регистры RG1, RG3 – RG7 синхронизируются положительными фронтами тактового сигнала CLK.

Предположим, что в начале такта T_0 в регистр RG1 (рис. 7.2, б) поступил очередной код Y_0 для сравнения с флаговым кодом J , постоянно хранящемся в регистре

RG2. К моменту окончания такта T_0 (с некоторым запасом) на выходах элементов Исключающее ИЛИ сформирован результат поразрядного сравнения кодов Y_0 и J . В начале следующего такта T_1 этот результат запоминается в регистре RG3. Одновременно с этим данные в регистре RG1 сдвигаются на один разряд, в результате в регистре RG1 формируется код Y_1 .

К моменту окончания такта T_1 на входах регистров RG4 и RG3 сформированы установившиеся сигналы, отображающие промежуточные результаты обработки кодов Y_0 и Y_1 . Эти сигналы запоминаются в регистрах в начале такта T_2 , при этом в регистре RG1 формируется очередной код Y_2 и т. д. Таким образом, в начале такта T_5 в регистре RG7 зафиксирован окончательный результат обработки кода Y_0 , а в регистрах RG6 – RG3 – промежуточные результаты обработки кодов $Y_1 – Y_4$. Иными словами, процесс обработки, как и при использовании “настоящего” конвейера, распадается на ряд простых и быстро выполняемых операций.

Рассмотренное решение (рис. 7.2, б) позволяет регистрировать точные совпадения кодов Y и J . Однако на практике точные совпадения могут наблюдаться не всегда из-за наличия ошибок в канале связи. Поэтому для исчерпывающей оценки результатов поиска необходимы интеллектуальные системы, проявляющие “терпимость” к ошибкам. В частности, аппаратура поиска должна не только давать ответ типа “да – нет”, но и оценивать степень сходства сравниваемых кодов в случае их несовпадения.

Схема, представленная на рис. 7.3 [52], осуществляет конвейерную обработку битового потока данных DATA и в каждом такте формирует двоичное число $Z_3 – Z_0$ (Z_0 – младший разряд), которое отражает степень близости текущего кода к восьмиразрядному флаговому коду.

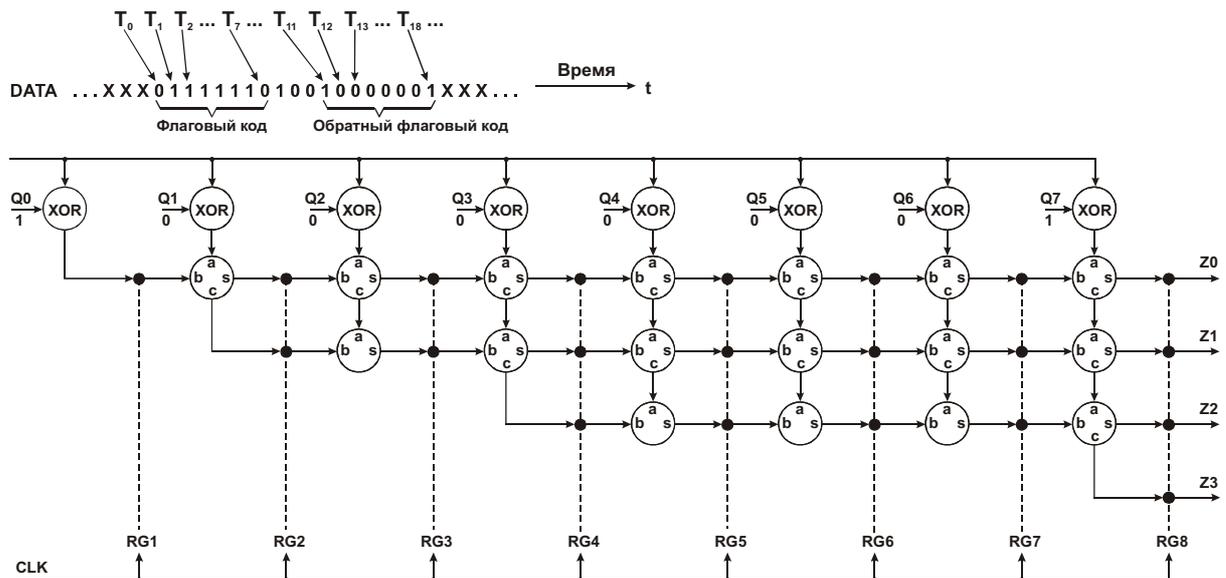


Рис. 7.3. Конвейерная схема распознавания флага в битовом потоке данных – первый вариант

Схема содержит восемь элементов Исключающее ИЛИ (XOR), семнадцать двухвходовых битовых сумматоров и восемь регистров RG1 – RG8. Разряды регистров выполнены на D-триггерах, условно показанных точками. Все триггеры синхронизируются положительными фронтами тактового сигнала CLK, сопровождающего биты данных DATA. Первые входы элементов Исключающее ИЛИ объединены и предназначены для приема последовательных данных DATA. На вторые входы этих элементов подан инвертированный флаговый код; прямой флаговый код равен 01111110_2 . Сумматор выполняет арифметическую операцию сложения двух однобитовых чисел a и b : $s = a \oplus b$; $c = a \& b$, где s – сумма, c – перенос в следующий разряд.

Процесс прохождения через схему битовой последовательности, приведенной в верхней части рис. 7.3, поясняется Таблица 7.1. табл. 7.1. Символ “X” на рисунке и в таблице обозначает произвольное значение бита.

Таблица 7.1. табл. 7.1

Последовательность состояний конвейерной схемы (рис. 7.3)

t	DATA	Коды на входах регистров							
		RG1	RG2	RG3	RG4	RG5	RG6	RG7	RG8
T ₀	0	1	X	X	X	X	X	X	X
T ₁	1	0	2	X	X	X	X	X	X
T ₂	1	0	1	3	X	X	X	X	X
T ₃	1	0	1	2	4	X	X	X	X
T ₄	1	0	1	2	3	5	X	X	X
T ₅	1	0	1	2	3	4	6	X	X
T ₆	1	0	1	2	3	4	5	7	X
T ₇	0	1	0	1	2	3	4	5	8
T ₈	1	0	2	1	2	3	4	5	5
T ₉	0	1	0	2	1	2	3	4	6
T ₁₀	0	1	1	0	2	1	2	3	5
T ₁₁	1	0	2	2	1	3	2	3	3
T ₁₂	0	1	0	2	2	1	3	2	4
T ₁₃	0	1	1	0	2	2	1	3	3
T ₁₄	0	1	1	1	0	2	2	1	4
T ₁₅	0	1	1	1	1	0	2	2	2
T ₁₆	0	1	1	1	1	1	0	2	3
T ₁₇	0	1	1	1	1	1	1	0	3
T ₁₈	1	0	2	2	2	2	2	2	0
T ₁₉	X	X	X	X	X	X	X	X	X
T ₂₀	X	X	X	X	X	X	X	X	X
T ₂₁	X	X	X	X	X	X	X	X	X

Предположим, что в тактах T₀ – T₇ на вход конвейерной схемы поступает последовательность битов 01111110, совпадающая с флаговой. В результате ее обработки, как показано в таблице, в такте T₇ на входе регистра RG8 формируется число 8, соответствующее последовательному восьмикратному совпадению поступивших битов DATA с соответствующими битами флагового кода. Этот результат получен следующим образом.

В такте T₀ DATA = 0, сигнал Q₀ постоянно равен лог. 1, поэтому на входе одноразрядного регистра RG1 сформирован сигнал лог. 1 ($0 \oplus 1 = 1$). Так как исходное состояние регистров RG1 – RG8 не определено, то значения сигналов на входах реги-

стров RG2 – RG8 могут быть произвольными (см. символы “X” в первой строке таблицы).

В такте T_1 (вторая строка таблицы) DATA = 1, на входе регистра RG1 сформирован сигнал лог. 0 ($1 \oplus 1 = 0$). На вход регистра RG2 поступает число 2, так как на входы левого (по схеме) сумматора поданы две единицы. Первая из них поступает из регистра RG1 как результат вычисления, выполненного в предыдущем такте, вторая получена суммированием по модулю два сигналов DATA = 1 и $Q1 = 0$. Состояния сигналов на входах регистров RG3 – RG8 по-прежнему не определены.

В такте T_2 (третья строка таблицы) DATA = 1, на входах регистров RG1 – RG3 сформированы числа 0, 1 и 3. Из структуры данных в таблице видно, что с течением времени число неопределенных состояний (“X”) кодов в регистрах уменьшается, при этом они замещаются последовательно возрастающими числами 1, 2, 3 и т. д., размещенными по диагонали таблицы. Такая закономерность формирования чисел связана с тем, что в данном примере на вход схемы поступает код, совпадающий с флаговым. Поэтому амплитуда фронта проходящей по конвейеру “волны вычислений” неуклонно растет.

В конце такта T_7 на входе регистра RG8 сформирован код 8, который запоминается в этом регистре в начале такта T_8 . Как показано в правом столбце таблицы, в последующих тактах в регистре RG8 формируются коды 5, 6, 5, 3, 4, 3, 4, 2, 3, 3, 0. Последний (нулевой) код соответствует полному отсутствию совпадений битов кода DATA с битами флагового кода в период $T_{11} - T_{18}$.

Чтобы убедиться в правильности полученной последовательности кодов в регистре RG8 и пояснить их происхождение, рассмотрим модель вычислений, приведенную на рис. 7.4.

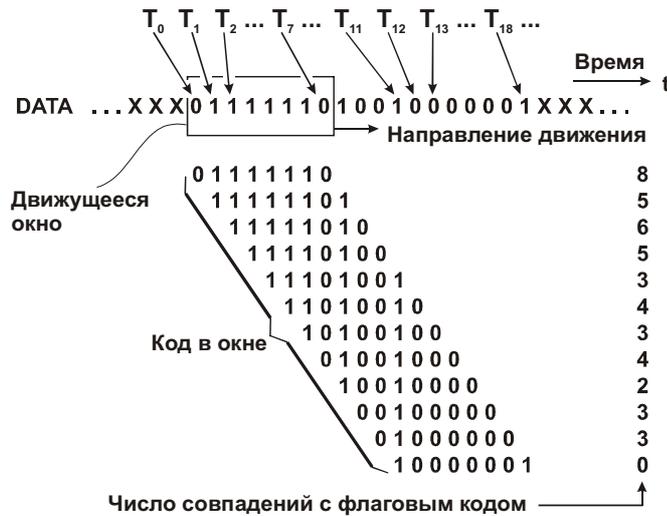


Рис. 7.4. Модель вычислений, поясняющая процесс конвейерной обработки кода DATA (см. рис. 7.3)

В данной модели конвейерной обработки кода DATA этот код просматривается через движущееся восьмиразрядное окно. Если окно зафиксировано в показанном на рисунке положении, то сквозь него просматривается код 01111110, совпадающий с флаговым, и, следовательно, совпадающий с ним восьми разрядах. В следующем такте окно перемещается на один разряд вправо, поэтому через него виден код 11111101. Этот код совпадает с флаговым в пяти разрядах. Как видно из рисунка, в последующих тактах числа (6, 5, 3, ...), отражающие совпадения, полностью соответствует числам, приведенным в правом столбце таблицы.

Недостатком рассмотренной схемы (см. рис. 7.3) является сравнительно невысокая тактовая частота ее работы. Действительно, длительность такта (время между двумя положительными фронтами сигнала CLK) должна быть достаточной для надежного

		-	-	-	X	X	X	X	X	X	-		
		-	-	-	-	J	J	J	J	J	0		
		-	-	-	-	-	-	-	-	-	1		
T ₄₀	1	0	X	X	X	X	X	X	X	X	X	X	
		-	X	X	X	X	X	X	X	X	-		-
		-	-	X	X	X	X	X	X	X	X		X
		-	-	-	X	X	X	X	X	X	X		-
		-	-	-	-	X	X	X	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₁	0	J	0	X	X	X	X	X	X	X	X	X	
		-	0	X	X	X	X	X	X	X	-		-
		-	-	X	X	X	X	X	X	X	X		X
		-	-	-	X	X	X	X	X	X	X		-
		-	-	-	-	X	X	X	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₂	0	J	J	0	X	X	X	X	X	X	X	X	
		-	J	0	X	X	X	X	X	X	-		-
		-	-	0	X	X	X	X	X	X	X		X
		-	-	-	X	X	X	X	X	X	X		-
		-	-	-	-	X	X	X	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₃	0	J	J	J	0	X	X	X	X	X	X	X	
		-	J	J	0	X	X	X	X	X	-		-
		-	-	J	0	X	X	X	X	X	X		X
		-	-	-	0	X	X	X	X	X	X		-
		-	-	-	-	X	X	X	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₄	0	J	J	J	J	0	X	X	X	X	X	X	
		-	J	J	J	0	X	X	X	X	-		-
		-	-	J	J	0	X	X	X	X	X		X
		-	-	-	J	0	X	X	X	X	X		-
		-	-	-	-	0	X	X	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₅	0	J	J	J	J	J	0	X	X	X	X	X	
		-	J	J	J	J	0	X	X	X	-		-
		-	-	J	J	J	0	X	X	X	X		X
		-	-	-	J	J	0	X	X	X	X		-
		-	-	-	-	J	0	X	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₆	0	J	J	J	J	J	J	0	X	X	X	X	
		-	J	J	J	J	J	0	X	X	-		-
		-	-	J	J	J	J	0	X	X	X		X
		-	-	-	J	J	J	0	X	X	X		-
		-	-	-	-	J	J	0	X	X	X		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₇	1	J	J	J	J	J	J	J	0	X	X	X	
		-	J	J	J	J	J	J	0	-	-		
		-	-	J	J	J	J	J	0	X	X		

		-	-	-	J	J	J	J	0	X	-		
		-	-	-	-	J	J	J	0	X	X		
		-	-	-	-	-	-	-	-	-	X		
T ₄₈	X	X	X	X	X	X	X	X	X	0	X	X	
		-	X	X	X	X	X	X	X	X	-		-
		-	-	J	J	J	J	J	J	J	0		X
		-	-	-	J	J	J	J	J	J	0		-
		-	-	-	-	J	J	J	J	J	0		X
		-	-	-	-	-	-	-	-	-	-		X
T ₄₉	X	X	X	X	X	X	X	X	X	X	0	0	
		-	X	X	X	X	X	X	X	X	-		-
		-	-	X	X	X	X	X	X	X	X		0
		-	-	-	X	X	X	X	X	X	X		-
		-	-	-	-	J	J	J	J	J	J		0
		-	-	-	-	-	-	-	-	-	-		0

Процессы, протекающие в схеме (рис. 7.5), в целом аналогичны рассмотренным при описании предыдущей схемы (рис. 7.3). Основное отличие связано с упоминавшейся ранее “отложенной” обработкой переносов из разряда в разряд, что увеличивает длину конвейера, но уменьшает время выполнения медленных операций. Из таблицы следует, что результат распознавания флага формируется на входе выходного регистра в такте T₉, а не в такте T₇, как в предыдущей схеме.

Отметим, что задержку комбинационных схем, включенных между регистрами, можно дополнительно уменьшить включением дополнительного восьмиразрядного регистра (с синхронизацией от сигнала CLK) в разрыв цепей между выходами элементов Иключающее ИЛИ и входами последующих элементов (D-триггера регистра RG1 и сумматоров). В этом случае задержки элементов Иключающее ИЛИ не будут складываться с задержками сумматоров. Можно пойти в этом направлении и далее – ввести регистры внутрь структур элементов Иключающее ИЛИ и сумматоров, чтобы свести задержку межрегистровой комбинационной схемы к задержке одного простейшего логического элемента типа И, ИЛИ, НЕ.

Рассмотрим некоторые модификации конвейерных схем. В схеме, приведенной на рис. 7.6, данные DATA поступают от приемника по двум каналам (а не по одному, как в предыдущих решениях). Такая организация потока данных свойственна, например, приемнику линейного сигнала с кодировкой 2B1Q. Правила кодирования следующие. Каждая пара выдаваемых в линию битов (“2B” – two binary) преобразуется в один из четырех (“1Q” – one of quaternary) уровней напряжения между проводами линии. В осциллограмме линейного сигнала при передаче случайных данных просматриваются четыре фиксированных уровня напряжения со случайными переходами между ними на границах между тактами. Приемник в каждом такте анализирует уровень сигнала и преобразует его в пару битов, которая одновременно выдается по двум каналам: DATA0 и DATA1.

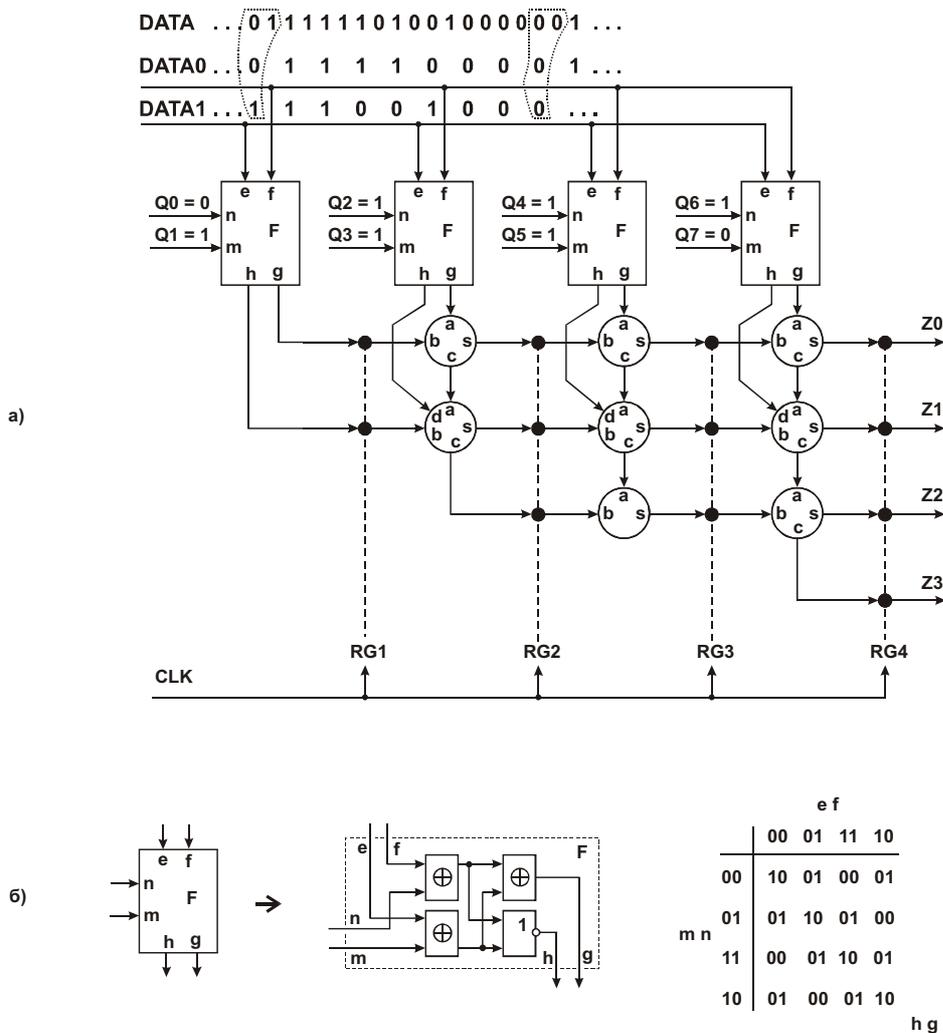


Рис. 7.6. Распознавание восьмибитового флага в битовом потоке данных: *a* – конвейерная схема – третий вариант; *b* – структура логического блока *F* и его таблица истинности

Данные из каналов DATA0 и DATA1 сравниваются логическими блоками *F* с соответствующими битами известного флагового кода. При одновременном совпадении бита *f* с битом *n* и бита *e* с битом *m* логический блок формирует двоичное число 2 ($hg = 10_2$). При неполном совпадении логический блок формирует число 1 ($hg = 01_2$). В отсутствие совпадений формируется нулевой код ($hg = 00_2$). Таким образом, логический блок осуществляет предварительную оценку степени совпадения кодов, которая далее учитывается при накоплении результирующей суммы, как уже было показано при описании предыдущих схем. В схеме (рис. 7.6) вместо двухвходовых применены три трехвходовых сумматора, выполняющих следующие функции: $s = a \oplus b \oplus d$; $c = a \& b + b \& d + a \& d$, где знак “+” соответствует логической операции ИЛИ, s – сумма, c – перенос в следующий разряд. Максимальное число в регистре RG4 равно восьми и соответствует опознанию флагового кода.

Схему, представленную на рис. 7.6 (и последующую, рис. 7.7), можно усовершенствовать для повышения тактовой частоты ее работы подобно тому, как это было сделано ранее (см. рис. 7.5), однако соответствующие решения здесь не приводятся.

Другая модификация конвейерной схемы (рис. 7.7) [52] позволяет работать с приемниками, имеющими встроенную аппаратуру контроля качества принятого сигнала. Если приемник гарантирует достоверность бита, выделенного из линейного сигнала, то комбинация сигналов $W1W2 = 00$ соответствует сигналу лог. 0, а комбинация $W1W2 = 11$ – сигналу лог. 1. Комбинации $W1W2 = 01$ и $W1W2 = 10$ формируются приемни-

ком при обнаружении ошибок, не позволяющих уверенно определить значение принятого бита.

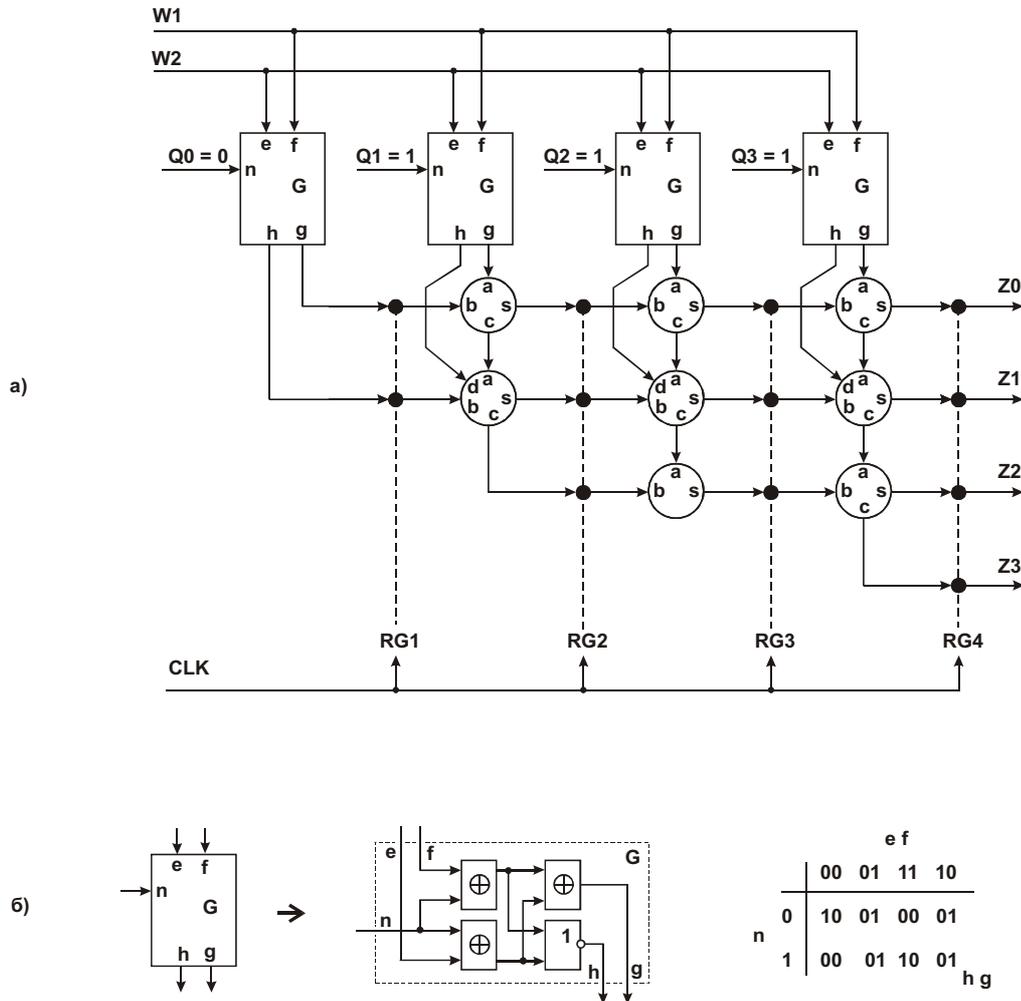


Рис. 7.7. Распознавание четырехразрядного флага в битовом потоке данных: а – конвейерная схема – четвертый вариант; б – структура логического блока G и его таблица истинности

Блок G формирует число 2 ($hg = 10_2$) при наличии сигнала, уверенно распознанного приемником, если этот сигнал логически совпадает с поданным на вход n. Если уверенно распознанный приемником сигнал не совпадает с сигналом, поданным на вход n, то на выходах блока G формируется нулевой код ($hg = 00_2$). Неуверенная работа приемника сопровождается выдачей на выходы логического блока числа 1 ($hg = 01_2$).

После предварительного анализа данные с выходов логических блоков G обрабатываются точно также, как и в предыдущих схемах. Результат обработки – поток четырехразрядных чисел на выходе регистра RG4 – оценивается последующими устройствами. При уверенном приеме сигнала и точном совпадении принятого кода с четырехразрядным флаговым кодом (0111) в регистре RG4 в определенный момент возникает число 8. Неуверенный прием или (и) несовпадение принятых битов с флаговыми приводят к уменьшению этого числа.

С другими решениями задачи распознавания заданных кодов или их последовательностей в проходящем потоке данных Вы можете ознакомиться в [71, 72].

разрядное слово, в котором, возможно, содержится искомый девятиразрядный флаг. В момент очередного продвижения потока байтов (при поступлении сигнала CLK/8) наблюдаемое 16-разрядное слово смещается на один байт в сторону новых данных.

В каждом такте сигнала CLK/8 байты BYTE_OLD и BYTE_NEW анализируются для обнаружения в них левого и правого фрагментов флага (000000001). Для анализа используются блоки памяти RAM_OLD и RAM_NEW с соответствующими кодировками. Байты BYTE_OLD и BYTE_NEW используются для адресации ячеек этих блоков. Число предполагаемых совпадений соответствующих байтов с фрагментами флага отображается четырехразрядными кодами данных Q_OLD и Q_NEW, которые считываются из блоков памяти. Каждый из этих кодов принадлежит диапазону 0 – 8. Эти коды арифметически суммируются, в результате формируется двоичное число Y, по которому можно судить о наличии (или отсутствии) флага в паре регистров RG2 – RG3.

Чтобы получить окончательный результат анализа, число Y сравнивается с числом J (отображающим разрядность флага), предварительно загруженным в регистр длины флага. В данном примере использован девятиразрядный флаг, поэтому J = 9. Сигнал распознавания флага формируется при $Y \geq J$. При этом информация о размещении флага в наблюдаемом 16-разрядном слове (BYTE_OLD – BYTE_NEW) содержится в коде Q_OLD или (и) Q_NEW (в данном примере – в последнем).

Далее рассмотрены подробности этого решения.

Как показано в верхней части рис. 7.9, возможны восемь вариантов размещения флага в байтах BYTE_OLD – BYTE_NEW. Неоднозначность размещения связана с тем, что начальное состояние делителя частоты сигнала CLK не определено, и “нарезка” битовой последовательности на байты может начаться с произвольного места. В нижней части рисунка показаны схемы вычисления переменных Q_OLD и Q_NEW. Эти переменные отражают предполагаемое число битовых совпадений кодов с левым и правым фрагментами флага. Из верхней строки схемы вычисления переменной Q_OLD следует, что нулевой код в байте BYTE_OLD может (но не обязательно должен) соответствовать обнаружению восьми нулевых битов флага, поэтому $Q_OLD = 8$. Следующая строка показывает возможность совпадения кода с флагом в семи разрядах и т. д. Аналогичные рассуждения применимы и к схеме вычисления переменной Q_NEW.

Условие распознавания флага, как уже отмечалось, состоит в том, что

$$Y = Q_OLD + Q_NEW \geq 9.$$

Чтобы убедиться в его правильности, рассмотрим все благоприятные (соответствующие обнаружению флага) сочетания фрагментов кодов, приведенных в нижней части рис. 7.9. Эти сочетания обозначим двузначными числами, в которых первое соответствует коду Q_OLD, а второе – коду Q_NEW. Так, например, число 88 соответствует сочетанию фрагментов кодов 0000 0000 (BYTE_OLD) и 0000 0001 (BYTE_NEW). Полный 16-разрядный код равен 0000 0000 0000 0001. В правой части этого кода присутствует искомый девятиразрядный флаг 000000001. Его местоположение однозначно определяется кодом $Q_NEW = 8$. Еще один пример: числу 37 соответствует 16-разрядный код XXXX 1000 0000 001X, в котором содержится искомый флаг, и т. д. (Символы “X” соответствуют произвольным значениям битов.) Итак, благоприятные сочетания следующие:

88, 87, 86, 85, 84, 83, 82, 81,
78, 77, 76, 75, 74, 73, 72,
68, 67, 66, 65, 64, 63,
58, 57, 56, 55, 54,
48, 47, 46, 45,
38, 37, 36,
28, 27,
18.

границей, разделяющей байт BYTE_NEW на две равные части (так как иное размещение границ привело бы к другому результату вычисления кода BYTE_NEW).

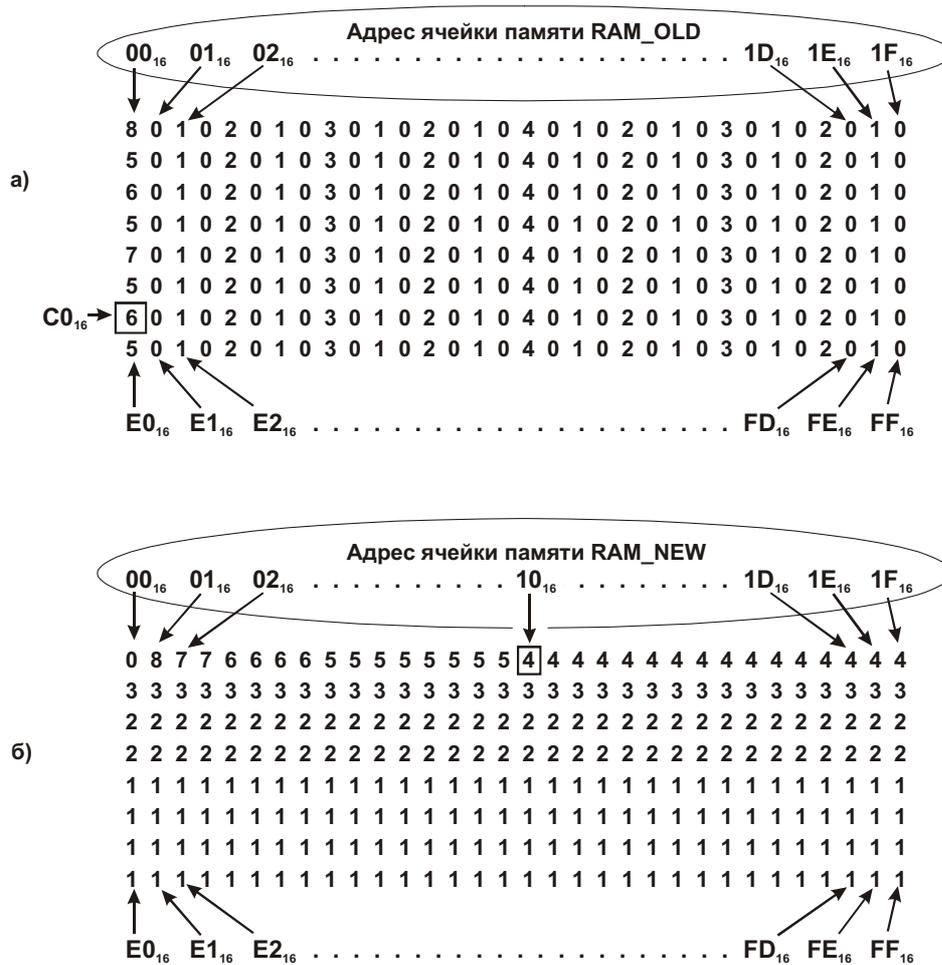


Рис. 7.10. Кодировка блоков памяти RAM_OLD (а) и RAM_NEW (б)

Отметим, что в данном примере левая граница флага “размыта”, т. е. может сливаться с цепочкой соседних нулевых битов, как в только что рассмотренном примере. Поэтому определение положения флага по его левой границе в данном случае невозможно.

В [55] рассмотрены также более сложные решения, при которых флаг размещается в нескольких байтах.

С точки зрения пользователя транспортной системы флаг представляется лишним элементом, и поэтому неплохо было бы если не избавиться от него (как ни странно, и это возможно, см. п. 7.4), то хотя бы уменьшить его длину, чтобы повысить скорость передачи данных.

7.2. Минимизация длины флага начала кадра

В рассмотренных далее решениях (п. 7.2.1 и 7.2.2) длина флага уменьшена до одного бита, при этом длина кадра постоянна. Разумеется, такой флаг не уникален. После установления синхронизации приемнику точно известно ожидаемое положение очередного флага и вычислено его значение (0 или 1). В первом решении значения флаговых битов чередуются во времени; во втором – представлены псевдослучайной последовательностью битов, что позволяет вести нумерацию кадров.

7.2.1. Использование чередующихся нулевых и единичных битов в качестве флагов начала кадров

Предлагаемое решение задачи минимизации длины флага основано на использовании чередующихся нулевых и единичных битов в качестве флагов начала кадров [14] и поясняется схемой, приведенной на рис. 7.11. Низкоскоростные потоки данных из каналов 1 – N поступают на входы мультиплексора MUX 1 (MUX 2). Сумма этих потоков в виде последовательности кадров пересылается с относительно высокой скоростью по линии связи к удаленному мультиплексору MUX 2 (MUX 1), который распознаёт кадр и распределяет данные по соответствующим каналам.

В предлагаемом решении битстаффинг не нужен, поэтому передаваемый по линии кадр имеет фиксированную длину, флаг представлен одним битом в каждом кадре.

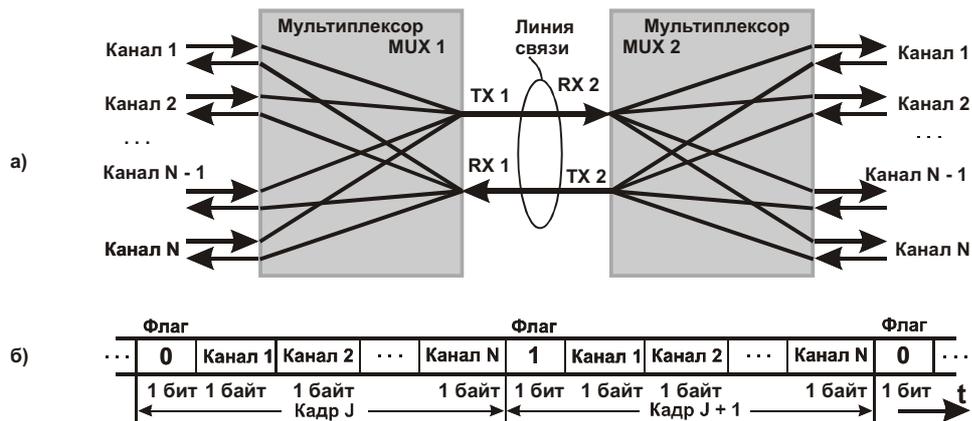


рис. 7.11. Мультиплексная передача данных с временным разделением каналов: а – структурная схема; б – структура одного из потоков данных в линии связи (TX 1 → RX 2 или TX 2 → RX 1)

Идея заключается в следующем. В установившемся режиме после вхождения мультиплексоров в логическое соединение, а затем и в кадровый синхронизм каждый мультиплексор помечает передаваемые им в линию последовательные кадры чередующимися одноразрядными флаговыми битами, как показано на рис. 7.11, б. Некоторый кадр с номером J (нумерация условная) содержит флаг, равный лог. 0. Следующий кадр с номером J + 1 имеет флаг, равный лог. 1. Последующие кадры содержат флаги, равные соответственно лог. 0, лог.1, лог. 0 и т. д.

Принимая кадр из линии, каждый мультиплексор проверяет соответствие полученного значения флагового бита ожидаемому. Например, после получения кадра с флаговым битом, равным лог. 0, предсказывается поступление кадра с флаговым битом, равным лог. 1. Позиция ожидаемого флагового бита известна, так как кадр имеет фиксированную длину, а предыдущая точка отсчета границы кадра уже отслежена в предыстории.

В отсутствие ошибок наблюдается правильное чередование флаговых битов. Ошибки, например проскальзывания, в конечном счете приведут к нарушению такого чередования и, следовательно, будут обнаружены, но, возможно, не сразу. При этом синхронизация будет временно потеряна и затем вновь восстановлена.

Вхождение мультиплексоров в кадровую синхронизацию

Для начала нужно установить связь между мультиплексорами по линии. Линия может быть выделенной или коммутируемой, содержащей ретрансляторы или иные устройства.

Каждый мультиплексор (MUX 1, MUX 2) имеет встроенный модем, подключенный к линии (модемы на рисунках не показаны). Один из модемов является вызывающим, т. е. инициатором установления связи, другой – вызываемым. В результате начального взаимодействия модемов (например по протоколу V.42 XID) и переговоров между ними, в некоторый момент мультиплексоры оказываются на связи и готовы передавать друг другу данные.

Нас сейчас интересует последующая фаза взаимодействия между мультиплексорами – установление кадровой синхронизации.

Последовательность событий такова.

1. Сразу после установления связи между модемами мультиплексоры начинают посылать друг другу непрерывные и неструктурированные потоки G, состоящие из лог. 1 (рис. 7.12, а, рис. 7.13).

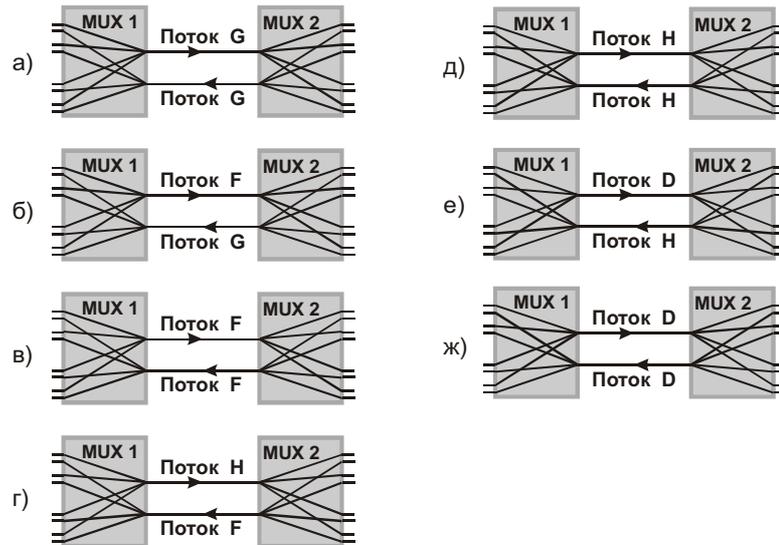


Рис. 7.12. Стадии а – ж установления кадровой синхронизации между мультиплексорами MUX 1 и MUX 2. Структура потоков F, G, H и D приведена на рис. 7.13

2. Один из мультиплексоров, например тот, по чьей инициативе организована связь (в данном примере мультиплексор MUX 1), через некоторое время начинает формировать поток F (см. рис. 7.12, б, рис. 7.13). Мультиплексор MUX 2 по-прежнему выдает поток G.

3. Мультиплексор MUX 2 обнаруживает в поступающем на его вход потоке F периодическое появление комбинаций 101111101 и расценивает второй лог. 0 каждой такой комбинации как флаг начала смежной пары кадров. Зная длину кадра, мультиплексор MUX 2 расставляет недостающие (единичные) флаги, как это сделано на рис. 7.13. Задача наполовину решена – мультиплексор MUX 2 теперь способен распознавать границы кадров, присылаемых мультиплексором MUX 1. Факт установления кадровой синхронизации по направлению MUX 1 → MUX 2 внешне проявляется в том, что мультиплексор MUX 2 вместо потока G начинает выдавать поток F (см. рис. 7.12, в).

4. Мультиплексор MUX 1 обнаруживает в поступающем на его вход потоке F периодическое появление комбинаций 101111101 и расценивает второй лог. 0 каждой такой комбинации как флаг начала смежной пары кадров. Зная длину кадра, мультиплексор MUX 1 расставляет недостающие (единичные) флаги, как это сделано на рис. 7.13. Теперь синхронизация установлена в обоих направлениях, и можно начинать переход к рабочему режиму. Для этого мультиплексор MUX 1 оповещает партнера об успешном установлении двусторонней кадровой синхронизации переходом к выдаче

потока Н с сохранением достигнутой ранее кадровой синхронизации (рис. 7.13). Новое состояние системы показано на рис. 7.12, *з*.

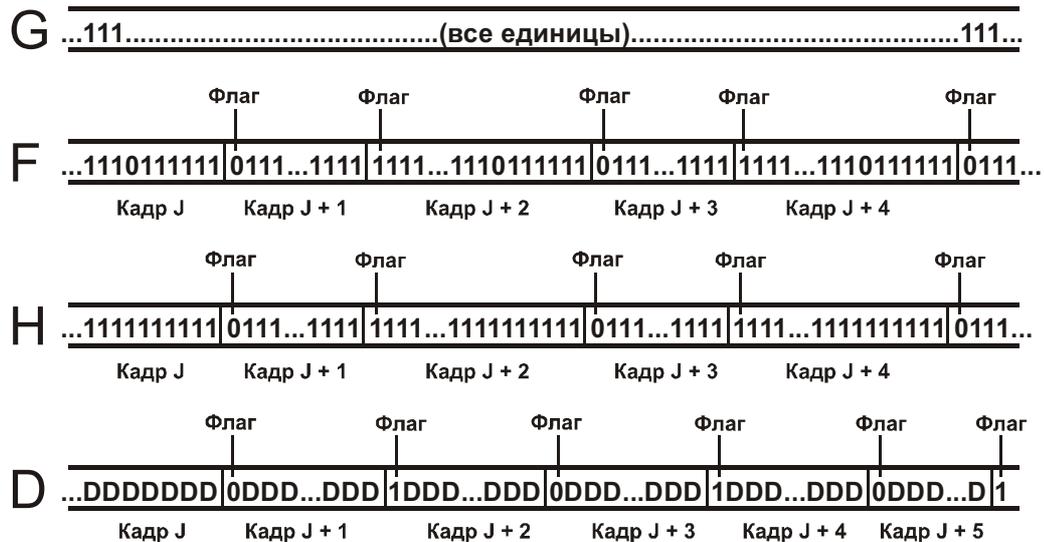


рис. 7.13. Потоки фиксированных данных (диаграммы G, F, H), используемые на этапе установления синхронизации, и поток полезных данных (диаграмма D)

5. Мультиплексор MUX 2 подтверждает готовность перехода к рабочему режиму выдачей партнеру потока Н с сохранением достигнутой ранее кадровой синхронизации (см. рис. 7.12, *д*).

6. Мультиплексор MUX 1 обнаруживает на входе поток Н и переходит к выдаче потока D (см. рис. 7.13) с сохранением достигнутой ранее кадровой синхронизации. Этот поток соответствует рабочему режиму временного разделения каналов 1 – N (см. рис. 7.11). Состояние системы показано на рис. 7.12, *е*.

7. Мультиплексор MUX 2 через некоторое время после начала выдачи потока Н (см. п. 5) переключается на выдачу потока D (см. рис. 7.13) с сохранением достигнутой ранее кадровой синхронизации. Этот поток соответствует рабочему режиму временного разделения каналов 1 – N (см. рис. 7.11). Состояние системы показано на рис. 7.12, *ж*.

Таким образом, синхронизация достигнута, система обслуживает каналы 1 – N.

Отметим, что каждый мультиплексор содержит внутренний таймер, который на соответствующих этапах установления синхронизации контролирует время ожидания правильных ответных реакций партнера. При превышении допустимого времени ожидания мультиплексор переходит к выдаче партнеру неструктурированного потока G (см. рис. 7.13). Партнер теряет синхронизацию и также переходит к выдаче потока G. В результате оба мультиплексора оказываются в рассмотренном ранее исходном состоянии (см. рис. 7.12, *а*), и попытка вхождения в синхронизм повторяется.

Потеря и восстановление синхронизации

Если в процессе или после установления взаимной синхронизации мультиплексоров хотя бы один из них обнаруживает неправильную последовательность флаговых битов (отличную от последовательности вида ...010101...), то это означает, что синхронизация потеряна.

В качестве примера поведения системы при потере синхронизации рассмотрим последовательность событий, приведенную на рис. 7.14.

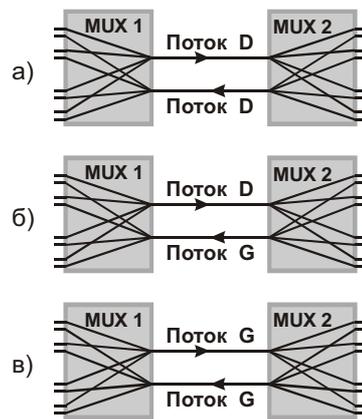


рис. 7.14. Последовательность состояний системы при потере синхронизации мультиплексором MUX 2

1. В исходном состоянии (рис. 7.14, а) оба мультиплексора находятся в рабочем режиме и формируют потоки D (см. рис. 7.13 и рис. 7.11).

2. В некоторый момент мультиплексор MUX 2 обнаруживает ошибку в последовательности полученных флаговых битов: вместо ожидаемого лог. 0 принята лог. 1, или наоборот. Так как мультиплексор MUX 2 не может справиться с проблемой в одиночку (у него нет ориентиров для поиска очередного флагового бита), он обращается за помощью к мультиплексору MUX 1 выдачей ему неструктурированного потока G (рис. 7.14, б).

3. Мультиплексор MUX 1 не обнаруживает во входном потоке G флаговых битов и теряет синхронизацию с мультиплексором MUX 2 (столь грубая просьба о помощи воспринята). В ответ на потерю синхронизации мультиплексор MUX 1 посылает в мультиплексор MUX 2 поток G. Таким образом, ситуация усугубляется – синхронизация отсутствует уже в обоих направлениях (рис. 7.14, в). Мультиплексоры передают друг другу потоки G. А это и есть начальное условие для запуска уже рассмотренной процедуры установления синхронизации (см. рис. 7.12, а).

Через определенное время, заданное таймером мультиплексора MUX 1, система переходит в состояние, показанное на рис. 7.12, б, и т. д. Процесс завершается полным восстановлением синхронизации (см. рис. 7.12, ж).

Отметим, что для установления синхронизации можно воспользоваться методом статистического анализа потока принимаемых данных (см. п. 7.3).

7.2.2. Использование псевдослучайных битов в качестве флагов начала кадров

В только что описанной системе передачи данных последовательность передаваемых кадров сопровождается чередующимися одноразрядными флаговыми битами: "...010101...". В установившемся режиме приемник следит за правильностью следования флаговых битов и обрабатывает принимаемые кадры. Однако однородный поток кадров в большинстве случаев необходимо структурировать, т. е. пометить в нем более крупные информационные объекты, например, группы кадров. Начало группы можно пометить, например, установкой в единицу некоторого служебного бита в заголовке кадра, но такое решение неэкономично. Нельзя ли совместить использование одноразрядных флаговых битов с разметкой относительного положения кадров в их непрерывном потоке?

Чтобы решить эту задачу, следует как-то преобразовать однородную последовательность "...010101...". Для введения признака начала группы кадров можно было бы

ввести в эту последовательность преднамеренное нарушение, например заменой нуля единицей или наоборот. Однако такая замена не обеспечивает хорошей отличимости внесенного нарушения от искажений сигнала в результате действия помех в линии. Еще одна проблема заключается в том, что аппаратура формирования кадровых битов и их распознавания должна быть достаточно простой, процесс вхождения в синхронизацию должен обладать быстрой сходимостью. Этим требованиям отвечает рассмотренное далее решение [75, 76], рис. 7.15.

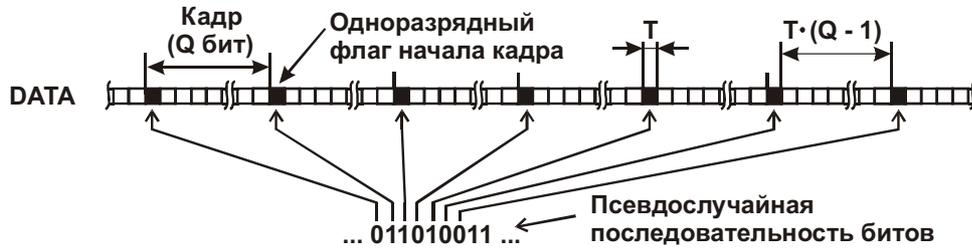


рис. 7.15. Схема, поясняющая идею использования псевдослучайных битов в качестве флагов начала кадров; DATA – поток битов между передатчиком и приемником

Так же, как и в предыдущем решении (см. рис. 7.11) начало кадра обозначается флаговым битом. Но в данном случае цепь из флаговых битов нетривиальна по структуре и формируется с помощью генератора псевдослучайной последовательности битов (рис. 7.16).

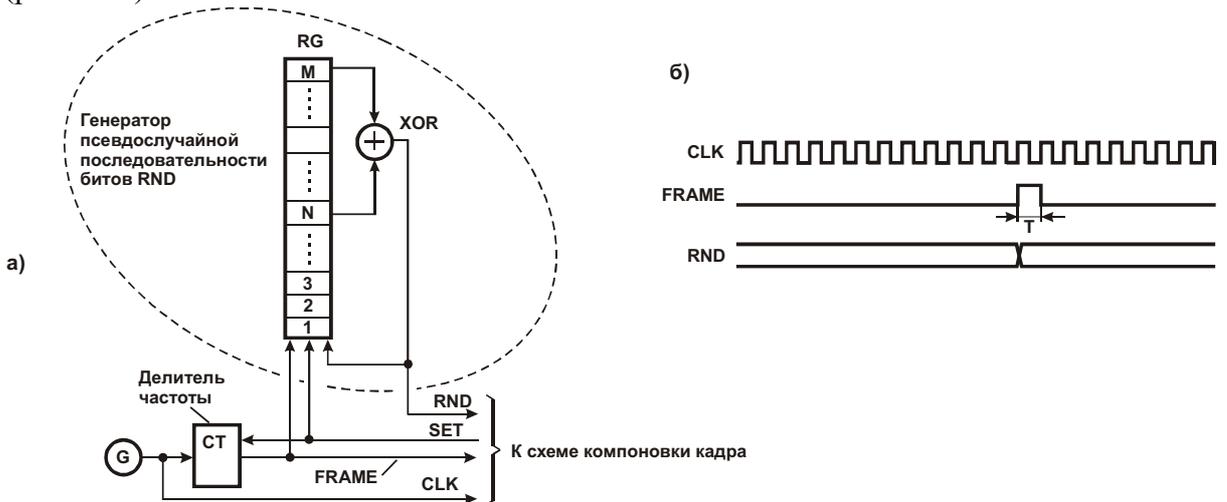


рис. 7.16. Формирование флаговых битов: а – схема; б – временная диаграмма

Предположим, что число битов в кадре равно Q , частота сигнала CLK от генератора G , задающего скорость передачи данных, равна F Гц. Длительность T битового интервала составляет $1/F$ с. Генератор псевдослучайной последовательности битов RND построен по классической схеме (она подробно рассмотрена в п. 8.4.1) на основе M -разрядного сдвигового регистра RG и элемента Иключающее ИЛИ (XOR), входы которого соединены с выходами разрядов M и N регистра. Выход элемента XOR соединен с входом данных регистра RG . Делитель частоты (счетчик по модулю Q) снижает частоту F сигнала CLK в Q раз и формирует кадровый импульс $FRAME$ длительностью T , который поступает в схему компоновки кадра и одновременно воздействует на вход синхронизации регистра RG генератора псевдослучайной последовательности битов. По положительному фронту сигнала $FRAME$ этот генератор формирует очередной псевдослучайный бит RND . Схема компоновки кадра при обнаружении импульса $FRAME$ считывает бит RND и использует его для вставки очередного флага в поток данных, как было показано на рис. 7.15.

По сигналу SET регистр RG устанавливается в состояние 111...1, а делитель частоты – в нулевое состояние. Такая установка позволяет задать исходную точку в системе псевдослучайного отсчета передаваемых кадров. В этой системе отсчета последовательность кадров неявно нумеруется кодами в регистре RG. Эти коды формируются в заранее известном и строго определенном порядке, зависящем от разрядности регистра и точки подключения к нему обратной связи. Так, предварительно установленный в состояние 111...1 регистр RG после воздействия первого импульса FRAME переходит в состояние 011...1 (первый разряд – слева), после второго импульса – в состояние 001...1 и т. д. В полном цикле работы генератора псевдослучайной последовательности битов в регистре однократно формируются все возможные коды, за исключением нулевого. Полученные таким способом M-разрядные номера кадров в явном виде не передаются в удаленный приемник, но могут быть восстановлены им при реставрации псевдослучайной последовательности битов, как будет показано далее.

В протоколе обмена данными номера кадров определяют их статус. Например, при $M = 5$ по линии связи непрерывно передаются группы из $2^5 - 1 = 31$ кадр. В каждой группе кадр с номером 11111_2 рассматривается как начальный. Приемная аппаратура распознаёт эти кадры по их номерам и трактует их содержимое по-разному, в соответствии с принятым протоколом.

Сформированная последовательность кадров с одноразрядными флаговыми битами (рис. 7.15) передается по линии связи в приемную аппаратуру. Для распознавания границ и номеров кадров применяется схема (рис. 7.17) на основе такого же генератора псевдослучайной последовательности, как и в схеме формирования флаговых битов.

