

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

---

Государственное образовательное учреждение высшего профессионального образования  
"Московский государственный институт электронной техники (технический университет)"  
(МИЭТ)

Б.В. Шевкопляс

**СБОРНИК ЗАДАЧ**  
по учебному курсу  
“Синхронизация в телекоммуникационных системах”  
кафедры ТКС (телекоммуникационные системы)

Учебное пособие

Утвержден Советом УМО по образованию в области  
телекоммуникаций 28.02.2008 г

Москва 2009

Приведены более ста задач и вопросов по учебному курсу “Синхронизация в телекоммуникационных системах” кафедры ТКС (телекоммуникационные системы) московского государственного института электронной техники (технического университета МИЭТ).

Учебное пособие рекомендовано УМО по образованию в области телекоммуникаций в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки дипломированных специалистов 210400 — Телекоммуникации.

Может быть полезно аспирантам и техническим специалистам, желающим проконтролировать и расширить свои базовые знания в части освоения практических задач построения телекоммуникационных систем и их составных частей.

Учебное пособие может использоваться при проведении собеседований и экзаменов по отдельным читаемым на кафедре курсам или при приёме государственного экзамена по их совокупности.

© Автор, 2009

# Содержание

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| <b>1. Предисловие .....</b>                                                    | <b>4</b>   |
| <b>2. Элементы теории, вопросы, задачи .....</b>                               | <b>5</b>   |
| 2.1. Взаимодействие между устройствами типа DTE и DCE.....                     | 5          |
| 2.2. Вопросы и задачи к разделу 2.1.....                                       | 22         |
| 2.3. Способы кодирования данных для их передачи по каналу связи .....          | 33         |
| 2.4. Вопросы и задачи к разделу 2.3.....                                       | 38         |
| 2.5. Вероятностная синхронизация.....                                          | 40         |
| 2.6. Вопросы и задачи к разделу 2.5.....                                       | 42         |
| 2.7. Передача данных по фантомным цепям каналов связи .....                    | 44         |
| 2.8. Вопросы и задачи к разделу 2.7.....                                       | 51         |
| 2.9. BER-тестеры .....                                                         | 52         |
| 2.10. Вопросы и задачи к разделу 2.9.....                                      | 56         |
| 2.11. Предотвращение заикливания синхросигналов.....                           | 58         |
| 2.12. Вопросы и задачи к разделу 2.11.....                                     | 62         |
| 2.13. Скремблирование — дескремблирование данных .....                         | 64         |
| 2.14. Вопросы и задачи к разделу 2.13.....                                     | 90         |
| 2.15. Усовершенствование систем с попутной синхронизацией.....                 | 94         |
| 2.16. Вопросы и задачи к разделу 2.15.....                                     | 97         |
| 2.17. Структурирование потоков данных.....                                     | 98         |
| 2.18. Вопросы и задачи к разделу 2.17.....                                     | 111        |
| 2.19. Программное управление потоком данных .....                              | 113        |
| 2.20. Вопросы и задачи к разделу 2.19.....                                     | 118        |
| 2.21. Выделение синхросигнала и данных из принимаемого сигнала .....           | 119        |
| 2.22. Вопросы и задачи к разделу 2.21.....                                     | 123        |
| 2.23. Сопряжение разноскоростных устройств.....                                | 124        |
| 2.24. Вопросы и задачи к разделу 2.23.....                                     | 127        |
| 2.25. Канал E1 .....                                                           | 129        |
| 2.26. Вопросы и задачи к разделу 2.25.....                                     | 134        |
| 2.27. Объединение удаленных сегментов сети Ethernet 10 Base T.....             | 136        |
| 2.28. Вопросы и задачи к разделу 2.27.....                                     | 141        |
| 2.29. Проскальзывания синхронизации .....                                      | 142        |
| 2.30. Вопросы и задачи к разделу 2.29.....                                     | 150        |
| 2.31. Мультиплексирование разнородных потоков данных (E1, Ethernet, V.35)..... | 151        |
| 2.32. Вопросы и задачи к разделу 2.31.....                                     | 153        |
| <b>3. Ответы и решения .....</b>                                               | <b>161</b> |
| <b>4. Литература .....</b>                                                     | <b>191</b> |

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

*К. Прутков*

## 1. Предисловие

Опыт работы со студентами старших курсов МИЭТ — московского государственного института электронной техники (технического университета) — показывает, что теоретическая сторона читаемых на кафедре ТКС (телекоммуникационные системы) курсов лекций в ряде случаев воспринимается ими в значительной степени абстрактно, без должной привязки к конкретным практическим задачам.

Настоящий сборник призван до некоторой степени восполнить этот недостаток и показать студентам, что полученная ими теоретическая подготовка позволяет решать достаточно сложные инженерные задачи.

Данное учебное пособие предназначено для студентов, получающих образование по специальностям 2010 (Многоканальные телекоммуникационные системы) и 2018 (Защищенные системы связи). Пособие может быть полезно аспирантам и техническим специалистам, желающим проконтролировать и расширить базовые знания в части освоения практических задач построения телекоммуникационных систем и их составных частей.

Приведенные вопросы и задачи могут предлагаться студентам при проведении собеседований и экзаменов по отдельным курсам или при приеме государственного экзамена по совокупности читаемых на кафедре курсов.

Каждая тема сопровождается ссылкой на источники информации — книги, статьи, патенты, стандарты или иные документы, в которых содержатся сведения, необходимые и достаточные для решения тематической группы задач или конкретной задачи. По мере возможности (с учётом ограниченного объёма данного учебного пособия) часть необходимых материалов цитируется в начале соответствующих разделов. Однако для лучшего понимания постановки задач и методов их решения рекомендуется дополнительно изучить материалы в соответствии с приведенным списком литературы.

Раздел “Ответы и решения” позволяет студентам на этапе подготовки к экзамену сопоставить собственный ход рассуждений с предлагаемым в данном учебном пособии. При проведении экзамена студенту предоставляется только вопрос или условие задачи без ответа или решения.

Для решения некоторых задач следует воспользоваться калькулятором.

Автор выражает благодарность инициатору данного издания, заведующему кафедрой телекоммуникационных систем, д. т. н., профессору В.В. Баринову за организацию работ по формированию данного учебного пособия, а также за замечания и предложения по улучшению его содержания.

Отзывы, замечания и предложения присылайте, пожалуйста, по адресу электронной почты кафедры ТКС МИЭТ: [tcs@miee.ru](mailto:tcs@miee.ru).

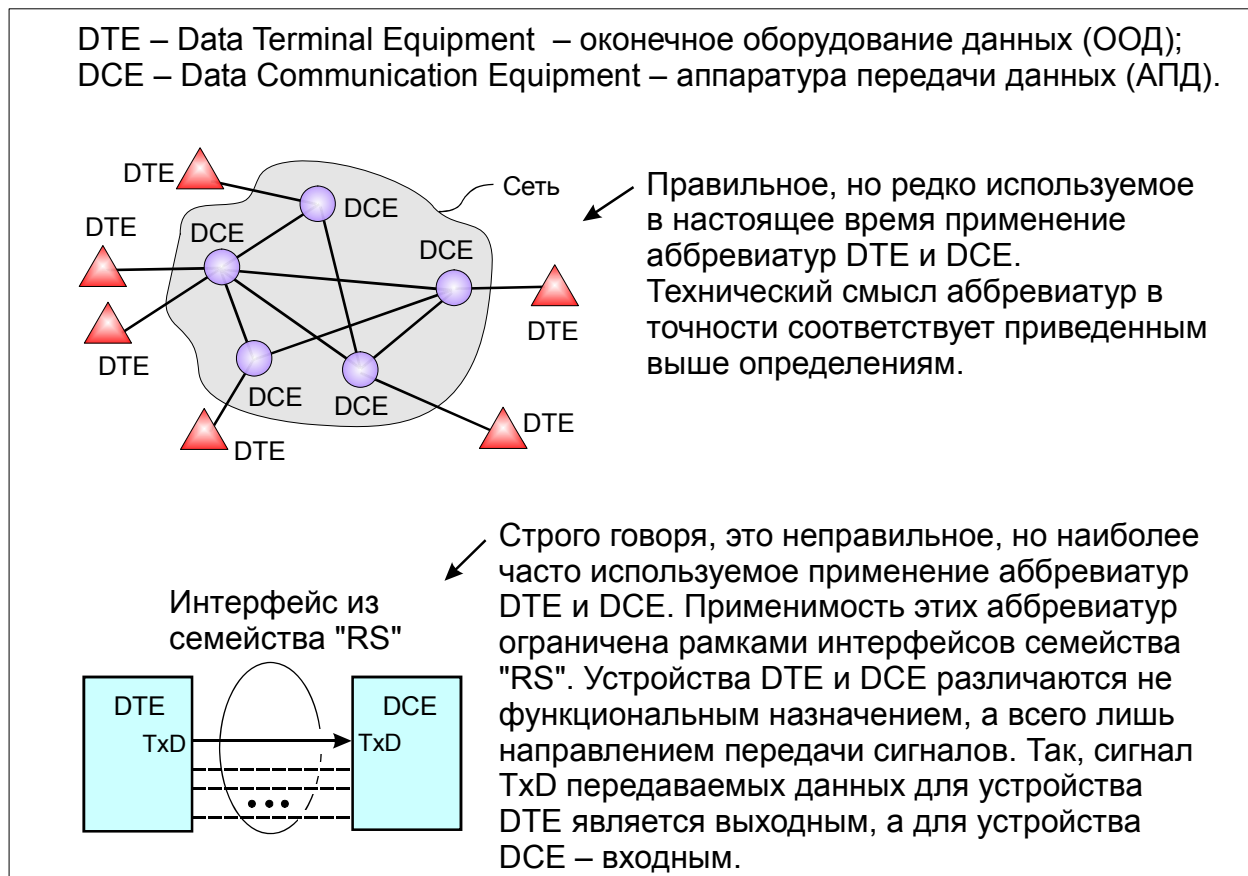
## 2. Элементы теории, вопросы, задачи

### 2.1. Взаимодействие между устройствами типа DTE и DCE

Литература к разделу 2.1: [1], гл. 1 — 3; [4]

#### 2.1.1. Введение в раздел 2.1

Исторически сложилось так, что аббревиатуры DTE и DCE хотя и имеют однозначную расшифровку, но в зарубежной и отечественной литературе применяются по-разному (Рис. 2.1).



**Рис. 2.1. Неоднозначность трактовки аббревиатур DTE и DCE**

Далее аббревиатуры DTE и DCE используются только применительно к интерфейсам семейства RS (RS-232, V.35 и другим).

На Рис. 2.2 показана схема соединения устройств типа DTE и DCE и приведены основные сигналы интерфейса RS-232. Общая цепь сигнальной земли на рисунке 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 направлениями передачи сигналов интерфейса RS-232 (или иного интерфейса из семейства RS). Всего лишь этим!

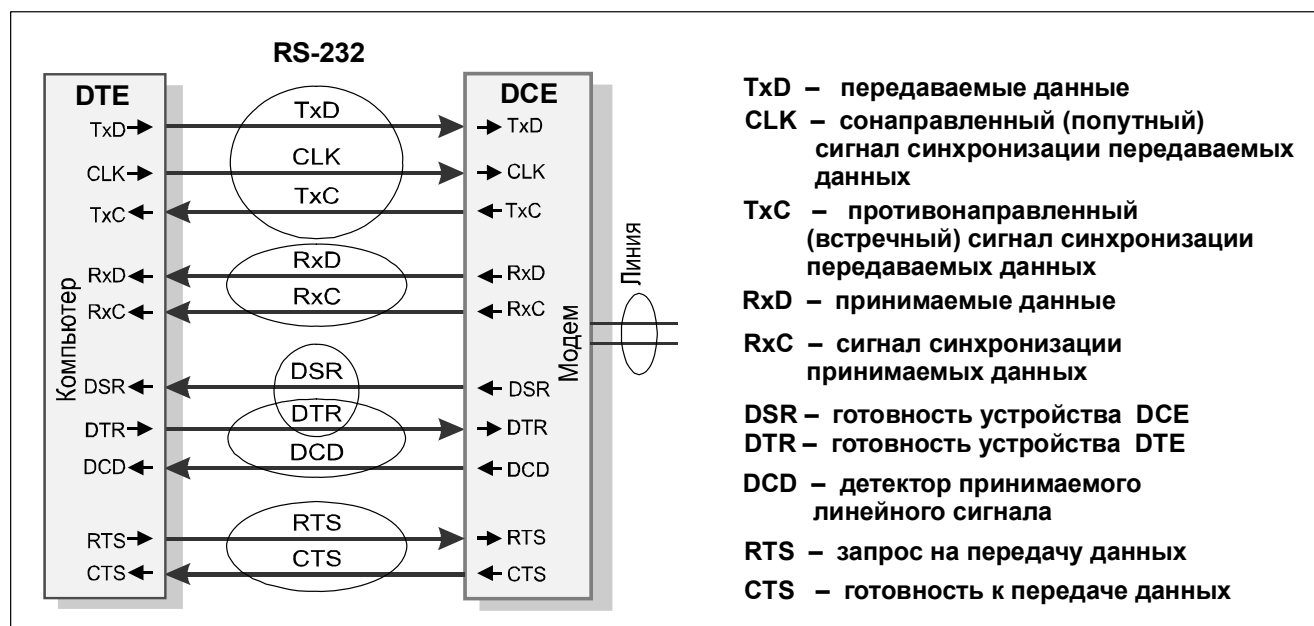


Рис. 2.2. Схема соединения устройств типа DTE и DCE

Для удобства запоминания различий между устройствами DTE и DCE можно воспользоваться шутливыми правилами:

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. 2.2, 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 Channel received line signal Detector), то оно явно коммуникационное (DCE); устройство, которое поглощает этот сигнал, не иначе как терминальное (DTE).

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. 2.2 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 — 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, TxC и RxC (см. Рис. 2.2).

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

### *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 — Transmitted Data — данные, передаваемые из устройства DTE в устройство DCE в асинхронном или синхронном режиме.

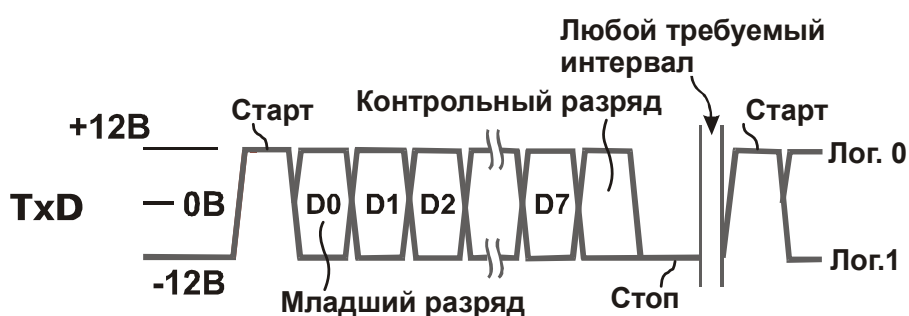
В синхронном режиме границы и середины битовых интервалов соответствуют положительным и отрицательным фронтам синхросигнала, передаваемого по отдельной цепи. Этот режим преимущественно используется в высокопроизводительных системах передачи данных.

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.

## 2.1.2. 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 передаваемые данные представлены потоком символов, каждый из которых снабжен служебными битами Старт и Стоп (Рис. 2.3).



**Рис. 2.3. 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 в скоростных версиях интерфейса.

Приемник синхронизируется положительным фронтом (переходом напряжения от -12 В к +12 В) сигнала Старт. Зная длительность битового интервала  $T = 1/V$  и формат посылки, при-

емник 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.

### 2.1.3. 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. 2.4, а).

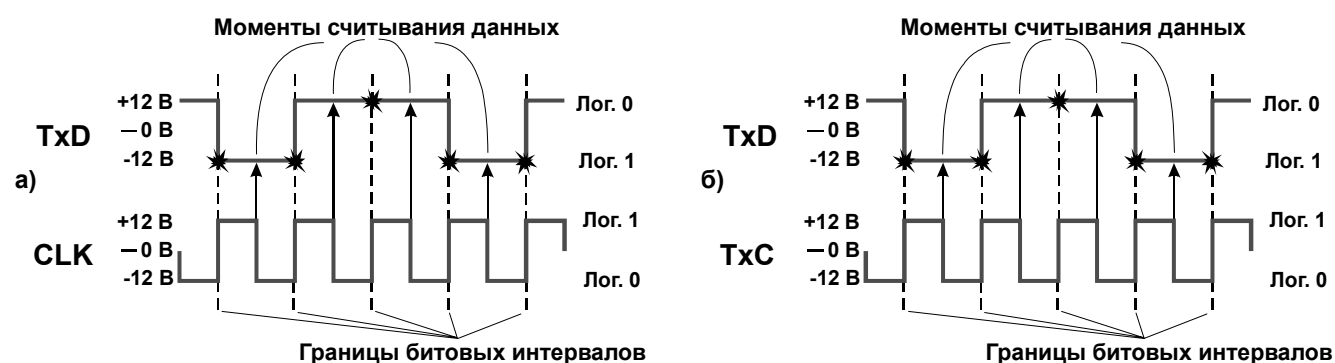


Рис. 2.4. Временные диаграммы передачи данных TxD в синхронном режиме:  
а — вариант 1; б — вариант 2

CLK — 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. 2.4, б).

TxC — Transmitter signal element timing (DCE) — сигнал синхронизации передаваемых данных TxD, формируемый устройством DCE. Получив положительный фронт этого сигнала, пе-

редатчик 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. Скважность этих сигналов обычно равна двум.

#### 2.1.4. 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. 2.4, 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 несущественны 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.

#### 2.1.5. 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. 2.5.

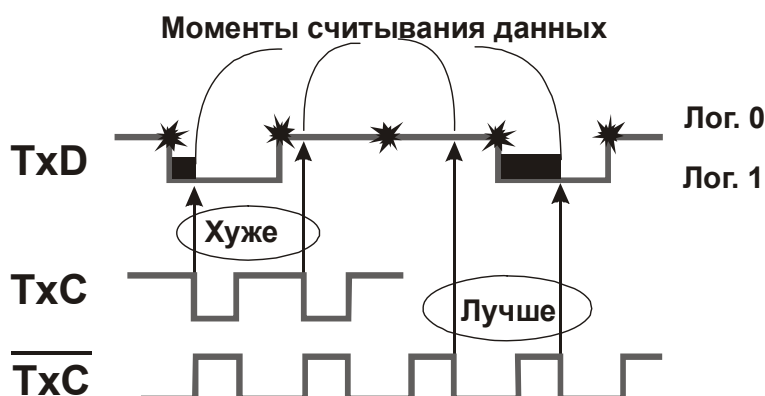


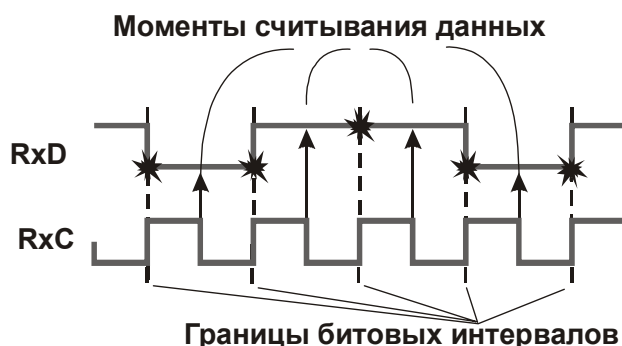
Рис. 2.5. Реальная временная диаграмма. Скважность синхросигналов отлична от двух, а данные поступают с задержкой

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 подстерегать другая опасность — приближая отрицательный фронт синхро-

сигнала к концу битового интервала (т. е. увеличивая длину затемненного прямоугольника), мы рискуем зарегистрировать нестабильные данные.

Рассмотрим сигналы RxD и RxC.

RxD — Received data — данные, принимаемые устройством DTE в асинхронном или синхронном режиме. Асинхронный режим передачи сигналов RxD аналогичен рассмотренному ранее (см. Рис. 2.3, на котором обозначение TxD теперь следовало бы изменить на RxD). Синхронный режим также аналогичен рассмотренному ранее первому варианту передачи сигналов TxD (Рис. 2.6).



**Рис. 2.6. Временная диаграмма передачи данных RxD в синхронном режиме**

RxC — 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.

### 2.1.6. Пары сигналов DTR — DSR и DTR — DCD

Начнем с определений.

DTR — Data terminal ready — готовность устройства DTE к обмену данными.

DSR — Data set ready — готовность устройства DCE к обмену данными.

DCD — 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.

### 2.1.7. 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 — Request to send — запрос на передачу данных;

CTS — 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 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).

### 2.1.8. 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. 2.2) 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.

Пока остаётся открытым вопрос о том, как приёмник отличает истинные команды Xon и Xoff от их копий, являющихся обычными данными. Об этом — позже (см. п. 2.19).

### 2.1.9. 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. 2.2, 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. При достаточном “интеллекте” устройств в поток символов можно вводить “уникальные” коды, которые расцениваются приемником 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.

### 2.1.10. Электрические уровни сигналов интерфейса RS-232

Возвращаясь к Рис. 2.2, отметим, что, согласно интерфейсу RS-232, электрические уровни всех входящих в него сигналов отсчитываются от сигнальной земли SG — signal ground or

common return. Управляющие сигналы передаются в прямом коде, т. е. наличие сигнала соответствует типовому уровню 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. 2.3). 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:  $\pm 12$  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, TxC и CTS (см. Рис. 2.2) 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, TxC 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. 2.2). Однако эта схема не является единственно возможной. Рассмотрим другие схемы соединения взаимодействующих устройств.

### 2.1.11. 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 будет описано далее.

Рассмотрим типовые варианты сопряжения двух устройств типа DTE, представленные на Рис. 2.7. 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, ко-

гда нет достаточной информации об особенностях реализации интерфейса соединяемых устройств.

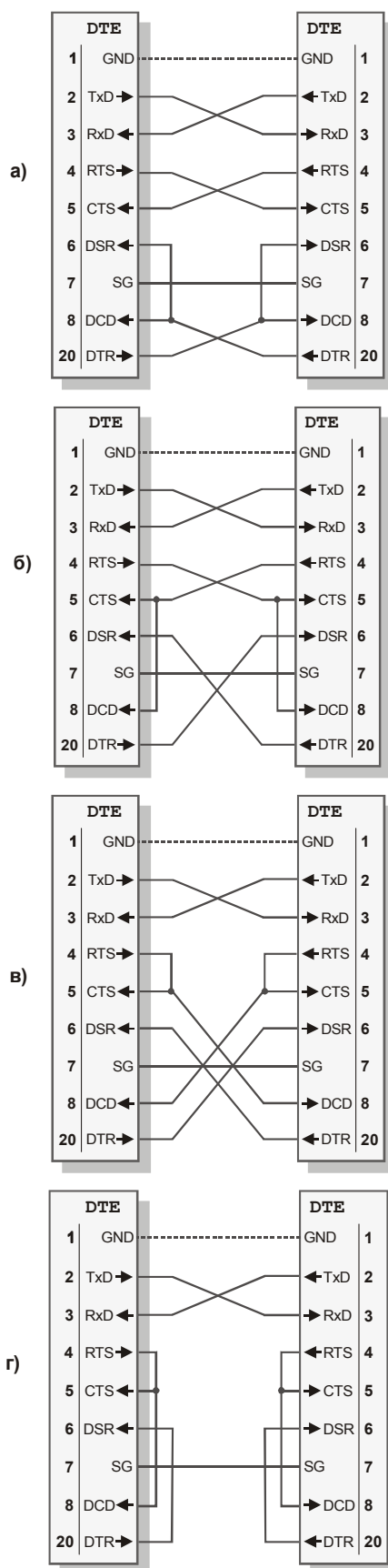


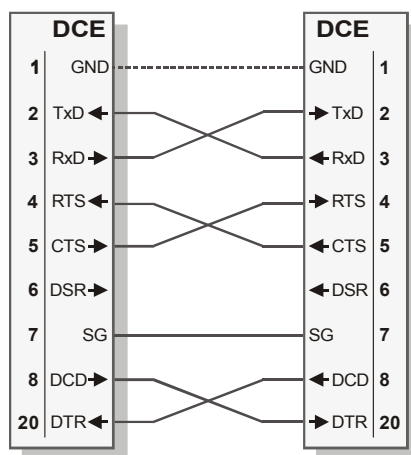
Рис. 2.7. Некоторые варианты соединения устройств типа DTE

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.

Схемы сопряжения устройств типа DCE по смыслу близки рассмотренным ранее (см. Рис. 2.7). 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. 2.8.



**Рис. 2.8. 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.

### 2.1.12. 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. 2.2). 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. 2.7 и Рис. 2.8), 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 — RxC или TxC — RxC, 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. 2.9 — Рис. 2.14).

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.

### 2.1.13. Системы с внутренней синхронизацией

В схеме, приведенной на Рис. 2.9, 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.

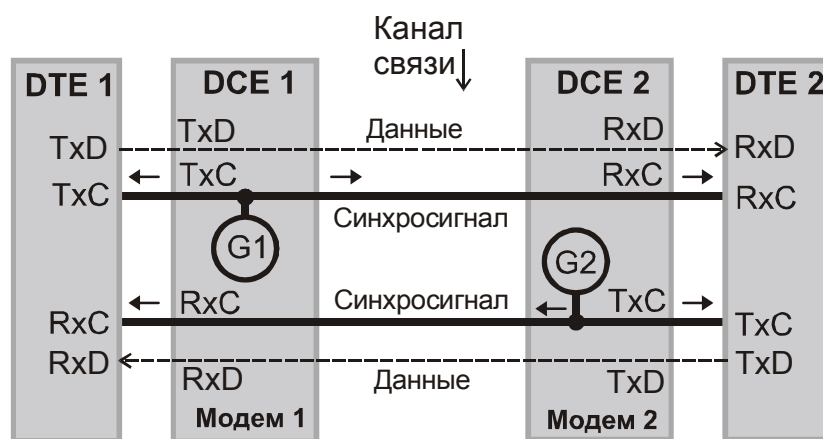


Рис. 2.9. Система с внутренней синхронизацией — первый вариант.

Под действием сигнала TxC от генератора G1 устройство DTE 1 выдает данные TxD в соответствии с временной диаграммой, приведенной на Рис. 2.4, б. Эти данные поступают в мо-

\* 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 [1].

дем 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 RxС, данные RxD передаются из модема 2 в устройство DTE 2 в соответствии с временной диаграммой, приведенной на Рис. 2.6. Схема симметрична, поэтому процесс передачи данных в обратном направлении аналогичен описанному.

Схема, показанная на Рис. 2.10, 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.

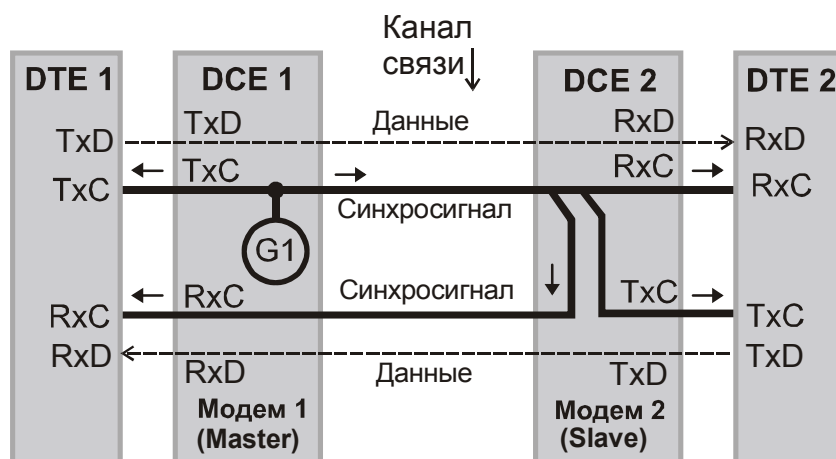


Рис. 2.10. Система с внутренней синхронизацией — второй вариант

Отметим, что сигналы TxС и RxС 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 принимаемые данные принадлежат взаимно сдвинутым по фазе синхросеткам, по которым определяются границы и центры битовых интервалов. Например, положительный фронт сигнала TxС 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 [1].

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

### 2.1.14. Системы с внешней синхронизацией

В схеме на Рис. 2.11 опорный генератор размещен в оконечном устройстве типа DCE 1. 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, приве-

денной на Рис. 2.10. Обратная передача данных из модема 1 в устройство DCE 1 сопровождается выделенным из канала сигналом RxC, который при поступлении в это устройство трактуется как сигнал CLK.

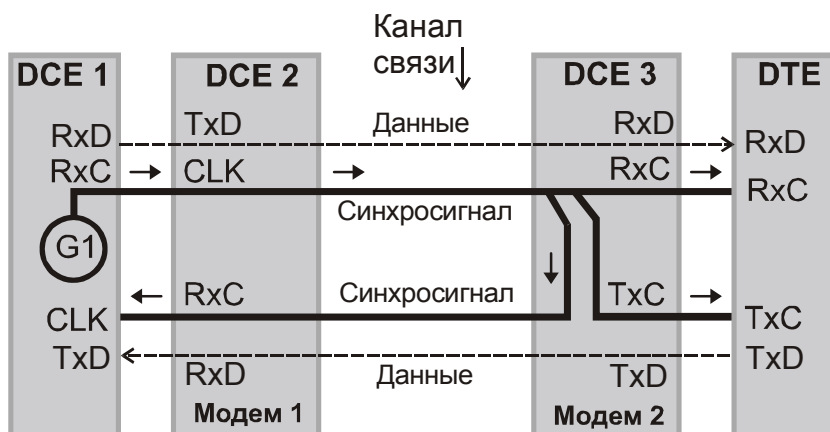


Рис. 2.11. Система с внешней синхронизацией — первый вариант

Схема на Рис. 2.12 отличается от приведенной на Рис. 2.11 двумя существенными особенностями.

Во-первых, генератор G1 формирует синхронные и синфазные сигналы RxC и TxС. 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 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 эта память 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”.

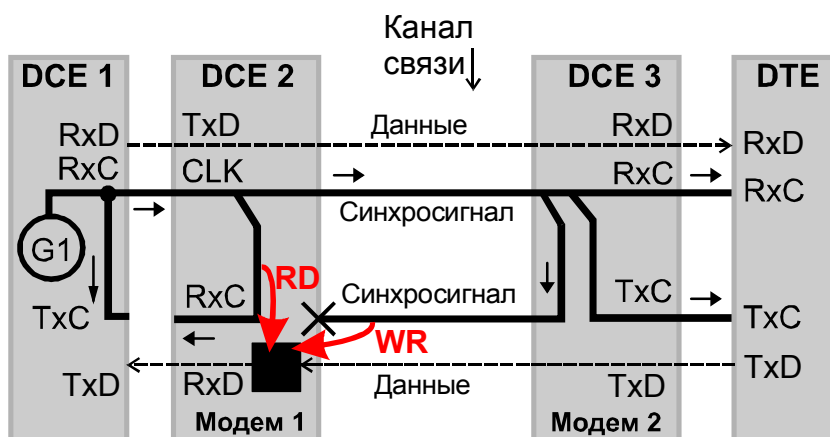


Рис. 2.12. Система с внешней синхронизацией — второй вариант

Данные считываются из буферной памяти под управлением сигналов от генератора G1, поступающих по цепи G1 — RxC — CLK. Эти данные сопровождают “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 RxС. Оно уверено в привязке поступающих от модема 1 данных к сигналу TxС. И эта уверенность оправдывается благодаря правильному выбору режима синхронизации буферной памяти.

### 2.1.15. Использование модема как устройства типа DTE

Пара модемов 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. 2.13).

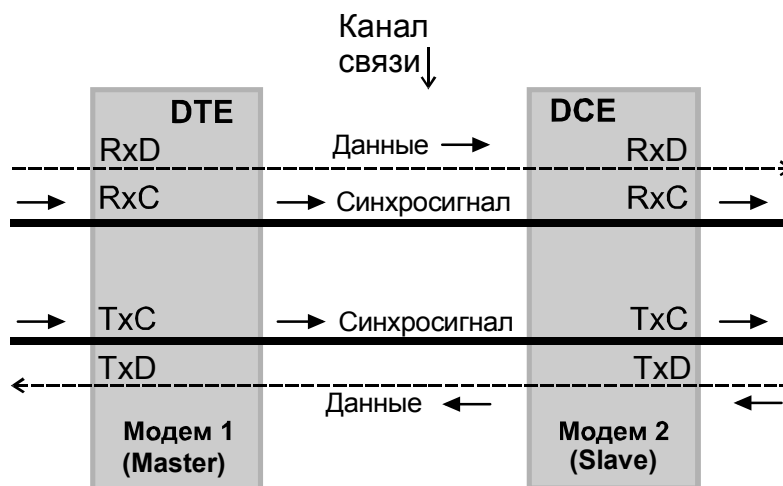


Рис. 2.13. Вариант подключения модемов к каналу связи

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 синхросигнал TxС распространяться в канале навстречу “своим” же данным (т. е. как бы “против течения”)? Может, но описание подробностей такого распространения выходит за рамки настоящего учебного пособия (см. [1]). Краткое описание идеи приведено на с. 173.

### 2.1.16. Система с двумя последовательно включенными каналами связи

Чтобы получить более общее представление о синхронизации систем, использующих приведенные решения, рассмотрим пример такой системы (Рис. 2.14). В ней применены две рассмотренные ранее схемы (см. Рис. 2.9 и Рис. 2.13). Последняя использована в качестве удлинителя для передачи сигналов RxС и TxС к удаленному оконечному устройству DTE 3. Пару модемов 2 и 3, расположенных недалеко друг от друга, можно рассматривать как ретранслятор.

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 заикливания синхросигналов описаны в [1]).

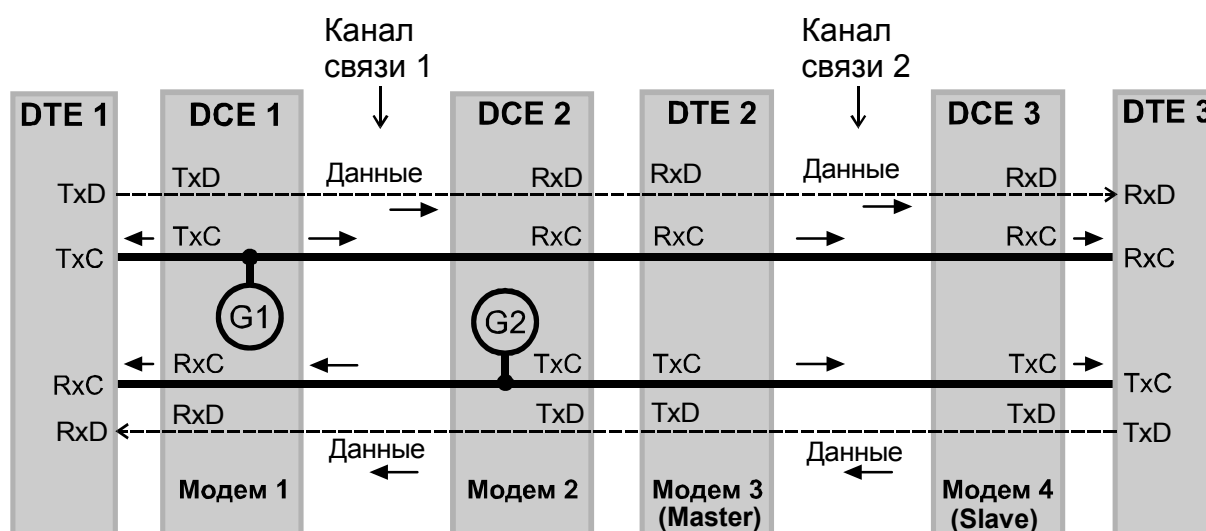


Рис. 2.14. Система с двумя последовательно включенными каналами связи

### 2.1.17. 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. 2.15, 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.

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.



**Рис. 2.15. Восстановление правильной синхронизации без обнаружения ошибки.**  
ST и SP — старт- и стоп-биты информационных посылок

В рассмотренном далее решении для восстановления правильной синхронизации и обнаружения этого события используется периодическая передача служебного кода 0000000 (Рис. 2.16). Этот код передается, например, в каждой тысячной посылке. При этом соблюдаются определенные правила, позволяющие отличить служебный код от обычного нулевого кода данных.



**Рис. 2.16. 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.

## 2.2. Вопросы и задачи к разделу 2.1

2.2.1. В системе передачи данных, показанной на Рис. 2.17, применены интерфейсы V.35.

Рассмотрите возможные варианты построения системы, при которых:

- компьютер 1 — типа DTE, скремблер — типа DTE;
- компьютер 1 — типа DTE, скремблер — типа DCE;
- компьютер 1 — типа DCE, скремблер — типа DTE;
- компьютер 1 — типа DCE, скремблер — типа DCE.

Замените символы “#” наименованиями сигналов. Структуры прямого и обратного каналов одинаковы.

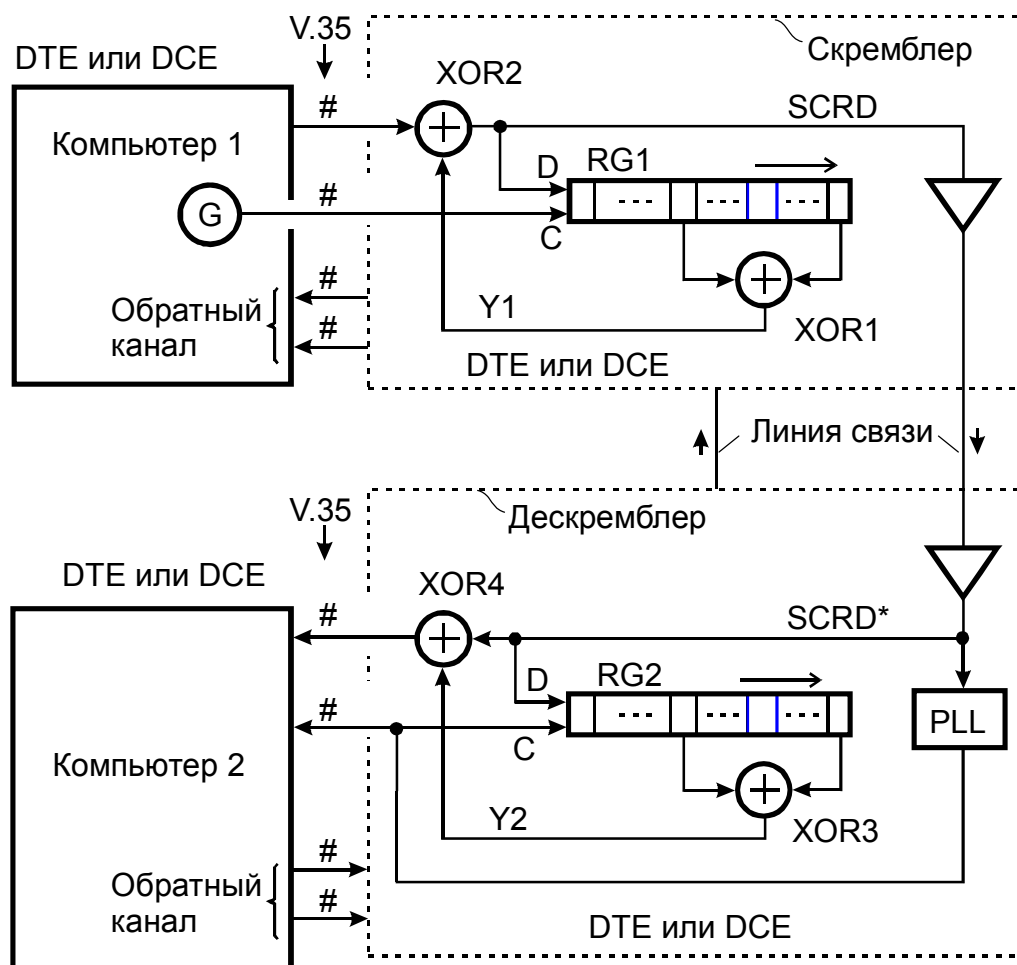


Рис. 2.17. Система передачи данных

Литература: [1], гл. 1 — 3; [4]

Решение задачи приведено в разделе 3 на с. 161

2.2.2. В системе передачи данных, показанной на Рис. 2.18, применены интерфейсы V.35.

Не нарушая синхронизацию, переместите генератор G из компьютера 1 в скремблер и рассмотрите варианты построения системы, при которых:

- компьютер 1 — типа DTE, скремблер — типа DTE;
- компьютер 1 — типа DTE, скремблер — типа DCE;
- компьютер 1 — типа DCE, скремблер — типа DTE;
- компьютер 1 — типа DCE, скремблер — типа DCE.

Замените символы “#” наименованиями сигналов. Структуры прямого и обратного каналов одинаковы. Все ли варианты реализуемы? Если не все, то почему?

Литература: [1], гл. 1 — 3; [4]

Решение задачи приведено в разделе 3 на с. 163

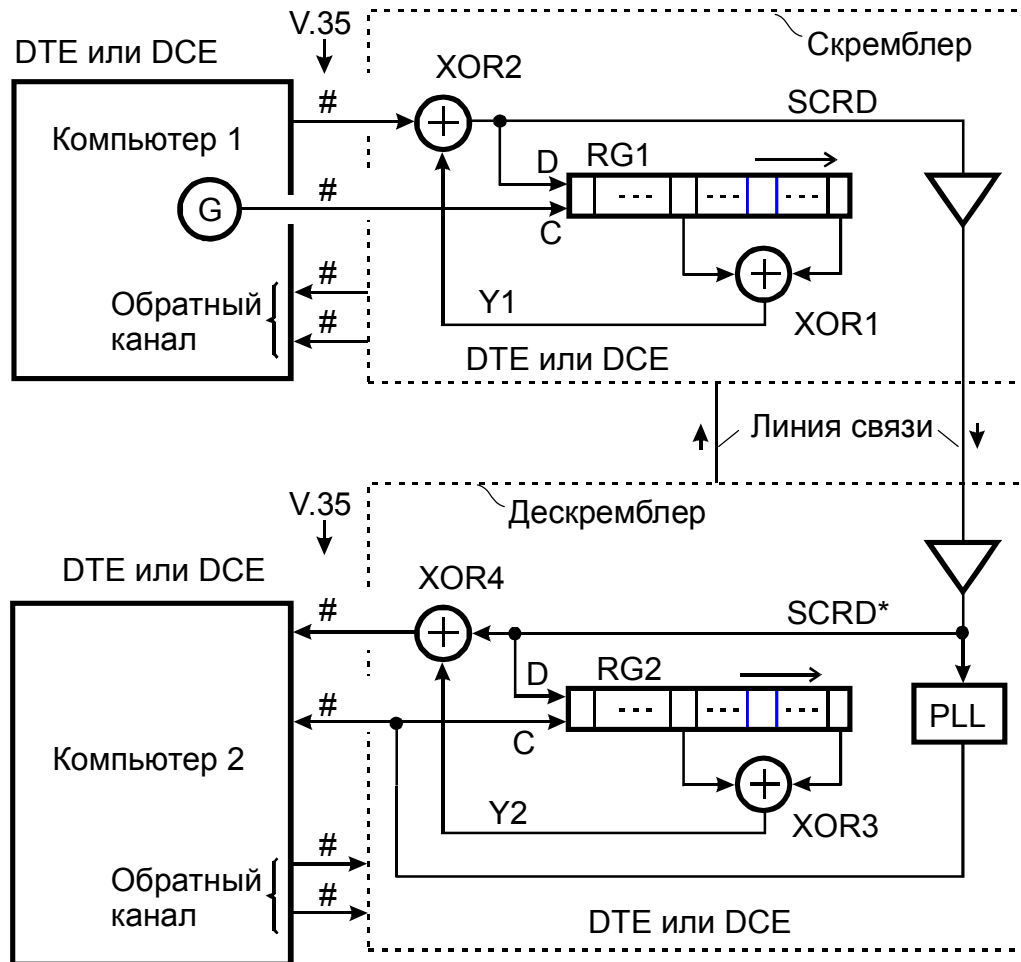


Рис. 2.18. Система передачи данных

2.2.3. На Рис. 2.19, а — д приведены варианты построения системы передачи данных, правильные и неправильные с точки зрения обеспечения синхронизации. Рассмотрите каждый вариант и оцените его правильность.