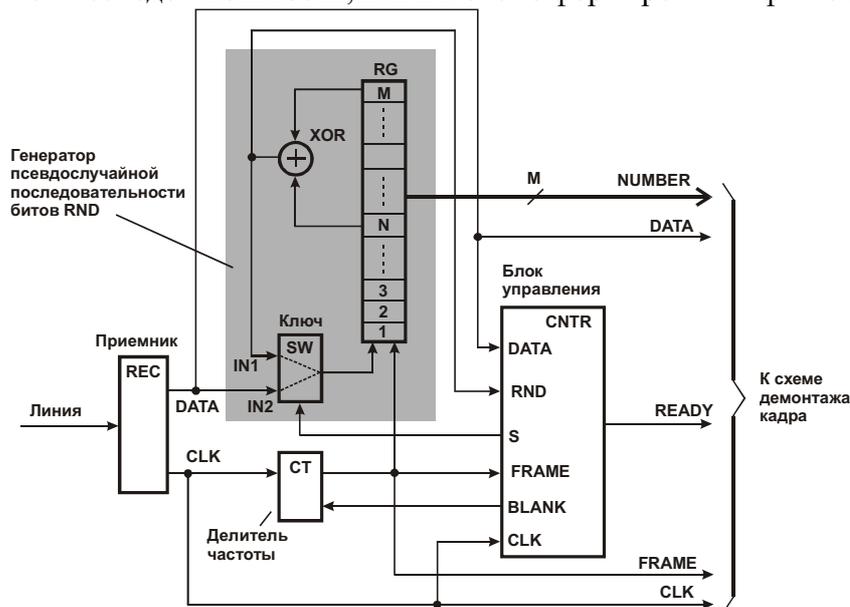


рис. 7.17. Схема распознавания межкадровых границ

Приемник REC усиливает принятый сигнал и выделяет из него синхросигнал CLK и биты DATA – собственно данные и вставленные в нужных местах псевдослучайные биты флагов начала кадров (см. рис. 7.15). Делитель частоты выполнен в виде счетчика по модулю Q (напомним, что Q – число битов в кадре). После установления правильной синхронизации восстановленные сигналы CLK, FRAME (признак наличия в потоке флагового бита) и RND (восстановленный псевдослучайный бит) находятся в тех же временных соотношениях, что и соответствующие сигналы в схеме формирования флаговых битов (см. временные диаграммы на рис. 7.16).

Сигнал BLANK длительностью в один период T сигнала CLK в необходимых случаях формируется блоком управления CNTR и притормаживает работу делителя

частоты на один такт. Это позволяет смещать импульс FRAME на один такт вправо относительно “неподвижной” последовательности битов DATA при поиске истинного положения флагового бита. Блок CNTR формирует также сигнал S управления электронным ключом SW. На этапе установления синхронизации ключ транслирует на выход сигнал с входа IN2. Когда синхронизация установлена, ключ передает на выход сигнал с входа IN1, т. е. переводит генератор псевдослучайной последовательности битов в автономный режим, при котором он нечувствителен к сигналам DATA. Параллельный M-разрядный код NUMBER на выходах регистра RG при правильной синхронизации отображает псевдослучайный номер кадра и с некоторой задержкой повторяет код в аналогичном регистре схемы формирования флаговых битов (см. рис. 7.16). Пока синхронизация не установлена, сигнал READY = 0; после установления синхронизации READY = 1.

Процесс установления синхронизации сопровождается серией довольно длительных экспериментов, которые проводятся приемной аппаратурой (эта аппаратура, разумеется, должна обладать достаточным “интеллектом”). Перед началом каждого эксперимента делается предположение о том, что некоторая периодически повторяющаяся позиция в потоке битов соответствует флаговой. Конечно, вероятность попадания в правильную сетку размещения флаговых битов с одной попытки мала и составляет $1/Q$. Но если “повезет”, то первый же эксперимент позволит войти в синхронизацию, т. е. правильно ориентироваться в потоке кадров, отслеживая их границы и номера. Если первый эксперимент оказался неудачным, то предполагаемая сетка размещения флаговых битов смещается блоком управления на один бит вправо. Проводится второй эксперимент и т. д. В худшем случае правильная синхронизация окажется возможной только после проведения последнего эксперимента с номером Q.

Предположим, что в начале эксперимента синхронизации нет, сигнал READY = 0, регистр RG и делитель частоты находятся в произвольных состояниях. Блок управления CNTR переводит ключ SW в режим передачи сигналов с входа IN2. При прохождении определенной позиции каждого кадра (эта позиция “подозревается” в качестве носителя флагового бита) на выходе делителя частоты формируется импульс FRAME. По его положительному фронту в первый разряд регистра RG принимается бит – кандидат на звание флагового. Одновременно с этим остальные биты этого регистра сдвигаются вверх на один разряд, бит из разряда M теряется.

По прошествии M кадров регистр RG заполняется принятыми битами, “подозреваемыми” в выполнении роли флаговых. Теперь возможны две ситуации.

1. Регистр действительно заполнен флаговыми битами группы из M соседних кадров. В этой ситуации содержимое регистра RG с некоторой задержкой повторяет код в аналогичном регистре схемы формирования флаговых битов (см. рис. 7.16). Существенно, что всякий раз в момент записи очередного бита в первый разряд регистра RG сигналы RND и DATA на входах IN1 и IN2 ключа SW совпадают. Это происходит благодаря тому, что в передающей и приемной аппаратуре применены одинаковые генераторы псевдослучайных последовательностей битов. Поэтому сигнал на входе IN1 формируется по тем же правилам, которым следует сигнал на входе IN2. Иными словами, сигнал RND можно рассматривать как результат предсказания значения очередного флагового бита, поступившего из линии.

Блок управления следит за совпадением предсказанных и фактически принятых флаговых битов. Если предсказания постоянно оправдываются при прохождении, например, 100 кадров, то предполагается, что синхронизация достигнута, сигнал READY устанавливается в состояние лог. 1. Блок управления переводит ключ SW в режим передачи сигнала с верхнего входа (IN1), так что генератор псевдослучайной последовательности битов переходит к автономной работе и не реагирует на ошибки, которые могут появляться в сигнале DATA. Это создает некоторую разумную инерционность

системы синхронизации и предотвращает выход и повторное вхождение в синхронизм при допустимом уровне ошибок в линии.

Теперь для дальнейшей обработки кадра имеется вся необходимая информация. Обработка, прежде всего, заключается в демонтаже кадра: вычеркивании флагового бита, определении номера кадра в группе, проверке и вычеркивании контрольных сумм и т. д.

При правильной синхронизации блок управления CNTR продолжает сопоставлять предсказанные и фактически принятые флаговые биты. Интенсивность несовпадений битов в этой ситуации не должна превышать некоторую заданную величину. Например, допустимым может считаться одно несовпадение на 100 тыс. кадров, в противном случае качество системы передачи данных расценивается как низкое. При резком повышении интенсивности несовпадений битов синхронизация считается потерянной, сигнал READY переводится в состояние лог. 0 и блок управления начинает новую серию экспериментов по восстановлению синхронизации.

2. *Регистр заполнен не флаговыми битами.* Эта ситуация приводит к множественным несовпадениям предсказанных и фактически полученных битов, причем вероятность формирования непрерывной последовательности ложных случайных совпадений быстро уменьшается со временем и с ней можно не считаться. Сигнал READY остается в нулевом состоянии, эксперимент признается неудачным. Блок управления CNTR формирует сигнал BLANK, который приостанавливает работу делителя частоты на один такт. После этого начинается новый эксперимент, при котором флаговые биты ищутся в соседних (сдвинутых вправо на один такт) позициях кадров и т. д.

Подводя итоги материалам, рассмотренным в п. 7.2, отметим следующее.

1. Длину флага можно сократить до одного бита, но для его распознавания нужен достаточно высокий “интеллект” приемной аппаратуры.

2. Использование псевдослучайных последовательностей флаговых битов позволяет не только распознавать границы кадров, но и вести их относительную нумерацию в пределах выделенных групп. Нумерация позволяет приемнику классифицировать принимаемые кадры по их номерам в соответствии с принятым протоколом обмена. Например, приемник может отличать “обычные” кадры от служебных, а среди них различать кадры различных типов. В многоканальной системе передачи данных (с мультиплексированием каналов) номер кадра позволяет приемнику определить, к какому каналу относятся содержащиеся в кадре данные, и т. д. Пример системы передачи данных, использующей псевдослучайные флаговые биты, рассмотрен в п. 7.9.

7.3. Использование раздробленного флага начала кадра

Вновь вернемся к вопросу синхронизации приемника с передатчиком. Напомним, что поток передаваемых по линии связи битов состоит из структурных единиц (например кадров). Идея построения “обычного” кадра поясняется рис. 7.18. Кадр представляет собой группу битов с границами p и q . Флаг размещен в начале кадра, представляет собой уникальным кодом и имеет фиксированную длину; его границы – a и b . За флагом следуют слова $b - c$, $c - d$, $d - e$ и т. д.

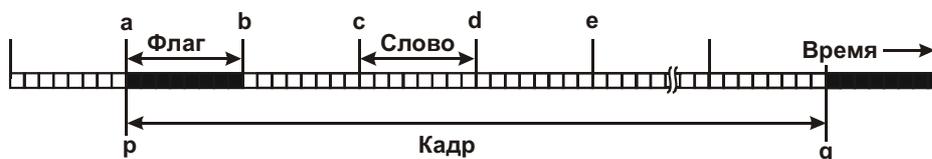


рис. 7.18. Пример структуры кадра.

Как следует из примера, рассмотренного в п. 7.2, длину флага можно сократить до одного бита, при этом битстаффинг не нужен. Здесь мы приведем родственное, но более общее решение, в котором многоразрядный флаг существует, но в виде разобренных битов. Такое решение позволяет быстро устанавливать синхронизацию между устройствами и сохранять ориентиры (биты флага) при частичном повреждении кадра из-за помех в линии.

7.3.1. Применение неуникального флагового кода

Рассмотрим предлагаемую структуру кадра (рис. 7.19) [20].

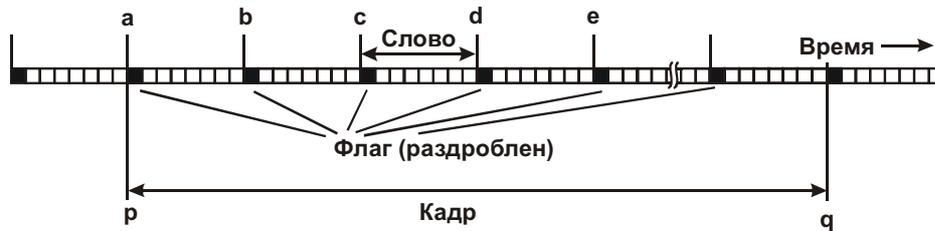


Рис. 7.19. Структура кадра с раздробленным на отдельные биты флагом

В отличие от традиционной структуры кадра (рис. 7.18), в данном случае флаг фиксированной длины раздроблен на отдельные биты. Эти биты дополнительно служат метками начала слов кадра. В то же время совокупность флаговых битов позволяет достаточно надежно распознать кадр как целое. Длина слова фиксирована, битстаффинг не нужен.

Приемник распознаёт флаг с использованием вероятностных оценок. При анализе входного потока битов он выявляет в нем устойчивые закономерности. Проще говоря, приемник как бы просматривает поток, показанный на рис. 7.19, сквозь непрозрачную маску, в которой вырезаны отверстия, соответствующие черным квадратикам на рисунке. Маска может накрывать, например, 10 кадров, так что при восьмиразрядном флаге в прорези маски попадает цепочка из $10 \times 8 = 80$ битов.

Шаг за шагом перемещая маску вдоль исследуемого потока, не позже чем через семь шагов увидим сквозь ее отверстия правильную флаговую комбинацию битов, повторенную 10 раз, что с высокой вероятностью подтверждает факт обнаружения исходной разметки кадров. (Как видим, число шагов поиска невелико, что способствует быстрому установлению синхронизации приемника с передатчиком.) В отсутствие ошибок передачи вероятность ложного обнаружения искомой цепочки из 80 битов в случайном потоке данных составляет 2^{-80} и уменьшается по мере дальнейшего перемещения маски. Это справедливо при использовании скремблера данных со стороны передатчика (см. п. 8.4). (Напомним, что скремблер передатчика преобразует данные таким образом, что они выглядят как случайные битовые последовательности с равновероятным появлением лог. 0 и лог. 1. Дескремблер приемника выполняет обратное преобразование.)

7.3.2. Построение кросс-корреляционной матрицы для распознавания раздробленного флага

Рассмотрим процесс распознавания границ кадров с демонстрацией последовательности поиска на простом примере.

Предположим, что передатчик формирует непрерывную последовательность кадров со структурой, показанной в верхней части рис. 7.20.

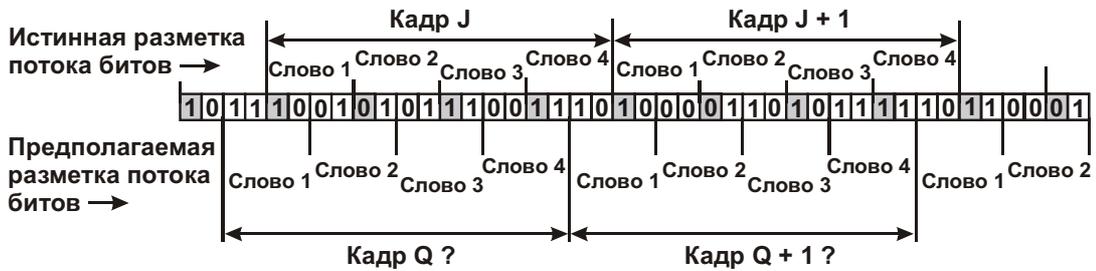


рис. 7.20. Упрощенный пример разметки потока битов на кадры и слова. Флаговые биты (рассредоточенные коды 1011_2) выделены серым фоном

Задача приемника состоит в том, чтобы путем набора и анализа статистической информации о входном потоке данных выявить границы кадров. При этом заранее известно, что: 1) кадр состоит из четырех 4-разрядных слов; 2) флаг представлен кодом 1011_2 , причем первым передается крайний левый бит.

Процесс поиска начинается с того, что приемник выдвигает предположение о том, что разметка потока битов уже известна и соответствует приведенной в нижней части рис. 7.20. Эта разметка начинается из произвольно выбранной точки. Как видно из рисунка, в данном примере имеется сдвиг на два бита влево относительно истинного положения границ кадров. Тем не менее приемник пока оперирует кадрами Q, Q + 1, Q + 2 и т. д. Знаки вопросов на рисунке отражают тот факт, что синхронизация еще не достигнута.

Кадр содержит 16 бит. Для удобства анализа он представлен матрицей размерностью 4×4 , как показано в верхней части рис. 7.21.

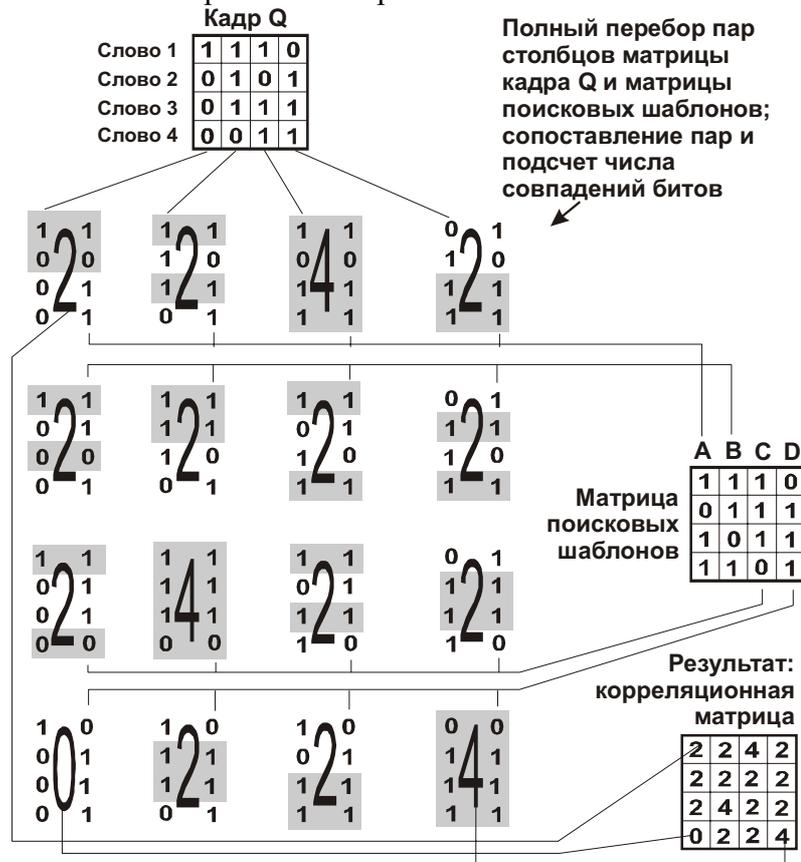


рис. 7.21. Схема вычисления корреляционной матрицы кадра Q. Серым фоном и укрупненными цифрами выделены совпадения и их число

Проведем мысленный эксперимент. Предположим, что кадр Q занял правильное положение, совпадающее с положением кадра J . Тогда искомая флаговая комбинация битов 1011 в матрице кадра Q разместится на своем месте – в первом (крайнем левом) столбце. При смещении кадра Q относительно кадра J влево на одну битовую позицию флаговая комбинация битов 1011 сместится во второй столбец матрицы. Дальнейшее смещение на одну позицию приводит к ситуации, отображенной на рис. 7.20 и рис. 7.21, когда искомый столбец матрицы – третий. Еще одно смещение – и флаг перемещается в четвертый столбец.

При следующем перемещении кадра Q влево на одну битовую позицию флаг вновь оказывается в первом столбце матрицы, но он циклически сдвинут на один разряд вправо и теперь имеет вид 1101. Продолжая серию перемещений кадра можно убедиться, что флаг может также приобретать формы 1110 и 0111. Таким образом, с учетом неопределенности выбора точки отсчета, флаговые (статистически устойчивые) комбинации битов в “системе координат” приемника могут быть такими: 1011, 1101, 1110 или 0111. Эти равноправные комбинации битов для опознания флага представлены на рис. 7.21 матрицей поисковых шаблонов размерностью 4×4 . В этой матрице столбец A соответствует истинному флагу, а остальные столбцы (B, C, D) – его циклически сдвинутым копиям.

Итак, имеются две матрицы: первая отображает содержимое кадра Q , вторая представляет собой набор из четырех шаблонов для поиска.

Задача состоит в том чтобы оценить, какой шаблон (A, B, C или D) наиболее подходит кадру. Получив самый подходящий шаблон и зная к какому именно столбцу он тяготеет, можно после аналогичного анализа ряда кадров откорректировать принятую разметку потока битов, т. е. выровнять кадры Q с кадрами J .

Задача “примерки” шаблонов решается методом простого перебора вариантов с регистрацией числа благоприятных исходов (совпадений кодов) в корреляционной матрице.

Начнем перебор. Выберем из матрицы поисковых шаблонов столбец A и поочередно сопоставим с ним все столбцы матрицы кадра Q . Первый (левый) столбец матрицы кадра Q (код 1000) частично совпадает со столбцом A (код 1011), число совпадений битов равно двум. Зафиксируем этот результат в верхней строке и первом (левом) столбце корреляционной матрицы.

Второй столбец матрицы кадра Q (код 1110) также частично совпадает со столбцом A (код 1011), число совпадений битов тоже равно двум. Зафиксируем этот результат в верхней строке и втором столбце корреляционной матрицы.

Продолжая этот процесс, заполним все позиции корреляционной матрицы. Можно ли сделать какие-то выводы? Пока лишь частичные, так как объем обработанной информации недостаточен для окончательной оценки ситуации.

Действительно, корреляционная матрица не содержит ярко выраженного элемента с максимальным численным значением. Имеем размытый максимум, представленный тремя элементами с численным значением, равным 4. Три шаблона из четырех оказались перспективными. Первый столбец матрицы кадра Q представляется безнадежно непохожим ни на один из шаблонов.

Чтобы получить больше информации, проанализируем подобным образом кадры $Q + 1$ и $Q + 2$ (рис. 7.22).

Как видим, корреляционная матрица кадра $Q + 2$ также имеет размытый максимум. Просуммируем соответствующие элементы трех корреляционных матриц и поместим результаты в соответствующие позиции кросс-корреляционной матрицы, отражающей обобщенный взгляд на три кадра.

Здесь вырисовывается уже более реалистичная картина опознания. Появилось большее число градаций переменных, сформировался максимум, равный 12 (отмечен

серым фоном). По трем кадрам, конечно, еще трудно судить о надежности результата, но напомним, что это всего лишь упрощенный пример.



рис. 7.22. Схема вычисления кросс-корреляционной матрицы группы кадров Q , $Q + 1$, $Q + 2$

Максимальное число (12) находится на пересечении первой строки и третьего столбца. Возвращаясь к схеме формирования корреляционной матрицы (см. рис. 7.21) можно утверждать, что наиболее подходящий эталон равен 1011, а наиболее подозрительным в переносе флага является третий столбец матрицы Q . Одновременно этим двум условиям удовлетворяет единственный вариант смещения кадра Q относительно кадра J , а именно, показанный на рис. 7.20. Остается сдвинуть кадр Q (и все последующие) на две битовые позиции вправо, после чего выявляются истинные границы кадров.

Подведем некоторые итоги. В рассмотренных структурах потоков данных (п. 7.1 – 7.3) флаг присутствует в явном виде и задается одним битом или раздробленной группой битов. Однобитовый флаг позволяет распознавать межкадровые границы и даже нумеровать кадры. Флаг, представленный раздробленной группой битов, позволяет уменьшить время вхождения приемника в синхронизацию с передатчиком. Кроме того, отдельные биты раздробленного флага служат признаками начала слов данных кадра. Это позволяет приемнику более уверенно ориентироваться в потоке данных в условиях повышенного уровня помех в канале связи и локализовать вероятные ошибки на уровне слов, которые примыкают к искаженным флаговым битам. Это может оказаться полезным в некоторых применениях (например, при передаче речи) когда частично поврежденный кадр можно использовать, отбросив его испорченный фрагмент, вместо того чтобы отбрасывать полный кадр.

7.4. Распознавание ячеек АТМ в битовом и байтовом потоках данных

7.4.1. Структура ячейки АТМ

Технология АТМ (Asynchronous Transfer Mode – асинхронный режим передачи) предусматривает размещение данных внутри 53-байтовых ячеек (рис. 7.23). Как показано на рисунке, ячейки следуют по линии непрерывным потоком, хотя возможны и иные варианты их транспортирования (об этом – позже).



рис. 7.23. Непосредственная передача ячеек ATM по линии связи

Ячейка состоит из заголовка и поля данных (соответственно 5 и 48 байтов). Заголовок содержит четыре байта для размещения управляющей информации (28 битов – номер логического канала плюс четыре флаговых бита) и один контрольный байт CRC (Cyclical Redundancy Check – циклический избыточный код).

Байт CRC вычисляется передатчиком и представляет собой контрольную сумму четырех предыдущих байтов. Приемник после получения ячейки проводит аналогичные вычисления и сопоставляет свой результат с имеющимся в ячейке кодом CRC. В отсутствие ошибок коды совпадают. Но как приемник определяет границы ячейки, ведь она не содержит флагов начала или конца?

7.4.2. Использование кода CRC в процессе распознавания границ ячеек

В общем виде методика поиска границ ячеек состоит в следующем. На начальном этапе поиска приемник просматривает поступающий поток ячеек (см. рис. 7.23) сквозь “непрозрачную маску с вырезанной в ней щелью”, сквозь которую видны 40 битов данных. Другими словами, перед приемником проходит “бегущая строка” длиной 5 байтов. В каждом такте строка смещается на один бит.

Приемник в каждом такте проверяет содержимое бегущей строки на предмет обнаружения в ней заголовка ячейки. Если соответствующий байт (самый новый в бегущей строке) представляет собой контрольную сумму (CRC) полученных ранее четырех байтов, то делается предположение о том, что заголовок найден. Если это предположение оправдано, то в дальнейшем моменты обнаружения заголовков будут следовать с шагом, равным длине ячейки. Если совпадение было случайным (его вероятность равна $1/256$), то оно не найдет подтверждения при последующей проверке событий на периодичность.

Как следует из рис. 7.23, истинный заголовок в наихудшем случае попадет в поле зрения приемника через $53 \times 8 - 1 = 423$ такта после начала поиска. Это соответствует ситуации, при которой поиск начат как раз в тот момент, когда предыдущий заголовок только что сместился на один бит и уже не полностью попадает в область бегущей строки.

После выделения цепи правильных совпадений на фоне случайных приемник точно знает положение границ ячеек. Синхронизация с передатчиком достигнута. Но описанный анализ потока битов продолжается, и каждый новый заголовок проверяется на правильность. Если в заголовке обнаружена корректируемая ошибка, то она исправляется (код CRC позволяет вычислить положение искаженного бита). Если ошибка некорректируемая, то ячейка уничтожается. Последовательность заголовков с некорректируемыми ошибками соответствует потере синхронизации, поэтому приемник возвращается к начальному этапу поиска границ ячеек.

Далее будут рассмотрены некоторые подробности аппаратного воплощения данной методики поиска.

7.4.3. Схема “деления” двоичных чисел

Код CRC удобно называть контрольной суммой, характеризующей некоторый массив данных. Если говорить более точно, то это – контрольный остаток от “деления” некоторого двоичного числа, построенного на основе массива, на двоичное число, выбранное в качестве делителя. Слово “деления” не напрасно взято в кавычки, так как соответствующая операция, как будет показано, выполняется довольно своеобразно (“лещенкой”, но вместо вычитания используется поразрядное логическое суммирование по модулю два). Результат этой операции в общем случае не совпадает с результатом обычного деления.

Применительно к ячейкам АТМ делитель представлен двоичным числом 100000111. Это число соответствует так называемому образующему полиному $X^8 + X^2 + X + 1$ (четыре его члена можно сопоставить с четырьмя единичными битами делителя). Полином, в свою очередь, имеет прямое отношение к структуре обратных связей в схеме “деления”, приведенной на рис. 7.24.

Схема содержит кольцевой сдвиговый регистр из восьми D-триггеров Т и три логических элемента Исключающее ИЛИ. На вход синхронизации поступает последовательность тактовых импульсов. В каждом такте положительный фронт импульса подтверждает истинность очередного бита данных D на входе регистра.

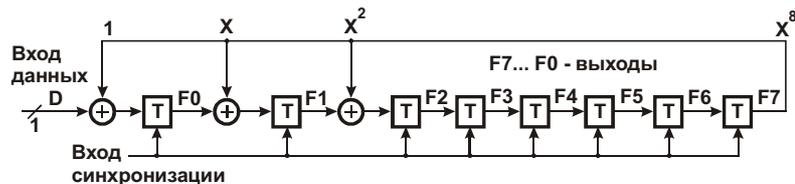


Рис. 7.24. Схема формирования (или проверки) контрольной суммы CRC

Состояние $F7(n+1) \dots F0(n+1)$ регистра в такте $n+1$ определяется его состоянием $F7(n) \dots F0(n)$ в предыдущем такте n , а также новым значением входного бита данных $D(n+1)$ и может быть выражено соотношениями (7.1):

$$\begin{aligned}
 F7(n+1) &= F6(n) \\
 F6(n+1) &= F5(n) \\
 F5(n+1) &= F4(n) \\
 F4(n+1) &= F3(n) \\
 F3(n+1) &= F2(n) \\
 F2(n+1) &= F1(n) \oplus F7(n) \\
 F1(n+1) &= F0(n) \oplus F7(n) \\
 F0(n+1) &= D(n+1) \oplus F7(n)
 \end{aligned}
 \tag{7.1}$$

Чтобы установить взаимосвязь процесса “деления” с последовательностью состояний схемы (см. рис. 7.24), рассмотрим примеры формирования кода CRC и его использования для проверки правильности принятых данных.

7.4.4. Формирование заголовка ячейки передатчиком

На начальном этапе формирования кода CRC передатчик дополняет исходные четыре байта заголовка ячейки пятым байтом 000...0, как показано на рис. 7.25. Это эквивалентно умножению исходного 32-разрядного числа на 2^8 . Полученное 40-разрядное число используется в качестве делимого; делитель равен 100000111.

Процесс преобразования 40-разрядного числа, как отмечалось, напоминает обычное деление и так же может выполняться “лесенкой”. Нули в старших разрядах делимого игнорируются. Копия делителя “пододвигается” под группу разрядов делимого, содержащую в старшем разряде лог. 1, независимо от значений последующих разрядов. Полученная пара кодов поразрядно суммируется по модулю два ($0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $1 \oplus 0 = 1$; $1 \oplus 1 = 0$).

Результат дополняется справа новыми битами делимого, которые для этого “сносятся” вниз таким образом, чтобы общее число битов нового фрагмента, подлежащего обработке, соответствовало разрядности делителя. Появление нескольких нулей в старших разрядах текущего результата, как и при обычном делении, приводит к ускоренному сдвигу очередной ступени “лесенки” вправо.

Целая часть частного не используется, остаток представляет собой байт CRC. Он “прикладывается” передатчиком к четырем исходным байтам (взамен ранее “приложенного” байта 000...0) и пересылается в линию. Первым передается старший разряд байта 1, последним – младший разряд байта CRC. В таком же порядке эта группа из пяти байтов (заголовок ячейки) поступает в приемник.

Чтобы сопоставить описанный процесс получения кода CRC с процессом “прокрутки” кода через кольцевой сдвиговый регистр (см. рис. 7.24), рассмотрим ряд состояний этого регистра (рис. 7.26, левая часть).

В исходном состоянии в регистре записан нулевой код. В первом такте ($n = 1$) по положительному фронту синхроимпульса в левый разряд регистра принимается старший бит 40-разрядного числа (делимого). Поскольку в нашем примере этот бит равен нулю, состояние регистра не изменяется. Во втором, третьем и четвертом тактах при вводе последующих нулевых битов делимого состояние регистра остается нулевым. В пятом такте старший значащий разряд (лог. 1) делимого заносится в регистр, и в последующих тактах наблюдается продвижение группы единиц в направлении выхода F7.

В двенадцатом такте лидирующая лог. 1 достигает выхода F7. Поэтому логические элементы Иключающее ИЛИ во входных каскадах регистра настраиваются на режим инвертирования соответствующих сигналов, распространяющихся слева направо. Инвертирование “проявляется” в тринадцатом такте, в результате в регистре фиксируется код 01001111. В этом коде старший разряд находится справа, что создает некоторые неудобства при его рассмотрении. Перепишем код в нормальном виде (старший разряд – слева): 11110010.

Этот код совпадает с полученным на первой ступени “лесенки деления”, а именно, с кодом, расположенным под первой горизонтальной чертой этой ступени. Продолжая сопоставление, можно заметить полное соответствие набора состояний регистра правилам “деления лесенкой”. Так, число ступеней равно одиннадцати. Столько же раз логические элементы Иключающее ИЛИ во входных каскадах регистра настраиваются на режим инвертирования соответствующих сигналов, распространяющихся слева направо. Столько же раз сигнал F7 равен лог. 1 (этот сигнал отображает последовательный код целой части частного, которая отбрасывается).

В сороковом такте ($n = 40$) в регистре фиксируется код CRC.

7.4.5. Проверка правильности заголовка ячейки приемником

Итак, рассматривается ситуация, при которой в приемник поступают пять байтов, из которых пятый – код CRC, построенный на основе обработки первого – четвертого байтов. Приемник решает такую задачу: действительно ли пятый байт представляет собой правильный код CRC, т. е. соответствует ли он контрольной сумме предшествующих четырех байтов?

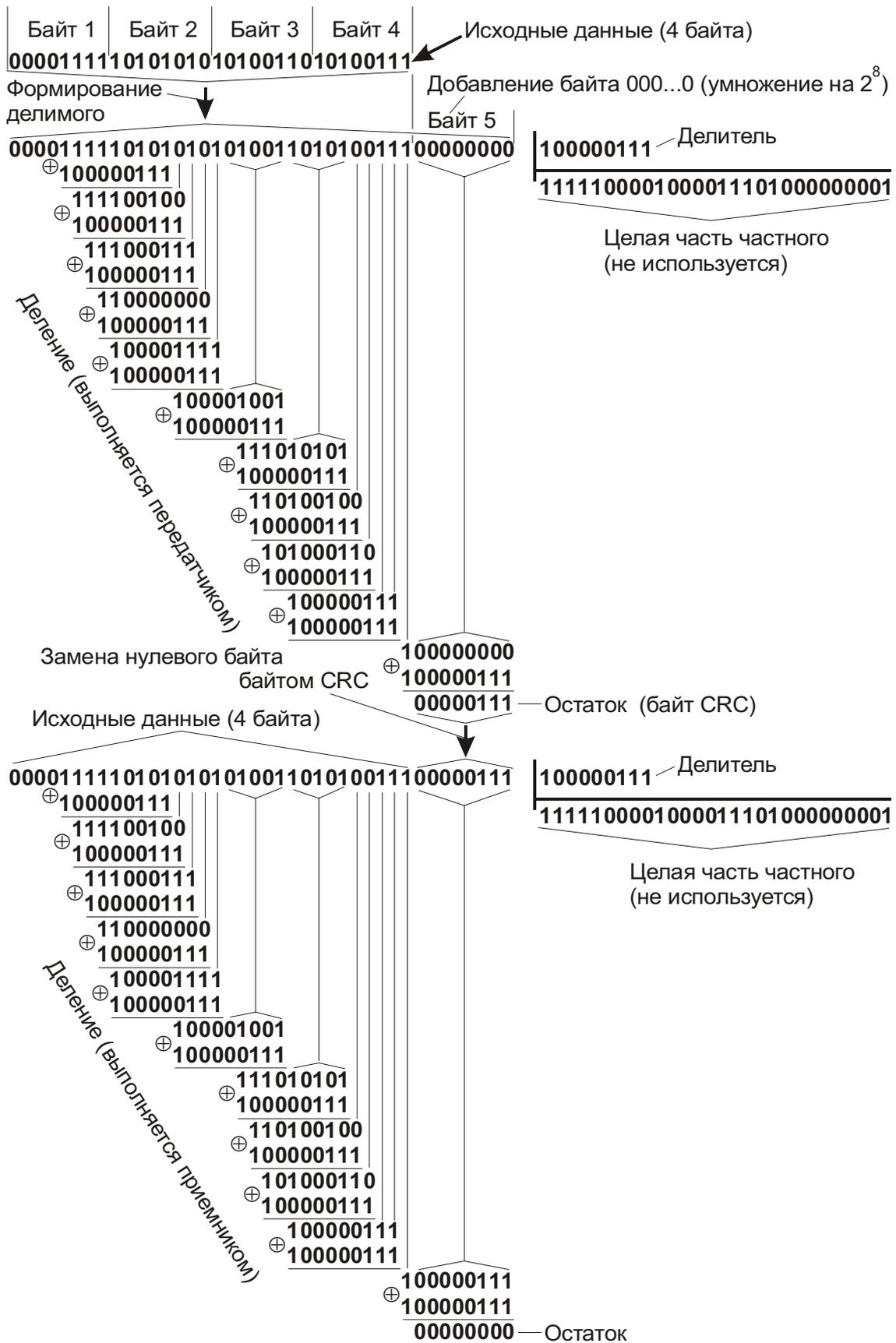


рис. 7.25. Формирование кода CRC передатчиком (верхняя часть рисунка) и проверка полученных данных приемником (нижняя часть)

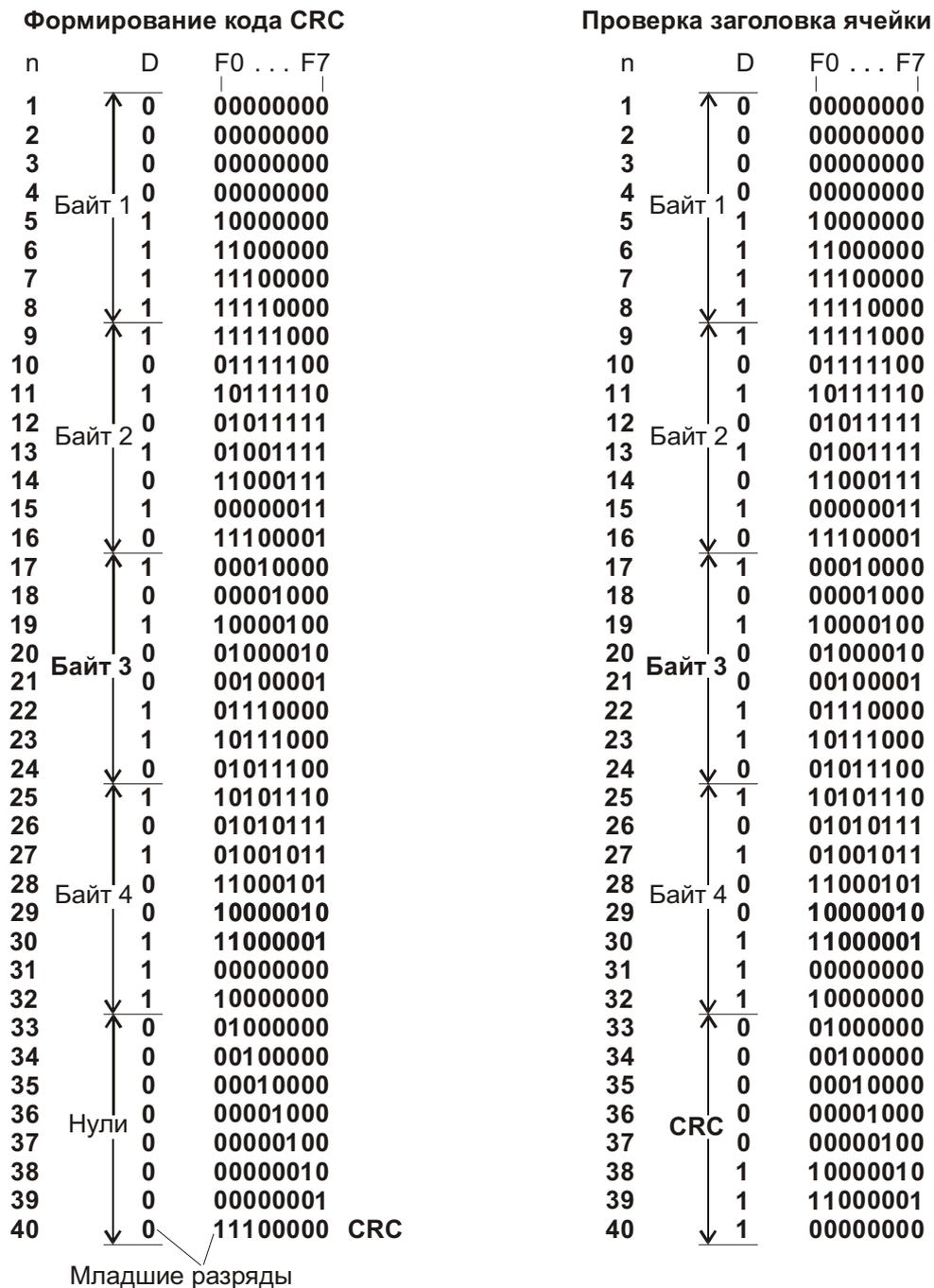


рис. 7.26. Формирование кода CRC передатчиком (левая часть рисунка) и проверка заголовка ячейки приемником (правая часть) с использованием кольцевого сдвигового регистра, показанного на рис. 7.24; n – номер такта

Вернемся к рис. 7.25 и рассмотрим его нижнюю часть. Делимое содержит в младшем байте код 00000111, который соответствует коду CRC, вычисленному передатчиком. Деление “лесенкой” на десяти ступенях дает те же результаты, что и ранее, но на одиннадцатой ступени получаем нулевой остаток. Это свидетельствует о том, что контрольная сумма правильная (что и требовалось проверить).

В правой части рис. 7.26 показан ряд состояний кольцевого сдвигового регистра (см. рис. 7.24) приемника при проверке правильности кода CRC. Отличия от группы состояний аналогичного регистра передатчика наблюдаются в тактах 38 – 40. В част-

ности, в контрольном, сороковом такте имеем нулевое содержимое регистра, что подтверждает правильность приема заголовка.

Отметим, что нулевые коды получены также в тактах 1 – 4, 31; но в нашем примере заранее известно, что эти такты промежуточные, поэтому моменты получения промежуточных нулевых кодов не являются для нас значимыми событиями. Иное дело – в “реальной жизни”, когда такие события не должны оставаться незамеченными.

7.4.6. Поиск заголовка в непрерывном битовом потоке данных

Рассмотренный ранее способ проверки правильности кода CRC с помощью схемы, приведенной на рис. 7.24, применим лишь в идеальной “статической” ситуации, когда заранее известно, что начальное состояние сдвигового регистра – нулевое, а в первом такте ($n = 1$) в регистр заносится старший бит заголовка.

Реальная ситуация динамична, в том смысле, что в поле зрения приемника находится бегущая строка из 40 битов. В каждом такте содержимое строки смещается на один бит. Поэтому в каждом такте нужно успеть проверить правильность пятого байта – предположительно кода CRC. Если применить рассмотренную ранее методику, то нужно в начале каждого такта установить кольцевой сдвиговый регистр в нуль, а оставшийся тактовый интервал разбить на 40 микротактов для выполнения процедуры, показанной в правой части рис. 7.26. Но такое решение слишком “прямолинейно”, чтобы быть эффективным.

Рассмотрим решение, предложенное в [23] (рис. 7.27). Здесь проверка очередного 40-разрядного кода выполняется за один такт в конвейерном режиме. Новый бит, видимый в бегущую строку, обычным образом учитывается в новой контрольной сумме; но одновременно с этим новая контрольная сумма корректируется из-за того, что один бит “вытолкнут” за пределы бегущей строки, и схема должна уничтожить его былой вклад в контрольную сумму. Теперь всё по порядку.

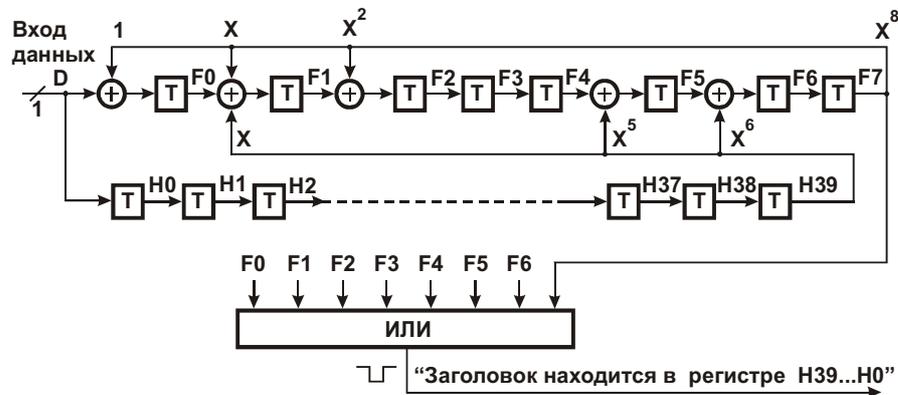


Рис. 7.27. Схема распознавания ячейки в непрерывном потоке битов. Цепи синхронизации триггеров не показаны

Сравнивая схемы, показанные на рис. 7.24 и рис. 7.27, можно заметить, что последняя представляет собой расширение первой. Кольцевой сдвиговый регистр содержит дополнительные цепи коррекции результата. Введен 40-разрядный сдвиговый регистр для хранения бегущей строки, в которой ожидается появление заголовка ячейки (точнее, кода с правильной контрольной суммой, претендующего на то, чтобы именоваться заголовком). Элемент ИЛИ регистрирует появление нулевого кода в кольцевом сдвиговом регистре. Все D-триггеры Т синхронизируются общим тактовым сигналом.

Состояние $F7(n+1) \dots F0(n+1)$ кольцевого сдвигового регистра в такте $n+1$ определяется состоянием $F7(n) \dots F0(n)$ этого регистра в предыдущем такте n , со-

стоянием разряда $H_{39}(n)$ в такте n , а также новым значением входного бита данных $D(n+1)$ и может быть выражено следующими соотношениями:

$$\begin{aligned}
 F_7(n+1) &= F_6(n) \\
 F_6(n+1) &= F_5(n) \oplus H_{39}(n) \\
 F_5(n+1) &= F_4(n) \oplus H_{39}(n) \\
 F_4(n+1) &= F_3(n) \\
 F_3(n+1) &= F_2(n) \\
 F_2(n+1) &= F_1(n) \oplus F_7(n) \\
 F_1(n+1) &= F_0(n) \oplus F_7(n) \oplus H_{39}(n) \\
 F_0(n+1) &= D(n+1) \oplus F_7(n)
 \end{aligned}
 \tag{7.2}$$

Процесс распознавания заголовка ячейки поясняется рис. 7.28.

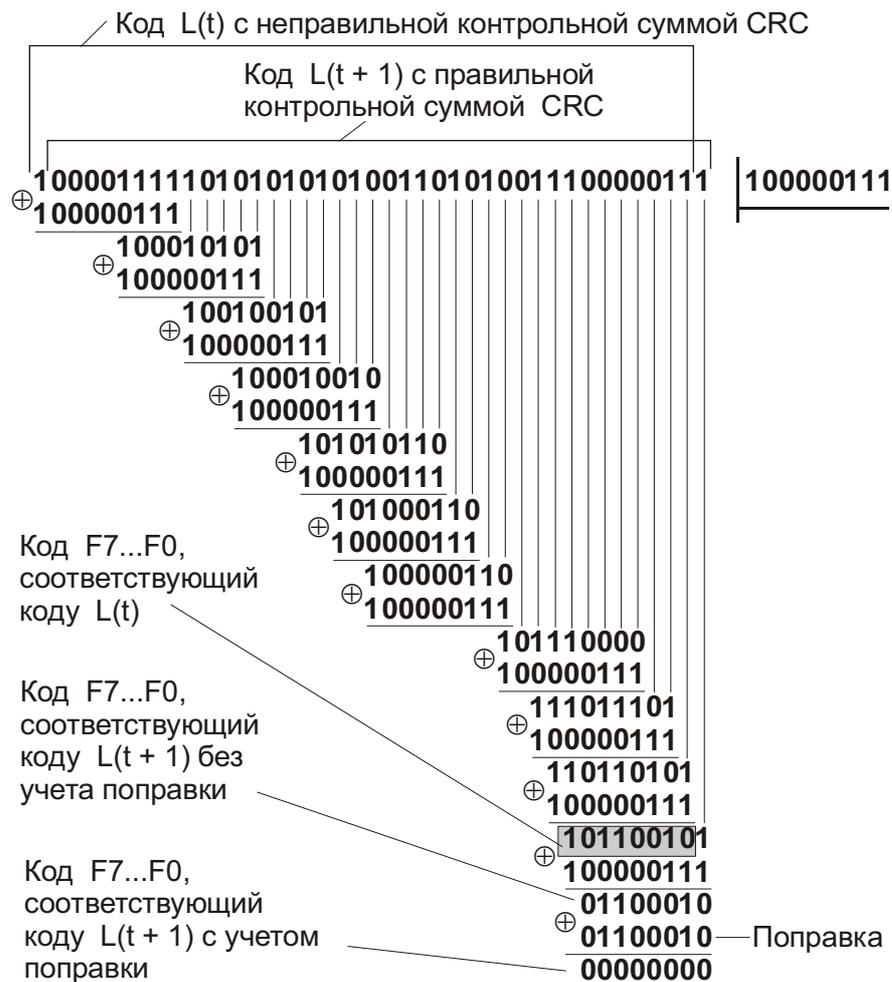


Рис. 7.28. Конвейерная обработка бегущей строки на примере анализа перекрывающихся кодов $L(t)$ и $L(t+1)$ с неправильной и правильной контрольными суммами

Предположим, что в некоторый момент t бегущая строка (регистр $H_{39} - H_0$) содержит код $L(t)$. Примем пока “на веру”, что в этот же момент в кольцевом сдвиговом регистре присутствует результат 40-тактной проверки кода $L(t)$ по приведенной ранее методике. Этот результат (код 10110010, выделенный на рисунке серым фоном) отличается от нулевого, что находится в согласии с тем, что пятый байт кода $L(t)$ не является контрольной суммой (CRC) четырех предыдущих байтов.

В момент $t+1$ в бегущей строке фиксируется код $L(t+1)$. Этот код мы уже рассматривали в предыдущем примере. Он содержит правильную контрольную сумму

(CRC), равную 00000111. Из рисунка следует, что простой ввод в кольцевой сдвиговый регистр очередного бита, т. е. простой (без поправки) переход от кода $L(t)$ к коду $L(t+1)$ приводит к результату, равному 01100010. Этот результат нельзя считать правильным, так как при правильной контрольной сумме результат должен быть нулевым. Таким образом, если не учитывать поправку, то операцию можно выполнить быстро, за один такт, но, к сожалению, неправильно.

В чем же тут дело? Проблема состоит в том, что результат, равный 01100010, получен при обработке не 40-, а 41-разрядного числа, в котором старший разряд принадлежит коду $L(t)$, а остальные – коду $L(t+1)$. Иными словами, в полученном результате неоправданно учитывается вклад в контрольную сумму исчезнувшего бита, отброшенного за пределы бегущей строки. Нужно уничтожить этот вклад. Но каков он?

Чтобы ответить на этот вопрос, отметим, что отброшенный разряд отображает число $2^{40} = 1000...0$ (сорок нулей после единицы). Если “разделить” (например, “лесенкой”) это число на наш делитель (100000111), то получим в остатке искомый вклад исчезнувшего бита. Он равен 01100010 (соответствующий образующий полином: $X^6 + X^5 + X$). Остается вычесть этот мешающий вклад (поправку) из результата обработки 41-разрядного числа. Как отмечалось, “деление” выполняется довольно своеобразно, и операция вычитания заменяется поразрядным суммированием чисел по модулю два.

Такое суммирование выполняется на заключительном этапе процедуры, показанной на рис. 7.28. Конечный результат совпадает с ожидаемым (нулевым). Как следует из схемы, приведенной на рис. 7.27, введение поправки совмещено во времени с вводом очередного бита данных D , поэтому проверка кода в бегущей строке выполняется за один такт. Если бит, покинувший бегущую строку, нулевой, то поправка не нужна.

Таким образом, если в такте j проверка содержимого бегущей строки выполнена правильно, то и в следующем такте $j+1$ гарантируется правильность результата (ошибками передачи пренебрегаем). Отсюда можно заключить, что схема (рис. 7.27) обеспечивает непрерывное слежение за содержимым бегущей строки, причем каждый 40-разрядный код анализируется за один такт. Конечно, для начального заполнения “конвейера” нужно потратить 40 тактов после предварительной установки всех триггеров T в нулевое состояние.

После установления надежной синхронизации приемника с передатчиком (на уровне установления границ ячеек) приемник использует код CRC для коррекции одиночных ошибок в заголовке. Это возможно, в частности, благодаря тому, что к моменту обнаружения ошибки заголовков хранится в регистре $H_{39} - H_0$, поэтому можно проинвертировать ошибочный бит, т. е. исправить его (цепи коррекции одиночных ошибок в заголовке ячейки на рисунке не показаны).

7.4.7. Поиск заголовка в непрерывном байтовом потоке данных

Размещение ячейки внутри кадра

Ячейки ATM могут передаваться либо непосредственно по линии связи, как было показано на рис. 7.23, либо в составе более крупных структурных единиц, например кадров SDH (рис. 7.29). (SDH - Synchronous Digital Hierarchy – синхронная цифровая иерархия, европейский стандарт на средства передачи данных.)

Здесь существенно то, что аппаратура обработки кадра распознаёт его начало и преобразует поток битов в поток байтов. Это позволяет снизить необходимую скорость обработки данных в восемь раз. Кадр построен так, что заранее неизвестно, где размещен блок ячеек ATM. Поэтому задача отыскания границ ячеек остается актуальной и при байтовом потоке данных.

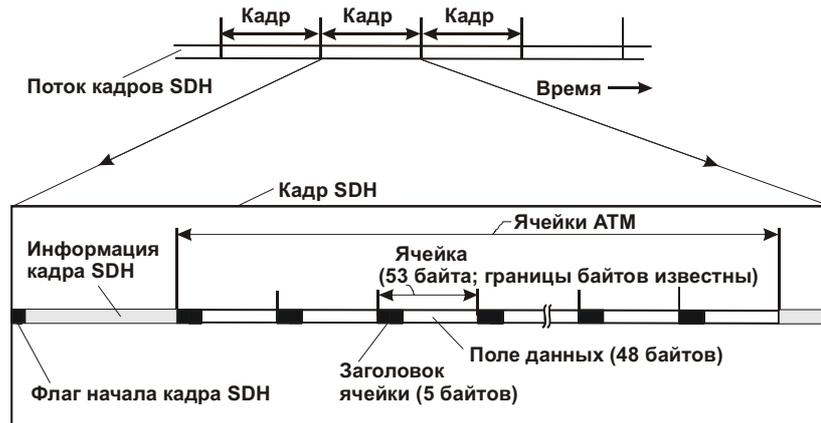


Рис. 7.29. Передача ячеек ATM по линии связи в упакованном виде. Ячейки помещены в кадры SDH

Схема распознавания заголовка ячейки в байтовом потоке данных

Решение, приведенное на рис. 7.30, представляет собой “распараллеленный” аналог рассмотренной ранее схемы (см. рис. 7.27).



Рис. 7.30. Схема распознавания заголовка ячейки в непрерывном потоке данных – байтовый вариант. Общая цепь синхронизации регистров RG1 – RG6 не показана

Группа параллельных регистров RG5 – RG1 представляет собой байтовый сдвиговый регистр для хранения содержимого бегущей строки. Регистр RG6 хранит текущий результат вычисления циклического кода. Схема содержит две последовательно включенные группы логических элементов Иключающее ИЛИ для вычисления циклического кода с учетом поправки, аналогично тому, как это делалось в предыдущей схеме, но в байтовом воплощении. Восьмивходовый логический элемент ИЛИ регистрирует появление нулевого кода в регистре RG6.

Структура других составных частей этой схемы приведена на рис. 7.31.

Схемы вычисления циклического кода и байта поправки содержат по восемь элементов Иключающее ИЛИ. Конфигурация связей в этих схемах определяется приведенными далее логическими выражениями, которые мы получим на основе рассмотренных ранее соотношений (7.2).

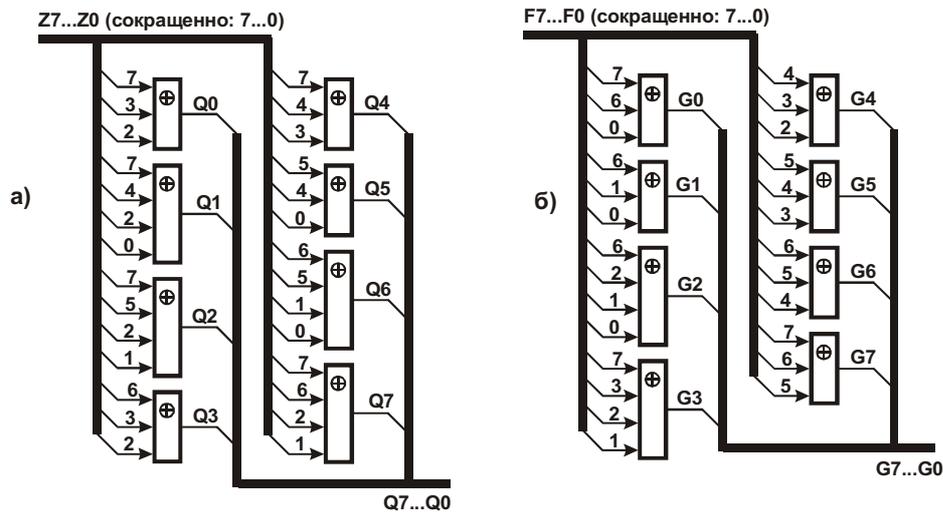


Рис. 7.31. Составные части схемы, приведенной на рис. 7.30: а – схема вычисления байта поправки; б – схема вычисления циклического кода

Логические соотношения для перехода от битового потока данных к байтовому

Напомним, что соотношения (7.2) описывают процесс вычисления циклического кода при последовательной обработке данных схемой, приведенной на рис. 7.27. Такт n в соотношениях (7.2) соответствует предыдущему состоянию схемы, такт $n + 1$ отображает текущее состояние. В этих же терминах предыдущее состояние “байтовой” схемы (рис. 7.30) характеризуется следующим набором переменных:

$$F_7(n), F_6(n), F_5(n), F_4(n), F_3(n), F_2(n), F_1(n), F_0(n); \\ Z_7(n), Z_6(n), Z_5(n), Z_4(n), Z_3(n), Z_2(n), Z_1(n), Z_0(n).$$

Текущее состояние входного байта данных D простирается от “настоящего времени” ($n + 8$) на семь тактов в “прошедшее”:

$$D(n + 8), D(n + 7), D(n + 6), D(n + 5), D(n + 4), D(n + 3), D(n + 2), D(n + 1).$$

Текущее состояние регистра RG_6 характеризуется следующим набором переменных:

$$F_7(n + 8), F_6(n + 8), F_5(n + 8), F_4(n + 8), F_3(n + 8), F_2(n + 8), F_1(n + 8), F_0(n + 8).$$

Установим зависимость текущего состояния регистра RG_6 от текущего состояния входного байта данных D и предыдущего состояния схемы, принимая во внимание следующие соотношения:

$$Z_7(n) = H_{39}(n), Z_6(n) = H_{38}(n), Z_5(n) = H_{37}(n), \dots, Z_0(n) = H_{32}(n)$$

$$H_{39}(n) = H_{38}(n - 1) = H_{37}(n - 2) = H_{36}(n - 3) = H_{35}(n - 4) = \dots$$

Первая цепь равенств принята по определению, вторая описывает работу битового сдвигового регистра.

Запишем первое соотношение из (7.2) в следующем виде:

$$F7(n + 8) = F6(n + 7). \quad (7.3)$$

С учетом второго соотношения из (7.2) равенство (7.3) можно продолжить:

$$F7(n + 8) = F6(n + 7) = F5(n + 6) \oplus H39(n + 6). \quad (7.4)$$

Продолжаем проникать во все более глубокие “археологические слои”, учитывая оставшиеся соотношения (7.2). Цепь равенств (7.4) приобретает следующий вид, при котором каждая последующая строка отображает очередной, более глубокий “слой” (некоторое исключение представляет последняя строка):

$$\begin{aligned} F7(n + 8) &= \\ &= F6(n + 7) = \\ &= F5(n + 6) \oplus H39(n + 6) = \\ &= F4(n + 5) \oplus H39(n + 5) \oplus H38(n + 5) = \\ &= F3(n + 4) \oplus H38(n + 4) \oplus H37(n + 4) = \\ &= F2(n + 3) \oplus H37(n + 3) \oplus H36(n + 3) = \\ &= F1(n + 2) \oplus F7(n + 2) \oplus H36(n + 2) \oplus H35(n + 2) = \\ &= F0(n + 1) \oplus F7(n + 1) \oplus H39(n + 1) \oplus F6(n + 1) \oplus H35(n + 1) \oplus H34(n + 1) = \\ &= D(n + 1) \oplus F7(n) \oplus F6(n) \oplus H38(n) \oplus F5(n) \oplus H39(n) \oplus H34(n) \oplus H33(n). \end{aligned}$$

После упорядочения в последней строке последовательности “слагаемых”, получим:

$$F7(n + 8) = D(n + 1) \oplus F7(n) \oplus F6(n) \oplus F5(n) \oplus H39(n) \oplus H38(n) \oplus H34(n) \oplus H33(n).$$

Аналогичные “раскопки глубин истории” можно провести и в отношении переменных $F6(n + 8)$, $F5(n + 8)$, $F4(n + 8)$, ..., $F0(n + 8)$. В результате получим полный набор функций перехода схемы (см. рис. 7.30) от предыдущего состояния к текущему:

$$\begin{aligned} F7(n+8) &= D(n + 1) \oplus F7(n) \oplus F6(n) \oplus F5(n) \oplus H39(n) \oplus H38(n) \oplus H34(n) \oplus H33(n) \\ F6(n+8) &= D(n + 2) \oplus F6(n) \oplus F5(n) \oplus F4(n) \oplus H38(n) \oplus H37(n) \oplus H33(n) \oplus H32(n) \\ F5(n+8) &= D(n + 3) \oplus F5(n) \oplus F4(n) \oplus F3(n) \oplus H37(n) \oplus H36(n) \oplus H32(n) \\ F4(n+8) &= D(n + 4) \oplus F4(n) \oplus F3(n) \oplus F2(n) \oplus H39(n) \oplus H36(n) \oplus H35(n) \\ F3(n+8) &= D(n + 5) \oplus F7(n) \oplus F3(n) \oplus F2(n) \oplus F1(n) \oplus H38(n) \oplus H35(n) \oplus H34(n) \\ F2(n+8) &= D(n + 6) \oplus F6(n) \oplus F2(n) \oplus F1(n) \oplus F0(n) \oplus H39(n) \oplus H37(n) \oplus H34(n) \oplus H33(n) \\ F1(n+8) &= D(n + 7) \oplus F6(n) \oplus F1(n) \oplus F0(n) \oplus H39(n) \oplus H36(n) \oplus H34(n) \oplus H32(n) \\ F0(n+8) &= D(n + 8) \oplus F7(n) \oplus F6(n) \oplus F0(n) \oplus H39(n) \oplus H35(n) \oplus H34(n) \end{aligned} \quad (7.5)$$

Соотношения (7.5) подтверждают правильность выбора общей структуры схемы распознавания заголовка ячейки (см. рис. 7.30). Действительно, “по большому счету” в этих соотношениях просматриваются три группы слагаемых (по модулю два):

- 1) байт входных данных $D(n + 8)$, $D(n + 7)$, ..., $D(n + 1)$;
- 2) байт, полученный на основе суммирования по модулю два определенных разрядов кода $F7(n)$, $F6(n)$, $F5(n)$, ..., $F0(n)$. Этот байт формируется на выходе схемы вычисления циклического кода;
- 3) байт, полученный на основе суммирования по модулю два определенных разрядов кода $H39(n)$, $H38(n)$, $H37(n)$, ..., $H32(n)$. Этот байт формируется на выходе схемы вычисления поправки.

Теперь можно установить соответствие между соотношениями (7.5) и схемными решениями, приведенными на рис. 7.31.

Рассмотрим, например, цепь формирования старшего разряда схемы вычисления байта поправки (см. рис. 7.31, а, правый нижний логический элемент Исключающее ИЛИ). На входы этого элемента поступают сигналы $Z7, Z6, Z2, Z1$ или, что то же самое, $N39, N38, N34, N33$. Это согласуется с группой соответствующих переменных в первой строке соотношений (7.5).

Цепь формирования младшего разряда схемы вычисления циклического кода (см. рис. 7.31, б) построена на основе трехходового логического элемента Исключающее ИЛИ, размещенного слева вверху. На входы этого элемента поступают сигналы $F7, F6$ и $F0$. Это согласуется с группой соответствующих переменных в последней строке соотношений (7.5).

7.5. Распознавание и восстановление искаженных кадров при передаче данных по радиоканалу

В системе мобильной телефонной связи (рис. 7.32) предусмотрена одновременная передача “оцифрованных” речевых и иных данных по радиоканалу. Данные, не относящиеся к передаче речи, пересылаются в паузах между словами и фразами [28]. Дальнейшее описание относится именно к таким данным.



рис. 7.32. Фрагмент системы мобильной телефонной связи

Потоки данных состоят из последовательностей кадров. Кадр первоначально формируется в памяти передающего устройства и представляет собой группу байтов, в которой первый и два последних служебные, а остальные содержат полезные данные, объединенные в сообщение (рис. 7.33).

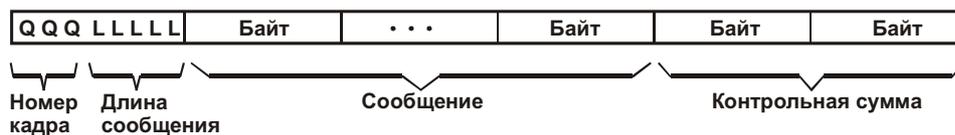


рис. 7.33. Структура кадра до его выдачи в радиоканал

Биты Q первого байта задают порядковый номер кадра, биты L определяют длину сообщения. Если все биты L нулевые, то сообщение не содержит ни одного байта. Такой “АСК-кадр” (от Acknowledge – ответ) рассматривается как подтверждение успешного приема кадра, содержащего сообщение. (Ответный кадр пересылается в противоположном направлении по отношению к кадру, несущему сообщение.) Два последних байта представляют собой контрольную сумму (16-разрядный циклический код CRC) всех предшествующих байтов кадра.

Дальнейшая подготовка кадра к выдаче в радиоканал состоит в его дроблении на отдельные асинхронные посылки (см. п. 1.2, рис. 1.5). При этом каждый байт преобразуется в две посылки, как показано на

рис. 7.34.

Преобразование состоит в следующем. Сначала байт $D1 \dots D8$ делится на два полубайта: $D1 \dots D4$ и $D5 \dots D8$. Им присваиваются четырехразрядные порядковые номера: $N1 \dots N4$ и $M1 \dots M4$. Эти номера “склеиваются” с соответствующими полубайтами, в результате образуются два байта:

$$N1 \dots N4 D1 \dots D4 \text{ и } M1 \dots M4 D5 \dots D8.$$

На заключительной стадии преобразования к обоим байтам добавляются стартовые биты ST, биты P контроля по четности (нечетности) и стоповые биты ST.

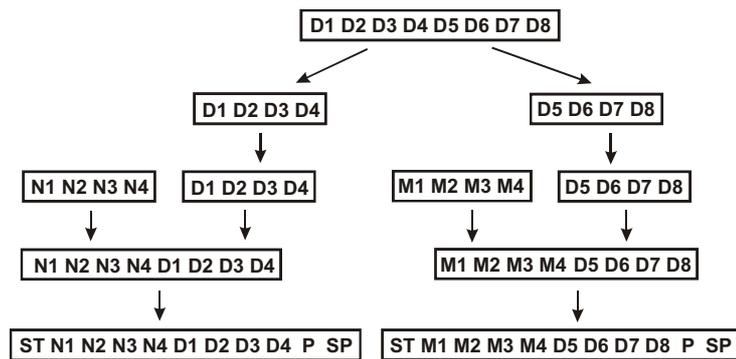


рис. 7.34. Преобразование байта в две start-стоповые посылки

В результате преобразования исходный кадр дробится на множество посылок (их число равно удвоенному числу байтов кадра). Они выдаются в радиоканал в паузах между “всплесками” передачи речевой информации. Поэтому поток данных неравномерен и может содержать как следующие друг за другом вплотную, так и разрозненные группы посылок или отдельные посылки. В процессе передачи по радиоканалу часть посылок может быть потеряна из-за возможных замираний сигнала или действия помех. Приемник может также получать ложные посылки, которые вклинились в неравномерный поток правильных. Чтобы обеспечить надежный обмен данными в условиях замираний сигнала и действия помех, предлагаются три простые, но примечательные идеи.

Идея 1. Проверка правильности порядка следования посылок слежением за динамикой изменения номеров полубайтов. Как было показано, каждая посылка содержит полубайт данных и предшествующий ему четырехразрядный номер этого полубайта. В отсутствие ошибок приемник должен регистрировать некоторую заранее заданную последовательность номеров полубайтов, например такую: ... 0, 1, 2, 3, ..., F, 0, ... (использована шестнадцатиричная форма записи чисел). Если получена последовательность ...0, 1, 2, 7, 3, 4, 5, 6, ..., то можно сделать вывод о том, что посылка с порядковым номером 7 нарушает динамику изменения номеров и явно лишняя. Поэтому приемник просто отбрасывает ее. Аналогично в последовательности ... 0, 1, 2, 4, 5, 6, ... недостает цифры 3. Это означает, что соответствующая посылка потеряна по пути к приемнику. Поэтому передача ошибочного кадра повторяется в связи с неполучением подтверждения правильного приема данных.

Идея 2. Расширение диапазона нумерации посылок при ограниченной разрядности номеров. С помощью четырехразрядного двоичного кода можно отображать числа в диапазоне $0 - F_{16}$. Поэтому на первый взгляд представляется естественной простая последовательная нумерация посылок: ... 0, 1, 2, ..., 9, A, B, C, D, E, F, 0, 1, 2, Однако такая нумерация повторяется с периодом, равным 16, а кадр может быть представлен десятками и сотнями посылок. Действительно, можно построить достаточно большой кадр, если принять условие, что длина сообщения (см. код LLLL на рис. 7.33) выражается не в отдельных байтах, а в более крупных единицах, например, в группах из четырех байтов.

Можно ли с помощью четырехразрядных кодов пронумеровать сотню или более посылок так, чтобы приемник мог уверенно определять положение любой из них относительно начала кадра? На первый взгляд, ответ на этот вопрос может быть только отрицательным. Но решение есть, и в этом мы сейчас убедимся.

Пример. Предположим, что составляющие кадр посылки пронумерованы с помощью такой последовательности номеров:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 0, 3, 6, 9, C, F, 2, 5, 8, B, E, 1, 4, 7, A, D, 0, 5, A, F, 4, 9, ..., 0, 7, E, 5, C, ...

Первые 16 посылок пронумерованы в обычном порядке, начиная с нулевого номера. Следующие 16 посылок также нумеруются, начиная с нуля, но номер наращивается с шагом, равным трем. Следующие 16 посылок нумеруются с нуля с шагом 5, следующие – с шагом 7, затем с шагами 9, 11, 13 и 15. В этом примере в качестве шагов выбраны восемь последовательно возрастающих чисел, не превышающих 15 и не имеющих общего делителя с числом 16, т. е. нечетные числа. Таким образом, получена *уникальная* последовательность из $16 \times 8 = 128$ *неуникальных* номеров. Эту последовательность можно продолжить, применив какие-либо заранее учтенные в протоколе обмена дополнительные правила ее формирования. Например, можно повторить последовательность, поменяв местами номера, размещенные в первых и последних пяти позициях каждой группы из 16 посылок и т. д. Иными словами, возможности построения такого рода уникальных последовательностей номеров практически безграничны.

Приемник просматривает полученную последовательность и проверяет ее правильность. Как уже было показано, зная вид последовательности, можно выявить и отбросить лишние номера и соответствующие посылки, если они есть, и зафиксировать пропущенные номера посылок. Так как последовательность уникальна, то можно отыскать ее начало и конец, т. е. идентифицировать первую и последнюю посылки кадра.

Отметим, что рассмотренная идея применима также и к нумерации кадров – наличие трех битов *QQQ* для указания номера кадра вовсе не означает, что соответствующий диапазон составляет только восемь номеров.

Идея 3. Усовершенствованное мажоритарное восстановление ошибочных байтов кадра. Если приемник получил все посылки, относящиеся к кадру, то он реконструирует байты в последовательности, обратной той, которая была приведена на рис. 7.34. Для этого он отбрасывает служебные биты *ST*, *P* и *SP* выбранной пары посылок, затем уничтожает их номера *N1 – N4* и *M1 – M4* и склеивает полубайты. В результате формируется первоначальный байт *D1 – D8*.

После формирования всех байтов кадра приемник проверяет его контрольную сумму (см. рис. 7.33). Если контрольная сумма правильная, то приемник посылает уведомление об этом источнику кадра. (Напомним, что в качестве уведомления используется кадр с нулевым кодом *LLLLL* длины сообщения.)

Если контрольная сумма неправильная, или в кадре отсутствуют некоторые байты, то приемник не посылает уведомление, сохраняет имеющуюся информацию о кадре и ожидает поступления копии ошибочного кадра. При поступлении копии проверяется ее контрольная сумма. Если она правильная, то посылается уведомление об этом, в противном случае приемник сопоставляет две ошибочные версии кадра и пытается собрать из них один полноценный кадр. Если это удалось сделать и контрольная сумма правильная, то посылается уведомление, иначе приемник ждет поступление новой копии и т. д.

При наличии нескольких копий кадра применяется обычный метод мажоритарного выбора, при котором используется “голосование по большинству”. Например если два одноименных байта из трех совпадают, то их значение принимается в качестве истинного. Если все три байта разные, то голосование переносится на уровень битов. Это позволяет в некоторых ситуациях, когда одиночные ошибки распределены по разным битам, восстановить правильный байт. Но если, например, из пяти одноименных байтов два совпадают между собой, но отличаются от трех других, которые, в свою очередь, не совпадают друг с другом в любых сочетаниях, то эти совпадающие байты принимаются в качестве истинных, хотя, строго говоря, они – в меньшинстве. Конечно, окончатель-

ное решение об отсутствии ошибок в кадре принимается после проверки его контрольной суммы.

7.6. Распознавание флагового кода, содержащего ошибки

В некоторых системах, например в системе мобильной связи, передача кадров часто происходит на фоне помех, которые искажают отдельные биты или группы битов. В частности, не исключена возможность появления искажений во флаговых кодах, обозначающих начало кадра. С этими искажениями нельзя не считаться, так как если ориентироваться только на получение безошибочных флаговых кодов, то из-за их нерегулярного поступления трудно установить или поддерживать ранее установленную кадровую синхронизацию. Задача распознавания флаговых кодов, в том числе, содержащих ошибки, может быть решена с помощью схемы, приведенной на рис. 7.35, а [34].

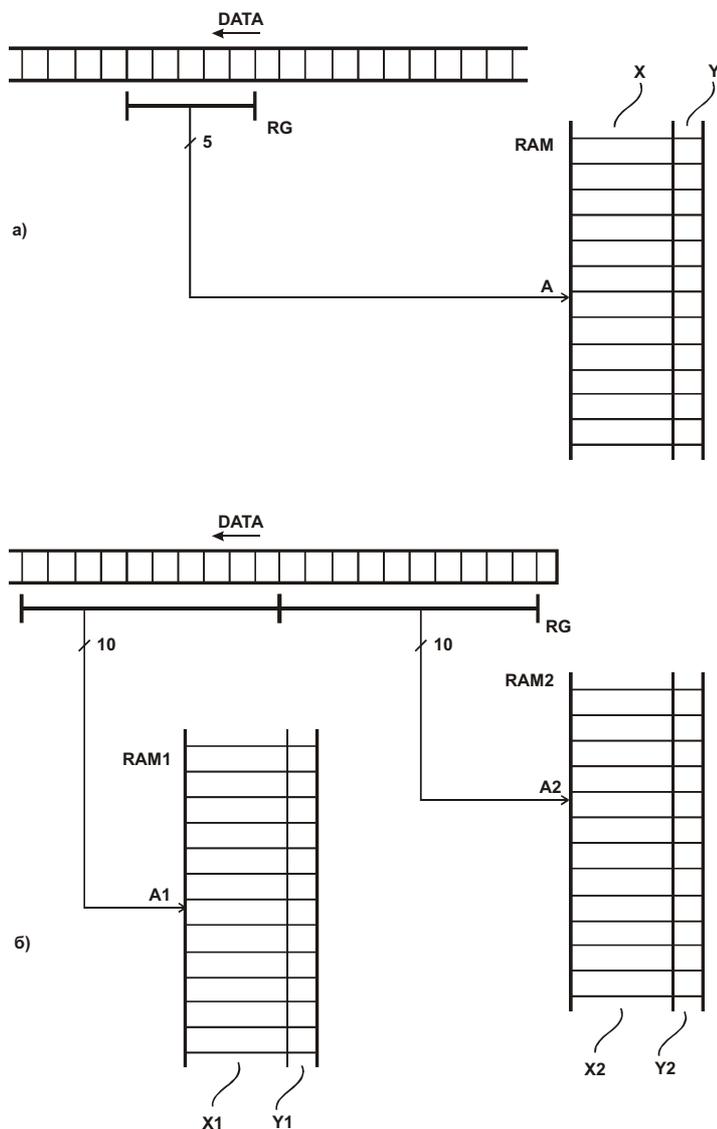


Рис. 7.35. Схема поиска флагового кода: а – с использованием одного блока памяти; б – с использованием двух блоков памяти

Входной поток данных DATA проходит через сдвиговый регистр RG. Этот процесс можно интерпретировать как просмотр движущейся последовательности битов через неподвижное пятиразрядное окно (направление движения данных показано

стрелкой). Параллельный код из регистра RG поступает на адресные входы блока памяти RAM.

Предположим, что в искомом флаговом коде 10101 (малая разрядность выбрана для удобства иллюстрации идеи) допустима одиночная ошибка. Тогда помимо этого кода схема поиска должна распознавать еще пять кодов, отличающихся от искомого в одном из разрядов: 00101, 11101, 10001, 10111 и 10100. Для распознавания правильного и пяти близких к нему ошибочных кодов в память предварительно записывается информация, представленная в Таблица 7.3 табл. 7.3.

Таблица 7.3 табл. 7.3

Таблица кодировки блока памяти (см. рис. 7.35, а)

Адрес	Эталонный код (X)	Число ошибок (Y)
00000	-	-
00001	-	-
00010	-	-
00011	-	-
00100	-	-
00101	10101	1
00110	-	-
00111	-	-
01000	-	-
01001	-	-
01010	-	-
01011	-	-
01100	-	-
01101	-	-
01110	-	-
01111	-	-
10000	-	-
10001	10101	1
10010	-	-
10011	-	-
10100	10101	1
10101	10101	0
10110	-	-
10111	10101	1
11000	-	-
11001	-	-
11010	-	-
11011	-	-
11100	-	-
11101	10101	1
11110	-	-
11111	-	-

Примечание. Символ “-” обозначает отсутствие данных и может кодироваться отдельным битом

Из таблицы следует, что точное или приближенное (с ошибкой в одном разряде) совпадение входного кода с заданным флаговым кодом сопровождается выдачей из памяти эталонного кода X и числа ошибок Y. В данном примере эталонный код уника-

лен (10101), поэтому из таблицы можно было бы исключить второй столбец. Однако эталонных кодов может быть несколько, например когда каждый из них соответствует кадру определенного иерархического уровня. В этом случае по считанным из памяти кодам X и Y определяется ближайший эталонный код и степень близости к нему входного кода. На основе считанных из памяти данных накапливается статистическая информация, по которой можно провести правильное разбиение потока данных на кадры.

С увеличением длины флагового кода растет требуемый объем блока памяти. Так, для анализа 20-разрядного кода необходим блок памяти объемом 2^{20} ячеек. Чтобы уменьшить этот объем, можно вместо одного применить несколько параллельно включенных блоков памяти. Как показано на рис. 7.35, б, 20-разрядный код из регистра RG разделен на две группы по десять разрядов. Каждая группа используется в качестве адреса для выбора одной из $2^{10} = 1024$ ячеек в соответствующих блоках памяти RAM1 и RAM2. Эти блоки осуществляют сравнение фрагментов входного и эталонных кодов.

Такое решение, однако, предполагает наличие дополнительных схем (на рисунке не показаны) для согласованной оценки результатов сравнения фрагментов кодов. Так, условие $Y1 = Y2 = 0$ соответствует точному совпадению входного кода с одним из эталонных только в том случае, когда 20-разрядный код $X1 X2$ также совпадает с этим же эталонным кодом. (Иными словами, нужно исключить из рассмотрения совпадения фрагментов входного кода с соответствующими фрагментами, принадлежащими разным эталонным кодам.) Ошибки, обнаруженные во фрагментах входного кода (коды $Y1$ и $Y2$), должны суммироваться для последующего сравнения с максимально допустимым числом.

Таким образом, в условиях повышенного уровня помех приемник проявляет определенную “терпимость” к ошибкам во флаговых кодах. Он пытается поддерживать достигнутую ранее или возобновить утраченную синхронизацию с передатчиком. Принятие решения о наличии или потере синхронизации основывается на основе анализа статистических данных об ошибках.

7.7. Поиск флага в потоке данных, передаваемых по волоконно-оптической линии связи

Задача состоит в обнаружении флаговой комбинации битов во входном потоке данных. Но специфика ее решения [19] обусловлена очень высокой скоростью передачи данных, при которой длительность битового интервала составляет доли наносекунды.

Устройство для распознавания флаговой комбинации битов (рис. 7.36) содержит оптический разветвитель (рис. 7.37), двенадцать оптических линий задержки, два фотоприемника, дифференциальный усилитель и компаратор. В данном примере флаг представлен 13-разрядным кодом 1111100110101. При обнаружении флага формируется сигнал $S = 1$.

Данные D передаются по волоконно-оптической линии связи в виде световых сигналов (рис. 7.38). Наличие света соответствует лог. 1, отсутствие света – лог. 0. Каждый бит размещен в отведенном ему временном интервале.

Оптический разветвитель разделяет входной световой поток на 13 равных по интенсивности частей. (На входе устройства можно установить усилитель для повышения мощности потока.) Каждая часть разделенного светового потока попадает в соответствующее выходное оптоволокно (линию). Верхняя (по схеме) линия (см. рис. 7.36) предназначена для непосредственной передачи сигнала на один из входов фотоприемника F2. Задержка передачи сигнала по этой линии условно принимается равной нулю (“нуль, умноженный на T ” или $0T$; T – длительность битового интервала).

Задержка передачи сигнала по второй – тринадцатой линиям составляет соответственно $1T$, $2T$, $3T$, . . . , $12T$. Задержка сигнала в каждой линии определяется числом последовательно включенных петель из оптоволоконна. Каждая петля задерживает сигнал на время, равное длительности одного битового интервала T . Такая структура позволяет развернуть последовательный код в параллельный.

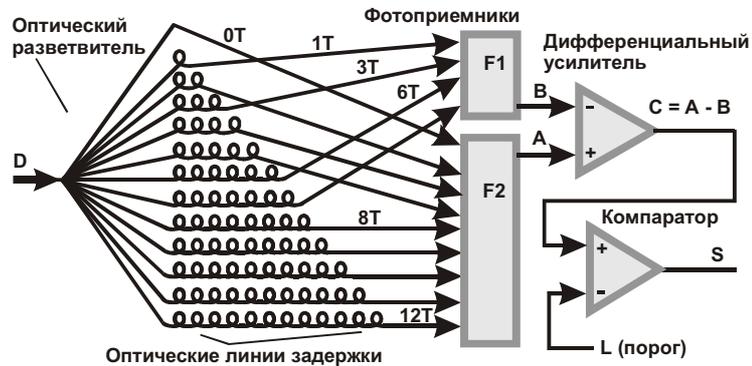


Рис. 7.36. Схема устройства для распознавания флаговой комбинации битов во входном потоке данных D , поступающем по волоконно-оптической линии связи

Действительно, одновременно с поступлением новейшего бита на входы фотоприемников поступают: новый, задержанный на один такт ($1T$); предшествующий ему и задержанный на два такта ($2T$) и т. д. и, наконец, наистарейший бит, задержанный на 12 тактов ($12T$). Таким образом, на входах фотоприемников образуется 13-разрядное окно, через которое можно просматривать последовательность входных данных D . При этом наистарейший бит соответствует принятому на вход устройства в интервале t_0 (см. рис. 7.38), а новейший бит – принятому в интервале t_{12} .

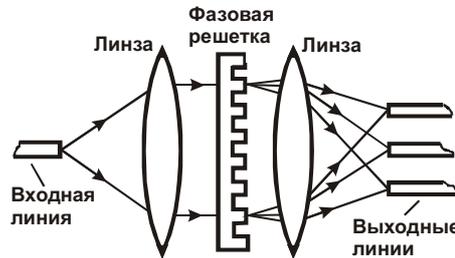


рис. 7.37. Конструкция оптического разветвителя

Выбранный в данном примере вариант подключения фотоприемников к предыдущим каскадам устройства определяется искомым флаговым кодом (1111100110101). Девять единичных битов этого кода собираются фотоприемником $F2$, а четыре нулевых – фотоприемником $F1$. Электрический сигнал на выходе фотоприемника пропорционален суммарной мощности световых потоков на его входах.

При поступлении искомого флагового кода фотоприемник $F1$ не получает световой энергии ни по одному из входов. Напротив, фотоприемник $F2$ получает световые потоки от всех входных линий. Такая комбинация сигналов уникальна и сопровождается максимально возможной разностью напряжений между точками A и B на входах дифференциального усилителя (см. рис. 7.38). Остается только зарегистрировать максимальную разность, и задача решена.

Для этого разностный сигнал $C = A - B$ подается на компаратор и сравнивается с заданным порогом L . Сигнал опознания $S = 1$ формируется при превышении порога сигналом C . Порог выбирается настолько высоким, что сигнал $S = 1$ формируется только при точном опознании флагового кода.

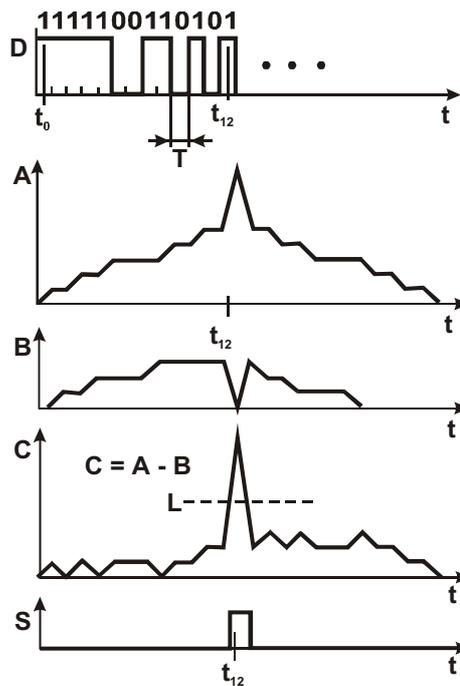


Рис. 7.38. Временные диаграммы сигналов при распознавании флага

Снижение порога позволяет вести не точный, а приближенный поиск нужной последовательности битов во входном потоке. Например, сигнал $S = 1$ при определенном снижении порога может формироваться не только в результате поступления флагового кода (1111100110101), но и любого близкого к нему, содержащего одиночную ошибку (например, 1111100110100, ошибка в младшем разряде). Это может оказаться полезным при повышенном уровне ошибок в канале связи. Полученные результаты приближенного поиска сортируются, и, с учетом предыстории, среди них выявляются наиболее правдоподобные, периодически повторяющиеся, которые интерпретируются как флаги.

В [19] показано, что при оптимальном выборе флагового кода можно получить максимально контрастные картины опознания, при которых пик разностного сигнала C выражен наиболее ярко.

7.8. Передача данных вместо избыточных битов синхронизации кадра

Как уже отмечалось, данные, передаваемые по транспортной сети, группируют в структурные единицы, которые в зависимости от используемой технологии называют контейнерами, ячейками, пакетами, кадрами и т. п. В любом случае удаленный узел – приемник данных должен распознать начало передаваемой по сети структурной единицы.

Рассмотрим одну из наиболее простых структурных единиц – кадр, предусмотренный рекомендацией ITU-T V.110 (рис. 7.39, а).

Кадр построен в виде матрицы из десяти строк и восьми столбцов. В пересечении каждой строки и столбца размещен один бит, так что кадр содержит 80 бит. Первым передается крайний левый бит верхней строки (лог. 0), последним – крайний правый бит B_{63} нижней строки. Биты $B_1 - B_{63}$ представляют собой передаваемые данные; $S_1 - S_8$ – так называемые биты синхронизации.

В начале кадра всегда содержится флаг – кодовая комбинация из восьми лог. 0 и последующей лог. 1 (00000001). Такая комбинация не должна встречаться в любом другом месте кадра; это гарантируется введением битов $S_1 - S_8$ синхронизации, каж-

дый из которых представлен лог. 1. Из приведенного на рис. 7.39, б примера видно, что в каждой строке матрицы, начиная со второй, содержится по крайней мере одна лог. 1, так что флаговая комбинация просто не умещается где-либо в ненадлежащем месте. Как видим, все сделано правильно; но “работа по правилам” не всегда дает наилучший результат. Действительно, левый столбец матрицы является избыточной константой, рассчитанной на наихудший случай, когда все биты В1 – В63 представлены лог. 0 либо среди них имеется незначительное число лог. 1. Нельзя ли уменьшить избыточность левого столбца?

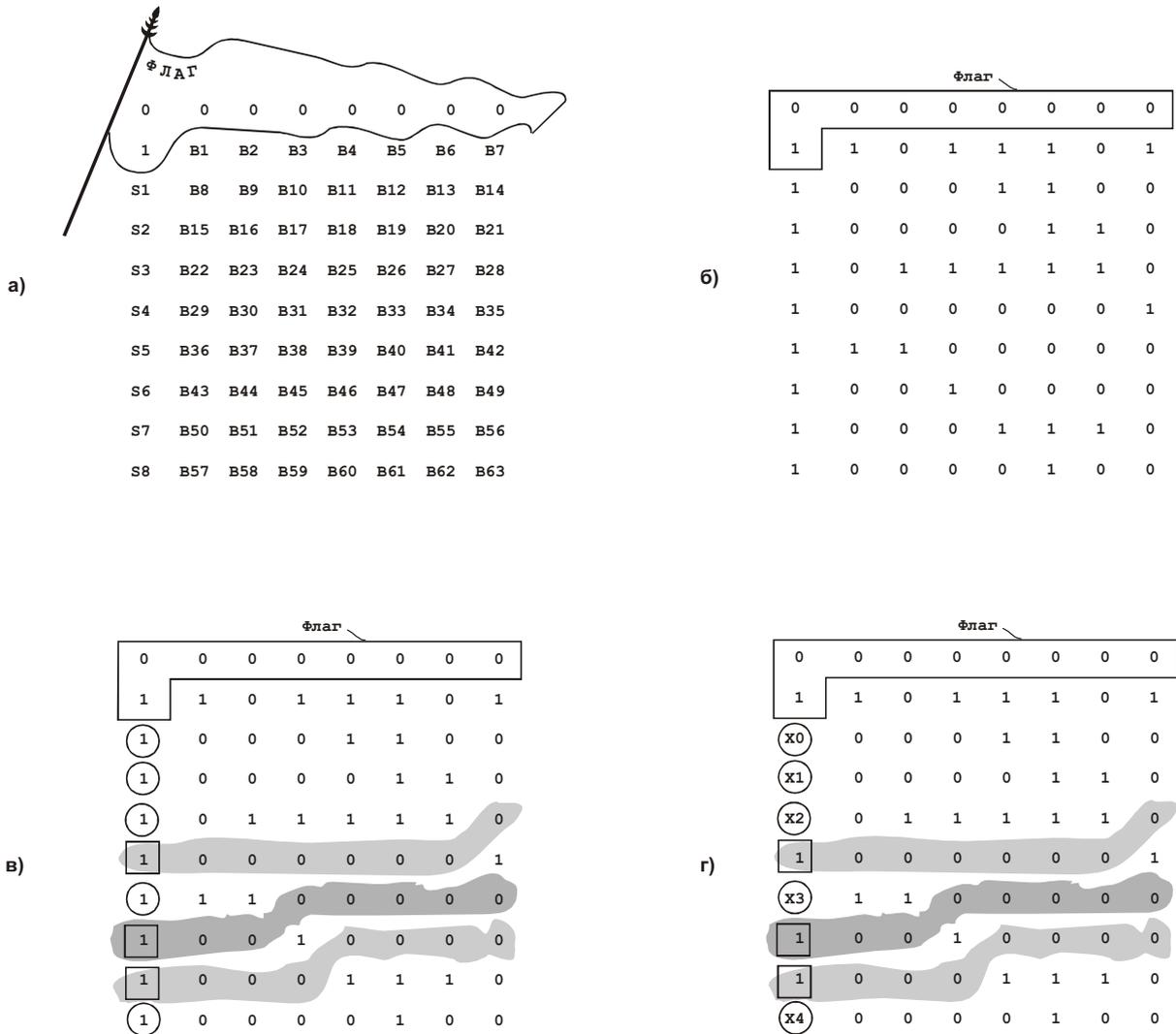


рис. 7.39. Передача данных вместо избыточных битов синхронизации кадра: а – общая структура кадра в соответствии с рекомендацией ITU-T V.110; б – кодовый пример кадра; в – поиск битов синхронизации, которые должны безусловно принимать значения лог. 1 (выделены квадратиками); г – определение позиций для передачи данных (X0 – X4) вместо битов синхронизации (выделены кружочками)

Обратимся к рис. 7.39, в и проанализируем значимость битов синхронизации, начиная с первого S1 (крайнего левого бита третьей сверху строки матрицы). Он, как и предусмотрено рекомендацией V.110, равен лог. 1. Но, просматривая его ближайшее окружение, видим, что ложного опознания флага не произойдет, если биту S1 присвоить значение, равное лог. 0.

Действительно, в третьей сверху строке матрицы при S1 = 0 появится непрерывная последовательность из четырех лог. 0, но это по-прежнему гарантирует удаленный

приемник кадра от ложного опознания флага (не только на своем законном месте). Отметим факт свободы выбора значения бита S_1 , выделив его на рисунке кружочком.

Подобные рассуждения применимы также к битам S_2 , S_3 , S_5 и S_8 , выделенным на рисунке кружочками. Эти биты могли бы принимать не только единичные, но и нулевые значения, не мешая удаленному приемнику распознать флаг только на своем законном месте – в начале кадра. В то же время биты S_4 , S_6 и S_7 (выделены квадратиками) должны безусловно принимать значения лог. 1. В противном случае сформируются достаточно длинные последовательности из лог. 0 (выделены тремя затененными областями), которые, например, при $S_4 = S_6 = S_7 = 0$ приведут к трехкратному ложному опознанию начала кадра в его теле.

Таким образом, становится понятной идея, предложенная в [3]. Ее можно сформулировать так: в передаваемом кадре V_{110} обычно можно отыскать вакантные места для размещения от одного до восьми битов – своего рода “неучтенных пассажиров” транспортной системы. В нашем примере имеем пять таких битов $X_0 - X_4$ (рис. 7.39, *з*).

Теперь возникает вопрос: как удаленный приемник сможет понять, какие биты полученного кадра обведены кружочками, а какие – квадратиками? Ведь информация о форме обвода в явном виде не передается по линии! Попробуем с этим разобраться.

Алгоритм работы приемника таков.

1. Приемник, постоянно прослушивая линию, обнаруживает поступление флага (кода 00000001), после чего принимает кадр (см. рис. 7.39, *а*) и записывает его в первой области памяти, выделенной для временного хранения кадров.

2. Приемник запоминает биты $S_1 - S_8$ во второй области памяти, выделенной для их временного хранения, а на места этих битов в матрице помещает лог. 1. Таким образом, приемник “видит” матрицу, показанную на рис. 7.39, *б*.

3. Приемник проводит описанный ранее анализ необходимости существования каждого бита синхронизации и, если не было ошибок передачи данных $V_1 - V_{63}$, приходит к тем же результатам, которые были приведены на рис. 7.39, *в*. В нашем примере приемник приходит к выводу, что биты S_4 , S_6 и S_7 должны быть представлены лог. 1.

Он проверяет, так ли это, извлекая из второй области памяти соответствующие биты. (Любое несовпадение расценивается как ошибка передачи, и кадр отбрасывается.) Попутно выявляются позиции битов, обведенных кружочками. Теперь остается только извлечь из второй области памяти соответствующие биты ($S_1 - S_3$, S_5 , S_8) и скомпоновать из них искомый код $X_0 \dots X_4$ (см. рис. 7.39, *з*). Таким образом, “неучтенные пассажиры” благополучно прибыли на место.

Можно предложить много вариантов использования кодов $X_0 \dots X_i$. Например:

1) коды $X_0 \dots X_i$ переменной длины, передаваемые в последовательности кадров, можно рассматривать как фрагменты некоторого, в общем случае независимого, массива данных. Иными словами, можно организовать второй, параллельный канал передачи данных, который работает не в ущерб первому, основному;

2) каждый код $X_0 \dots X_i$ можно рассматривать как некое контролирующее приложение к данным $V_1 - V_{63}$, которые передаются в этом же кадре. Например, разряд X_0 может представлять собой бит контроля по четности всей группы $V_1 - V_{63}$. Разряды X_1 и X_2 могут соответственно служить битами контроля по четности подгрупп $V_1 - V_{31}$ и $V_{32} - V_{63}$ и т. д. с уменьшением длины контролируемой области.

Иное возможное распределение функций разрядов: бит X_i контролирует по четности данные, размещенные в своей и предыдущей строках матрицы. Такое решение позволяет обнаруживать ошибки “на лету”, не дожидаясь окончания приема полного кадра.

Возможны варианты использования кодов $X_0 \dots X_i$ для уменьшения вероятности ложного опознания флага при наличии ошибок передачи [3].

7.9. Способы размещения низкоскоростного потока данных в высокоскоростном потоке кадров

Для передачи данных от некоторого источника, например, от низкоскоростного аналого-цифрового преобразователя (АЦП), к удаленному регистрирующему устройству, например, компьютеру, может использоваться высокоскоростной канал связи (рис. 7.40). Канал содержит передающую и приемную аппаратуру (например, два мультиплексора), подключенную к противоположным сторонам линии связи. Разумеется, канал связи передает потоки данных и от других источников (на рисунке не показаны). Задача состоит в рациональном размещении низкоскоростного потока данных D в высокоскоростном потоке кадров, передаваемых по линии. Рассмотрим два способа решения этой задачи [29].

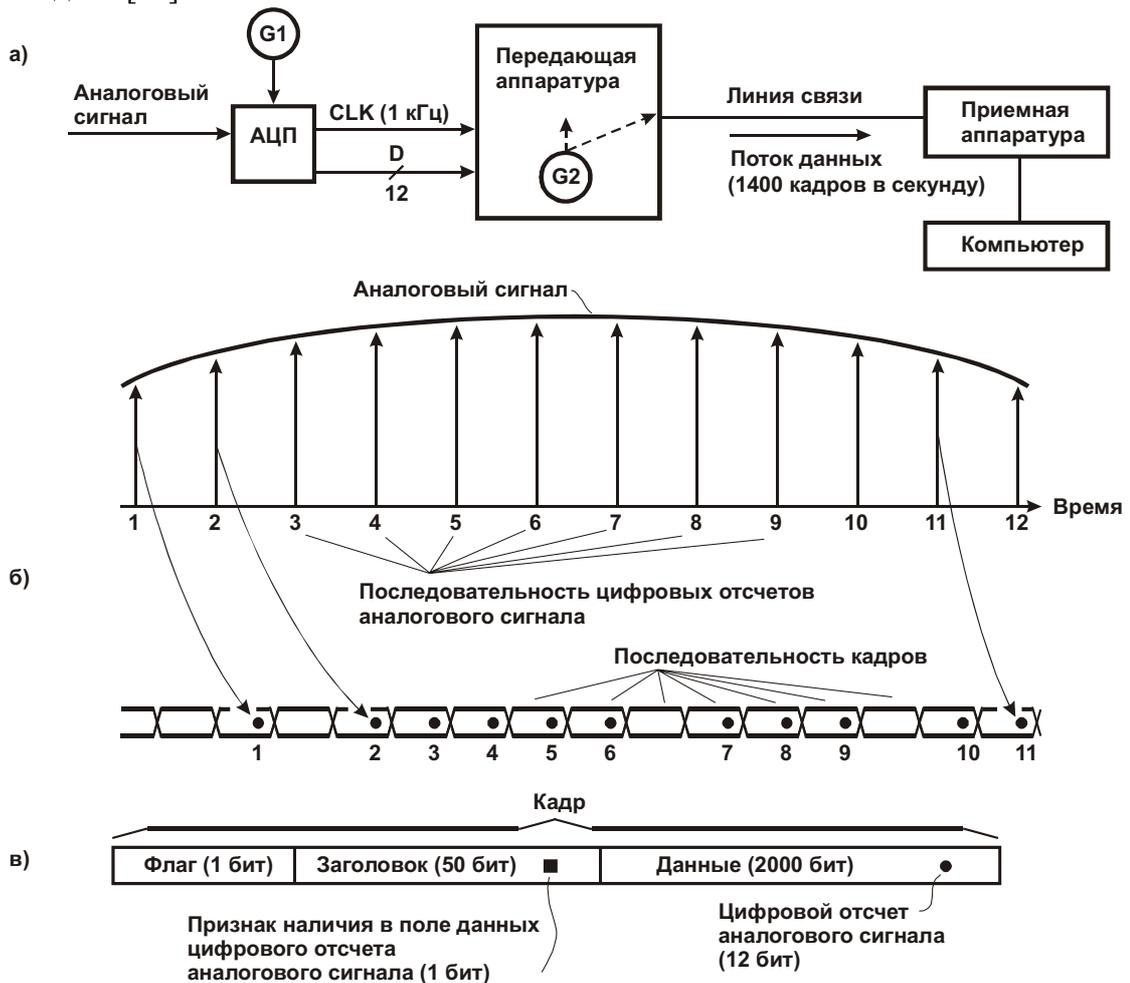


Рис. 7.40. Передача низкоскоростного потока данных от аналого-цифрового преобразователя к удаленному компьютеру (вариант 1): *а* – схема; *б* – размещение потока данных в потоке кадров; *в* – структура кадра

Первый способ. Как показано на рис. 7.40, АЦП формирует непрерывную последовательность 12-разрядных цифровых отсчетов D аналогового сигнала. Синхронизация АЦП осуществляется от генератора синхросигналов $G1$. Отсчеты следуют с частотой 1кГц и сопровождаются синхросигналом CLK.

Передающая аппаратура синхронизируется сигналом от генератора $G2$. Данные поступают в линию связи в виде непрерывной последовательности кадров со скоро-

стью 1400 кадров в секунду. Кадр содержит одноразрядный флаг (см. п. 7.2.1), обозначающий его начало, 50-разрядный заголовок со служебными данными и поле данных (2000 бит).

Передающая аппаратура принимает очередной отсчет D аналогового сигнала и размещает его в определенной группе разрядов поля данных ближайшего формируемого кадра. В каждый кадр может быть помещен только один отсчет. Факт его размещения помечается установкой в состояние лог. 1 определенного бита в заголовке кадра. Как следует из рис. 7.40, б, отсчеты переносятся в поток кадров с нерегулярным темпом, по мере их поступления от АЦП и наличия “свободных мест” в транспортной системе.

Приемная аппаратура анализирует поступающие кадры. При обнаружении в заголовке кадра признака наличия цифрового отсчета аналогового сигнала соответствующий 12-разрядный код извлекается из поля данных и передается в компьютер для дальнейшей обработки. Если кадр не содержит данных от АЦП, то отведенное для них место используется для передачи данных от других источников.

Второй способ. Передача данных от АЦП в компьютер может синхронизироваться одним генератором G (рис. 7.41), при этом скорость поступления данных, как и в предыдущем примере (рис. 7.40, а), не совпадает со скоростью потока кадров. Однако в данном случае возможно регулярное размещение данных в потоке кадров. Это означает, что в каждой группе из M кадров размещаются N цифровых отсчетов аналогового сигнала. В примере, приведенном на рис. 7.41, поток разбивается на группы по девять кадров ($M = 9$), а в каждой группе размещаются четыре отсчета аналогового сигнала ($N = 4$).

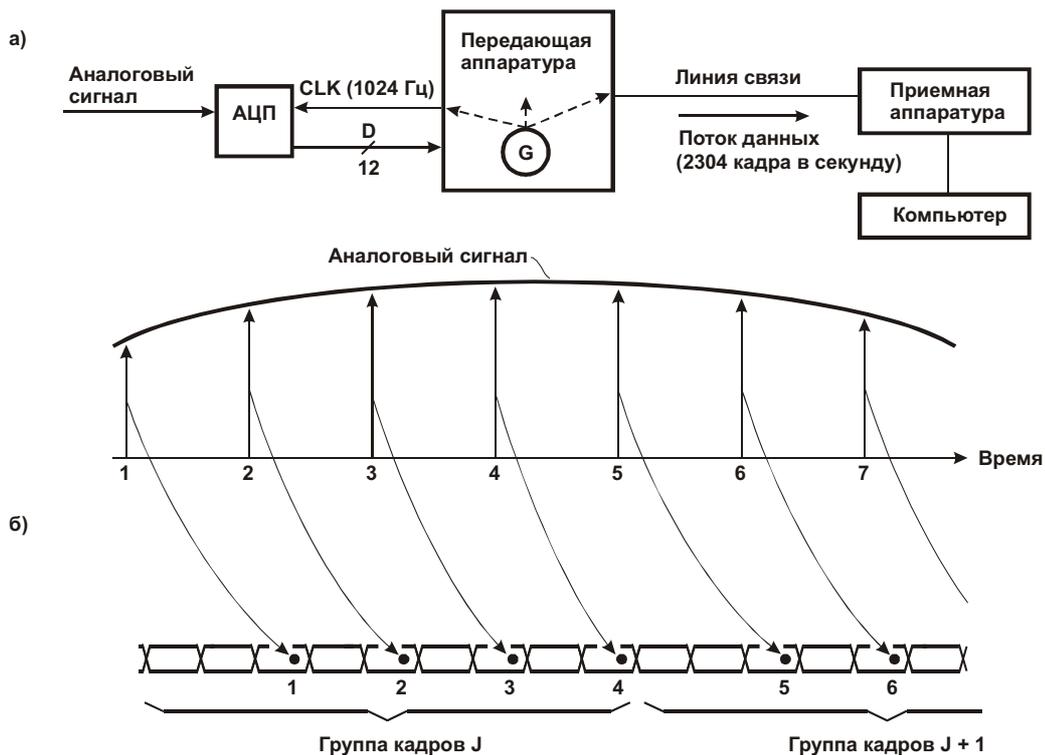


Рис. 7.41. Передача низкоскоростного потока данных от аналого-цифрового преобразователя к удаленному компьютеру (вариант 2): а – схема; б – размещение потока данных в потоке кадров

Убедимся в правильности такого разбиения. За одну секунду по линии связи передаются 2304 кадра или 256 групп по девять кадров. В каждой группе содержатся четы-

ре отсчета, поэтому их общее число составит $256 \times 4 = 1024$, что соответствует скорости поступления данных от АЦП.

В общем случае при некратном отношении частоты F_F следования кадров к частоте F_S следования цифровых отсчетов аналогового сигнала выполняется условие $F_F/F_S = A + b/c$, где $F_F/F_S > 1$, A – целая часть отношения частот, b/c – несократимая дробь. Это условие можно записать в следующем виде: $F_F/F_S = (Ac + b)/c$, где $Ac + b = M$, $c = N$.

Разметку потока кадров на группы можно выполнить по меньшей мере двумя способами.

1. Первый кадр каждой группы может помечаться установкой в единичное состояние некоторого служебного бита в его заголовке. Приемная аппаратура, обнаружив признак начала группы кадров, получает ориентир для поиска и выделения данных от АЦП. Это возможно благодаря тому, что размещение этих данных в каждой группе кадров одинаково и заранее известно. Так, в приведенном на рис. 7.41 примере первый и второй кадры каждой группы не содержат данных от АЦП, в третьем кадре данные имеются и т. д.

2. Для более экономной разметки потока кадров на группы можно использовать псевдослучайные одноразрядные флаги (см. п. 7.2.2). Экономия заключается в том, что применение псевдослучайной последовательности флагов позволяет приемной аппаратуре находить начало группы кадров только на основе анализа этой последовательности. При этом нет необходимости помечать первый кадр группы служебным битом в заголовке.

Применительно к рассматриваемой системе передачи данных (рис. 7.41) следует генерировать псевдослучайную последовательность битов с периодом повторения, равным или кратным девяти. Точнее, последовательность не обязательно должна быть псевдослучайной в строгом смысле слова (см. п. 8.4.1). В данном случае достаточно принять за основу классическую псевдослучайную последовательность с периодом повторения, равным $2^4 - 1 = 15$ бит и выделить из нее только девять следующих подряд элементов, отбросив шесть остальных. Один из вариантов формирования такой “усеченной” последовательности поясняется рис. 7.42.

После начальной установки в регистре RG некоторого ненулевого кода (цепи установки на рисунке не показаны) классический генератор псевдослучайной битовой последовательности (рис. 7.42, *а*) может находиться в 15 состояниях (см. таблицу в левой верхней части рисунка), которые периодически повторяются. Копии этих состояний можно получить при просмотре выходной последовательности битов через четырехразрядное движущееся окно, как показано в правой верхней части рисунка.

Чтобы исключить “лишние” состояния регистра RG , в схему генератора введены дополнительные элементы (рис. 7.42, *б*). Элемент I формирует сигнал лог. 1 при обнаружении в регистре кода $0011_2 = 3_{10}$. В начале следующего такта, по положительному фронту сигнала CLK триггер $D1$ устанавливается в единицу, что вызывает безусловный переход регистра в состояние $1111 = 15$. При переходе синхросигнала CLK в состояние лог. 0 устанавливается в единицу триггер $D2$, что вызывает установку в нуль триггера $D1$ и снятие сигнала установки кода 1111 с входа регистра. В начале следующего такта в регистре формируется код $1110 = 14$, а в середине этого такта сигнал лог. 0 переписывается из триггера $D1$ в триггер $D2$.

В последующих тактах регистр последовательно проходит состояния $12, 8, 1, 2, 4, 9, 3$, затем описанный цикл генерации кодов повторяется. В результате число различных состояний регистра сокращено до девяти, что и требовалось. Отметим, что в данном случае последовательность состояний регистра не соответствует кодам, наблюдаемым при просмотре выходной последовательности битов через движущееся четырехразрядное окно. Однако это не приводит к каким-либо неопределенностям, так как эти коды можно однозначно сопоставить с состояниями регистра.