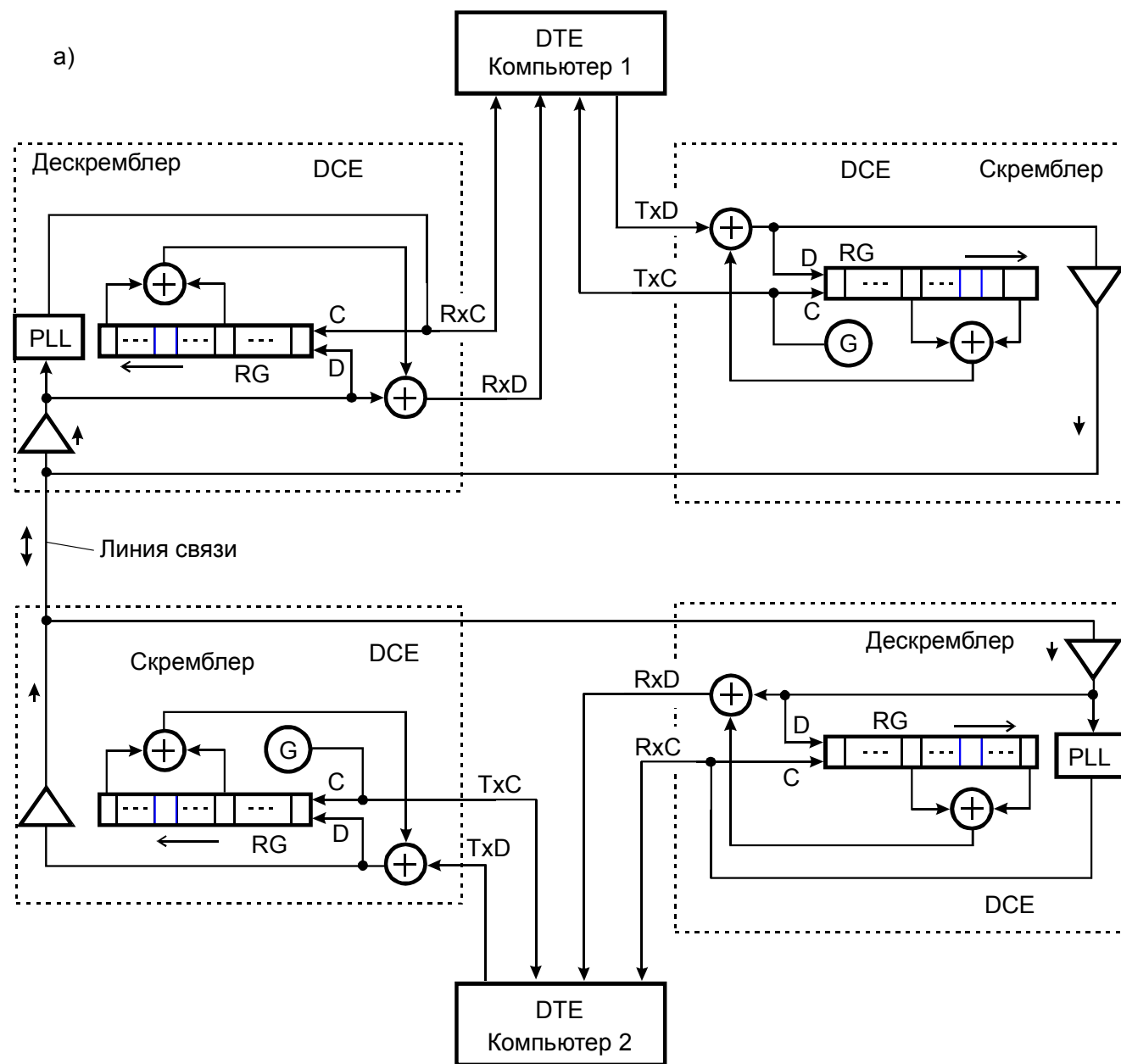


Рис. 2.19, а (см. Рис. 2.19, б)

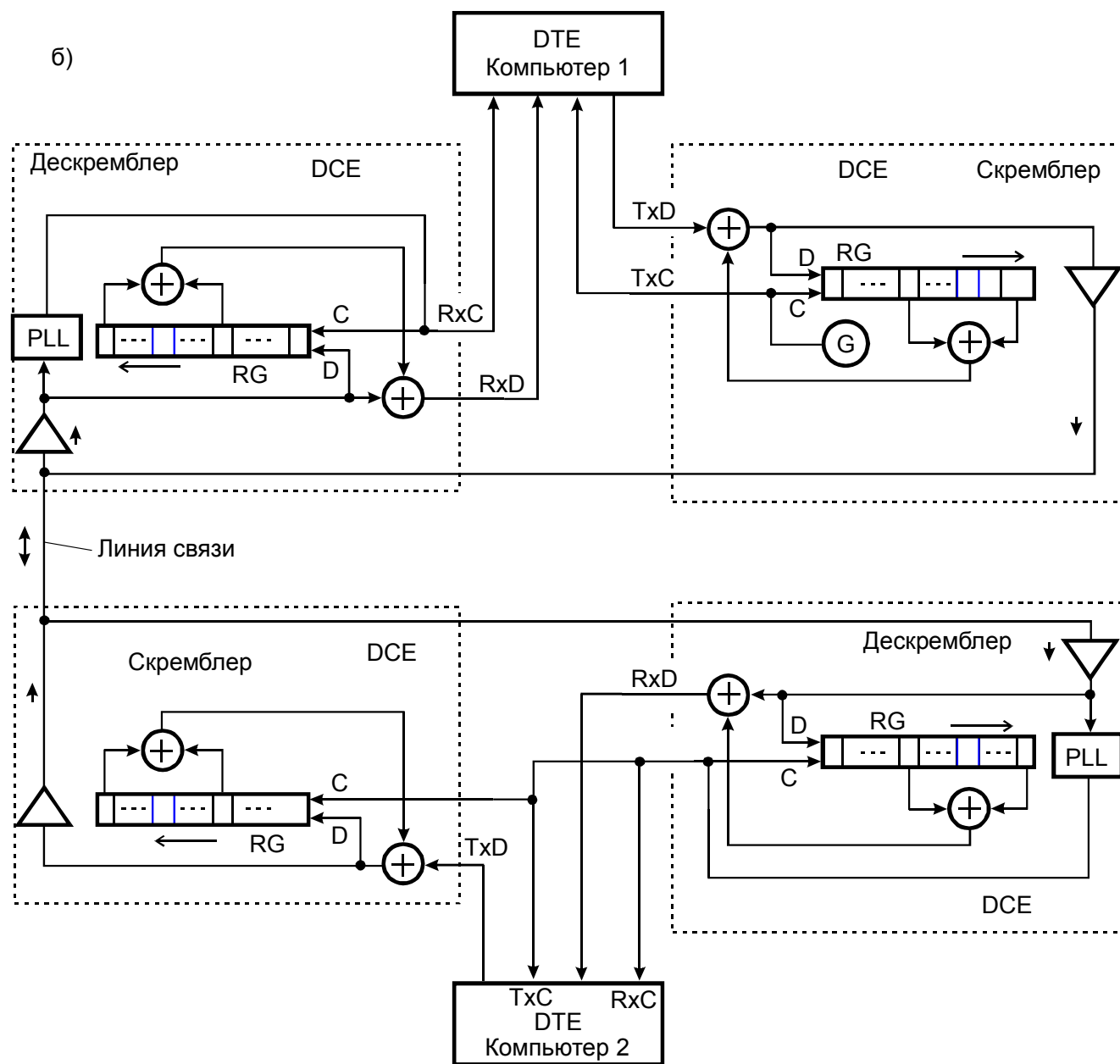


Рис. 2.19, б (см. Рис. 2.19, в)

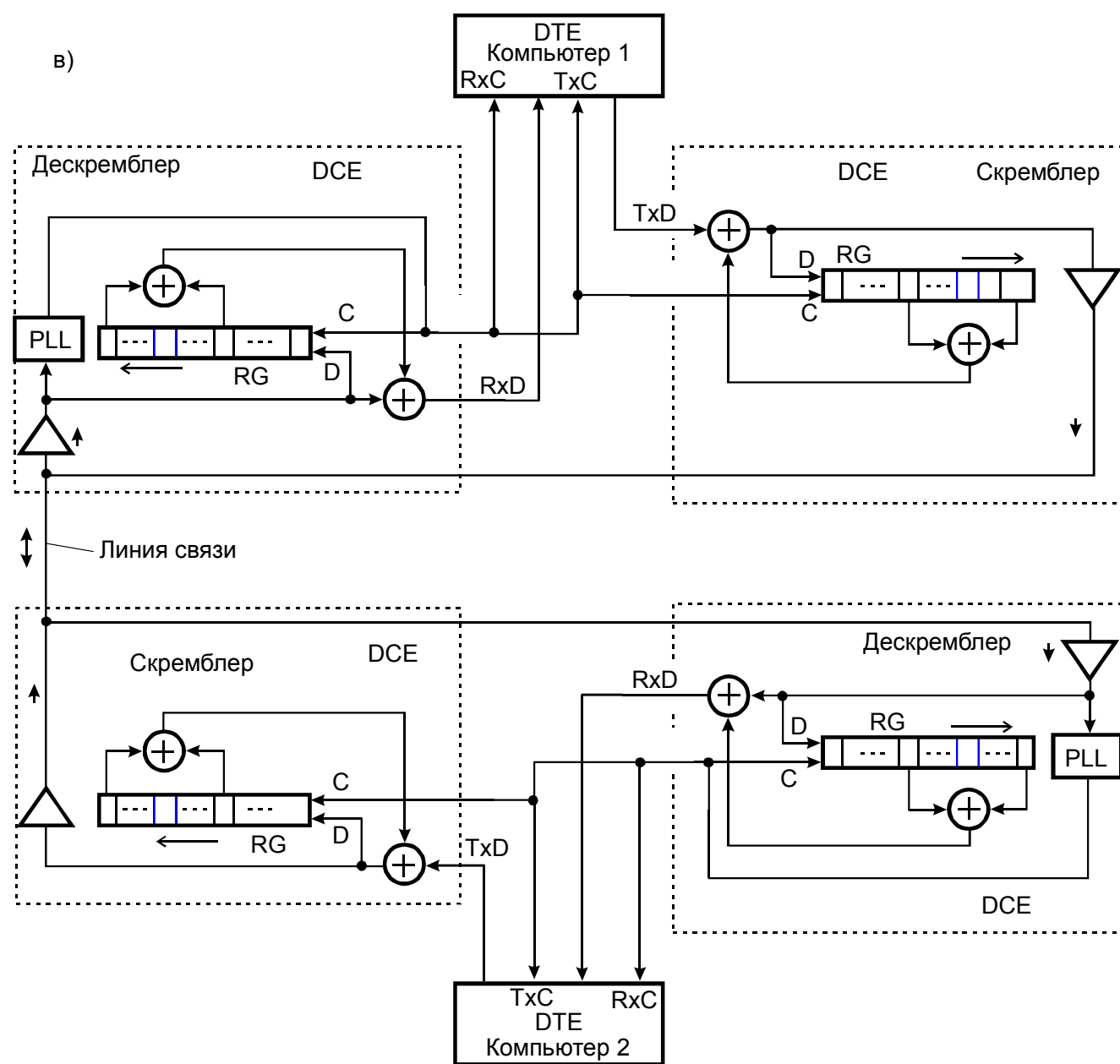


Рис. 2.19, в (см. Рис. 2.19, г)

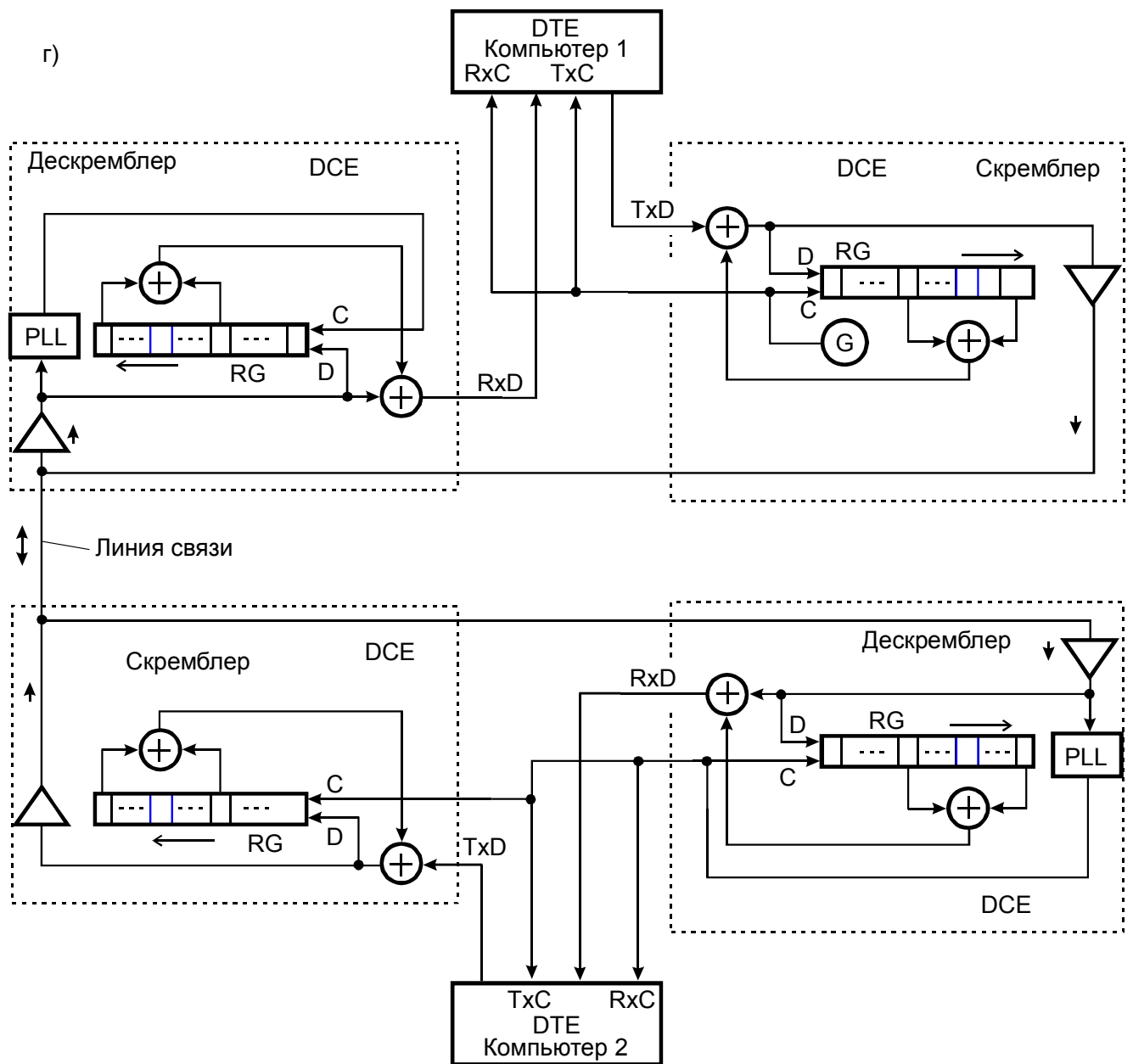


Рис. 2.19, г (см. Рис. 2.19, д)

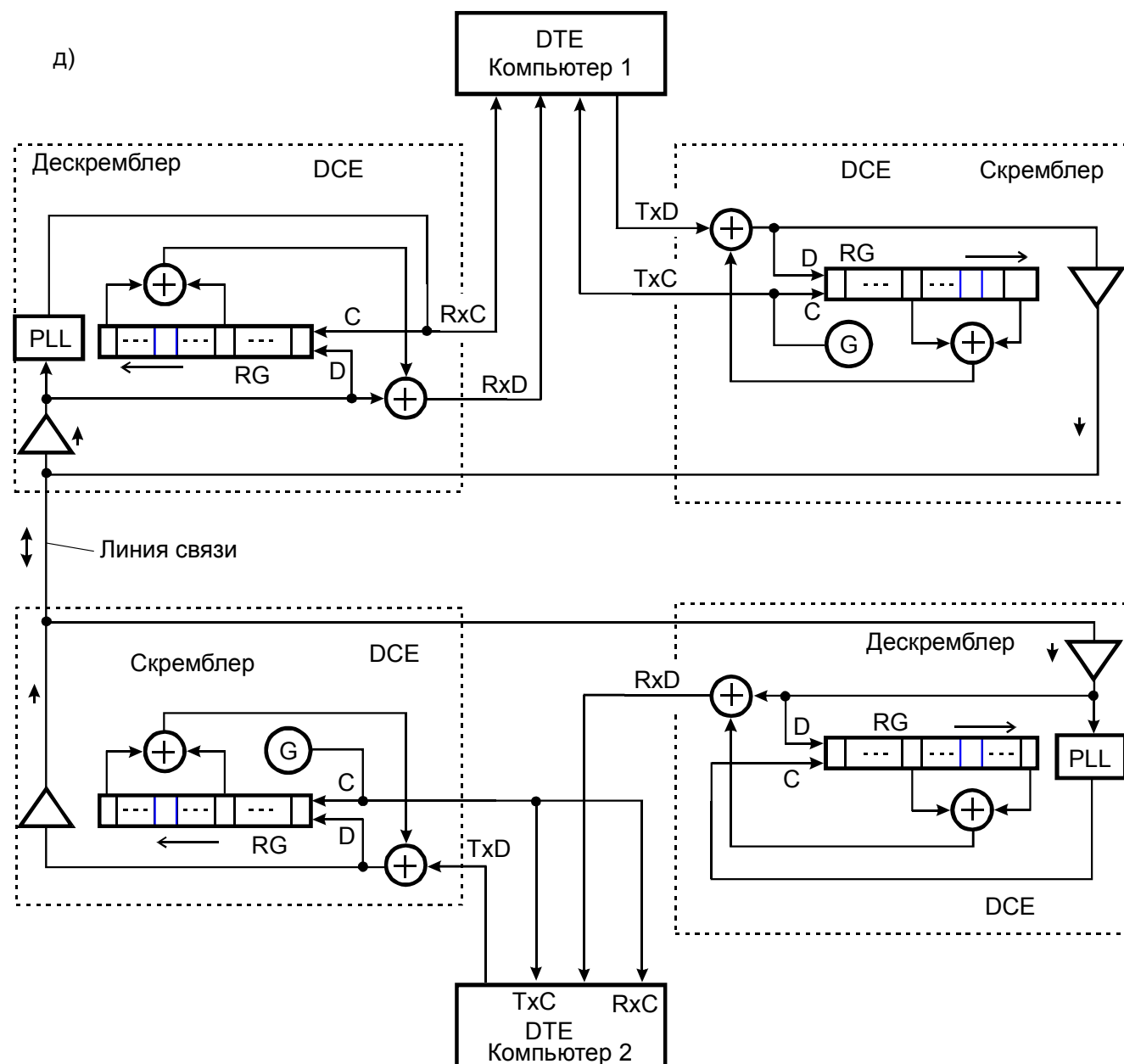


Рис. 2.19. Варианты (а — д) построения системы передачи данных, правильные и неправильные

Литература: [1], гл. 1 — 3; [4]

Решение задачи приведено в разделе 3 на с. 163

2.2.4. Какой из двух режимов асинхронного порта RS-232 обеспечивает более высокую скорость передачи пользовательских данных:

первый режим: скорость передачи — 38,4 кбит/с, семибитовый код данных, один стоп-бит, контроль по чётности (нечётности) отсутствует;

второй режим: скорость передачи — 38,4 кбит/с, восьмибитовый код данных, два стоп-бита, имеется бит контроля по чётности (нечётности).

Вычислите фактическую максимальную скорость передачи данных при использовании того и другого режима.

Литература: [1], гл. 1

Решение задачи приведено в разделе 3 на с. 164

2.2.5. В системе передачи данных, показанной на Рис. 2.20, переместите генераторы G1 и G2 соответственно в устройства DTE1 и DTE3. Приведите в соответствие с внесёнными изменениями направления передачи сигналов и их наименования. Каким образом сигнал синхронизации, обозначенный символом “#”, распространяется против течения потока данных?

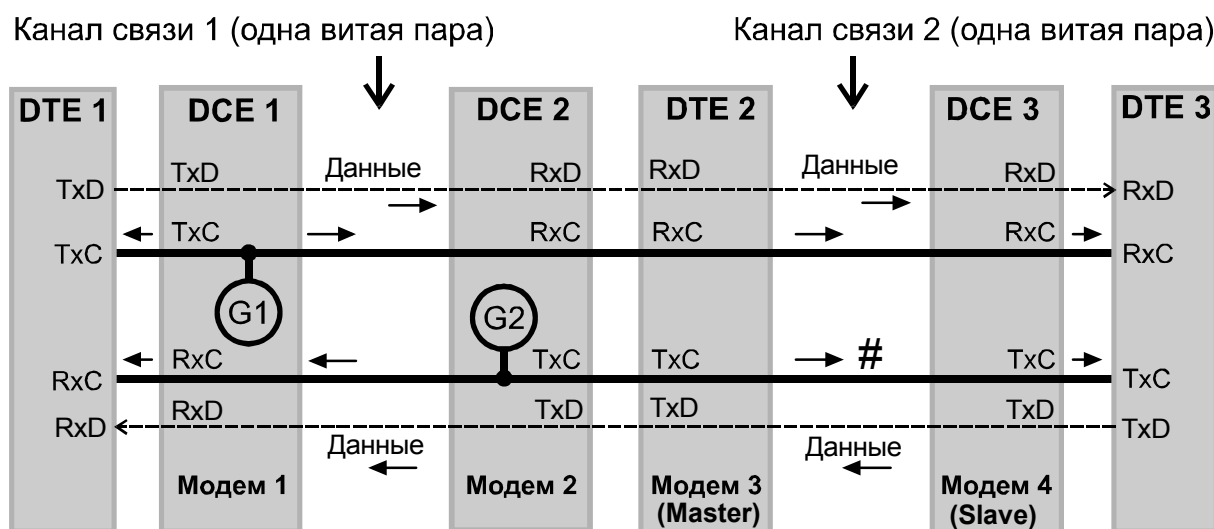


Рис. 2.20. Исходная система передачи данных с двумя последовательно включёнными каналами связи

Литература: [1], гл. 1 — 3

Решение задачи приведено в разделе 3 на с. 164

- 2.2.6. Из системы передачи данных, показанной на Рис. 2.21, удалите генератор G1. Оба направления передачи данных должны синхронизироваться от генератора G2. Приведите в соответствие с внесёнными изменениями направления передачи сигналов и их наименования, если это необходимо. Каким образом сигнал синхронизации, обозначенный символом “#”, распространяется против течения потока данных?

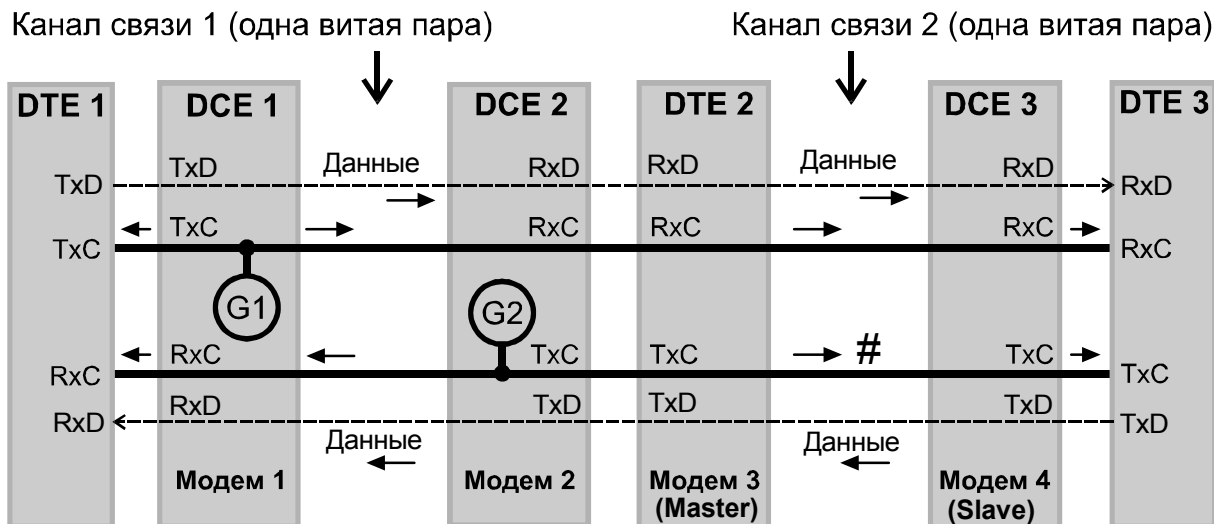


Рис. 2.21. Исходная система передачи данных с двумя последовательно включёнными каналами связи

Литература: [1], гл. 1 — 3

Решение задачи приведено в разделе 3 на с. 164

- 2.2.7. Из системы передачи данных, показанной на Рис. 2.22, удалите генератор G2. Оба направления передачи данных должны синхронизироваться от генератора G1. Приведите в соответствие с внесёнными изменениями направления передачи сигналов и их наименования, если это необходимо. Каким образом сигнал синхронизации, обозначенный символом “#”, распространяется против течения потока данных?

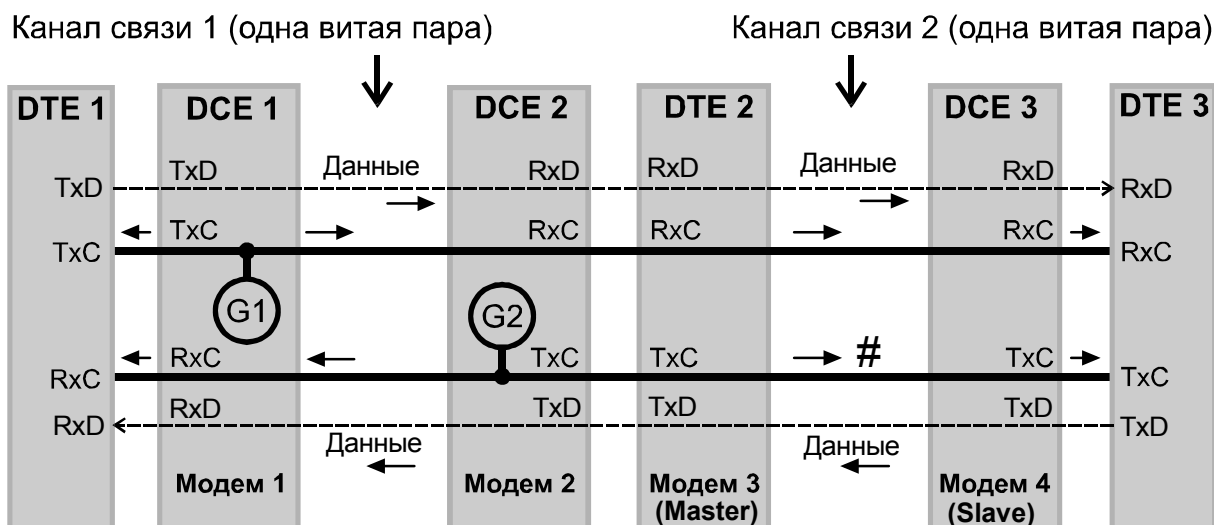


Рис. 2.22. Исходная система передачи данных с двумя последовательно включёнными каналами связи

Литература: [1], гл. 1 — 3;

Решение задачи приведено в разделе 3 на с. 165

- 2.2.8. В системе передачи данных, показанной на Рис. 2.23, переместите генераторы G1 и G2 соответственно в устройства DTE1 и DTE3. Устройства DCE замените устройствами DTE. Приведите в соответствие с внесёнными изменениями направления передачи сигналов и их наименования. Каким образом сигнал синхронизации, обозначенный символом “#”, распространяется против течения потока данных?

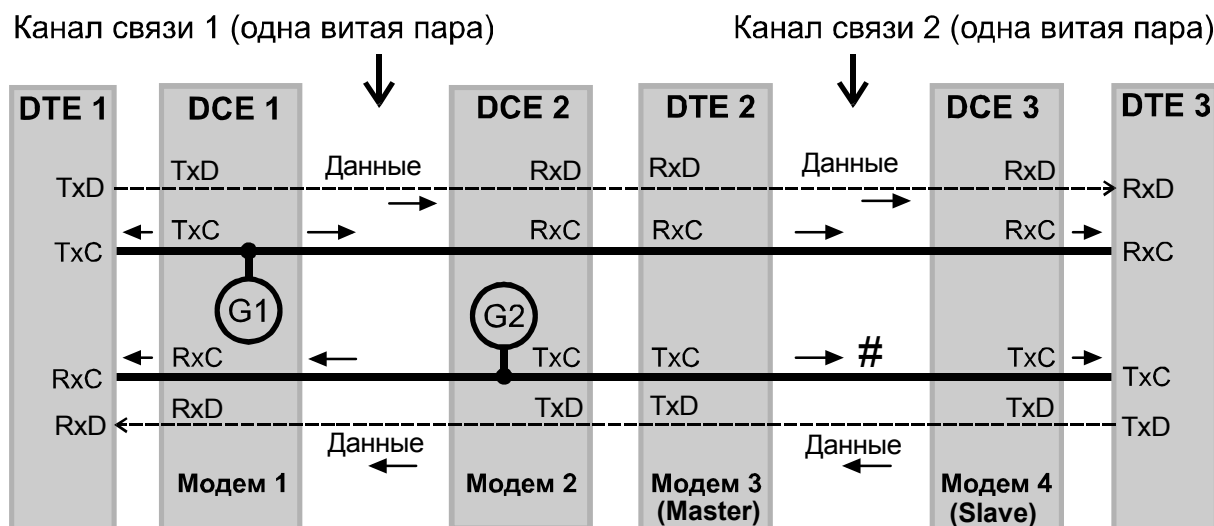


Рис. 2.23. Исходная система передачи данных с двумя последовательно включёнными каналами связи

Литература: [1], гл. 1 — 3

Решение задачи приведено в разделе 3 на с. 165

- 2.2.9. В системе передачи данных, показанной на Рис. 2.24, переместите генераторы G1 и G2 соответственно в устройства DTE1 и DTE3. Затем все устройства DTE замените устройствами DCE. Приведите в соответствие с внесёнными изменениями направления передачи сигналов и их наименования. Каким образом в исходной схеме сигнал синхронизации, обозначенный символом "#", распространяется против течения потока данных?

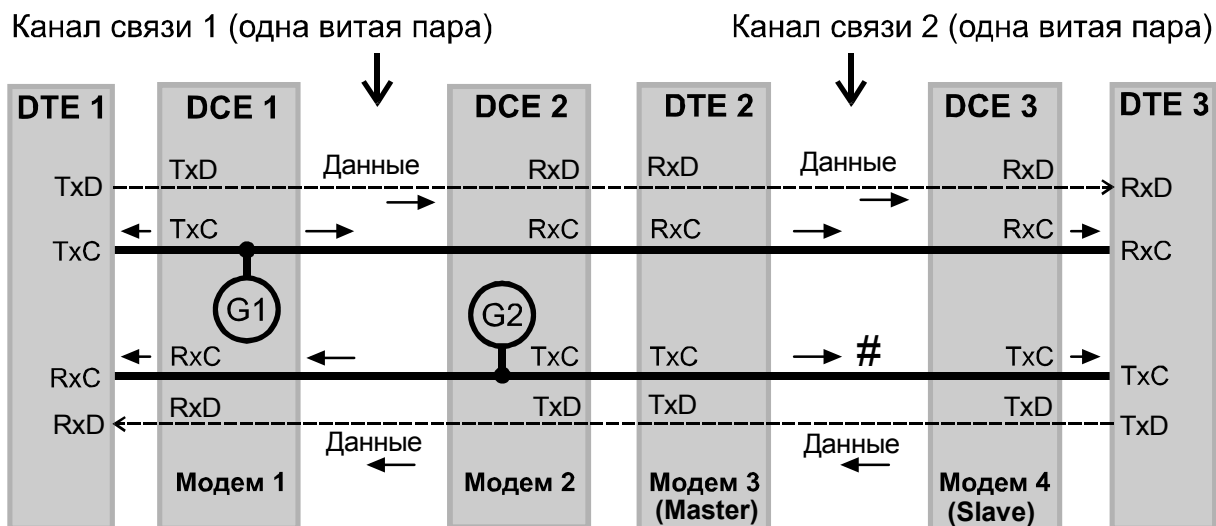


Рис. 2.24. Исходная система передачи данных с двумя последовательно включёнными каналами связи

Литература: [1], гл. 1 — 3

Решение задачи приведено в разделе 3 на с. 166

- 2.2.10. Известно, что восстановление потерянной синхронизации приёмника с передатчиком в асинхронном интерфейсе семейства RS (Рис. 2.25,) при отсутствии пауз между посылками гарантированно достигается в период передачи 7-разрядного нулевого кода данных. Если данные случайны, то вероятность обнаружения нулевого кода в любой 9-битовой старт-стоповой посылке равна  $1/2^7$ . Оцените длину цепи посылок, которая гарантирует установление синхронизации с вероятностью не менее 0,999.



Рис. 2.25. Восстановление правильной синхронизации с обнаружением ошибки

Литература: [1], гл. 1 — 3

Решение задачи приведено в разделе 3 на с. 166

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

Литература к разделу 2.3: [1], гл. 8, [4]

### 2.3.1. Введение в раздел 2.3

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. 2.26).



Рис. 2.26. Структура типового последовательного канала связи

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

### 2.3.2. Униполярный код NRZ

Простейшим линейным кодом является униполярный код типа NRZ (Non Return to Zero), показанный на Рис. 2.27, 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. 2.27, б) при той же амплитуде сигнала, равной  $A$ !

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).

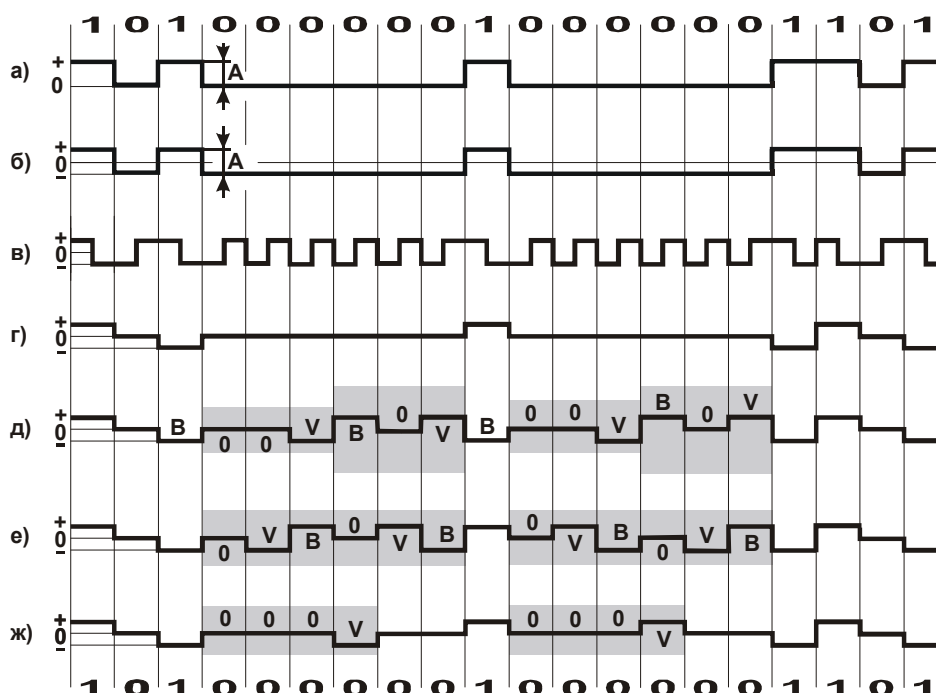


Рис. 2.27. Наиболее распространенные линейные коды: а — униполярный код NRZ; б — биполярный код NRZ; в — код Манчестер-II; г — код AMI; д — код B3ZS; е — код B6ZS; ж — код HDB3; затемненными прямоугольниками выделены “заготовки”

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

### 2.3.3. Биполярный код NRZ

Биполярный сигнал NRZ (Рис. 2.27, 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.

### 2.3.4. 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. 2.27, 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 устраняются сразу все отмеченные ранее недостатки. Поскольку число положительных и отрицательных импульсов на любом достаточно большом от-

резке 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. уже нет опасности потери синхронизации при передаче длинных цепочек нулей или единиц.

Сигнал 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.

### 2.3.5. Код AMI

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

На Рис. 2.27, *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.

### 2.3.6. Коды BNZS, HDB3

Потеря синхронизации при передаче длинной последовательности нулей предотвращается так: цепочки нулей передатчик заменяет определенными “заготовками”, которые представляют собой “отрезки” стандартных временных диаграмм. Коды AMI, в которых цепочка из  $N$  нулей заменяется определенной подстановкой, называются BNZS-кодами (Bipolar with  $N$  Zeroes Substitution).

В коде B3ZS (Рис. 2.27, *д*) каждые три последовательных нуля подменяются либо комбинацией B0V, либо 00V. Символ B обозначает импульс, который отвечает правилам кодирования AMI, символ V - импульс, который нарушает правила кодирования AMI (совпадает по полярности с предыдущим).

Выбор одной из этих двух “заготовок” проводится так, чтобы, во-первых, число импульсов B между двумя последовательно расположенными импульсами V было нечетным, и, во-вторых, чтобы полярность импульсов V чередовалась.

В коде B6ZS (Рис. 2.27, *е*) каждые шесть последовательных нулей подменяются комбинацией 0VB0VB.

Коды BNZS получили широкое распространение в компьютерных сетях США и Канады: линии T1 — 1,544 Мбит/с, T1C — 3,152 Мбит/с, LD-4 — 274,176 Мбит/с, T4 — 274,176 Мбит/с.

В странах Западной Европы широко используется код HDB3 для работы на скоростях 2,048 и 8,448 Мбит/с. Этот код очень похож на BNZS, поскольку максимально допустимое число нулей, стоящих в цепочке, равно трем.

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

двумя последовательными импульсами  $V$  было нечетным, и, во-вторых, чтобы полярность импульсов  $V$  чередовалась (Рис. 2.27, ж).

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

### 2.3.7. 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. Рассмотрим одно из таких решений [1]. 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, приведенной на Рис. 2.28.

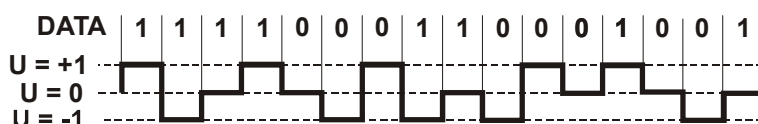


Рис. 2.28. Временная диаграмма сигнала в линии

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. 2.29, 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.

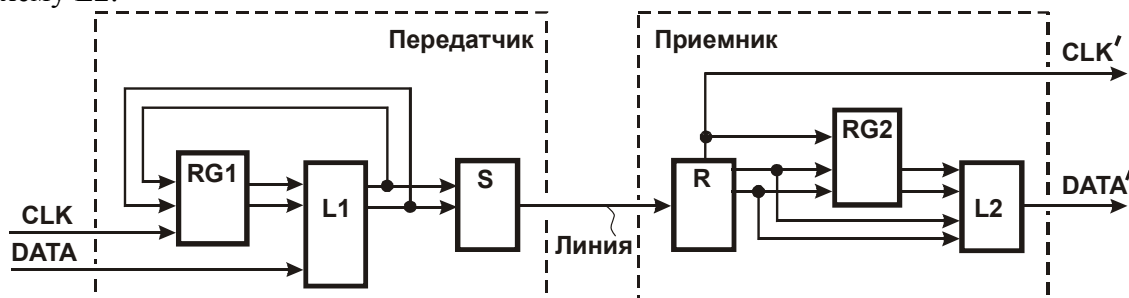


Рис. 2.29. Система передачи данных

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 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. 2.30.

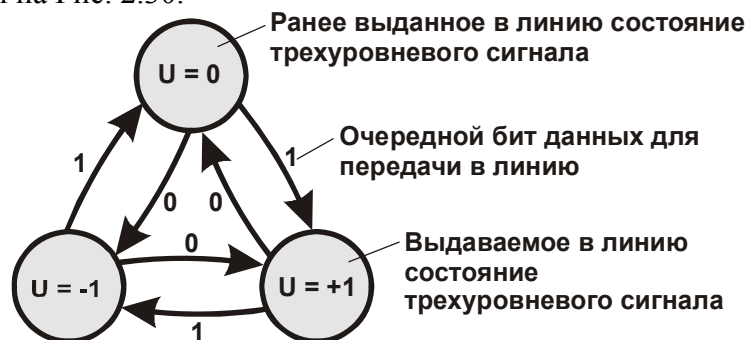


Рис. 2.30. Диаграмма состояний передатчика

Передатчик может находиться в трех состояниях, выделенных кружками. Эти состояния обозначены в соответствии с принятыми ранее сокращениями (см. Рис. 2.28).

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 двухуровневые (см. Рис. 2.29) 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. 2.31).

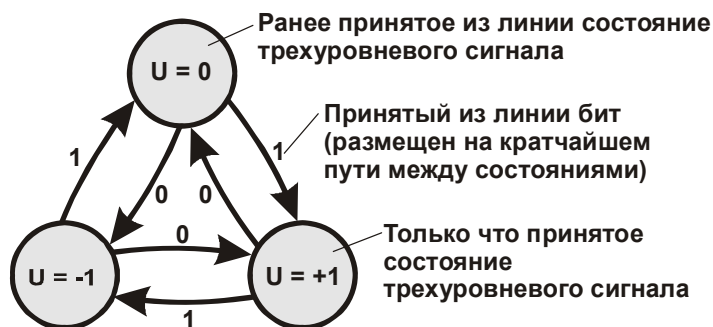


Рис. 2.31. Диаграмма состояний приемника

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 [4].

## 2.4. Вопросы и задачи к разделу 2.3

- 2.4.1. Покажите, что биполярное кодирование энергетически более выгодно по сравнению с однополярным. Каков выигрыш в мощности, требуемой для передачи сигнала?

Литература: [1], гл. 8

Решение задачи приведено в разделе 3 на с. 167

- 2.4.2. На Рис. 2.32, а показан сигнал “Манчестер-II”, на Рис. 2.32, б — д — варианты его разделения на битовые интервалы. Какой вариант правильный? Обоснуйте свой выбор. Напишите последовательность передаваемых битов.

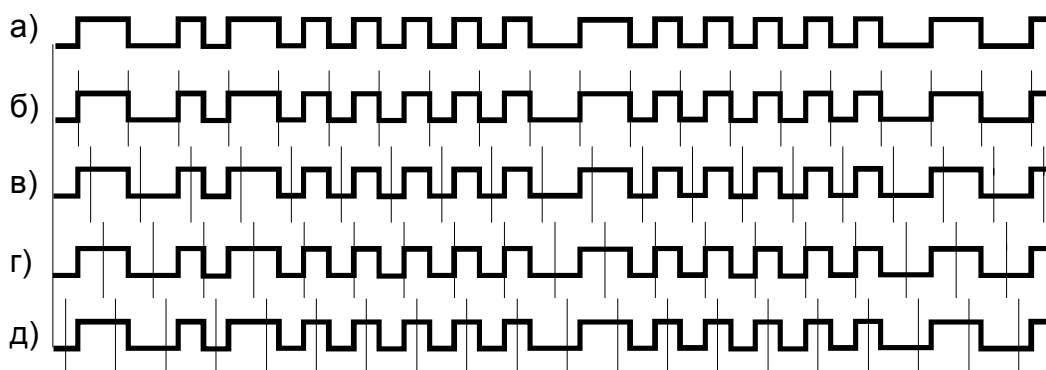


Рис. 2.32. Сигнал “Манчестер-II” (а) и предполагаемые варианты (б — д) его разделения на битовые интервалы

Литература: [1], гл. 8

Решение задачи приведено в разделе 3 на с. 167

- 2.4.3. На Рис. 2.33 показан сигнал “Манчестер-II”, содержащий ошибку. Покажите её положение. Напишите последовательность передаваемых битов.



Рис. 2.33. Сигнал “Манчестер-II”

Литература: [1], гл. 8

Решение задачи приведено в разделе 3 на с. 168

- 2.4.4. Начертите временную диаграмму сигнала B6ZS, соответствующую передаче последовательности ...000000110000001100000011000000110... .

Литература: [1], гл. 8

Решение задачи приведено в разделе 3 на с. 168

- 2.4.5. Начертите временные диаграммы сигналов B3ZS и B6ZS, соответствующих передаче длинной цепочки лог. 0.  
Для определённости предполагаем, что предыстория сигнала завершается положительным импульсом V.

Литература: [1], гл. 8

Решение задачи приведено в разделе 3 на с. 168

- 2.4.6. Начертите временную диаграмму сигнала B3ZS, соответствующего передаче последовательности 1000100010001000100010001000... . Для определённости предполагаем, что предыстория сигнала завершается отрицательным импульсом V.

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 169*

- 2.4.7. Начертите временную диаграмму сигнала B3ZS, соответствующего передаче последовательности 1100011000110001100011000110001100011000... . Для определённости предполагаем, что предыстория сигнала завершается отрицательным импульсом V.

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 169*

- 2.4.8. Начертите временную диаграмму сигнала HDB3, соответствующего передаче длинной цепочки лог. 0. Для определённости предполагаем, что предыстория сигнала завершается положительным импульсом V.

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 169*

- 2.4.9. Начертите временную диаграмму сигнала HDB3, соответствующего передаче последовательности 10000100001000010000100001000010000... . Для определённости предполагаем, что предыстория сигнала завершается отрицательным импульсом V.

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 170*

- 2.4.10. Начертите временную диаграмму сигнала HDB3, соответствующего передаче последовательности 11000011000011000011000011000011000011000011000... . Для определённости предполагаем, что предыстория сигнала завершается отрицательным импульсом V.

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 170*

- 2.4.11. Начертите временную диаграмму сигнала B6ZS, соответствующего передаче последовательности ...000000100000010000001000000... .

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 171*

- 2.4.12. Пользуясь правилами кодирования, описанными в п. 2.3.7, начертите временную диаграмму сигнала, соответствующего передаче последовательности ...1011011100111000... . Однозначно ли предполагаемое решение задачи?

Литература: [1], гл. 8

*Решение задачи приведено в разделе 3 на с. 171*

## 2.5. Вероятностная синхронизация

Литература к разделу 2.5: [4, 5]

### 2.5.1. Введение в раздел 2.5

Вероятностная синхронизация в телекоммуникационных системах позволяет уменьшить долю служебных битов в потоках данных

Термин “вероятностная синхронизация” (“стохастическая синхронизация”) используется в технике при описании некоторых процессов, протекающих в сложных системах. В данном случае этот термин отражает смысл предлагаемой идеи, которая состоит в том, что удалённые друг от друга телекоммуникационные устройства, например мультиплексоры, взаимно координируют свои действия в результате одновременной реакции на некоторые случайные события, являющиеся “побочными продуктами” передачи потока скремблированных данных.

Скремблирование — это шифрация потока данных, после которой он выглядит как поток случайных битов. Последовательности битов в исходном массиве данных, как регулярные, так и нерегулярные, обратимо разрушаются, так что вероятности появления логической единицы и логического нуля в каждой последующей битовой позиции скремблированного потока одинаковы и не зависят от предыстории. Известно, что применительно к телекоммуникационным системам скремблирование повышает надёжность битовой синхронизации между устройствами, подключёнными к противоположным сторонам канала связи, и уменьшает уровень помех, излучаемых на соседние линии многожильного кабеля [4].

Проведенное автором исследование ранее не рассмотренных в литературе функциональных возможностей систем со скремблированием данных показало, что в результате некоторого усовершенствования таких систем достигается весьма полезное свойство самосинхронизации приёмника с передатчиком на уровне распознавания информационных кадров или иных структурных единиц. Полезность состоит в уменьшении доли служебных битов в потоках данных.

В передаваемом по каналу связи скремблированном потоке данных с некоторой средней периодичностью можно обнаруживать любые заранее заданные сочетания битов. Так, при однократной выборке 30-разрядного кода из проходящего потока данных вероятность его совпадения с 30-разрядным эталоном составит  $2^{-30} \approx 10^{-9}$ . При скорости потока 10 Гбит/с средняя частота событий одновременного (с точностью до задержки передачи) обнаружения заданного кода на разных сторонах канала связи составляет 10 Гц. Это означает, что передатчик и приёмник со средней периодичностью 10 Гц одновременно получают некие метки времени, которые самопроизвольно порождаются “полезными” или иными произвольными данными, а не создаются, как это делается в настоящее время, введением в поток служебной синхронизирующей информации, например, последовательностей флаговых кодов.

В рамках дисциплины “Синхронизация в телекоммуникационных системах” рассмотрены примеры применения идеи вероятностной синхронизации для решения следующих задач.

1. *Синхронизация системы “скремблер — дескремблер”.* Для предотвращения размножения ошибок, возникающих в канале связи, и защиты от неблагоприятных кодовых ситуаций в таких системах используют логически изолированные от линии генераторы псевдослучайных последовательностей битов. При этом в существующих системах для синхронизации работы этих генераторов используют служебные информационные кадры, что снижает скорость передачи полезных данных. Применение вероятностной синхронизации позволяет сохранить все преимущества системы “скремблер — дескремблер” с изолированными генераторами псевдослучайных последовательностей битов и повысить пропускную способность канала связи. Последнее достигается исключением служебных кадров и программных средств установления и поддержания синхронизации между скремблером и дескремблером [4].

2. *Вставка команд в поток данных.* Задача заключается в распознавании на дальней стороне канала связи команд, вставленных передатчиком в поток данных. В существующих системах

для маркировки команд используется служебная информация, полученная в результате применения к массиву исходных данных операций битстаффинга или байтстаффинга. Использование вероятностной синхронизации позволяет исключить из потока данных служебную информацию и вставлять команды в случайные моменты обнаружения заданных кодов в скремблированном потоке данных [5].

3. *Разграничение байтов в битовом потоке данных.* Для разграничения байтов обычно используются служебные коды, например, флаги начала кадров в стандарте E1. Применение вероятностной синхронизации позволяет устанавливать и всякий раз при обнаружении в скремблированном потоке заданных кодов пересматривать временные границы между байтами или иными структурными единицами. События обнаружения заданных кодов следуют сравнительно редко (например, с частотой 10 Гц), поэтому потери (составляющие в среднем 3,5 бита при каждом согласовании временных сеток передатчика и приёмника) составляют пренебрежимо малую часть общего потока данных, передаваемых по каналу связи. При скорости передачи данных 10 Гбит/с избыточность составляет 3,5 бита на  $10^9$  бит [7].

4. *Разграничение каналов в мультиплексированном потоке данных.* В двухканальной системе “мультиплексор — демультиплексор”, предназначенной для объединения — разделения потоков данных при их передаче по линии связи, как и в ранее упомянутых здесь системах, применение вероятностной синхронизации позволяет сократить число избыточных битов до пренебрежимо малого уровня, исключить программные средства поддержки синхронизации и уменьшить время её восстановления в случае потери [8].

5. *Ускорение передачи информационных кадров.* В системе передачи данных с попутной синхронизацией (линии TxD — CLK) по линии TxD предлагается передавать информационные кадры “стык-в-стык” без использования каких-либо разделительных служебных битов. При этом по линии CLK помимо сигналов битовой синхронизации дополнительно передаются как полезные данные, так и коды, позволяющие приёмнику отыскать границы между информационными кадрами. Эти коды формируются в случайные моменты обнаружения в линии TxD заранее заданных последовательностей битов [9].

## 2.6. Вопросы и задачи к разделу 2.5

2.6.1. В системе передачи данных, показанной на Рис. 2.34, применена вероятностная синхронизация. Определите скорость передачи командных битов при следующих условиях:

- скорость передачи битов по линии связи равна 1 Гбит/с;
- каждая командная вставка содержит 4 бита;
- разрядность регистров RG1 ... RG4 — 20 бит;
- число кодов, анализируемых дешифраторами, — 3.

Литература: [4, 5]

Решение задачи приведено в разделе 3 на с. 171

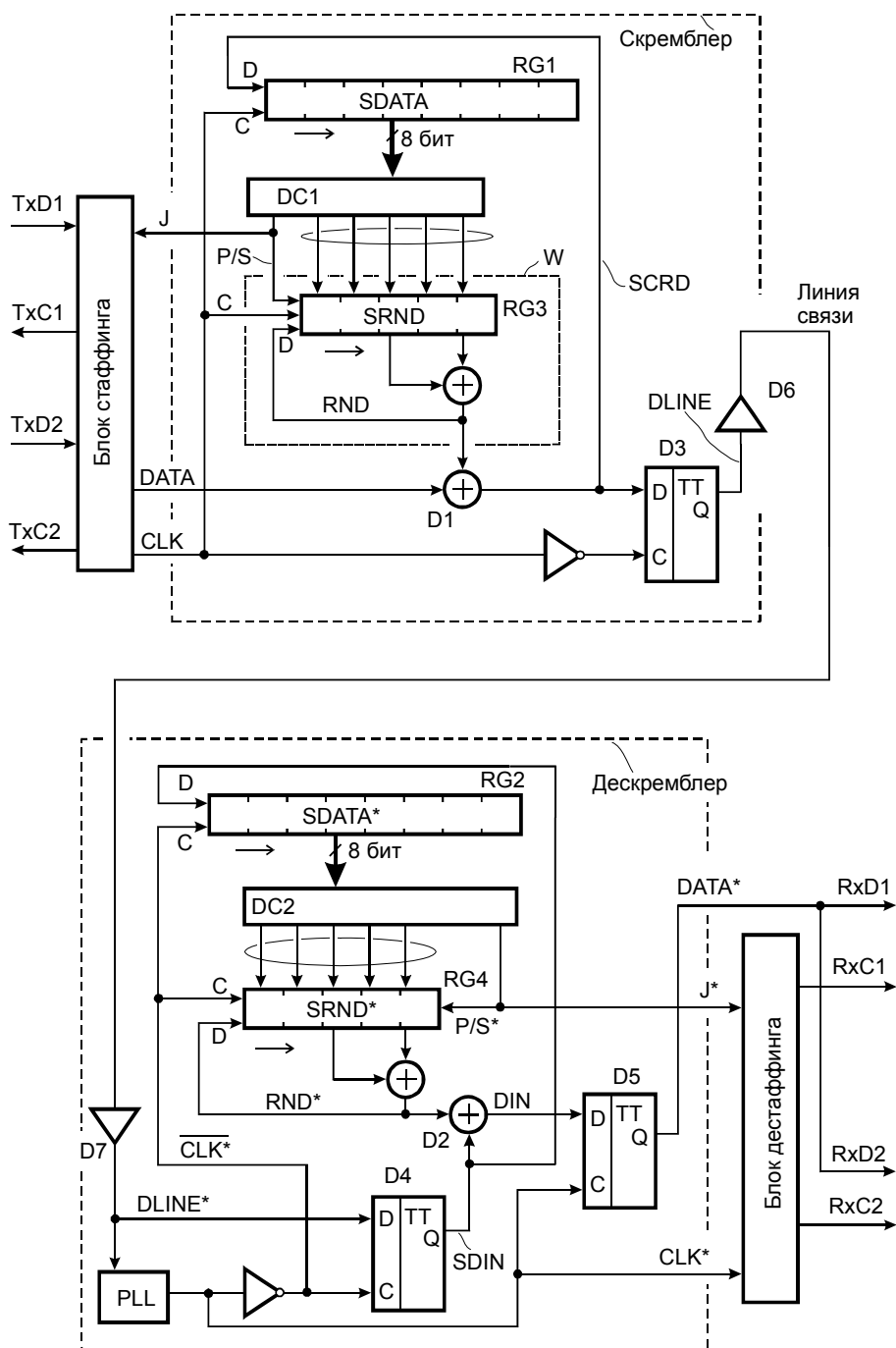


Рис. 2.34. Система передачи данных (к задаче 2.6.1)

- 2.6.2. В системе передачи данных (Рис. 2.35) применена вероятностная синхронизация. Скремблер и дескремблер выполнены по схеме с изолированными генераторами псевдослучайных последовательностей битов. Скорость передачи данных — 10 Гбит/с, разрядность синхронизирующего кода — 30 бит. Вычислите среднее время между импульсами вероятностной синхронизации. В чём её преимущество?

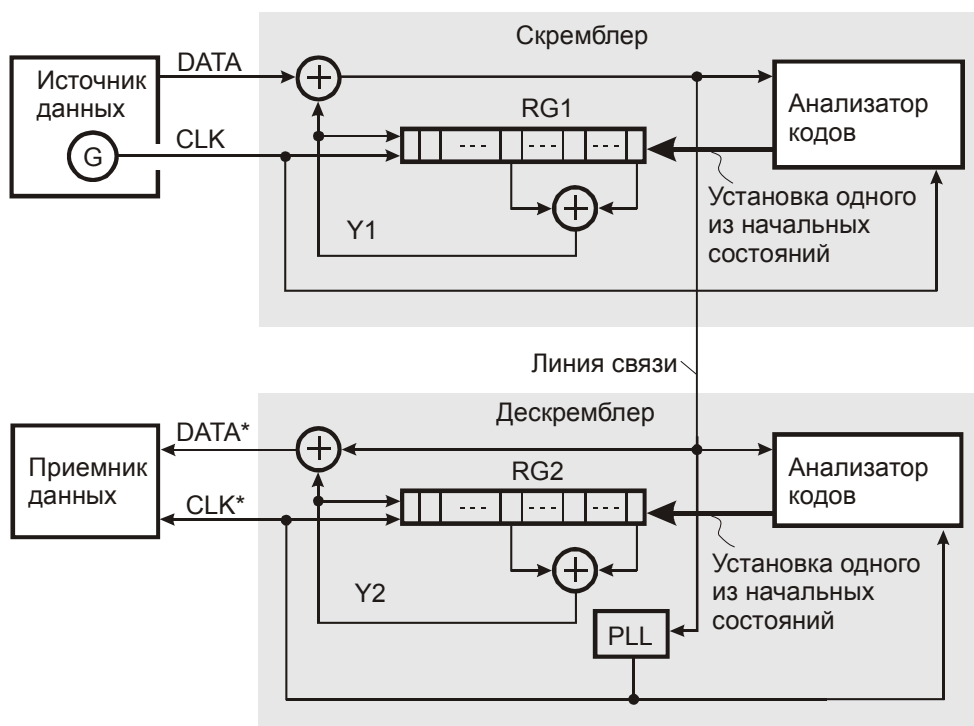


Рис. 2.35. Система передачи данных с вероятностной синхронизацией

Литература: [4]

Решение задачи приведено в разделе 3 на с. 171

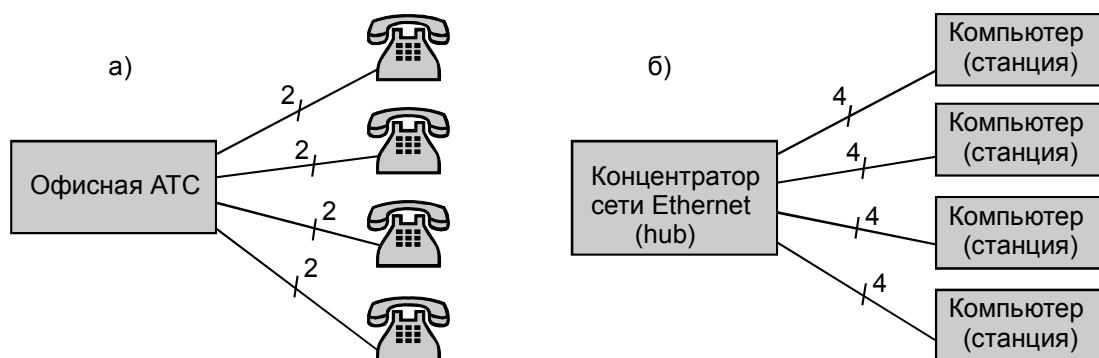
## 2.7. Передача данных по фантомным цепям каналов связи

Литература к разделу 2.7: [6]

### 2.7.1. Введение в раздел 2.7

Известно, что кабельные линии связи, содержащие группу витых пар медных проводов, обладают некоторыми “скрытыми возможностями”. Они заключаются в том, что помимо использования витых пар по прямому назначению (для передачи данных) каждая из них может рассматриваться как отдельный провод. Совокупность таких проводов позволяет без дополнительного увеличения числа жил кабеля сформировать дополнительные (фантомные) каналы связи или (и) цепи питания удалённых устройств. Основные идеи построения фантомных каналов и цепей предложены более 30 лет назад и, по существу, с тех пор мало изменились. Тем не менее, рассмотренные здесь новые решения могут оказаться полезными при проектировании систем передачи данных. В частности, это относится к использованию общей кабельной инфраструктуры для построения офисной телефонной сети и сети Ethernet 10 Base T.

Рабочие места сотрудников некоторого офиса, расположенного в одном здании, обычно оснащаются индивидуальными компьютерами, объединёнными локальной сетью, и телефонными аппаратами, аналоговыми или цифровыми. Телефонные аппараты с помощью двухпроводных линий связи соединяются с офисной АТС (Рис. 2.36, а), которая имеет несколько выходов в городскую телефонную сеть (эти выходы на рисунке не показаны).



**Рис. 2.36. Структуры связей офисной телефонной сети (а) и локальной сети Ethernet 10 Base T (б)**

Каждый компьютер (Рис. 2.36, б) соединяется с концентратором (hub) локальной сети Ethernet 10 Base T с помощью двух витых пар проводов. По каждой паре данные передаются в одну сторону. Как видно из рисунка, структуры связей офисной телефонной сети и локальной сети топологически схожи. Поэтому целесообразно использовать одни и те же витые пары проводов как для построения локальной сети, так и для связи офисной АТС с телефонными аппаратами на рабочих местах.

Одно из решений задачи построения общей кабельной инфраструктуры основано на частотном разделении сигналов локальной сети и сигналов офисной АТС (Рис. 2.37). Каждое рабочее место содержит компьютер и телефонный аппарат. Телефонный аппарат подключается к офисной АТС через фильтры низких частот и линию связи, выполненную в виде витой пары проводов. Верхняя граница спектра передаваемых сигналов при работе с аналоговыми телефонными аппаратами не превышает нескольких килогерц.

Эта же линия связи предназначена для дуплексной передачи сигналов между модемами. Спектр модемных сигналов занимает полосу от десятков килогерц до десятков мегагерц. Благодаря фильтрам сигналы разделяются по соответствующим направлениям и не взаимодействуют между собой.

Такой способ совмещения сигналов требует применения пары модемов для каждой линии связи, что может оказаться экономически невыгодным. Более простой вариант совмещения

сигналов концентратора локальной сети с сигналами офисной АТС основан на применении фантомных цепей (Рис. 2.38).

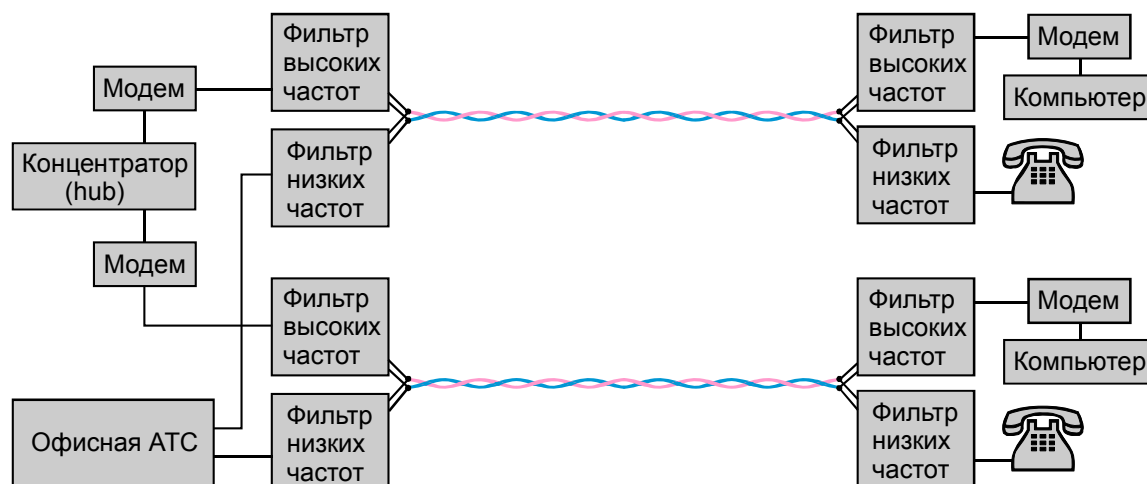


Рис. 2.37. Частотное разделение сигналов концентратора локальной сети и сигналов офисной АТС

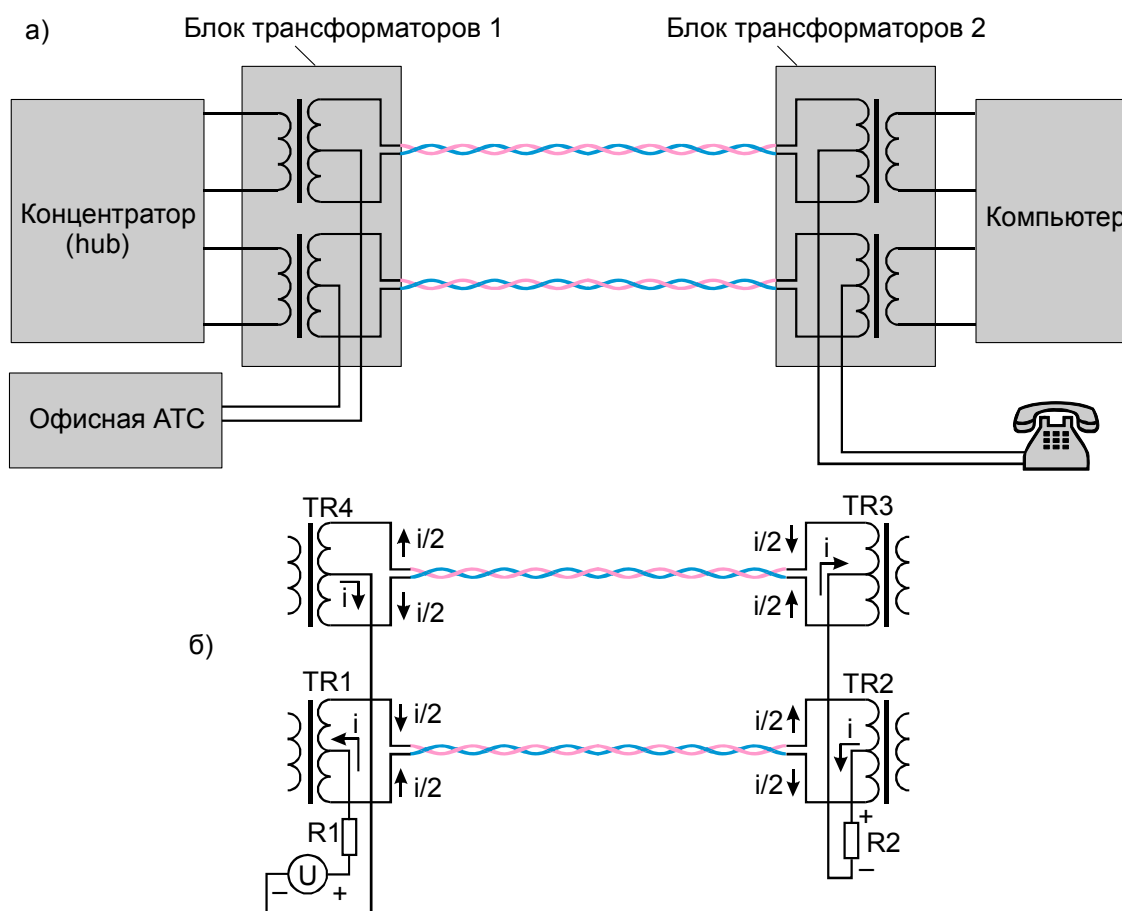


Рис. 2.38. Использование фантомной цепи для совмещения передачи сигналов концентратора локальной сети с сигналами офисной АТС (первый вариант): а — функциональная схема; б — схема протекания тока по фантомной цепи

## 2.7.2. Основная идея совмещения передачи сигналов

В схеме, показанной на Рис. 2.38, передача данных между концентратором и компьютером осуществляется по двум витым парам проводов через блоки трансформаторов. По каждой

витой паре данные передаются в одну сторону. Одновременно с этим офисная АТС взаимодействует с телефонным аппаратом, цифровым или аналоговым в зависимости от типа АТС. При таком взаимодействии каждая витая пара выполняет функцию отдельного провода, соединяющего АТС с телефонным аппаратом. Поясним сказанное с использованием эквивалентной схемы, приведенной на Рис. 2.38, б.

В эквивалентной схеме офисная АТС представлена генератором напряжения  $U$  и резистором  $R1$ ; телефонный аппарат заменён резистором  $R2$ . Ток  $i$  от положительного полюса генератора напряжения  $U1$  протекает через резистор  $R1$  в среднюю точку обмотки трансформатора  $TR1$  и растекается на два одинаковых, но противоположных по направлению тока  $i/2$ . Магнитные поля от этих токов взаимно уничтожаются в сердечнике трансформатора  $TR1$ , поэтому его подмагничивания не происходит.

Далее оба тока  $i/2$  текут по нижней линии связи (как по единому проводу) к трансформатору  $TR2$ , проходят через половины его обмоток в противоположных направлениях и суммируются в средней точке. Суммарный ток протекает через нагрузочный резистор  $R2$  и создаёт на нём падение напряжения (принимаемый сигнал). Далее ток протекает через трансформаторы  $TR3$  и  $TR4$  и возвращается к отрицательному полюсу источника напряжения  $U$ . Таким образом, офисная АТС “не мешает” передаче сигналов между концентратором и компьютером.

В рассмотренной схеме (Рис. 2.38) для взаимодействия офисной АТС с  $N$  телефонными аппаратами необходимы  $2N$  витых пар проводов. Применение схемы, показанной на Рис. 2.39, позволяет уменьшить это число до  $N + 1$ . Уменьшение достигнуто благодаря применению “общего провода”, частью которого является верхняя линия связи 1.

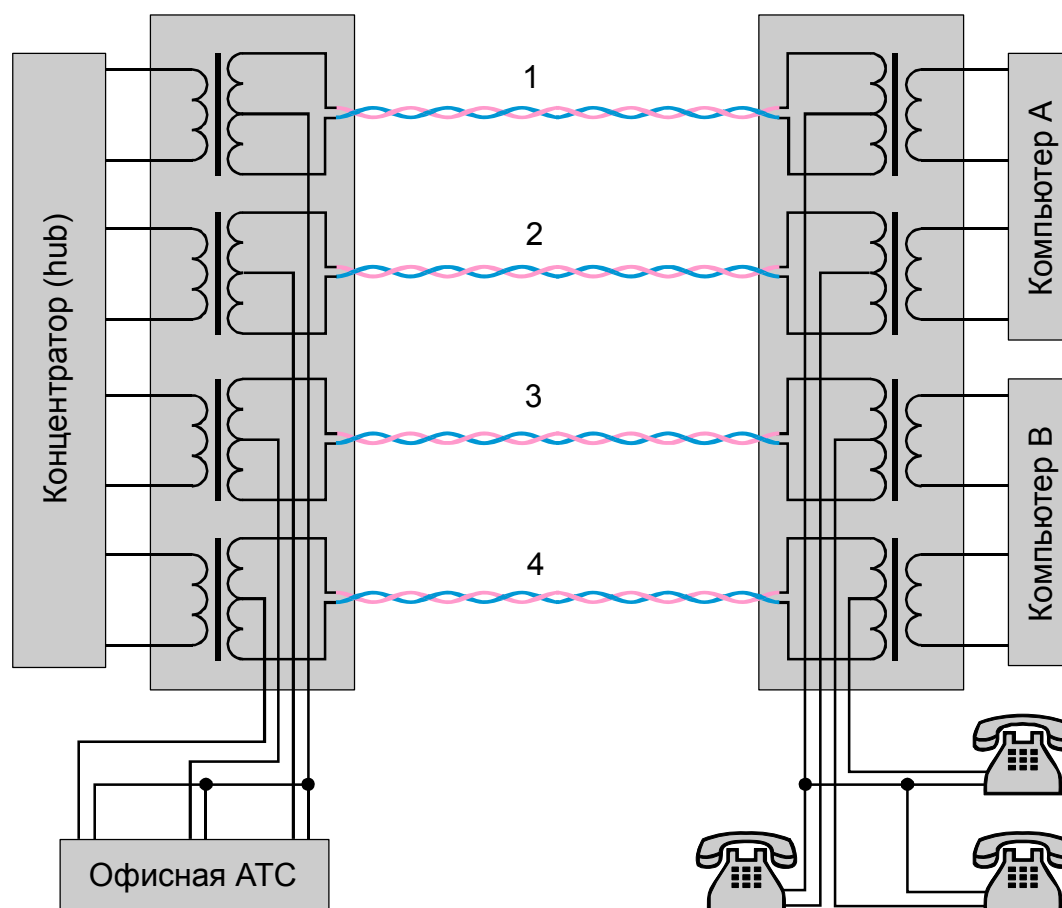


Рис. 2.39. Использование фантомных цепей для совмещения передачи сигналов концентратора локальной сети с сигналами офисной АТС (второй вариант)

### 2.7.3. Варианты конструктивных решений

Вернёмся к схеме, показанной на Рис. 2.38, *а*, и рассмотрим некоторые варианты конструктивного выполнения её составных частей. Блоки трансформаторов могут представлять собой конструктивно-законченные устройства либо входить в состав одного из устройств — концентратора, офисной АТС, компьютера или телефонного аппарата.

Блок трансформаторов 2 (Рис. 2.38, *а*) может быть размещён в сетевой карте компьютера (Рис. 2.40). В этом случае на корпусе сетевой карты дополнительно устанавливается розетка для подключения кабеля (шнура) телефонного аппарата. Аналогично блок трансформаторов можно разместить в телефонном аппарате (Рис. 2.41). Тогда на корпусе телефонного аппарата устанавливается дополнительная розетка для подключения кабеля сетевой карты компьютера.

Блок трансформаторов 2 (Рис. 2.38, *а*) можно разместить в настенной коробке, содержащей две розетки, как показано на Рис. 2.42. Первая розетка предназначена для подключения кабеля телефонного аппарата, вторая — для подключения кабеля сетевой карты компьютера.

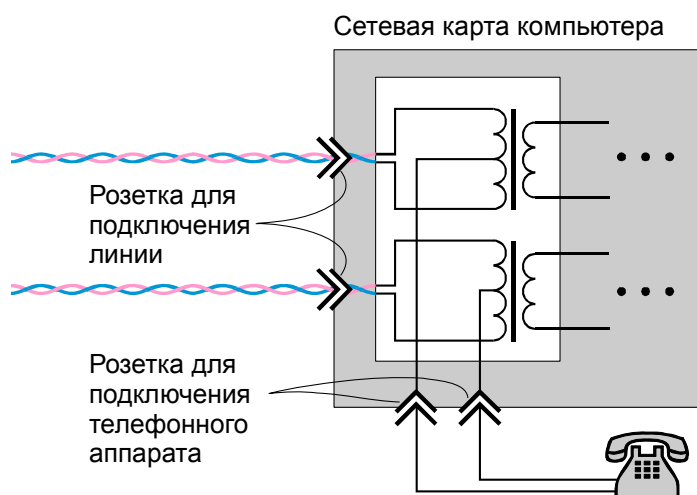


Рис. 2.40. Размещение блока трансформаторов 2 (Рис. 2.38, *а*) в сетевой карте компьютера

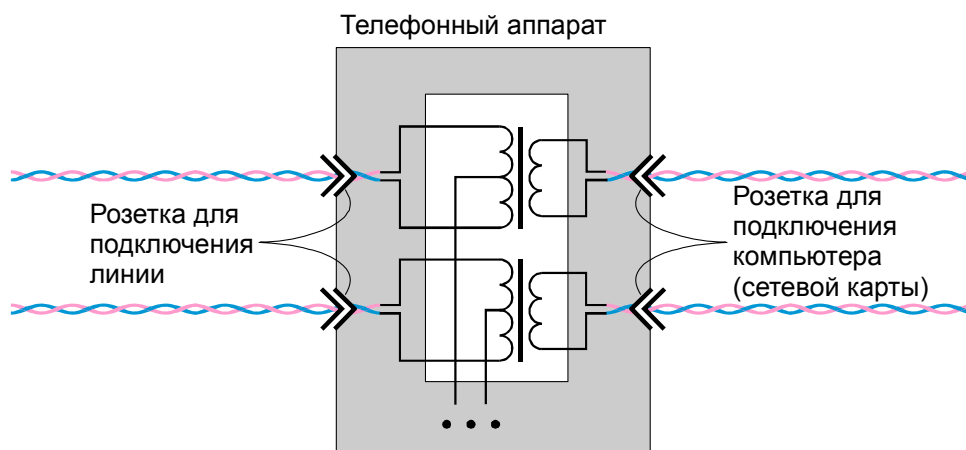
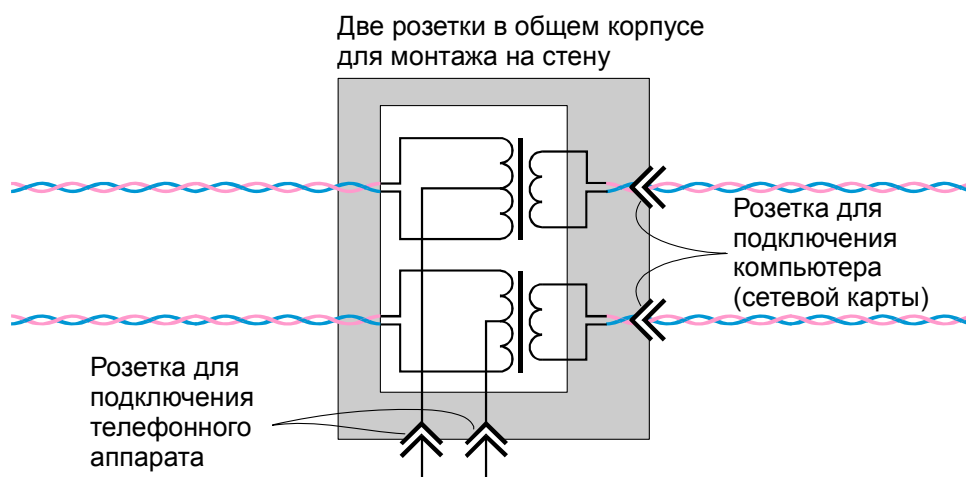
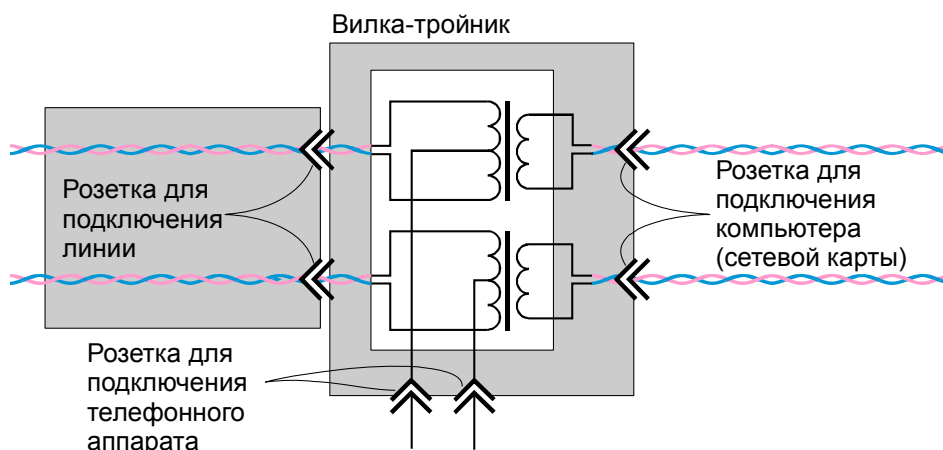


Рис. 2.41. Размещение блока трансформаторов 2 (Рис. 2.38, *а*) в телефонном аппарате



**Рис. 2.42. Размещение блока трансформаторов 2 (Рис. 2.38, а) в настенной коробке**

Чтобы не вводить изменения в существующую разводку связей локальной сети, можно воспользоваться схемой, показанной на Рис. 2.43. Блок трансформаторов 2 (Рис. 2.38, а) в данном случае размещается в вилке-тройнике, содержащем две розетки для подключения кабеля телефонного аппарата и кабеля сетевой карты компьютера.

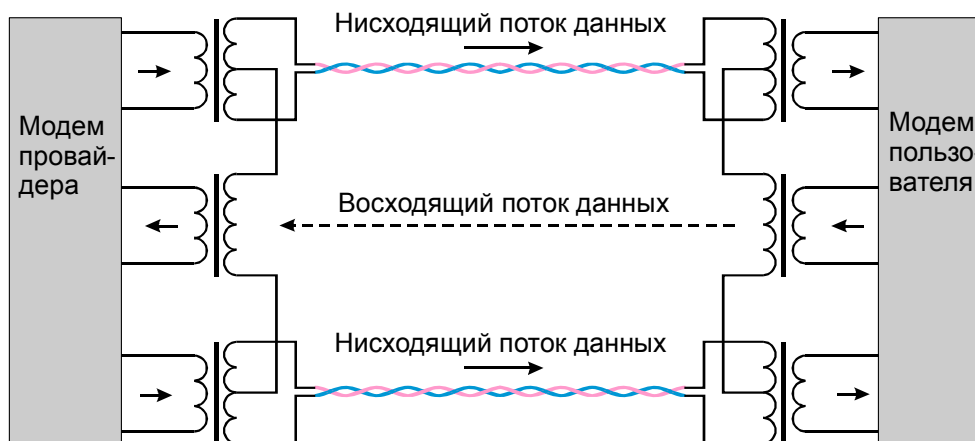


**Рис. 2.43. Размещение блока трансформаторов 2 (Рис. 2.38, а) в вилке-тройнике**

В асимметричной системе передачи (ADSL), показанной на Рис. 2.44, суммарная скорость нисходящих потоков данных превышает скорость восходящего потока. Нисходящие потоки в явном виде передаются по витым парам. Для передачи данных в обратном направлении используется фантомная цепь.

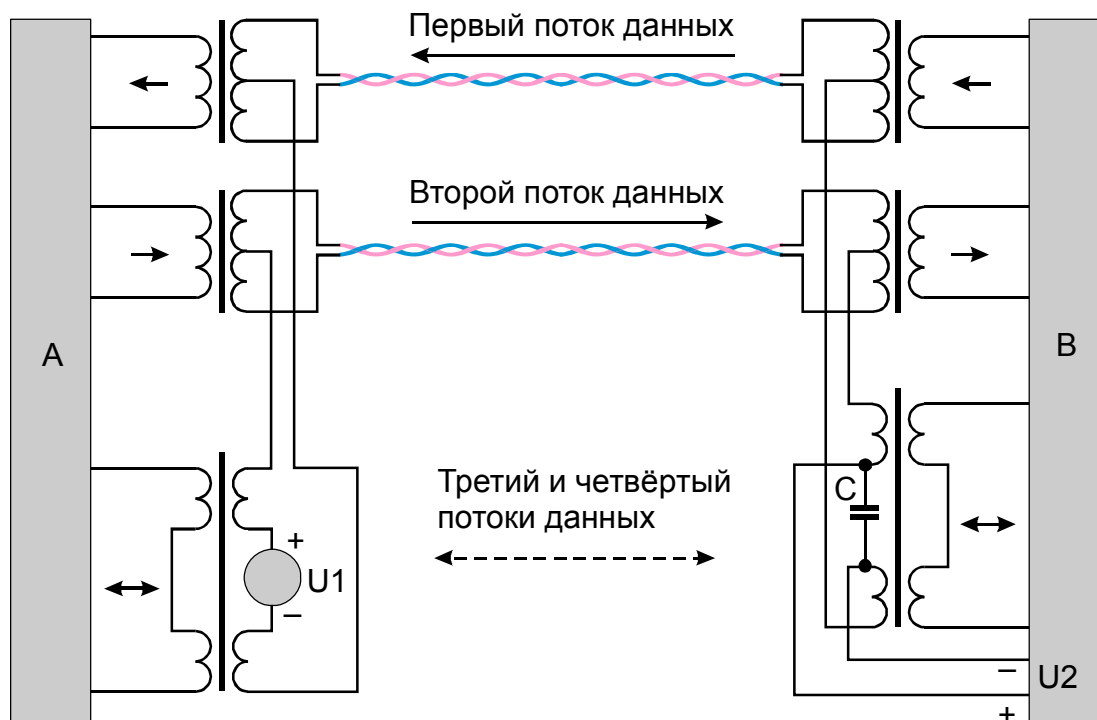
#### 2.7.4. Развитие идеи совмещения сигналов

Скорость передачи восходящего потока данных может несколько превышать скорость каждого из нисходящих потоков. Напомним, что в фантомной цепи оба провода витой пары рассматриваются как единый провод с удвоенным сечением жилы, а чем толще жила, тем больше дальность связи или допустимая скорость передачи. Кроме того, погонная ёмкость между проводами (витыми парами) фантомной линии меньшая, чем погонная ёмкость между проводами витой пары, что также благоприятно отражается на характеристиках линии.



**Рис. 2.44. Применение фантомной цепи в асимметричной системе передачи данных (ADSL)**

В схеме, показанной на Рис. 2.45, имеются два телекоммуникационных устройства А и В, между которыми передаются четыре потока данных. Первый и второй потоки данных в явном виде передаются по двум витым парам проводов в соответствующих направлениях. Третий и четвёртый потоки в дуплексном режиме передаются по фантомной цепи. Постоянное напряжение  $U_1$  от источника питания через эту же фантомную цепь поступает в устройство В в виде напряжения  $U_2$ .



**Рис. 2.45. Применение фантомной цепи для передачи данных и напряжения питания удалённого устройства В**

В заключение отметим, что задача совмещения передачи обычных цифровых и “голосовых” данных решается разными способами. В глобальных сетях широко применяется технология VoIP (Voice-Over-Internet-Protocol). Согласно этой технологии звуковые сигналы от микрофона телефонного аппарата усиливаются, преобразуются из аналоговой формы в цифровую, сжимаются, помещаются в информационные пакеты данных, маршрутизируются и передаются абоненту, где происходит их обратное преобразование. Применительно к локальным сетям такое решение сложно и дорого; кроме того, качество передачи звукового сигнала зависит от трафика и при больших нагрузках на сеть может снижаться.

Кабельная инфраструктура локальной сети позволяет наряду с данными одновременно передавать по ней сигналы офисной АТС (или иные) без применения преобразований, подоб-

ных принятым в технологии VoIP. Задача построения дополнительных каналов связи решается на аппаратном уровне — либо с применением частотного разделения сигналов (Рис. 2.37), либо с использованием фантомных цепей, что более экономично.

Фантомные цепи обеспечивают повышенную дальность связи (или повышенную скорость передачи) благодаря удвоенному сечению жилы медного проводника (витая пара рассматривается как единый провод) и меньшей погонной ёмкости между витыми парами, чем между проводами витой пары. При этом несколько снижается защищённость фантомного канала связи от действия внешних помех (ёмкостных, индуктивных, электромагнитных) и низкочастотных магнитных полей. При построении фантомного канала следует уделять внимание симметрии входящих в него цепей и элементов. Так, витые пары канала должны иметь одинаковую длину, применяемые трансформаторы должны обладать близкими параметрами и т. п.

## 2.8. Вопросы и задачи к разделу 2.7

- 2.8.1. Фрагмент локальной сети Ethernet 10 Base T, показанный на Рис. 2.46, позволяет использовать одну и ту же проводную инфраструктуру как для обмена данными между компьютерами и концентратором, так и для связи офисной АТС с телефонными аппаратами. Чем ограничивается длина линий связи в этом фрагменте (ограничениями, связанными с правилами проектирования сети Ethernet 10 Base T, пренебрегаем)?

Литература: [6]

Решение задачи приведено в разделе 3 на с. 171

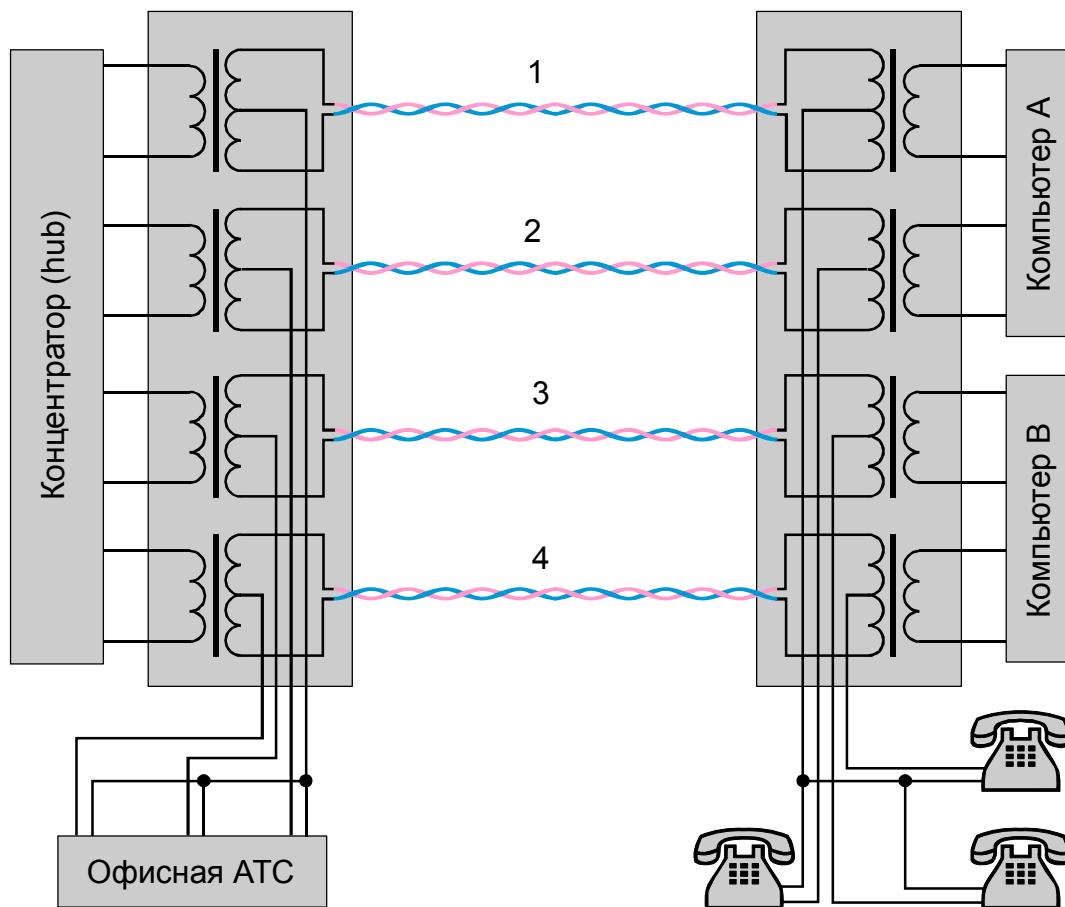


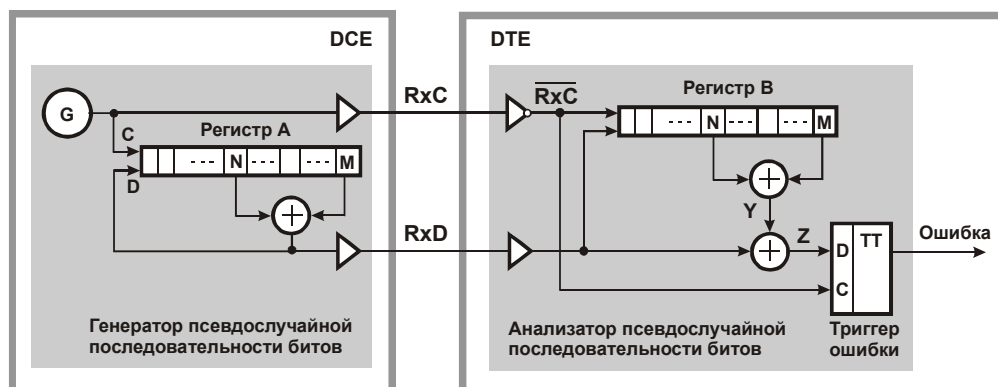
Рис. 2.46. Использование фантомных цепей для совмещения передачи сигналов концентратора локальной сети Ethernet 10 Base T с сигналами офисной АТС

## 2.9. BER-тестеры

Литература к разделу 2.9: [1], гл. 8; [4]

### 2.9.1. Введение в раздел 2.9

Как показано на Рис. 2.47, для тестирования цепей RxC и RxD использованы генератор и анализатор псевдослучайных последовательностей битов. По существу, генератор и анализатор представляют собой скремблер и дескремблер [4], причем скремблируется сигнал “Постоянный нуль”, т. е. последовательность нулевых битов. В отсутствие ошибок передачи сигналов RxC и RxD сигнал на выходе триггера ТТ также должен быть нулевым.