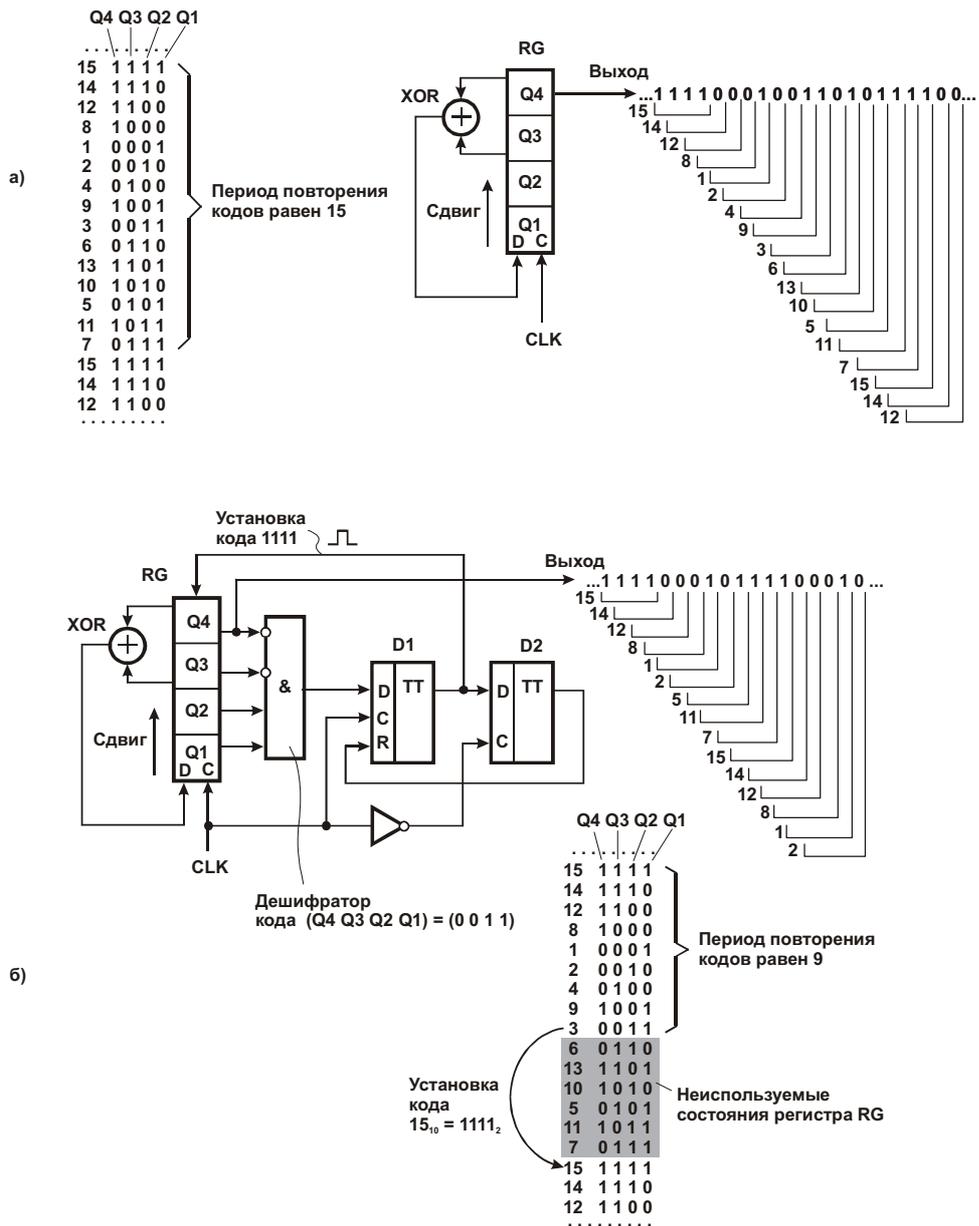


Рис. 7.42. Генераторы, формирующие: *а* – псевдослучайную битовую последовательность максимальной длины с периодом повторения, равным 15; *б* – битовую последовательность с периодом повторения, равным 9

Последовательность генерируемых таким способом битов размещается передающей аппаратурой во флаговых позициях формируемых кадров – по одному биту на каждый кадр. Эта последовательность выделяется из кадров приемной аппаратурой и просматривается ею через четырехразрядное окно (см. рис. 7.42, б). Согласно протоколу обмена, один из кодов в окне, например код 1111, является признаком начала группы из девяти кадров. Обнаружив такой код, приемная аппаратура “узнаёт”, что последний принятый кадр является первым в группе (возможны и иные договоренности). Разумеется, что передающая аппаратура должна знать, что к этому моменту она действительно переслала именно первый кадр группы. Таким образом, осуществляется кадровая синхронизация между приемной и передающей аппаратурой.

Рассмотренные решения позволяют согласовать темп поступления данных от низкоскоростного источника с темпом их передачи по высокоскоростной транспортной системе. Для такого согласования можно выделить один служебный бит в каждом кад-

ре. Но можно и не вводить служебные биты, если использовать псевдослучайные флаги начала кадров.

7.10. Уменьшение числа операций, выполняемых при распознавании флага начала кадра

Распознавание многоразрядных флагов начала кадра в потоке данных обычно реализуется с помощью микропроцессорного устройства. Это устройство выполняет некоторую циклическую программу, предусматривающую ряд операций извлечения данных из буферной памяти и их сопоставления с заранее известным флаговым кодом. Уменьшение общего числа таких операций позволяет снизить тактовую частоту микропроцессора и уменьшить интенсивность считывания данных из буферной памяти, что, в свою очередь, снижает потребляемую устройством мощность.

Рассмотрим один из способов уменьшения числа операций, выполняемых при распознавании флага начала кадра [41]. Чтобы показать преимущество этого способа перед традиционным, предположим, что тот и другой реализуются с использованием одной и той же аппаратуры обработки сигнала, поступающего из линии (рис. 7.43, а). Прежде всего, из этого сигнала выделяется битовый поток данных DATA и сопровождающий его синхросигнал CLK. Далее поток битов преобразуется в поток байтов D, сопровождаемый синхросигналом CLK/8 с выхода делителя частоты на восемь. Поток байтов поступает в буферную память и считывается из нее для дальнейшей обработки (см. штриховую линию на рисунке). Устройство распознавания флага выполнено на основе микропроцессора. Оно анализирует проходящий через буферную память поток данных и при обнаружении флага начала кадра формирует сигнал FOUND.

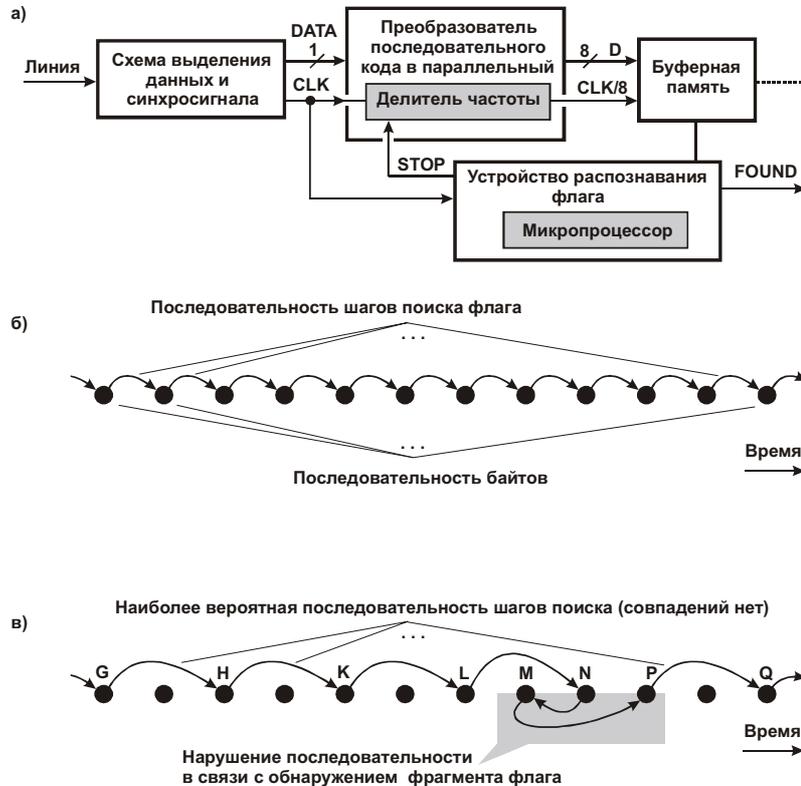


рис. 7.43. Распознавание флага начала кадра: а – структурная схема аппаратуры обработки сигнала из линии; б, в – диаграммы, соответствующие традиционному и предлагаемому способам поиска

Исходное состояние делителя частоты может быть произвольным, поэтому первоначальное разделение битового потока данных DATA на байты D, вероятнее всего, неверно, т. е. не совпадает с разделением, выполненным удаленным передатчиком данных в линию. Следствием этого является невозможность регулярного обнаружения флагов начала кадров в потоке байтов D, что достижимо только при правильном определении границ байтов.

Чтобы найти истинные границы байтов, устройство распознавания флага проводит серию экспериментов по поиску регулярной последовательности флагов. В каждом эксперименте границы байтов смещаются на один бит, поэтому число экспериментов не превышает семи, а при удачном стечении обстоятельств сразу же выясняется, что предполагаемые границы байтов совпадают с истинными границами. Для того чтобы сместить границы байтов на один бит, устройство распознавания флага формирует сигнал STOP, который приостанавливает работу делителя частоты на один такт сигнала CLK.

Предположим, что границы байтов в битовом потоке определены правильно, а флаг представлен 32-разрядным кодом

$$0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 1011\ 0011_2 = 00\ 00\ 01\ B3_{16}.$$

Традиционный способ поиска флага предписывает микропроцессору последовательно просматривать байты, размещенные в буферной памяти (рис. 7.43, б) и сравнивать их с байтами кода $00\ 00\ 01\ B3_{16}$, начиная с первого (00_{16}). Успешное распознавание флага сопровождается последовательным совпадением четырех байтов данных с байтами этого кода.

Предлагаемый способ поиска предусматривает проведение беглого оценочного просмотра последовательности байтов. Просмотр становится более детальным только в окрестностях тех точек (байтов), которые подозреваются на причастность к переносу флага (рис. 7.43, в). В результате число выполняемых операций, условно показанных на рисунке стрелками, уменьшается, что и требовалось. Рассмотрим оба способа поиска подробнее.

Традиционный способ поиска флага описывается диаграммой состояний, приведенной на рис. 7.44. Стрелками помечены разрешенные переходы между состояниями. Блок-схема алгоритма поиска показана на рис. 7.45. Предположим, что в начале поиска устройство находится в состоянии 0. Если очередной байт равен 00_{16} , то осуществляется переход в состояние 1, в противном случае сохраняется состояние 0. При повторном обнаружении байта 00_{16} происходит переход в состояние 2, в противном случае – в состояние 0 и т. д. Эти же условия соответствуют переходам между блоками 1 – 4 на блок-схеме алгоритма. Обнаружение флага соответствует переходу в состояние 4 или, что то же самое, выходу из блока 8 по ветви “Да”.

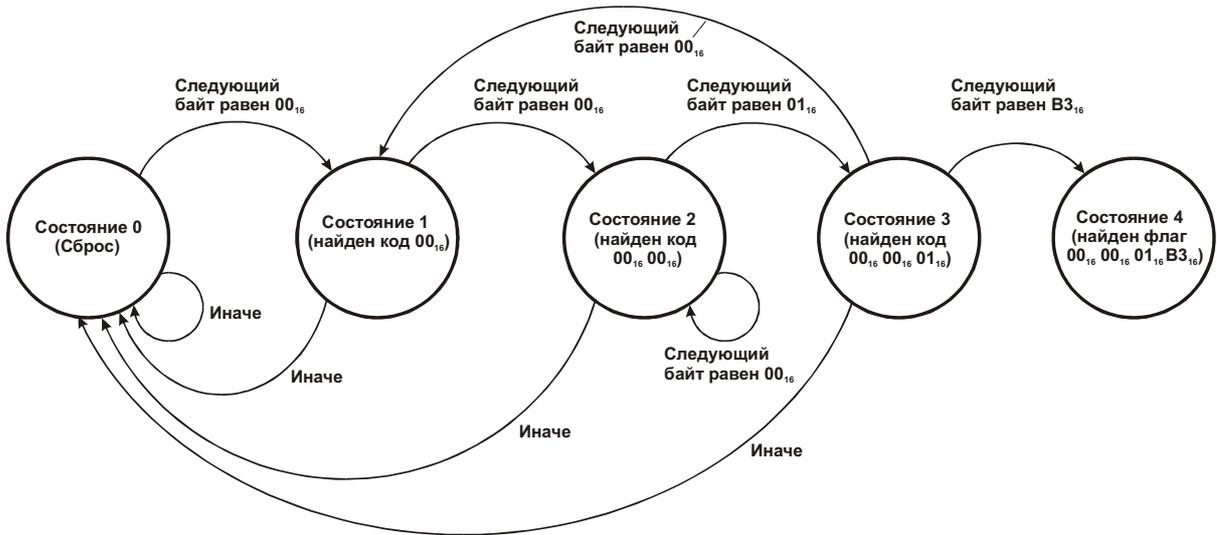


рис. 7.44. Диаграмма состояний устройства распознавания флага (традиционный способ поиска)

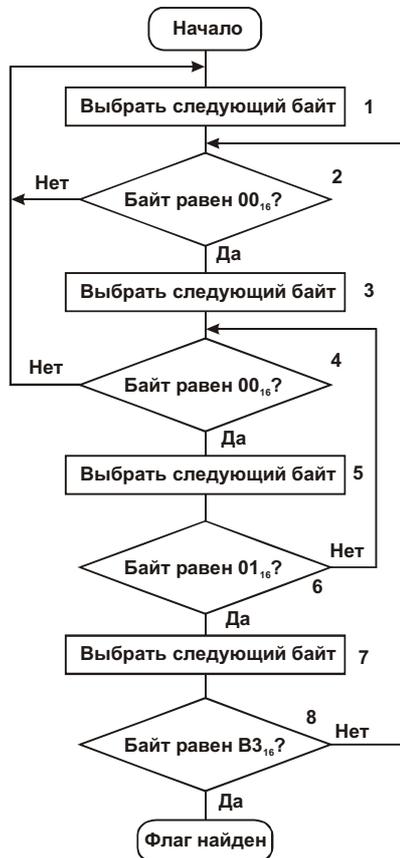


рис. 7.45. Блок-схема алгоритма поиска флага (традиционный вариант)

Диаграмма состояний и блок-схема предлагаемого алгоритма поиска флага приведены на рис. 7.46 и рис. 7.47.

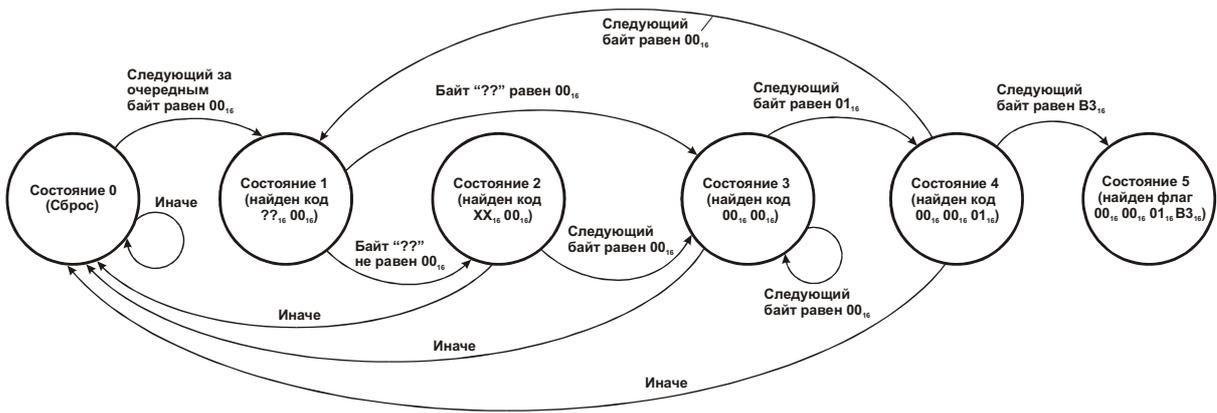


рис. 7.46. Диаграмма состояний устройства распознавания флага (предлагаемый способ поиска)

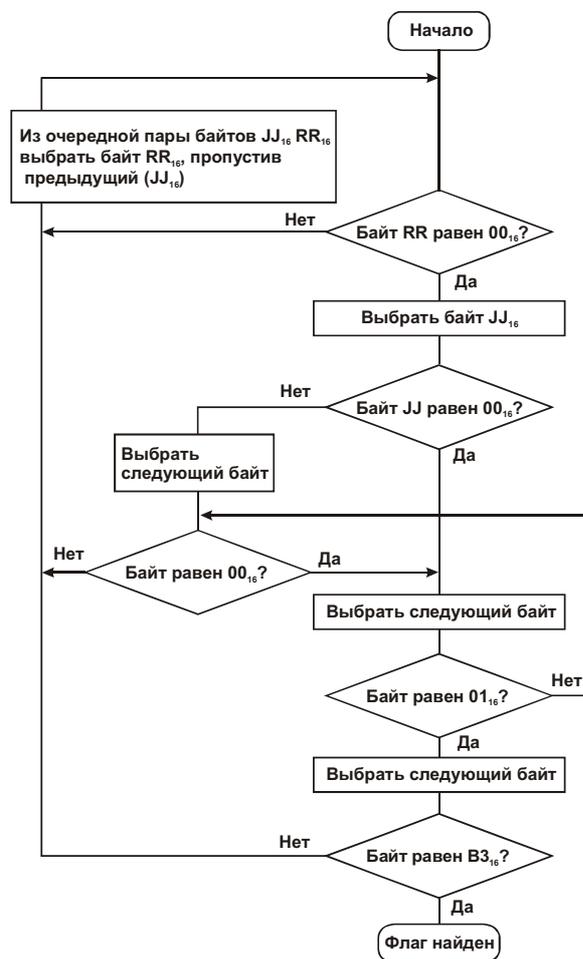


рис. 7.47. Блок-схема алгоритма поиска флага (предлагаемый вариант)

Отличие предлагаемого способа поиска от традиционного, как уже отмечалось, состоит в том, что вместо полного контроля последовательности байтов проводится ее грубый просмотр, а подробности анализируются лишь по мере необходимости уточнения ситуации. Такой способ поиска правомерен в силу следующих причин.

1. Флаги встречаются в потоке данных сравнительно редко, особенно если они представлены уникальными кодами (в случае применения битстаффинга), а кадры содержат большое число байтов. Поэтому столь же редко возникает необходимость полного анализа четырехбайтового кода в связи с обнаружением флага.

2. Необходимым (но не достаточным) условием обнаружения флага является регистрация двух размещенных рядом нулевых байтов. Поэтому для проверки этого условия можно проверять не все байты подряд, а продвигаться по их последовательности с шагом, равным двум байтам. Данные в кадрах можно рассматривать как случайные – это безусловно справедливо, если применено их скремблирование. Тогда вероятность обнаружения нулевого байта равна $1/256$. Иными словами, ускоренное продвижение по последовательности байтов – скорее норма, чем исключение.

Для пояснения сказанного вернемся к диаграмме, показанной на рис. 7.43, в. В процессе поиска флага последовательно анализируются ненулевые байты G, H, K и L, расположенные в буферной памяти с интервалом в две ячейки. На диаграмме состояний (рис. 7.46) этот этап поиска соответствует неудачным попыткам выхода устройства из состояния 0. После байта L выбирается нулевой байт N, что соответствует переходу устройства в состояние 1. При этом возникает необходимость проверить предыдущий байт M, так как появилась некоторая надежда обнаружить флаг. Однако этот байт при проверке оказывается ненулевым, устройство переходит в состояние 2. Далее выбирается байт P, но он также оказывается ненулевым, поэтому устройство возвращается в состояние 0. После этого ускоренный анализ продолжается – выбирается байт Q и т. д.

Таким образом, предлагаемый способ значительно сокращает число обращений к памяти и уменьшает объем вычислительных операций, выполняемых микропроцессором при поиске флага.

8. Способы кодирования данных для их передачи по каналу связи

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

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

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

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

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

8.1.1. Структура последовательного канала связи

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

В зависимости от расстояния данные, передаваемые по линии, могут однократно или многократно подвергаться ретрансляции с целью восстановления амплитуды и временных характеристик (рис. 8.1) [21, 71].

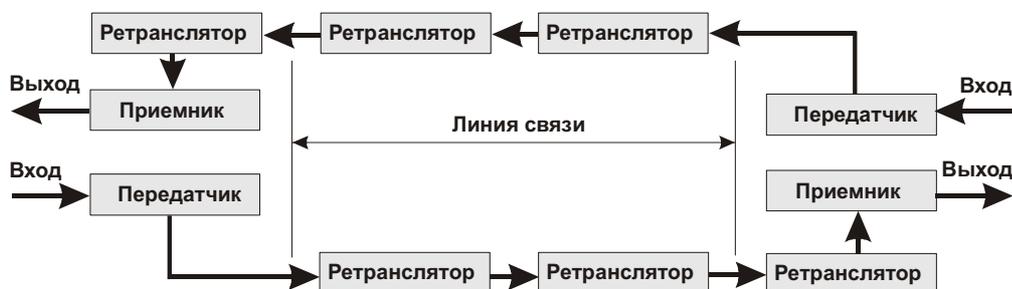


рис. 8.1. Структура типового последовательного канала связи

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

8.1.2. Униполярный код NRZ

Простейшим линейным кодом является униполярный код типа NRZ (Non Return to Zero), показанный на рис. 8.2, *а*. В этом коде нули представлены отсутствием импульса (напряжение, близкое нулю), а единицы – наличием импульса (некоторое положительное напряжение). Этот код имеет четыре недостатка.

1. Средняя мощность, выделяемая на нагрузочном резисторе R (на рисунке не показан), равна $A^2/2R$, где A – амплитуда импульса напряжения. Число 2 в знаменателе дроби соответствует равновероятному появлению лог. 0 и лог. 1 в потоке данных. Результат неутешительный. Резистор R рассеивает тепловую энергию в два раза интенсивнее, чем при биполярном кодировании (см. рис. 8.2, *б*) при той же амплитуде сигнала, равной A !

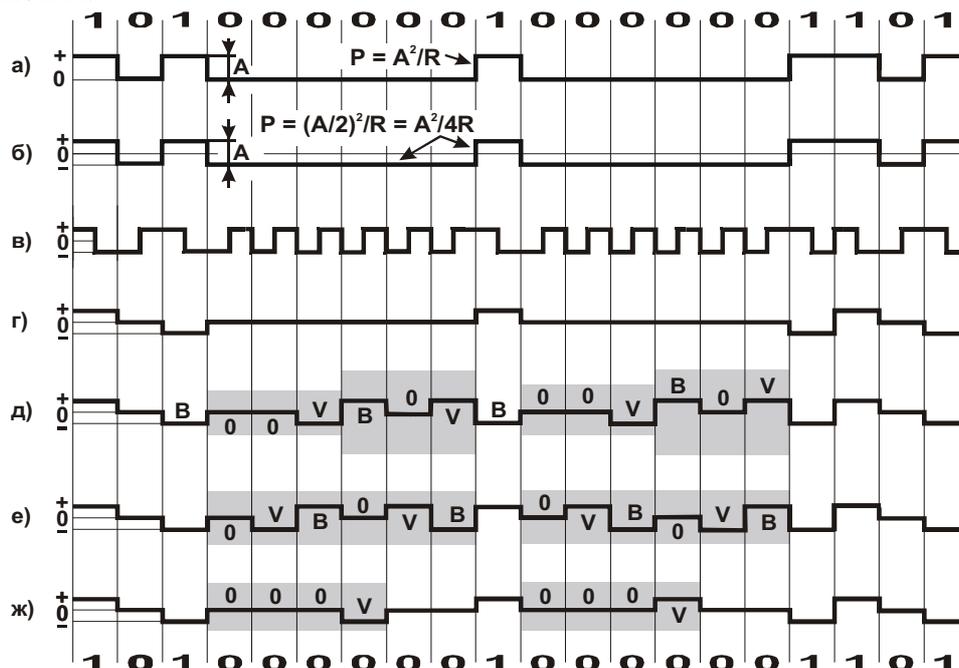


рис. 8.2. Наиболее распространенные линейные коды: *а* – униполярный код NRZ; *б* – биполярный код NRZ; *в* – код Манчестер-II; *г* – код AMI; *д* – код B3ZS; *е* – код B6ZS; *ж* – код HDB3; затемненными прямоугольниками выделены “заготовки”

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

3. Ретрансляторы и приемники надежно восстанавливают синхронизирующую временную сетку только тогда, когда паузы между изменениями сигнала не слишком велики. Изменение сигнала после незначительной паузы позволяет всякий раз корректировать “ход часов” ретранслятора или приемника. С увеличением паузы надежность “службы времени” падает. Например, после передачи серии из 10 тыс. нулей приемник, вероятнее всего, не сможет точно определить, находится ли последующая единица на позиции 9999, 10000 или 10001. То же относится и к передаче длинных цепочек из лог. 1. Другими словами, при передаче достаточно большой последовательности нулей или единиц приемник (или ретранслятор) теряет синхронизацию с передатчиком (или ретранслятором).

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

8.1.3. Биполярный код NRZ

Биполярный сигнал NRZ (рис. 8.2, б) по сравнению с униполярным обладает лучшими энергетическими характеристиками. Единица представлена положительным уровнем напряжения, нуль – отрицательным. Нагрузочный резистор R в данном случае постоянно рассеивает тепло, так как на нем независимо от передаваемого кода присутствует напряжение $A/2$ той или иной полярности. Средняя мощность, выделяемая на нагрузочном резисторе, равна $(A/2)^2/R = A^2/4R$, т. е. половине средней мощности униполярного сигнала, хотя перепад уровней тот же самый.

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

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

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

8.1.4. Код Манчестер-II

Примером кода с избыточностью, введенной согласно только что упомянутому первому способу, является код Манчестер-II. Форма биполярного сигнала при передаче кода Манчестер-II показана на рис. 8.2, в. Единица кодируется отрицательным перепадом сигнала в середине битового интервала, нуль – положительным перепадом. На границах битовых интервалов сигнал, если это необходимо, изменяет значение, готовясь к отображению очередного бита в середине следующего битового интервала.

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

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

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

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

8.1.5. Код AMI

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

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

последовательности единиц синхронизация не теряется. Обнаруживаются ошибки, нарушающие правильную последовательность знакочередующихся сигналов.

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

8.1.6. Коды BNZS, HDB3

Потеря синхронизации при передаче длинной последовательности нулей предотвращается так: цепочки нулей передатчик заменяет определенными “заготовками”, которые представляют собой “отрезки” стандартных временных диаграмм. Коды АМІ, в которых цепочка из N нулей заменяется определенной подстановкой, называются BNZS-кодами (Bipolar with N Zeroes Substitution).

В коде В3ZS (рис. 8.2, *д*) каждые три последовательных нуля подменяются либо комбинацией В0V, либо 00V. Символ В обозначает импульс, который отвечает правилам кодирования АМІ, символ V - импульс, который нарушает правила кодирования АМІ (совпадает по полярности с предыдущим).

Выбор одной из этих двух “заготовок” проводится так, чтобы, во-первых, число импульсов В между двумя последовательно расположенными импульсами V было нечетным, и, во-вторых, чтобы полярность импульсов V чередовалась.

В коде В6ZS (рис. 8.2, *е*) каждые шесть последовательных нулей подменяются комбинацией 0VB0VB.

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

Каждые четыре последовательных нуля подменяются комбинацией 000V либо В00V. Выбор той или иной комбинации проводится так, чтобы, во-первых, число импульсов В между двумя последовательными импульсами V было нечетным, и, во-вторых, чтобы полярность импульсов V чередовалась (рис. 8.2, *ж*).

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

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

Как следует из ранее сказанного, для надежного восстановления синхросигнала приемником желательно так закодировать данные, чтобы сигнал в линии изменялся как можно чаще, в идеальном случае – в каждом битовом интервале. Одно из таких решений с использованием трехуровневого кодирования сигнала предложено в [64]. Между двумя проводами линии может присутствовать отрицательное, нулевое или положительное напряжение или $U = -1$, $U = 0$, $U = +1$. Данное решение интересно тем, что созданы гарантии изменения уровня сигнала при переходе от одного битового интервала к другому независимо от вида передаваемой последовательности битов, что подтверждается временной диаграммой рис. 8.3.

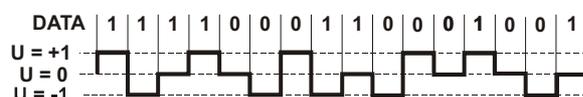


рис. 8.3. Временная диаграмма сигнала в линии

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

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

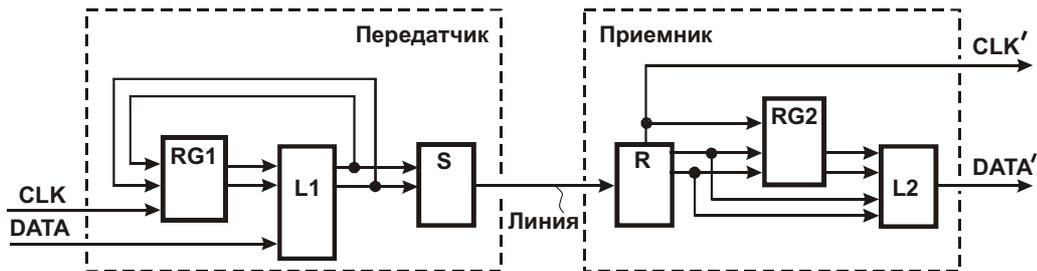


рис. 8.4. Система передачи данных

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

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

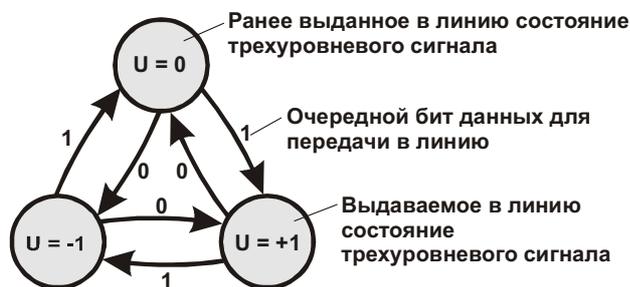


рис. 8.5. Диаграмма состояний передатчика

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

Стрелками обозначены переходы из одного состояния в другое. Цифра 0 или 1 около стрелки соответствует значению очередного бита DATA. Из рисунка следует, что при передаче цепочки битов 111...1 траектория переходов по диаграмме соответствует

движению по часовой стрелке, а при передаче цепочки 000...0 – движению в обратном направлении. Передача случайных данных сопровождается “блужданием” между тремя состояниями. Существенно, что не бывает ситуаций, при которых одно и то же состояние повторяется в соседних тактах.

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

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

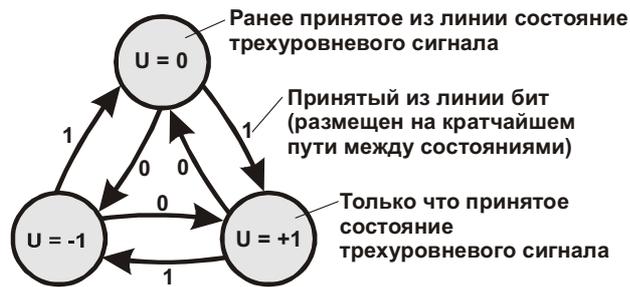


рис. 8.6. Диаграмма состояний приемника

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

Особенность этой схемы кодирования – декодирования состоит в том, что при передаче цепочки битов вида 010101... все импульсы будут иметь одинаковую полярность, зависящую от предыстории. Это означает, что в сигнале появится постоянная составляющая, что для многих систем недопустимо. Чтобы избежать этого, можно применить скремблирование данных на входе передатчика и их дескремблирование на выходе приемника. Напомним, что применение этих операций позволяет получить псевдослучайный поток битов, в котором устранены нежелательные закономерности их чередования (см. п. 8.4).

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

8.3.1. Скремблирование полярностей импульсов

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

Энергия передаваемого по линии сигнала сосредоточена в некоторой спектральной полосе. Для уменьшения влияния на соседние линии желательно как можно более равномерно распределить энергию в этой полосе, без выраженных спектральных пиков. Если это условие выполнено, то источник сигнала можно грубо представить в виде

бесконечно большого числа генераторов разной частоты, причем каждый генератор имеет бесконечно малую мощность. Результирующий сигнал помехи имеет характер шума.

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

Эти компоненты могут появляться, например, в сигналах АМІ, T1 или MLT-3 при передаче длинной последовательности лог. 1, как показано затененными областями на рис. 8.7.

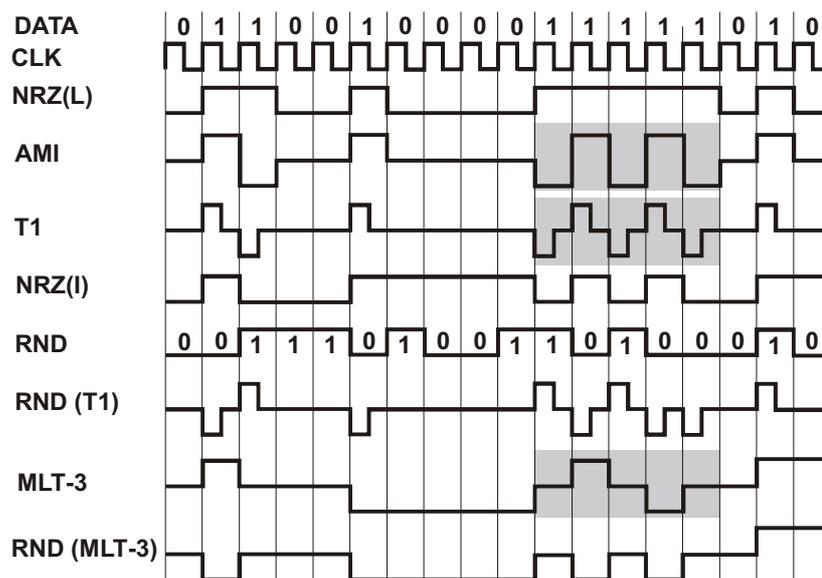


Рис. 8.7. Временные диаграммы передачи данных DATA с использованием различных кодов; RND – сигнал на выходе генератора псевдослучайной последовательности битов

В этих областях невооруженным глазом просматриваются прообразы синусоидальных сигналов, несущих основную энергию. Периоды сигналов АМІ и T1 при передаче длинной последовательности лог. 1 равны двум битовым интервалам. Период сигнала MLT-3 равен четырем битовым интервалам.

Длинные последовательности лог. 1 можно “разрушить” применением скремблирования, т. е. особой шифрации данных, после которой любые исходные последовательности выглядят как случайные (см. п. 8.4). Для восстановления исходных данных приемник должен выполнить обратную операцию (дескремблирование). При этом необходима синхронная работа шифратора и дешифратора, что несколько усложняет задачу.

Предлагаемое в [26] решение также предусматривает разрушение периодического сигнала при передаче длинной последовательности лог. 1, но выполняется оно иначе. Скремблируются не данные, а полярности передаваемых по линии импульсов. В зависимости от значения некоторого псевдослучайного бита выбирается либо положительная, либо отрицательная полярность. Приемник безразличен к полярности импульса и реагирует только на его наличие. Поэтому для восстановления данных приемнику не нужно знать вид псевдослучайной последовательности, использованной при шифрации полярностей! Иными словами, осуществляется некое “скремблирование без последующего дескремблирования” (что на первый взгляд представляется лишенным

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

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

Как уже отмечалось в п. 8.1, код NRZ (в данном случае он обозначен как NRZ(L)) отображает лог. 0 и лог. 1 соответственно низким и высоким уровнями напряжения. В коде AMI лог. 0 отображается отсутствием напряжения, а лог. 1 – положительным или отрицательным импульсом, причем полярности соседних импульсов чередуются. Код T1 отличается от AMI длительностью импульса.

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

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

Схема преобразования кода NRZ(L) в коды NRZ(I) и MLT-3 приведена на рис. 8.8, а. Каждый из двух последовательно соединенных D-триггеров включен в режиме делителя частоты. На выходе Q первого триггера формируется код NRZ(I). На входы передатчика подаются сигналы “+” и “-”, которые преобразуются соответственно в положительные и отрицательные импульсы трехуровневого сигнала MLT-3.

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

Схема, показанная на рис. 8.8, б, отличается от предыдущей тем, что на D-вход второго триггера (первый триггер не показан) подается псевдослучайная последовательность битов RND. При RND = 1 в момент формирования положительного фронта сигнала NRZ(I) выбирается положительная полярность импульса в линии, при RND = 0 – отрицательная. Последовательность битов RND синхронизирована сигналом CLK и формируется, например, генератором на основе сдвигового регистра с логическими элементами Исключающее ИЛИ в цепях обратных связей [65]. Такое решение приводит к случайному чередованию полярностей импульсов кода RND(MLT-3) в отличие от их регулярного чередования в коде MLT-3. Схема формирования сигнала RND(T1), показанная на рис. 8.8, в, построена аналогично и отличается наличием дополнительного логического элемента И, предназначенного для укорочения положительных импульсов кода NRZ(I).

Схема, представленная на рис. 8.8, г, позволяет дешифровать коды MLT-3 или RND(MLT-3), т. е. преобразовывать их в обычный код NRZ(L). На выходе приемника формируются положительные импульсы “+” и “-”, которые соответствуют разнополярным входным сигналам. Приемник также формирует синхросигнал CLK, например, с помощью генератора с фазовой автоподстройкой частоты (см. гл. 9).

Логический элемент ИЛИ суммирует импульсы “+” и “-”, так что их первоначальная полярность не учитывается. В этом, пожалуй, и заключена основная предпосылка создания рассмотренного решения: полярность импульсов в линии может быть произвольной, так как приемник не обращает на нее внимания. А если это так, то можно случайным образом распределить полярности передаваемых импульсов и тем самым пода-

вить периодические составляющие сигнала. Единственное ограничение состоит в том, что для исключения постоянной составляющей сигнала в линии среднее число положительных и отрицательных импульсов в любом достаточно большом интервале времени должно быть одинаковым. Это условие в данном случае выполнено.

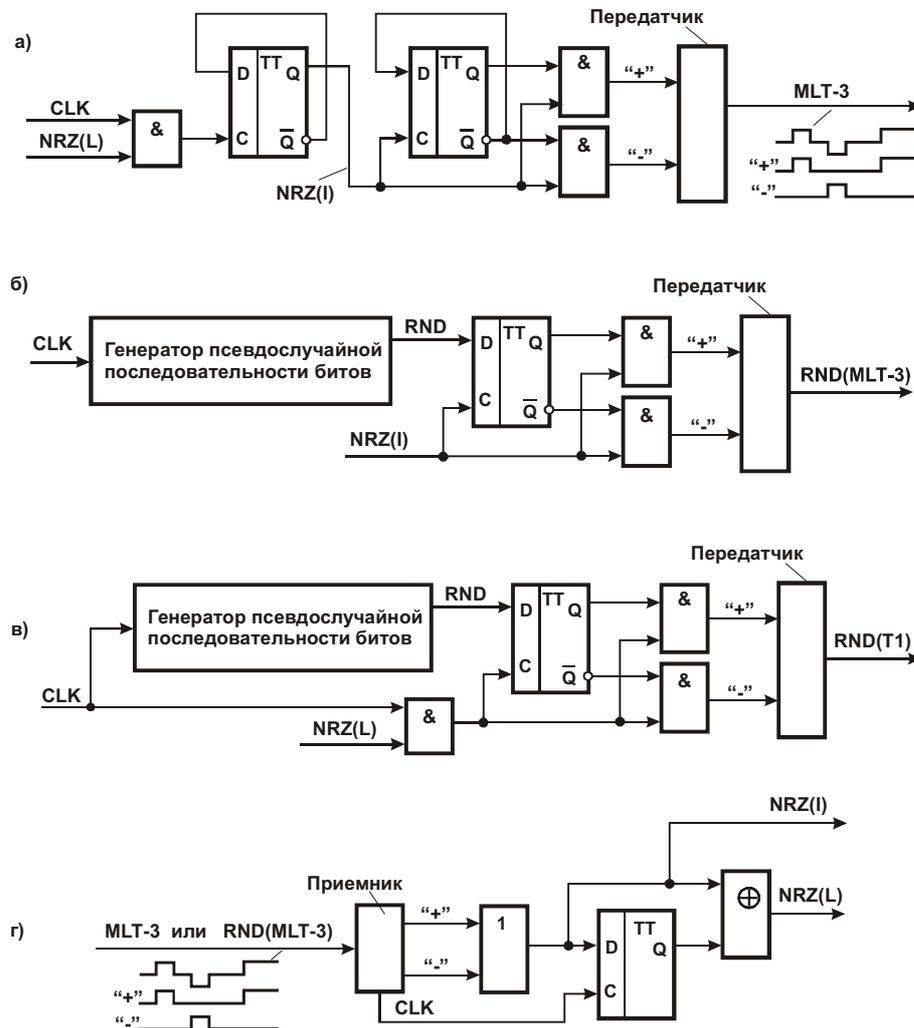


Рис. 8.8. Упрощенные схемные решения: *а* – формирователь кодов NRZ(I), MLT-3; *б* – формирователь кода RND(MLT-3) с псевдослучайным чередованием полярностей импульсов; *в* – формирователь кода RND(T1) с псевдослучайным чередованием полярностей импульсов; *г* – дешифратор кода MLT-3 или RND(MLT-3)

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

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

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

8.3.2. Двубинарное кодирование

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

В схеме, показанной на рис. 8.9, потребитель данных находится на некотором удалении от оптоволоконной линии связи. Для приема данных потребителю выделена витая пара проводов в многожильном кабеле (рассматриваем только одно направление

передачи). На выходе интерфейса FDDI (Fiber Distributed Data Interface – распределенный интерфейс передачи данных по волоконно-оптическим каналам) данные представлены кодом NRZ(I) и сопровождающим его синхросигналом CLK (см. рис. 8.7).

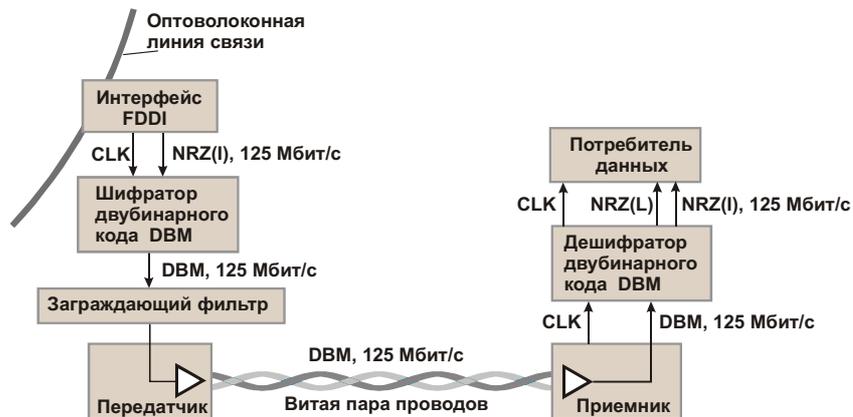


рис. 8.9. Схема высокоскоростной передачи данных в двоичном коде с использованием витой пары проводов

Проблема заключается в том, что непосредственная передача сигнала NRZ(I) со скоростью 125 Мбит/с по витой паре проводов создает повышенный уровень помех на соседних жилах кабеля. Ситуация усугубляется в отсутствие полезных данных, когда передается заполняющая паузу непрерывная последовательность лог. 1. Эта последовательность соответствует частоте сигнала NRZ(I), равной половине скорости передачи данных или 62,5 МГц. На этой частоте сигнал легко преодолевает паразитные емкостные и индуктивные связи и наводится на соседние провода кабеля. Поэтому следовало бы применить какой-либо дополнительный способ кодирования для снижения частоты сигнала в отсутствие данных и разравнивания его спектра при наличии данных. Рассмотренное далее трехуровневое двоичное кодирование DBM (duobinary modulation) и включение заграждающего фильтра позволяют в значительной мере снизить уровень излучаемых помех. По способу построения код DBM во многом схож с описанными в п. 8.3.1 кодами MLT-3 и RND(MLT-3).

Как показано на рис. 8.9, код NRZ(I) с выхода интерфейса FDDI преобразуется шифратором в код DBM. Сигнал с выхода шифратора проходит через заграждающий R-L-C-фильтр, разравнивающий спектр сигнала, передатчик и по линии связи (витой паре проводов) поступает в приемник. Приемник выделяет из него синхросигнал CLK и данные, представленные в коде DBM. Дешифратор кода DBM формирует коды NRZ(I) и NRZ(L). Скорость передачи данных во всем тракте постоянна и равна 125 Мбит/с.

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

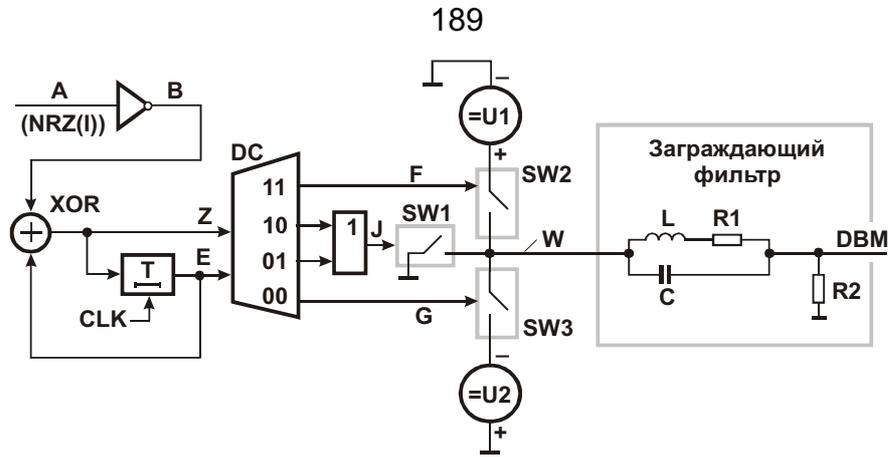


рис. 8.10. Схема шифратора двубинарного кода DBM и структура заграждающего фильтра

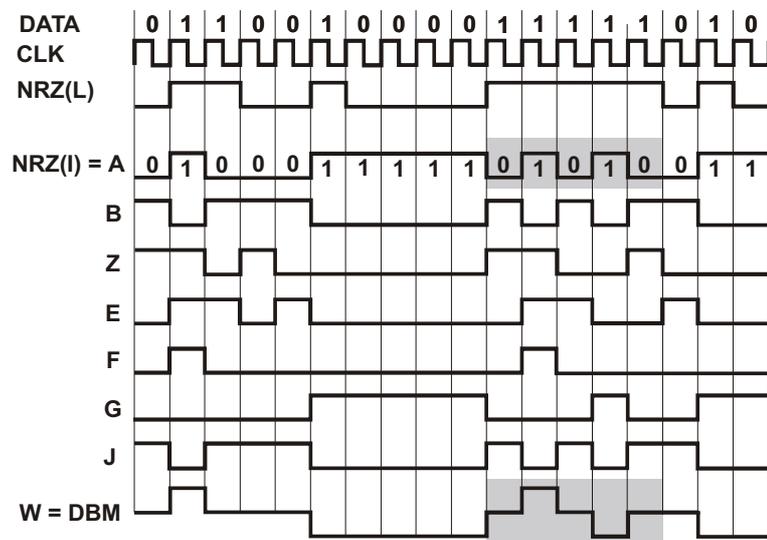


рис. 8.11. Временные диаграммы формирования двубинарного кода DBM

Входной сигнал A инвертируется и поступает на первый вход элемента XOR. Сигнал Z с выхода этого элемента задерживается на один период сигнала CLK (например, с помощью D-триггера) и подается на второй вход элемента XOR. Дешифратор DC в зависимости от сочетания сигналов Z и E формирует сигнал на одном из четырех выходов. При $Z = E = 0$ сигнал $G = 1$ замыкает ключ $SW3$, поэтому на выход W шифратора поступает отрицательное напряжение от источника $U2$. При $Z \neq E$ сигнал $J = 1$ замыкает ключ $SW1$, на выход шифратора поступает нулевое напряжение. При $Z = E = 1$ сигнал $F = 1$ замыкает ключ $SW2$, на выход шифратора поступает положительное напряжение от источника $U1$.

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

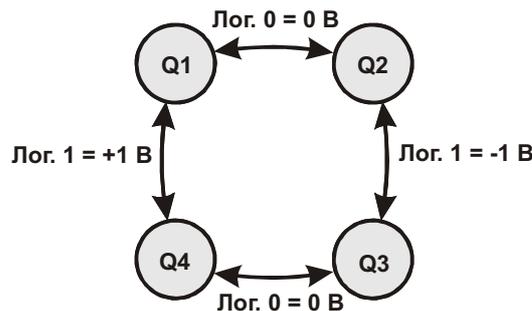


рис. 8.12. Диаграмма состояний шифратора двубинарного кода DBM

Шифратор может находиться в одном из четырех состояний $Q1 - Q4$. Если, например, шифратор пребывает в состоянии $Q1$, то при поступлении на вход A сигнала лог. 1 на его выходе W формируется положительное напряжение $+1$ В (величина условная). Этот факт отражен обозначением “Лог. 1 = +1 В” около двунаправленной связи между узлами $Q1$ и $Q4$. В этой ситуации шифратор переходит в состояние $Q4$.

Если шифратор находится в состоянии $Q1$, то при поступлении на вход A сигнала лог. 0 на его выходе W формируется нулевое напряжение 0 В. Этот факт отражен обозначением “Лог. 0 = 0 В” около двунаправленной связи между узлами $Q1$ и $Q2$. В данной ситуации шифратор переходит в состояние $Q2$. Переходы между состояниями $Q2$ и $Q3$ возможны при поступлении на вход A сигналов лог. 1, но эти переходы сопровождаются выдачей отрицательного напряжения (-1 В) на выход W . Переходы между состояниями $Q3$ и $Q4$ возможны при поступлении на вход A шифратора сигналов лог. 0.

Из диаграммы состояний следует, что если на вход A подана последовательность лог. 0, то шифратор последовательно переходит из состояния $Q1$ в состояние $Q2$ и обратно либо из состояния $Q3$ в состояние $Q4$ и обратно. Эти ситуации внешне неразличимы, так как на выходе шифратора в любом случае сформировано нулевое напряжение. Если на вход A подана последовательность лог. 1, то шифратор последовательно переходит из состояния $Q1$ в состояние $Q4$ и обратно либо из состояния $Q2$ в состояние $Q3$ и обратно. Эти ситуации различаются полярностью выходного напряжения.

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

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

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

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

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

В соблюдении приведенных правил можно убедиться при сопоставлении временных диаграмм сигналов A и W на рис. 8.11. Из этих диаграмм также следует, что при передаче непрерывной последовательности лог.1 ($DATA = 11\dots 1$) частота основной гармоники сигнала NRZ(I) равна половине скорости передачи данных или 62,5 МГц. При этих же условиях частота основной гармоники сигнала DBM равна четверти скорости передачи данных или 31,25 МГц. (Интересующие нас области временных диаграмм выделены серым фоном.) Амплитуда этой гармоники достаточно высока по сравнению с остальными, поэтому без заметного искажения формы сигнала ее можно несколько снизить с помощью заграждающего фильтра.

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

Двубинарное кодирование с фильтрацией выходного сигнала позволяет сместить его энергетический спектр в область более низких частот по сравнению с другими ре-

шениями. Так, 78 % энергии сигнала сосредоточено в полосе частот ниже 30 МГц, а 90 % энергии – в полосе частот ниже 42,6 МГц. Напомним, что скорость передачи данных составляет 125 Мбит/с!

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

8.4. Передача данных с использованием скремблера и дескремблера

Скремблирование может выполняться с различными целями. Наиболее распространенная цель – защита передаваемых данных от несанкционированного доступа. Для ее достижения разработано множество методов кодирования и схемных решений. Но нас интересует иная задача, связанная с “разравниванием” спектра сигнала и повышением надежности синхронизации приемника с источником передаваемых по линии данных. Применительно к этой задаче цель скремблирования состоит в исключении из потока данных длинных последовательностей лог. 0, лог. 1 и периодически повторяющихся групп битов. Для этого необходимо преобразовать данные так, чтобы они выглядели как случайные, т. е. лишены какой-либо видимой закономерности.

8.4.1. Генераторы псевдослучайных битовых последовательностей

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

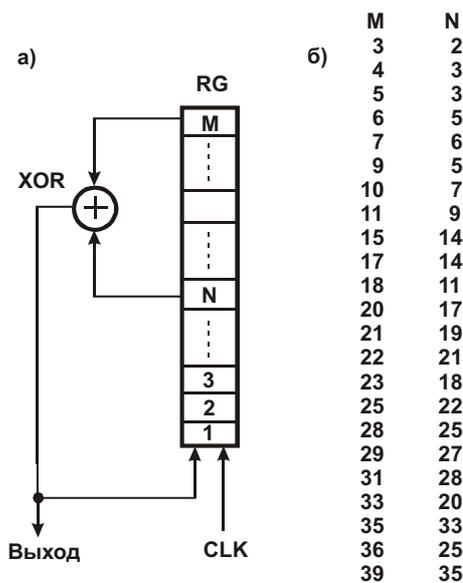


Рис. 8.13. Генератор псевдослучайной битовой последовательности максимальной длины: а – схема; б – таблица для выбора промежуточной точки подключения обратной связи

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

Последовательность максимальной длины обладает следующими свойствами.

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

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

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

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

8.4.2. Скремблер и дескремблер с неизолрованными генераторами псевдослучайных битовых последовательностей

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

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

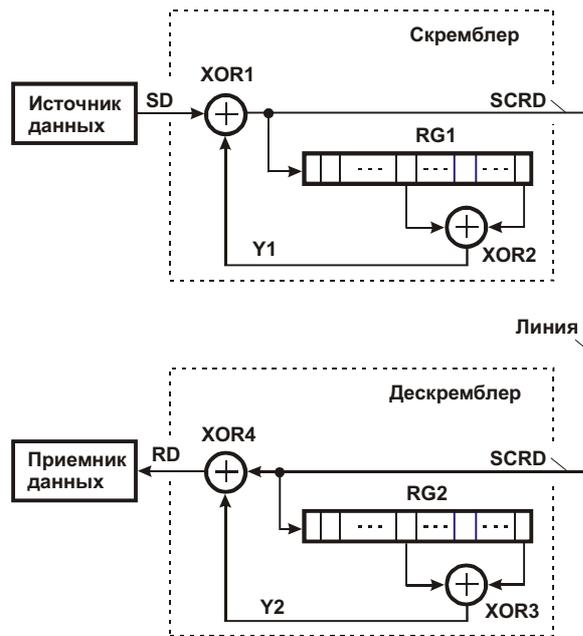


рис. 8.14. Система передачи данных, в которой скремблер и дескремблер содержат неизолированные генераторы псевдослучайных битовых последовательностей

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

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

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

8.4.3. Скремблер и дескремблер с изолированными генераторами псевдослучайных битовых последовательностей

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

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

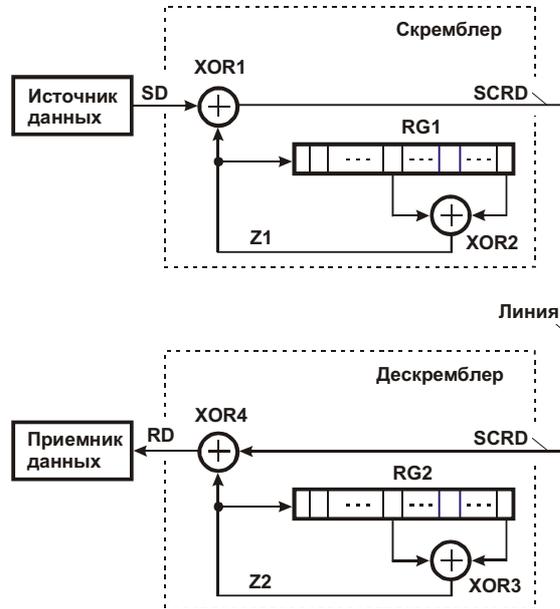


Рис. 8.15. Система передачи данных, в которой скремблер и дескремблер содержат изолированные генераторы псевдослучайных битовых последовательностей

8.4.4. Скремблер и дескремблер с неизолрованными генераторами – улучшенный вариант

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

Скремблер содержит сдвиговый регистр $RG1$ с логическими элементами Иключающее ИЛИ ($XOR1$ и $XOR2$) в цепи обратной связи, а также два двоичных счетчика.

Счетчик лог. 0 устанавливается в нуль всякий раз, когда скремблированный сигнал данных $SCRD = 1$. Если $SCRD = 0$, то содержимое счетчика увеличивается на единицу по фронту сигнала $CLK1$. При накоплении заданного числа единиц (например пяти) счетчик автоматически устанавливается в нулевое состояние и формирует импульс SET установки в единицу некоторого разряда (или группы разрядов) сдвигового регистра. Таким образом, счетчик лог. 0 служит детектором цепочек лог. 0 заданной длины. При обнаружении такой цепочки корректируется код в сдвиговом регистре.

Счетчик лог. 1 построен симметрично. Он устанавливается в нуль всякий раз, когда скремблированный сигнал данных $SCRD = 0$. Если $SCRD = 1$, то содержимое счетчика увеличивается на единицу по фронту сигнала $CLK1$. При накоплении заданного числа единиц (например пяти) счетчик автоматически устанавливается в нулевое состояние и формирует импульс $RESET$ установки в нуль некоторого разряда (или груп-

пы разрядов) сдвигового регистра. Счетчик лог. 1 служит детектором цепочек лог. 1 заданной длины.

Дескремблер построен аналогично. Он дополнительно содержит схему выделения синхросигнала CLK2 из скремблированного сигнала SCRD. Эта схема может быть выполнена на основе петли фазовой автоподстройки частоты PLL (Phase Locked Loop).

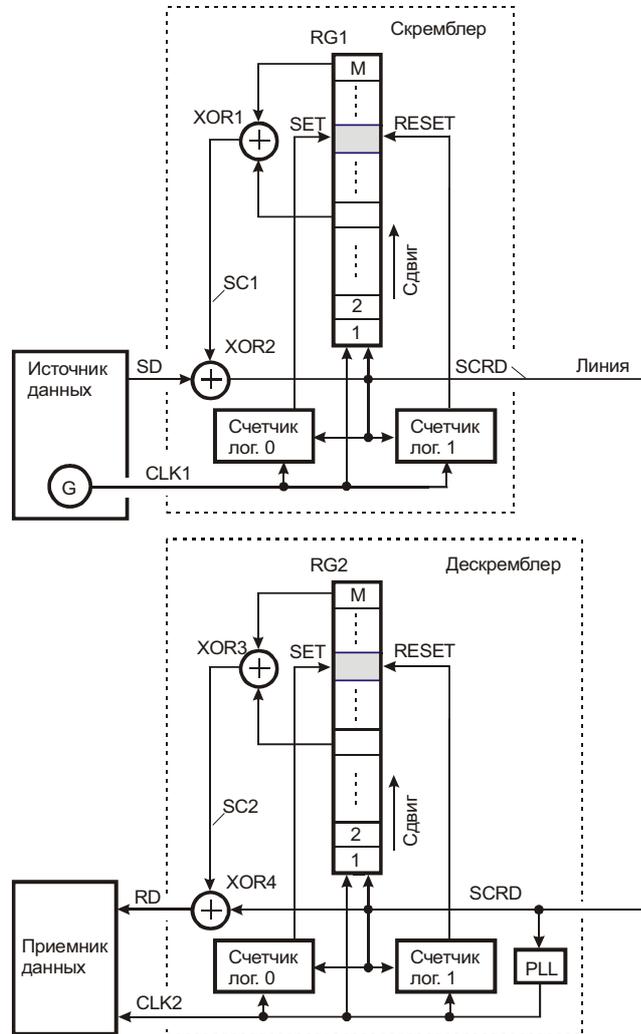


рис. 8.16. Система передачи данных, в которой скремблер и дескремблер содержат неизолированные генераторы псевдослучайных битовых последовательностей (улучшенный вариант)

Система передачи данных функционирует следующим образом. Источник данных формирует синхронный битовый поток SD и соответствующий синхросигнал CLK1. Этот поток проходит через логический элемент XOR2. На второй вход этого элемента поступает последовательность скремблирующих битов SC1. Суммарный (скремблированный) поток SCRD передается по линии и поступает в дескремблер.

После заполнения регистра RG2 информация в нем в точности совпадает с той, которая присутствует в регистре RG1. В дальнейшем все изменения информации в этих регистрах происходят синхронно, так как на их входы подается один и тот же сигнал SCRD (разумеется, с учетом задержки передачи по линии связи). Благодаря этому, $SC2 = SC1$. Логический элемент XOR4 формирует сигнал принимаемых данных RD, который повторяет исходный сигнал SD. Это следует из того, что

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

Уточним роль счетчиков лог. 0 и лог. 1, о которых уже кратко упоминалось. Предположим, что эти счетчики исключены из схем скремблера и дескремблера. Схема остается работоспособной при условии, что поток SD не содержит некоторых опасных последовательностей сигналов. Рассмотрим эти последовательности.

При работе системы не исключено, что поступающие от источника данные SD таковы, что логический элемент XOR2 скремблера в M последовательных тактах сформирует сигнал лог. 0 (M – разрядность сдвигового регистра). Тогда сдвиговый регистр RG1 (а синхронно с ним и регистр RG2) заполнится нулевыми битами. Если после этого источник сигнала начнет передавать длинную последовательность лог. 0, то на обоих входах логического элемента XOR2 будут постоянно присутствовать нулевые сигналы, сигнал SCRD также в течение длительного времени будет оставаться нулевым, что крайне нежелательно.

Аналогичная ситуация возможна и после случайного заполнения сдвигового регистра единичными битами. При последующей передаче длинной последовательности сигналов SD = 1 на выходе логического элемента XOR2 поддерживается сигнал лог. 1, который в каждом такте записывается в регистр, подтверждая его состояние “Все единицы”.

Введение счетчиков позволяет исключить возможность заполнения регистра RG1 одинаковыми битами (лог. 0 или лог. 1). Поэтому нет опасности фиксации уровня сигнала в линии при последующей выдаче источником данных длинной последовательности лог. 0 или лог. 1. Но это, к сожалению, не означает, что задача получения гарантированно изменяющегося сигнала SCRD решена “полностью и окончательно”. Действительно, теоретически можно преднамеренно синтезировать сколь угодно длинную последовательность сигналов SD, совпадающую или противофазную последовательности сигналов SC1, какой бы сложной она ни была (ведь ее можно заранее вычислить, зная структуру скремблера и его начальное состояние). В результате такого синтеза получим неизменный сигнал SCRD на протяжении любого желаемого интервала времени! Точно так же можно было бы синтезировать периодический сигнал SCRD вида 010101... для создания максимального уровня перекрестных помех в соседних проводах многожильного кабеля (например с целью тестирования системы). Но так как начальное состояние регистра RG1 источнику данных не известно, на практике такой синтез невозможен.

Вероятность случайного формирования нескремблируемых последовательностей битов источником данных зависит от разрядности скремблера и может быть небольшой, но с ней нельзя не считаться при проектировании телекоммуникационных устройств.

8.4.5. Синхронизация изолированных генераторов скремблера и дескремблера

В системе передачи данных, показанной на рис. 8.17, применены изолированные генераторы псевдослучайных битовых последовательностей. Их синхронизация осуществляется с использованием аппаратных и программных средств приемной стороны.

К этим средствам относятся мультиплексор MUX и программно-управляемый выход приемника данных, на котором формируется сигнал F. При нормальной работе системы приемник данных постоянно поддерживает на выходе сигнал $F = 0$. На выход мультиплексора транслируется сигнал Z2, генератор псевдослучайной битовой последовательности на основе регистра RG2 изолирован от внешних воздействий. Поэтому схема эквивалентна рассмотренной ранее (см. рис. 8.15).

Предположим теперь, что в исходном состоянии дескремблер не синхронизирован со скремблером. Такая ситуация может возникнуть, например, после включения напряжения питания аппаратуры приемной стороны, после сбоя тактового генератора де-

скремблера из-за воздействия помех на линию связи или по иным причинам. В отсутствие синхронизации между скремблером и дескремблером содержимое регистров $RG1$ и $RG2$ не совпадает, поток принимаемых данных RD ошибочен и не совпадает с потоком передаваемых данных SD .

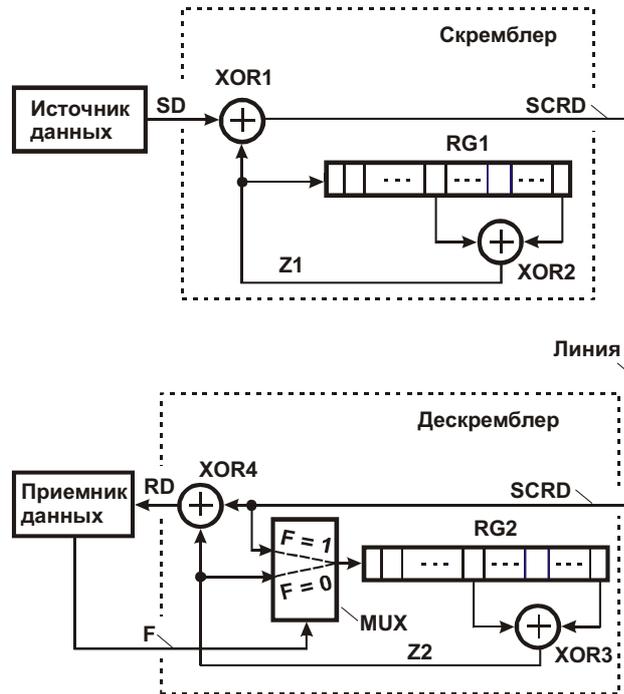


Рис. 8.17. Система передачи данных, в которой скремблер и дескремблер содержат изолированные генераторы псевдослучайных битовых последовательностей. Введены средства синхронизации этих генераторов

При обнаружении устойчивого хаотического потока данных RD (в котором нет обусловленного протоколом разделения на кадры и т. п.), приемник формирует сигнал $F = 1$. Вследствие этого на вход регистра $RG2$ транслируется сигнал скремблированных данных $SCRД$, как в схеме на рис. 8.14.

Протокол обмена предусматривает пересылку данных в виде последовательности кадров. Группы обычных кадров перемежаются со служебными кадрами. Например, после группы из 63 обычных кадров следует один служебный. Он, в частности, содержит синхронизирующую последовательность из нулевых битов. При выдаче этих битов ($SD = 0$) в скремблер элемент $XOR1$ выполняет функцию повторителя сигнала $Z1$. Поэтому в данном случае скремблированный сигнал $SCRД$ представляет собой фрагмент “истинной” псевдослучайной битовой последовательности, в том смысле, что она не смешана с потоком произвольных данных и порождается только генератором скремблера.

Эта последовательность загружается в регистр $RG2$, так как $F = 1$. После того как содержимое регистров $RG1$ и $RG2$ оказывается одинаковым, сигнал $Z2$ начинает повторять сигнал $Z1$. Синхронизация достигнута. После заполнения регистра $RG2$ на вход приемника данных подается непрерывная последовательность лог. 0, так как $RD = SD$. После уверенного обнаружения последовательности лог. 0 приемник формирует сигнал $F = 0$ и тем самым переключает генератор дескремблера в режим изолированной работы. Теперь синхронизация не только достигнута, но и “сохранена”. Для гарантии окончания процесса установления синхронизации источник данных еще некоторое время продолжает выдачу последовательности лог. 0, а затем приступает к передаче данных согласно принятому в системе протоколу.

8.4.6. Применение системы скремблер – дескремблер для тестирования канала связи

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

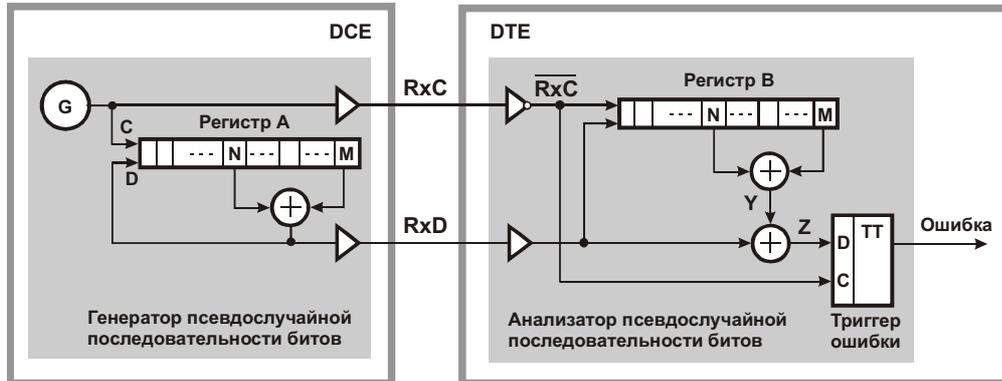


рис. 8.18. Система контроля передачи данных и синхросигнала между устройствами типа DCE и DTE

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

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

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

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

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

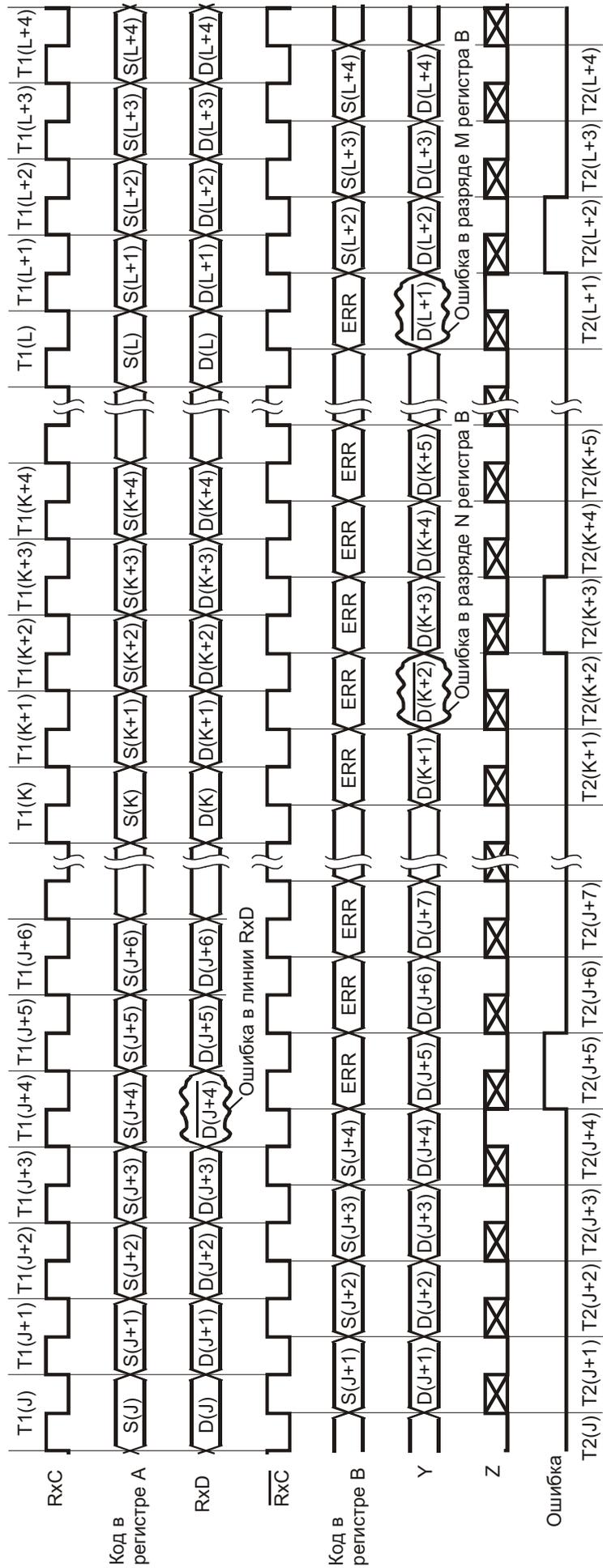


рис. 8.19. Временные диаграммы передачи и приема тестовых битовых последовательностей

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

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

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

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

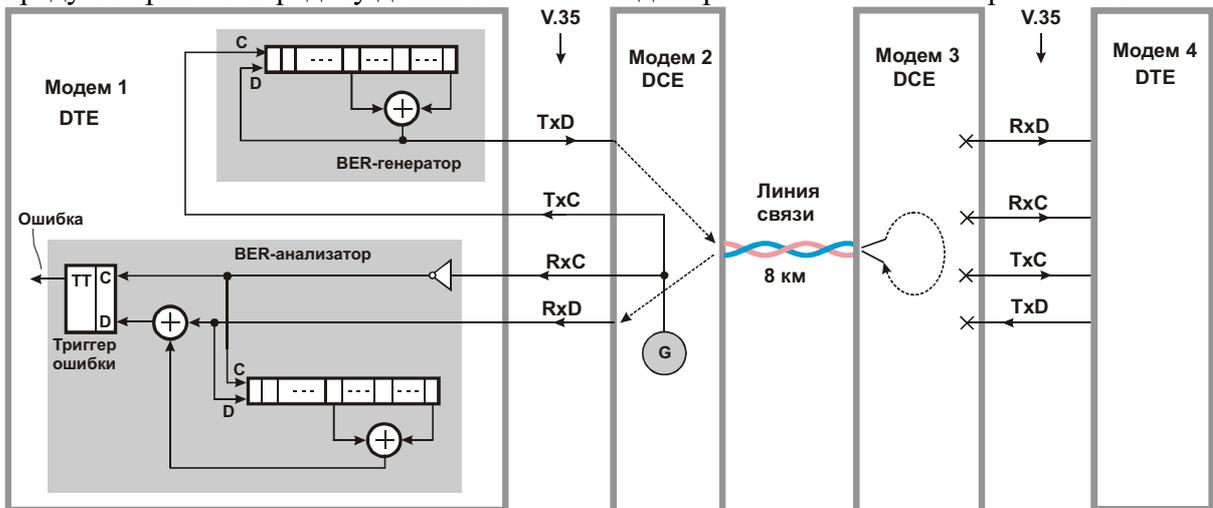


рис. 8.20. Включение BER-тестера в телекоммуникационную систему

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

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

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

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

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

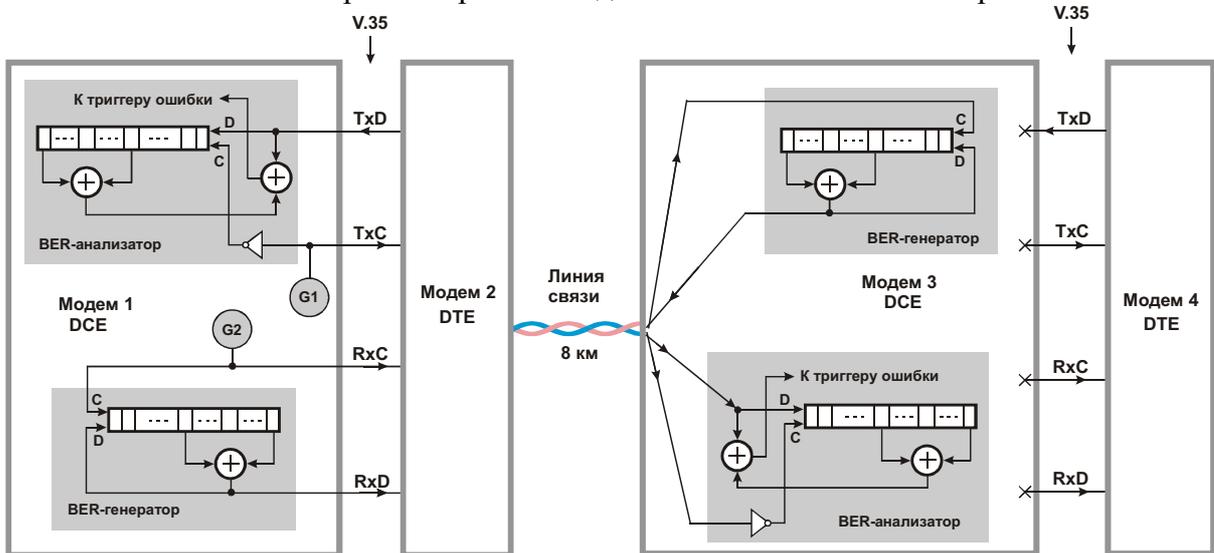


рис. 8.21. Включение двух BER-тестеров в телекоммуникационную систему

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

У читателя могут возникнуть сомнения в том, что ВЕР-генератор третьего модема выдает в линию псевдослучайную тестовую последовательность битов со скоростью, задаваемой генератором G1. Действительно, данные в смеси с синхросигналом передаются по “верхнему” каналу справа налево, а генератор G1 находится слева. Каким образом синхросигнал от генератора G1 может распространяться по линии против течения потока данных? – Ответ на этот вопрос мы уже рассматривали, см. п. 3.5.2.

8.5. Кодирование данных и синхросигнала для параллельной передачи по группе витых пар проводов

Далее приведен ряд схемных решений, обеспечивающих передачу данных по группе витых пар проводов. Сначала рассмотрена основная схема, а затем предложены ее усовершенствованные варианты. Повышенная степень детализации предлагаемых схемных решений обусловлена тем, что они не описаны в литературе.

8.5.1. Схема кодирования – декодирования

Сигналы можно передавать по нескольким витым парам проводов без использования кодирования, выделив для каждого сигнала одну витую пару. В схеме, приведенной на рис. 8.22, трехразрядный параллельный код DIN и сопровождающий его синхросигнал CIN передаются по четырем витым парам проводов к удаленному абоненту. Выходной код DOUT и принятый синхросигнал COUT совпадают с входными с точностью до задержки передачи.

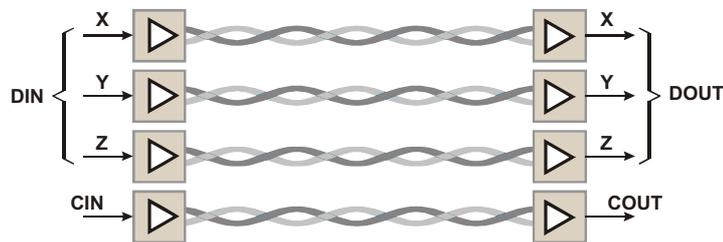


Рис. 8.22. Схема параллельной передачи данных и синхросигнала по витым парам проводов без применения кодирования

При увеличении разрядности кода DIN такое решение становится слишком неэкономичным. Для уменьшения числа витых пар проводов можно воспользоваться схемой, приведенной на рис. 8.23 [74].

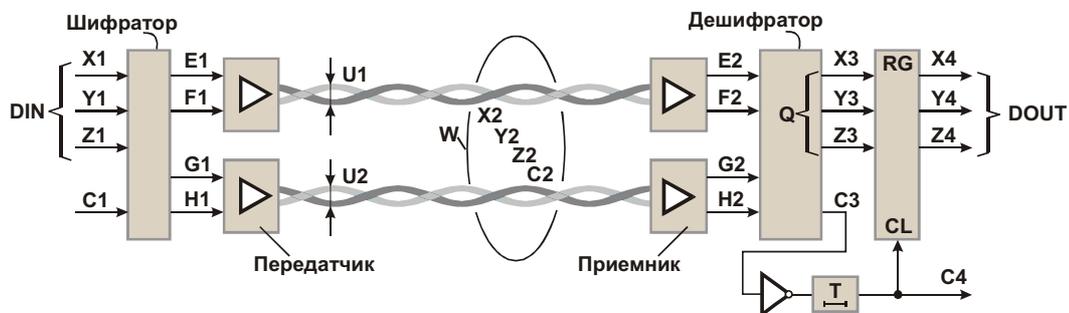


Рис. 8.23. Схема параллельной передачи кодированных данных и синхросигнала по витым парам проводов

В этой схеме применено трехуровневое кодирование сигнала в линии. Напряжение U_1 (U_2) между проводами витой пары может быть отрицательным, нулевым или положительным (сокращенно: $-$, 0 , $+$). Это напряжение формируется передатчиком в зависимости от сочетания битов $E_1 F_1$ ($G_1 H_1$) на его входах следующим образом. При $E_1 = F_1 = 0$ ($G_1 = H_1 = 0$) напряжение U_1 (U_2) нулевое, при $E_1 = 0$, $F_1 = 1$ ($G_1 = 0$, $H_1 = 1$) – отрицательное, при $E_1 = 1$, $F_1 = 0$ ($G_1 = 1$, $H_1 = 0$) – положительное. Код $E_1 F_1 G_1 H_1$ формируется шифратором из входного кода $X_1 Y_1 Z_1 C_1$ (данные, синхросигнал), как показано в левой части Таблица 8.1 табл. 8.1.

Таблица 8.1 табл. 8.1

Состояния шифратора, линии связи и дешифратора при параллельной передаче трех битов данных и синхросигнала

Коды на входах и выходах шифратора								Сигналы в линии		Коды на входах и выходах дешифратора							
C1	X1	Y1	Z1	E1	F1	G1	H1	U1	U2	H2	G2	F2	E2	Z3	Y3	X3	C3
1	×	×	×	0	0	0	0	0	0	0	0	0	0	×	×	×	1
0	0	0	0	0	0	0	1	0	–	1	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	+	0	1	0	0	1	0	0	0
0	0	1	0	0	1	0	0	–	0	0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1	–	–	1	0	1	0	1	1	0	0
0	1	0	0	0	1	1	0	–	+	0	1	1	0	0	0	1	0
0	1	0	1	1	0	0	0	+	0	0	0	0	1	1	0	1	0
0	1	1	0	1	0	0	1	+	–	1	0	0	1	0	1	1	0
0	1	1	1	1	0	1	0	+	+	0	1	0	1	1	1	1	0

Примечание. Символ “ \times ” соответствует произвольному значению бита

Приемники и дешифратор осуществляют обратное преобразование сигналов, представленных напряжениями U_1 и U_2 в линии. В этом можно убедиться сопоставлением левой и правой частей таблицы – они симметричны. В центральной части таблицы показаны девять возможных комбинаций сигналов в линии. Первая комбинация (0 0) выбрана для отображения единичного состояния синхросигнала. Символы “ \times ” в левой части таблицы означают, что при $C_1 = 1$ входные сигналы DIN не воспринимаются шифратором. Эти же символы в правой части таблицы показывают, что при выдаче синхросигнала C_3 выходные данные $X_3 Y_3 Z_3$ не определены и не могут использоваться для выдачи абоненту. Остальные восемь комбинаций сигналов в линии распределены для отображения восьми состояний трехразрядного кода $X_1 Y_1 Z_1$ при условии, что $C_1 = 0$. Шифратор и дешифратор могут быть выполнены на основе ПЗУ, программируемой логики или построены из обычных логических элементов.

Как следует из временных диаграмм, показанных на рис. 8.24, сигналы, передаваемые по линии, поочередно отображают синхросигнал и данные. Преобразами этих зашифрованных тремя уровнями напряжения сигналов (C_2 , X_2 , Y_2 , Z_2) в группе W являются сигналы на входах шифратора. Как уже отмечалось, группа выходных сигналов дешифратора не определена при наличии синхросигнала. Этот факт отмечен затемненными областями на временной диаграмме группы Q сигналов X_3 , Y_3 , Z_3 на выходе шифратора.

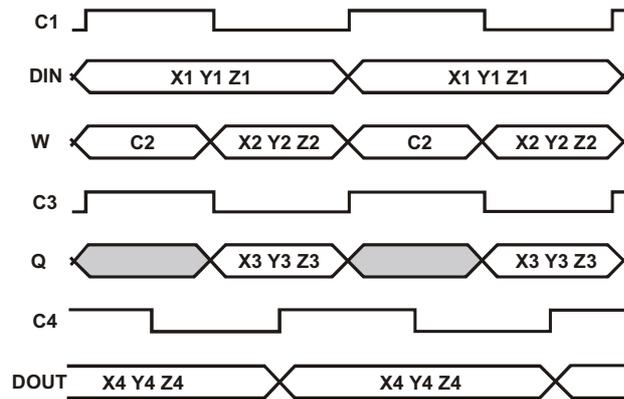


рис. 8.24. Временные диаграммы передачи данных и синхросигнала

Для устранения областей неопределенности этих сигналов применен параллельный регистр RG. Данные принимаются в него по положительному фронту сигнала C4, который формируется из сигнала C3 после его инвертирования и задержки на четверть периода. В результате сигналы DOUT и C4 с некоторой задержкой повторяют сигналы DIN и C1.

8.5.2. Усовершенствованная схема – первый вариант

Описанный способ уменьшения числа витых пар проводов линии применим и к схемам с большей разрядностью. В общем случае при трехуровневом кодировании число состояний линии, содержащей N витых пар проводов, равно 3^N . С помощью этих состояний можно передать M -разрядный двоичный код DIN (см. рис. 8.23), где $M = \lfloor \log_2 3^N \rfloor$, прямые скобки обозначают целую часть заключенного в них числа. Так, при $N = 2, 3, 4, \dots, 10$ разрядность M передаваемого двоичного кода составляет 3, 4, 6, 7, 9, 11, 12, 14, 15 (Таблица 8.2 табл. 8.2). Напомним, что одно из избыточных состояний линии в приведенном ранее примере использовалось для кодирования единичного состояния синхросигнала.

Таблица 8.2 табл. 8.2

Параметры перехода от троичного к двоичному представлению кодов

Число витых пар проводов N	Число состояний линии 3^N	Допустимая разрядность двоичного кода M	Число состояний двоичного кода 2^M	Избыточность троичного представления двоичного кода	
				абсолютная $3^N - 2^M$	относительная $((3^N - 2^M) / 3^N) \cdot 100\%$
2	9	3	8	1	11,1
3	27	4	16	11	40,7
4	81	6	64	17	21,0
5	243	7	128	115	47,3
6	729	9	512	217	29,8
7	2187	11	2048	139	6,3
8	6561	12	4096	2465	37,5
9	19683	14	16384	3299	16,8
10	59049	15	32768	26281	44,5

Из Таблица 8.2 табл. 8.2 следует, что при $N > 2$ в троичном представлении двоичного кода имеется избыточность, в основном, значительная. Ее можно использовать для повышения информационной нагрузки на линию связи при передаче потока данных.

Первый вариант усовершенствованной схемы предусматривает использование избыточных состояний линии для построения дополнительного канала связи. Телекоммуникационная система (рис. 8.25), в которую включена схема передачи данных, содержит четыре оконечных устройства DTE1 – DTE4, например четыре компьютера. В процессе работы системы данные передаются из устройства DTE1 в устройство DTE3 по основному (первому) каналу. Одновременно с этим по дополнительному (второму) каналу данные передаются из устройства DTE2 в устройство DTE4. (Для передачи данных в обратном направлении необходима вторая схема передачи данных, включенная встречно.)

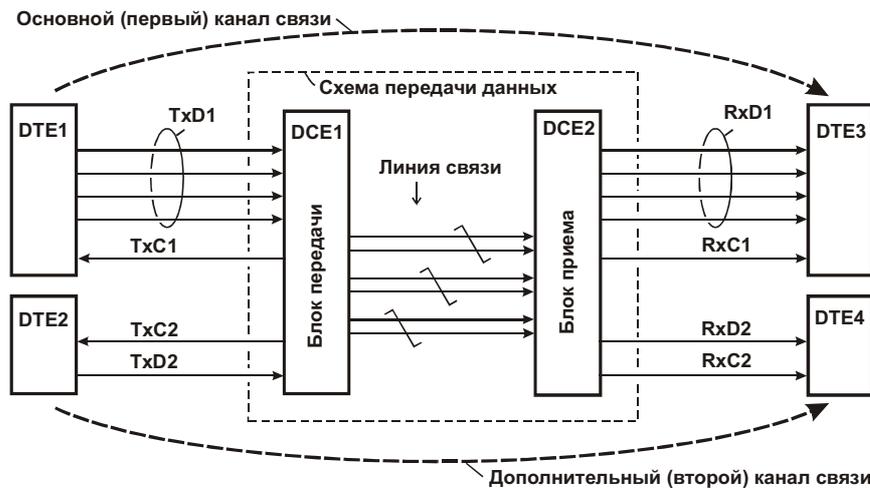


рис. 8.25. Пример включения усовершенствованной схемы передачи данных (первый, третий варианты) в телекоммуникационную систему

Сигнал TxC1 обеспечивает синхронное поступление данных TxD1 первого канала из устройства DTE1 в блок передачи данных DCE1. Сигнал TxC2 синхронизирует поступление данных TxD2 второго канала из устройства DTE2 в блок DCE1. Положительные фронты сигнала TxC1 задают границы битовых интервалов для каждого входного сигнала из группы TxD1. Положительные фронты сигнала TxC2 задают границы битовых интервалов сигнала TxD2. Отрицательные фронты сигналов TxC1 и TxC2 гарантируют истинность соответствующих данных TxD1 и TxD2. Аналогичные функции выполняют синхросигналы RxC1 и RxC2 сопровождения данных RxD1 и RxD2. Как будет показано, поток данных по первому каналу связи имеет постоянную скорость, в отличие от потока данных по второму каналу. Скорость передачи битов по второму каналу зависит от кодов в первом канале и при случайном равномерном распределении этих кодов составляет 62,5% скорости передачи четырехразрядных слов данных по первому каналу.

На рис. 8.26 – рис. 8.28 приведены схемы и временные диаграммы, поясняющие детали предлагаемого решения.

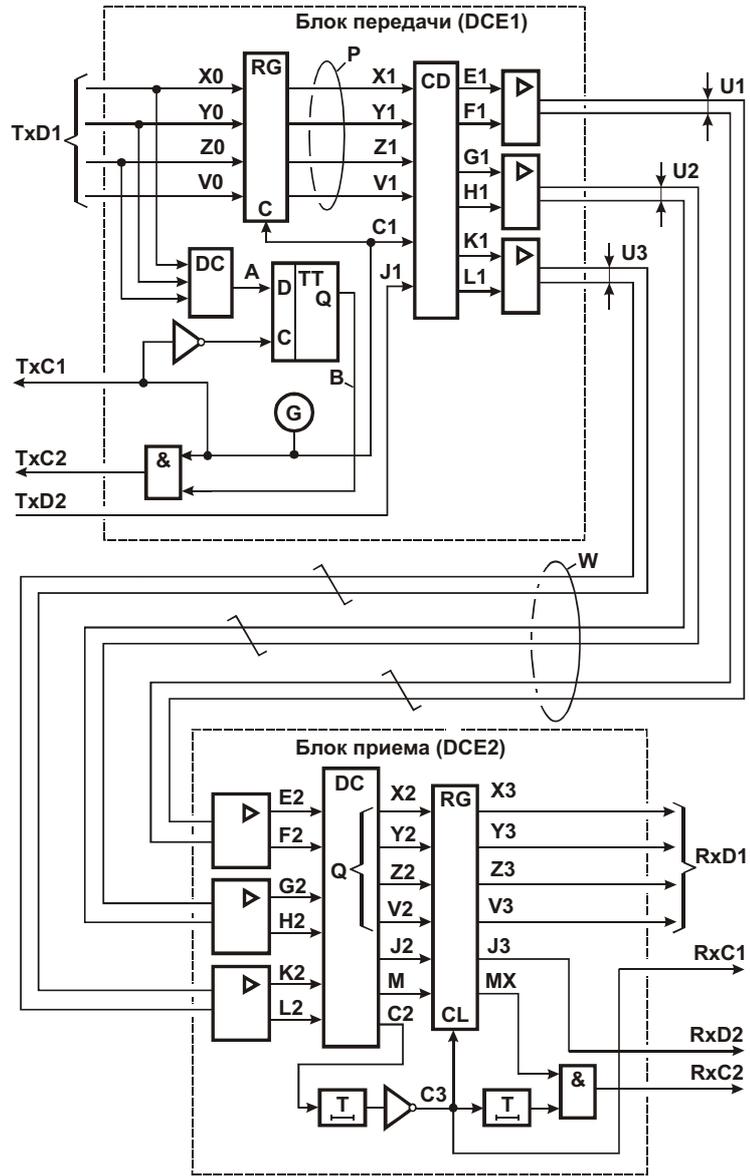


Рис. 8.26. Усовершенствованная схема параллельной передачи кодированных данных и синхросигнала по витым парам проводов (первый вариант)

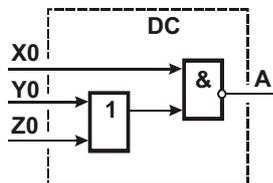


Рис. 8.27. Схема дешифратора блока DCE1 передачи данных

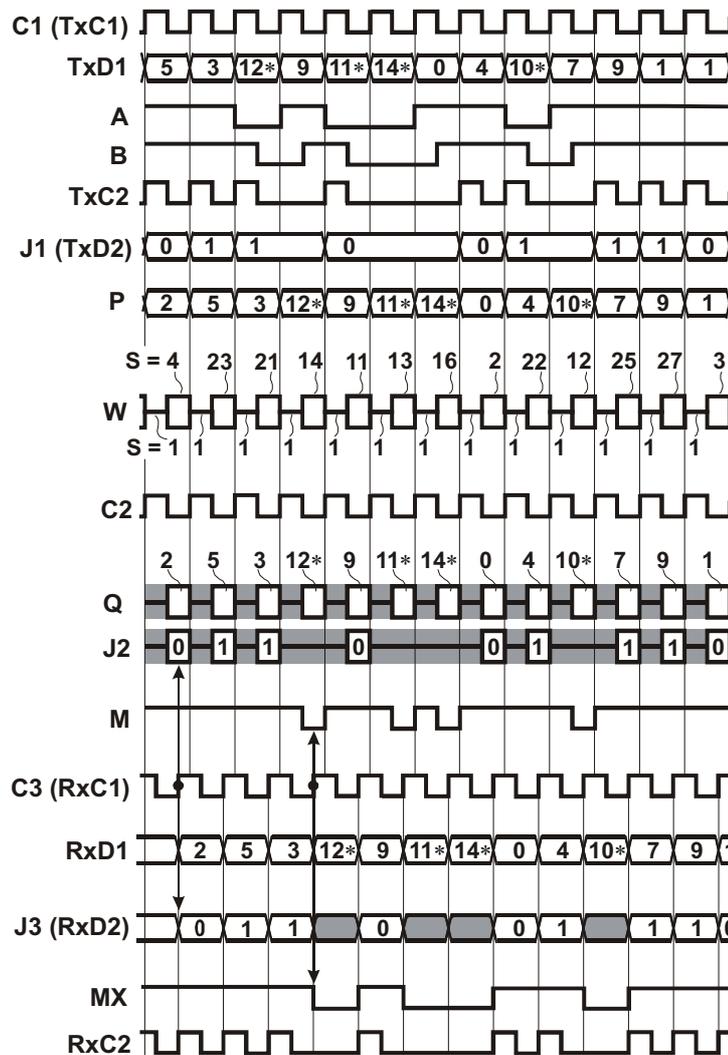


рис. 8.28. Временные диаграммы передачи данных

Все процессы, протекающие при передаче данных, синхронизируются от генератора G блока DCE1 (см. рис. 8.26). По положительному фронту сигнала TxC1 устройство DTE1 формирует очередной четырехразрядный двоичный код $X_0 Y_0 Z_0 V_0$ данных TxD1 (X_0 – старший разряд). В примере, приведенном на временных диаграммах (см. рис. 8.28), от устройства DTE1 под управлением сигнала TxC1 поступает такая последовательность кодов TxD1: $0101_2 = 5$, $0011_2 = 3$, $1100_2 = 12$, ..., $0001_2 = 1$. При этом коды, помеченные символами “*”, препятствуют распространению данных по дополнительному каналу в соответствующих битовых интервалах.

Регистр RG блока DCE1 принимает данные по положительному фронту сигнала TxC1, поэтому в этом регистре фиксируется “старый” код TxD1, прежде чем на его входы поступит “новый” код. (По аналогичной схеме построены обычные сдвиговые регистры с общей цепью синхронизации.) В результате данные P на выходах регистра формируются с задержкой на один такт по сравнению с данными TxD1 на его входах (см. диаграмму сигналов P, повторяющую диаграмму сигналов TxD1 со сдвигом вправо на один период сигнала TxC1).

Данные TxD1 анализируются дешифратором DC блока DCE1. Если данные принадлежат диапазону 0, 1, 2, ..., 9, то на выходе дешифратора формируется сигнал $A = 1$. Если данные принадлежат диапазону 10, 11, 12, ..., 15, то на выходе дешифратора формируется сигнал $A = 0$ (см. временную диаграмму этого сигнала). Как показано на рис. 8.27, дешифратор может быть выполнен на двух логических элементах. Сигнал A с выхода дешифратора принимается в триггер по положительному фронту ин-

вертированного сигнала $TxC1$. Сигнал $B = 0$ с выхода триггера закрывает элемент И по нижнему входу и препятствует формированию сигнала $TxC2$ в тех ситуациях, когда необходимо приостановить передачу данных по второму каналу. Поэтому данные $TxD2$ запрашиваются у устройства DTE2 и поступают от него с переменным темпом по мере возможности их передачи по второму каналу.

Сигналы P , $C1$ и $J1$ ($TxD2$) поступают на входы шифратора CD. Цепь, состоящая из шифратора, выходных и входных усилителей, обслуживающих линию связи, и дешифратора блока DCE2 выполняет преобразование кодов в соответствии с Таблица 8.3 табл. 8.3

Таблица 8.3 табл. 8.3
Состояния шифратора, линии связи и дешифратора при передаче данных и синхросигнала схемой, приведенной на рис. 8.2б

№ п/п	Коды на входах и выходах шифратора										Состояние линии		Коды на входах и выходах дешифратора																		
	C	J	X	Y	Z	V	P	E	F	G	H	K	L	S	U1-U3	L	K	H	G2	F	E	Q	V	Z	Y	X	J	C	M		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	000	0	0	0	0	0	0	0	×	×	×	×	×	×	1	1	
2	0	0	0	0	0	0	0	0	0	0	0	0	1	00-	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
3	0	0	0	0	0	1	1	0	0	0	0	1	0	00+	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
4	0	0	0	0	1	0	2	0	0	0	1	0	0	0-0	0	0	1	0	0	0	0	2	0	1	0	0	0	0	0	1	1
5	0	0	0	0	1	1	3	0	0	0	1	0	1	0--	1	0	1	0	0	0	0	3	1	1	0	0	0	0	0	1	1
6	0	0	0	1	0	0	4	0	0	0	1	1	0	0-+	0	1	1	0	0	0	0	4	0	0	1	0	0	0	0	1	1
7	0	0	0	1	0	1	5	0	0	1	0	0	0	0+0	0	0	0	1	0	0	0	5	1	0	1	0	0	0	0	1	1
8	0	0	0	1	1	0	6	0	0	1	0	0	1	0+-	1	0	0	1	0	0	0	6	0	1	1	0	0	0	0	1	1
9	0	0	0	1	1	1	7	0	0	1	0	1	0	0++	0	1	0	1	0	0	0	7	1	1	1	0	0	0	0	1	1
10	0	0	1	0	0	0	8	0	1	0	0	0	0	-00	0	0	0	0	0	1	0	8	0	0	0	1	0	0	0	1	1
11	0	0	1	0	0	1	9	0	1	0	0	0	1	-0-	1	0	0	0	0	1	0	9	1	0	0	1	0	0	0	1	1
12	0	×	1	0	1	0	10*	0	1	0	0	1	0	-0+	0	1	0	0	0	1	0	10*	0	1	0	1	×	0	0	0	0
13	0	×	1	0	1	1	11*	0	1	0	1	0	0	--0	0	0	1	0	0	1	0	11*	1	1	0	1	×	0	0	0	0
14	0	×	1	1	0	0	12*	0	1	0	1	0	1	---	1	0	1	0	0	1	0	12*	0	0	1	1	×	0	0	0	0
15	0	×	1	1	0	1	13*	0	1	0	1	1	0	--+	0	1	1	0	0	1	0	13*	1	0	1	1	×	0	0	0	0
16	0	×	1	1	1	0	14*	0	1	1	0	0	0	+00	0	0	0	1	1	0	0	14*	0	1	1	1	×	0	0	0	0
17	0	×	1	1	1	1	15*	0	1	1	0	0	1	-+-	1	0	0	1	1	0	0	15*	1	1	1	1	×	0	0	0	0
18	0	1	0	0	0	0	0	0	1	1	0	1	0	-++	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1	1
19	0	1	0	0	0	1	1	1	0	0	0	0	0	+00	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1	1
20	0	1	0	0	1	0	2	1	0	0	0	1	0	+0-	1	0	0	0	0	1	0	2	0	1	0	0	1	0	0	1	1
21	0	1	0	0	1	1	3	1	0	0	0	1	0	+0+	0	1	0	0	0	1	0	3	1	1	0	0	1	0	0	1	1
22	0	1	0	1	0	0	4	1	0	0	1	0	0	+0-	0	0	1	0	0	1	0	4	0	0	1	0	1	0	0	1	1
23	0	1	0	1	0	1	5	1	0	0	1	0	1	+--	1	0	1	0	0	1	0	5	1	0	1	0	1	0	0	1	1
24	0	1	0	1	1	0	6	1	0	0	1	1	0	+--	0	1	1	0	0	1	0	6	0	1	1	0	1	0	0	1	1
25	0	1	0	1	1	1	7	1	0	1	0	0	0	+00	0	0	0	1	0	1	0	7	1	1	1	0	1	0	0	1	1
26	0	1	1	0	0	0	8	1	0	1	0	0	1	++-	1	0	1	0	1	0	0	8	0	0	0	1	1	0	0	1	1
27	0	1	1	0	0	1	9	1	0	1	0	1	0	+++	0	1	0	1	1	0	0	9	1	0	0	1	1	0	0	1	1

Как и в предыдущей схеме (см. рис. 8.23), в линии связи применено трехуровневое кодирование сигналов. Напряжение U_1 (U_2 , U_3) между проводами витой пары может быть отрицательным, нулевым или положительным (сокращенно: $-$, 0 , $+$). Это напряжение формируется выходным усилителем блока DCE1 в зависимости от сочетания битов $E_1 F_1$ ($G_1 H_1$, $K_1 L_1$) на его входах следующим образом. При $E_1 = F_1 = 0$ ($G_1 = H_1 = 0$, $K_1 = L_1 = 0$) напряжение U_1 (U_2 , U_3) нулевое, при $E_1 = 0$, $F_1 = 1$ ($G_1 = 0$, $H_1 = 1$; $K_1 = 0$, $L_1 = 1$) – отрицательное, при $E_1 = 1$, $F_1 = 0$ ($G_1 = 1$, $H_1 = 0$; $K_1 = 1$, $L_1 = 0$) – положительное. Код $E_1 F_1 G_1 H_1 K_1 L_1$ формируется шифратором CD из входного кода $X_1 Y_1 Z_1 V_1 C_1 J_1$ (данные P , синхросигнал, данные TxD_2), как показано в левой части таблицы.

Входные усилители и дешифратор блока DCE2 выполняют обратное преобразование сигналов, представленных напряжениями U_1 , U_2 и U_3 в витых парах проводов линии связи. В графе S таблицы перечислены $3^3 = 27$ возможных состояний трехуровневых сигналов $U_1 - U_3$ в линии связи. Первое состояние ($S = 1$) соответствует нулевым напряжениям в линии: $(U_1 U_2 U_3) = (0 0 0)$. Это состояние выбрано для отображения единичного значения синхросигнала C_1 . Символы “ \times ” в левой части первой строки таблицы означают, что при $C_1 = 1$ входной код $X_1 Y_1 Z_1 V_1 J_1$ не воспринимается шифратором. Эти же символы в правой части первой строки таблицы показывают, что при выдаче восстановленного синхросигнала $C_2 = 1$ выходные данные $X_2 Y_2 Z_2 V_2 J_2$ также не определены. При этом сигнал M принимает значение лог. 1.

Состояния 2, 3, ..., 27, соответствующие условию $C_1 = 0$, используются для отображения данных первого и, в дополнение к этому, по возможности, второго каналов. Как следует из таблицы, коды P , принадлежащие диапазону 0 – 9, представлены двумя равноценными копиями. Первая копия размещена в строках с номерами 2 – 11 и сопровождается нулевым значением сигнала J_1 . Вторая копия размещена в строках с номерами 18 – 27 и сопровождается единичным значением сигнала J_1 . Передача первой или второй копии кода по первому каналу совмещена с передачей одного бита данных ($TxD_2 = J_1$) по второму каналу. Коды P , принадлежащие диапазону 10 – 15, представлены в единственном экземпляре и размещены в строках 12 – 17. При передаче этих кодов состояние сигнала J_1 не учитывается (что отражено символами “ \times ” в столбце “ J_1 ” таблицы). Иными словами, в данных ситуациях параллельная передача данных по второму каналу приостанавливается.

Как показано на временной диаграмме сигналов W линии, в первых половинах тактов состояние линии $S = 1$, так как $C_1 = 1$. Во второй половине первого такта $C_1 = 0$, $P = 2$, $J = 0$. Этому сочетанию сигналов соответствует четвертая строка таблицы. Из нее следует, что состояние линии $S = 4$. Аналогично, во второй половине второго такта $C_1 = 0$, $P = 5$, $J = 1$, в результате в линии формируется состояние $S = 23$ (+ – –) и т. д.

Дешифратор блока DCE2 восстанавливает синхросигнал C_2 (совпадающий с синхросигналом C_1 с точностью до задержки передачи) и формирует сигналы Q , J_2 и M (см. соответствующие временные диаграммы). Сигналы Q и J_2 содержат области неопределенности, выделенные на диаграммах затемненными участками.

Для устранения неопределенности сигналов Q применен параллельный регистр. Данные принимаются в него по положительному фронту сигнала C_3 , который формируется из сигнала C_2 после его задержки на четверть периода и последующего инвертирования. В результате сигналы RxD_1 и RxC_1 с некоторой задержкой повторяют сигналы TxD_1 и TxC_1 .

Сигнал C_3 далее проходит через элемент задержки для исключения “гонок”. При наличии единичного сигнала MX элемент И формирует периодический сигнал RxC_2 . При $MX = 0$ из сигнала RxC_2 исключаются положительные импульсы, соответствующие областям неопределенности сигнала J_3 . Таким образом регулируется поток данных по дополнительному каналу.

При равномерном случайном распределении кодов $TxD1$ вероятность их попадания в диапазон $0 - 9$ равна $10/16 = 0,625$. Каждое попадание в этот диапазон сопровождается передачей одного бита данных по дополнительному каналу. Поэтому скорость передачи битов по дополнительному каналу составляет $62,5\%$ скорости передачи четырехразрядных кодов $TxD1$ по основному каналу. Протокол передачи данных может предусматривать передачу нулевых кодов $TxD1$ в незанятом состоянии первого канала. Тогда дополнительный канал используется в каждом такте синхросигнала, т. е. в нем полностью исключаются приостановки передачи данных.

Для исключения неблагоприятных ситуаций, при которых в течение длительных промежутков времени коды попадают в диапазон $10 - 15$, можно применить скремблирование (см. п. 8.4) одного или двух старших разрядов данных $TxD1$ перед их подачей на входы блока $DCE1$.

8.5.3. Усовершенствованная схема – второй вариант

Недостатком рассмотренной ранее схемы (см. рис. 8.23) является низкая скорость передачи данных по линии связи. Это связано с тем, что в линии связи данные чередуются во времени с синхросигналом. Поэтому половина потенциальной пропускной способности канала связи теряется. В схеме, приведенной на рис. 8.29, по линии связи передаются только данные и, в необходимых случаях, признаки повтора кодов. При этом гарантируется изменение состояния линии на границах тактовых интервалов, что позволяет с высокой надежностью восстанавливать синхросигнал блоком приема ($DCE2$).

Все процессы, протекающие при передаче данных, синхронизируются сигналом TxC от генератора G блока $DCE1$. По положительному фронту этого сигнала источник данных (на рисунке не показан) формирует очередной трехразрядный двоичный код $X0 Y0 Z0$ передаваемых данных TxD ($X0$ – старший разряд). В примере, приведенном на временных диаграммах (рис. 8.30), от источника данных под управлением сигнала TxC поступает такая последовательность кодов: $101_2 = 5$, $011_2 = 3$, $001_2 = 1$, ..., $011_2 = 3$. При этом коды “6” и “4” повторяются в четырех и двух смежных тактах.

Регистр RG блока $DCE1$ принимает данные по положительному фронту сигнала TxC , поэтому в этом регистре фиксируется “старый” код TxD , прежде чем на его входы поступит “новый” код. В результате данные K на выходах регистра формируются с задержкой на один такт по сравнению с данными TxD на его входах.

Компаратор сравнивает старый код с новым. При их совпадении формируется сигнал $J = 1$. Этот сигнал сдвигается на один такт и “прореживается”, если его длительность превышает один период синхросигнала. Результирующий сигнал $W1$, формируемый триггерами, поступает на шифратор CD вместе с данными и несет информацию о повторах кодов (6, 6, 6, 6 и 4, 4). Цепь, состоящая из шифратора, выходных и входных усилителей, обслуживающих линию связи, и дешифратора блока $DCE2$ выполняет преобразование кодов в соответствии с Таблица 8.4 табл. 8.4.