**Рис. 2.47. 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 Рис. 2.48. 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 (наблюдается “отрицательная задержка”!). 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.

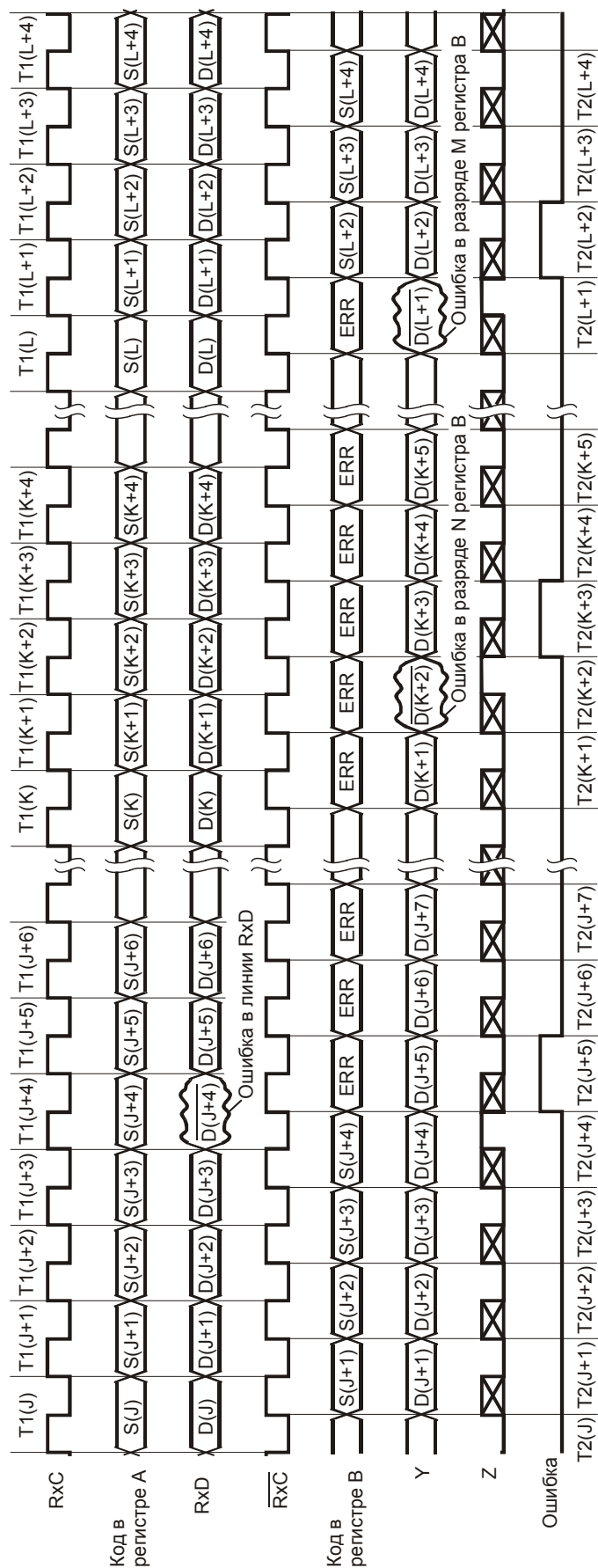


Рис. 2.48. Временные диаграммы передачи и приема тестовых битовых последовательностей

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.

## 2.9.2. Варианты включения BER-тестера в телекоммуникационную систему

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 системе, показанной на Рис. 2.49, 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.

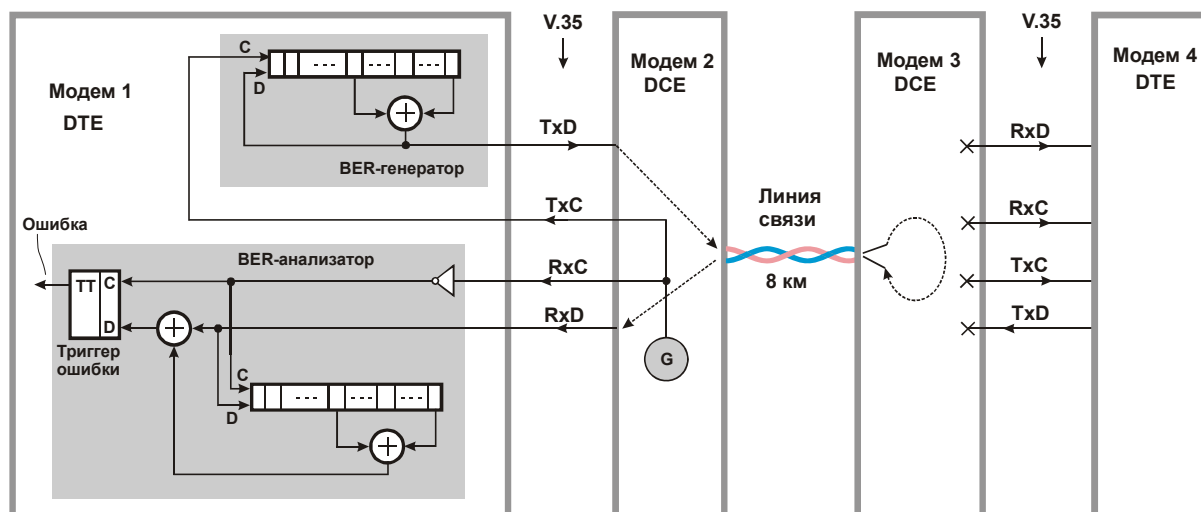


Рис. 2.49. Включение BER-тестера в телекоммуникационную систему

В первом модеме включен режим BER-тестера. Это означает, что по положительным фронтам синхросигнала TxC в линию передаваемых данных с выхода BER-генератора выдаются псевдослучайные биты данных TxD. Параллельно с этим BER-анализатор этого же модема проверяет правильность последовательности битов RxD, сопровождаемых синхросигналом TxC.

Все процессы, протекающие в системе, синхронизируются от генератора G, размещенного во втором модеме.

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

В отсутствие ошибок данные проходят от первого модема к третьему и обратно без искажений. Сигнал на выходе триггера ошибки постоянно равен нулю. Любое нарушение правильной последовательности данных или (и) существенное искажение синхросигнала на входах BER-анализатора регистрируется триггером ошибки в виде одного или нескольких импульсов Ошибка. Отметим, что 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. 2.50, 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.

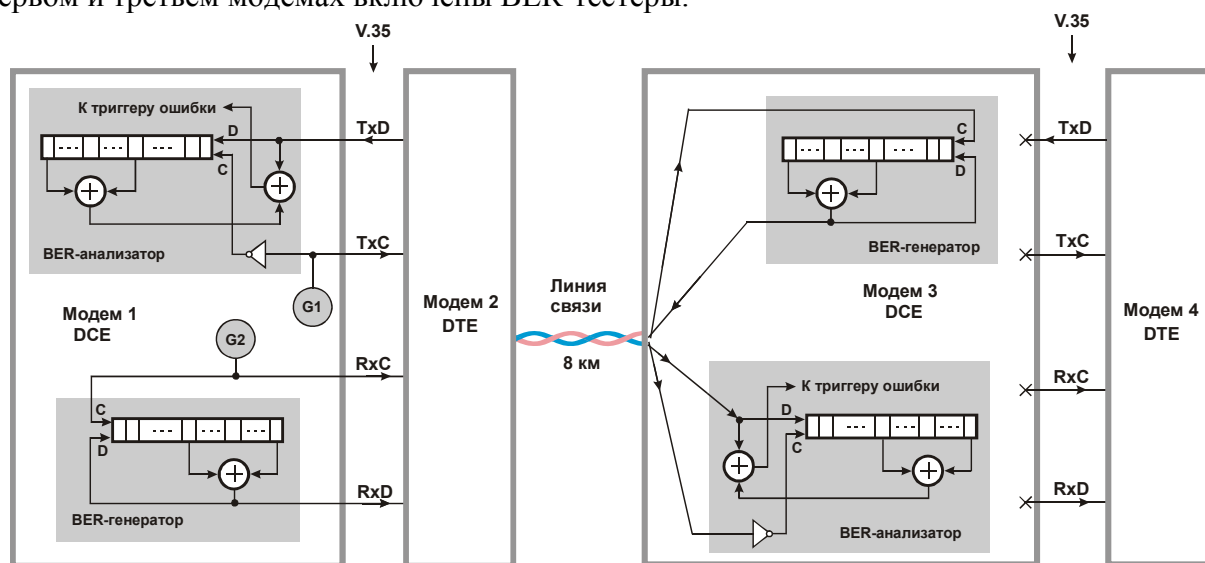


Рис. 2.50. Включение двух BER-тестеров в телекоммуникационную систему

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 BER-генератор третьего модема выдает в линию псевдослучайную тестовую последовательность битов со скоростью, задаваемой генератором G1. Действительно, данные в смеси с синхросигналом передаются по “верхнему” каналу справа налево, а генератор G1 находится слева. Каким образом синхросигнал от генератора G1 может распространяться по линии против течения потока данных? — Ответ на этот вопрос рассмотрен в [1]. Краткое описание идеи приведено на с. 173.

## 2.10. Вопросы и задачи к разделу 2.9

2.10.1. На Рис. 2.51 показана система контроля (BER-тестер) передачи данных и синхросигнала между устройствами DTE и DCE.

В этой системе наблюдается “отрицательная задержка” распространения сигнала, в том смысле, что информация в регистре В приёмника формируется на половину такта раньше, чем та же информация в регистре А передатчика.

Объясните этот факт.

Начертите схему, в которой генератор G перемещён в устройство DCE.

Наблюдается ли теперь “отрицательная задержка”?

Литература: [1], гл. 8

Решение задачи приведено в разделе 3 на с. 172

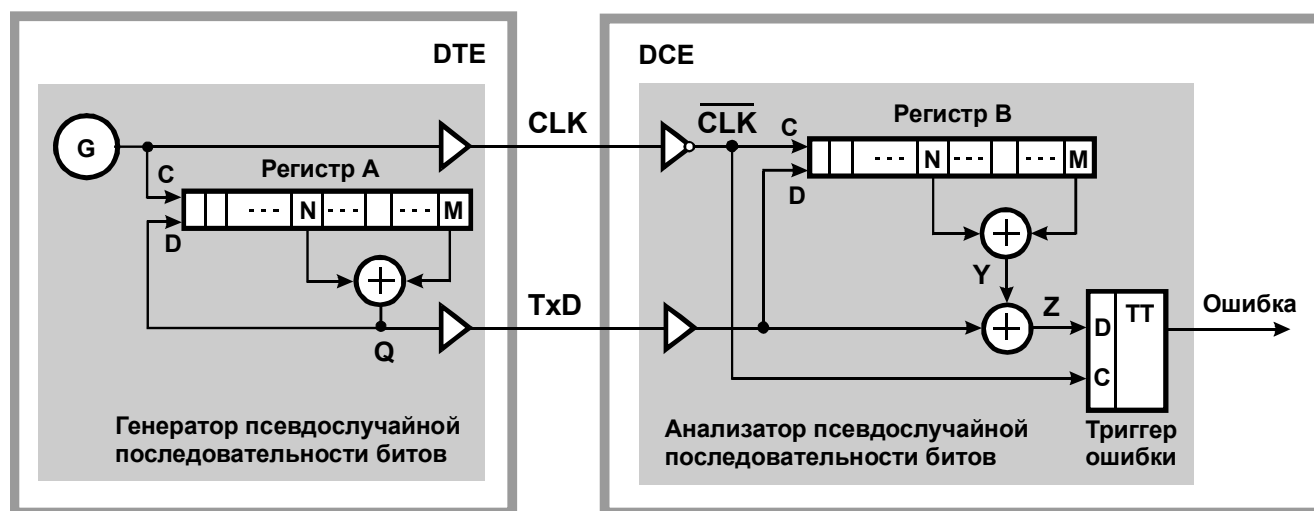


Рис. 2.51. Система контроля передачи данных и синхросигнала между устройствами типа DTE и DCE

2.10.2. На Рис. 2.52 показано включение BER-тестера в телекоммуникационную систему.

- 1) Каковы временные соотношения между сигналами RxC и RxD?
- 2) Зависят ли они от длины линии связи?
- 3) Зависит ли работа BER-тестера от длины линии связи?
- 4) Как учитывается в работе BER-тестера задержка тестового сигнала, вызванная его двойным пробегом по линии связи?

Литература: [1], гл. 8

Ответы приведены в разделе 3 на с. 173

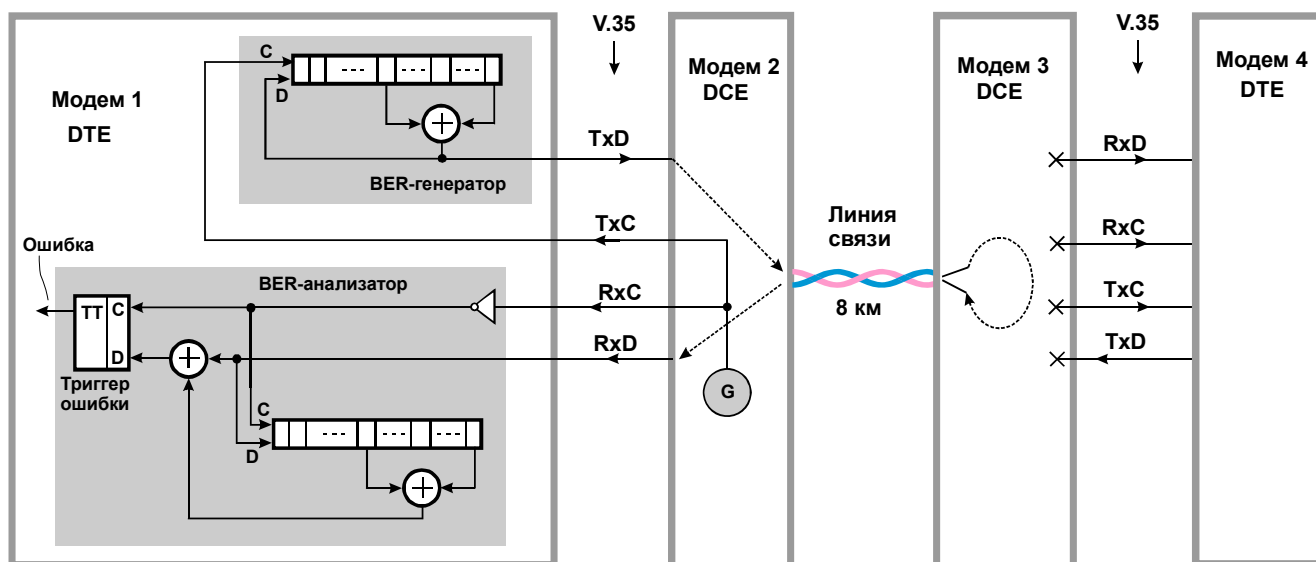


Рис. 2.52. Включение BER-тестера в телекоммуникационную систему

2.10.3. На Рис. 2.53 показано встречное включение двух BER-тестеров в телекоммуникационную систему.

Каким образом сигнал TxC с генератора G1 проходит на вход синхронизации BER-генератора модема 3 (против течения потока тестовых данных)?

Литература: [1], гл. 8

Ответ приведен в разделе 3 на с. 173

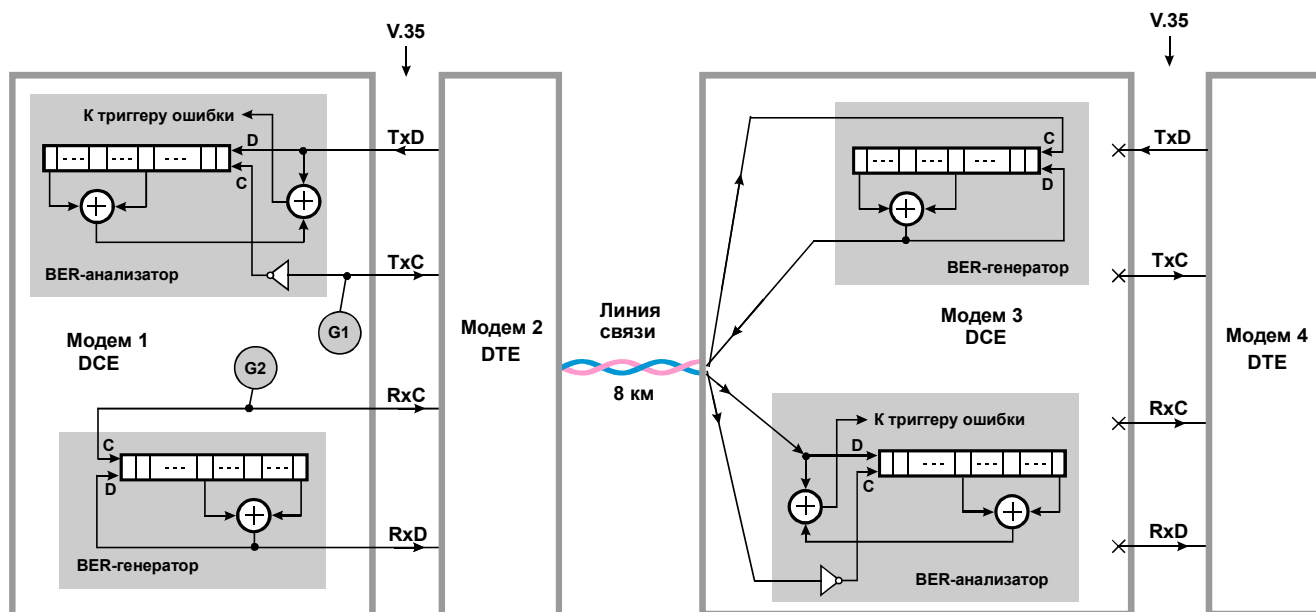


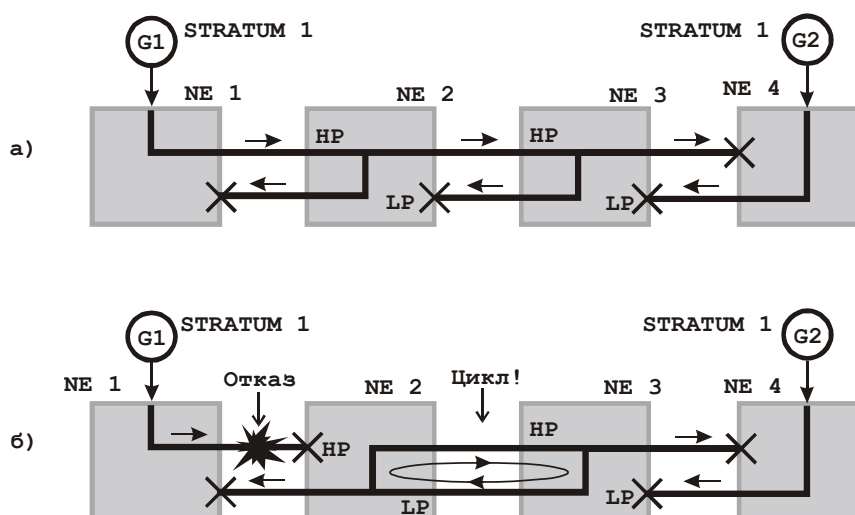
Рис. 2.53. Встречное включение двух BER-тестеров в телекоммуникационную систему

## 2.11. Предотвращение заикливания синхросигналов

Литература к разделу 2.11: [1], гл.6

### 2.11.1. Введение в раздел 2.11

При построении глобальной сети разработчики стремятся к тому, чтобы каждый её узел синхронизировался сигналом наиболее высокого уровня точности и стабильности, например, сигналом уровня STRATUM 1. Но 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. 2.54.



**Рис. 2.54. Схема, поясняющая процесс возникновения ошибок синхронизации — циклов, лишенных опорного источника сигналов: а — исходное (работоспособное) состояние цепи из сетевых устройств NE (Network Equipment); 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. 2.54, а) для синхронизации выходных сигналов используются импульсы, выделенные с высокоприоритетных входов HP. Синхросигналы 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. 2.54 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. 2.54, 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 уст-

ройства 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 НР 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.

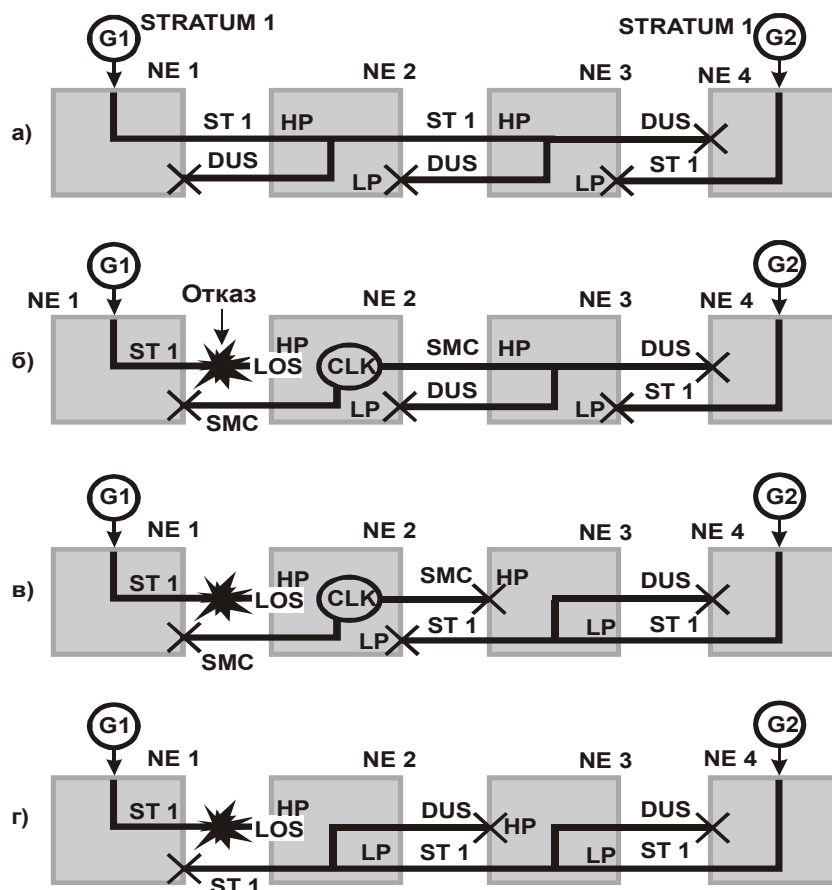
Далее рассмотрены метод и схема, исключающие формирование циклических трасс распространения синхросигналов при отказе канала связи.

### **2.11.2. Автоматическое предотвращение заикливания синхросигналов**

Мы только что рассмотрели сравнительно простую ситуацию, когда неправильное решение, приведшее к заикливанию синхросигнала (см. Рис. 2.54, 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. 2.55).

Исходные условия работы участка сети совпадают с приведенными в предыдущем примере (сравните Рис. 2.54, а и Рис. 2.55, 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 мнемонические коды, отражающие статус передаваемого синхросигнала.



**Рис. 2.55. 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, переносащий данный кадр от одного устройства NE<sub>i</sub> 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. 2.55, а) не допускает проведения в жизнь ошибочного решения, принятого в предыдущем примере (см. Рис. 2.54). 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. 2.55, 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 соседние узлы о случив-

шемся, 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. 2.55, 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. 2.55, 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<sub>2</sub>, 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<sub>2</sub>), 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 задачи общей синхронизации.

Помимо описанных статусных признаков могут передаваться сведения о траектории распространения синхросигнала, т. е. об адресах узлов сети, которые пройдены им на пути к приёмнику [21]. Это позволяет приёмнику, во-первых, просмотреть список адресов и убедиться, что он сам не числится в нём. В противном случае обнаруживается ошибка — заикливание синхросигнала. Во-вторых, приёмник подсчитывает число узлов, преодоленных синхросигналом. В некоторых стандартах предусмотрено условие, что это число не должно превышать 20. Если приёмник обнаруживает, что условие нарушено, то он не использует такой синхросигнал.

## 2.12. Вопросы и задачи к разделу 2.11

2.12.1. Фрагмент компьютерной сети, показанный на Рис. 2.56, в результате отказа линии связи между узлами NE2 и NE3 временно теряет работоспособность. Используя способ автоматического предотвращения заикливания синхросигналов, покажите, как и в течение скольких шагов протекает процесс адаптации сети к отказу.

Литература: [1], гл. 6

Решение задачи приведено в разделе 3 на с. 174

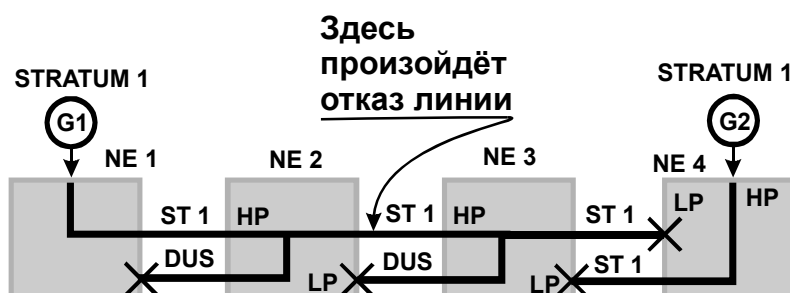


Рис. 2.56. Исходный фрагмент компьютерной сети (до возникновения отказа)

2.12.2. Фрагмент компьютерной сети, показанный на Рис. 2.57, в результате отказа генератора G1 синхросигналов временно теряет работоспособность. Используя способ автоматического предотвращения заикливания синхросигналов, покажите, как и в течение скольких шагов протекает процесс адаптации сети к отказу.

Литература: [1], гл. 6

Решение задачи приведено в разделе 3 на с. 174



Рис. 2.57. Исходный фрагмент компьютерной сети (до возникновения отказа)

2.12.3. Фрагмент компьютерной сети, показанный на Рис. 2.58, в результате отказа генератора G2 синхросигналов временно теряет работоспособность. Используя способ автоматического предотвращения заикливания синхросигналов, покажите, как и в течение скольких шагов протекает процесс адаптации сети к отказу.

Литература: [1], гл. 6

Решение задачи приведено в разделе 3 на с. 175

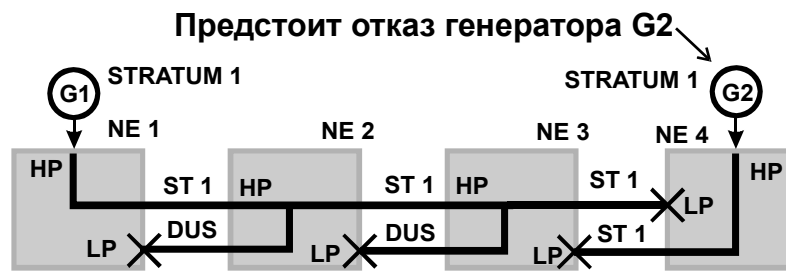


Рис. 2.58. Исходный фрагмент компьютерной сети (до возникновения отказа)

## 2.13. Скремблирование — дескремблирование данных

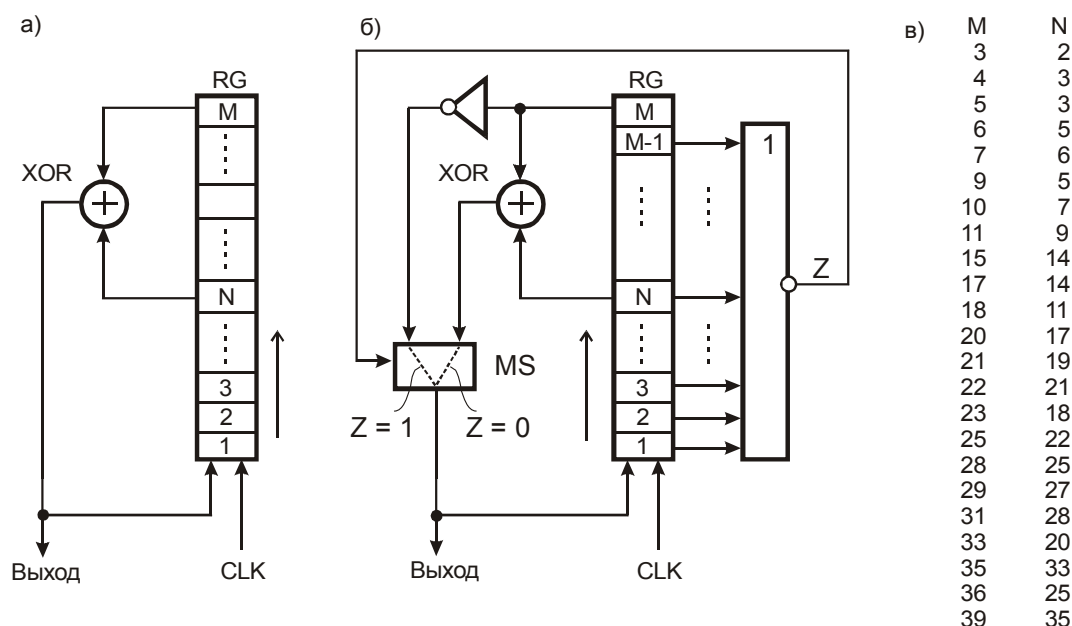
Литература к разделу 2.13: [1], гл. 8; [4, 10 — 17]

### 2.13.1. Введение в раздел 2.13

Скремблирование — это шифрация потока данных, в результате которой он выглядит как поток случайных битов. Последовательности битов в исходном массиве данных, как регулярные, так и нерегулярные, обратимо разрушаются, так что вероятности появления логической единицы и логического нуля в каждой последующей битовой позиции потока одинаковы и не зависят от предыстории. Применительно к телекоммуникационным системам скремблирование повышает надёжность синхронизации устройств, подключенных к противоположным сторонам линии связи, и уменьшает уровень помех, излучаемых на соседние линии многожильного кабеля. Есть и иная область применения скремблеров — защита передаваемой информации от несанкционированного доступа; однако эта область здесь не рассматривается. Ниже приведены схемы классических и модернизированных скремблеров и дескремблеров, описаны преимущества, связанные с их применением, рассмотрены меры защиты систем передачи скремблированных данных от злонамеренного пользователя.

### 2.13.2. Генераторы псевдослучайных последовательностей битов

Скремблеры и дескремблеры (шифраторы и дешифраторы особого класса) обычно построены на основе генераторов псевдослучайных последовательностей битов. Генераторы чаще всего выполняются с использованием  $M$ -разрядных сдвиговых регистров RG с цепями обратной связи (Рис. 2.59).



**Рис. 2.59.** Генераторы псевдослучайных битовых последовательностей, формирующие: а — последовательность длиной  $2^M - 1$  бит, б — последовательность длиной  $2^M$  бит; в — таблица для выбора промежуточной точки N подключения цепи обратной связи

Приведенные схемы различаются длинами периодов генерируемых последовательностей битов. В схеме, показанной на Рис. 2.59, а [10], регистр RG исходно установлен в некоторое ненулевое состояние (цепь начальной установки не показана). Под действием положительных фронтов синхросигнала CLK хранимый в регистре код непрерывно циркулирует в нём и одновременно видоизменяется благодаря преобразованию битов логическим элементом Иключающее ИЛИ (XOR). Генерируемая последовательность битов снимается с выхода этого элемента

или с выхода любого разряда регистра. Направление сдвига данных в регистре показано стрелкой. В полном цикле работы генератора в регистре однократно формируются все возможные  $M$ -разрядные коды, за исключением нулевого. Циклы следуют один за другим без пауз.

В общем случае при использовании  $M$ -разрядного регистра цепь обратной связи подключается к разрядам с номерами  $M$  и  $N$  ( $M > N$ ). Для того чтобы на выходе генератора (Рис. 2.59, *а*) формировалась псевдослучайная последовательность битов с периодом повторения, равным  $2^M - 1$ , следует выбирать точки подключения цепи обратной связи в соответствии с таблицей, приведенной на Рис. 2.59, *в*, которая описывает ряд генераторов различной разрядности.

Псевдослучайная последовательность битов с периодом повторения, равным  $2^M - 1$ , обладает следующими свойствами.

1. В полном цикле ( $2^M - 1$  тактов) число лог. 1, формируемых на выходе генератора, 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 лог. 0 и лог. 1 на выходе генератора 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  $\frac{1}{2}$ .

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

Усовершенствованная схема (Рис. 2.59, *б* [11, 12]) формирует псевдослучайную последовательность битов с периодом повторения, равным  $2^M$ . К этой схеме также применима таблица, приведенная на Рис. 2.59, *в*. В регистре  $RG$  в определенном порядке формируются все возможные коды, включая нулевой. Схема дополнительно содержит элемент ИЛИ — НЕ, инвертор и мультиплексор  $MS$ . Сигнал  $Z$  на выходе элемента ИЛИ — НЕ задаёт направление передачи данных через мультиплексор. При  $Z = 0$  на выход мультиплексора транслируется сигнал с выхода элемента Иключающее ИЛИ, а при  $Z = 1$  — сигнал с выхода инвертора.

До тех пор пока на входах элемента ИЛИ — НЕ присутствует хотя бы одна лог. 1, на его выходе сформирован сигнал  $Z = 0$ . В этом случае мультиплексор  $MS$  в каждом такте передаёт в освободившийся (нижний) разряд сдвигового регистра бит с выхода элемента Иключающее ИЛИ так же как и в схеме, показанной на Рис. 2.59, *а*.

В некотором такте  $i$  в регистре фиксируется код, содержащий единственную лог. 1, размещённую в разряде  $M - 1$ . Так как в разрядах  $M$  и  $N$  присутствуют лог. 0, то на выходе элемента Иключающее ИЛИ сформирован сигнал лог. 0, который к началу такта  $i + 1$  поступает на вход регистра. В начале такта  $i + 1$  лог. 1 перемещается из разряда  $M - 1$  в разряд  $M$ , на входах элемента ИЛИ — НЕ формируется нулевой код. Сигнал  $Z = 1$  переводит мультиплексор  $MS$  в состояние, при котором на вход нижнего разряда сдвигового регистра поступает бит с выхода инвертора. В данном случае этот бит равен 0, поэтому в такте  $i + 2$  в регистре фиксируется нулевой код.

К началу такта  $i + 3$  на вход сдвигового регистра с выхода инвертора поступает лог. 1, поэтому по положительному фронту синхросигнала  $CLK$  в регистре фиксируется код, содержащий лог. 0 во всех разрядах, кроме первого. Сигнал  $Z$  вновь принимает нулевое значение, мультиплексор переключается в состояние передачи сигнала с выхода элемента Иключающее ИЛИ и т. д. Таким образом, регистр проходит через все состояния, включая нулевое. Возможны и иные варианты построения генераторов с числом состояний, равным  $2^M$  [11, 12].

Наиболее распространены два вида систем “скремблер — дескремблер”: с неизолированными и изолированными (от линии связи) генераторами псевдослучайных последовательностей битов (Рис. 2.60, Рис. 2.61) [13, 1].

### 2.13.3. Система с неизолированными генераторами

В системе, показанной на Рис. 2.60, скремблер и дескремблер содержат фрагменты рассмотренного ранее генератора (см. Рис. 2.59, а) псевдослучайных последовательностей битов. В скремблере цепь обратной связи генератора на основе сдвигового регистра RG1 дополнительно содержит элемент Иключающее ИЛИ XOR2. В дескремблере применен аналогичный генератор на основе сдвигового регистра RG2 с разомкнутой цепью обратной связи.

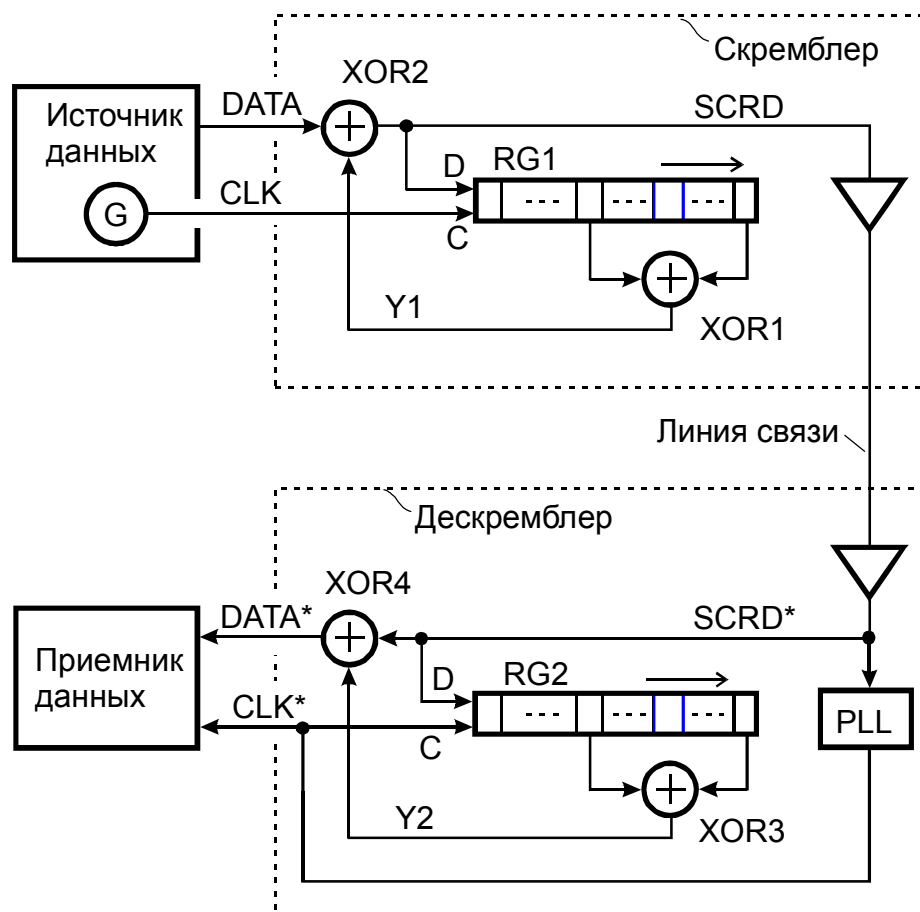


Рис. 2.60. Система “скремблер — дескремблер” с неизолированными генераторами псевдослучайных последовательностей битов

Все процессы, протекающие в системе, синхронизируются от тактового генератора G, размещенного в источнике данных (возможно также его размещение в скремблере). Тактовый генератор формирует сигнал CLK — непрерывную последовательность тактовых импульсов со скважностью, равной двум. В каждом такте по положительному фронту сигнала CLK на вход скремблера подается “новый” бит передаваемых данных DATA, а код в его сдвиговом регистре RG1 продвигается на один разряд вправо, причём в этот же момент в освободившийся разряд заносится “старый” бит данных, просуммированный по модулю два со “старым” битом Y с выхода элемента XOR1.

Строго говоря, на границах между битовыми интервалами на выходе элемента XOR2 могут формироваться короткие ложные импульсы в результате неодновременного формирования “новых” сигналов на его входах (сигнал Y1 приходит чуть позже сигнала DATA). Для устранения ложных импульсов можно ввести в цепь сигнала SCRД D-триггер, синхронизируемый отрицательным фронтом сигнала CLK (триггер на рисунке не показан). Короткими ложными импульсами пока пренебрегаем для упрощения изложения основных идей построения систем “скремблер — дескремблер”.

Если источник данных посылает в скремблер длинную последовательность сигналов лог. 0 (DATA = 0), то элемент XOR2 можно рассматривать как повторитель сигнала Y1. Тогда регистр RG1 фактически оказывается замкнутым в кольцо и генерирует точно такую же псевдослучай-

ную последовательность битов, как и в рассмотренной ранее схеме генератора, приведенной на Рис. 2.59, *а*. Отметим, что в этой ситуации при неблагоприятном стечении обстоятельств есть опасность потери работоспособности скремблера, если в регистре RG1 к началу передачи последовательности сигналов лог. 0 зафиксирован нулевой код. (Об этом — позже.)

Если от источника данных поступает произвольная битовая последовательность, то она взаимодействует с последовательностью битов с выхода элемента XOR1. В результате формируется новая (скремблированная) последовательность битов данных SCRD, по структуре близкая случайной. Эта последовательность, в свою очередь, продвигается по регистру RG1, формирует поток битов Y1 на выходе элемента XOR1 и т. д.

Скремблированная последовательность битов SCRD проходит через передающий усилитель и по линии связи поступает в дескремблер, где проходит через приёмный усилитель. Линия связи может быть выполнена, например, в виде витой пары проводов многожильного кабеля городской телефонной сети. С помощью генератора PLL (Phase Locked Loop) с фазовой автоподстройкой частоты из входного сигнала SCRD\* выделяется тактовый сигнал CLK\*, который передаётся на синхронизирующие входы регистра RG2 и приёмника данных.

Генератор PLL с фазовой автоподстройкой частоты может быть построен по одной из известных схем (см., например, [14]). Он предназначен для формирования высокостабильного синхросигнала CLK\* на основе непрерывного слежения за входным сигналом SCRD\*. В данном случае отрицательный фронт сигнала CLK\* привязан к моментам изменения сигнала SCRD\* ( $0 \rightarrow 1$  или  $1 \rightarrow 0$ ), так что положительный фронт сигнала CLK\* формируется в середине битового интервала сигнала SCRD\*, что соответствует его установившемуся значению. Сдвиг данных в регистре RG2 и приём очередного бита SCRD\* в его освободившийся разряд происходят по положительному фронту сигнала CLK\*. Дескремблированные данные DATA\* поступают в приемник данных и фиксируются в нем по положительным фронтам сигнала CLK\*. Благодаря достаточной инерционности генератора PLL сигнал CLK\* практически нечувствителен к “дрожанию фазы” сигнала SCRD\* и иным его кратковременным искажениям, вызванным помехами в линии связи.

Потоки данных DATA и DATA\* совпадают с точностью до задержки передачи. Действительно, в установившемся режиме в сдвиговых регистрах RG1 и RG2 присутствуют одинаковые коды, так как на входы D этих регистров поданы одни и те же данные SCRD = SCRD\* (с учётом задержки передачи), а тактовая частота одна и та же. Поэтому  $Y2 = Y1$  и с учетом этого

$$DATA^* = SCRD^* \oplus Y2 = SCRD \oplus Y2 = (DATA \oplus Y1) \oplus Y2 = DATA \oplus Y1 \oplus Y1 = DATA \oplus 0 = DATA.$$

Рассмотренный способ скремблирования — дескремблирования данных не требует применения какой-либо специальной процедуры начальной кодовой синхронизации, после которой коды в обоих регистрах становятся одинаковыми и, следовательно, начинает выполняться условие  $Y2 = Y1$ . Синхронизация достигается автоматически после заполнения регистров одинаковыми данными. Это, пожалуй, единственное преимущество данной схемы перед классической схемой с изолированными генераторами (Рис. 2.61).

К сожалению, есть и существенные недостатки. О первом из них уже вскользь упоминалось — это плохая устойчивость по отношению к некоторым неблагоприятным кодовым ситуациям, которые могут возникнуть как при нормальной работе системы, так и в результате злого умысла пользователя. (Этот недостаток и меры его устранения будут рассмотрены далее.)