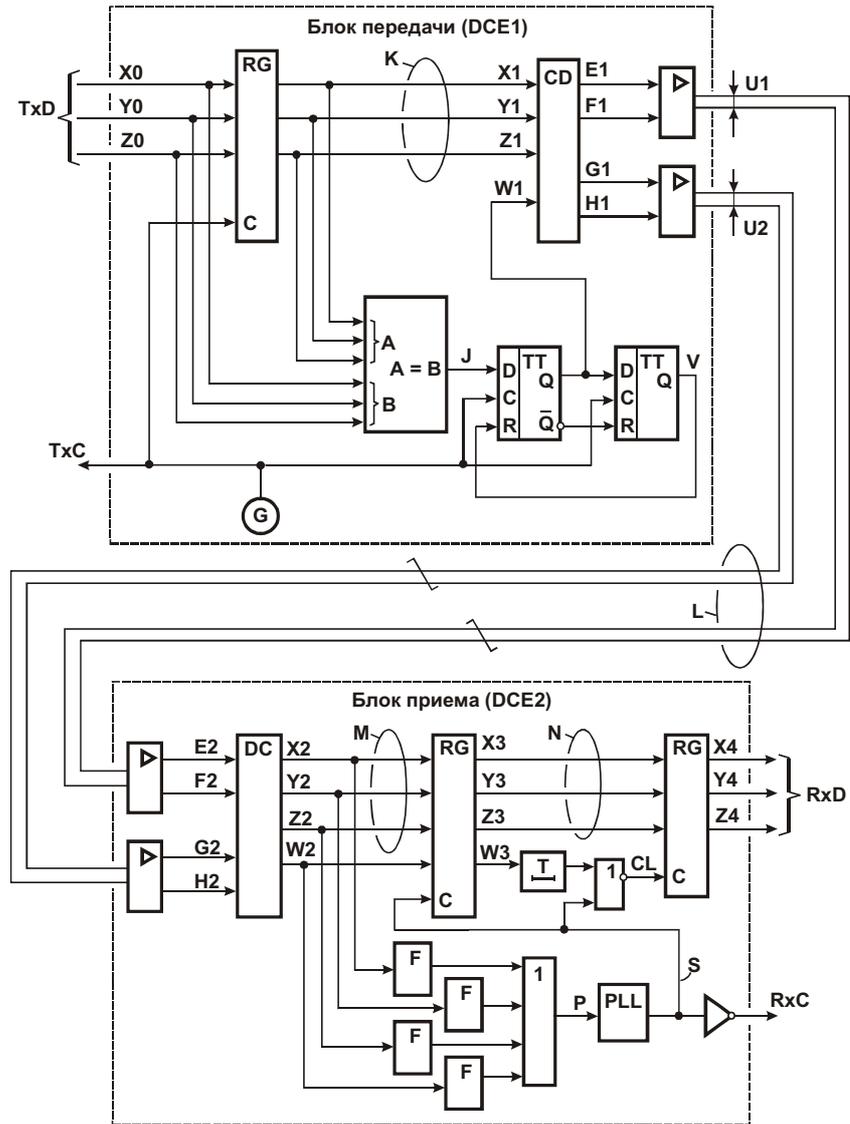


Рис. 8.29. Усовершенствованная схема параллельной передачи кодированных данных и синхросигнала по витым парам проводов (второй вариант)

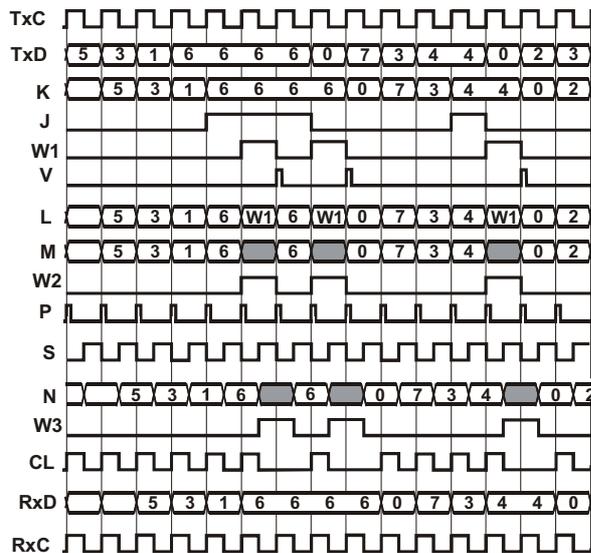


Рис. 8.30. Временные диаграммы передачи данных

Таблица 8.4 табл. 8.4
Состояния шифратора, канала связи и дешифратора при параллельной передаче трех битов данных и признака повтора кода схемой, приведенной на рис. 8.29

Коды на входах и выходах шифратора								Сигналы в линии		Коды на входах и выходах дешифратора							
W1	X1	Y1	Z1	E1	F1	G1	H1	U1	U2	H2	G2	F2	E2	Z2	Y2	X2	W2
1	×	×	×	0	0	0	0	0	0	0	0	0	0	×	×	×	1
0	0	0	0	0	0	0	1	0	–	1	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	+	0	1	0	0	1	0	0	0
0	0	1	0	0	1	0	0	–	0	0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1	–	–	1	0	1	0	1	1	0	0
0	1	0	0	0	1	1	0	–	+	0	1	1	0	0	0	1	0
0	1	0	1	1	0	0	0	+	0	0	0	0	1	1	0	1	0
0	1	1	0	1	0	0	1	+	–	1	0	0	1	0	1	1	0
0	1	1	1	1	0	1	0	+	+	0	1	0	1	1	1	1	0

Как и в предыдущих схемах, напряжение U_1 (U_2) между проводами витой пары проводов может быть отрицательным, нулевым или положительным (сокращенно: –, 0, +). Это напряжение формируется выходным усилителем блока DCE1 в зависимости от сочетания битов $E_1 F_1 (G_1 H_1)$ на его входах следующим образом. При $E_1 = F_1 = 0$ ($G_1 = H_1 = 0$) напряжение U_1 (U_2) нулевое, при $E_1 = 0, F_1 = 1$ ($G_1 = 0, H_1 = 1$) – отрицательное, при $E_1 = 1, F_1 = 0$ ($G_1 = 1, H_1 = 0$) – положительное. Код $E_1 F_1 G_1 H_1$ формируется шифратором CD из входного кода $X_1 Y_1 Z_1 W_1$ (данные, признак повтора), как показано в левой части таблицы.

Входные усилители и дешифратор блока DCE2 осуществляют обратное преобразование сигналов, представленных напряжениями U_1 и U_2 . В этом можно убедиться сопоставлением левой и правой частей таблицы – они симметричны. В центральной части таблицы показаны девять возможных комбинаций сигналов в линии. Первая комбинация (0 0) произвольно выбрана для отображения признака $W_1 = 1$ повтора кода данных. Символы “×” в левой части таблицы означают, что при $W_1 = 1$ входные сигналы K не воспринимаются шифратором.

Эти же символы в правой части таблицы показывают, что при выдаче дешифратором признака $W_2 = 1$ повтора кода выходные данные $X_2 Y_2 Z_2$ не определены и не могут использоваться для выдачи абоненту. Остальные восемь комбинаций сигналов в линии произвольно распределены для отображения восьми состояний трехразрядного кода $X_1 Y_1 Z_1$ при условии, что $W_1 = 0$.

Сигналы в линии связи в каждом такте несут информацию о передаваемом трехразрядном коде либо о том, что в данном такте код повторяется. Сигналы M на выходе дешифратора содержат области неопределенности, выделенные серым фоном; им соответствуют единичные состояния сигнала W_2 . Так как признак повтора кодируется в линии уникальным сочетанием сигналов и не может длиться более одного такта (благодаря “прореживанию”), то создается гарантия того, что в каждом такте состояние канала изменяется, т. е. отлично от предыдущего. Для регистрации границ тактовых интервалов использованы формирователи F импульсов, элемент ИЛИ и блок PLL фазовой автоподстройки частоты. Формирователи импульсов реагируют на любой фронт входного сигнала выдачей на входы элемента ИЛИ кратковременных сигналов лог. 1. По-

этому на выходе элемента ИЛИ формируется непрерывная последовательность импульсов.

Эта последовательность поступает на вход синхронизации блока фазовой автоподстройки частоты (PLL). Блок PLL может быть выполнен по одной из известных схем (см. гл. 9). Он предназначен для формирования высокостабильного синхросигнала S на основе непрерывного слежения за входным сигналом P . В данном примере отрицательный фронт сигнала S привязан к положительному фронту сигнала P . Благодаря достаточной инерционности блока PLL сигнал S практически нечувствителен к “дрожанию фазы” сигнала P и иным его кратковременным искажениям, вызванным помехами в канале связи.

По положительному фронту сигнала S установившиеся на выходе дешифратора сигналы фиксируются в промежуточном регистре. Далее, если нет повтора, код переписывается из промежуточного в выходной регистр. При наличии повтора элемент ИЛИ – НЕ блокируется незначительно задержанным сигналом $W3 = 1$. Поэтому сигнал CL синхронизации выходного регистра не периодичен. Как следует из диаграмм, выходные данные RxD повторяют входные данные TxD с точностью до задержки передачи. При этом сигнал RxC обеспечивает правильную синхронизацию данных RxD .

8.5.4. Усовершенствованная схема – третий вариант

Усовершенствования по п. 8.5.2 и 8.5.3 можно объединить в едином решении. Правда такое объединение не удастся получить простой “механической сборкой” готовых схем – поэтому здесь приводится одно из возможных синтетических схемных решений. Применительно к линии из трех витых пар проводов это решение позволяет в каждом такте работы линии передавать четыре бита по основному каналу и в среднем 0,625 бита по дополнительному каналу (итого 4,625 бит/такт). Чтобы избежать повторов при описании уже рассмотренных деталей, дальнейшее изложение приводится по возможности в сжатом виде в расчете на то, что читатель знаком с предыдущими решениями по п. 8.5.

Подключение схемы передачи данных (третий вариант) к телекоммуникационной системе можно выполнить согласно рис. 8.25. Схемы блоков передачи и приема данных и используемого в них дешифратора приведены на рис. 8.31 – рис. 8.33.

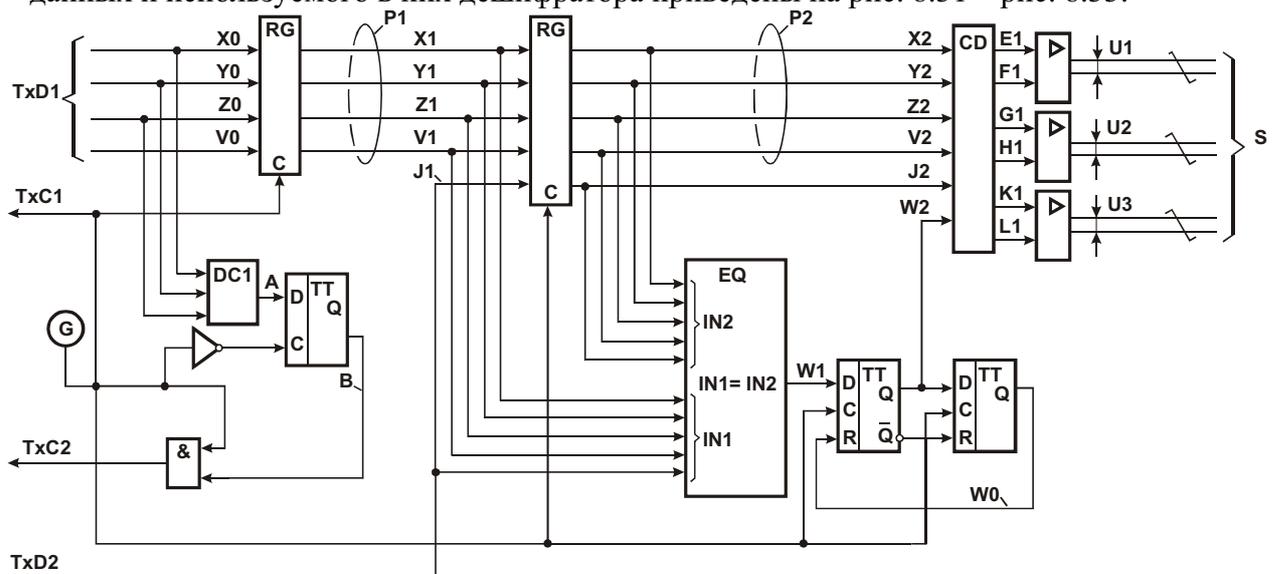


рис. 8.31. Блок передачи данных третьего варианта схемы

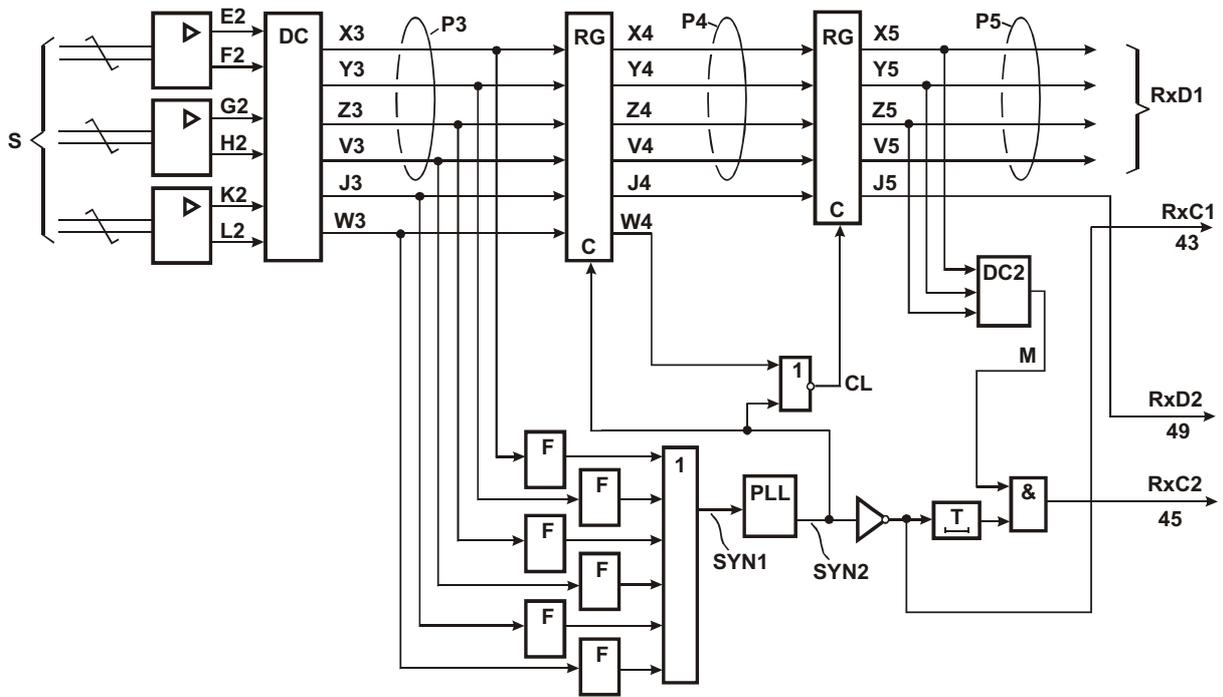


рис. 8.32. Блок приема данных третьего варианта схемы

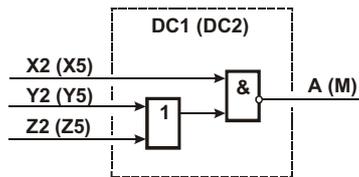


рис. 8.33. Схема дешифратора DC1 (DC2), применяемого в третьем варианте схемы

Преобразование кодов при их передаче по линии описывается Таблица 8.5 табл. 8.5. Временные диаграммы, соответствующие различным кодовым ситуациям, приведены на

рис. 8.34 – рис. 8.36.

Состояния шифратора CD, линии связи и дешифратора DC при передаче данных (третий вариант схемы)

№ п/п	Коды на входах и выходах шифратора CD											Состояние линии			Коды на входах и выходах дешифратора DC														
	W 2	J 2	X 2	Y 2	Z 2	V 2	P2	E 1	F 1	G 1	H 1	K 1	L 1	S	U1 – U3	L 2	K 2	H 2	G 2	F 2	E 2	P3	V 3	Z 3	Y 3	X 3	J 3	W 3	
1	1	×	×	×	×	×	×	0	0	0	0	0	0	1	000	0	0	0	0	0	0	×	×	×	×	×	×	×	1
2	0	0	0	0	0	0	0	0	0	0	0	1	0	2	00–	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	1	0	0	0	0	0	0	3	00+	0	1	0	0	0	0	1	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	1	0	0	0	4	0–0	0	0	1	0	0	0	2	0	1	0	0	0	0	0
5	0	0	0	0	1	1	0	0	0	1	0	1	0	5	0––	1	0	1	0	0	0	3	1	1	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	1	1	0	0	6	0–+	0	1	1	0	0	0	4	0	0	1	0	0	0	0
7	0	0	0	1	0	1	0	0	1	0	0	0	0	7	0+0	0	0	0	1	0	0	5	1	0	1	0	0	0	0
8	0	0	0	1	1	0	0	0	1	0	0	1	0	8	0+–	1	0	0	1	0	0	6	0	1	1	0	0	0	0
9	0	0	0	1	1	1	0	0	1	0	1	0	0	9	0++	0	1	0	1	0	0	7	1	1	1	0	0	0	0
10	0	0	1	0	0	0	0	0	1	0	0	0	0	10	–00	0	0	0	0	1	0	8	0	0	0	0	1	0	0
11	0	0	1	0	0	1	0	0	1	0	0	1	0	11	–0–	1	0	0	0	1	0	9	1	0	0	0	1	0	0
12	0	×	1	0	1	0	0	1	0	0	1	0	0	12	–0+	0	1	0	0	1	0	10*	0	1	0	1	0	×	0
13	0	×	1	0	1	1	0	0	1	0	1	0	0	13	––0	0	0	1	0	1	0	11*	1	1	0	1	×	×	0
14	0	×	1	1	0	0	0	1	0	1	0	1	0	14	–––	1	0	1	0	1	0	12*	0	0	1	1	×	×	0
15	0	×	1	1	0	1	0	1	0	1	1	0	0	15	––+	0	1	1	0	1	0	13*	1	0	1	1	×	×	0
16	0	×	1	1	1	0	0	1	1	0	0	0	0	16	–+0	0	0	0	1	1	0	14*	0	1	1	1	×	×	0
17	0	×	1	1	1	1	0	0	1	1	0	1	0	17	–+–	1	0	0	1	1	0	15*	1	1	1	1	×	×	0
18	0	1	0	0	0	0	0	0	1	1	0	1	0	18	–++	0	1	0	1	1	0	0	0	0	0	0	1	0	0
19	0	1	0	0	0	1	1	1	0	0	0	0	0	19	+00	0	0	0	0	0	1	1	1	0	0	0	1	0	0
20	0	1	0	0	1	0	0	0	1	0	0	1	0	20	+0–	1	0	0	0	0	1	2	0	1	0	0	1	0	0
21	0	1	0	0	1	1	0	0	1	0	0	1	0	21	+0+	0	1	0	0	0	1	3	1	1	0	0	1	0	0
22	0	1	0	1	0	0	0	1	0	0	1	0	0	22	+–0	0	0	1	0	0	1	4	0	0	1	0	1	0	0
23	0	1	0	1	0	1	0	1	0	1	0	1	0	23	+––	1	0	1	0	0	1	5	1	0	1	0	1	0	0
24	0	1	0	1	1	0	0	1	1	0	1	1	0	24	+–+	0	1	1	0	0	1	6	0	1	0	1	0	1	0
25	0	1	0	1	1	1	1	0	1	0	0	0	0	25	+ +0	0	0	0	1	0	1	7	1	1	1	0	1	0	0
26	0	1	1	0	0	0	0	1	0	1	0	0	1	26	+ +–	1	0	0	1	0	1	8	0	0	0	0	1	1	0
27	0	1	1	0	0	1	1	1	0	1	1	0	1	27	+ ++	0	1	0	1	0	1	9	1	0	0	1	1	1	0

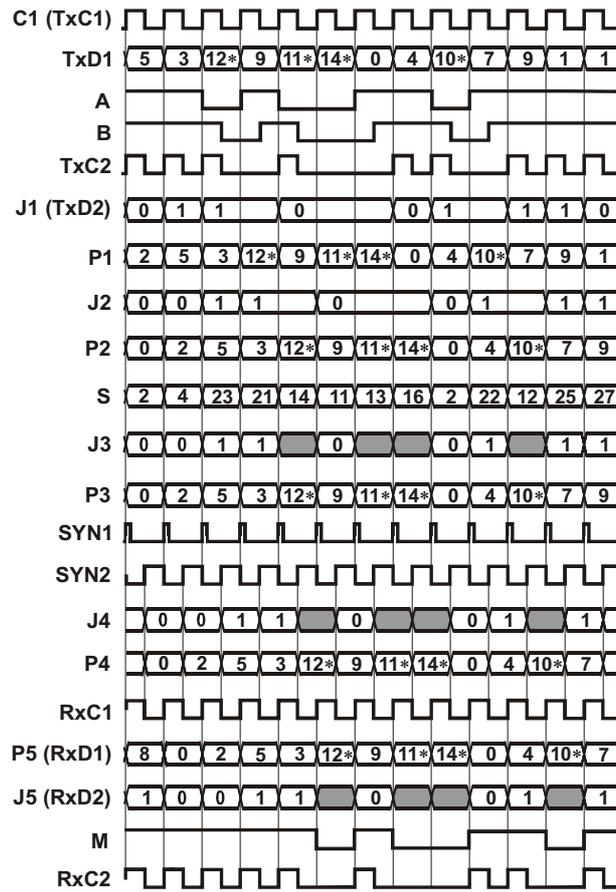


рис. 8.34. Временные диаграммы передачи данных с приостановками работы второго канала (третий вариант схемы)

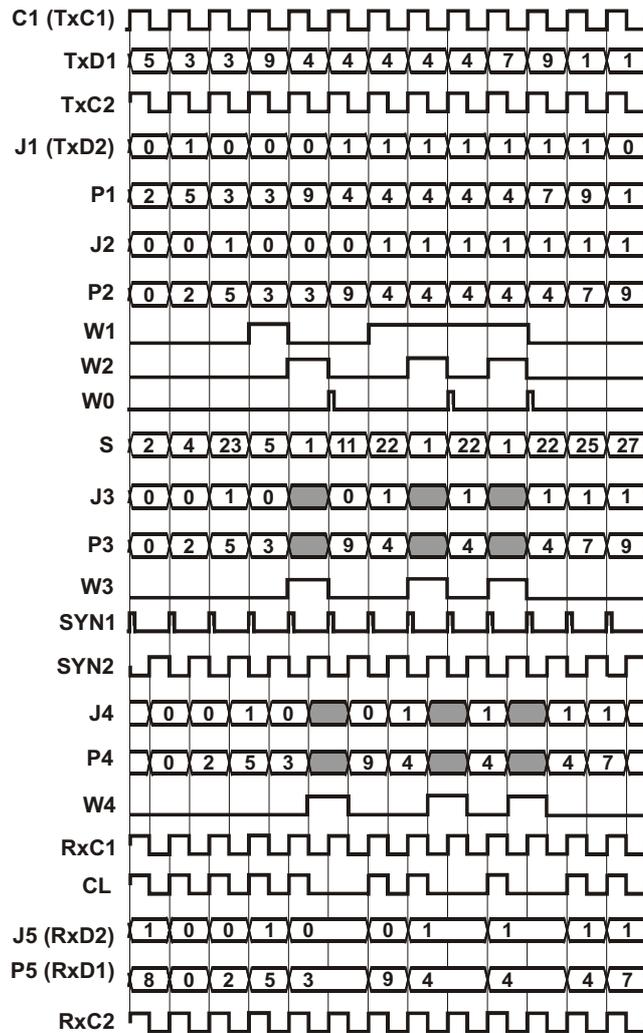


Рис. 8.35. Временные диаграммы передачи данных с повторами передаваемых кодов в смежных тактах (третий вариант схемы)

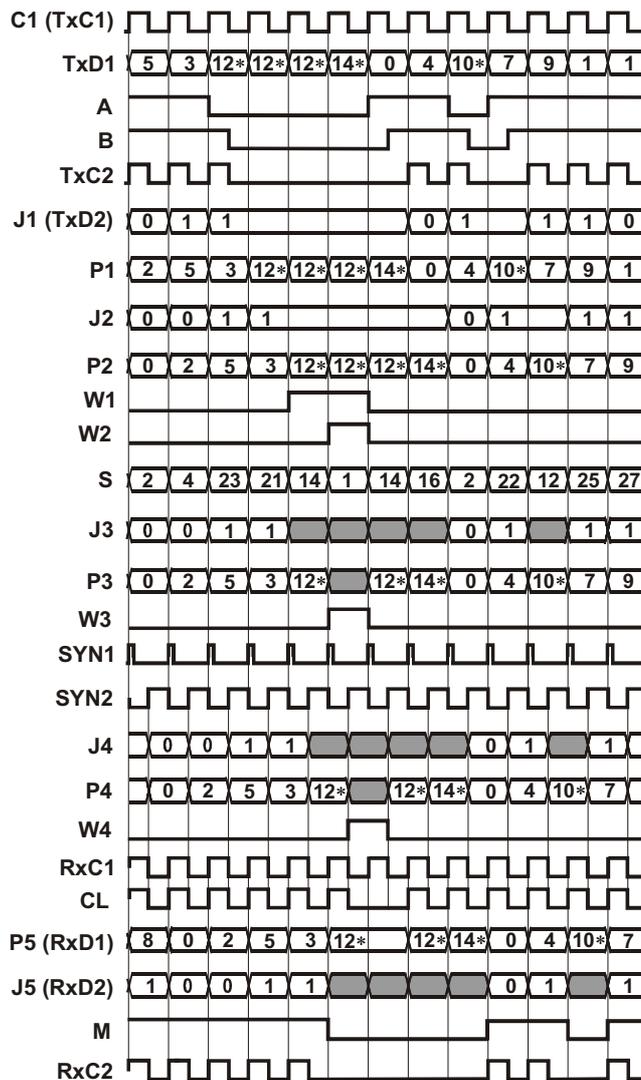


Рис. 8.36. Временные диаграммы передачи данных с приостановками работы второго канала и повторами кодов в смежных тактах (третий вариант схемы)

8.5.5. Усовершенствованная схема – четвертый вариант

Этот вариант предусматривает поочередную передачу четырех битов из первого и трех битов из второго каналов. Таким образом, за два такта передаются семь битов или в среднем 3,5 бита за один такт. По сравнению с предыдущим вариантом средняя скорость передачи данных снижена на $4,625 - 3,5 = 1,125$ бит/такт, но упрощены схемные решения и применено аппаратное кодирование – декодирование признаков начала информационных кадров. Как и ранее, описание приводится по возможности в сжатом виде в расчете на то, что читатель знаком с предыдущими решениями по п. 8.5.

Схема передачи данных (рис. 8.37) подключена к четырем оконечным устройствам DTE1 – DTE4, например к четырем компьютерам. В процессе работы системы данные TxD1 и признак TxF1 начала кадра передаются из устройства DTE1 в устройство DTE3 по основному (первому) каналу. Одновременно с этим по дополнительному (второму) каналу данные TxD2 и признак TxF2 начала кадра передаются из устройства DTE2 в устройство DTE4. (Для передачи информации в обратном направлении необходима вторая схема передачи данных, включенная встречно.)

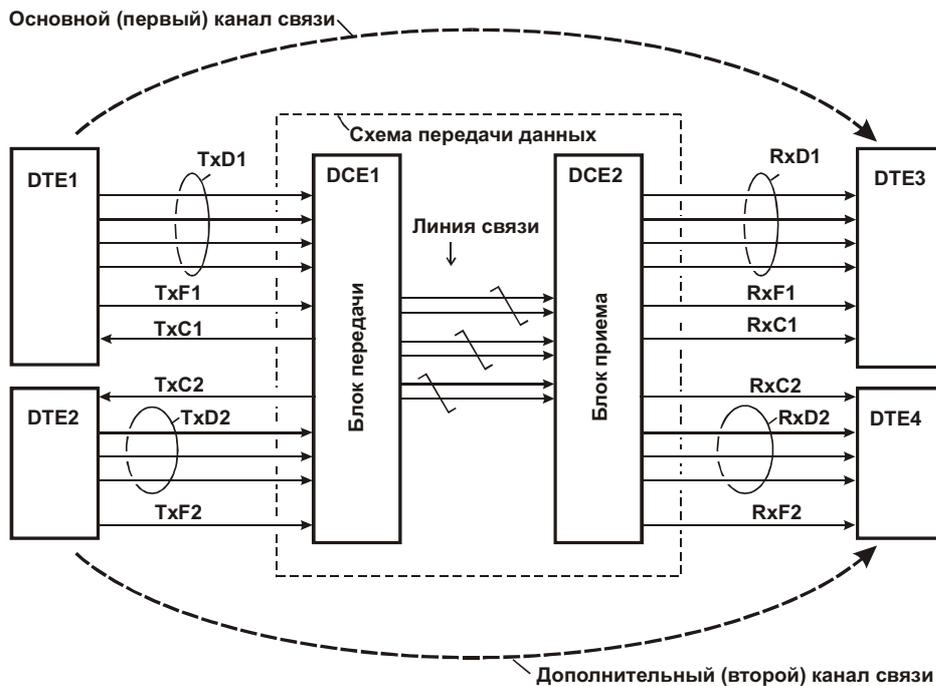


Рис. 8.37. Пример включения усовершенствованной схемы передачи данных (четвертый вариант) в телекоммуникационную систему

Сигнал $TxC1$ обеспечивает синхронную передачу данных $TxD1$ и признака $TxF1$ начала кадра из устройства DTE1 в блок передачи. Аналогично сигнал $TxC2$ обеспечивает синхронную передачу данных $TxD2$ и признака $TxF2$ начала кадра из устройства DTE2 в этот же блок. Положительные фронты сигнала $TxC1$ задают границы битовых интервалов для каждого входного сигнала первого канала. Положительные фронты сигнала $TxC2$ задают границы битовых интервалов для каждого входного сигнала второго канала. Аналогичные функции выполняет синхросигнал $RxC1$ ($RxC2$) сопровождения данных $RxD1$ ($RxD2$) и признаков $RxF1$ ($RxF2$) начала кадров.

Схема передачи данных и временные диаграммы ее работы приведены на рис. 8.38 и рис. 8.39. Кодирование и декодирование данных при их передаче по линии выполняется в соответствии с Таблица 8.6 табл. 8.6.

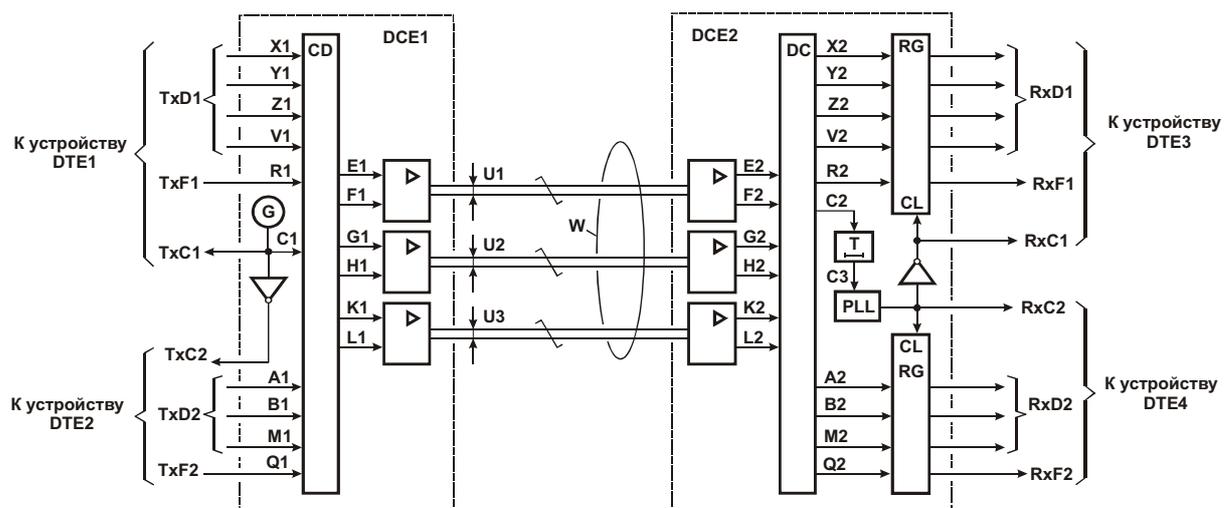


Рис. 8.38. Схема передачи данных (четвертый вариант)

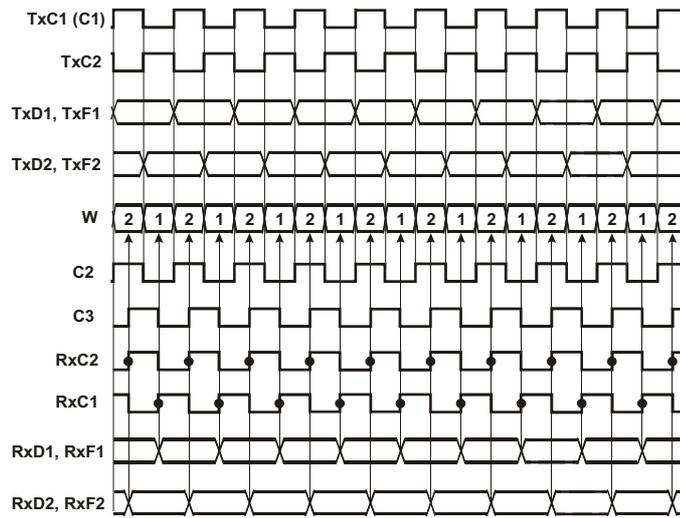


рис. 8.39. Временные диаграммы работы схемы передачи данных (четвертый вариант).
 На временной диаграмме сигналов W в линии связи цифрами 1 и 2 обозначены данные, передаваемые по первому и второму каналам

Таблица 8.6 табл. 8.6

Состояния шифратора, линии связи и дешифратора при передаче данных схемой, приведенной на рис. 8.38

№ п/п	Коды на входах и выходах шифратора												Состояние линии		Коды на входах и выходах дешифратора																				
	R	Q	C	X	Y	Z	V	A	B	M	E	F	G	H	K	L	S	U1-U3	L	K	H	G	F	E	M	B	A	V	Z	Y	X	C	Q	R	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	0	1	0	0	0	1	0	x	x	0	0	0	0	0	x	0		
2	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	-	0	0	1	0	0	1	0	x	x	1	0	0	0	0	x	0		
3	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	-	0	0	1	0	1	0	x	x	0	0	1	0	0	0	x	0		
4	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	-	0	1	0	1	0	0	x	x	1	1	0	0	0	0	x	0		
5	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	-	+	0	1	0	1	0	x	x	0	0	1	0	0	0	x	0		
6	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	-	+	0	0	1	1	0	x	x	1	0	1	0	0	0	x	0		
7	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	-	+	0	1	0	1	0	x	x	0	0	1	1	0	0	x	0		
8	0	0	0	0	1	1	1	0	0	1	0	0	1	0	0	0	-	+	0	1	0	1	0	x	x	1	1	1	0	0	0	x	0		
9	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	+	0	0	0	0	1	0	x	x	0	0	0	1	0	0	0	x	0	
10	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	+	0	1	0	0	1	0	x	x	1	0	0	1	0	0	0	x	0	
11	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	+	0	0	1	0	0	1	x	x	0	0	1	0	1	0	0	x	0	
12	0	0	0	1	0	1	1	0	0	1	0	0	0	1	0	0	+	-	0	1	0	0	1	x	x	1	1	0	1	0	0	0	x	0	
13	0	0	0	1	1	0	0	0	0	1	0	0	1	0	1	0	+	-	0	1	1	0	1	x	x	0	0	1	1	0	0	0	x	0	
14	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	+	0	0	0	0	1	0	x	x	1	0	1	1	0	0	0	x	0	
15	0	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	+	+	0	1	0	0	1	x	x	0	0	1	1	1	0	0	0	x	0
16	0	0	0	1	1	1	1	0	0	1	0	0	1	0	1	0	+	+	0	1	0	1	0	x	x	1	1	1	1	0	0	0	x	0	
17	x	0	0	1	x	x	x	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	x	x	x	1	0	x	0	
18	x	0	0	1	x	x	x	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	x	x	x	x	1	0	x	0	
19	x	0	0	1	x	x	x	0	1	0	0	0	0	1	0	0	0	-	0	0	1	0	0	0	1	0	x	x	x	x	1	0	x	0	
20	x	0	0	1	x	x	x	0	1	0	0	0	0	1	0	0	0	-	0	1	0	0	0	1	1	0	x	x	x	x	1	0	x	0	
21	x	0	0	1	x	x	x	0	1	0	0	0	0	1	0	0	0	-	+	0	1	0	0	0	0	1	x	x	x	x	1	0	x	0	
22	x	0	0	1	x	x	x	0	1	0	0	1	0	0	0	0	0	+	0	0	0	1	0	0	1	x	x	x	x	1	0	x	0		
23	x	0	0	1	x	x	x	0	1	0	0	1	0	0	0	1	0	+	0	0	1	0	0	0	1	x	x	x	x	1	0	x	0		
24	x	0	0	1	x	x	x	0	1	1	0	1	0	0	1	0	0	+	0	1	0	1	0	1	x	x	x	x	1	1	0	x	0	x	
25	1	x	0	x	x	x	x	0	1	x	0	0	0	0	0	0	-	0	0	0	0	1	0	x	x	0	0	0	0	0	0	x	1	x	
26	x	1	0	x	x	x	x	0	0	x	0	0	0	0	0	0	+	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	x	x

Все процессы, протекающие при передаче данных, синхронизируются от генератора G. По положительному фронту сигнала TxC1 на входы первого канала устройства поступает очередной четырехразрядный двоичный код X1 Y1 Z1 V1 данных TxD1 (X1 – старший разряд) и в начале каждого информационного кадра – признак TxF1 = 1 (R1 = 1). Аналогично по положительному фронту сигнала TxC2 на входы второго канала устройства поступает очередной трехразрядный двоичный код A1 B1 M1 данных TxD2 (A1 – старший разряд) и в начале каждого информационного кадра – признак TxF2 = 1 (Q = 1).

Сигналы с входов устройства и с выхода генератора G поступают на входы шифратора. Так же как и в предыдущих примерах, применено трехуровневое кодирование сигналов в линии связи. Напряжение U1 (U2, U3) между проводами витой пары может быть отрицательным, нулевым или положительным (сокращенно: –, 0, +). Это напряжение формируется усилителем в зависимости от сочетания битов E1 F1 (G1 H1, K1 L1) на его входах следующим образом. При E1 = F1 = 0 (G1 = H1 = 0, K1 = L1 = 0) напряжение U1 (U2, U3) нулевое, при E1 = 0, F1 = 1 (G1 = 0, H1 = 1; K1 = 0, L1 = 1) – отрицательное, при E1 = 1, F1 = 0 (G1 = 1, H1 = 0; K1 = 1, L1 = 0) – положительное. Код E1 F1 G1 H1 K1 L1 формируется шифратором из входного кода, как показано в левой части Таблица 8.6 табл. 8.6.

Входные усилители и дешифратор осуществляют обратное преобразование сигналов, представленных напряжениями U1, U2 и U3 в витых парах проводов линии связи. В графе S таблицы перечислены 26 из $3^3 = 27$ возможных состояний трехуровневых сигналов U1 – U3 в линии связи (одно из состояний в данном примере не используется, хотя оно может отображать какой-либо дополнительный признак). Первое состояние (S = 1) соответствует напряжениям (U1 U2 U3) = (– 0 –); второе – напряжениям (U1 U2 U3) = (– 0 +) и т. д.

Символы “x” в левой части таблицы означают, что соответствующие входные сигналы игнорируются шифратором. В частности, как показано в строках 1 – 16, при R1 = C1 = 0 шифратор оперирует кодом TxD1 (X1 Y1 Z1 V1) и не воспринимает сигналы с входов Q, A1, B1, M1. Аналогично при Q1 = 0 и C1 = 1 (см. строки 17 – 24) шифратор воспринимает код TxD2 (A1 B1 M1) и игнорирует остальные входные сигналы. Сочетание сигналов R1 = 1 и C1 = 0 (строка 25) однозначно трактуется как признак начала передаваемого по первому каналу информационного кадра независимо от состояний сигналов на остальных входах. Аналогично строка 26 определяется условием Q = C = 1 и описывает передачу признака начала кадра по второму каналу.

Правая часть таблицы, в основном, представляет собой зеркальную копию левой части. Исключение составляют две последние строки (24 и 25), в которых некоторым символам “x” в левой части поставлены в соответствие символы “0” в правой части. Это гарантирует формирование нулевых кодов RxD1 и RxD2 в тех ситуациях, когда передаются признаки RxF1 и RxF2 начала кадров.

Из таблицы и временных диаграмм следует, что по мере изменения логического значения тактового сигнала TxC1 (C1) в линию попеременно поступает информация из первого и второго каналов. При этом сигнал C2 на выходе дешифратора повторяет сигнал C1 с точностью до задержки передачи. Сигнал C3 на выходе элемента задержки сдвинут на четверть периода относительно сигнала C2 для совмещения его положительных и отрицательных фронтов с установившимися значениями информационных сигналов на входах выходных регистров первого и второго каналов.

Сигнал C3 поступает на вход синхронизации блока PLL фазовой автоподстройки частоты. Благодаря достаточной инерционности этого блока сигнал RxC2 практически нечувствителен к “дрожанию фазы” сигнала C3 и иным его кратковременным искажениям, вызванным помехами в линии связи. По положительным фронтам сигналов RxC2 и RxC1 информация с выходов дешифратора фиксируется в выходных регистрах и поступает к оконечным устройствам DTE3 и DTE4.

Передаваемые по первому и второму каналам данные группируются в кадры. Длина кадра может быть постоянной или переменной. Для обозначения начала кадра, передаваемого по первому каналу, используются сигналы TxF1 (на передающей стороне) и RxF1 (на

приемной стороне). Как следует из таблицы, при $T_x F_1 = 1$ данные $T_x D_1$ не воспринимаются на передающей стороне и доопределяются нулями на приемной стороне. То же справедливо и по отношению ко второму каналу.

Благодаря аппаратной передаче признаков начала кадров исключена необходимость выполнения операции битстаффинга. Напомним, что эта операция связана с введением служебных битов в передаваемый массив данных для создания уникальных (флаговых) кодовых комбинаций с целью обозначения начала информационных кадров. Поэтому в данном случае также исключена передача служебных битов по линии связи, последующего программного или аппаратного распознавания флаговых комбинаций, обнаружения и вычеркивания служебных битов.

Подведем некоторые итоги.

1. Основной вариант схемы передачи данных (рис. 8.23) характеризуется следующими признаками. По линии, состоящей из N витых пар проводов, в каждом такте передается одно из 3^N состояний трехуровневого сигнала. Одно из состояний используется для передачи синхросигнала, остальные – для передачи групп битов данных. Передача синхросигнала чередуется с передачей групп битов данных. При $N = 3$ используются все 3^3 состояний трехуровневого сигнала, однако с увеличением N появляются избыточные состояния, которые предлагается использовать в усовершенствованных вариантах схем.

2. Первый вариант усовершенствованной схемы (рис. 8.26) предусматривает использование избыточных состояний трехуровневого сигнала для построения дополнительного канала связи. Передача данных по этому каналу совмещена во времени с передачей основного потока данных и осуществляется при наличии в последнем кодовых комбинаций, разрешающих работу дополнительного канала связи. Чтобы гарантировать появление таких комбинаций, входные данные предварительно скремблируются.

3. Второй вариант схемы (рис. 8.29) позволяет повысить скорость передачи данных благодаря исключению состояния трехуровневого сигнала, отображающего синхросигнал. Поэтому во втором варианте схемы нет чередования этого состояния с передачей групп битов данных (скорость передачи повышена в два раза). Синхросигнал содержится в потоке данных в неявном виде как результат изменения состояния канала связи в каждом такте. Чтобы гарантировать такие изменения при передаче повторяющихся данных, последние в необходимых случаях заменяются кодом повтора, который подменяет эти данные.

4. Третий вариант схемы (рис. 8.31, рис. 8.32), по существу, объединяет отмеченные ранее усовершенствования по первому и второму вариантам. Такое объединение, однако, не удастся получить простой “механической сборкой” готовых схем, поэтому описано одно из возможных синтетических решений, сочетающих преимущества рассмотренных ранее вариантов.

5. Четвертый вариант схемы (рис. 8.38) также предусматривает использование избыточных состояний трехуровневого сигнала для построения дополнительного канала связи. Данные из основного и дополнительного каналов связи передаются поочередно, их скремблирования не требуется. Синхросигнал передается в неявном виде. По сравнению с предыдущим вариантом, средняя скорость передачи данных несколько снижена, но упрощены схемные решения и расширены функциональные возможности – два состояния трехуровневого сигнала выделены для передачи по каждому каналу признаков начала информационных кадров.

9. Выделение синхросигнала и данных из принимаемого сигнала

В этой главе рассмотрены примеры построения схем для выделения синхросигнала и данных из сигнала, передаваемого по каналу связи. В большинстве случаев схемы выполнены на основе генераторов с фазовой автоподстройкой частоты. Варианты схем генераторов различны, но решаемая с их помощью задача одна и та же – она состоит в правильной привязке синхросигнала к битовым интервалам принимаемых данных.

Два варианта схем, приведенные в п. 9.1, могут использоваться в высокоскоростных системах передачи данных по волоконно-оптическим линиям связи. Второй вариант позволяет также проектировать древовидные структуры для демультиплексирования входного потока данных либо для преобразования последовательного потока данных в параллельный. В п. 9.2 приведены схемы выделения синхросигнала и данных на основе сдвиговых регистров, в п. 9.3 – схемы на основе одноконтурного и двухконтурных генераторов синхросигналов, в п. 9.4 – схема с повышенной точностью воспроизведения синхросигнала при работе на переменную емкостную нагрузку. В п. 9.5 рассмотрен пример применения коррелятора для повышения качества распознавания синхроимпульсов во входном сигнале. В конце главы (п. 9.6) приведены примеры построения схем шифратора и дешифраторов кода Манчестер-II (см. п. 8.1.4).

9.1. Выделение синхросигнала и данных схемами на основе генераторов с фазовой автоподстройкой частоты

Генератор с фазовой автоподстройкой частоты (рис. 9.1) [38] содержит фазовый компаратор PD, инвертор, генераторы одинаковых по величине токов I_1 , I_2 , ключи на транзисторах VT1 и VT2, фильтр низких частот и синхрогенератор VCO, управляемый напряжением. Сигнал DIN в коде NRZ (см. п. 8.1) поступает из линии на вход фазового компаратора. Этот сигнал не должен оставаться в неизменном состоянии (лог. 0 или 1) на протяжении более или менее длительного интервала времени. Например, при проектировании системы передачи данных может быть заранее известно, что максимальная длительность неизменного состояния сигнала не должна превышать двадцати битовых интервалов. (Напомним, что для обеспечения гарантированного изменения состояния сигнала в любом заранее заданном интервале времени применяют скремблирование этого сигнала перед его выдачей в линию, см. п. 8.4.)

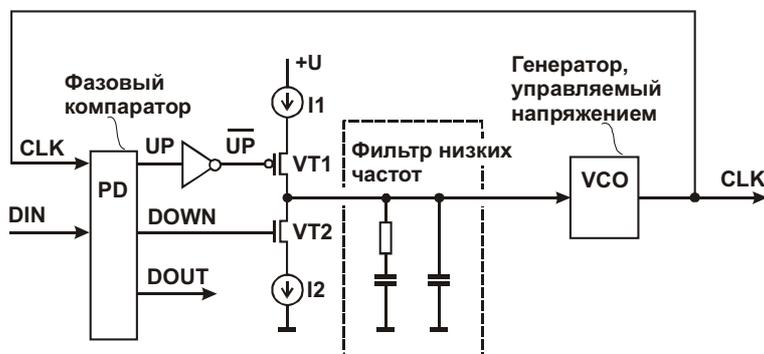


рис. 9.1. Схема генератора с фазовой автоподстройкой частоты (первый вариант)

Фазовый компаратор сопоставляет сигнал DIN с синхросигналом CLK от генератора VCO. При правильном фазовом соотношении этих сигналов на выходе данных фазового компаратора формируется сигнал DOUT, соответствующий данным в линии, при этом гра-

ницы его битовых интервалов задаются отрицательными фронтами сигнала CLK. Чтобы достичь, а затем и поддерживать правильное фазовое соотношение между сигналами DIN и CLK, фазовый компаратор непрерывно отслеживает ошибку взаимного расположения фронтов этих сигналов и формирует управляющие сигналы ускорения UP и замедления DOWN темпа работы генератора VCO.

Генератор VCO при номинальном управляющем напряжении, примерно равном $U/2$, формирует синхросигнал номинальной (ожидаемой) частоты. Повышение и понижение напряжения на управляющем входе генератора приводит к незначительному повышению и снижению частоты. Это позволяет подстраивать фазу синхросигнала в нужном направлении.

Чтобы пояснить, каким образом, варьируя частоту сигнала, можно корректировать его фазу, проведем “мысленный эксперимент”, в котором частота сигнала CLK на выходе генератора VCO в незначительных пределах регулируется переменным резистором R (рис. 9.2). При некотором положении движка переменного резистора, близком к среднему, частота сигнала CLK совпадает с частотой “идеального” сигнала CLKI, к которому необходимо подстроиться. Сигнал CLKI поступает на вход синхронизации осциллографа и отображается его верхним лучом. При равенстве частот сигналов CLK и CLKI нижняя осциллограмма неподвижна относительно верхней, при этом, вероятнее всего, наблюдается некоторый фазовый сдвиг сигнала CLK относительно сигнала CLKI. На рисунке показана одна из возможных ситуаций, при которой сигнал CLK отстает по фазе от сигнала CLKI примерно на 110 градусов.

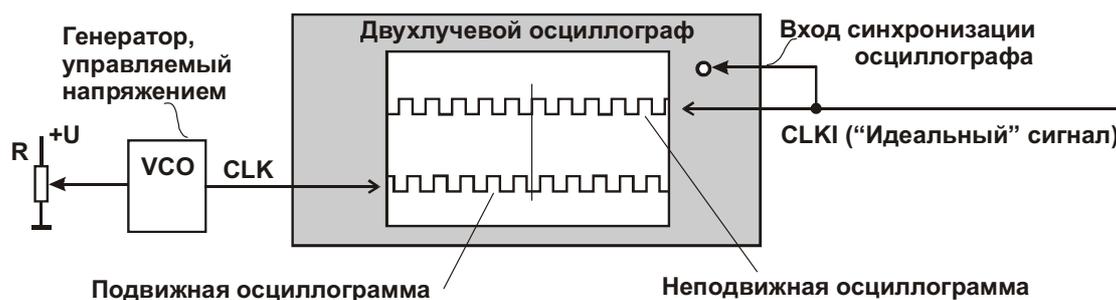


рис. 9.2. Схема, поясняющая процесс фазовой подстройки частоты

Чтобы скомпенсировать несовпадение фаз, немного увеличим частоту сигнала CLK перемещением движка переменного резистора вверх. Фронты этого сигнала будут формироваться чуть раньше, нижняя осциллограмма начнет медленно перемещаться влево относительно неподвижной верхней, фазовый сдвиг начнет уменьшаться. Остается только приостановить перемещение в момент, когда фазовый сдвиг станет нулевым. Для этого в нужный момент следует вернуть движок переменного резистора в исходное положение, соответствующее равенству частот.

Сигналы UP и DOWN (рис. 9.1) управляют транзисторными ключами, регулирующими поступление токов I_1 и I_2 на вход фильтра низких частот. При $UP = 0$, $DOWN = 0$ транзисторы VT1 и VT2 выключены, источники тока изолированы от фильтра и генератора VCO, напряжение на входе этого генератора остается постоянным либо незначительно изменяется из-за выравнивания напряжений на конденсаторах фильтра, если исходно эти напряжения не совпадали. При $DOWN = 0$ и поступлении серии импульсов $UP = 1$, транзистор VT1 включается по каждому импульсу, транзистор VT2 постоянно выключен, импульсы тока I_1 заряжают конденсаторы фильтра, напряжение на входе генератора VCO повышается, частота сигнала CLK незначительно возрастает. При $UP = 0$ и поступлении серии импульсов $DOWN = 1$ ситуация противоположная – транзистор VT2 открывается по каждому импульсу, транзистор VT1 постоянно закрыт, импульсы тока I_2 разряжают конденсаторы, частота сигнала CLK незначительно снижается. При $UP = 1$, $DOWN = 1$ оба транзистора включены и, поскольку $I_1 \approx I_2$, токи практически компенсируются и напряжение на входе генератора VCO почти не изменяется. В реальных условиях работы генератора

(рис. 9.1) импульсы UP и DOWN, в основном, чередуются во времени и чаще всего не перекрываются. Благодаря фильтру низких частот, импульсные воздействия на частоту генератора VCO со стороны фазового компаратора усредняются. Заметный вклад в процесс регулирования частоты сигнала CLK может внести только усредненное по времени устойчивое расхождение этих воздействий.

Рассмотрим первый вариант схемы фазового компаратора PD (рис. 9.3). Схема содержит последовательно включенные D-триггеры D1, D3, инвертор D2 и элементы Исключающее ИЛИ D4 и D5. Входные данные DIN принимаются в триггер D1 по положительным фронтам сигнала CLK и переписываются в триггер D3 по отрицательным фронтам этого сигнала. Элементы D4 и D5 формируют положительные импульсы UP и DOWN (сигналы лог. 1) в периоды несовпадений сигналов на входах и выходах соответствующих триггеров D1 и D3. С увеличением этих периодов возрастает длительность соответствующих импульсов (фактически применена широтно-импульсная модуляция).

Таким образом, сигналы UP и DOWN отражают фазовые соотношения между тремя сигналами – DIN и двумя его производными, полученными с использованием положительного и отрицательного фронтов сигнала CLK. При правильном соотношении сигналов DIN и CLK сигналы UP и DOWN вносят одинаковые, но противоположные вклады в процессы ускорения и замедления работы генератора VCO, поэтому частота и фаза синхросигнала остается неизменной. При смещении фазы синхросигнала длительность и число импульсов UP или DOWN возрастает, что приводит к компенсации смещения.