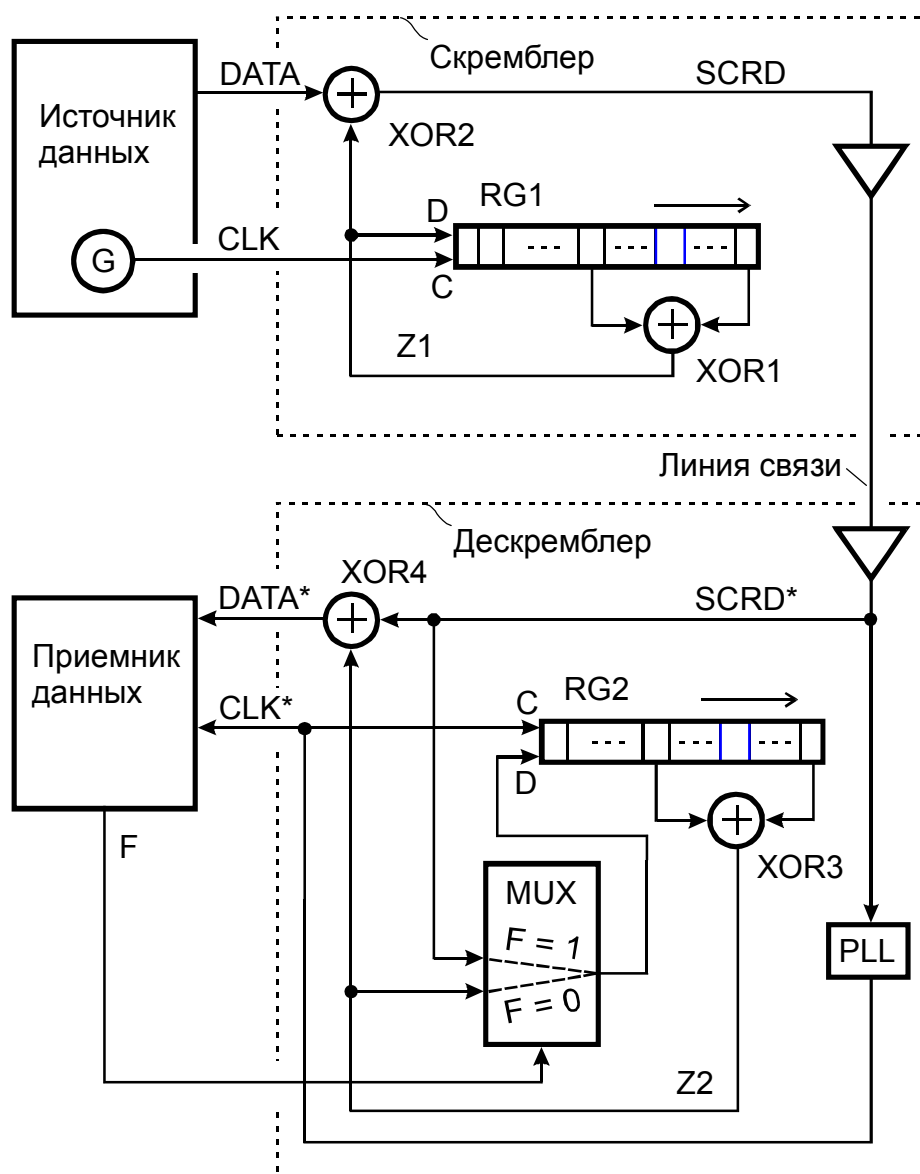
Второй недостаток состоит в размножении ошибок. При появлении одиночной ошибки в линии связи идентичность содержимого регистров RG1 и RG2 временно нарушается, но затем автоматически восстанавливается, как только правильные данные вновь заполняют регистр RG2. Однако в процессе продвижения ошибочного бита по сдвиговому регистру RG2, а именно, в периоды его попадания сначала на один, а затем на другой вход элемента XOR3 сигнал Y2 дважды принимает неправильное значение. Это приводит к размножению одиночной ошибки — она впервые появляется в сигнале DATA\* в момент поступления из линии и затем возникает еще два раза при последующем двукратном искажении сигнала Y2.

### 2.13.4. Система с изолированными генераторами

В системе, показанной на Рис. 2.61, применены изолированные от линии связи генераторы псевдослучайных битовых последовательностей. Преимущества этой системы перед предыдущей заключаются в том, что, во-первых, ошибки, поступающие из линии, не размножаются. Во-вторых, как будет показано, такая система более устойчива по отношению к неблагоприятным последовательностям битов, которые формируются либо в силу случайных стечений обстоятельств либо в результате преднамеренных действий пользователя.

Недостаток этой системы — сложность установления кодовой синхронизации. Отметим, что в конце статьи предложено решение, лишенное этого недостатка — оно предусматривает автоматическое установление и поддержание синхронной работы изолированных генераторов псевдослучайных битовых последовательностей.

Начальная кодовая синхронизация системы (Рис. 2.61) осуществляется с использованием аппаратных средств дескремблера и программных средств источника и приёмника данных. К аппаратным средствам относятся мультиплексор MUX и программно-управляемый выход приёмника данных, на котором формируется управляющий сигнал  $F$ . При нормальной работе системы приемник данных постоянно поддерживает на выходе сигнал  $F = 0$ . На выход мультиплексора транслируется сигнал  $Z2$  с выхода элемента Иключающее ИЛИ XOR3, генератор псевдослучайной битовой последовательности на основе регистра RG2 изолирован от внешних воздействий со стороны линии связи.



**Рис. 2.61. Система “скремблер — дескремблер” с изолированными генераторами псевдослучайных последовательностей битов**

Предположим, что в исходном состоянии дескремблер не синхронизирован со скремблером. Такая ситуация может возникнуть, например, после включения напряжения питания аппаратуры приемной стороны, после ошибки в работе генератора PLL дескремблера из-за воздействия помех на линию связи или по иным причинам. В отсутствие кодовой синхронизации между скремблером и дескремблером содержимое регистров RG1 и RG2 не совпадает, поток принимаемых данных DATA\* ошибочен и не совпадает с потоком передаваемых данных DATA.

При обнаружении устойчивого “хаоса” в данных DATA\* (когда в потоке нет обусловленного протоколом обмена разделения на информационные кадры и т. п.) приемник формирует сигнал  $F = 1$ . Вследствие этого мультиплексор начинает транслировать на вход D регистра RG2 сигнал скремблированных данных SCRД\*, как в ранее рассмотренной системе (см. Рис. 2.60).

Протокол обмена предусматривает пересылку данных в виде последовательности кадров. Группы обычных кадров перемежаются со служебными кадрами. Например, после группы из 1000 обычных кадров следует один служебный. Он, в частности, содержит синхронизирующую последовательность из некоторого числа (например, 256) нулевых битов. При выдаче этих битов ( $DATA = 0$ ) в скремблер элемент XOR2 выполняет функцию повторителя сигнала Z1 с выхода элемента XOR1. 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 DATA и порождается только генератором

скремблера.

Эта последовательность загружается в регистр RG2 и проходит через него, так как  $F = 1$ . После того как содержимое регистров RG1 и RG2 оказывается одинаковым, сигнал Z2 начинает повторять сигнал Z1. Кодовая синхронизация достигнута. На вход приемника данных подается непрерывная последовательность лог. 0, так как  $DATA^* = DATA = 0$ . После уверенного обнаружения достаточно длинной (например, содержащей 220 бит) последовательности лог. 0 приемник данных формирует сигнал  $F = 0$  и тем самым возвращает генератор псевдослучайной последовательности битов дескремблера в режим изолированной работы. Теперь кодовая синхронизация не только достигнута, но и “сохранена” благодаря логической изоляции регистра RG2 от линии связи. После окончания передачи служебного (синхронизирующего) кадра источник данных приступает к передаче группы из 1000 обычных кадров согласно принятому в системе протоколу обмена.

Таким образом, в рассмотренной системе для поддержания синхронной работы сдвиговых регистров скремблера и дескремблера (в случае нарушения синхронизации или при первоначальном включении приёмной части системы) необходимо периодически прерывать передачу полезных данных и передавать по линии связи служебные информационные кадры, содержащие достаточно длинные цепочки синхронизирующих битов ( $DATA \equiv 0$ ). В результате уменьшается эффективная скорость передачи данных по линии, усложняется протокол обмена. Кроме того, с увеличением интервалов между служебными кадрами (это способствует более эффективной передаче пользовательских данных) увеличивается время ожидания этих кадров дескремблером в случае потери кодовой синхронизации. В течение времени ожидания передача полезных данных невозможна.

Прежде чем предложить решение, лишённое этих недостатков, рассмотрим преимущества, которые достигаются благодаря применению скремблирования данных, а также обсудим вопросы устойчивости скремблеров по отношению к нежелательным последовательностям битов.

### 2.13.5. Первое преимущество скремблирования — повышение устойчивости синхронизации

При передаче данных по линии связи применяют различные способы кодирования. В частности, широкое распространённое получило NRZ-кодирование и его модификации [1]. Для передачи нулевых и единичных битов выделяются одинаковые интервалы времени — “битовые интервалы”. В каждом битовом интервале в зависимости от значения передаваемого бита (лог. 1 или лог. 0) между проводами медной витой пары (двухпроводной линии) присутствует положительное или отрицательное напряжение. Применительно к оптоволоконным линиям, в каждом битовом интервале по оптическому волокну передаётся или не передаётся световой поток.

Такое кодирование неприменимо в тех случаях, по линии могут передаваться очень длинные цепочки из одинаковых битов. Тогда состояние линии будет оставаться неизменным на протяжении длительного интервала времени и синхронизация приёмника с передатчиком может нарушиться. Действительно, приемник надёжно восстанавливают синхросигнал только тогда, когда паузы между изменениями сигнала в линии не слишком велики. Изменение сигнала в линии после незначительной паузы позволяет всякий раз корректировать “ход часов” приемника. С увеличением паузы надёжность “службы времени” падает. Например, после передачи серии из 10 тыс. нулей приемник, вероятнее всего, не сможет с уверенностью определить, находится ли последующая единица на позиции 9999, 10000 или 10001. То же относится и к передаче длинных цепочек лог. 1. Другими словами, при передаче достаточно большой последовательности нулей или единиц приемник теряет синхронизацию с передатчиком.

На практике данные группируют в кадры постоянной или переменной длины. Каждый кадр содержит некоторую служебную информацию, например флаговый код начала кадра, причём эта информация заведомо не является последовательностью одинаковых битов. Это облегчает поддержание синхронизации, так как возможная однородность пользовательских данных пе-

риодически нарушается заведомо неоднородными служебными данными. Тем не менее, для повышения надёжности синхронизации желательно исключить из потока пользовательских данных длинные последовательности нулевых или единичных битов. Это и делается с помощью скремблирования — тогда цепочки нулевых или единичных битов (и не только они) преобразуются в псевдослучайные битовые последовательности, в которых вероятность изменения уровня сигнала в каждом последующем битовом интервале (по отношению к текущему уровню) равна  $\frac{1}{2}$ .

В подтверждение сказанного об устойчивости синхронизации рассмотрим схему выделения синхросигнала и данных из сигнала в линии (Рис. 2.62, а) [17, 1].

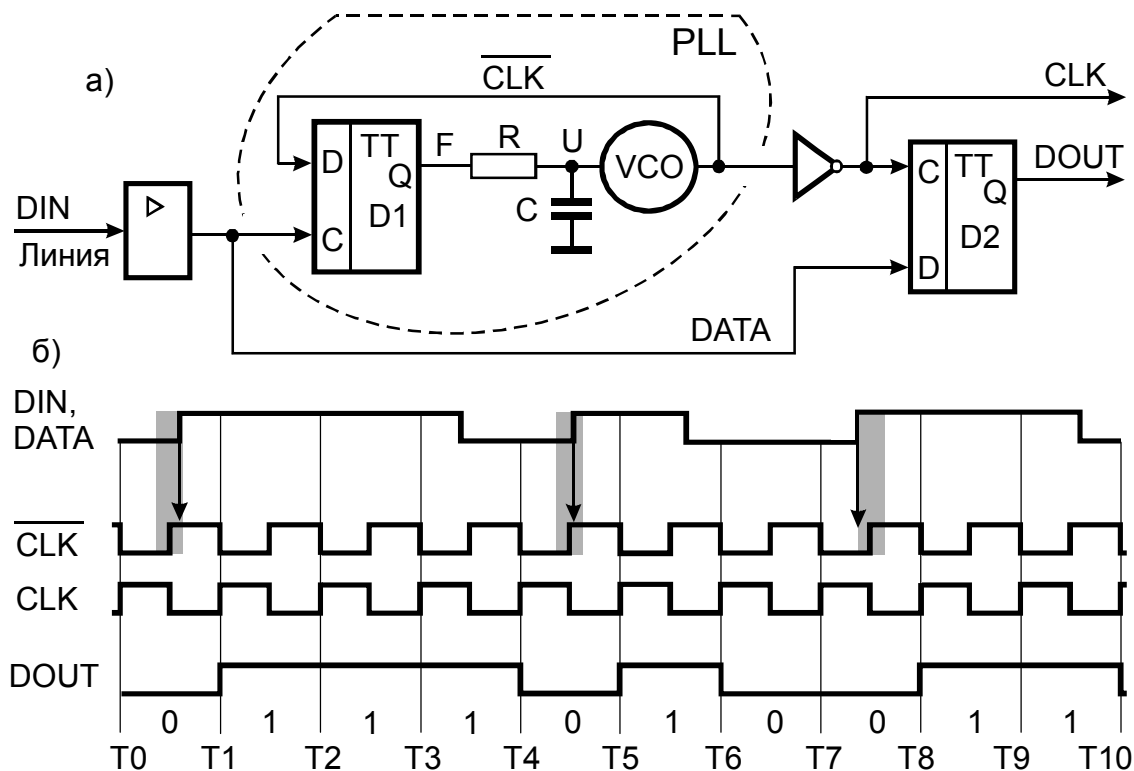


Рис. 2.62. Выделение синхросигнала CLK и данных DOUT из входного сигнала DIN, представленного кодом NRZ: а — схема; б — временная диаграмма

Схема содержит входной усилитель, генератор PLL с фазовой автоподстройкой частоты, инвертор и триггер D2. Генератор PLL включает в себя триггер D1, фильтр низких частот (RC-цепь) и генератор VCO (Voltage Controlled Oscillator), управляемый напряжением U. Триггеры D1 и D2 принимают данные с входов D по положительному фронту сигнала на входе C. Генератор VCO формирует выходной синхросигнал CLK (Рис. 2.62, б). Частота этого сигнала в значительных пределах может изменяться в зависимости от напряжения U на его управляющем входе. С повышением этого напряжения частота уменьшается, и наоборот.

Данные DATA представлены кодом NRZ. Границы битовых интервалов примерно соответствуют моментам формирования отрицательных фронтов сигнала CLK, а центры этих интервалов — моментам T0 — T10.

Обратите внимание на нестандартный режим работы триггера D1. Обычно на вход C синхронизации D-триггера подается синхроимпульс, а на вход D — данные. При этом, согласно техническим условиям на триггер, к моменту формирования положительного фронта синхроимпульса сигнал на входе D данных должен заблаговременно принять установившееся значение в течение некоторого “времени предустановки” и сохранять это значение в течение “времени удержания”, которое отсчитывается от того же фронта. Здесь же реализован некий противоположный режим. В качестве данных на D-вход триггера поступает синхросигнал, а в качестве синхросигнала — данные. При этом в идеальном случае положительный фронт сигнала на входе C должен совпадать с положительным фронтом сигнала на входе D.

Как показано на Рис. 2.62, *б* затемнёнными прямоугольниками, существуют области неопределённости моментов формирования положительных фронтов сигнала данных. Неопределённость обусловлена так называемым *джиттером* (дрожанием фронтов [1]). Джиттеру в равной мере подвержены и отрицательные фронты входного сигнала, но они нас в данном случае не интересуют.

Предположим, что генератор VCO уже вошёл в синхронизацию с сигналом, поступающим из линии. В этом случае положительные фронты сигнала DATA с равной вероятностью совмещены во времени с нулевыми или единичными уровнями инвертированного сигнала CLK. На рисунке вертикальные стрелки показывают, что два положительных фронта сигнала DATA совмещены с единичным уровнем инвертированного сигнала CLK, а третий — с нулевым уровнем. При равновероятном приёме в триггер D1 лог. 0 и лог. 1 сигнал F1 на его выходе постоянно изменяется, но в среднем пребывает в состоянии лог 0 столько же времени, сколько и в состоянии лог. 1. При этом напряжение U на выходе интегрирующей RC-цепи примерно равно половине напряжения, соответствующего уровню лог. 1 на выходе триггера D1.

Предположим теперь, что вертикальные стрелки в большинстве своем указывают на нулевые состояния синхросигнала с выхода генератора VCO. Это означает, что следует слегка увеличить частоту синхросигнала, что приведет к его незначительному фазовому опережению, т. е. к некоторому “сжатию влево” его временной диаграммы. Это и произойдёт благодаря тому, что в данной ситуации сигнал управления F будет преимущественно нулевым, напряжение U снизится, частота сигнала CLK незначительно увеличится.

В противоположной ситуации временная диаграмма синхросигнала исходно чуть смещена влево относительно показанной на рисунке. Тогда стрелки должны преимущественно 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, напряжение U повышается, частота незначительно снижается.

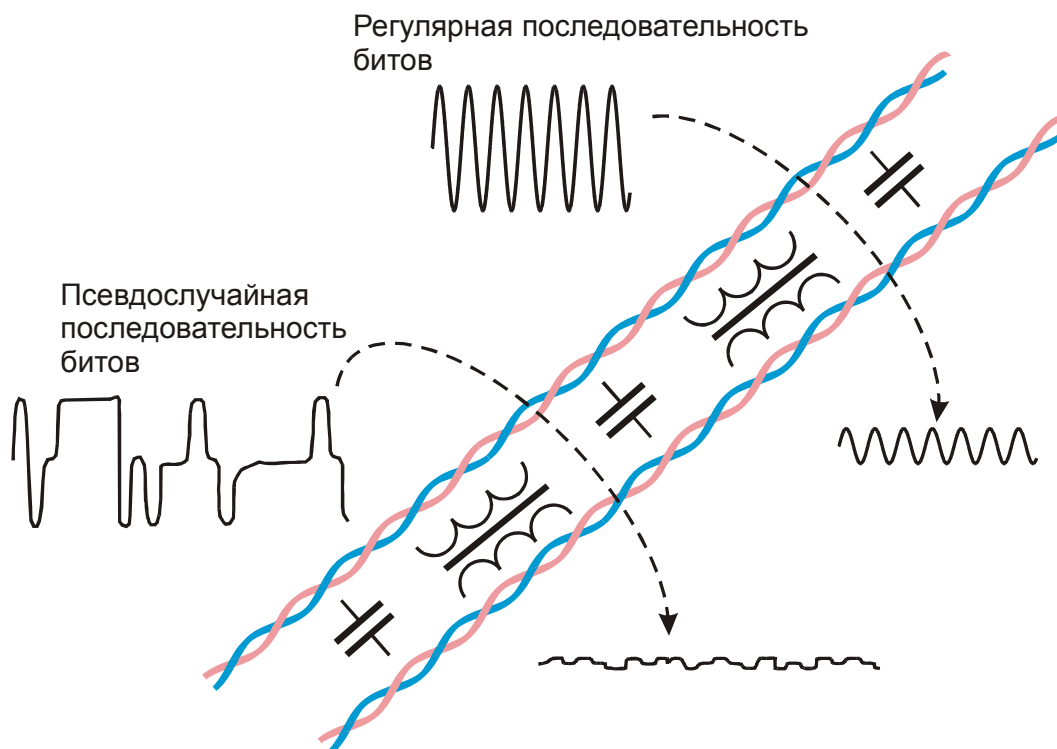
В результате малых постоянных колебаний около равновесного состояния осуществляется статистически наиболее правильная привязка синхросигнала CLK к сигналу DATA. Такая привязка обеспечивает стабильность данных на D-входе триггера D2 в момент прохождения положительного фронта сигнала CLK. Таким образом, на выходе схемы формируется синхросигнал и данные, выделенные из входного сигнала DIN.

Если сигнал DIN в течение длительного времени остаётся неизменным, то управление генератором VCO теряется, так как постоянно действующий на него через интегрирующую цепь нулевой или единичный сигнал F вызывает неуклонное повышение или снижение частоты синхросигнала. Скремблирование передаваемого по линии сигнала предотвращает его длительную фиксацию, т. е. повышает надёжность синхронизации.

### 2.13.6. Второе преимущество скремблирования — снижение уровня излучаемых помех

При распространении сигнала по витой паре медных проводов кабеля городской телефонной сети неизбежна передача части его энергии на соседние витые пары через паразитные элементы — емкостные и индуктивные, условно показанные на Рис. 2.63 в виде конденсаторов и трансформаторов. При этом наблюдается интересный эффект: регулярная последовательность битов создаёт более сильные помехи на соседних линиях, чем эта же последовательность, предварительно обработанная скремблером, т. е. преобразованная в псевдослучайную.

В данном примере использован один из способов трёхуровневого кодирования данных [1], но всё сказанное далее в равной мере относится и к другим способам. Важно то, что при заданном способе кодирования форма сигнала в линии зависит от вида передаваемых данных и может в предельных случаях приближаться либо к синусоиде, либо к шумовому сигналу, не содержащему ярко выраженных периодических составляющих.



**Рис. 2.63. Перекрёстные помехи: передача регулярных последовательностей битов сопровождается повышенным уровнем помех, наводимых на соседние витые пары проводов телефонного кабеля**

Далее для упрощения рассуждений предполагаем, что некоторая регулярная последовательность битов представлена чисто синусоидальным сигналом, что недалеко от истины.

Как видно из рисунка, и регулярный, и псевдослучайный сигналы имеют одинаковую амплитуду, но уровни наводимых этими сигналами помех различаются в несколько раз. При этом форма помехи от синусоидального сигнала повторяет его, а помеха от псевдослучайного сигнала явно отличается по форме от своего прототипа. Почему так происходит? Чем один сигнал “безопаснее” другого с точки зрения наводимых им помех?

Чтобы разобраться в этом, можно рассуждать “от противного”. Предположим, что как синусоидальный, так и псевдослучайный сигналы при паразитном прохождении на соседнюю линию кабеля в равной мере ослабляются и наследуют формы породивших их сигналов. Но чтобы это было именно так, паразитная система передачи сигналов с одной линии на другую должна была бы представлять собой идеальный линейный усилитель (с коэффициентом усиления, меньшим единицы).

Поясним сказанное двумя известными утверждениями.

*Первым необходимым условием неискаженной передачи сложного сигнала через некоторый усилитель является равномерность амплитудно-частотной характеристики этого усилителя в достаточно широкой полосе частот.* Это утверждение не требует особых доказательств. В его справедливости можно убедиться при прослушивании музыки, если с помощью регуляторов тембра “завалить” низкочастотные и высокочастотные компоненты аудиосигнала в усилительном тракте. Звук станет похожим на воспроизводимый из телефонной трубки из-за искажений первоначальной формы сигнала.

*Второе необходимое условие неискаженной передачи сложного сигнала — график фазочастотной характеристики усилителя должен быть представлен прямой линией, проходящей через начало координат.* Здесь, по-видимому, нужны некоторые пояснения. Не прибегая к строгим математическим выкладкам, рассмотрим упрощенные модели передачи сигнала через идеальный и неидеальный усилители (Рис. 2.64). В этих моделях усилители имеют единичные коэффициенты усиления, что несущественно.

Как показано на Рис. 2.64, *а*, входной сигнал  $C$  усилителей представлен суммой двух компонентных сигналов  $A$  и  $B$ , каждый из которых содержит ярко выраженную синусоидальную составляющую. (Строго говоря, следовало бы использовать “бесконечные” графики синусоидальных компонентных и суммарного сигналов, но это повлекло уменьшение наглядности рисунков.) Форма сигнала  $C^*$  на выходе идеального усилителя (Рис. 2.64, *б*) должна совпадать с формой входного сигнала  $C$ , следовательно, компонентные сигналы  $A^*$  и  $B^*$ , образующие в сумме сигнал  $C^*$ , должны находиться в тех же временных соотношениях, что и соответствующие сигналы  $A$ ,  $B$  и  $C$ .

Для выполнения этого условия необходимо чтобы задержка  $T_1$  распространения обоих компонентных сигналов через усилитель была одинакова. Эту задержку можно выразить как в единицах времени (наносекундах, микросекундах и т. д.), так и в числе периодов  $T_A$  или  $T_B$  низкочастотной и высокочастотной синусоидальных составляющих сигналов  $A$  и  $B$ . Последнее означает, что низкочастотное колебание, входящее в состав компонентного сигнала  $A$ , задержано на  $T_1/T_A$  периодов, а высокочастотное колебание, входящее в состав компонентного сигнала  $B$ , задержано на  $T_1/T_B$  периодов. Период синусоидального сигнала составляет 360 градусов или  $2\pi$  радиан в зависимости от принятых единиц измерения. Таким образом, задержку  $T_1$  можно трактовать как фазовый сдвиг  $\varphi_A$  сигнала  $A$  на  $(T_1/T_A) \times 2\pi$  [рад] или как фазовый сдвиг  $\varphi_B$  сигнала  $B$  на  $(T_1/T_B) \times 2\pi$  [рад]. Так как  $T_B < T_A$ , то высокочастотный сигнал должен иметь больший фазовый сдвиг при распространении через усилитель по сравнению с фазовым сдвигом низкочастотного сигнала.

Покажем, что зависимость фазового сдвига от частоты компонентного синусоидального сигнала линейна. Если  $F_A = 1/T_A$  и  $F_B = 1/T_B$  — частоты низкочастотного и высокочастотного колебаний компонентных сигналов на входе усилителя, и единицей измерения фазового сдвига является радиан, то можно записать, что

$$\varphi_B = (T_1/T_B) \times 2\pi;$$

$$\varphi_A = (T_1/T_A) \times 2\pi;$$

$$\varphi_B - \varphi_A = (T_1/T_B) \times 2\pi - (T_1/T_A) \times 2\pi = T_1 \times 2\pi (1/T_B - 1/T_A) = k \times (F_B - F_A), \text{ где } k = T_1 \times 2\pi.$$

В полученном соотношении при  $F_A = 0$   $\varphi_A = 0$ , так как для постоянной составляющей сигнала фазовая задержка равна нулю. С учётом этого получаем условие  $\varphi_B = k \times F_B$ , которое представляет собой искомую фазочастотную характеристику идеального усилителя и графически отображается в виде прямой, проходящей через начало координат с углом наклона по отношению к горизонтальной оси, равным  $\arctg k$  (Рис. 2.64, *в*).

Таким образом, условие равенства временных задержек  $T_1$  всех компонентных синусоидальных сигналов (для неискаженной передачи их суммы) трансформируется в условие линейности фазочастотной характеристики усилителя.

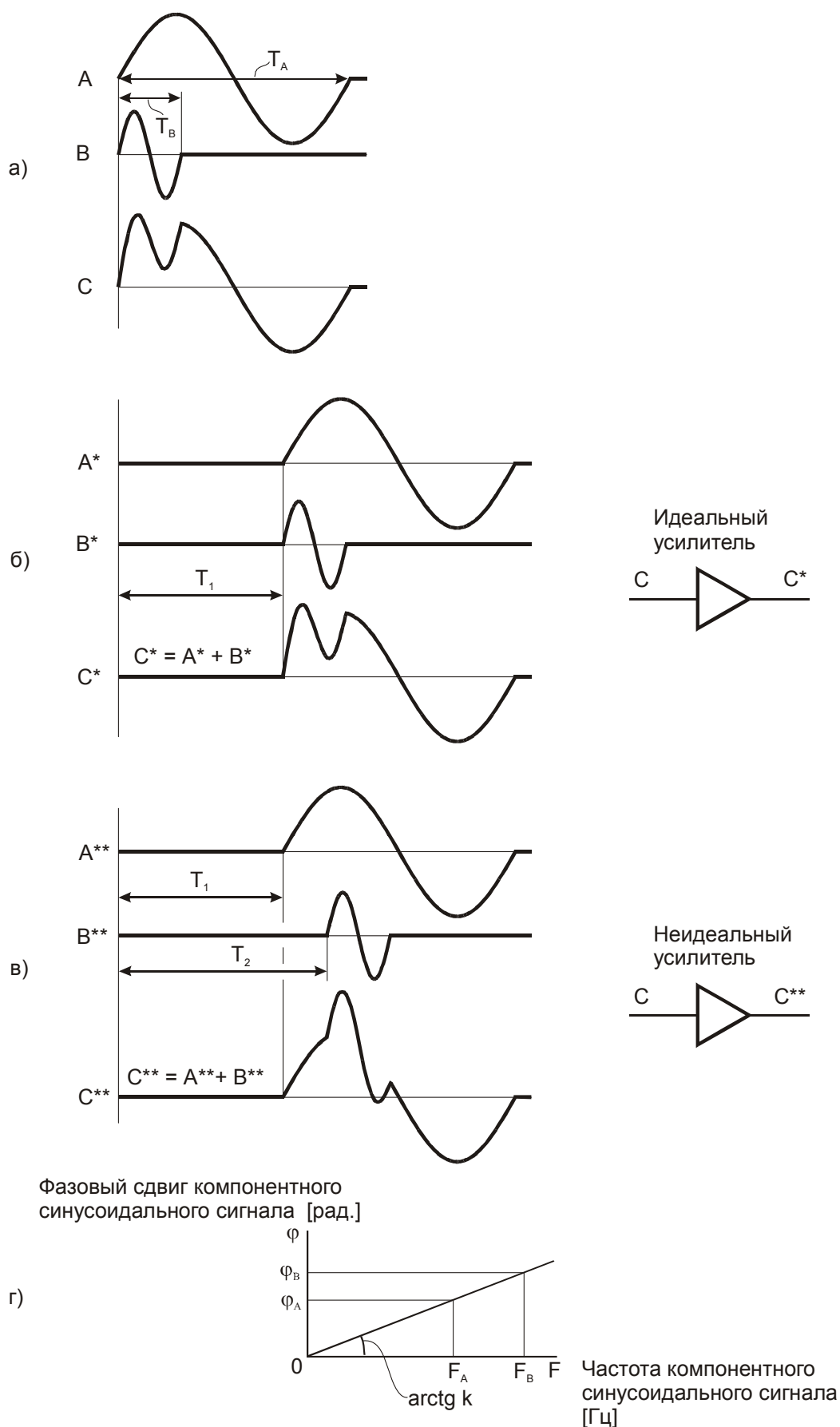
Если временные задержки ( $T_1$  и  $T_2$ , Рис. 2.64, *в*) компонентных синусоидальных сигналов не одинаковы, т. е. фазочастотная характеристика усилителя не линейна, то сумма  $C^{**}$  этих сигналов не соответствует входному сигналу  $C$ .

Компонентные сигналы  $A$  и  $B$  помимо ярко выраженных синусоидальных составляющих содержат бесконечно большое число иных гармоник, о которых не упоминалось в приведенных рассуждениях исключительно с целью упрощения иллюстрации того факта, что фазочастотная характеристика идеального усилителя должна быть линейной.

Возвращаясь к вопросу о том, почему скремблирование уменьшает уровень помех наводимых на соседние витые пары кабеля, теперь можно утверждать следующее.

1. Для того чтобы как синусоидальный, так и скремблированный сигналы в равной мере передавались в виде помех на соседние пары, необходимо выполнение рассмотренных выше весьма специфических условий передачи, которые, к счастью, на практике не выполняются. Действительно, совершенно невероятно, чтобы система передачи на основе распределённых паразитных параметров кабеля вела себя точно так же как идеальный линейный усилитель.

2. Псевдослучайный сигнал можно представить состоящим из большого числа синусоидальных компонентных сигналов малой амплитуды. Каждый компонентный сигнал достигает точки, в которой измеряется уровень помехи, со случайным фазовым сдвигом и случайным коэффициентом ослабления, поэтому помехи от компонентных сигналов в значительной мере взаимно уничтожаются, их сумма относительно невелика.



**Рис. 2.64.** Передача сигнала через идеальный и неидеальный усилители: а — компоненты входного сигнала и их сумма; б, в — компоненты выходных сигналов и их суммы, полученные на выходах идеального и неидеального усилителей; г — фазочастотная характеристика идеального усилителя

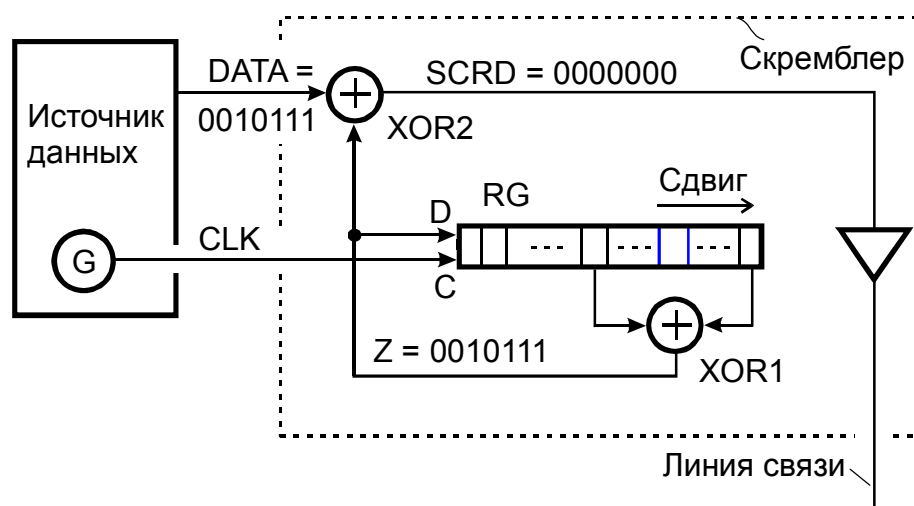
### 2.13.7. Непреднамеренное и умышленное нарушение работы скремблера с изолированным генератором псевдослучайной последовательности битов

Ранее кратко упоминалось о том, что скремблирование не всегда даёт ожидаемый результат, который, как предполагается, состоит в том, что выходная последовательность битов имеет характер “несмолкающего шума”. Действительно, возможны ситуации, при которых сигнал на выходе скремблера остаётся неизменным (постоянный лог. 0 или лог. 1) в течение десятков или даже сотен или тысяч тактов (периодов синхросигнала).

Поясним сказанное (Рис. 2.65). Если последовательность битов данных DATA совпадает со скремблирующей последовательностью битов Z, то последовательность SCRD скремблированных битов окажется состоящей только из лог. 0. Действительно, элемент Иключающее ИЛИ XOR2 формирует сигнал лог. 0 всякий раз, когда на его входы поступают одинаковые биты. Аналогично если биты данных DATA противоположны битам Z, то скремблированная последовательность будет содержать только единичные биты. В обоих случаях сигнал в линии будет оставаться неизменным, и, возможно, довольно долго, что может привести к потере синхронизации приёмника с передатчиком.

Так что же, скремблер настолько уязвим, что его применение теряет смысл?

Чтобы ответить на этот вопрос, следует различать непреднамеренное и умышленное разрушение скремблированных последовательностей битов. Первое не предполагает злого умысла со стороны искушённого пользователя системы передачи данных — сигнал на выходе скремблера перестаёт изменяться в силу чисто вероятностных событий, частоту которых нетрудно вычислить. Умышленное воздействие на скремблер со стороны “хакера” (злонамеренного пользователя) состоит в подборе такой последовательности битов данных DATA, при которой сигнал на выходе скремблера остаётся неизменным; при этом длина последовательности может составлять тысячи бит, что было бы практически невероятно при чисто случайном её происхождении.



**Рис. 2.65. Фрагмент схемы (Рис. 2.61), поясняющий уязвимость скремблера по отношению к некоторым последовательностям битов DATA**

Действительно, предположим, что последовательность битов DATA случайна. Вероятность совпадения очередной пары битов DATA и Z равна  $\frac{1}{2}$ . Вероятность совпадения цепочек из двух битов равна  $\frac{1}{4}$  и т. д. Вероятность последовательного совпадения N битов на входах элемента XOR2 равна  $2^{-N}$ . При  $N = 40$  эта вероятность составляет менее  $10^{-12}$ . Если скорость передачи равна 10 Гбит/с =  $10^{10}$  бит/с, то на выходе скремблера паузы длиной 40 бит будут наблюдаться в среднем один раз в сто секунд.

При  $N = 60$  средний период следования пауз длиной 60 бит превысит  $10^8$  с и составит более трёх лет. Таким образом, если генератор PLL (см. Рис. 2.62) с фазовой автоподстройкой частоты достаточно стабилен и его частота не уходит из допустимых границ в течение пауз длиной,

скажем, 65 битовых интервалов, то система “скремблер — дескремблер” если и будет терять синхронизацию по причине неудачных кодовых ситуаций, то в среднем один раз в сто лет. С этим вполне можно мириться, не правда ли?

Но хакер может сократить это время до долей секунды, сконструировав “практически невозможные” с точки зрения теории вероятностей последовательности, вызывающие паузы длинной в тысячи бит. Такие последовательности способны нарушить синхронизацию системы передачи данных, если не приняты надлежащие меры защиты скремблера.

Проследим за возможными действиями хакера, направленными на подавление работы скремблера.

Предположим, что разрядность регистра RG (Рис. 2.65) невелика и равна, например, семи битам. Этот регистр может находиться в одном из  $2^7 - 1 = 127$  состояний (нулевое состояние исключено). Так как структура связей между элементами скремблера не представляет секрета, то для каждого возможного начального состояния регистра можно синтезировать сколь угодно длинную последовательность битов DATA, совпадающую с последовательностью Z (или противоположной ей). Длина этой последовательности может составлять, например, 1 кбит. Все последовательности, по сути, одинаковы и различаются лишь начальными точками отсчёта, которые определяются начальным состоянием регистра RG. Для подавления работы скремблера достаточно иметь в распоряжении только одну последовательность, соответствующую некоторому заданному состоянию регистра RG, например, состоянию “все единицы”.

Теперь остаётся периодически вставлять эту последовательность в поток пользовательских данных. Источник данных помещает синтезированную хакером последовательность битов в кадр, который передаётся в скремблер в составе потока данных DATA. Рано или поздно (в среднем, в одной реализации из 127) обстоятельства сложатся так, что к моменту выдачи первого бита синтезированной последовательности регистр RG окажется в благоприятном состоянии “все единицы”. Цель будет достигнута.

Так как потоки битов DATA и Z одинаковы, то в линии на протяжении более 1000 тактов будет постоянно присутствовать сигнал лог. 0, поэтому синхронизация между приёмником и передатчиком, вероятнее всего, нарушится, система передачи данных на некоторое время, например порядка десяти секунд, перестанет функционировать. Столь длительный перерыв может быть связан с необходимостью частичного или полного выполнения процедуры установления потерянной связи между устройствами. В эту процедуру входят такие операции как посылка в линию тестовых сигналов, адаптация параметров приёмников к параметрам принимаемых сигналов, настройка устройств компенсации эхо-сигналов, установление битовой и кадровой синхронизации и др.

Если атаки со стороны хакера последуют непрерывно одна за другой, то система передачи данных окажется вообще непригодной для использования, так как всякий раз, едва вернувшись в рабочий режим, она вновь окажется парализованной.

Одна из мер защиты скремблера от атак хакера — увеличение разрядности регистра RG, скажем, до 64 бит. Тогда время ожидания благоприятной для атаки ситуации может превысить время жизни самого хакера.

### **2.13.8. Непреднамеренное и умышленное нарушение работы скремблера с неизолированным генератором псевдослучайной последовательности битов**

Скремблер, построенный по схеме с неизолированным генератором псевдослучайной последовательности битов (Рис. 2.66), в большей степени подвержен опасности “подавления” по сравнению со скремблером на основе изолированного генератора. Это связано с тем, что опасными последовательностями битов для него являются те, которые часто встречаются на практике — это “все нули” или “все единицы”. Вероятность появления в потоке данных цепи из 1000 нулей или единиц несравнимо более высокая, чем вероятность появления некоторого заданного “случайного” 1000-разрядного кода.

Это следует из того, что при передаче данных далеко не все процессы протекают в соответствии с теорией вероятностей. Так, длинные цепочки нулей или единиц обычно формируются не в результате случайных стечений обстоятельств, а в результате выполнения действий, предусмотренных протоколами обмена данными в системе. Цепочки нулей или единиц могут соответствовать отсутствию данных, сообщениям об ошибках, отображать нулевые или вышедшие за диапазон измерения значения физических величин и т. п.

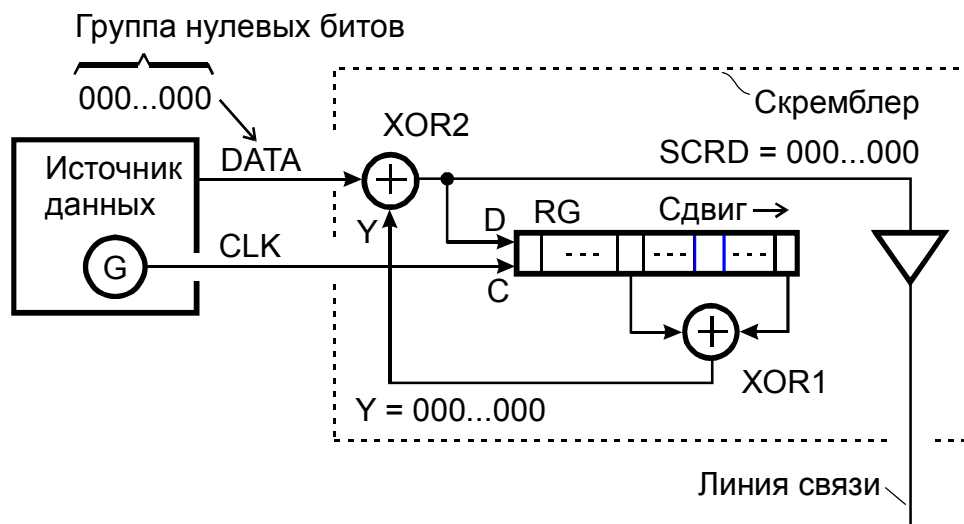


Рис. 2.66. Фрагмент схемы (Рис. 2.60), поясняющий процесс подавления процедуры скремблирования

В скремблере, показанном на Рис. 2.66, регистр RG может находиться в любом состоянии, в частности, в нулевом. Если сразу после перехода регистра RG в нулевое состояние на вход DATA подать серию лог. 0, то скремблирование будет подавлено, сигнал в линии останется неизменным. Это же относится и к передаче серии из лог. 1, если в регистре RG непосредственно перед этим сформировался код “все единицы”. Поэтому при периодической передаче достаточно длинных цепей из нулевых или единичных битов и небольшой разрядности регистра RG (например, равной семи битам), в среднем в одном случае из 128 скремблирование будет подавлено.

Таким образом, схема, показанная на Рис. 2.66, и без “помощи” хакера может потерять работоспособность при передаче длинных однородных последовательностей битов DATA. Один из вариантов защиты скремблера состоит в увеличении разрядности регистра RG с целью уменьшения вероятности формирования в нём нулевого или единичного кода. Этот вариант, правда, не единственный. Рассмотрим два альтернативных решения.

### 2.13.9. Первый вариант защиты скремблера

Скремблер (Рис. 2.67) [15, 1] содержит сдвиговый регистр RG1 с элементами Искключающее 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 SCRД = 1. Если SCRД = 0, то содержимое счетчика увеличивается на единицу по фронту сигнала CLK1 от тактового генератора G. 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 SCRД = 0. Если SCRД = 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 им-

пульс RESET установки в нуль некоторого разряда (или группы разрядов) сдвигового регистра. Счетчик лог. 1 служит детектором цепочек лог. 1 заданной длины.

Дескремблер построен аналогично. Он дополнительно содержит генератор PLL с фазовой автоподстройкой частоты для выделения синхросигнала CLK2 из скремблированного сигнала SCRD.

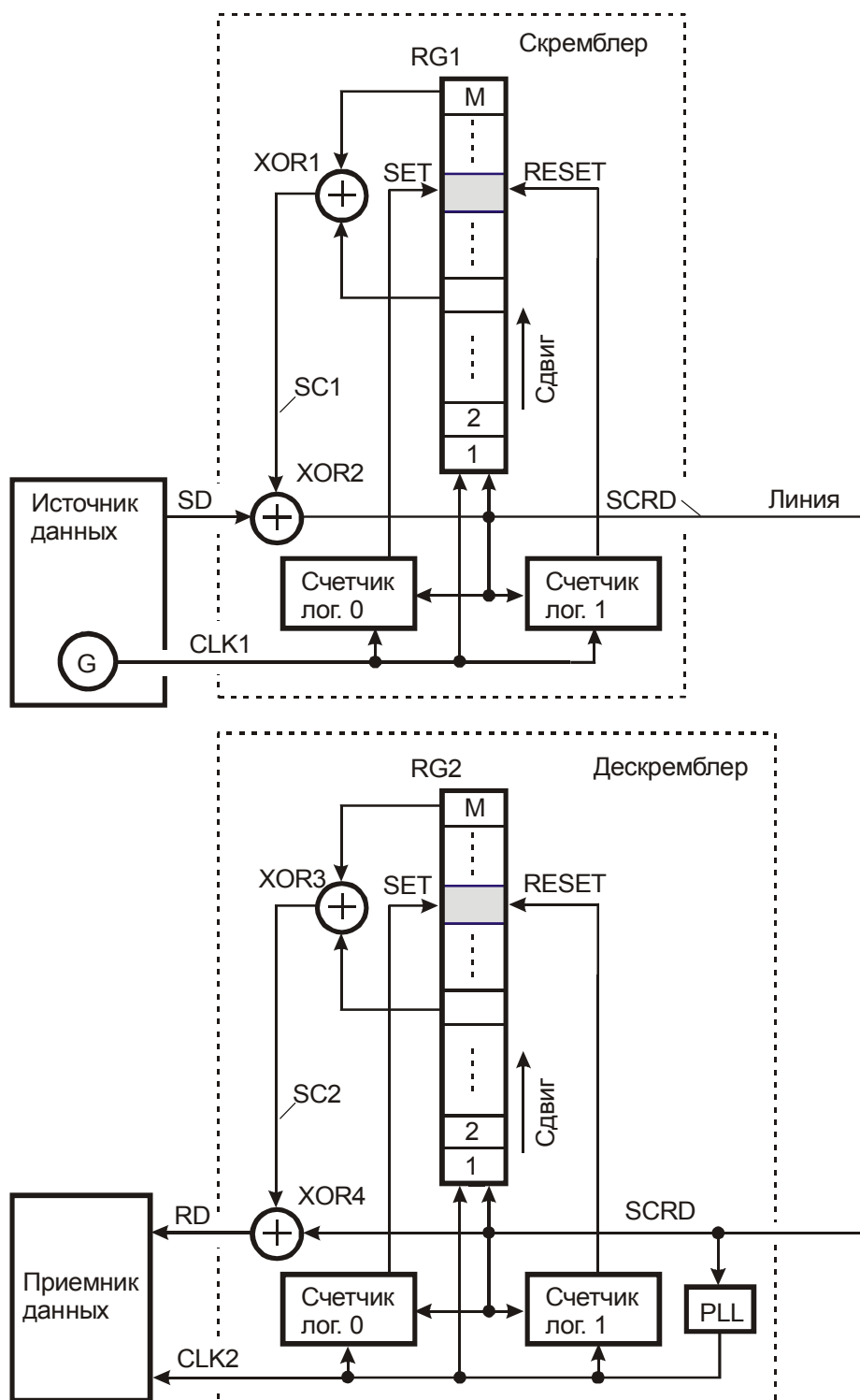


Рис. 2.67. Улучшенный вариант системы передачи данных с неизолрованными генераторами псевдослучайных последовательностей битов

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 последова-

тельность 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. Предположим, что эти счетчики исключены из схем скремблера и дескремблера. Схема остается работоспособной при условии, что поток SD не содержит некоторых “опасных” битовых последовательностей. Рассмотрим эти последовательности.

При работе системы не исключено, что поступающие от источника данные SD таковы, что логический элемент XOR2 скремблера в М последовательных тактах сформирует сигнал лог. 0 (М — разрядность сдвигового регистра). Тогда сдвиговый регистр RG1 (а синхронно с ним и регистр RG2) заполнится нулевыми битами. Если после этого источник данных 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 недопустимо.

Аналогичная ситуация возможна и после случайного заполнения сдвигового регистра единицами 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 перебрав все его возможные начальные состояния). 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 витых парах телефонного кабеля (например, с целью тестирования или попытки подавления работы систем, подключенных к этим парам).

### 2.13.10. Второй вариант защиты скремблера

Ранее было показано, что одной из мер защиты скремблера от потери работоспособности является увеличение разрядности сдвигового регистра. Это уменьшает вероятность формирования в регистре особых кодов, которые являются своего рода ключами для доступа к выходному потоку данных скремблера. Но на практике такое решение далеко не всегда осуществимо, так как скремблер, который следовало бы модернизировать, может быть встроен в действующую аппаратуру, не доступную пользователю. В таком случае можно воспользоваться решением, приведенным на Рис. 2.68 [16].

Для защиты системы “скремблер 1 — дескремблер 1”, показанной в правой части рисунка и построенной с использованием семиразрядных сдвиговых регистров, введена вторая, внешняя система “скремблер 2 — дескремблер 2”. Вторая система построена на основе 40-разрядных сдвиговых регистров RG. Обратите внимание на схемы включения этих регистров — скремблирующий Y1 и дескремблирующий Y2 сигналы формируются непосредственно в регистрах

RG, без использования элементов Иключающее ИЛИ, как в классической схеме (см. Рис. 2.60, Рис. 2.66).

Данные DATA от источника последовательно проходят через два скремблера, передаются на противоположную сторону линии связи, дважды дескремблируются и в виде потока DATA\* поступают в приёмник данных.

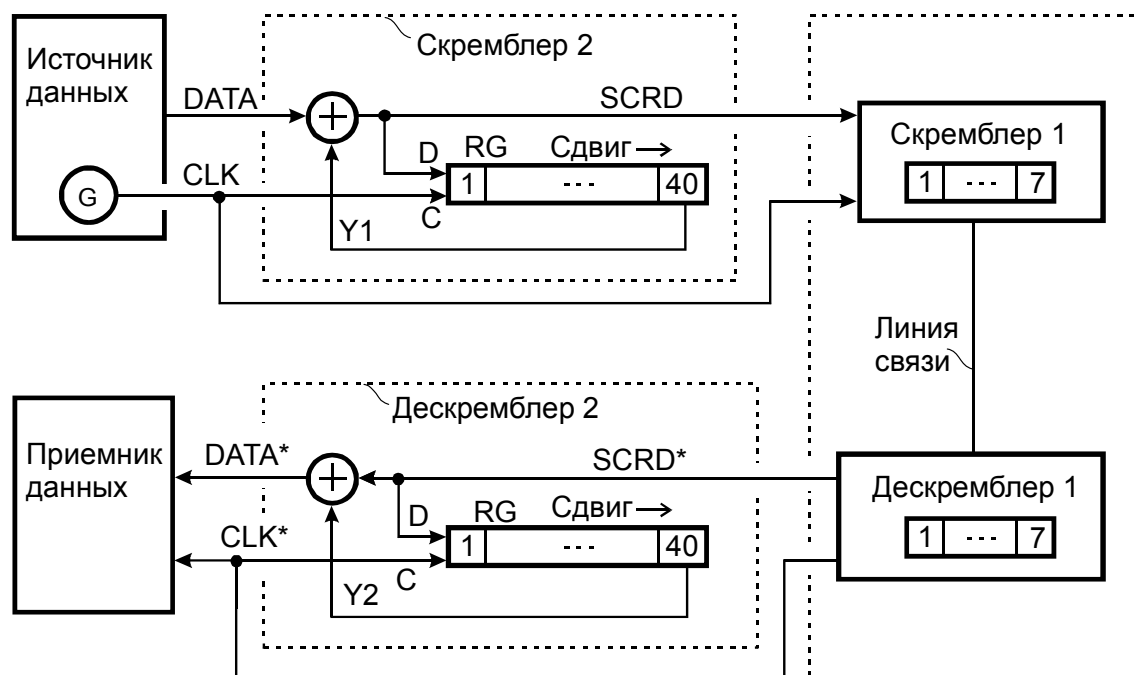


Рис. 2.68. Двухкаскадное скремблирование — дескремблирование

Система “скремблер 1 — дескремблер 1” выполнена по схеме с самосинхронизацией. Самосинхронизация достигается использованием неизолированных (Рис. 2.60) или изолированных генераторов псевдослучайных последовательностей битов, если последние включены по схеме, показанной на Рис. 2.69.

Состояние регистров скремблера не может быть известно хакеру. Оно зависит от всей предыстории работы системы, которую, возможно, включили в действие год тому назад. Чтобы создать цепочку нулей или единиц в линии связи, необходимо чтобы к моменту начала её формирования регистры обоих скремблеров содержали некоторые заданные коды. Вероятность такого события равна  $2^{-40} \times 2^{-7} = 2^{-47}$ . Поэтому хакеру пришлось бы пересылать в приёмник данных в среднем  $2^{47}$  или более  $10^{14}$  длинных цепей из нулевых битов, чтобы добиться однократных потерь синхронизации системы. Слишком малопродуктивная работа, не так ли?

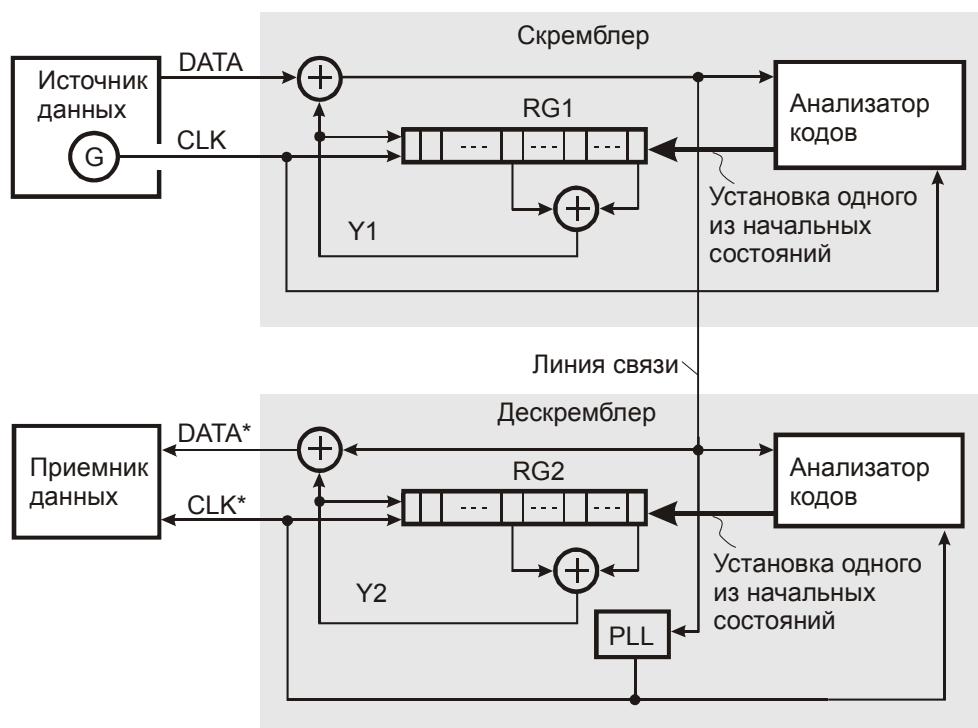
### 2.13.11. Вероятностная синхронизация системы “скремблер — дескремблер” с изолрованными генераторами псевдослучайных последовательностей битов

Как было показано, система “скремблер — дескремблер” с изолрованными от линии связи генераторами псевдослучайных битовых последовательностей (см. Рис. 2.61) обладает некоторыми преимуществами по сравнению с системой на основе неизолрованных генераторов (см. Рис. 2.60). Основное преимущество состоит в том, что ошибки, поступающие из линии, не размножаются. Кроме того, такая система более устойчива по отношению к неблагоприятным последовательностям битов, которые формируются в силу случайных стечений обстоятельств либо по злому умыслу хакера. Однако есть и существенный недостаток — сложность обеспечения кодовой синхронизации между скремблером и дескремблером.

Это связано с тем, что в “классической” системе для поддержания синхронной работы сдвиговых регистров скремблера и дескремблера необходимо периодически прерывать передачу полезных данных и передавать по линии связи служебные информационные кадры, содер-

жащие достаточно длинные цепочки синхронизирующих битов. Это уменьшает эффективную скорость передачи данных по линии, усложняет протокол обмена и требует значительного времени ожидания дескремблером служебного кадра в случае потери синхронизации. В течение этого времени передача пользовательских данных невозможна.

Предлагаемое решение (Рис. 2.69) лишено этих недостатков. Синхронизация системы на основе изолированных генераторов выполняется автоматически и не требует введения в поток передаваемых данных каких-либо служебных кадров. Это позволяет повысить эффективную скорость передачи данных и исключить из системы программные средства установления и поддержания синхронизации. Кроме того, уменьшаются потери данных при восстановлении синхронизации в случае её нарушения.



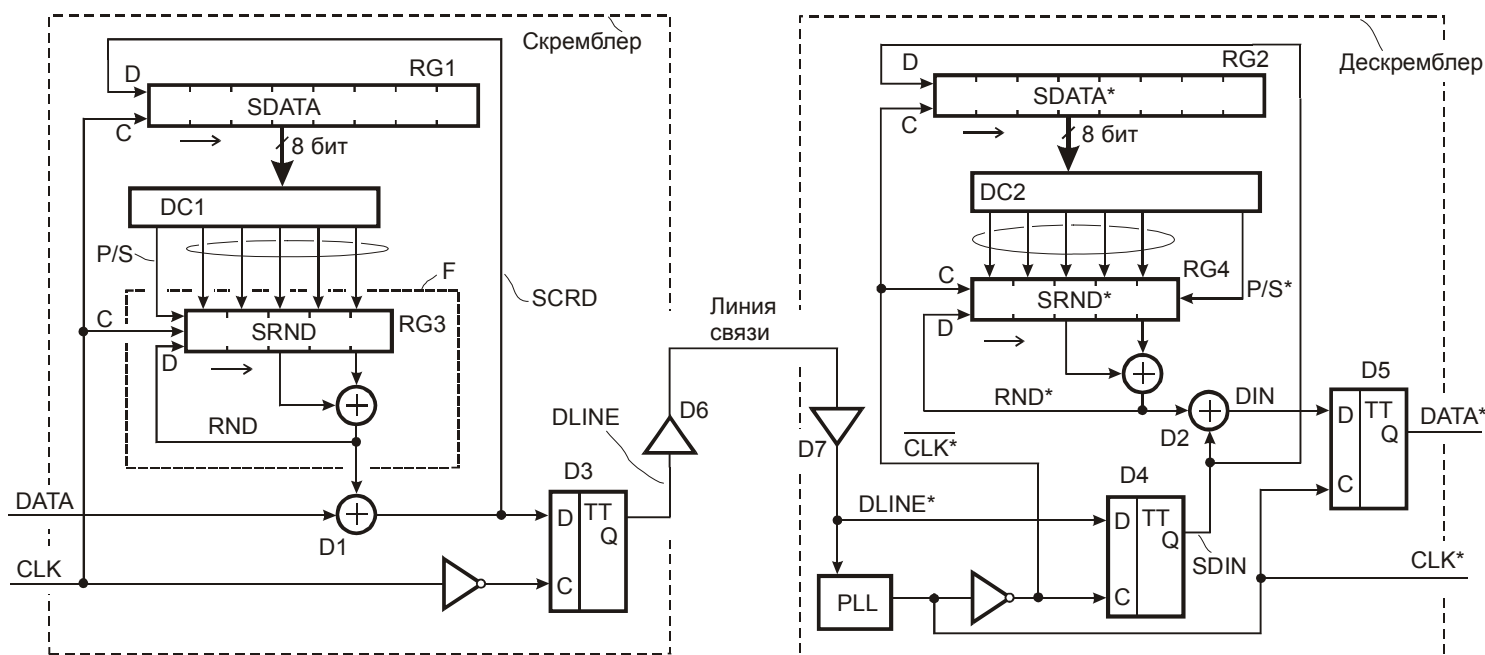
**Рис. 2.69. Система “скремблер — дескремблер” с самосинхронизацией изолированных генераторов псевдослучайных последовательностей битов**

В общем виде идея построения системы такова. Скремблер и дескремблер содержат изолированные от линии связи генераторы псевдослучайной последовательности битов с одинаковой структурой обратных связей. Скремблированный поток битов постоянно просматривается двумя одинаковыми анализаторами кодов, размещёнными в скремблере и дескремблере, с целью отыскания в нем заранее заданных кодов. Обнаружение каждого такого кода скремблером и дескремблером приводит к одновременной установке обоих генераторов псевдослучайной последовательности битов в определённое состояние, соответствующее этому коду. Таким образом, генераторы в случайные моменты одновременно устанавливаются в одинаковые состояния по мере передачи полезных данных, без применения какой-либо служебной процедуры установления синхронизации.

Если разрядность искоемых кодов достаточно велика (она не связана с разрядностью регистров RG1 и RG2), то одновременная коррекция состояний генераторов происходит сравнительно редко по мере обнаружения этих кодов в случайном потоке битов. Другими словами, большую часть времени генераторы работают в режиме “естественного” последовательного перехода от предыдущего состояния генератора к последующему. Если кодовая синхронизация не была нарушена, то моменты одновременной установки генераторов в новые начальные состояния лишь подтверждает её. Если кодовая синхронизация была ранее потеряна, то она восстанавливается при первом же обнаружении одного из заданных кодов в потоке скремблированных данных.

### 2.13.12. Детализация системы с вероятностной синхронизацией

Рассмотрим предлагаемую систему подробнее (Рис. 2.70).



**Рис. 2.70. Детализация схемы, приведенной на Рис. 2.69**

Сдвиговые регистры RG1 и RG2 совместно с дешифраторами DC1 и DC2 представляют собой анализаторы кодов. Регистры RG1 и RG2 предназначены для временного хранения фрагментов SDATA и SDATA\* потока скремблированных данных. В установившемся режиме эти фрагменты одинаковы (совпадают с точностью до задержки передачи). Прием очередного бита в регистр RG1 (RG2) происходит по положительному фронту сигнала на его синхронизирующем входе С. Одновременно с приёмом очередного бита с входа D ранее хранимые данные сдвигаются на один разряд вправо (по стрелке). Выдвигаемые из регистров биты теряются.

Разрядность регистра RG1 (RG2) выбрана равной восьми, хотя она может быть большей или меньшей. Динамику работы регистра RG1 можно проследить по таблице его состояний (Рис. 2.71).

| SCRD | SDATA <sub>2</sub> | SDATA <sub>16</sub> |
|------|--------------------|---------------------|
| 0    | 00010000           | 10                  |
| 1    | 00001000           | 08                  |
| 0    | 10000100           | 84                  |
| 0    | 01000010           | 42                  |
| 0    | 00100001           | 21                  |
| 1    | 00010000           | 10                  |
| 1    | 10001000           | 88                  |
| 0    | 11000100           | C4                  |
| 1    | 01100010           | 62                  |
| 0    | 10110001           | B1                  |
| 1    | 01011000           | 58                  |
| 0    | 10101100           | AC                  |
| 0    | 01010110           | 56                  |
| 1    | 00101011           | 2B                  |
| 1    | 10010101           | 95                  |
| 1    | 11001010           | CA                  |
| 0    | 11100101           | E5                  |
| 0    | 01110010           | 72                  |

**Рис. 2.71. Таблица состояний регистра RG1**

Генератор F псевдослучайной последовательности битов скремблера построен на основе сдвигового регистра RG3. Аналогичный генератор дескремблера выполнен на основе сдвигового регистра RG4.

Сдвиговые регистры RG3 и RG4 предназначены для временного хранения псевдослучайных кодов SRND и SRND\*. В установившемся режиме эти коды одинаковы (совпадают с точностью до задержки передачи). Прием очередного бита в регистр RG3 (RG4) с входа D происходит по положительному фронту сигнала на синхронизирующем входе С при условии, что на его управляющем входе P/S (P/S\*), задающем режим параллельного или последовательного приема данных, присутствует сигнал лог. 0. Одновременно с приёмом очередного бита с входа D происходит сдвиг ранее хранимого кода на один разряд вправо (по стрелке).

Если на управляющем входе P/S (P/S\*) регистра RG3 (RG4) присутствует сигнал лог. 1, то по положительному фронту сигнала на синхронизирующем входе С в регистр принимается параллельный код с группы входов, обведенной на рисунке овалом. В данном примере построения устройства разрядность регистра RG3 (RG4) выбрана равной пяти, хотя она может быть большей или меньшей. При этом точки подключения элемента Иключающее ИЛИ к регистру RG3 (RG4) выбираются в соответствии с таблицей, представленной на Рис. 2.59, в.

В таблице (Рис. 2.72, а) представлен список состояний генератора псевдослучайной последовательности битов на основе сдвигового регистра RG3. Диаграмма состояний этого генератора (Рис. 2.72, б) отражает перемещение указателя А текущего состояния по кольцевому пути; линии В и С разделяют диаграмму на четыре примерно равных сектора.

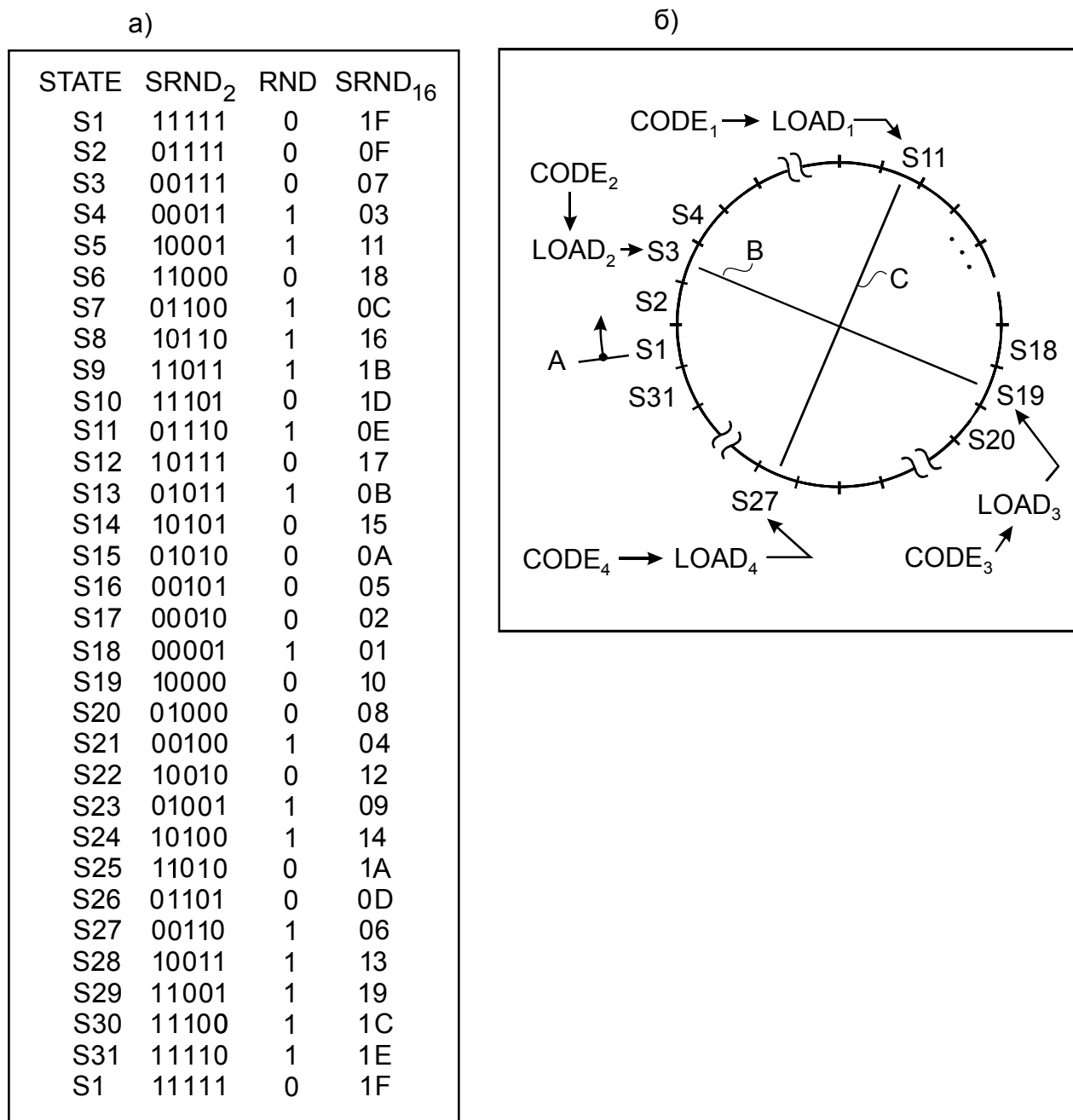


Рис. 2.72. Таблица (а) и диаграмма (б) состояний генератора псевдослучайной последовательности битов

Начальное состояние регистра RG3 может быть любым, в частности, нулевым. Выход из нулевого состояния происходит при записи в регистр ненулевого параллельного кода. Программа инициализации скремблера может предусматривать выдачу на его вход некоторого кода CODE<sub>1</sub>, который распознаётся дешифратором DC1. Если в регистре RG3 первоначально присутствовал нулевой код, то код CODE<sub>1</sub> без изменения проходит через элемент D1 и последовательно загружается в регистр RG1.

Дешифратор DC1 реагирует на него переводом регистра RG3 в режим параллельной загрузки (P/S = 1) и формированием ненулевого кода LOAD<sub>1</sub>, который затем принимается в регистр. Таким образом, генератор F выходит из запрещенного состояния 000...0. Если первоначальное состояние регистра RG3 было ненулевым, то выдача кода CODE<sub>1</sub> на вход скремблера оказывается бесполезной, но не приводит к каким-либо нежелательным последствиям. Возможна также и аппаратная установка регистра RG3 в ненулевое состояние (соответствующий вход установки начального состояния регистра не показан).

Начальное состояние регистра RG4 также может быть любым, в том числе, нулевым. Это состояние обновляется, т. е. становится заведомо ненулевым при обнаружении дешифратором DC2 в скремблированном потоке данных одного из заранее заданных кодов ( $CODE_1$  и, возможно, других).

Элементы Иключающее ИЛИ D1 и D2 формируют скремблированный SCRD и дескремблированный DIN сигналы данных.

Триггеры D3, D4 и D5 принимают биты данных с входа D по положительному фронту сигнала на входе синхронизации C. Триггеры D3 и D5 формируют выходные сигналы DLINE и DATA\*, в которых на границах между битовыми интервалами сигнал может измениться только один раз, в то время как входные сигналы SCRD и DIN этих триггеров на границах между битовыми интервалами могут изменяться многократно из-за неодновременного протекания переходных процессов на входах каждого из элементов D1 и D2.

Триггер D4 практически полностью устраняет джиттер входного сигнала (“дрожание” фронтов на границах между битовыми интервалами) благодаря тому, что приём бита в этот триггер происходит в центре битового интервала, когда переходные процессы сигнала DLINE\* уже закончились. Остаточный джиттер сигнала SDIN на выходе триггера D4 определяется неидеальностью сигнала CLK\* на выходе генератора PLL с фазовой автоподстройкой частоты. Исходные состояния триггеров D3 — D5 произвольны.

Генератор PLL с фазовой автоподстройкой частоты предназначен для формирования высокостабильного синхросигнала CLK\* на основе непрерывного слежения за входным сигналом DLINE\*. Положительный фронт сигнала CLK\* привязан к моментам изменения сигнала DLINE\*. Отрицательный фронт сигнала CLK\* формируется в середине битового интервала сигнала DLINE\*, что соответствует его установившемуся значению.

Благодаря достаточной инерционности генератора PLL сигнал CLK\* практически нечувствителен к джиттеру сигнала DLINE\* и иным его кратковременным искажениям, вызванным помехами в линии связи.

Дешифратор DC1 (DC2) предназначен для выделения в потоке скремблированных данных, проходящем через сдвиговый регистр RG1 (RG2), определенных кодов  $CODE_1$ ,  $CODE_2$ , ...,  $CODE_J$ . При обнаружении дешифратором DC1 (DC2) указанных кодов на его выходах (обведены овалом) формируется соответствующий M-разрядный код  $LOAD_1$ ,  $LOAD_2$ , ...,  $LOAD_J$  для последующей параллельной загрузки сдвигового регистра RG3 (RG4). В данном примере построения системы  $J = 4$ ,  $M = 5$ . При обнаружении любого кода  $CODE_1$ ,  $CODE_2$ , ...,  $CODE_J$  дешифратор DC1 (DC2) формирует также единичный сигнал на входе P/S (P/S\*) управления режимом работы регистра RG3 (RG4), подготавливая его к параллельному приёму данных по положительному фронту очередного синхроимпульса на входе C.

Усилитель D6 (D7) предназначен для передачи (приёма) скремблированного сигнала данных в линию (из линии). Параметры усилителей D6 и D7 определяются типом линии связи, которая может быть выполнена в виде витой пары проводов, коаксиального или оптоволоконного кабеля и т. п.

### 2.13.13. Описание работы системы

Далее приведено описание работы системы (Рис. 2.69, Рис. 2.70).

На входы скремблера поступают данные DATA и сопровождающий их сигнал синхронизации CLK. Положительные фронты сигнала CLK (моменты T0, T1, ..., T18 на Рис. 2.73) соответствуют границам между битовыми интервалами сигнала данных DATA. По положительным фронтам сигнала CLK изменяется содержимое регистра RG1 (см. диаграмму сигнала SDATA), генератор F переходит в новое состояние. При этом формируется очередной псевдослучайный бит RND, который складывается по модулю два с битом данных DATA и преобразуется в скремблированный бит данных SCRD. По окончании переходных процессов, в момент формирования отрицательного фронта сигнала CLK бит SCRD принимается в триггер D3 (см. диаграмму сигнала DLINE) и через усилитель D6 передается в линию связи.

В интервале времени  $T_8 — T_9$  дешифратор DC1 формирует сигнал лог. 1 на входе P/S управления режимом работы регистра RG3, подготавливая его к приёму параллельных данных в момент  $T_9$ .

В отсутствие параллельной загрузки генератор F псевдослучайной последовательности битов последовательно, циклически проходит через ряд состояний  $S_1, S_2, S_3, \dots, S_{31}, S_1, S_2$  и т. д., как показано в таблице и на диаграмме (Рис. 2.72, а, б). В состоянии  $S_1$  (см. первую строку таблицы, а также указатель A на диаграмме) в регистре RG3 хранится пятиразрядный двоичный код  $11111_2 = 1F_{16}$ , на выходе RND генератора F сформирован сигнал лог. 0. В следующем такте указатель A перемещается по часовой стрелке и фиксируется на соседней позиции, генератор F переходит в состояние  $S_2$ , при котором  $SRND = 01111_2 = 0F_{16}$ ,  $RND = 0$  и т. д. Этот процесс циклически повторяется, указатель A вращается по кругу, последовательно проходя все возможные состояния  $S_i$ .

Параллельная загрузка регистра RG3 в произвольном такте приводит к принудительной установке генератора в одно из заданных состояний, в данном примере, в состояния  $S_3, S_{11}, S_{19}$  или  $S_{27}$ . Эти состояния предпочтительно выбираются так, чтобы на диаграмме (Рис. 2.72, б) дуги  $S_3 — S_{11}, S_{11} — S_{19}, S_{19} — S_{27}$  и  $S_{27} — S_3$  имели примерно равную длину (см. линии В и С, которые разделяют окружность на четыре примерно равные части). В процессе работы системы указатель A сравнительно редко, с равной вероятностью устанавливается в эти состояния, а в промежутках между такими установками продолжает равномерное перемещение по диаграмме в направлении по часовой стрелке.

Выбор нескольких (а не одного) заданных состояний, в которые генератор переходит в моменты его параллельной загрузки, целесообразен в тех случаях, когда число состояний генератора достаточно велико, и в течение полного оборота указателя A вероятность параллельной загрузки регистра RG3 близка к единице. Поэтому если указатель A периодически “срывается” с равномерного вращения в одно и то же заданное состояние, то вероятность того, что он успеет совершить хотя бы один полный оборот, становится невысокой. Иными словами, некоторые состояния генератора F будут использоваться реже, чем другие, а тогда отмеченные ранее (при описании генератора, см. Рис. 2.59) свойства “канонической” псевдослучайной последовательности битов будут в некоторой степени утеряны, что нежелательно. Наличие нескольких фиксированных точек установки, равномерно распределенных по диаграмме, выравнивает вероятности использования всех возможных состояний генератора F.

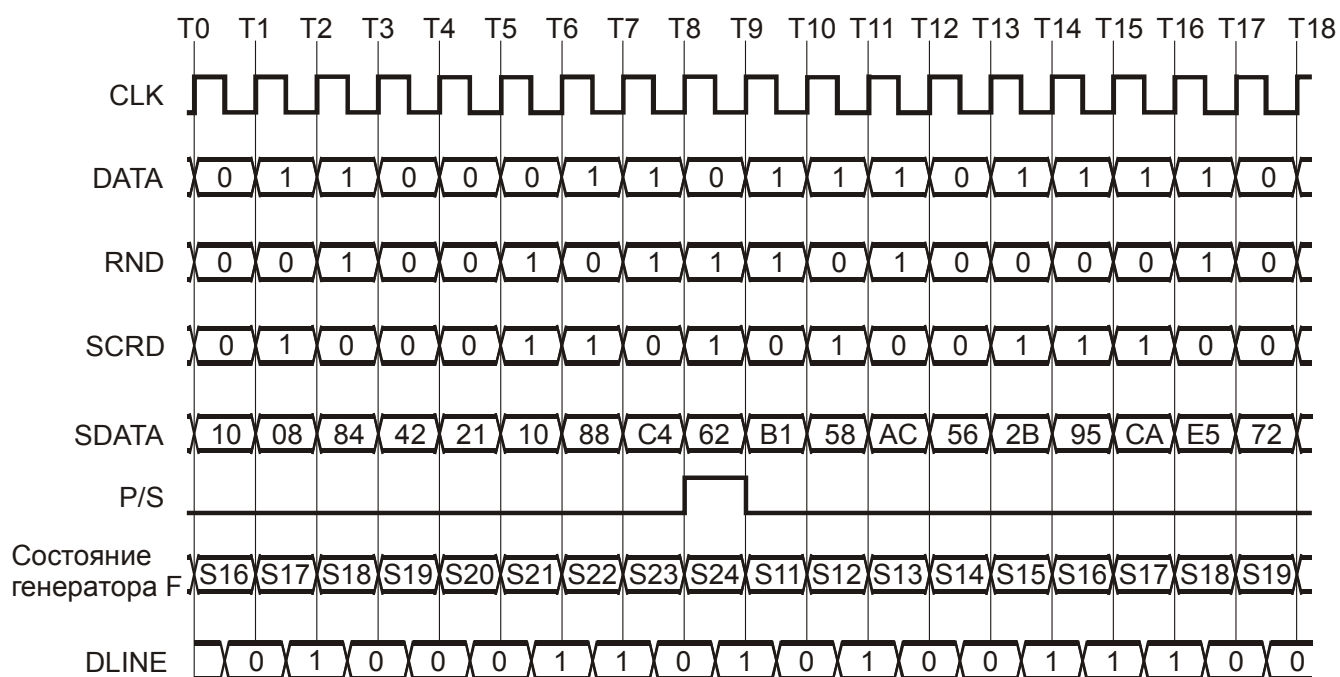


Рис. 2.73. Временные диаграммы работы скремблера

Как показано на диаграммах сигналов SDATA и P/S (Рис. 2.73), одним из кодов, вызы-

вающих принудительную установку генератора F в фиксированное состояние, является код  $SDATA = CODE_1 = 62_{16} = 01100010_2$ . Этот код присутствует в регистре RG1 в интервале времени T8 — T9 и, как уже отмечалось, дешифратор DC1 реагирует на него подготовкой регистра RG3 к приёму параллельного кода  $LOAD_1$ . Этот код в данном примере равен  $0E_{16} = 01110_2$  и соответствует состоянию S11 генератора F (см. таблицу на Рис. 2.72, а). Таким образом, в момент T9 цепь последовательных переходов ... S16, S17, ..., S23, S24 разрывается, и, вместо перехода в очередное состояние S25, генератор F “перескакивает” в состояние S11. После этого формируется новая цепь последовательных переходов: S11, S12, ..., S18, S19, ... — вплоть до возникновения очередной ситуации, при которой эта цепь разрывается, а затем образуется следующая цепь с одним из начальных состояний S3, S11, S19 или S27 и т. д.

Принятые из линии скремблированные данные DLINE\* синхронизируют генератор PLL с фазовой автоподстройкой частоты, в результате на его выходе формируется сигнал CLK\* (Рис. 2.74). Сигнал SDIN на выходе триггера D4 повторяет сигнал DLINE\* с задержкой на половину периода синхросигнала, при этом сигнал SDIN, как уже отмечалось, практически не содержит фазовых искажений (джиттера). Скремблированные данные SDIN последовательно проходят через регистр RG2. После его заполнения данные  $SDATA^*$  совпадают с данными SDATA в регистре RG1 скремблера.

Это следует из того, что, во-первых, источник данных для обоих регистров общий — выход элемента Иключающее ИЛИ D1 и, во-вторых, ничто не препятствует одновременному (с точностью до задержки передачи) заполнению обоих регистров (RG1 и RG2) данными от этого источника. Так как дешифраторы DC1 и DC2 идентичны, а данные на их входах одинаковы, то сигналы на выходах этих дешифраторов также совпадают с точностью до задержки передачи. Поэтому рассмотренный ранее процесс установки генератора F в определенное состояние протекает также и в дескремблере. В интервале времени T8 — T9 (Рис. 2.74) на входе P/S\* регистра RG4 формируется сигнал лог. 1, в момент T9 в регистр принимается параллельный код  $0E_{16}$ , соответствующий состоянию S11.

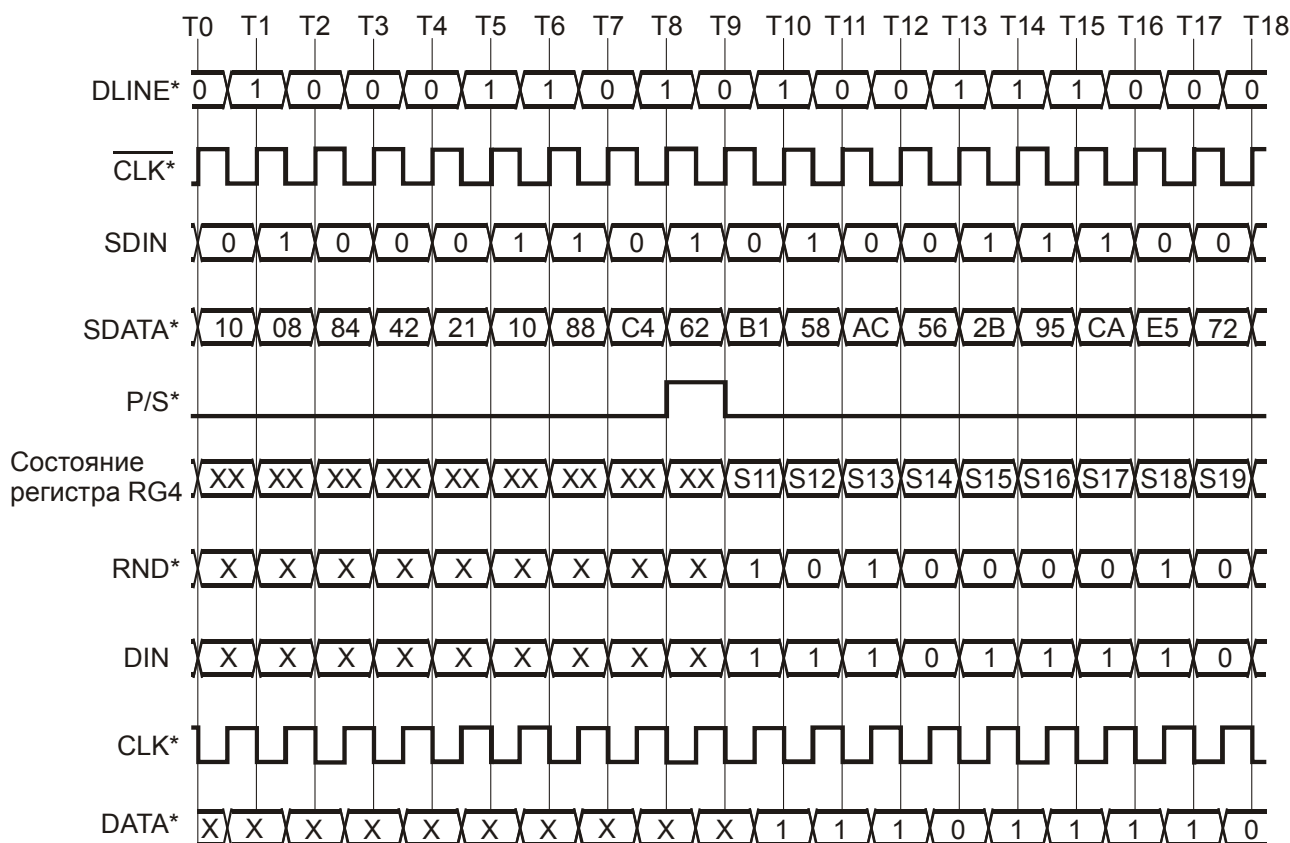


Рис. 2.74. Временные диаграммы работы дескремблера

Независимо от предыстории состояния генератора псевдослучайной последовательности битов дескремблера, начиная с момента  $T_9$  этот генератор синхронизируется с генератором  $F$  скремблера, в том смысле, что формируемые обоими генераторами последовательности битов совпадают. Неопределенные состояния и сигналы в начальный период, когда кодовая синхронизация между генераторами отсутствовала, помечены на диаграммах (Рис. 2.74) символами “X”.

Начиная с момента  $T_9$  скремблирующая  $RND$  и дескремблирующая  $RND^*$  последовательности битов совпадают, поэтому сигнал  $DIN$  дескремблированных данных совпадает с сигналом  $DATA$  на входе скремблера. Выходной сигнал  $DATA^*$  данных, “очищенный” от возможных многократных переключений на границах между битовыми интервалами, поступает на выход устройства и сопровождается сигналом  $CLK^*$ . Таким образом, входные сигналы  $DATA$  и  $CLK$  преобразуются в совпадающие с ними (с точностью до задержки передачи) выходные сигналы  $DATA^*$  и  $CLK^*$ .

Частота следования моментов синхронной установки регистров  $RG3$  и  $RG4$  в одинаковые состояния (моментов синхронизации) зависит от скорости передачи данных, а также от разрядности и числа  $J$  кодов  $CODE_1, CODE_2, \dots, CODE_J$ , распознаваемых дешифраторами  $DC1$  и  $DC2$ .

При  $J = 1$  и разрядности регистра  $RG1$  ( $RG2$ ), равной 20 бит, в скремблированном потоке данных в среднем в каждой цепи из  $2^{20} \approx 10^6$  бит будет встречаться один искомый код, равный  $CODE_1$ . При скорости передачи данных, равной 1 Мбит/с, средняя частота следования моментов синхронизации составляет примерно 1 Гц. При  $J = 4$  частота моментов синхронизации увеличивается в четыре раза.

Для уменьшения вероятности ложного распознавания кодов  $CODE_1, CODE_2, \dots, CODE_J$  дешифратором  $DC2$  дескремблера в связи с поступлением из линии связи ошибочных битов разрядность регистров  $RG1$  и  $RG2$  можно увеличить.

Применение предлагаемой системы “скремблер — дескремблер” позволяет повысить скорость передачи полезных данных и уменьшить их потери при восстановлении нарушенной синхронизации благодаря исключению из потока данных служебной синхронизирующей информации.

## 2.14. Вопросы и задачи к разделу 2.13

- 2.14.1. В системе передачи данных, показанной на Рис. 2.75, применены скремблер и дескремблер с изолированными генераторами псевдослучайных последовательностей битов;  $M = 5$ ,  $N = 3$ .
1. Какова вероятность получения в линии связи цепи из 10000 нулей в ситуациях, когда на вход скремблера подаётся цепь из:
    - а) 10000 случайных битов;
    - б) 10000 нулевых битов;
    - в) 10000 единичных битов?
  2. Каковы последствия получения в линии связи цепи из 10000 нулей?
  3. Какими должны быть действия злоумышленника для получения в линии 10000 нулевых сигналов? Достижима ли эта цель? Как можно противодействовать её достижению?
  4. Кратко опишите процесс установления кодовой синхронизации между скремблером и дескремблером.
  5. Каковы последствия искажения одного бита в линии связи?
  6. Существуют ли запрещённые коды в регистре RG1 (RG2)?