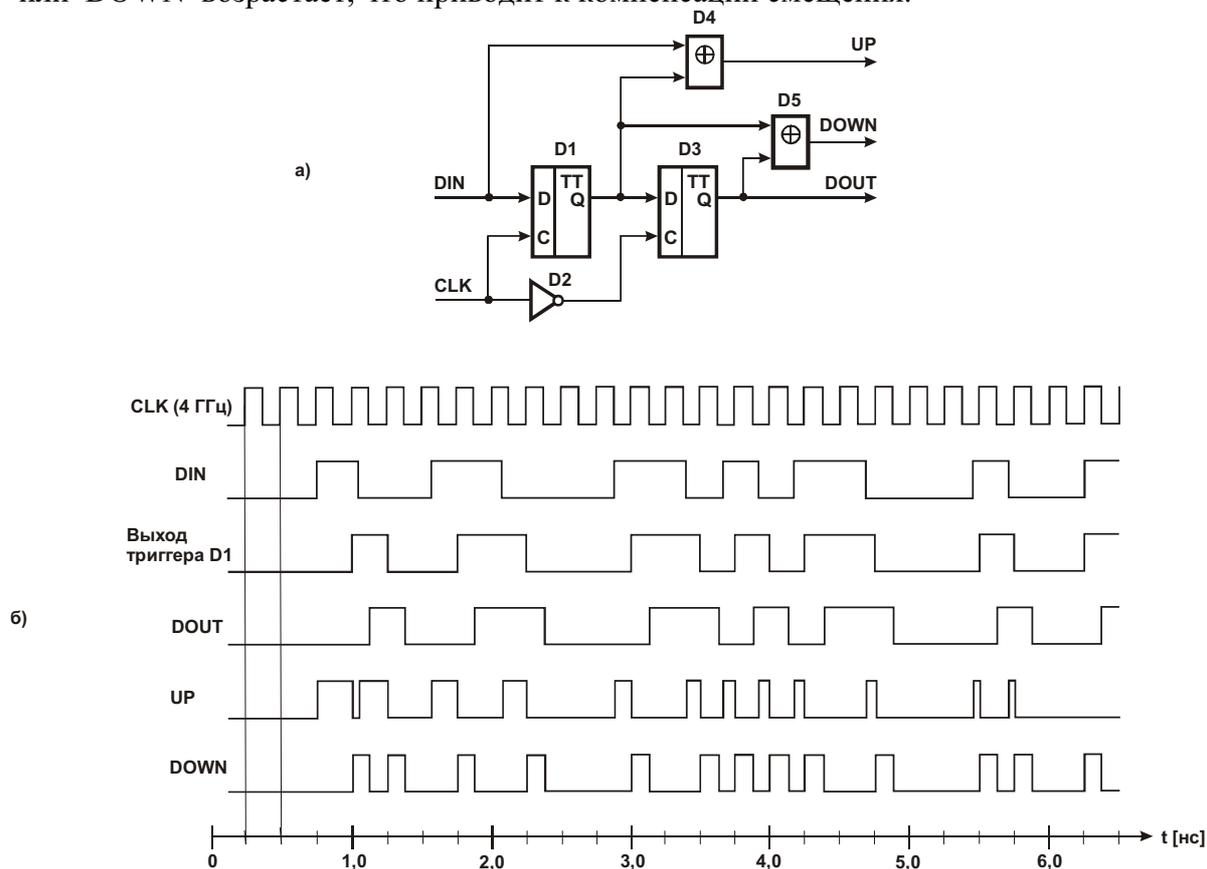


рис. 9.3. Фазовый компаратор PD: *а* – схема (первый вариант); *б* – временные диаграммы

Недостатком схемы (рис. 9.3) является высокая частота синхронизации. В приведенном примере частота сигнала CLK составляет 4 ГГц (период синхросигнала равен 0,25 нс). Эту частоту можно уменьшить вдвое, если использовать для приема данных как положительные, так и отрицательные фронты синхросигнала [38]. (Отметим, что подобная идея, применительно к иной задаче, была рассмотрена ранее, см. п. 3.3.1 [63].)

Прежде чем перейти к описанию второго варианта схемы фазового компаратора, рассмотрим его основную часть – формирователь сигналов UP и DOWN (рис. 9.4), работающий на пониженной частоте (2 ГГц).

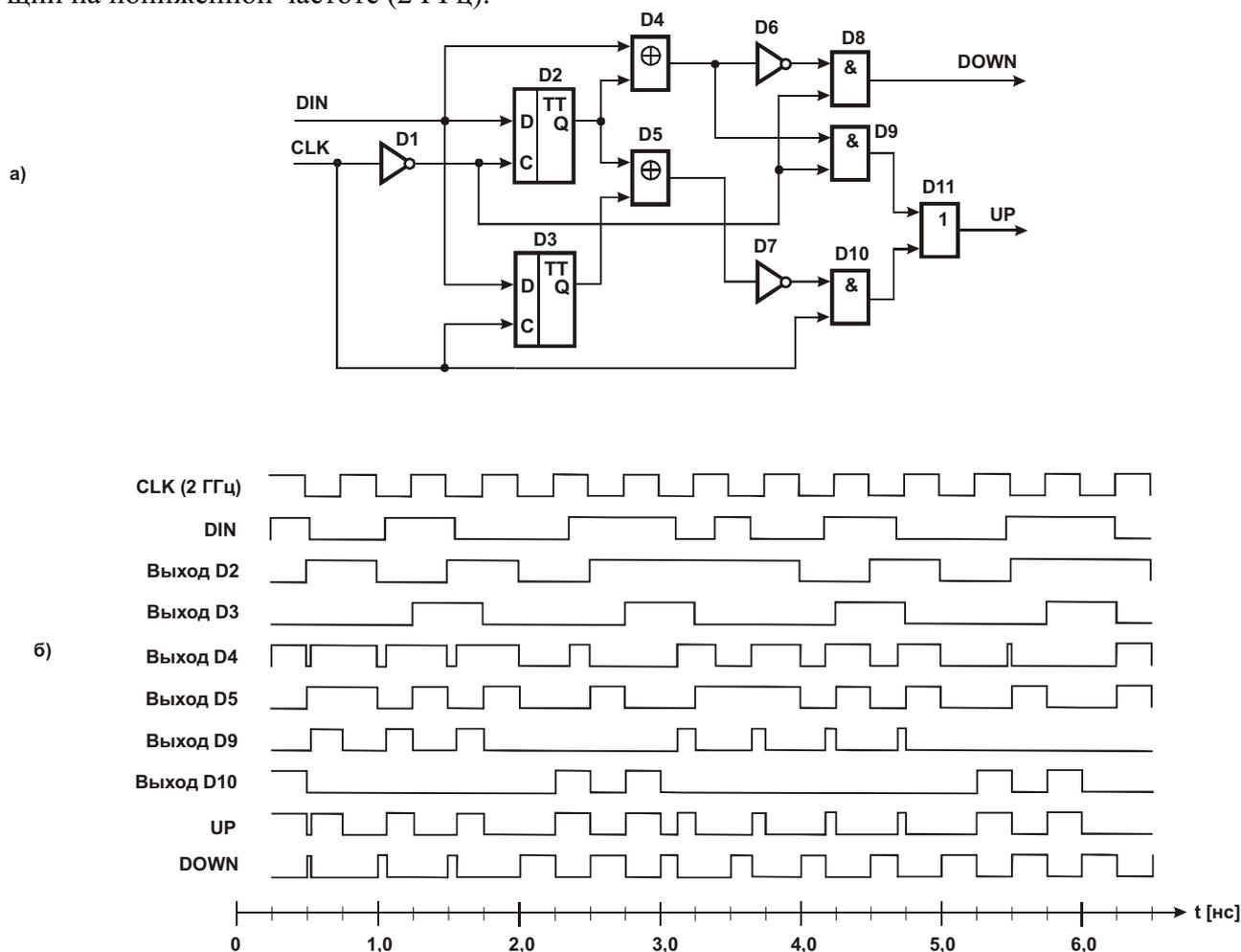


рис. 9.4. Формирователь сигналов UP и DOWN, с синхронизацией от сигнала CLK пониженной частоты (2 ГГц)

Входной сигнал DIN по положительному фронту сигнала CLK записывается в триггер D3, а по отрицательному фронту – в триггер D2. Элемент Иключающее ИЛИ D4 формирует сигнал лог. 1 при несовпадении сигналов на входе и выходе триггера D2. Элемент Иключающее ИЛИ D5 формирует сигнал лог. 1 при несовпадении сигналов на выходах триггеров D2 и D3. Элементы D6 – D11 формируют сигналы UP и DOWN на основе сигналов с выходов элементов D4, D5, а также и прямого и инверсного сигналов CLK.

Второй вариант схемы фазового компаратора (рис. 9.5) содержит только что рассмотренный формирователь сигналов UP и DOWN (рис. 9.4), а также D-триггер D12.

Эта схема обладает следующими полезными особенностями. Во-первых, как уже отмечалось, при той же скорости поступления данных по линии, что и в схеме, показанной на рис. 9.3, она оперирует синхросигналом, частота которого уменьшена в два раза. Во-вторых, выходной поток данных распределен по двум каналам. Биты данных DIN, считанные из линии по положительным фронтам сигнала CLK, временно запоминаются в триггере D3 и выдаются из него в виде потока DOUT1. Биты данных DIN, считанные из линии по отрицательным фронтам сигнала CLK, последовательно проходят через триггеры D2 и D12 и образуют поток DOUT0.

В чем состоит польза от разделения каналов? Во-первых, благодаря такому разделению, последующая обработка данных из обоих каналов может осуществляться параллельно, но вдвое медленнее по сравнению с обработкой данных из одноканального потока. Напри-

В этой структуре элементарный демультиплексор первого уровня принимает сигнал DIN и управляет генератором VCO, включенным в контур фазовой автоподстройки частоты точно также, как и в рассмотренной ранее схеме (рис. 9.1). Для синхронизации двух демультиплексоров второго уровня и четырех демультиплексоров третьего уровня частота синхросигнала CLK от генератора VCO снижается в два и четыре раза с помощью делителя частоты на D-триггерах D1 и D2. Так как выходные сигналы UP и DOWN демультиплексоров второго и третьего уровней не используются, формователи этих сигналов (рис. 9.5, элементы D6 – D11) можно из них исключить. Структуру (рис. 9.6) можно также использовать в качестве преобразователя битового потока данных DIN в байтовый DATA0 – DATA7.

9.2. Выделение синхросигнала и данных схемой на основе сдвиговых регистров

Схема, показанная на рис. 9.7 [25], предназначена для выделения синхросигнала SYNC и данных DOUT из сигнала DIN, поступающего через приемник из линии. Для начала предположим, что линейный сигнал представлен кодом NRZ (см. рис. 8.2). При этом допустим, что в коде NRZ максимальное число следующих подряд лог. 0 или лог. 1 не превышает, например, шести.

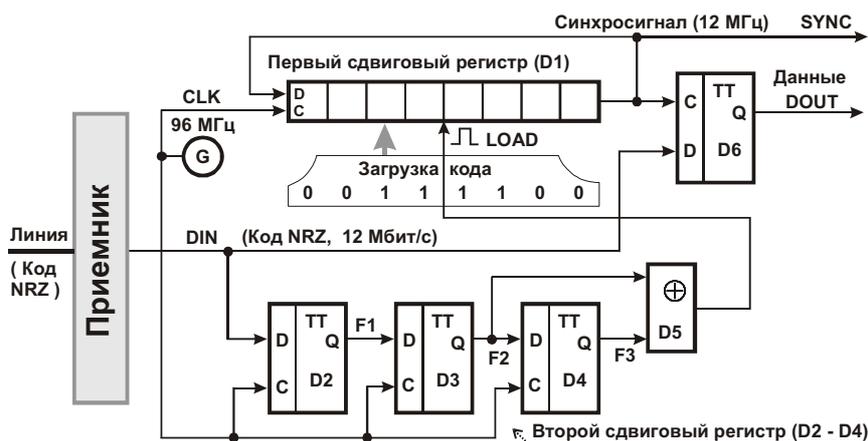


рис. 9.7. Схема выделения синхросигнала и данных из линии. Входной сигнал представлен кодом NRZ

Схема построена на основе двух сдвиговых регистров. Первый регистр D1 состоит из восьми включенных в “кольцо” D-триггеров с общими входами C синхронизации и объединенными входами LOAD управления асинхронной загрузкой. Циклический сдвиг кода в регистре D1 выполняется в отсутствие сигнала LOAD (LOAD = 0) по положительному фронту синхросигнала CLK от кварцевого генератора G. При наличии сигнала LOAD = 1 регистр D1 устанавливается в состояние 00111100 независимо от состояния сигнала CLK.

Частота сигнала CLK (96 МГц) в данном примере в восемь раз (по числу разрядов регистра D1) превышает скорость передачи данных по линии (12 Мбит/с). Сдвиговый регистр D1 фактически выполняет функцию делителя частоты на восемь, так как в нем в идеальной ситуации непрерывно циркулирует код 00111100: в течение четырех тактов сдвига SYNC = 1, затем в течение следующих четырех тактов SYNC = 0 и т. д. При этом фаза сигнала SYNC привязана к сигналу LOAD, который, в свою очередь, формируется при любом изменении состояния входного сигнала DIN.

Второй сдвиговый регистр выполнен на триггерах D2 – D4 с объединенными входами C синхронизации. Логический элемент Исключающее ИЛИ D5 формирует сигнал

лог. 1 при несовпадении сигналов на его входах. Рассмотрим процесс выделения синхросигналов и данных из сигнала DIN (рис. 9.8).

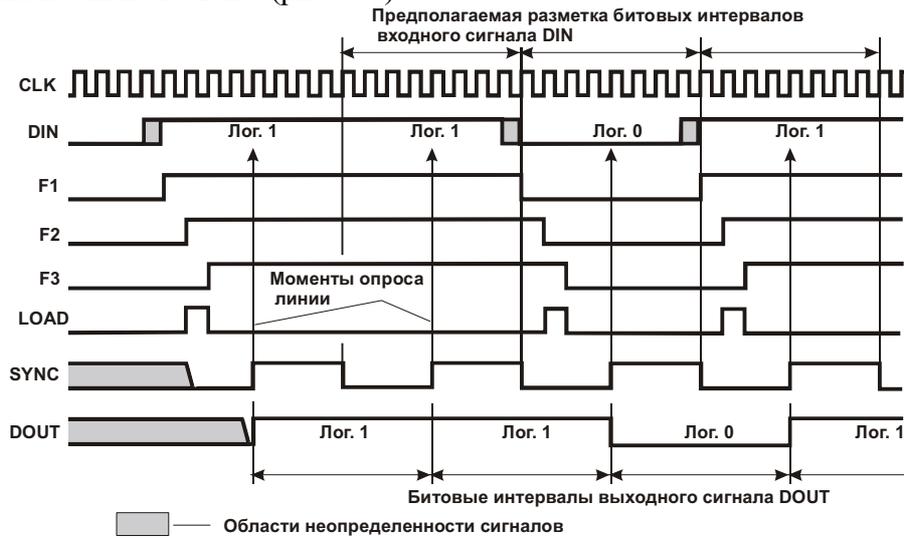


рис. 9.8. Временные диаграммы выделения синхросигналов и данных из линии

В исходном состоянии, когда нет изменений сигнала DIN (или когда эти изменения были слишком давно), сигналы SYNC и DOUT не определены. Изменение сигнала DIN (в данном примере переход из нулевого в единичное состояние) регистрируется по ближайшему положительному фронту сигнала CLK. Это проявляется в том, что в триггере D2 фиксируется сигнал $F1 = 1$.

Отметим, что в результате неблагоприятного сочетания сигналов на входах этого триггера, а именно, когда в момент поступления положительного фронта сигнала CLK сигнал DIN нестабилен, триггер D2 может сформировать кратковременный ложный сигнал. Иными словами, этот триггер может начать процесс “защелкивания”, но не завершить его. Однако кратковременный ложный сигнал не будет распространяться в последующие каскады сдвигового регистра благодаря общей цепи синхронизации. Полноценное защелкивание сигнала лог. 1 в триггере D1 окажется возможным только в следующем такте.

Сигнал $F1 = 1$ в двух последующих тактах продвигается к выходу второго сдвигового регистра. При этом в период неравнозначности сигналов F2 и F3 формируется сигнал загрузки сдвигового регистра D1 $LOAD = 1$, синхросигнал SYNC принимает нулевое значение. Через два такта после окончания сигнала $LOAD = 1$ формируется положительный фронт сигнала SYNC, гарантирующий правильную регистрацию бита данных DOUT в триггере D6. Начиная с этого момента, достигаются нужные соотношения между выходными и входными сигналами. Чтобы убедиться в правильности этих соотношений, рассмотрим временную диаграмму, приведенную на рис. 9.9.

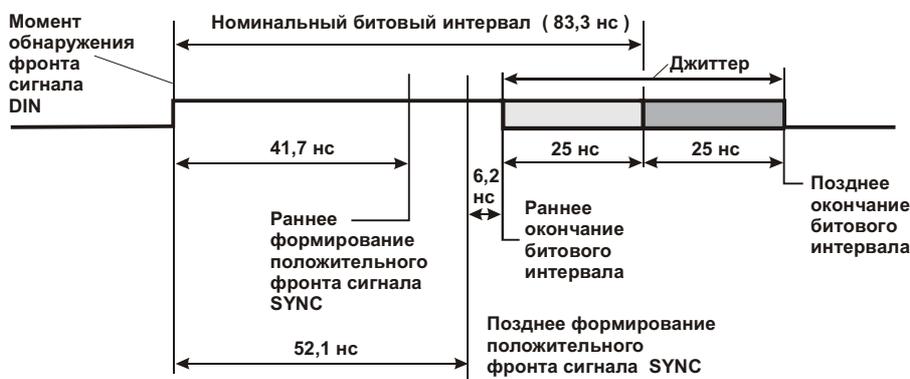


рис. 9.9. Временная диаграмма выделения бита данных с учетом джиттера и некоторой неопределенности момента формирования положительного фронта сигнала SYNC

прямоугольником) составит не два, а три периода сигнала CLK. Это, однако, учитывается допусками (см. рис. 9.9) и не нарушает работы устройства.

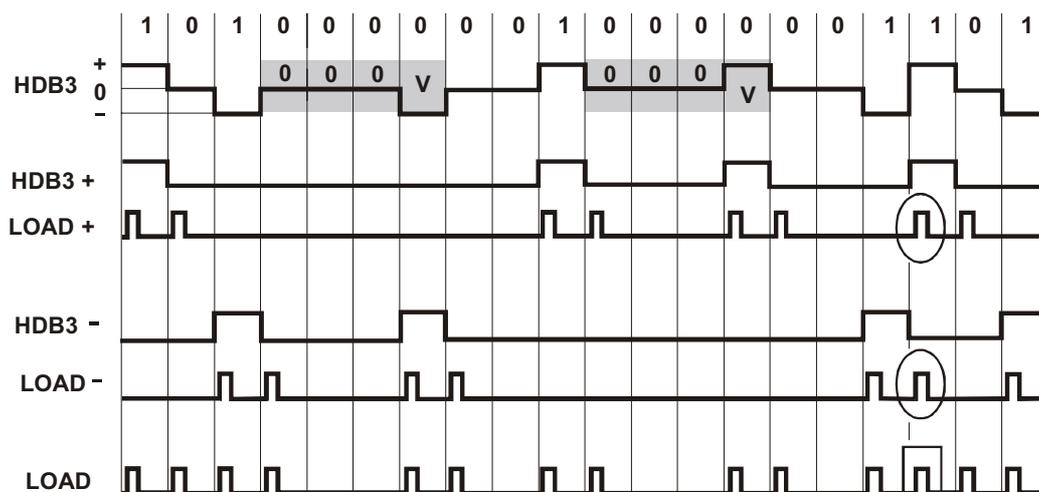


рис. 9.11. Временные диаграммы формирования сигнала LOAD

Выходные сигналы D+ и D- синхронизированы сигналом SYNC и соответствуют положительным и отрицательным импульсам на входе устройства. Дальнейшая обработка сигналов D+ и D- заключается в восстановлении исходной битовой последовательности (показанной в верхней части рис. 9.11) и может выполняться микропроцессором.

9.3. Одноконтурная и двухконтурные схемы выделения синхросигнала

В схеме, приведенной на рис. 9.12, выделение синхросигнала и данных из линии основано на использовании контура фазовой автоподстройки частоты.

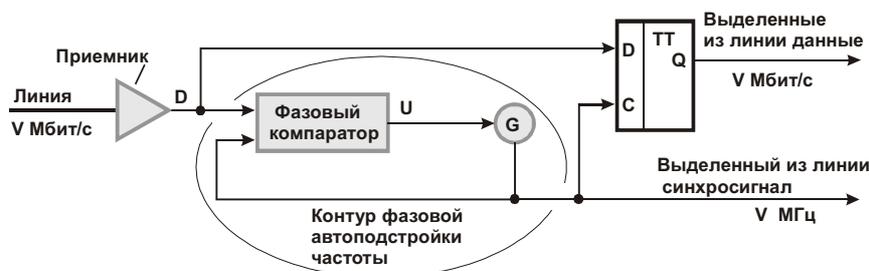


рис. 9.12. Схема, поясняющая принцип выделения синхросигнала и данных с помощью контура фазовой автоподстройки частоты

В установившемся режиме генератор G, управляемый напряжением U, формирует непрерывную последовательность синхроимпульсов. Их фронты (например отрицательные) с высокой точностью совпадают во времени с моментами изменения сигнала D на выходе приемника. Моменты изменений в общем случае нерегулярны и зависят от передаваемой последовательности битов. При правильном выборе способа кодирования линейного сигнала существует гарантия хотя бы одного изменения сигнала D в течение некоторого заданного промежутка времени. Например, при использовании кода B6ZS (см. п. 8.1.6) гарантируется наличие хотя бы одного импульса (двух изменений сигнала) в течение любых пяти последовательных битовых интервалов.

Фазовый компаратор оценивает степень совпадения фронтов сигналов на своих входах и в зависимости от результата формирует некоторое управляющее напряжение U. На-

пример, при хорошем совпадении фронтов, когда коррекции не требуется, $U = 2,5$ В. При необходимости ускорить или замедлить темп выработки синхроимпульсов напряжение повышается или снижается. Диапазон изменения напряжения может составлять $(0,5 - 4,5)$ В. Фазовый компаратор содержит фильтр низких частот и поэтому обладает достаточной инерционностью, чтобы не реагировать на возможные мешающие факторы: “дрожание” фазы входного сигнала, импульсные помехи в линии и т. п.

В идеальном случае положительные фронты выделенного из линии синхросигнала соответствуют центрам битовых интервалов, поэтому для выделения данных используется D-триггер.

Контур фазовой автоподстройки частоты представляет собой достаточно “тонкий инструмент”. Этот контур должен иметь очень узкую полосу захвата – порядка долей герца. Иными словами, для вхождения устройства в синхронизм с линейным сигналом нужно, чтобы исходная частота генератора G почти не отличалась от входной частоты. В противном случае на входах фазового компаратора будут наблюдаться случайные фазовые соотношения между сигналами, что не позволит выбрать нужное направление коррекции частоты. Из этого следует, что генератор G должен быть высокостабилизированным, но это противоречит требованию его хорошей управляемости со стороны фазового компаратора.

Чтобы устранить данное противоречие, применяют схемы с двумя контурами автоподстройки. Одна из таких схем [54] приведена на рис. 9.13.

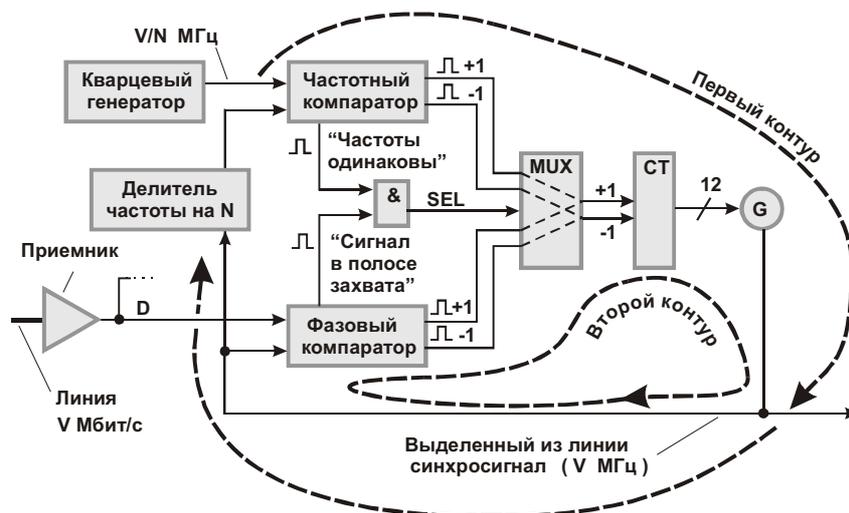


рис. 9.13. Двухконтурная схема выделения синхросигнала из линии - первый вариант

Первый контур предназначен для предварительной настройки генератора G на номинальную частоту синхросигнала. После завершения такой настройки вместо первого контура включается второй, аналогичный рассмотренному ранее (см. рис. 9.12).

В данном примере частота генератора G регулируется кодом с выхода 12-разрядного двоичного счетчика СТ. При нормальной работе устройства код на выходе счетчика примерно соответствует середине диапазона счета и может изменяться в незначительных пределах. Содержимое счетчика увеличивается или уменьшается на единицу младшего разряда при поступлении импульса на вход $+1$ или -1 . В отсутствие импульсов код в счетчике остается неизменным. Импульсы вырабатываются либо частотным, либо фазовым компаратором в зависимости от того, какой контур активизирован. При $SEL = 0$ включен первый контур, при $SEL = 1$ – второй.

Кварцевый генератор формирует на первом входе частотного компаратора сигнал частотой V/N , где V – скорость передачи данных по линии, $N \geq 2$ – целое число. Сигнал той же частоты поступает на второй вход этого компаратора. Снижение сравниваемых час-

тот позволяет упростить схему частотного компаратора и снизить потребляемую им мощность.

Предположим, что в исходном состоянии в линии присутствует полезный сигнал, но синхронизация не установлена; генератор G формирует сигнал, имеющий заметное отклонение (например, на 2 Гц) от нужной частоты. В этой ситуации оба компаратора обнаруживают неблагоприятные сочетания сигналов на входах, логический элемент I получает от обоих компараторов сигналы лог. 0. Так как $SEL = 0$, мультиплексор MUX транслирует на входы счетчика CT сигналы с выходов частотного компаратора.

Частотный компаратор определяет знак и величину частотной ошибки и формирует последовательность импульсов на одном из выходов. Код в счетчике CT изменяется в направлении уменьшения ошибки. В конечном счете ошибка уменьшается до допустимых пределов, частотный компаратор формирует сигнал “Частоты одинаковы”. Одновременно с этим или с небольшой задержкой фазовый компаратор формирует признак “Сигнал в полосе захвата”, означающий, что фазовый компаратор выявил закономерность между моментами изменения сигналов на обоих входах и поэтому способен корректировать ее в нужном направлении.

В результате совпадения указанных условий формируется сигнал $SEL = 1$, мультиплексор переключается на трансляцию сигналов с фазового компаратора. Далее осуществляется точная подстройка генератора, при которой выходной сигнал привязывается к входному по фазе. Это позволяет выделять данные с помощью D -триггера, как было показано ранее (см. рис. 9.12).

При нормальной работе устройства, как уже отмечалось, включен второй контур управления генератором G . Но при потере входного сигнала в линии частота нестабилизированного генератора G , оставшегося без управления, начинает непрерывный дрейф в ту или иную сторону. Так как во втором контуре нет опорной частоты, фазовый компаратор не посылает корректирующие импульсы в счетчик CT . Поэтому если, например, в результате “замирания” сигнала в линии потеряна группа из 70 битов, то дрейф приведет к выходу частоты генератора G из полосы захвата фазового компаратора. Тогда после возобновления правильной последовательности сигналов в линии вся описанная ранее процедура грубой и точной настройки генератора G будет повторена, что связано с неоправданно большими затратами времени.

Схема, приведенная на рис. 9.14 [54], обладает лучшей устойчивостью к “замираниям” сигнала в линии. Это связано с тем, что нестабилизированный генератор $G2$ контура фазовой автоподстройки частоты ни при каких обстоятельствах не остается без управления.

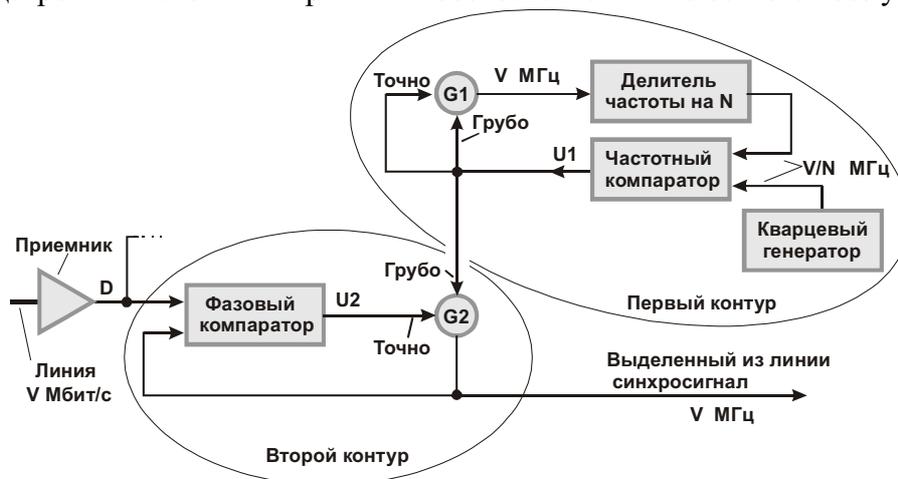


рис. 9.14. Двухконтурная схема выделения синхросигнала из линии - второй вариант

Схема построена на основе двух нестабилизированных генераторов $G1$ и $G2$. Эти генераторы размещены в одном кристалле интегральной схемы и имеют одинаковую топо-

логию. Поэтому они обладают очень близкими электрическими параметрами, что существенно для данного решения. Генератор G1 (G2) содержит аналоговые входы грубой и точной подстройки частоты. При нормальной работе устройства регулирующие напряжения U_1 и U_2 находятся примерно в середине диапазона регулировки (например близки 2,5 В при диапазоне (0,5 – 4,5) В). Входы грубой и точной подстройки генератора G1 объединены.

В первом, автономном контуре управления отслеживается частота кварцевого генератора, так что генератор G1 формирует стабилизированную частоту V , близкую скорости передачи данных в линии. Поскольку генераторы G1 и G2 имеют практически одинаковые электрические параметры, генератор G2 также настроен на эту частоту по входу грубой подстройки. В результате независимо от уровня напряжения U_2 частота сигнала на выходе генератора G2 отличается от нужной всего лишь на доли герца и в любой ситуации не выходит за пределы полосы захвата фазового компаратора!

Таким образом, фазовый компаратор всегда готов к работе, даже после длительных “замираний” сигнала в линии. Исключен дрейф частоты в отсутствие сигнала в линии, уменьшено время вхождения в синхронизм после обнаружения этого сигнала.

Генераторы G1 и G2 выполнены на МДП-транзисторах по схеме симметричного мультивибратора (рис. 9.15).

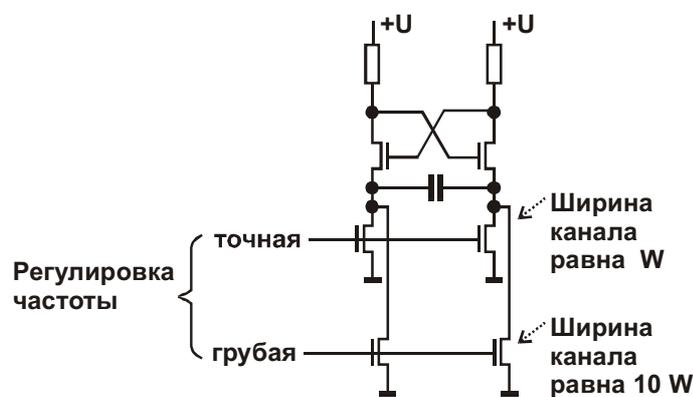


рис. 9.15. Схема генератора

Четыре нижних транзистора используются в качестве двух сдвоенных переменных резисторов, с помощью которых можно регулировать частоту сигнала. Ширина канала транзисторов грубой регулировки частоты (нижняя пара) в 10 раз превышает ширину канала транзисторов точной регулировки.

Поэтому сопротивления каналов и соответствующие диапазоны регулировки частоты отличаются примерно в 10 раз.

9.4. Схема выделения синхросигнала с компенсацией нагрузочной емкости

Выделенный из линии синхросигнал может поступать к нескольким микросхемам, размещенным на одной или нескольких печатных платах устройства, например мультиплексора. При этом паразитная емкость проводника, несущего синхросигнал, и входные емкости элементов – приемников этого сигнала увеличивают длительности его фронтов. Это эквивалентно задержке синхросигнала, нарушающей его точную привязку к данным, поэтому задержка должна быть скомпенсирована. На рис. 9.16 приведен пример схемы компенсации задержки синхросигнала [56].

Схема содержит генератор G с фазовой автоподстройкой частоты, микропроцессор, группу электронных ключей SW, набор компенсирующих конденсаторов с емкостями C, 2C, 4C, 8C, два усилителя сигнала SYNC и два фазовых компаратора EQ1 и EQ2. Усилители выполнены в одной интегральной схеме и поэтому имеют практически одинаковые электрические характеристики. Каждый ключ управляется соответствующим сигналом из группы сигналов Q, поступающих из микропроцессора. Входные данные DATA поступают из линии и в неявном виде содержат синхросигнал. В цепи передачи выделенного из линии синхросигнала CLK присутствует заранее не известная паразитная емкость CX.

В зависимости от комбинации управляющих сигналов Q к цепи передачи сигнала CL подключается та или иная подгруппа компенсирующих конденсаторов. В установленном

режиме их суммарная емкость с помощью микропроцессора подбирается примерно равной паразитной емкости C_X . При $Q = 0000$ все конденсаторы отключены, при $Q = 1111$ все конденсаторы подключены и их суммарная емкость составляет $15C$, где C – емкость, определяющая шаг подбора. Максимальная компенсирующая емкость ($15C$) должна превышать максимальную паразитную емкость C_X .

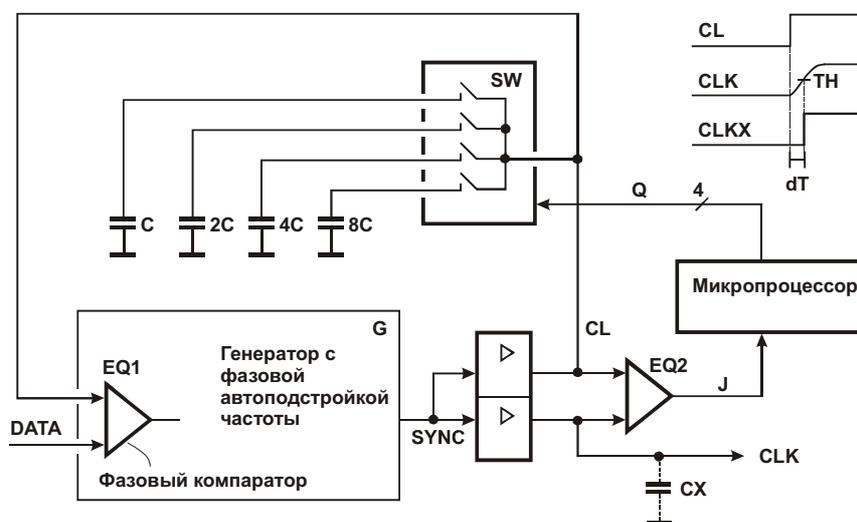


Рис. 9.16. Схема компенсации задержки выделенного синхросигнала

Предположим, что в исходном состоянии $Q = 0$, компенсирующие конденсаторы изолированы от цепи CL передачи сигнала обратной связи генератора с фазовой автоподстройкой частоты. Предположим также, что паразитная емкость C_X оказывает заметное влияние на длительность фронтов сигнала CLK . В этой ситуации сигналы CL и $DATA$ находятся в правильных фазовых соотношениях, которые отслеживаются компаратором $EQ1$ генератора G . Однако сигнал CLK имеет пологий фронт, как показано на рисунке. Порог TH срабатывания приемников этого сигнала лежит примерно в середине между уровнями лог. 0 и 1. Поэтому момент срабатывания приемников смещен относительно положительного фронта сигнала CL на время dT . Иными словами, сигнал CLK эквивалентен некоторому сигналу $CLKX$, полученному из сигнала CL путем его сдвига на время dT .

Задача заключается в минимизации этого времени. Фазовый компаратор $EQ2$ реагирует на отрицательный сдвиг фазы сигнала CL относительно сигнала CLK выдачей на вход микропроцессора признака $J = 0$. Микропроцессор следит за состоянием этого признака и последовательно наращивает код Q , что приводит к последовательному увеличению компенсирующей емкости. Фронт сигнала CL начинает приближаться по форме к фронту сигнала CLK . В некоторый момент фазовое соотношение между этими сигналами изменяет знак, признак J переходит в состояние лог. 1. Обнаружив этот переход, микропроцессор фиксирует код Q на достигнутом уровне.

В дальнейшем микропроцессор периодически, например один раз в секунду, подбирает величину компенсирующей емкости таким образом, чтобы компаратор $EQ2$ находился на грани срабатывания.

Рассмотренное решение позволяет автоматически отслеживать изменения паразитной емкости C_X при изменении температуры, влажности, при отключении части нагрузки и т. п., что обеспечивает надежную привязку выделенного из линии синхросигнала к данным.

9.5. Распознавание синхросигнала в искаженном входном сигнале с помощью симметрирующего коррелятора

Корреляторы обычно применяют в системах распознавания образов для сопоставления и оценки сходства эталонного и искомого изображений. Далее рассмотрено применение коррелятора для решения задачи распознавания синхросигнала в искаженном входном сигнале. Точнее, задача заключается в отыскании в проходящем битовом потоке “очертаний” синхросигнала с заданными параметрами и привязки восстановленного синхросигнала к центрам битовых интервалов.

В примере, приведенном на рис. 9.17, коррелятор входит в состав схемы распознавания синхросигнала и содержит 16-разрядные регистры RG1, RG2, компараторы EQ0 – EQ15, сумматор ADD. Входные данные DATA “мелкими шагами” (составляющими доли битового интервала) проходят через сдвиговой регистр RG1. В регистре RG2 зафиксирован эталонный код для сравнения с кодом в регистре RG1. Поразрядное сравнение кодов выполняется компараторами EQ0 – EQ15. Компаратор формирует сигнал лог. 1 при совпадении битов на его входах. Сумматор ADD подсчитывает число совпадений битов в регистрах и выдает результат SUM на компаратор для сравнения с некоторым порогом TH. Если $SUM > TH$, то $Y = 1$, в противном случае $Y = 0$. Сигнал $Y = 1$ подтверждает близость кодов в регистрах RG1 и RG2 и представляет собой метку времени для привязки к ней искомого синхросигнала. Сигнал Y с выхода коррелятора поступает на вход генератора PLL с фазовой автоподстройкой частоты, который восстанавливает синхросигнал CL.

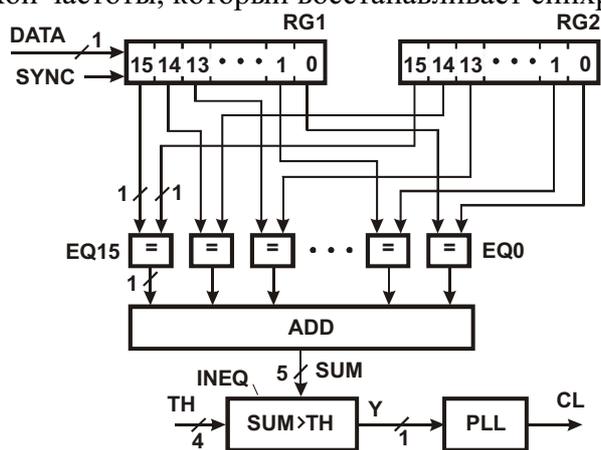


Рис. 9.17. Схема распознавания синхросигнала в потоке данных – первый вариант

Рассмотрим процесс распознавания синхросигнала более детально. На вход коррелятора поступают два сигнала: DATA и SYNC. Первый принят из линии; он в неявном виде содержит синхросигнал CL, который следует восстановить. Второй формируется местным автономным генератором; его частота в восемь раз превышает частоту синхросигнала CL. Для большей наглядности описания работы коррелятора предположим, что сигнал DATA представлен кодом NRZ (см. 8.1.2) вида ...010101... Каждый бит этого кода представлен на рис. 9.18, а восемью одноименными отсчетами (цепочками из лог. 0 или 1).

Сигнал DATA проходит через сдвиговой регистр RG1 с шагом, равным 1/8 части битового интервала кода NRZ. Такое продвижение удобно рассматривать как просмотр сигнала DATA через движущееся окно. Содержимое окна сравнивается с эталоном 000011111110000 из регистра RG2. Этот эталон рассчитан на появление в коде NRZ комбинации битов “010”. Задача заключается в фиксации момента наибольшей схожести кодов в регистрах RG1 и RG2. Как показано в первой строке Таблица 9.1 табл. 9.1, в некотором такте (n-8) сигнала SYNC код в регистре RG1 противоположен эталонному, поэтому нет ни одного совпадения битов, следовательно $SUM = 0$.

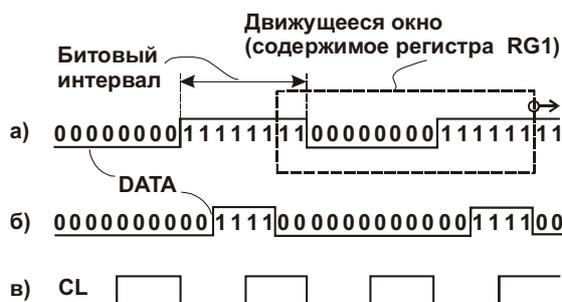


Рис. 9.18. Временные диаграммы, поясняющие принцип действия коррелятора: *а* – сигнал DATA без искажений; *б* – сигнал DATA с искажениями; *в* – восстановленный синхросигнал

Таблица 9.1 табл. 9.1

Процесс формирования сигнала SUM при обработке сигнала DATA, показанного на рис. 9.18, *а*, схемой, приведенной на рис. 9.17

Номер такта	Код в регистре RG1	Значение сигнала SUM
n-8	1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1	0
n-7	1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1	2
n-6	1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1	4
n-5	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1	6
n-4	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0	8
n-3	0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0	10
n-2	0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0	12
n-1	0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0	14
n	0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0	16
n+1	0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0	14
n+2	0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0	12
n+3	0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0	10
n+4	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1	8

В следующем такте (n-7) совпадения наблюдаются в разрядах 3 и 11, поэтому $SUM = 2$. По мере продвижения кода в регистре RG1 наблюдается увеличение числа совпадений, и в такте с номером *n* коды полностью совпадают, сигнал SUM достигает максимального значения, равного 16. В последующих тактах код в регистре RG1 все более отличается от эталонного, сигнал SUM возвращается к нулевому значению и т. д. В отсутствие искажений сигнала DATA порог ТН можно выбрать равным 15. Тогда положительный фронт сигнала Y с хорошей точностью попадает в середину битового интервала, в котором передается единичный сигнал. Снижая порог, можно уменьшить требуемую степень схожести кодов в регистрах RG1 и RG2, однако при этом снижается точность попадания сигнала Y в середину битового интервала.

Генератор PLL с фазовой автоподстройкой частоты формирует сигнал CL так, чтобы его положительные фронты совпадали с положительными фронтами сигнала Y или запаздывали на заданный интервал, что необходимо при снижении порога ТН. Сигнал Y в общем случае формируется с нерегулярным периодом, зависящим от характера передаваемых данных. Передающая сторона должна обеспечивать гарантированное появление хотя бы одной эталонной комбинации (сочетания битов “010” в коде NRZ) на протяжении, например, любых двадцати битовых интервалов. Напомним, что в данном примере рассматривается последовательность ...010101..., обеспечивающая формирование сигнала Y в каждом втором битовом интервале.

Выраженный максимум сигнала SUM, однако, быстро размывается при типичном искажении входного сигнала DATA, показанном на рис. 9.18, б. Такое искажение связано, например, со смещением порога срабатывания приемника сигнала из линии. Действительно, импульсы в линии имеют не прямоугольную, а трапецеидальную форму. Если порог срабатывания приемника смещен в сторону повышения, то сигналы лог. 1 окажутся более короткими, чем сигналы лог. 0, и наоборот. Как следует из рисунка, сигнал лог. 1 представлен четырьмя (а не восемью) единичными отсчетами в регистре RG1, хотя передается тот же самый, что и ранее, код NRZ: ...010101... . Работа коррелятора в этих условиях поясняется табл. 9.2.

Таблица 9.2 табл. 9.2

Процесс формирования сигнала SUM при обработке сигнала DATA, показанного на рис. 9.18, б, схемой, приведенной на рис. 9.17

Номер такта	Код в регистре RG1	Значение сигнала SUM
n-8	1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1	2
n-7	1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1	2
n-6	1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1	2
n-5	0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1	5
n-4	0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0	8
n-3	0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0	10
n-2	0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0	12
n-1	0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0	12
n	0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0	12
n+1	0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0	12
n+2	0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0	12
n+3	0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0	10
n+4	0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0	8

Как следует из таблицы, сигнал SUM имеет размытый максимум, равный 12; он наблюдается на протяжении пяти соседних тактов. Это не позволяет с хорошей точностью выделить момент, соответствующий середине битового интервала. Этот недостаток устранен в схеме симметрирующего коррелятора, показанной на рис. 9.19, рис. 9.20 [32].

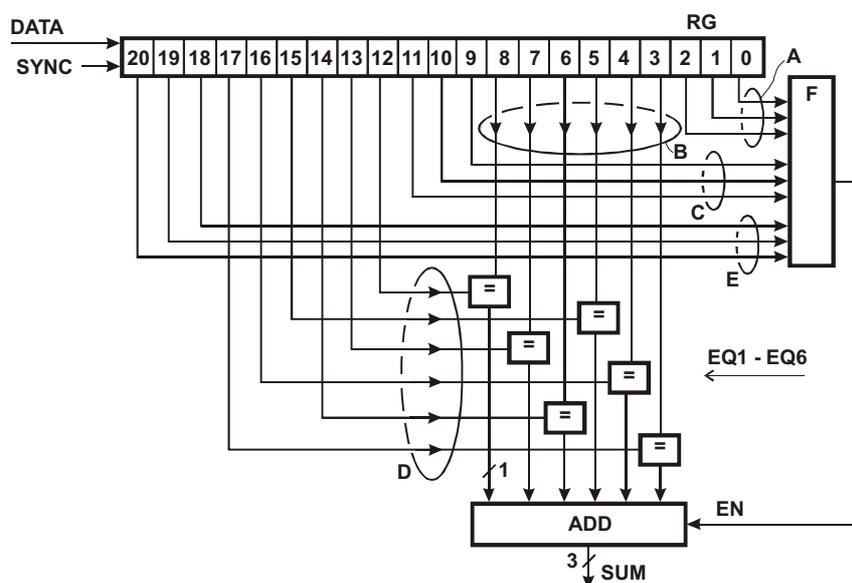


Рис. 9.19. Схема распознавания синхросигнала в потоке данных – второй вариант

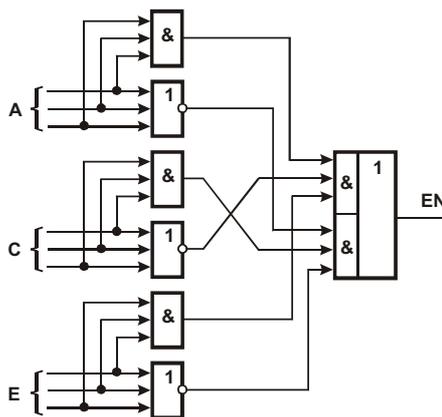


рис. 9.20. Схема логического блока F (см. рис. 9.19)

В схеме (рис. 9.19), в отличие от предыдущей, отсутствует регистр для хранения эталона (!). Последовательность отсчетов входного сигнала продвигается через регистр RG и анализируется логическим блоком F, компараторами EQ1 – EQ6 и сумматором ADD. Принцип действия коррелятора поясняется рис. 9.21.

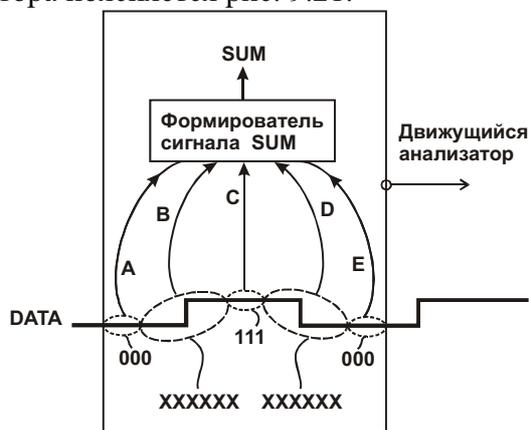


рис. 9.21. Схема, поясняющая принцип действия коррелятора (см. рис. 9.19)

При формировании сигнала SUM анализируются группы отсчетов A, B, C, D, E. Прежде всего проверяется одновременное выполнение трех условий: $A = 000$, $C = 111$, $E = 000$ или $A = 111$, $B = 000$, $C = 111$. Если условия выполнены, то это означает, что положение движущегося анализатора примерно соответствует показанному на рисунке или аналогичному, смещенному на один битовый интервал сигнала NRZ. В этом случае анализ продолжается. Теперь сопоставляются группы отсчетов B и D, представленные на рисунке знаками "X". Определяется степень симметрии отсчетов в этих группах. Если сигнал NRZ не искажен (как показано на рисунке), то $B = 000111$, $D = 111000$. Сравнение этих групп с учетом зеркальной перестановки разрядов в одной из них показывает максимально возможное число совпадений, равное шести. Если хотя бы одно названных условий ($A = 000$, $C = 111$, $E = 000$ или $A = 111$, $B = 000$, $C = 111$) не выполнено, то анализ прекращается и сигнал SUM устанавливается в ноль.

Предположим, что длительность сигналов лог. 1 в коде NRZ чуть уменьшилась за счет соответствующего увеличения длительности сигналов лог. 0. Тогда получим, например, такое соотношение кодов B и D: $B = 000011$, $D = 110000$. Эти коды вновь оказались симметричными, несмотря на асимметрию сигнала DATA, т. е. разрешающая способность коррелятора не уменьшилась при обработке искаженного сигнала!

Рассмотрим работу коррелятора более подробно. Код DATA проходит через сдвиговый регистр RG под управлением синхросигнала SYNC, частота которого в девять раз превышает скорость передачи данных по линии. Группы разрядов B и D с помощью компара-

торов EQ1 – EQ6 поразрядно сравниваются в “зеркальном” порядке относительно центрального (десятого) разряда регистра: разряд 8 сравнивается с разрядом 12, разряд 7 – с разрядом 13 и т. д. Группы разрядов А, С и Е анализируются логическим блоком F. Сигнал EN = 1 формируется при A = E = 000 и C = 111 или при A = E = 111 и C = 000. Сумматор ADD при EN = 1 подсчитывает число единичных сигналов на входах (все входы имеют равные единичные веса). Если EN = 0, то работа сумматора запрещена, на его выходе постоянно присутствует нулевой код.

Прохождение неискаженного и искаженного сигналов DATA через коррелятор поясняется Таблица 9.3 табл. 9.3 и Таблица 9.4 табл. 9.4.

Таблица 9.3 табл. 9.3

Процесс формирования сигнала SUM при обработке неискаженного сигнала DATA схемой, приведенной на рис. 9.19

Номер такта	Код в регистре RG	Значение сигнала SUM	Значение сигнала EN
n-3	1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1	0	1
n-2	1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1	2	1
n-1	1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1	4	1
n	1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1	6	1
n+1	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1	4	1
n+2	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1	2	1
n+3	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1	0	1

Таблица 9.4 табл. 9.4

Процесс формирования сигнала SUM при обработке искаженного сигнала DATA схемой, приведенной на рис. 9.19

Номер такта	Код в регистре RG	Значение сигнала SUM	Значение сигнала EN
n-3	1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1	0	0
n-2	1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	0	0
n-1	1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	4	1
n	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1	6	1
n+1	1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1	4	1
n+2	1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1	0	0
n+3	1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1	0	0

Из сопоставления таблиц следует, что, несмотря на внесенные искажения, связанные с нарушением симметрии сигналов лог. 0 и 1 в коде NRZ, коррелятор по-прежнему формирует хорошо выраженный сигнал опознания центра битового интервала. При этом распознаются две (а не одна) комбинации битов в коде NRZ: 010 и 101. Это увеличивает частоту следования импульсов привязки на входе генератора с фазовой автоподстройкой частоты и способствует повышению качества восстановленного синхросигнала.

9.6. Шифратор и дешифраторы кода Манчестер-II

Рассмотрим процессы формирования и последующей дешифрации линейного сигнала типа Манчестер-II. В результате дешифрации из линейного сигнала извлекаются данные и сопровождающие их синхроимпульсы. Напомним о некоторых примечательных свойствах кода Манчестер-II.

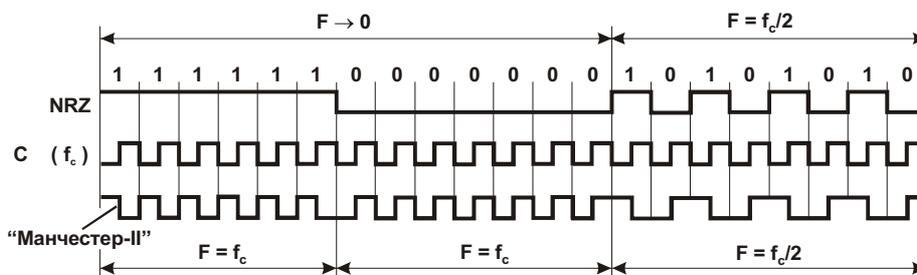


рис. 9.22. Сравнение частотных свойств сигналов NRZ и Манчестер-II

Сигнал Манчестер-II (рис. 9.22) можно получить суммированием по модулю два (с помощью логического элемента Исключающее ИЛИ) сигналов NRZ и C (данных и синхросигнала). Основная особенность этого сигнала состоит в том, что он никогда не “замирает”. При передаче длинной цепочки единиц частота сигнала Манчестер-II совпадает с частотой f_c синхросигнала C. То же наблюдается и при передаче длинной цепочки нулей. Поэтому даже в том случае, когда частота исходного сигнала NRZ стремится к нулю ($F \rightarrow 0$), частота сигнала Манчестер-II остается постоянной (моментами смены фазы сигнала пренебрегаем). При передаче чередующейся последовательности нулей и единиц частота сигнала Манчестер-II совпадает с частотой сигнала NRZ.

Идея поддержания синхронизма между приемником и передатчиком поясняется рис. 9.23. На рисунке показаны три битовых интервала T. Предположим, что приемник уже синхронизирован с передатчиком. Тогда, обнаружив в середине первого битового интервала фронт сигнала (он обязательно должен быть: положительный либо отрицательный), приемник запускает внутренние часы.