Литература: [1], гл. 8; [4]

*Решение задачи приведено в разделе 3 на с. 175*

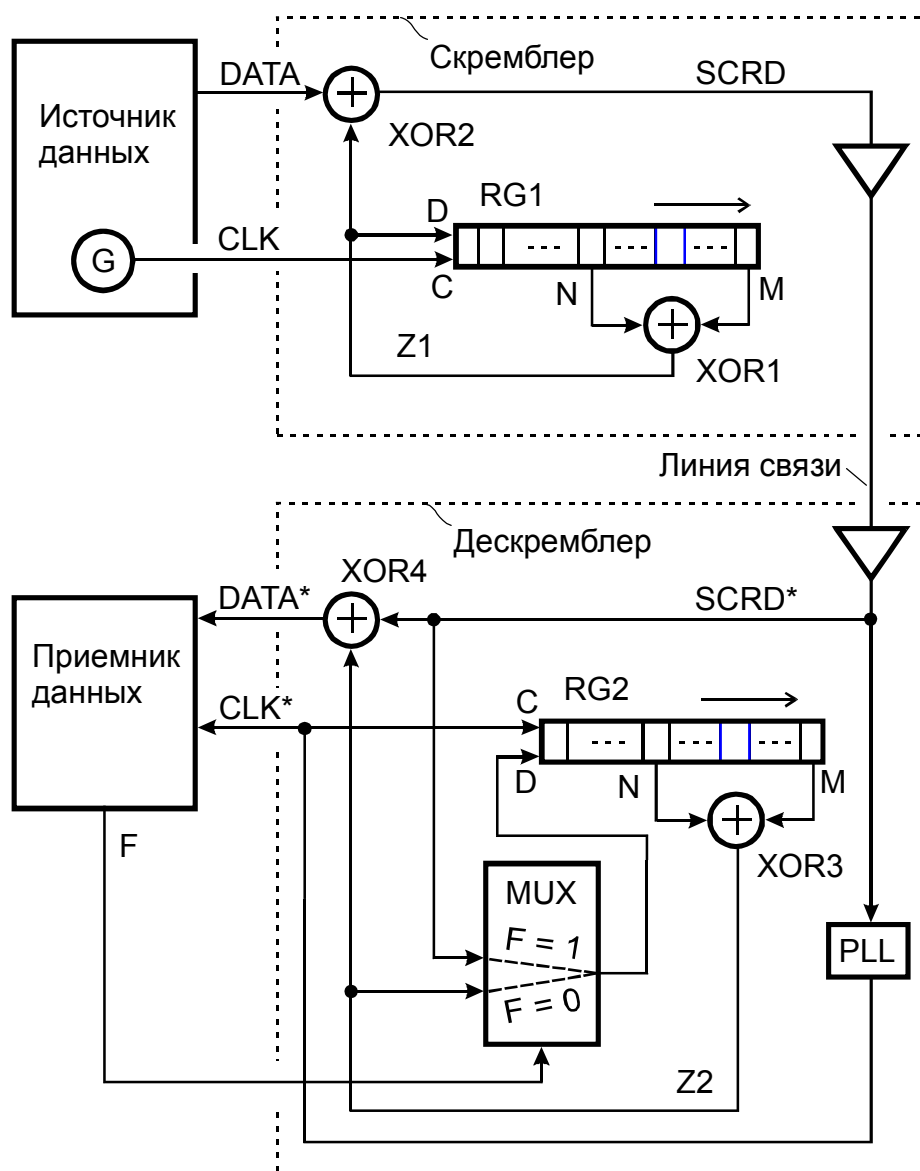


Рис. 2.75. Система передачи данных

- 2.14.2. В системе передачи данных, показанной на Рис. 2.76, применены скремблер и дескремблер с неизолированными генераторами псевдослучайных последовательностей битов;  $M = 5$ ,  $N = 3$ .
- Какова вероятность получения в линии связи цепи из 10000 нулей в ситуациях, когда на вход скремблера подаётся цепь из:
    - 10000 случайных битов;
    - 10000 нулевых битов;
    - 10000 единичных битов?
  - Каковы последствия получения в линии связи цепи из 10000 нулей?
  - Назовите два способа защиты этой системы от злонамеренного пользователя.
  - Каково время вхождения данной системы в кодовую синхронизацию и как развивается этот процесс?
  - Каковы последствия искажения одного бита в линии связи?
  - Существуют ли запрещённые коды в регистре RG1 (RG2)?

Литература: [1], гл. 8; [4]

Решение задачи приведено в разделе 3 на с. 176

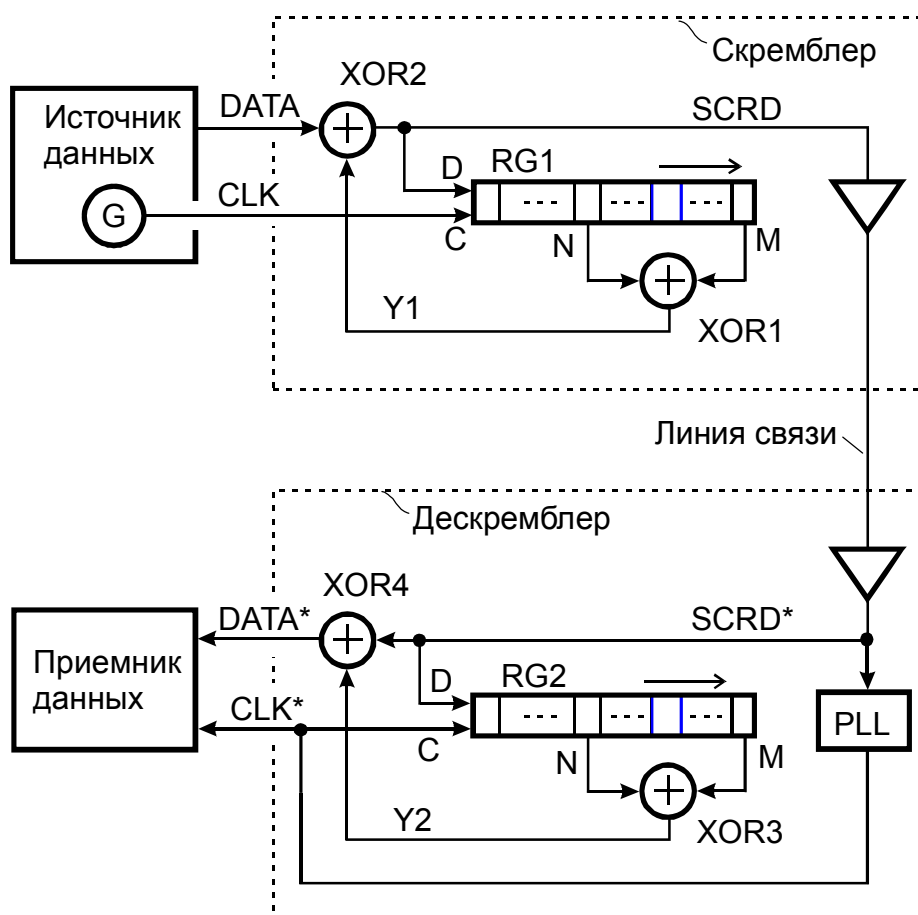


Рис. 2.76. Система передачи данных

2.14.3. В схеме генератора псевдослучайной последовательности битов, приведенной на Рис. 2.77,  $M = 39$ ,  $N = 35$  [1]. Напишите последовательность длиной 50 бит, формируемую на выходе генератора, если его начальное состояние равно:

- 1) 000...0;
- 2) 111...1.

Каков период повторения битов?

Литература: [1], гл. 8; [4]

Решение задачи приведено в разделе 3 на с. 177

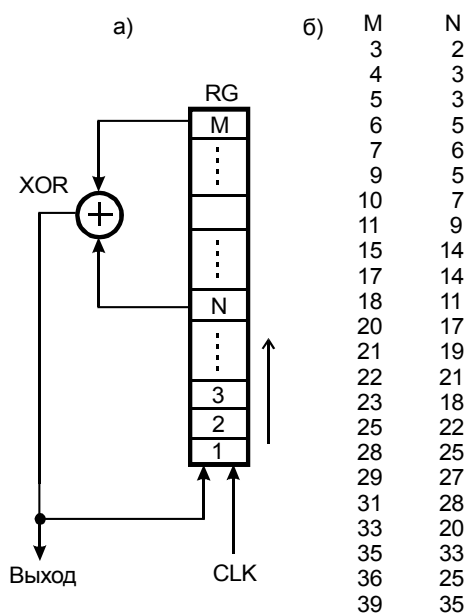


Рис. 2.77. Генератор (а) псевдослучайной последовательности битов и таблица (б) для выбора промежуточной точки N подключения цепи обратной связи

2.14.4. В схеме генератора псевдослучайной последовательности битов, приведенной на Рис. 2.78,  $M = 39$ ,  $N = 35$  [1]. Напишите последовательность длиной 50 бит, формируемую на выходе генератора, если его начальное состояние равно:

1) 111...1;

2) 000...010 (единица в разряде  $M - 1$  регистра RG, остальные разряды содержат нули).

Каков период повторения битов?

Литература: [1], гл. 8; [4]

Решение задачи приведено в разделе 3 на с. 177

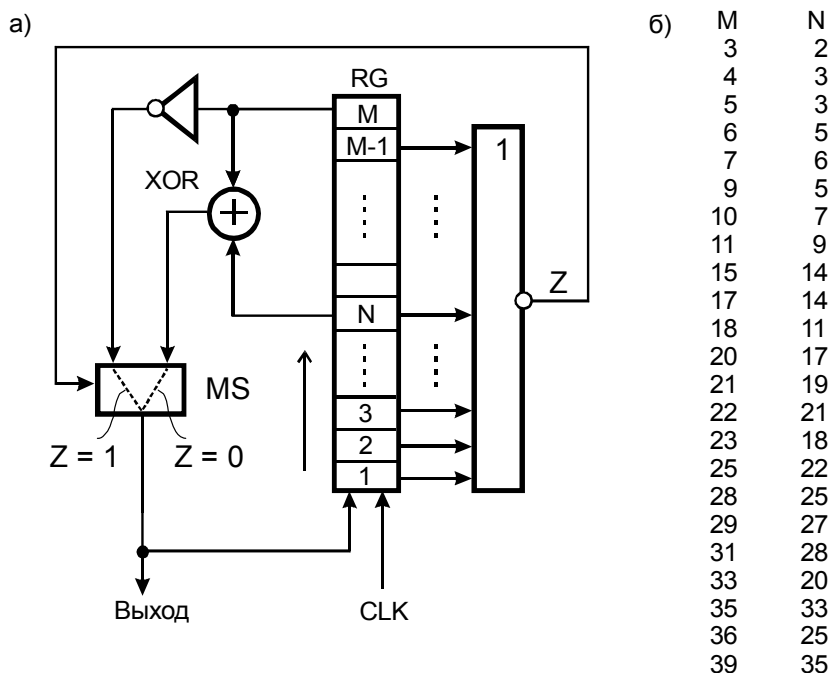


Рис. 2.78. Генератор (а) псевдослучайной последовательности битов и таблица (б) для выбора промежуточной точки N подключения цепи обратной связи

## 2.15. Усовершенствование систем с попутной синхронизацией

Литература к разделу 2.15: [1], гл. 3

### 2.15.1. Введение в раздел 2.15

В интерфейсах семейства RS для синхронной передачи данных из устройства 1 в устройство 2 (Рис. 2.79, а, б), например, из компьютера в модем, используются две линии: TxD и CLK.

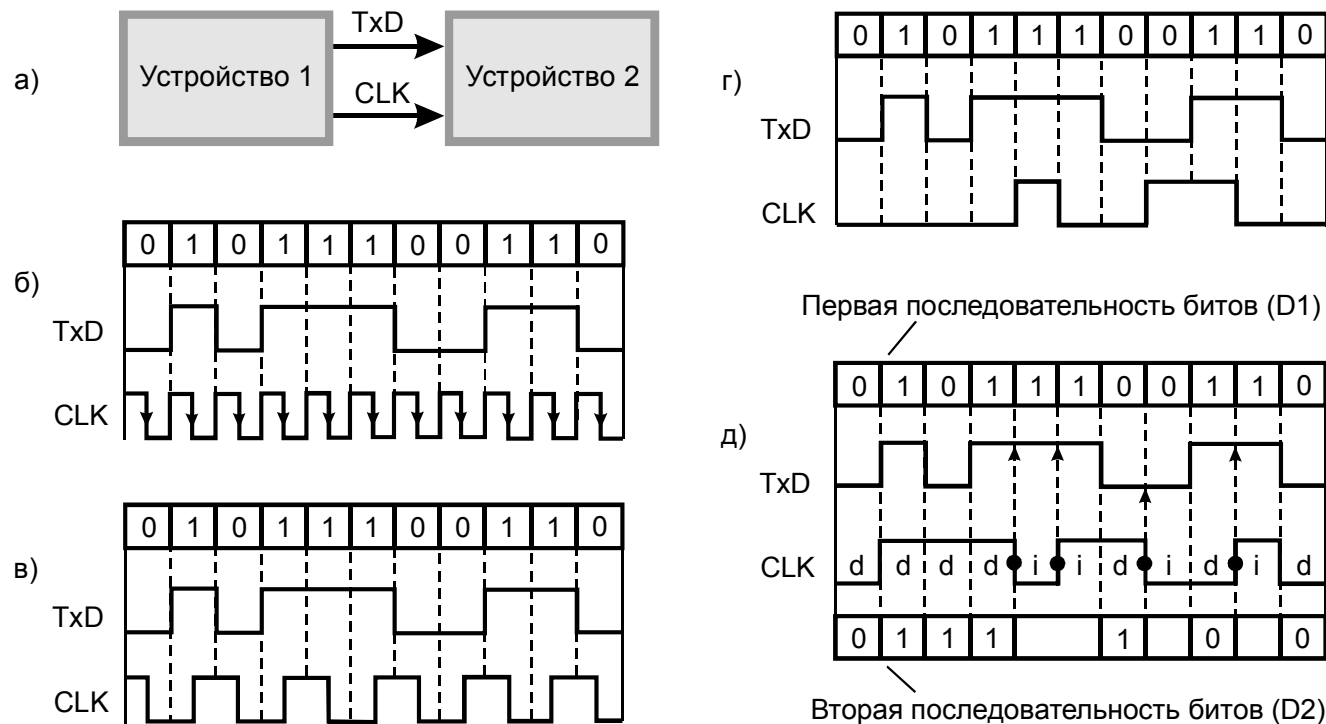


Рис. 2.79. Попутная передача данных и синхросигнала: а — схема; б — д — варианты временных диаграмм передачи данных

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

Недостатками такого способа передачи данных являются низкая скорость и высокий уровень помех, излучаемых на соседние провода многожильного кабеля, если он использован в составе канала связи. Первый недостаток связан с тем, что передача каждого бита данных сопровождается двумя (а не одним) изменениями состояния синхросигнала CLK. Второй недостаток: в спектре синхросигнала доминирует ярко выраженный энергетический максимум на основной частоте. Это приводит к тому, что из-за паразитных (ёмкостных и индуктивных) связей с соседними проводами многожильного кабеля в них формируется достаточно выраженный сигнал помехи [4].

### 2.15.2. Первый вариант усовершенствования

На диаграммах, приведенных на Рис. 2.79, в, для обозначения центров битовых интервалов используются как фронты, так и срезы синхросигнала CLK. Фронты и срезы синхросигнала CLK должны соответствовать установившимся значениям сигнала TxD. Точнее, должны гарантироваться достаточные интервалы времени предварительной установки и удержания сигнала TxD соответственно до и после изменения сигнала CLK.

Сравнивая временные диаграммы, приведенные на Рис. 2.79, *в* и *б*, можно заключить, что при одинаковой скорости передачи данных частота сигнала CLK уменьшена вдвое. Это позволяет увеличить длину линии связи. Можно привести и иную трактовку тех же нововведений: при неизменной частоте сигнала CLK скорость передачи данных удвоена.

Однако только что рассмотренный способ передачи данных (Рис. 2.79, *в*) также обладает отмеченным ранее недостатком — повышенным уровнем излучаемых помех при передаче периодического сигнала CLK.

### 2.15.3. Второй вариант усовершенствования

Этот недостаток устранён при передаче данных в соответствии с диаграммой на Рис. 2.79, *г*. В данном случае сигнал CLK, как и ранее, выполняет функцию разграничения битовых интервалов, но он изменяется только в тех ситуациях, при которых сигнал данных TxD остается неизменным. Иными словами, на любой границе любого битового интервала изменяется либо сигнал TxD, либо сигнал CLK. Приемник отслеживает изменения этих сигналов и поэтому в полной мере располагает информацией о местоположении границ между битовыми интервалами.

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

### 2.15.4. Третий вариант усовершенствования

На диаграммах, приведенных на Рис. 2.79, *д*, сигнал CLK по-прежнему в необходимых случаях разграничивает битовые интервалы, но одновременно с этим используется для передачи дополнительного потока данных. По линиям TxD и CLK передаются две независимые последовательности данных D1 и D2. Первая последовательность D1 (основная) скремблирована и передается с гарантированной скоростью, определяемой частотой следования битовых интервалов. Вторая последовательность D2 (дополнительная) может не скремблироваться и передается со средней скоростью, равной половине скорости передачи первой последовательности.

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

Поясним сказанное. В период, соответствующий первым четырем битовым интервалам, сигнал TxD изменяется, поэтому сигнал CLK можно использовать как “транспортное средство” для передачи первых четырех битов (dddd) = 0111 последовательности D2. Эти биты могут иметь произвольные значения.

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

Устройство 2 принимает сигналы из линий TxC и CLK, запоминает их и “прорисовывает” временные диаграммы, показанные на Рис. 2.79, *д*. Зная правила кодирования, по этим диаграммам устройство определяет, какие перепады сигнала CLK являются обязательными и со-

проводят передачу битов  $i$  синхронизации, а какие сопровождают передачу “полезных” битов  $d$ . Поэтому восстановление данных, принадлежащих последовательностям D1 и D2, не сопряжено с какими-либо трудностями.

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

Таким образом, в среднем каждый второй битовый интервал пригоден для передачи бита  $d$  данных второго канала D2. В результате суммарная скорость передачи данных между устройствами 1 и 2 (Рис. 2.79, *а*) увеличена в 1,5 раза по сравнению со способами, соответствующими Рис. 2.79, *в* и *г*. При этом сигналы TxD и CLK имеют “размытые” энергетические спектры, что способствует снижению уровня помех, излучаемых на соседние провода кабеля.

## 2.16. Вопросы и задачи к разделу 2.15

2.16.1. В системе передачи, показанной на Рис. 2.80, применена попутная передача данных и синхросигнала. Проведите сравнительный анализ приведенных на рисунке временных диаграмм по критериям скорости передачи, сложности реализации и уровням излучаемых помех.

Как вычисляется пропускная способность дополнительного канала передачи данных D2 (Рис. 2.80, д)?

Постройте диаграммы, аналогичные приведенным на Рис. 2.80, б — д, применительно к следующим последовательностям битов на линии TxD:

первая ситуация: 00000000000;

вторая ситуация: 11111111111;

третья ситуация: 01010101010;

четвёртая ситуация: 00110011001.

Литература: [1], гл. 3

Решение задачи приведено в разделе 3 на с. 177

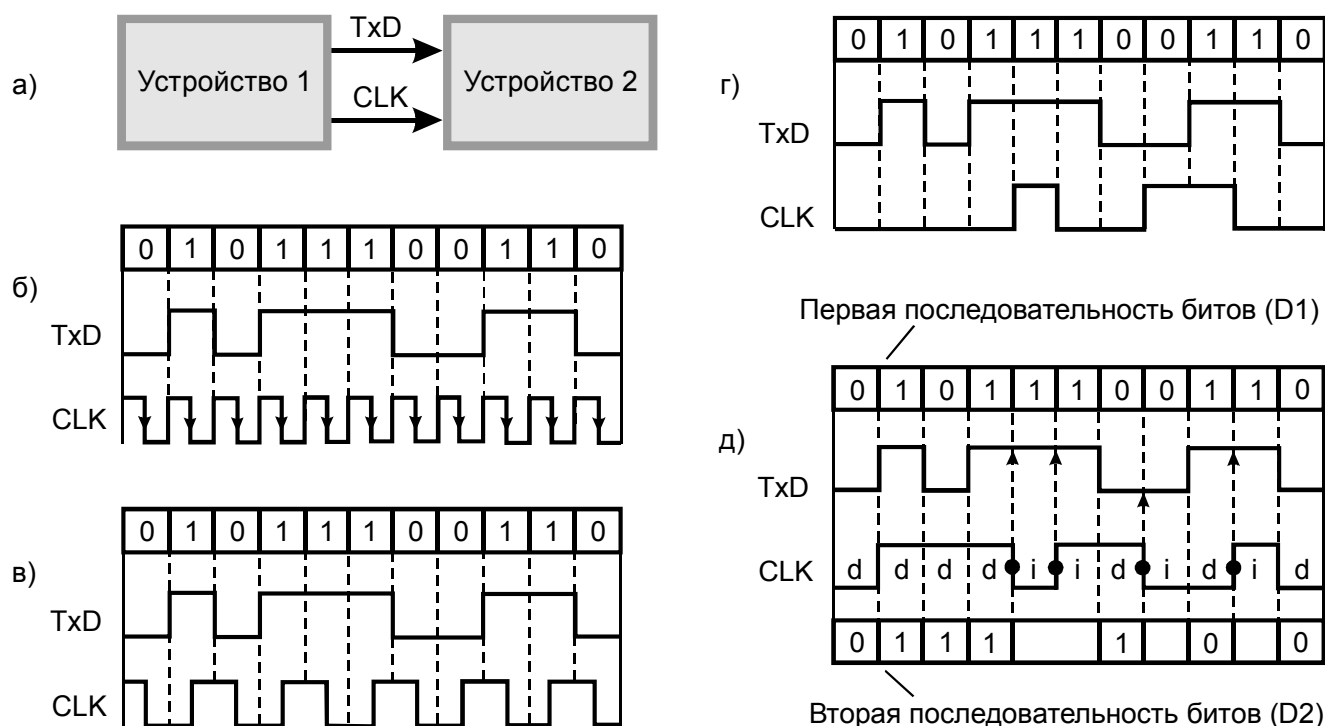


Рис. 2.80. Попутная передача данных и синхросигнала: а — схема; б — д — варианты временных диаграмм передачи данных

## 2.17. Структурирование потоков данных

Литература к разделу 2.17: [1], гл. 7; [2, 18]

### 2.17.1. Введение в раздел 2.17

Флаг начала каждого информационного кадра, передаваемого по каналу связи, представляет собой фиксированный код, который распознаётся приёмником и позволяет ему ориентироваться в непрерывном потоке битов, поступающем от передатчика. Флаг может быть уникальным или неуникальным. Уникальный флаг гарантирует отсутствие его ложных копий в теле кадра. Эти копии, если они первоначально существовали, преднамеренно и обратимо искажаются передатчиком перед их посылкой в канал связи. Для этого применяется вставка в такие копии “лишних” битов — битстаффинг. Использование неуникального флага также допустимо; при этом в потоке битов истинные флаги распознаются на фоне ложных в результате набора статистических данных. Это, однако, приводит к задержке установления кадровой синхронизации между передатчиком и приёмником. Далее, во-первых, предложено улучшить широко распространённую процедуру битстаффинга и, во-вторых, рассмотрено использование расширенного неуникального флага для ускоренного установления синхронизации между приёмником и передатчиком.

В синхронной дуплексной системе передачи данных (Рис. 2.81, а) телекоммуникационные устройства А и В одновременно посылают друг другу по каналу связи непрерывные потоки битов. Канал связи в данном примере выполнен в виде витой пары проводов кабеля городской телефонной сети; в качестве устройств А и В использованы модемы.

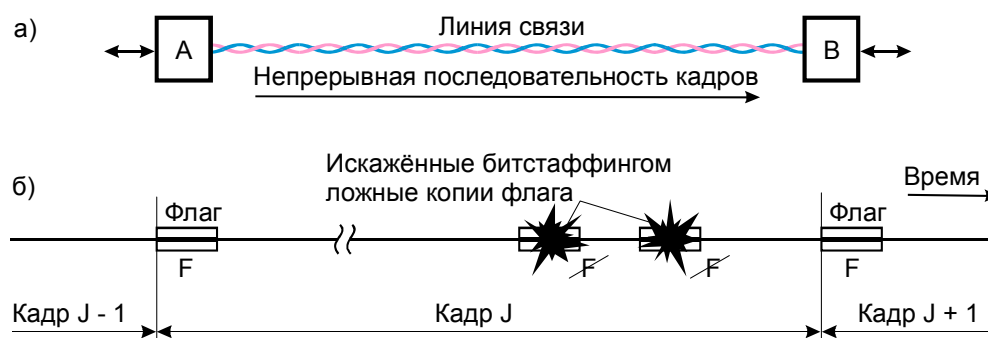


Рис. 2.81. Передача данных между устройствами А и В: а — схема; б — структура потока битов

Система передачи может быть симметричной или асимметричной. В любом случае пользовательские данные помещаются в кадры. Каждый кадр начинается с флага F — заранее заданной комбинации служебных битов, равной, например, 01111110. Окончание кадра также может помечаться флагом, но здесь этот вариант не рассматривается. Если данных нет, то по каналу связи передаются “пустые” кадры. В качестве таких вырожденных кадров может передаваться непрерывная последовательность не перекрывающихся или перекрывающихся флагов: ...011111100111111001111110011... или ...0111111011111110111111101111...

Далее для определённости предполагается, что данные передаются из устройства А (передатчика) в устройство В (приёмник).

Флаговые биты с точки зрения пользователя системы бесполезны и даже вредны, так как для их передачи необходимо расходовать часть пропускной способности канала связи. Поэтому в некоторых системах [1] длину флага уменьшают до одного бита и даже полностью исключают флаг в том виде, в котором он показан на Рис. 2.81, б. Но в таких системах время первоначального установления (или восстановления потерянной) кадровой синхронизации между устройствами А и В относительно велико. Для уменьшения этого времени длину флага увеличивают или (и) делают его “уникальным”.

Далее рассматриваются кадры с “нормальными” и расширенными флагами. В первой части раздела 2.17 предложен усовершенствованный битстаффинг — способ обеспечения уникальности флага; во второй части рассмотрено использование расширенного неуникального флага для ускоренного установления синхронизации между приемником и передатчиком.

### 2.17.2. Улучшение “классического” битстаффинга

Как уже отмечалось, уникальность флага гарантирует отсутствие в теле кадра его случайных копий. Эти копии могут первоначально существовать, так как передаваемые данные, как пользовательские, так и служебные, в общем случае произвольны. Для устранения этих копий они преднамеренно и обратимо искажаются применением битстаффинга (Рис. 2.81, б). Широкое распространение получил вариант битстаффинга, показанный в примере на Рис. 2.82.

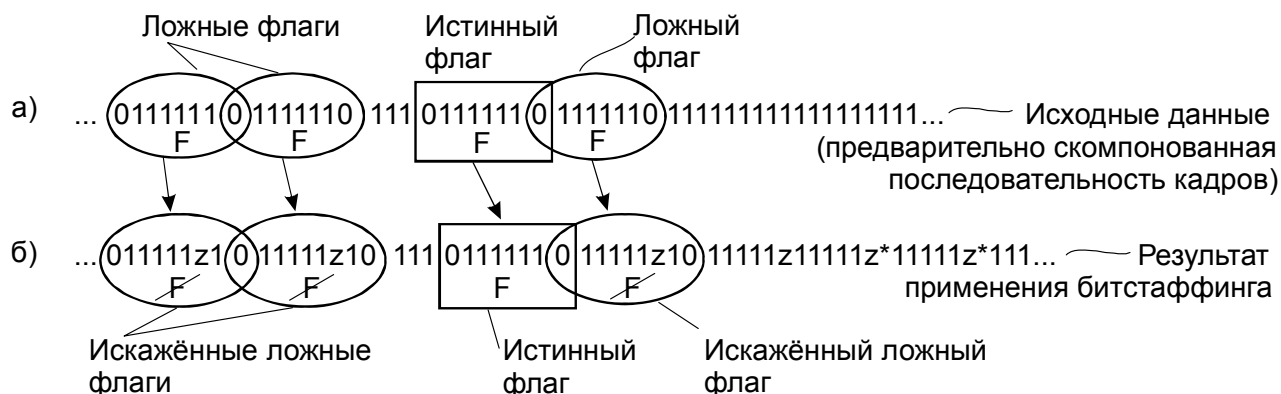


Рис. 2.82. Первый (традиционный) вариант битстаффинга

При формировании кадра в его начало помещается флаг, который выбирается равным 01111110 и в дальнейшем остаётся неизменным. Далее кадр или группа кадров анализируется с целью выявления “особых” кодов. Особый код (для краткости — код Q) представляет собой комбинацию битов, частично совпадающую с флаговой:  $Q = 011111$ . К найденным кодам Q передатчик приписывает справа “лишний” нуль (обозначаемый здесь и далее символом z) независимо от значения последующего бита. При этом образуются последовательности битов 011111z0 или 011111z1. К истинному флагу z-бит не добавляется, так как флаг не подвергается битстаффингу и поэтому становится уникальным — теперь он однозначно распознаётся на фоне полученного “прозрачного” массива битов.

Приёмник выявляет флаговые коды (01111110) и расценивает их как признаки начала кадров. Далее приёмник проводит операцию поиска и последующего исключения z-битов. Обнаружив коды Q и следующие за ними z-биты (они по определению являются “лишними”), приёмник вычёркивает их и тем самым восстанавливает исходную последовательность битов.

В кодовой ситуации, показанной на Рис. 2.82, а, помимо истинного флага в исходных данных легко просматриваются его ложные копии. Передатчик преднамеренно искажает их введением z-битов (Рис. 2.82, б), поэтому копии не воспринимаются приёмником как флаги, что и требуется.

Биты  $z^* = 0$  могут вводиться или не вводиться в последовательность единиц в зависимости от принятого протокола обмена. В первом случае предполагается, что в качестве нуля в коде Q может присутствовать как “обычный” лог. 0, принадлежащий исходным данным, так и z-бит. Во втором случае принимается соглашение о том, что z-бит не может входить в код Q. Возможны и иные договорённости, предусмотренные конкретным протоколом обмена. Например, если вслед за кодом Q приёмник обнаруживает две лог. 1, (комбинация 01111111), то он анализирует следующие биты, в которых могут кодироваться какие-либо команды или признаки (например, команда аварийного завершения работы).

Однако такой способ кодирования неэффективен, так как требует большого числа избыточных битов. Это связано, во-первых, с тем, что код Q имеет разрядность, равную шести (а не семи, как предлагается далее) и встречается в случайных данных достаточно часто. Во-вторых,

появляется необходимость разделения цепочек из лог. 1 в пользовательских данных битами  $z^*$  на пятиразрядные группы (что является излишним в предлагаемом далее решении).

Если исходные данные в теле кадра рассматривать как случайные, то вероятность обнаружения шестиразрядного кода  $Q$  в любой случайно выбранной группе из шести битов равна  $2^{-6} = 1/64$ . Иными словами, при последовательном просмотре исходных данных код  $Q$  будет обнаруживаться передатчиком в среднем один раз в каждой цепочке из 64 битов. Как было показано, обнаружение кода  $Q$  сопровождается введением в тело кадра одного  $z$ -бита. Если длина кадра составляет, например, 12 тыс. бит (такую длину может иметь кадр локальной сети Ethernet), то среднее число  $z$ -битов, которое потребуется в него ввести, составит  $12000/64 \approx 187$ . Нельзя ли уменьшить это число?

### 2.17.3. Анализ флаговых кодов

Для решения этой задачи увеличим длину кода  $Q$  на один разряд:  $Q = 0111111$ , сохранив флаг неизменным ( $01111110$ ). По аналогии с избыточным битом  $z = 0$  используем бит  $s = 1$ , который добавляется к каждому найденному коду  $Q$  (кроме входящего в состав истинного флага), как показано на Рис. 2.83. Флаг при этом становится уникальным и однозначно распознаётся приёмником. Приёмник отыскивает все коды  $Q$  (кроме входящего в состав истинного флага) и без проверки каких-либо дополнительных условий вычёркивает примыкающие к ним справа биты  $s$ . Таким образом, внесённые искажения исправляются, и приёмник восстанавливает правильные исходные данные.

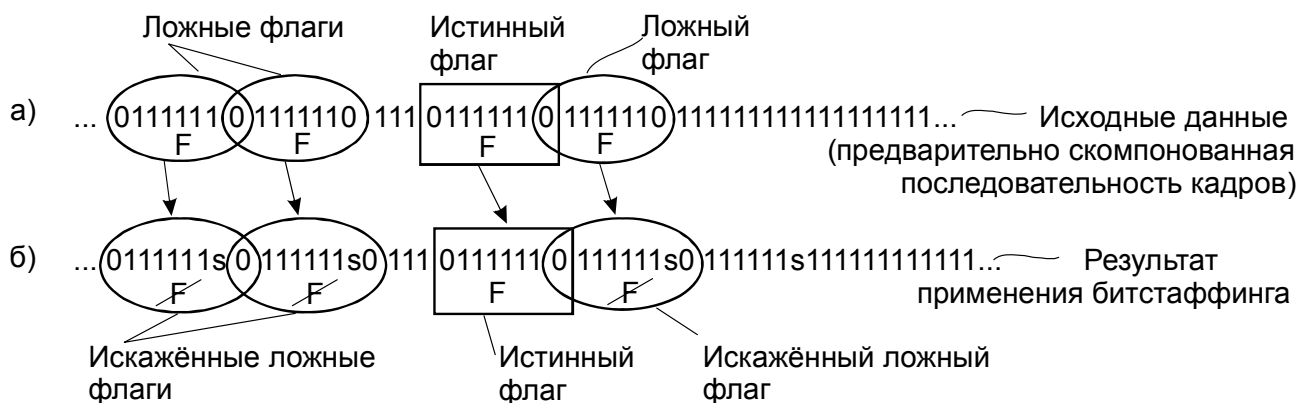


Рис. 2.83. Второй (предлагаемый) вариант битстаффинга

При таком варианте битстаффинга вероятность обнаружения кода  $Q$  в произвольной семиразрядной группе битов из массива случайных данных составляет  $2^{-7} = 1/128$ . Предполагая, что, как и в предыдущем примере, длина кадра составляет 12 тыс. бит, получим, что среднее число избыточных битов, которое требуется ввести в кадр, равно  $12000/128 \approx 94$ , т. е. уменьшено в два раза по сравнению с традиционным решением.

Экономия избыточных битов, как видим, весьма существенная. Чтобы обеспечить возможность оперативного ввода в тело кадра каких-либо команд или признаков (например, команды аварийного завершения работы), можно предложить разные способы. Например, можно предусмотреть в протоколе обмена дополнительную вставку в тело кадра двух флагов с общим нулевым битом ( $011111101111110$ ), вслед за которой передаётся байт команды. Эти флаги не подвергаются битстаффингу и поэтому легко распознаются приёмником. При этом пустые кадры можно обозначать не перекрывающимися флагами:

...011111100111111001111110011...

Теперь рассмотрим вопрос о выборе флагового кода. Можно ли в качестве флага использовать коды, отличные от традиционного ( $01111110$ )? Можно ли использовать все возможные коды или только часть из них?

Прежде чем ответить на эти вопросы покажем, что существуют, по крайней мере, “очень плохие”, “плохие” и “хорошие” флаговые коды.

Приведём пример “очень плохого” (неприемлемого) флагового кода: 00000001. Ему соответствует код  $Q = 0000000$  и нулевой избыточный бит (z). Предположим, что исходно в теле кадра имеется последовательность битов, совпадающая с флаговой: ...00000001... . После первого шага процедуры битстаффинга получим последовательность ...0000000z1... . В ней также есть совпадение с флагом, поэтому после второго шага получим ...0000000zz1, после третьего — 0000000zzz1, затем 0000000zzzz1 и т. д. Вставка избыточных битов, таким образом, будет длиться бесконечно, при этом избыточные биты следуют с шагом, равным единице.

Этот результат (шаг, равный единице) отражен цифрой 1 (обведённой жирной прямоугольной рамкой) в верхней строке табл. 1. Эта таблица отражает качество всех возможных флаговых кодов в восьмибалльной системе оценок. Строки и столбцы таблицы пронумерованы от 0 до  $F_{16}$  и соответствуют старшей и младшей четвёркам разрядов флагового кода.

Флаговый код  $11111110 = FE_{16}$ , обратный по отношению к только что рассмотренному ( $01_{16}$ ), столь же непригоден для использования. Ему соответствует код  $Q = 1111111$  и единичный избыточный бит (s). Оценка этого кода, как показано в таблице, также равна единице.

Таблица 1. Оценки качества флаговых кодов

|   | 0 | 1        | 2 | 3 | 4        | 5 | 6 | 7 | 8 | 9 | A | B        | C | D | E        | F |
|---|---|----------|---|---|----------|---|---|---|---|---|---|----------|---|---|----------|---|
| 0 | 8 | <b>1</b> | 8 | 7 | 8        | 6 | 8 | 7 | 8 | 5 | 8 | 7        | 8 | 6 | 8        | 7 |
| 1 | 4 | 5        | 8 | 7 | 8        | 6 | 8 | 7 | 8 | 5 | 8 | 7        | 8 | 6 | 8        | 7 |
| 2 | 5 | 6        | 8 | 4 | 8        | 3 | 8 | 7 | 5 | 6 | 8 | 7        | 8 | 6 | 8        | 7 |
| 3 | 5 | 6        | 4 | 7 | 8        | 6 | 8 | 7 | 5 | 6 | 8 | 7        | 8 | 6 | 8        | 7 |
| 4 | 6 | 7        | 8 | 5 | 6        | 4 | 8 | 7 | 3 | 7 | 8 | 5        | 6 | 7 | 8        | 7 |
| 5 | 6 | 7        | 8 | 5 | <b>2</b> | 7 | 8 | 7 | 6 | 7 | 8 | 5        | 6 | 7 | 8        | 7 |
| 6 | 6 | 7        | 5 | 7 | 6        | 7 | 8 | 4 | 6 | 7 | 5 | 7        | 3 | 7 | 8        | 7 |
| 7 | 6 | 7        | 5 | 7 | 6        | 7 | 4 | 7 | 6 | 7 | 5 | 7        | 6 | 7 | <b>8</b> | 7 |
| 8 | 7 | 8        | 7 | 6 | 7        | 5 | 7 | 6 | 7 | 4 | 7 | 6        | 7 | 5 | 7        | 6 |
| 9 | 7 | 8        | 7 | 3 | 7        | 5 | 7 | 6 | 4 | 8 | 7 | <b>2</b> | 5 | 8 | 7        | 6 |
| A | 7 | 8        | 7 | 6 | 5        | 8 | 7 | 6 | 7 | 8 | 7 | <b>2</b> | 5 | 8 | 7        | 6 |
| B | 7 | 8        | 7 | 6 | 5        | 8 | 7 | 3 | 7 | 8 | 4 | 6        | 5 | 8 | 7        | 6 |
| C | 7 | 8        | 6 | 8 | 7        | 8 | 6 | 5 | 7 | 8 | 6 | 8        | 7 | 4 | 6        | 5 |
| D | 7 | 8        | 6 | 8 | 7        | 8 | 6 | 5 | 7 | 8 | 3 | 8        | 4 | 8 | 6        | 5 |
| E | 7 | 8        | 6 | 8 | 7        | 8 | 5 | 8 | 7 | 8 | 6 | 8        | 7 | 8 | 5        | 4 |
| F | 7 | 8        | 6 | 8 | 7        | 8 | 5 | 8 | 7 | 8 | 6 | 8        | 7 | 8 | <b>1</b> | 8 |

$0F_{16} = 0000\ 1111_2$

$7F_{16} = 0111\ 1111_2$

$FF_{16} = 1111\ 1111_2$

“Плохому” флаговому коду  $54_{16}$  (оценка — два) соответствует код  $Q = 0101010$  и единичный избыточный бит (s). При неблагоприятной исходной последовательности ...0101010000000000... после первого шага процедуры битстаффинга получим ...0101010s00000000..., после второго — ...0101010s0s0000000..., после третьего — ...0101010s0s0s000000... и т. д. Как видим, процесс не бесконечен, избыточные биты следуют с шагом два, но их общее число, конечно, неоправданно велико. То же относится и к обратному флаговому коду  $AB_{16}$  — его оценка также соответствует двум баллам.

При использовании рассмотренного ранее флагового кода  $7E_{16}$  неблагоприятная последовательность данных ...0111111011111101111110... преобразуется в последовательность 0111111s0111111s0111111s0, в которой избыточные биты следуют с шагом, равным восьми. Оценка “восемь” соответствует “хорошему” флаговому коду.

В качестве “хорошего” флагового кода можно также использовать другие коды, например, код  $00_{16}$ . Неблагоприятная последовательность ...0000000000000000... преобразуется в последовательность ...0000000s0000000s0000000s..., поэтому код  $00_{16}$  также получает оценку “восемь”.

Из табл. 1 следует, что из 256 возможных флаговых кодов можно использовать только 74 кода, имеющие оценку “восемь”. Аналогичный анализ можно провести применительно к флаговым кодам большей или меньшей разрядности.

#### 2.17.4. Использование расширенного неуникального флага для ускоренного установления синхронизации между приемником и передатчиком

Как было показано, битстаффинг в большей или меньшей мере сопровождается передачей по каналу связи избыточных битов. Применение “классического” битстаффинга в неблагоприятной кодовой ситуации, при которой исходные данные представлены кодом “все единицы”, приводит к тому, что за каждой пятеркой единичных битов следует один избыточный — нулевой. Передача 1000 “полезных” (единичных) битов сопровождается передачей 200 служебных (нулевых). Снижение пропускной способности канала связи на величину, достигающую 20 %, нежелательно, поэтому во многих системах битстаффинг не применяется, следовательно, допускается совпадение флагового кода с кодами в теле кадра.

Если флаговый код не уникален, то приёмник должен обладать достаточным интеллектом для того, чтобы отличать истинные флаги от ложных. Если кадровая синхронизация приёмника с передатчиком уже достигнута, то она поддерживается благодаря тому, что истинный флаг следует во времени с определенным периодом, зависящим от длины кадра (постоянной или переменной, но заданной в самом кадре).

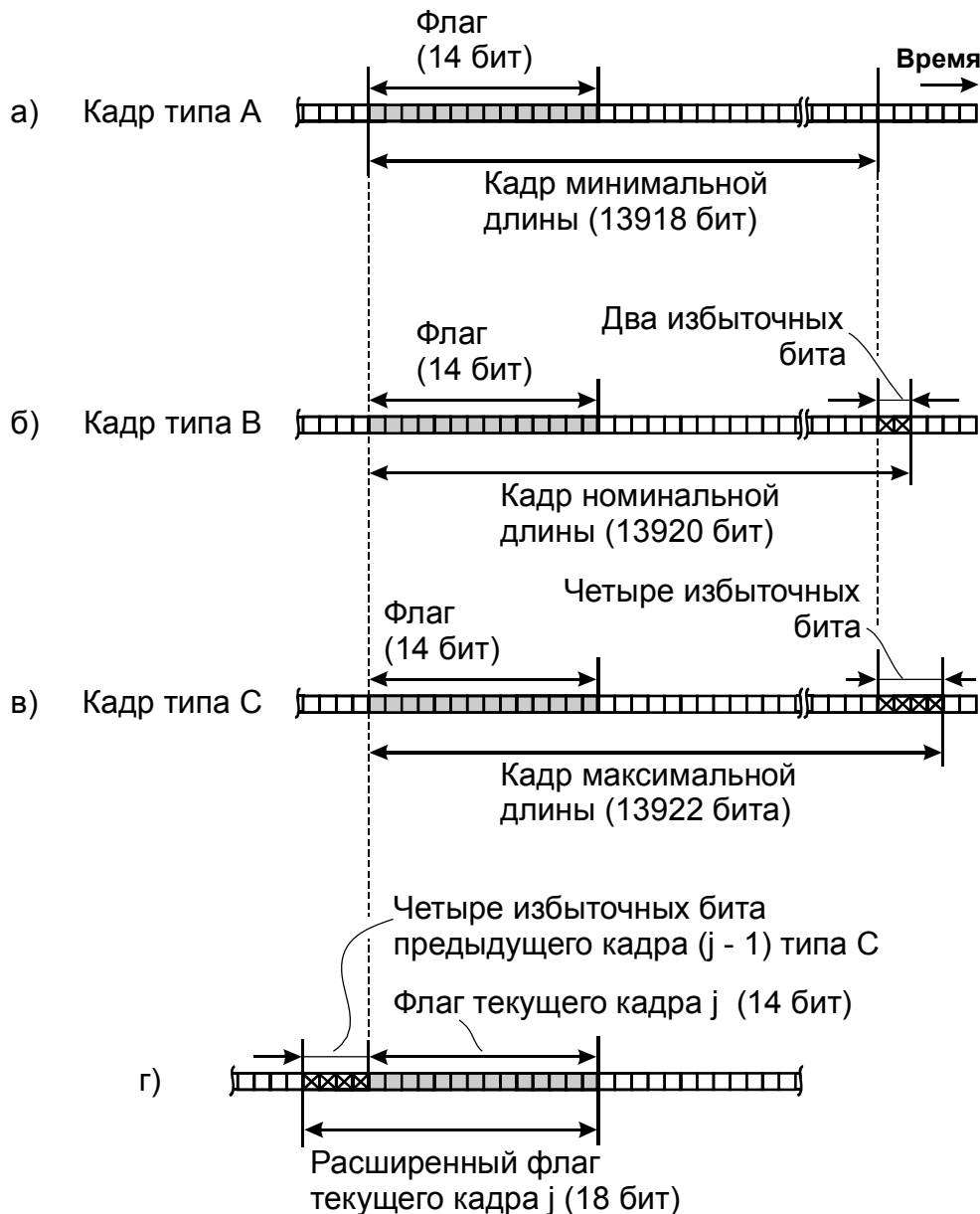
Если кадровая синхронизация приёмника с передатчиком еще не достигнута (или потеряна), то каждый код, встречающийся во входном потоке данных и совпадающий с флаговым, рассматривается приемником в качестве кандидата на истинный флаг начала кадра. Чтобы убедиться в правильности предположения об истинности флага, приемник проверяет своевременность его получения с учетом предыстории поиска, определяет длину очередного кадра, прогнозирует момент получения следующего флага и т. д. Чем больше ложных флагов в потоке данных, тем более сложен поиск среди них истинного флага, так как нужно одновременно или последовательно отслеживать историю многих событий обнаружения искомых кодов. Поэтому для упрощения и ускорения работы средств распознавания истинных флагов на фоне ложных желательно уменьшить число случайных флаговых кодов в потоке данных.