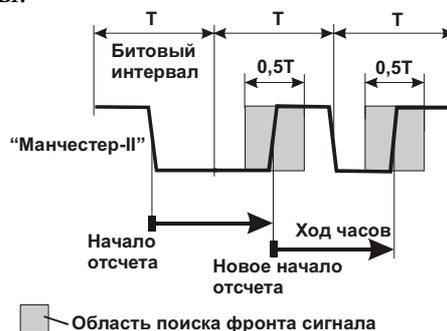


рис. 9.23. Временная диаграмма передачи кода Манчестер-II. Обеспечивается коррекция хода часов приемника при передаче каждого бита

Через интервал времени, равный T, обнаруживается положительный фронт сигнала (передается лог. 0). Приемник вновь запускает часы (“забывая” старые их показания), т. е. переносит начало отсчета времени на один битовый интервал вправо. Далее процесс повторяется. Как видим, не происходит накопления ошибок, связанных с отсчетом времени.

Поиск очередного фронта сигнала в середине следующего битового интервала ведется в некотором доверительном интервале. Из “геометрии” рисунка следует, что область поиска фронта сигнала может достигать $0,5T$. Это означает, что независимо от длины цепи передаваемых битов допустимое рассогласование хода часов передатчика и приемника может приближаться к $\pm 25\%$ (другими дестабилизирующими факторами пренебрегаем). Для сравнения: допустимое рассогласование хода часов передатчика и приемника при использовании старт-стоповой посылки (интерфейсы RS-232, RS-422 и другие) составляет всего 4%.

Перейдем к конкретным решениям. Рассмотрим упрощенную схему передачи данных в коде Манчестер-II (рис. 9.24). В схеме отсутствует гальваническая развязка с линией, не

показаны элементы электрического согласования и т. п.; тем не менее, она работоспособна при условии, что линия связи достаточно короткая.



рис. 9.24. Упрощенный пример построения канала связи для передачи данных в коде Манчестер-II (рассмотрено одно направление передачи)

Как уже отмечалось, шифратор кода Манчестер-II может быть выполнен на двухвходовом элементе Искключающее ИЛИ (рис. 9.25).

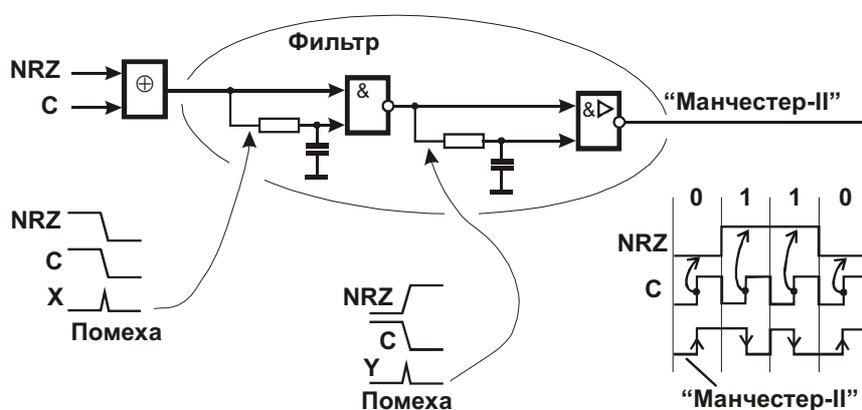


рис. 9.25. Шифратор кода Манчестер-II

Фильтр предназначен для подавления кратковременных импульсов, которые могут возникнуть из-за неидеального совпадения отрицательного фронта сигнала C с отрицательным или положительным фронтом сигнала NRZ. Дешифраторы кода Манчестер-II более сложны по логической структуре и чаще всего в явном виде не используют описанную ранее идею перезапуска часов в каждом битовом интервале (см. рис. 9.23).

Дешифратор кода Манчестер-II, первый вариант

Этот вариант [22] предложен в 1980 году, и к настоящему времени представляет, скорее, "познавательный", чем практический интерес в силу наглядности решения задачи дешифрации, пусть и с повышенными затратами аппаратуры.

Дешифратор кода Манчестер-II (рис. 9.26, а) [22] содержит формирователь F импульсов, счетный триггер СТ и D-триггер. Как следует из временной диаграммы, приведенной на рис. 9.26, б, отрицательные импульсы j на выходе формирователя импульсов возникают всякий раз, когда сигнал Манчестер-II изменяет значение (0 → 1 или 1 → 0). Сигнал k восполняет "недостающие" импульсы j.

"Интеллект" дешифратора, по существу, скрыт в структуре формирователя F импульсов. Поясним сказанное. Читатель может без труда нарисовать временную диаграмму сигнала k, используя только диаграмму сигнала j. Действительно, на временной диаграмме сигнала j визуально прослеживаются промежутки, в течение которых частота следования отрицательных импульсов снижается из-за их "недостаки". В эти промежутки и следует формировать импульсы k. Один из вариантов построения формирователя приведен в [22].

Так как импульс k поступает на вход S установки единицы счетного триггера СТ, то в момент t_0 этот триггер безусловно перейдет в единичное состояние, и в дальнейшем сигнал

C^* на его инверсном выходе будет в точности повторять сигнал C от источника данных (разумеется, с некоторой задержкой).

Начиная с момента t_1 , т. е. по прошествии одного периода тактовых импульсов от момента t_0 , код NRZ^* , снимаемый с выхода триггера ТТ, полностью совпадает с исходным кодом NRZ (с точностью до задержки передачи). Таким образом, для того чтобы заставить приемник войти в синхронизм с передатчиком, достаточно на линии NRZ создать переход сигнала из 0 в 1. Последующая цепочка битов любой длины, передаваемая по линии NRZ , будет в точности повторена на линии NRZ^* приемника. Это же относится и к синхросигналам.

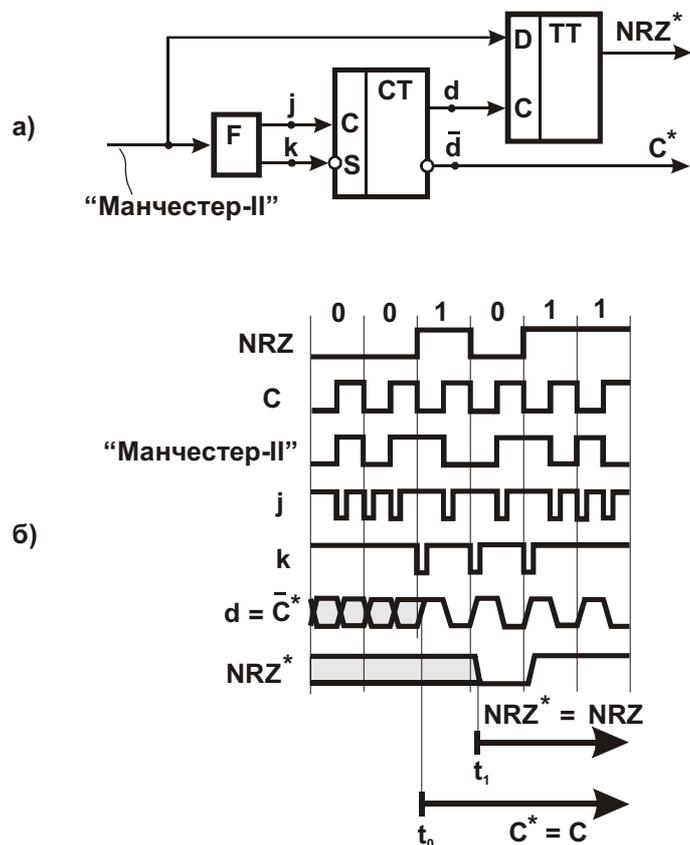


рис. 9.26. Дешифратор кода Манчестер-II (первый вариант):
а – схема; б – временные диаграммы

Недостатком рассмотренной схемы дешифратора является сложность реализации формирователя F импульсов. Аналоговое решение формирователя (использующее времязадающие RC-цепи) не обеспечивает высокой стабильности его работы при изменении температуры и питающего напряжения. Поэтому предпочтительна чисто цифровая реализация схемы формирователя.

Идея его построения может быть такой. Как было показано, формирователь должен оценивать своевременность появления импульсов j и восполнять недостающие импульсы. Для этого он должен оперировать достаточно короткими интервалами времени, составляющими, например, $1/8$ или $1/16$ части битового интервала кода Манчестер-II.

Внутренний таймер формирователя (на рисунке не показан) должен отсчитывать время от окончания очередного импульса j или k и ждать появления очередного импульса j в некотором доверительном интервале. Если этот импульс приходит, то таймер перезапускается, и цикл ожидания повторяется. Если же ожидаемого импульса j нет, то вместо него формируется импульс k . Он может размещаться не точно на том месте, где ожидался импульс j , а с некоторой задержкой, связанной с ожиданием.

Таким образом, формирователь импульсов имеет довольно сложную логическую структуру, при этом тактовая частота его работы должна значительно превышать скорость передачи данных по линии.

Дешифратор кода Манчестер-II, второй вариант

Схема дешифратора [78], приведенная на рис. 9.27, по сравнению с ранее рассмотренной, более проста и оперирует более низкой частотой внутреннего генератора G синхросигналов. Генератор формирует периодический сигнал W опроса линии и приема дешифрованных сигналов в выходной регистр RG . Тактовая частота генератора в данном случае превышает скорость передачи данных по линии в $N = 3,33$ раза. Это означает, что при скорости передачи данных по линии, равной 15 Мбит/с, частота сигнала W с выхода генератора G равна 50 МГц. С учетом возможного “дрожания фронтов” (джиттера) входного сигнала соотношение частот может быть несколько увеличено, например, до уровня $N = 4,0$. Генератор G работает автономно, т. е. он не синхронизирован входным сигналом с линии. Точность и стабильность генератора могут быть сравнительно низкими, важно только, чтобы максимальное отклонение частоты генератора в сторону ее снижения не приводило к уменьшению фактического значения параметра N ниже расчетного уровня.

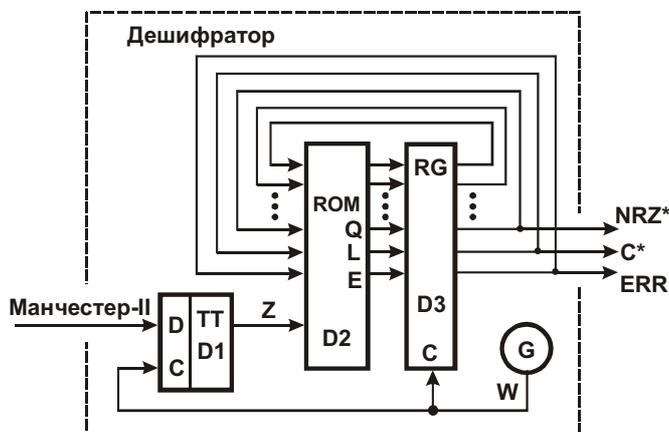


рис. 9.27. Дешифратор кода Манчестер-II (второй вариант)

Дешифратор выполнен на основе “конечного автомата” – комбинационной схемы $D2$ (в данном примере реализованной на основе ПЗУ) с подключенным к ее выходам регистром RG . Три разряда этого регистра выделены для хранения текущего состояния выходных сигналов NRZ^* и C^* , а также сигнала ERR обнаружения ошибки кодирования входного сигнала. Эти и другие разряды регистра RG отображают текущее состояние дешифратора и по цепям обратной связи передаются на адресные входы ПЗУ. По положительному фронту сигнала W с выхода генератора G в регистре RG фиксируется очередной код, а в триггер $D1$ заносится результат очередного опроса состояния линии. Сигнал Z с выхода этого триггера, также как и сигналы обратной связи, поступает на адресные входы ПЗУ.

При дешифрации кода Манчестер-II устройство проходит через множество состояний, каждое из которых зависит как от предыстории (от состояния регистра RG), так и от результата последнего опроса сигнала в линии (этот результат зафиксирован в триггере $D1$ в виде сигнала Z). В каждом такте сигнала W происходит вычисление новых значений сигналов NRZ^* и C^* , а также сигналов обратной связи, определяющих направления перехода к последующим состояниям. Общее число состояний дешифратора зависит от параметра N , от требуемых возможностей противодействия влиянию джиттера входного сигнала, от заданного уровня детализации ошибок в линии, если требуется их распознавание. Число состояний может лежать, например, в диапазоне от 32 до 256. Вместо ПЗУ может использоваться программируемая логическая матрица или “жесткая” структура из логических элементов.

Сигнал Манчестер-II (рис. 9.28), как было показано ранее (см. рис. 9.25), формируется суммированием по модулю два исходных сигналов NRZ и C (данных и синхросигнала) и по-

ступает на D-вход триггера D1 (см. рис. 9.27). Моменты опроса сигнала Манчестер-II не обязательно совпадают с его установившимися значениями, так как генератор G не синхронизирован сигналом с линии. Иными словами, положительный фронт сигнала опроса W может практически совпасть с положительным или отрицательным фронтом сигнала Манчестер-II. В этом случае в триггер D1 с равной вероятностью будет принят сигнал лог. 0 или 1.

Для определенности предположим, что в таких “спорных” ситуациях принятые в триггер D1 сигналы соответствуют значениям, показанным рядом с вертикальными стрелками на диаграмме сигнала Манчестер-II. Так, в начале такта T0* момент опроса сигнала Манчестер-II примерно совпадает с его первым положительным фронтом; при этом, как предполагаем, триггер D1 остается в нулевом состоянии (см. обозначение “0” рядом с первой слева стрелкой). Последующие опросы соответствуют фиксации в триггере D1 следующих состояний: 11100011011001001100 и т. д.

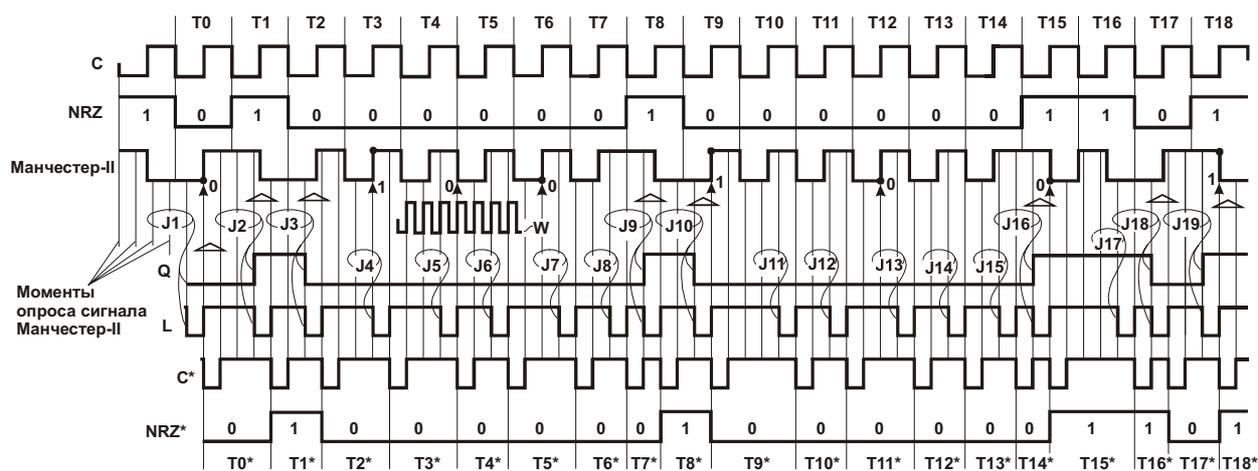


Рис. 9.28. Временные диаграммы формирования сигнала Манчестер-II и его последующей дешифрации схемой, приведенной на рис. 9.27

В отсутствие искажений входного сигнала и при заданном отношении $N = 3,33$ длительности битового интервала сигнала Манчестер-II к периоду сигнала W можно отметить следующее.

1. Низкочастотные компоненты сигнала Манчестер-II, т. е. импульсы или паузы между ними длительностью, равной одному битовому интервалу, опрашиваются тремя или четырьмя положительными фронтами сигнала W. Так, в интервале времени T17* получены четыре единичных “отсчета” входного сигнала, соответствующие последнему (на диаграмме) широкому положительному импульсу сигнала Манчестер-II. Пауза перед этим импульсом опрашивалась три раза группой отсчетов J18 и т. д.

2. Высокочастотные компоненты сигнала Манчестер-II, т. е. импульсы или паузы между ними длительностью, равной половине битового интервала, опрашиваются одним или двумя положительными фронтами сигнала W.

Таким образом, при $N = 3,33$ непрерывная последовательность отсчетов позволяет распознавать и различать низкочастотные и высокочастотные компоненты входного сигнала. Остаётся только вести непрерывный анализ этих отсчетов (с учетом предыстории) и на его основе восстанавливать исходный сигнал NRZ.

Для анализа данных Z с выхода триггера D1 и восстановления сигнала NRZ используется упоминавшийся ранее конечный автомат. В качестве текущего промежуточного результата анализа используется сигнал Q – прототип сигнала NRZ*, опережающий сигнал NRZ* на один период сигнала W. Истинность выходного сигнала NRZ* подтверждается положительными фронтами сигнала C* (эти фронты всегда соответствуют установившемуся значению сигнала NRZ*).

Рассмотрим правила формирования сигналов Q , C^* и NRZ^* , т. е., по существу, алгоритм функционирования конечного автомата. (Сигналы E и ERR обнаружения ошибки кодирования входного сигнала далее не рассматриваем для упрощения изложения.)

1. При обнаружении трех последовательных единичных отсчетов кода Манчестер-II (трех единичных сигналов $Z = 1$) формируется сигнал $Q = 1$, а также “отрицательный” импульс $L = 0$ длительностью, равной одному периоду сигнала W . Состояние $Q = 1$ поддерживается вплоть до выполнения условия 2.
2. При обнаружении трех последовательных нулевых отсчетов кода Манчестер-II (трех нулевых сигналов $Z = 0$) формируется сигнал $Q = 0$, а также импульс $L = 0$ длительностью, равной одному периоду сигнала W . Состояние $Q = 0$ поддерживается вплоть до выполнения условия 1.
3. В состоянии $Q = 1$ импульсы $L = 0$ длительностью, равной одному такту сигнала W , формируются всякий раз при обнаружении отрицательного фронта сигнала Манчестер-II, т. е. при обнаружении перехода сигнала Z из состояния лог. 1 в состояние лог. 0. Исключение составляет первый такой переход (см. моменты опросов, охваченные на диаграмме треугольниками) после установления состояния $Q = 1$; этот переход игнорируется.
4. В состоянии $Q = 0$ импульсы $L = 0$ длительностью, равной одному такту сигнала W , формируются всякий раз при обнаружении положительного фронта сигнала Манчестер-II, т. е. при обнаружении перехода сигнала Z из состояния лог. 0 в состояние лог. 1. Исключение составляет первый такой переход после установления состояния $Q = 0$; этот переход игнорируется.
5. Сигналы NRZ^* и C^* формируются из сигналов Q и L при фиксации последних в выходном регистре RG по положительным фронтам сигнала W . Поэтому сигналы NRZ^* и C^* задержаны относительно сигналов Q и L на один период сигнала W .

Рассмотрим применение этого алгоритма на примере конкретной кодовой ситуации, приведенной на рис. 9.28.

Предположим, что группа $J1$ из трех последовательных отсчетов сигнала Манчестер-II (три вертикальные линии, проходящие через овал с обозначением $J1$) соответствует его нулевому значению. Согласно правилу 2, по результату последнего опроса из группы $J1$ формируется сигнал $Q = 0$ и сигнал $L = 0$. В следующем такте сигнала W сигнал L безусловно восстанавливает исходное значение $L = 1$.

Согласно правилу 4, при $Q = 0$ начинается слежение за положительными фронтами сигнала Манчестер-II, причем первый такой фронт игнорируется. Первый положительный фронт сигнала Манчестер-II соответствует последовательным нулевому и единичному отсчетам, которые на диаграмме обведены крайним левым треугольником (это обозначение используется и далее для указания фронтов, которые игнорируются).

Следующим значимым событием является обнаружение группы $J2$ из трех единичных отсчетов. Согласно правилу 1, по результату последнего опроса из группы $J2$ формируется сигнал $Q = 1$ и сигнал $L = 0$. В следующем такте сигнала W сигнал L безусловно восстанавливает исходное значение $L = 1$.

Согласно правилу 3, при $Q = 1$ начинается слежение за отрицательными фронтами сигнала Манчестер-II, причем первый такой фронт игнорируется. Первый (после установления сигнала $Q = 1$) отрицательный фронт сигнала Манчестер-II соответствует последовательным единичному и нулевому отсчетам, которые игнорируются (на диаграмме обведены вторым слева треугольником).

Далее в результате обнаружения группы $J3$ из трех последовательных нулевых отсчетов (см. правило 2) формируются сигнал $Q = 0$ и третий импульс $L = 0$. Согласно правилу 4, первый обнаруженный положительный фронт входного сигнала (см. третий слева треугольник) игнорируется, а последующие положительные фронты, соответствующие группам отсчетов $J4 - J8$, вызывают формирование импульсов $L = 0$.

Группы отсчетов $J9$ и $J10$ вызывают формирование положительного импульса $Q = 1$ и сопровождающей его пары импульсов $L = 0$. Группы отсчетов $J11 - J15$ соответству-

ют обнаружению положительных фронтов входного сигнала на фоне сигнала $Q = 0$ (первый обнаруженный фронт игнорируется). Группы отсчетов J16 и J18 вызывают формирование положительного импульса $Q = 1$ и сопровождающей его пары импульсов $L = 0$.

Пара отсчетов J17 соответствует второму отрицательному фронту входного сигнала на фоне сигнала $Q = 1$ (первый отрицательный фронт игнорируется) и формированию импульса $L = 0$. Группа J19 из трех последовательных единичных отсчетов входного сигнала соответствует переходу сигнала Q в состояние лог. 1 и формированию последнего импульса $L = 0$.

Сигнал NRZ* задержан относительно сигнала NRZ: биты, размещенные во временных интервалах T_0, T_1, T_2 и т. д., преобразуются в соответствующие биты в интервалах T_0^*, T_1^*, T_2^* и т. д. Из диаграммы следует, что длительности битовых интервалов T_0^*, T_1^*, T_2^* и т. д. не одинаковы даже при отсутствии джиттера во входном сигнале. Так, длительности интервалов T_{15}^* и T_{16}^* различаются в 2,5 раза. При этом средняя длительность битового интервала сигнала NRZ* в точности соответствует длительности битового интервала сигнала NRZ и при $N = 3,33$ составляет 3,33 периода сигнала W . Иными словами, дешифратор формирует выходной сигнал со значительным джиттером, что, однако, не представляет опасности для устройства-потребителя данных, так как достоверность каждого бита кода NRZ* подтверждается положительным фронтом сигнала C^* .

Выводы

1. При построении дешифраторов кода Манчестер-II обычно используется один из двух подходов. Первый основан на использовании автономного генератора для опроса состояний входной линии. При этом частота сигнала с выхода этого генератора в 5 или более раз превышает скорость передачи данных по линии. Такое превышение приводит к относительно сложным схемным решениям с использованием быстродействующих элементов, потребляющих повышенную мощность. Второй подход основан на использовании генератора с фазовой автоподстройкой частоты, что приводит к усложнению схемных решений.
2. Дешифратор (рис. 9.27) также использует автономный генератор, но позволяет снизить частоту опроса сигнала в линии, что упрощает его структуру. Дешифратор построен на основе конечного автомата, алгоритм дешифрации хранится в памяти, что позволяет легко его изменять, например, для обнаружения ошибок кодирования или повышения устойчивости к искажениям входного сигнала.

10. Синхронизация передатчика и приемника тестовых сигналов при измерении длины линий передачи данных

Прежде чем применить модем либо иное телекоммуникационное устройство для работы на выделенной линии (витой паре проводов, проложенной непосредственно между абонентами, без использования какой-либо промежуточной аппаратуры усиления, коммутации, уплотнения и т. п.), желательно уточнить ее основные параметры. К этим параметрам прежде всего относятся длина линии и тип кабеля, на основе которого она выполнена. Если эти параметры точно известны, то поставщик телекоммуникационного устройства может с высокой достоверностью предсказать, применима ли эта линия для передачи данных между конкретными устройствами.

Для измерения длины кабельных линий передачи данных обычно применяют рефлектометры. Рефлектометр представляет собой своеобразный локатор. Он выдает в линию зондирующий импульс и регистрирует задержку получения отраженного от ее конца сигнала. Зная эту задержку и скорость распространения сигнала в линии, можно вычислить ее длину.

Эксперименты показывают, что рефлектометрический метод измерения длины стандартного телефонного кабеля типа ТПП-0,5 (представляющего набор витых пар проводов с

диаметром медной жилы 0,5 мм) не позволяет работать на дистанциях, превышающих 4 км, что явно недостаточно для практических целей, когда необходим диапазон, измеряемый десятками километров.

Это связано с тем, что, с одной стороны, короткий зондирующий импульс не может преодолеть достаточно длинную линию и вернуться обратно – он сильно затухает и не выделяется на фоне шумов. С другой стороны, с увеличением длительности зондирующего импульса диапазон длин измеряемых линий расширяется, но быстро входит в насыщение. Это обусловлено тем, что длительность зондирующего импульса становится соизмеримой с временем его распространения к удаленному концу линии и обратно. В результате получаемые рефлектограммы становятся непригодными для надежного распознавания отраженного импульса на фоне зондирующего из-за их интерференции и влияния других факторов.

Рассмотренные далее решения основаны на замене импульсного сигнала ступенчатым. При этом сигнал распространяется по витой паре проводов только в одну сторону. Начальная и конечная точки распространения сигнала могут находиться рядом, если линия содержит две витые пары проводов, объединенные на дальнем конце. В этой ситуации применим первый вариант измерителя (п. 10.1). Второй вариант (п. 10.2) позволяет работать с линиями, у которых начало и конец территориально разнесены. Для синхронизации приемника с источником тестового ступенчатого сигнала используются кадровые синхроимпульсы одного из каналов местного телецентра.

10.1. Измеритель длины петли передачи данных

Измеритель [73] содержит ведущий и ведомый блоки, подключенные к противоположным сторонам проверяемой линии передачи данных (рис. 10.1). Ведущий блок содержит микрокомпьютер (P1 ... P6 – группы сигналов входных и выходных портов), генератор G импульсов, передатчик Tx, приемник Rx, счетчик СТ и блок регистрации формы сигнала, выполненный на основе аналого-цифрового преобразователя АЦП (ADC), буфера типа FIFO и мультиплексора MUX. Ведомый блок содержит передатчик Tx и приемник Rx.

Рассмотрим работу составных частей устройства. Кварцевый генератор G формирует на выходах непрерывные последовательности импульсов А, В и С со скважностью, равной двум (рис. 10.2). Частота сигнала А выбирается достаточно низкой (например, 50 Гц или ниже, вплоть до долей герца) для того, чтобы длительность импульса, передаваемого по линии, значительно превышала время распространения сигнала по этой линии. Иными словами, можно считать, что фактически измеритель оперирует не импульсами (как в рефлектометрах), а редкими перепадами “статического” сигнала.

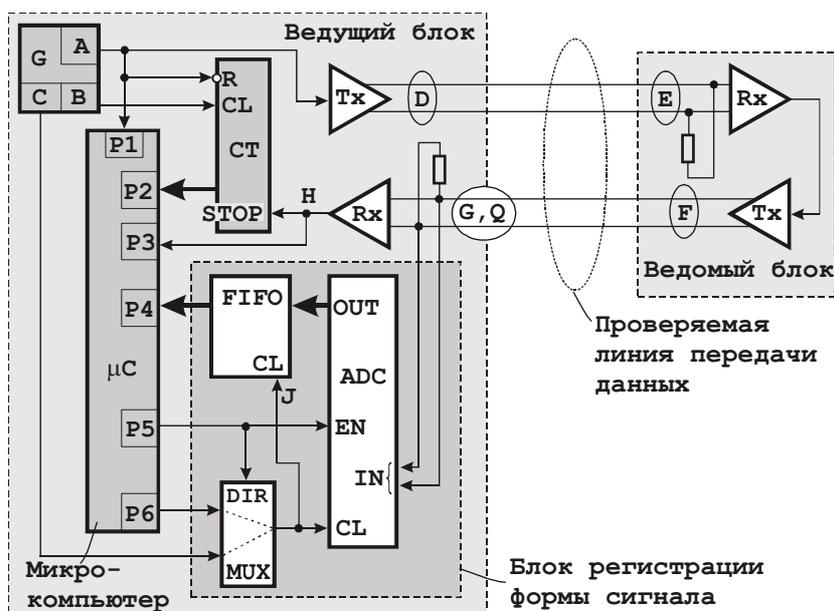


рис. 10.1. Схема измерения длины кабельной линии передачи данных – первый вариант

Частота сигнала В определяет точность регистрации интервала времени между фронтами переданного и полученного из линии сигналов и может составлять, например, 100 МГц. Частота сигнала С задает число точек, по которым воспроизводится форма сигнала (для распознавания начала фронта) после его прохождения по линии (рис. 10.3) и может составлять, например, 40 кГц.

Счетчик СТ прибавляет единицу к текущему содержимому по фронту сигнала В при условии, что $R = 1$ и $STOP = 0$. При $R = 1$ и $STOP = 1$ счет останавливается, т. е. прекращается прибавление единиц к содержимому счетчика. При $R = 0$ счетчик устанавливается в нуль независимо от состояния сигналов на других входах.

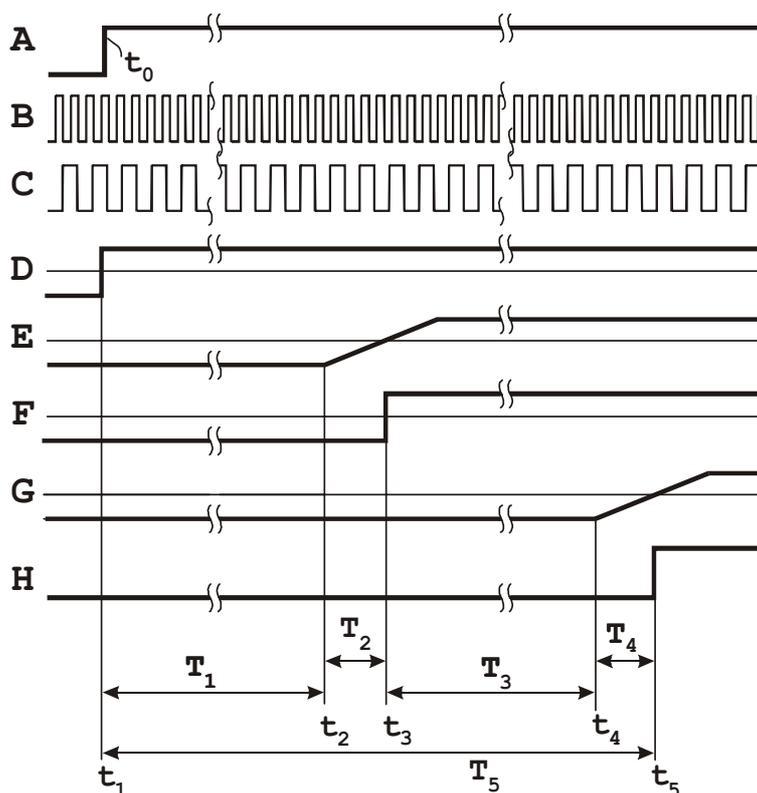


рис. 10.2. Временные диаграммы работы измерителя длины кабельной линии передачи данных

Передачики T_x имеют низкое выходное сопротивление. При переключении передатчика напряжение между его выходами изменяет знак. Длительность фронта выходного сигнала передатчика может быть преднамеренно увеличена до некоторого заданного значения (которое учитывается в окончательных расчетах длины линии), а его форма сглажена, чтобы уменьшить перекрестные помехи, наводимые на соседние линии кабеля. Для упрощения изложения предполагаем, что длительность фронта сигнала на выходах передатчиков пренебрежимо мала. Приемники выполняют функции компараторов и регистрируют моменты изменения полярности входного напряжения.

Блок регистрации формы сигнала при $P5 = 1$ постоянно отслеживает “недавнюю предысторию” напряжения на входах АЦП. Это позволяет расчетным путем с приемлемой точностью определить момент t_4 начала фронта сигнала Q (см. рис. 10.3).

Память типа FIFO работает по принципу конвейера. Под действием фронта сигнала J на входе синхронизации очередной код с выходов АЦП записывается в начало конвейера. В то же время в результате продвижения данных по конвейеру на его выход пересылается очередной код, соответствующий наиболее давней предыстории. Длина конвейера может составлять, например, 256 ячеек. При отсутствии динамики сигнала J содержимое памяти остается неизменным.

Аналого-цифровой преобразователь работает при наличии сигнала разрешения $EN = 1$. Очередной цикл преобразования начинается при поступлении сигнала $J = 1$. Мультиплексор при $R5 = 1$ передает на выход сигнал C с нижнего входа; при $R5 = 0$ на выход передается сигнал $P6$.

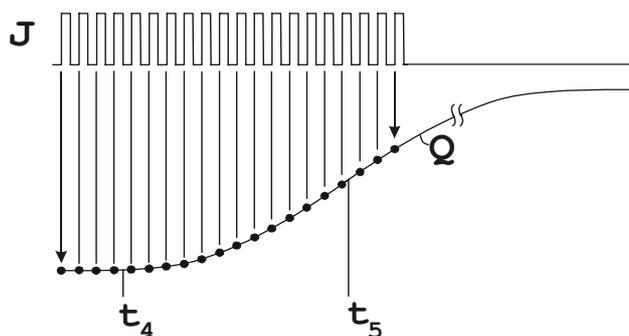


рис. 10.3. Временные диаграммы, поясняющие принцип действия блока регистрации формы сигнала

Принцип действия измерителя основан на регистрации задержки между посланным и принятым из линии перепадами уровней напряжения с учетом поправок на вносимые линией искажения фронтов сигнала. Умножив известную скорость распространения сигнала в линии на полученную расчетным путем (на основе измеренных параметров) задержку прохождения сигнала по линии (в одну сторону), можно вычислить ее длину.

В исходном состоянии, до момента t_0 (см. рис. 10.2), сигнал $A = 0$ удерживает счетчик в нулевом состоянии. (На диаграмме показан лишь фрагмент низкочастотного периодического сигнала A .) Микрокомпьютер находится в режиме ожидания перехода этого сигнала в состояние лог. 1. Сигнал $P5 = 1$ разрешает работу АЦП и настраивает мультиплексор на передачу сигнала C . Таким образом, АЦП и память FIFO непрерывно (с частотой дискретизации, определяемой сигналом C) отслеживают новейшую предысторию состояния сигнала на входах АЦП. Выдвигаемая из памяти информация не воспринимается микрокомпьютером и теряется как ненужная; при этом $P6 = 0$.

В момент t_0 формируется сигнал $A = 1$, который разрешает работу счетчика. Микрокомпьютер, получив этот сигнал, принимает к сведению факт начала цикла измерений и переходит к ожиданию его завершения, которое, как будет показано далее, сопровождается формированием сигнала лог. 1 на его входе $P3$.

В момент t_1 , близкий t_0 , напряжение D между выходами передатчика изменяет знак. Перепад напряжения начинает распространяться по первой витой паре проводов и в момент t_2 достигает дальнего конца линии. При этом фронт сигнала заметно “размывается”, что условно показано наклонным участком временной диаграммы, соответствующей сигналу E на входах удаленного приемника. После срабатывания приемника (компаратора) и передатчика (момент t_3) ведомого блока сигнал F восстановленной формы начинает распространяться в обратном направлении по второй витой паре проводов и в момент t_4 достигает входов приемника ведущего блока. Этот приемник срабатывает в момент t_5 .

Сигнал $H = 1$ с выхода приемника останавливает накопление единиц в счетчике, так что в нем фиксируется промежуток времени T_5 (см. рис. 10.2), выраженный в условных единицах – периодах сигнала B . Сигнал $H = 1$ также вызывает переход микрокомпьютера к программе завершения цикла измерения и обработки полученных результатов. Завершение цикла измерения заключается в приостановке работы блока регистрации формы сигнала, считывании данных из блока памяти и опросе счетчика.

Для приостановки работы блока регистрации формы сигнала микрокомпьютер формирует сигнал $P5 = 0$, запрещая работу АЦП и настраивая мультиплексор на передачу сигнала $P6$. Этот сигнал пока равен нулю, поэтому на выходе мультиплексора формируется временная диаграмма, приведенная на рис. 10.3. Процесс регистрации формы сигнала (см. диаграмму сигнала Q) приостанавливается через небольшой промежуток времени после момента t_5 срабатывания приемника из-за инерционности микрокомпьютера. При этом в блоке памяти хранятся данные, по которым в дальнейшем можно с некоторой точностью вычислить

длительность интервала $T_4 = t_5 - t_4$, соответствующего половине длительности фронта принятого сигнала (см. последовательность отсчетов, показанную рядом точек на диаграмме сигнала Q).

Считывание данных из блока памяти в микрокомпьютер происходит в сравнительно медленном темпе под управлением программы, которая периодически изменяет состояние сигнала Р6 и после каждого периода изменения считывает очередной отсчет, выдвинутый из блока памяти. После считывания всех данных или их части, необходимой для вычисления длительности промежутка времени $T_4 = t_5 - t_4$, микрокомпьютер опрашивает счетчик через входной порт, рассчитывает длину линии, выдает результаты на индикатор (на рисунке не показан) и переходит в описанное ранее состояние ожидания следующего положительного фронта сигнала А для повторения измерения (например в режиме усреднения результатов), либо выполняет иные действия, предписанные командами оператора.

Из временных диаграмм, представленных на рис. 10.2, следует, что, без учета задержек срабатывания передатчиков и приемников, время T_X распространения фронта сигнала до дальнего конца линии и обратно составляет

$$T_X = T_1 + T_3 = T_5 - T_2 - T_4.$$

Если предположить, что характеристики пар проводов линии примерно одинаковы, то можно считать, что $T_2 = T_4$; тогда $T_X = T_5 - 2T_4$. Время T_Y распространения сигнала в одну сторону при этом составит $T_Y = T_X/2 = T_5/2 - T_4$. Значения T_5 и T_4 определены по результатам измерений. Искомая длина L линии вычисляется по формуле

$$L = c \cdot T_Y / K,$$

где c – скорость света в вакууме; K – коэффициент укорочения, показывающий, во сколько раз скорость света в вакууме превосходит скорость распространения сигнала по кабелю (для кабеля типа ГПП-0,5 $K = 1,52$).

Если принятое ранее условие примерного равенства промежутков времени T_2 и T_4 не соблюдается (что маловероятно), то можно провести измерение повторно, поменяв местами витые пары проводов линии и усреднив результаты измерения.

Расширение диапазона измерений достигнуто благодаря тому, что по линии передаются редкие или даже однократные перепады уровней сигнала (а не импульсы, как в рефлектометре). При этом перепад уровней сигнала распространяется по петле в одном направлении, отраженные сигналы не могут его опередить и даже догнать, так что к финишу приходит полноценный по форме сигнал уменьшенной амплитуды и имеющий “размытый” фронт. Диапазон измерений можно дополнительно увеличивать повышением уровня передаваемого сигнала с одновременным контролируемым увеличением длительности его фронта. Можно формировать фронт в виде графика функции

$$y = \sin \omega t,$$

где ω – параметр, определяющий крутизну фронта; $-\pi/2 \leq \omega t \leq \pi/2$.

Увеличение длительности фронта и сглаживание его формы уменьшает нежелательное влияние проверяемой линии на соседние, размещенные в том же кабеле. Но даже при заметном влиянии, помехи окажутся редкими (или даже однократными).

10.2. Способ синхронизации измерителя длины кабельной линии передачи данных

В предыдущей схеме измерения источник и приемник тестового сигнала размещались рядом. Однако в кабеле не всегда имеется “лишняя” витая пара проводов для построения петли.

В схеме измерения, показанной на рис. 10.4 [42], начало и конец двухпроводной линии связи территориально разнесены. Для синхронизации работы передающего и приемного блоков использован сигнал местного телецентра.

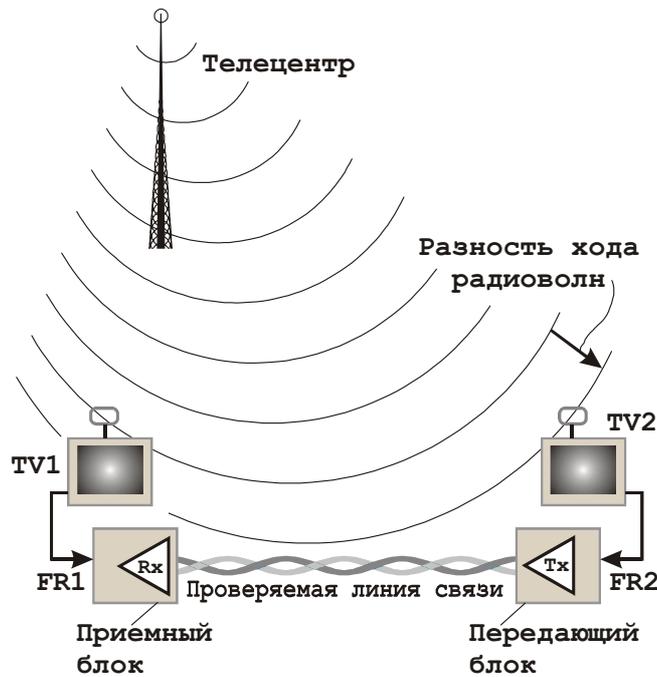


Рис. 10.4. Схема измерения длины кабельной линии передачи данных – второй вариант

Перед началом измерений телевизионные приемники TV1 и TV2 отключаются от передающего и приемного блоков, помещаются рядом и настраиваются на прием одной и той же местной телевизионной программы. С помощью осциллографа или иного измерительного прибора телевизионные приемники калибруются так, чтобы свести к минимуму разность фаз между их кадровыми синхроимпульсами FR1 и FR2.

Структура приемного и передающего блоков (рис. 10.5) близка описанной ранее структуре ведущего и ведомого блоков (см. п. 10.1).

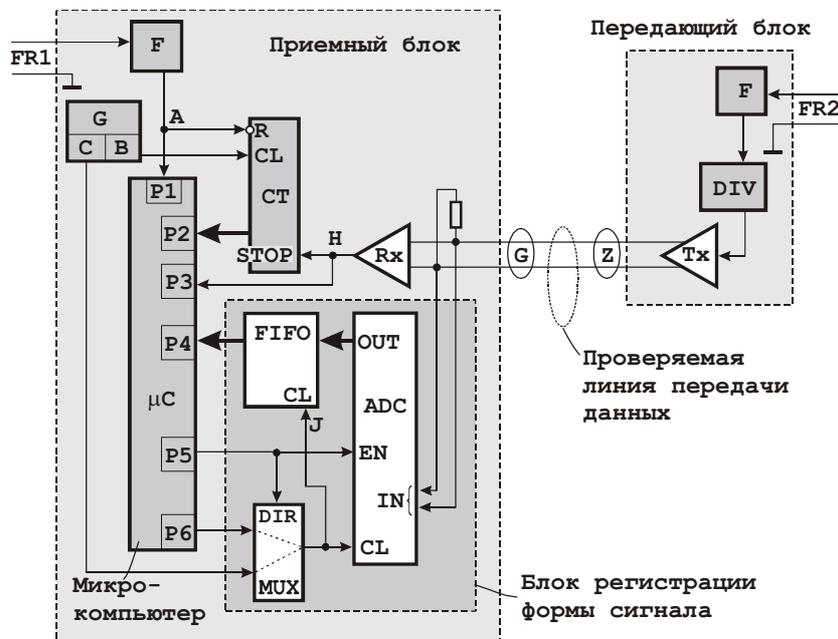


рис. 10.5. Структура приемного и передающего блоков

Введены формирователи F импульсов, делитель DIV частоты. Часть аппаратуры исключена. В приемном блоке цикл измерения начинается по сигналу FR1. В передающем блоке

выдача перепада напряжения в линию происходит по сигналу FR2 (возможно, “прореженному”, если коэффициент деления частоты делителем DIV превышает единицу). Приемный блок регистрирует время между поступлением тестового сигнала из линии и ближайшим предшествующим кадровым синхроимпульсом FR1.

Как и в предыдущем решении, для вычисления длины линии необходимо знать значение коэффициента укорочения и измерить время T_L распространения сигнала от начала линии до ее конца. Это время может быть получено проведением двух измерений с усреднением результатов.

При первом измерении расположение передающего и приемного блоков соответствует показанному на рис. 10.4. Временные диаграммы сигналов приведены на рис. 10.6, а. Разность хода радиоволн вызывает отставание кадрового синхроимпульса FR2 от импульса FR1 на некоторый промежуток времени T_D . Поэтому вместо истинного значения T_L получим искаженное: $T_M = T_L + T_D$, поскольку отсчет времени начинается по положительному фронту сигнала А еще до того как в линию выдан тестовый сигнал Z.

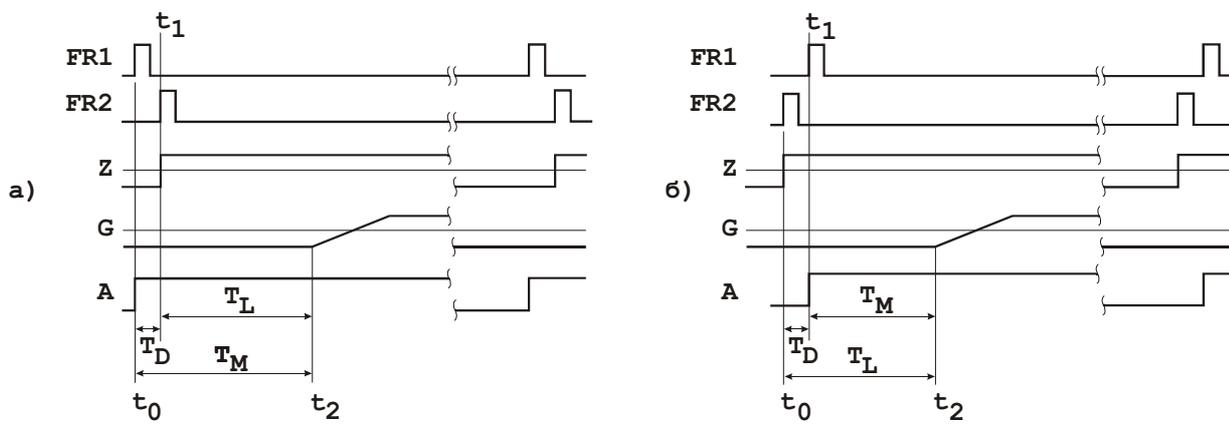


Рис. 10.6. Временные диаграммы измерения длины линии связи: а – при размещении передающего и приемного блоков в соответствии с рис. 10.4; б – после перестановки блоков

Второе измерение T_L проводится после перестановки передающего и приемного блоков относительно проверяемой линии. (Телевизионные приемники остаются на своих местах.) Поэтому разность хода радиоволн создает ошибку измерения той же величины, но противоположного знака, и вместо истинного значения T_L получим искаженное: $T_M = T_L - T_D$ (см. временные диаграммы, приведенные на рис. 10.6, б).

Усредняя результаты первого и второго измерений, получим истинное значение времени распространения сигнала от начала линии до ее конца: $(T_L + T_D + T_L - T_D)/2 = T_L$.

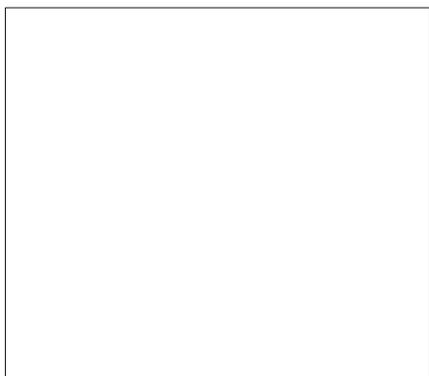
На практике комплект измерительной аппаратуры может состоять из двух одинаковых частей, каждая из которых содержит как передающий, так и приемный блоки. Это исключает необходимость механической перестановки блоков для проведения второго измерения. Отметим, что описанную ранее калибровку телевизионных приемников TV1 и TV2 можно не проводить, если при втором измерении поменять эти приемники местами. В этом случае взаимный “перекос” кадровых синхроимпульсов FR1 и FR2, вызванный неидентичностью телевизионных приемников, будет автоматически компенсирован при усреднении результатов.

Если коэффициент деления сигнала FR2 выбран большим единицы, то часть кадровых импульсов не используется для измерения. “Пропущенные” измерения соответствуют получению микрокомпьютером приемного блока сигнала с выхода приемника Rx за период следования кадровых синхроимпульсов (20 мс).

Литература

1. Патент США № 5.408.500.
2. <http://www.datum.com> – Сайт ф. Datum (США)
3. Патент США № 5.687.199.
4. Патент США № 6.088.412.
5. Патент США № 5.727.034.
6. Патент США № 5.367.534.
7. Патент США № 6.195.385 В1.
8. Патент США № 5.062.124.
9. <http://www.tiaonline.org> – Сайт ассоциации производителей телекоммуникационного оборудования TIA (The Telecommunication Industry Association)
10. Патент США № 5.933.468.
11. Патент США № 6.044.421.
12. Патент США № 5.982.833.
13. Патент США № 5.966.409.
14. Патент США № 5.822.328.
15. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб: Изд-во “Питер”, 2000. – 672 с.
16. Стандарт IEEE 802.3.
17. Рекомендация ITU-T G.703.
18. Патент США № 5.610.951.
19. Патент США № 4.442.550.
20. Патент США № 5.539.751.
21. Electronic Products. – 1982. – Vol. 25, N 5. – P. 63 – 67, 71 – 76.
22. Intern. J. of Electronics. – 1980. – Vol. 49, N 2. – P. 175 – 177.
23. Патент США № 5.771.249.
24. Патент США № 4.872.186.
25. Патент США № 5.825.834.
26. Патент США № 5.422.919.
27. Патент США № 6.011.808.
28. Патент США № 5.905.756.
29. Патент США № 6.400.727 В1.
30. Патент США № 5.463.645.
31. Патент США № 5.204.859.
32. Патент США № 5.901.189.
33. Патент США № 4.715.044.
34. Патент США № 5.943.377.
35. Патент США № 4.833.675.
36. Патент США № 4.771.442.
37. Патент США № 5.058.142.
38. Патент США № 6.421.404 В1.
39. Патент США № 5.781.595.
40. Патент США № 6.222.893 В1.
41. Патент США № 6.470.034 В1.
42. Патент РФ № 2.187.784.
43. Патент США № 4.578.797.
44. Патент США № 5.245.635.
45. Мультимедиа, 2(12), апрель 2002 г., с. 17 – 19.

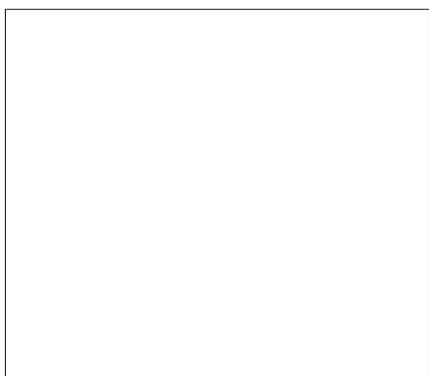
46. Патент США № 5.408.507.
47. Патент США № 4.370.617.
48. Патент США № 4.817.117.
49. Патент США № 6.134.234.
50. Патент США № 5.068.879.
51. Патент США № 5.956.348.
52. Патент США № 6.195.402 В1.
53. Патент США № 6.081.550.
54. Патент США № 6.215.835 В1.
55. Патентная заявка США № US 2002/0037064 А1.
56. Патент США № 6.002.732.
57. Патент США № 5.499.275.
58. Патент США № 5.517.521.
59. Патент США № 5.724.392.
60. Патент США № 5.724.360.
61. Стандарт MIL-STD-1553В (США).
62. Патент США № 5.602.859.
63. Патент США № 6.269.414 В1.
64. Патент США № 5.461.379.
65. Искусство схемотехники / П.Хоровиц, У.Хилл: В 3-х т. – М.: Мир, 1993. – 2 т.
66. Патент США № 4.920.545.
67. Патент США № 4.860.286.
68. Патент США № 4.599.723.
69. Патент США № 5.179.592.
70. Патент США № 5.530.959.
71. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения:
Справочник. – 2-е изд., перераб. и доп. – М.: Радио и связь, 1990. – 512 с.
72. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения:
Справочник. – Дополнение первое. – М.: Радио и связь, 1993. – 256 с.
73. Патент РФ № 2.183.052.
74. Патент США № 5.160.929.
75. Патент США № 4.590.601.
76. Патент США № 4.719.643.
77. <http://www.zelax.ru> – Сайт компании “Зелакс” (Россия).
78. Пат. США № 6.628.212.



Сухман Сергей Маратович окончил в 1987 г. Московский институт электронной техники. Один из основателей компании “Зелакс”, которая специализируется на разработке и производстве телекоммуникационного оборудования. Внес значительный вклад в развитие компании, обеспечив конкурентоспособность производимых устройств, вытеснивших с Российского рынка некоторые зарубежные аналоги. С 1992 г. является техническим директором компании. С.М. Сухман имеет ряд публикаций в ведущих отраслевых периодических изданиях по связи и телекоммуникациям. Не-

которые предложенные им технические решения запатентованы.

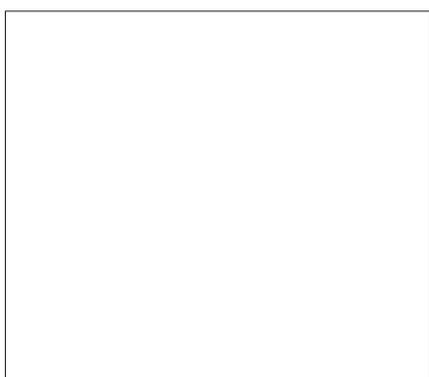
E-mail: suhman@zelax.ru



Бернов Аркадий Валентинович окончил в 1988 г. Московский институт электронной техники. Работал инженером-стажером в этом институте, затем ведущим инженером в ЗАО “Золотой шар”. В компании “Зелакс” работает с 1997 г., а с 2000 г. является заместителем ее технического директора. А.В. Бернов – автор технических решений и руководитель разработок хорошо известных потребителям модемов для выделенных линий: М-144, М-1, М-30. В настоящее

время проектирует серию модемов нового поколения. Автор более 20 печатных работ. Основная тема публикаций – современные технологии связи, новые принципы построения телекоммуникационной аппаратуры.

E-mail: bav@zelax.ru



Шевкопляс Борис Владимирович после окончания в 1969 г. Московского инженерно-физического института работал на предприятиях министерства электронной промышленности в г. Зеленограде. Принимал участие в разработке ряда мини- и микроЭВМ серии “Электроника НЦ”, центра коммутации сообщений (ЦКС) министерства гражданской авиации, был заместителем главного конструктора по схемотехнике диалоговых вычислительных комплексов (ДВК), разрабатывал микропроцессорную аппаратуру специального назначения. С 2000 г. работает ведущим инженером в

компании “Зелакс”. Автор 65 изобретений и трех книг “Микропроцессорные структуры: инженерные решения”.

E-mail: borissh@zelmail.ru