Для такого уменьшения следует увеличить разрядность флагового кода. Действительно, чем больше его разрядность, тем более вероятно, что обнаружен именно истинный (а не ложный) флаг начала кадра. Так, если поток битов имеет случайный характер, то любой заранее заданный восьмиразрядный код будет встречаться в нем в среднем через каждые  $2^8 = 256$  бит = 0,25 кбит. При длине кадра, равной 10 кбит, в нем, помимо истинного, будет обнаружено около  $10/0,25 = 40$  ложных флагов. Если при той же длине кадра увеличить разрядность флага до 20 бит, то его случайные копии будут встречаться в потоке данных со средним периодом  $2^{20} = 1$  Мбит. Один ложный флаг будет в среднем обнаруживаться в группе из  $2^{20}/10 \times 2^{10} = 0,1 \times 2^{10} = 102,4$  кадров. Иными словами, вероятность того, что с первой попытки поиска обнаружен истинный (а не ложный) флаг, достаточно высока и составляет  $1 - 1/102,4 \approx 0,99$ .

Флаг повышенной разрядности, таким образом, позволяет быстро синхронизировать приемник с передатчиком на уровне распознавания границ между кадрами. После установления синхронизации разрядность искомого флага может быть уменьшена, так как в подтверждение правильности достигнутой синхронизации наблюдается ряд закономерностей: вновь прибывающие флаги обнаруживаются на предсказанных временных позициях, структура принятых кадров соответствует протоколу, контрольные суммы полученных кадров в подавляющем большинстве случаев правильные, так как интенсивность ошибок в линии невелика, и т. д.

Рассмотрим идею использования флага переменной длины [18]. В стандарте SHDSL (Single-pair High-speed Digital Subscriber Line — высокоскоростная цифровая абонентская линия на основе одной витой пары медных проводов) предусмотрена передача кадров, которые могут не содержать или содержать два или четыре избыточных бита (Рис. 2.84, а — в). Если приемник и передатчик образуют систему “ведущий — ведомый”, т. е. синхронизированы от одного и того же генератора импульсов, то для передачи данных используются только кадры типа В, каждый из которых содержит два избыточных бита. Эти биты вплотную примыкают к флагу сле-

дующего кадра. Поэтому, зная значения избыточных битов, можно рассматривать их как двухразрядное расширение 14-разрядного флага. Такое расширение в среднем в четыре раза уменьшает число ложных флагов, содержащихся в теле кадра, что ускоряет вхождение приемника в кадровую синхронизацию с передатчиком.



**Рис. 2.84. Передача данных с использованием кадров: а — не содержащих избыточных битов; б, в — с двумя и четырьмя избыточными битами; г — содержащих флаг, расширенный избыточными битами**

Если приемник и передатчик синхронизируются от внутренних генераторов, имеющих почти одинаковые частоты (“плездохронная” система передачи данных), то для предотвращения проскальзываний используются кадры с переменным числом избыточных битов [1].

Стандарт SHDSL применительно к плездохронному режиму предусматривает передачу двух типов кадров: А и С. Кадр типа А не содержит избыточных битов, кадр типа С имеет четыре избыточных бита. Если частоты генераторов строго одинаковы (что возможно лишь теоретически), то среди передаваемых кадров 50% имеют тип А и 50% — тип С. При незначительном различии частот генераторов передатчика и приемника среди передаваемых кадров наблюдается столь же незначительное преобладание кадров типа А или С в зависимости от знака разности этих частот.

Как показано на Рис. 2.84, г, в плездохронном режиме избыточные биты кадра типа С (значения которых постоянны и заранее известны) вплотную примыкают к флагу следующего

за ним кадра (типа А или С). В среднем около 50% флагов состыкованы с избыточными битами предыдущего кадра, которые могут рассматриваться в качестве битов расширения флага. Таким образом, на этапе установления синхронизации может осуществляться поиск 18-разрядного (а не 14-разрядного) флага, что способствует его более быстрому обнаружению.

### 2.17.5. Распознавание и восстановление искаженных кадров при передаче данных по радиоканалу

В системе мобильной телефонной связи [1] (Рис. 2.85) предусмотрена одновременная передача “оцифрованных” речевых и иных данных по радиоканалу. Данные, не относящиеся к передаче речи, пересылаются в паузах между словами и фразами. Дальнейшее описание относится именно к таким данным.



Рис. 2.85. 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. 2.86).

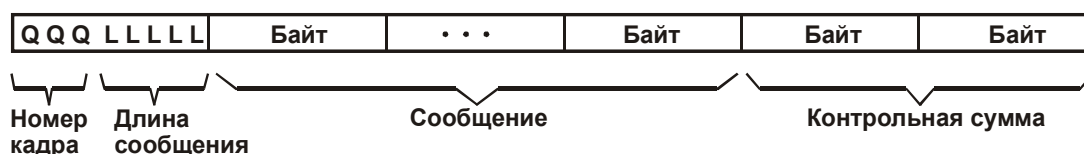


Рис. 2.86. Структура кадра до его выдачи в радиоканал

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 послышки. При этом каждый байт преобразуется в две послышки, как показано на Рис. 2.87.

Преобразование состоит в следующем. Сначала байт D1 ... D8 делится на два полубайта: D1 ... D4 и D5 ... D8. Им присваиваются четырехразрядные порядковые номера: N1 ... N4 и M1 ... 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.

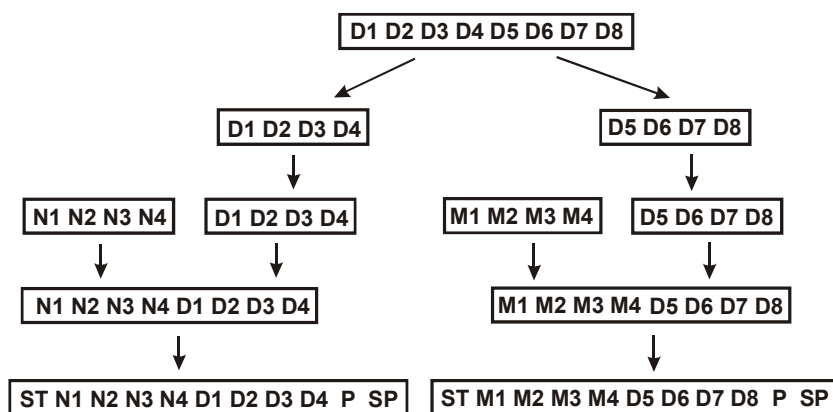


Рис. 2.87. Преобразование байта в две start-стоповые послылки

В результате преобразования исходный кадр дробится на множество посылок (их число равно удвоенному числу байтов кадра). Они выдаются в радиоканал в паузах между “всплесками” передачи речевой информации. Поэтому поток данных неравномерен и может содержать как следующие друг за другом вплотную, так и разрозненные группы посылок или отдельные послылки. В процессе передачи по радиоканалу часть посылок может быть потеряна из-за возможных замираний сигнала или действия помех. Приемник может также получать ложные послылки, которые вклинились в неравномерный поток правильных. Чтобы обеспечить надежный обмен данными в условиях замираний сигнала и действия помех, предлагаются три простые, но примечательные идеи.

*Идея 1. Проверка правильности порядка следования посылок слежением за динамикой изменения номеров полубайтов.* Как было показано, каждая послылка 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 шестнадцатеричная форма записи чисел). Если получена последовательность ...0, 1, 2, 7, 3, 4, 5, 6, ..., то можно сделать вывод о том, что послылка 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 послылка потеряна по пути к приемнику. Поэтому передача ошибочного кадра повторяется в связи с неполучением подтверждения правильного приема данных.

*Идея 2. Расширение диапазона нумерации посылок при ограниченной разрядности номеров.* С помощью четырехразрядного двоичного кода можно отображать числа в диапазоне 0 —  $F_{16}$ . Поэтому на первый взгляд представляется естественной простая последовательная нумерация посылок: ... 0, 1, 2, ..., 9, A, B, C, D, E, F, 0, 1, 2, ... Однако такая нумерация повторяется с периодом, равным 16, а кадр может быть представлен десятками и сотнями посылок. Действительно, можно построить достаточно большой кадр, если принять условие, что длина сообщения (см. код LLLLL на Рис. 2.86) 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 послылки 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, за-

тем 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. 2.87. 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. 2.86). Если контрольная сумма правильная, то приемник посылает уведомление об этом источнику кадра. (Напомним, что в качестве уведомления используется кадр с нулевым кодом LLLL 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 поступления новой копии и т. д.

При наличии нескольких копий кадра применяется обычный метод мажоритарного выбора, при котором используется “голосование по большинству”. Например если два одноименных байта из трех совпадают, то их значение принимается в качестве истинного. Если все три байта разные, то голосование переносится на уровень битов. Это позволяет в некоторых ситуациях, когда одиночные ошибки распределены по разным битам, восстановить правильный байт. Но если, например, из пяти одноименных байтов два совпадают между собой, но отличаются от трех других, которые, в свою очередь, не совпадают друг с другом в любых сочетаниях, то эти совпадающие байты принимаются в качестве истинных, хотя, строго говоря, они — в меньшинстве. Конечно, окончательное решение об отсутствии ошибок в кадре принимается после проверки его контрольной суммы.

### 2.17.6. Структура ячейки АТМ

Технология АТМ (Asynchronous Transfer Mode — асинхронный режим передачи) предусматривает размещение данных внутри 53-байтовых ячеек (Рис. 2.88). Как показано на рисунке, ячейки следуют по линии непрерывным потоком, хотя возможны и иные варианты их транспортирования.



Рис. 2.88. 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?

### 2.17.7. 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 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. 2.88, 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 в данном случае 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.

### 2.17.8. Структура кадра V.110

Кадр V.110 (Рис. 2.89, *a*) 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 V63 нижней строки. Биты B1 — B63 представляют собой передаваемые данные; S1 — S8 — биты синхронизации.

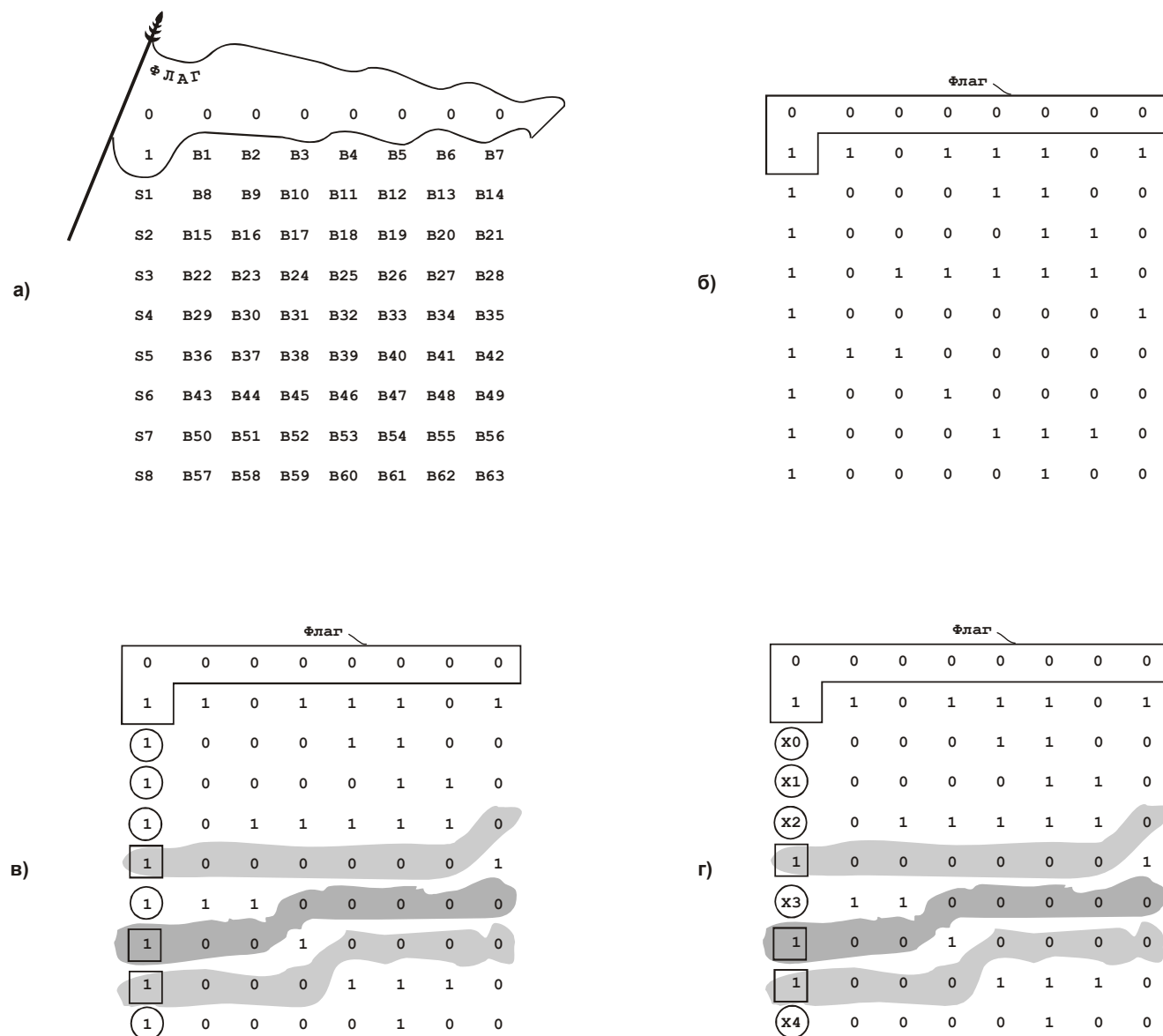
В начале кадра всегда содержится флаг — кодовая комбинация из восьми лог. 0 и последующей лог. 1 (000000001). Такая комбинация не должна встречаться в любом другом месте кадра; это гарантируется введением битов S1 — S8 синхронизации, каждый из которых представлен лог. 1. Из приведенного на Рис. 2.89, *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 B1 — B63 представлены лог. 0 либо среди них имеется незначительное число лог. 1. Нельзя ли уменьшить избыточность левого столбца?

### 2.17.9. Усовершенствование структуры кадра

Обратимся к Рис. 2.89, *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 S1, выделив его на рисунке кружочком.

Подобные рассуждения применимы также к битам S2, S3, S5 и S8, 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 S4, S6 и S7 (выделены квадратиками) должны безусловно принимать значения лог. 1. В противном случае сформируются достаточно длинные последовательности из лог. 0 (выделены тремя затененными областями), которые, например, при  $S4 = S6 = S7 = 0$  приведут к троекратному ложному опознанию флага начала кадра в его теле.



**Рис. 2.89. 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 квадратами); г — определение позиций для передачи данных (X0 — X4) вместо битов синхронизации (выделены кружочками)**

Таким образом, в передаваемом кадре V.110 обычно можно отыскать вакантные места для размещения от одного до восьми битов — своего рода “неучтенных пассажиров” транспортной системы. В нашем примере имеем пять таких битов X0 — X4 (Рис. 2.89, г).

Теперь возникает вопрос: как удаленный приемник сможет понять, какие биты полученного кадра обведены кружочками, а какие — квадратами? Ведь информация о форме обвода в явном виде не передается по линии! Попробуем с этим разобраться.

Алгоритм работы приемника таков.

1. Приемник, постоянно прослушивая линию, обнаруживает поступление флага (кода 000000001), после чего принимает кадр (см. Рис. 2.89, а) и записывает его в первой области памяти, выделенной для временного хранения кадров.

2. Приемник запоминает биты S1 — S8 во второй области памяти, выделенной для их временного хранения, а на места этих битов в матрице помещает лог. 1. Таким образом, приемник “видит” матрицу, показанную на Рис. 2.89, б.

3. Приемник проводит описанный ранее анализ необходимости существования каждого бита синхронизации и, если не было ошибок передачи данных B1 — B63, приходит к тем же

результатам, которые были приведены на Рис. 2.89, в. В данном примере приемник приходит к выводу, что биты S4, S6 и S7 должны быть представлены лог. 1.

Он проверяет, так ли это, извлекая из второй области памяти соответствующие биты. (Любое несовпадение расценивается как ошибка передачи, и кадр отбрасывается.) Попутно выявляются позиции битов, обведенных кружочками. Теперь остается только извлечь из второй области памяти соответствующие биты (S1 — S3, S5, S8) и скомпоновать из них искомый код X0 ... X4 (см. Рис. 2.89, г). Таким образом, “неучтенные пассажиры” благополучно прибыли на место.

Можно предложить много вариантов использования кодов X0 ... Xi. Например:

1) коды X0 ... Xi переменной длины, передаваемые в последовательности кадров, можно рассматривать как фрагменты некоторого, в общем случае независимого, массива данных. Иными словами, можно организовать второй, параллельный канал передачи данных, который работает не в ущерб первому, основному;

2) каждый код X0 ... Xi можно рассматривать как некое контролирующее приложение к данным B1 — B63, которые передаются в этом же кадре. Например, разряд X0 может представлять собой бит контроля по четности всей группы B1 — B63. Разряды X1 и X2 могут соответственно служить битами контроля по четности подгрупп B1 — B31 и B32 — B63 и т. д. с уменьшением длины контролируемой области.

Иное возможное распределение функций разрядов: бит Xi контролирует по четности данные, размещенные в своей и предыдущей строках матрицы. Такое решение позволяет обнаруживать ошибки “на лету”, не дожидаясь окончания приема полного кадра.

## 2.18. Вопросы и задачи к разделу 2.17

2.18.1. На Рис. 2.90 показана структура кадра, в котором предусмотрен трёхразрядный код его номера (QQQ).

1) Как с помощью этого кода пронумеровать, например, 100 таких кадров, следующих друг за другом?

2) Можно ли обойтись одним разрядом (Q) и пронумеровать 1000 кадров?

Литература: [1], гл. 7

Решение задачи приведено в разделе 3 на с. 179

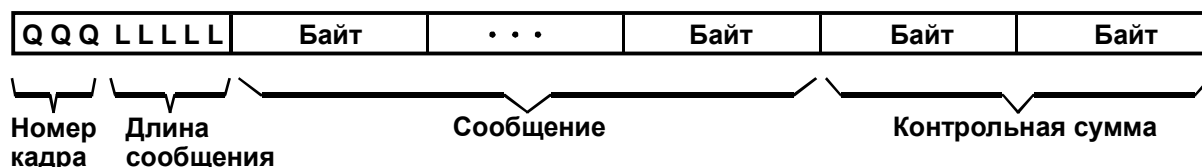


Рис. 2.90. Структура кадра

2.18.2. Примените битстаффинг к следующей последовательности пользовательских данных для исключения из неё копий флагового кода 01111110:

...011111111111011111010101101111100011111111111000000111111...

Литература: [2]

Решение задачи приведено в разделе 3 на с. 180

2.18.3. Может ли последовательность байтов 8В, А8, 78, 3D, 1В рассматриваться как заголовок АТМ-ячейки?

Литература: [1], гл. 7

Решение задачи приведено в разделе 3 на с. 180

2.18.4. Какова вероятность обнаружения предполагаемого заголовка АТМ-ячейки в потоке случайных данных при однократной выборке из него цепи из 40 бит?

Литература: [1], гл. 7

Решение задачи приведено в разделе 3 на с. 181

2.18.5. В системе передачи, показанной на Рис. 2.91, непрерывный поток случайных данных разделён на кадры одинаковой длины, равной  $N$  бит. Каждый кадр содержит флаг  $F$  длиной  $M$  бит. Битстаффинг не применяется, поэтому в теле кадра могут содержаться ложные флаги.

Оцените вероятность того, что после включения ранее выключенного приёмника первый же обнаруженный им флаг будет истинным, а не ложным. Рассмотрите две ситуации:

первая ситуация:  $M = 10$ ;  $N = 10^4$  бит;

вторая ситуация:  $M = 30$ ;  $N = 10^4$  бит.

Литература: [2]

Решение задачи приведено в разделе 3 на с. 181

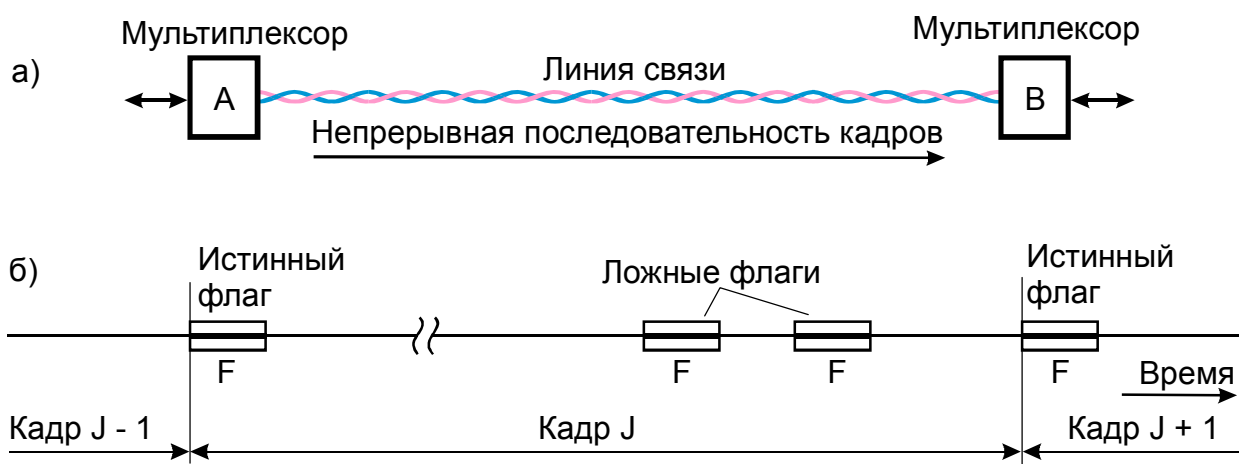


Рис. 2.91. Система передачи данных: а — схема; б — структура передаваемых кадров

2.18.6. Кадр ITU-T V.110 (Рис. 2.92) содержит флаг, группу битов синхронизации и поле данных.

Какие позиции битов синхронизации в данной кодовой ситуации избыточны и могут использоваться для передачи данных?

Литература: [1], гл. 7

Решение задачи приведено в разделе 3 на с. 181

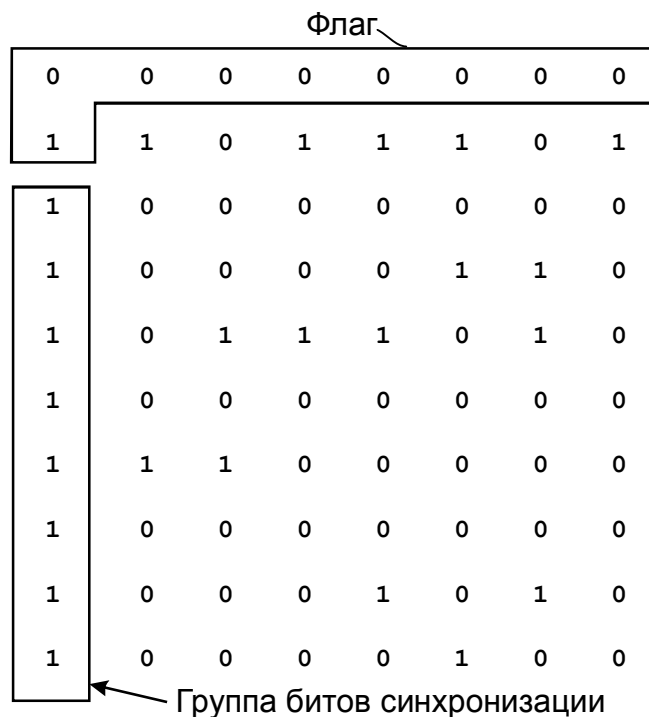


Рис. 2.92. Кадр ITU-T V.110

## 2.19. Программное управление потоком данных

Литература к разделу 2.19: [3]

### 2.19.1. Введение в раздел 2.19

Как решить задачу распознавания “истинных” служебных символов на фоне их случайных копий в потоке пользовательских данных? Запретить пользователю применять коды, совпадающие со служебными символами? Такое решение в общем случае неприемлемо, так как ограничивает “свободу кодирования” пользовательских данных. Просто уничтожать эти символы в массиве пользовательских данных? Тоже нельзя, ведь они несут полезную информацию. “Прятать” эти символы при передаче, а на противоположной стороне, при приёме, отыскивать их? Именно в этом и состоит рассмотренная далее идея решения задачи (Рис. 2.93).

Как показано на Рис. 2.93, а, компьютер и модем по мере необходимости вставляют в выходные потоки данных символы XON и XOFF для приостановки-возобновления входных потоков. Схема симметрична, поэтому для определённости рассмотрим передачу данных по линии TxD от компьютера к модему.

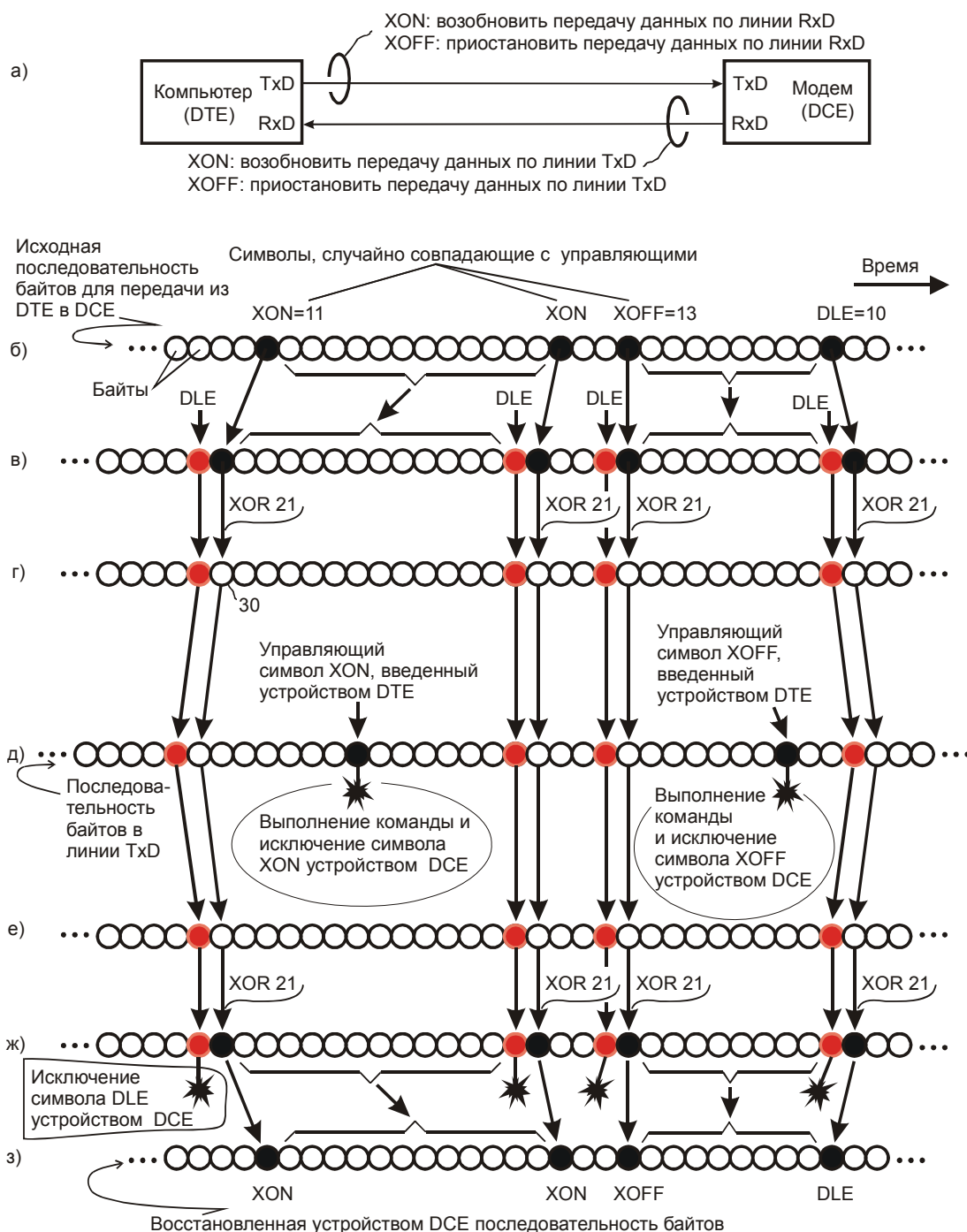
Предположим, что в памяти компьютера содержится предназначенная для передачи в модем последовательность байтов, каждый из которых соответствует одному из символов ASCII (Рис. 2.93, б). Байты (символы) обозначены на рисунке кружками, причём зачернённые кружки соответствуют символам, случайно совпадающим с символами XON, XOFF или DLE (Data Link Escape),  $DLE = 10_{16} = 0001.0000_2$  в коде ASCII. Если переслать такую последовательность символов в модем, то он воспримет символы XON и XOFF как команды управления, хотя они таковыми не являются, что приведёт к неправильной работе системы передачи данных. Поэтому далее решается задача маскировки “особых” символов (XON, XOFF и DLE) в исходной последовательности.

### 2.19.2. Байтстаффинг как средство обеспечения прозрачности потока данных

Как показано на Рис. 2.93, в, перед каждым особым символом в разрывы исходной последовательности вставляются символы DLE, выполняющие роль своеобразных флагов. Далее все особые символы исходной последовательности поразрядно суммируются по модулю два с кодом  $21_{16} = 0010.0001_2$  (это операция “Исключающее ИЛИ”, обычно обозначаемая аббревиатурой XOR). При этом коды 10, 11 и 13 особых символов преобразуются в коды 31, 30 и 32. В результате выполнения этих действий (называемых байтстаффингом — вставкой флагов и преобразованием байтов, следующих за ними) в памяти компьютера формируется последовательность символов, показанная на Рис. 2.93, г.

Эта последовательность примечательна тем, что она не содержит символов XON, XOFF и исходных символов DLE — эти символы “спрятаны” и помечены предупреждающими флагами — вновь введёнными символами DLE. При описании процедуры байтстаффинга такую последовательность обычно называют прозрачной, в том смысле, что в ней нет первоначально присутствовавших “загрязняющих примесей”, которые могли бы помешать правильному управлению потоком данных.

На следующем этапе (Рис. 2.93, д) поток символов передаётся в модем, причём в этот поток по мере необходимости в произвольных местах вводятся управляющие символы XON и XOFF. Эти символы теперь явно выделяются в “прозрачном” потоке и однозначно воспринимаются модемом как команды управления, после чего исключаются из потока и передаются на исполнение. Задача пересылки управляющих символов, таким образом, решена, и теперь остаётся восстановить в модеме исходную последовательность символов, совпадающую с показанной на Рис. 2.93, б.



**Рис. 2.93. Программное управление потоком данных: а — структурная схема системы передачи данных; б — исходная последовательность байтов; в, г — формирование последовательности байтов, не содержащей управляющих символов XON и XOFF; д — включение в последовательность управляющих символов XON и XOFF устройством DTE; е, ж — восстановление исходной последовательности байтов; з — восстановленная устройством DCE последовательность байтов, содержащая символы, случайно совпадающие с управляющими символами XON и XOFF**

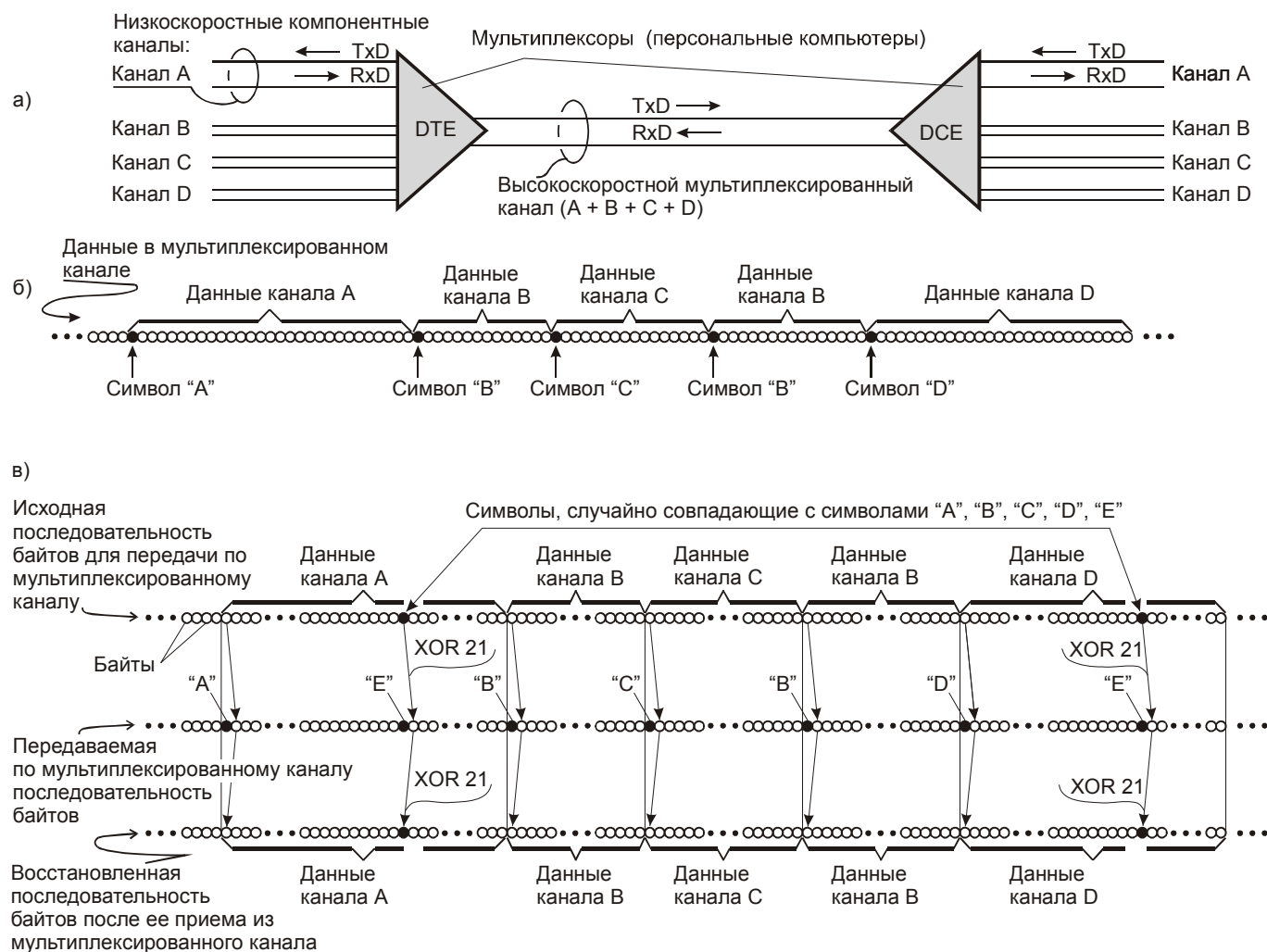
Для восстановления исходной последовательности модем отыскивает в потоке символы-флаги DLE и уничтожает их, а каждый сопряженный с флагом символ (соседний справа) поразрядно суммирует по модулю два с кодом  $21_{16}$  (Рис. 2.93, е, ж). В результате выполнения такой процедуры первоначально скрытые особые коды вновь приобретают первоначальный вид. Например, код  $11_{16}$  (Рис. 2.93, б) после первого поразрядного суммирования по модулю два с кодом  $21_{16}$  преобразуется в код  $30_{16}$  (Рис. 2.93, г), а после второго — восстанавливает исходное значение  $11_{16}$  (Рис. 2.93, ж, з). Иными словами, в процессе преобразования особые коды два-два

ды суммируются по модулю два с одной и той же константой ( $21_{16}$ ), что эквивалентно тому, что особые коды остаются неизменными, так как суммирование по модулю два двух одинаковых констант даёт нулевой результат.

Таким образом, байтстаффинг позволяет правильно управлять потоками, несмотря на возможное наличие в исходных данных “ложных” управляющих кодов. В худшем случае, когда исходные (пользовательские) данные представлены последовательностью, состоящей только из особых символов (XON, XOFF и DLE), каждый из них должен “упрятываться” и предваряться флаговым кодом DLE. Поэтому фактическая скорость передачи данных в плотном (без пауз) потоке символов по линии TxD уменьшится в два раза по сравнению с максимально возможной.

### 2.19.3. Мультиплексирование асинхронных данных в прозрачном потоке

Попробуем применить байтстаффинг для построения асинхронного мультиплексированного канала связи для объединения/разделения нескольких асинхронных потоков данных. Предположим, что в нашем распоряжении имеются два компьютера, каждый из которых содержит пять СОМ-портов (Рис. 2.94, а). Четыре из них имеют относительно невысокие скорости передачи данных, например, не превышающие 64 кбит/с; к этим портам подключены “компонентные” каналы А, В, С и D. Порты, объединённые мультиплексированным каналом связи, работают со скоростью, превышающей суммарную скорость всех компонентных каналов. При работе этой системы данные передаются между удалёнными друг от друга каналами (А — А, В — В, С — С, D — D) так, как будто эти каналы просто соединены соответствующими проводами.



**Рис. 2.94. Применение байтстаффинга для построения асинхронного мультиплексированного канала связи: а — система передачи данных; б — передаваемая по мультиплексированному каналу последовательность байтов; в — процессы разметки фрагментов мультиплексированной последовательности, исключения передатчиком и последующего восстановления приемником символов, случайно совпадающих с символами А — D и Е**

По мультиплексированному каналу связи с разделением во времени передаются данные от всех компонентных каналов. Каждый канал содержит линии передачи TxD и приёма RxD асинхронных данных. Данные передаются по этим линиям одновременно в обоих направлениях. Задача заключается в том, чтобы “разметить” мультиплексированный поток таким образом, чтобы в нём можно было распознать принадлежность данных тому или иному компонентному каналу. Схема симметрична, поэтому далее рассмотрена передача данных только в одном направлении — от компьютера DTE к компьютеру DCE.

Применим в этой системе передачи рассмотренную ранее идею байтстаффинга. Конечная цель — получить прозрачный поток, в котором символы А — D (выбраны произвольно) обозначают начала блоков данных от соответствующих каналов (Рис. 2.94, б). А для получения прозрачного потока, как мы знаем, следует временно “упрятать” подобные символы, случайно содержащиеся в исходном массиве данных.

Исходная последовательность байтов (символов) для передачи по мультиплексированному каналу связи хранится в памяти компьютера — передатчика (DTE). Перед выдачей в мультиплексированный канал эта последовательность несколько видоизменяется (Рис. 2.94, в). В ней преобразуются и помечаются символами Е коды, случайно совпадающие с кодами символов А — Е. После этого существует гарантия того, что последовательность не содержит “прежних” символов А — D и Е (если они были). Далее в последовательность вводятся “новые” символы А — D для разграничения блоков данных, относящихся к разным каналам.

Запись XOR 21, как и ранее, обозначает операцию поразрядного суммирования по модулю два кода, соответствующего началу стрелки, с кодом  $21_{16} = 0010.0001_2$ . Например, символ А, код которого равен  $41_{16} = 0100.0001_2$ , после поразрядного суммирования по модулю два с кодом  $21_{16}$  преобразуется в код  $60_{16} = 0110.0000_2$ . Напомним, что повторное применение операции XOR 21 по отношению к любому коду восстанавливает его первоначальное значение.

Вместо символов А — Е целесообразно использовать иные, редко употребляемые в текстовых документах символы. Это уменьшает вероятность совпадения пользовательских кодов с кодами особых символов, т. е. способствует уменьшению избыточности при подготовке прозрачной последовательности байтов для выдачи в мультиплексированный канал.

Таким образом, идея байтстаффинга упрощает построение достаточно сложных систем передачи данных — в нашем примере мультиплексированный канал связи выполнен на основе обычных персональных компьютеров с несколькими асинхронными СОМ-портами. Мультиплексирование и демultipлексирование данных выполняются только программными средствами.

## 2.20. Вопросы и задачи к разделу 2.19

2.20.1. Примените байтстаффинг к следующей последовательности пользовательских данных для исключения из неё копий команд управления:

FA, EE, 12, 10, 10, 10, 99, 13, 13, 05, 11, 10, 88, 11, 11, CC, DD, DD, 11

DLE = 10, XON = 11, XOFF = 13

Литература: [3]

*Решение задачи приведено в разделе 3 на с. 183*

2.20.2. Применяя метод программного управления потоком данных, введите команды XOFF и XON в начало и конец следующей последовательности пользовательских данных:

FA, EE, 12, 10, 10, 10, 99, 13, 13, 05, 11, 10, 88, 11, 11, CC, DD, DD, 11

DLE = 10, XON = 11, XOFF = 13.

Литература: [3]

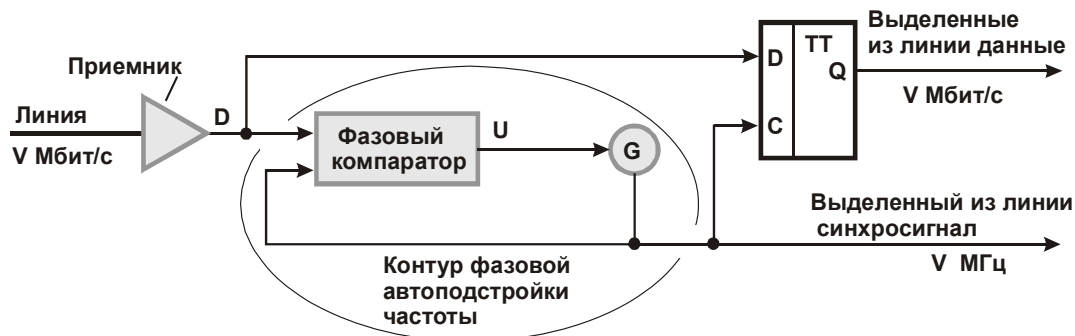
*Решение задачи приведено в разделе 3 на с. 183*

## 2.21. Выделение синхросигнала и данных из принимаемого сигнала

Литература к разделу 2.21: [1], гл. 9

### 2.21.1. Введение в раздел 2.21

В схеме, приведенной на Рис. 2.95, выделение синхросигнала и данных из линии основано на использовании контура фазовой автоподстройки частоты.



**Рис. 2.95. 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. 2.3.6) 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  $U$ . Например, 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 хорошо стабилизированным, 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 приведена на Рис. 2.96.

### 2.21.2. Двухконтурные схемы выделения синхросигнала

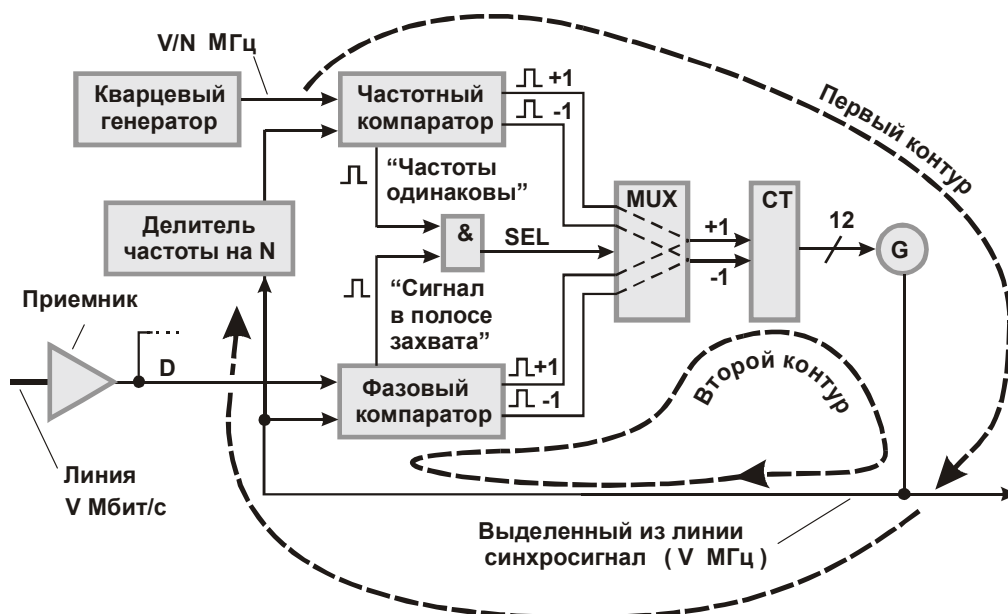


Рис. 2.96. Двухконтурная схема выделения синхросигнала из линии — первый вариант

Первый контур предназначен для предварительной настройки генератора  $G$  на номинальную частоту синхросигнала. После завершения такой настройки вместо первого контура включается второй, аналогичный рассмотренному ранее (Рис. 2.95).

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 частот 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 И получает от обоих компараторов сигналы лог. 0. Так как  $SEL = 0$ , мультиплексор MUX транслирует на входы счетчика СТ сигналы с выходов частотного компаратора.

Частотный компаратор определяет знак и величину частотной ошибки и формирует последовательность импульсов на одном из выходов. Код в счетчике СТ 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. 2.95).

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 СТ. 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. 2.97, 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.

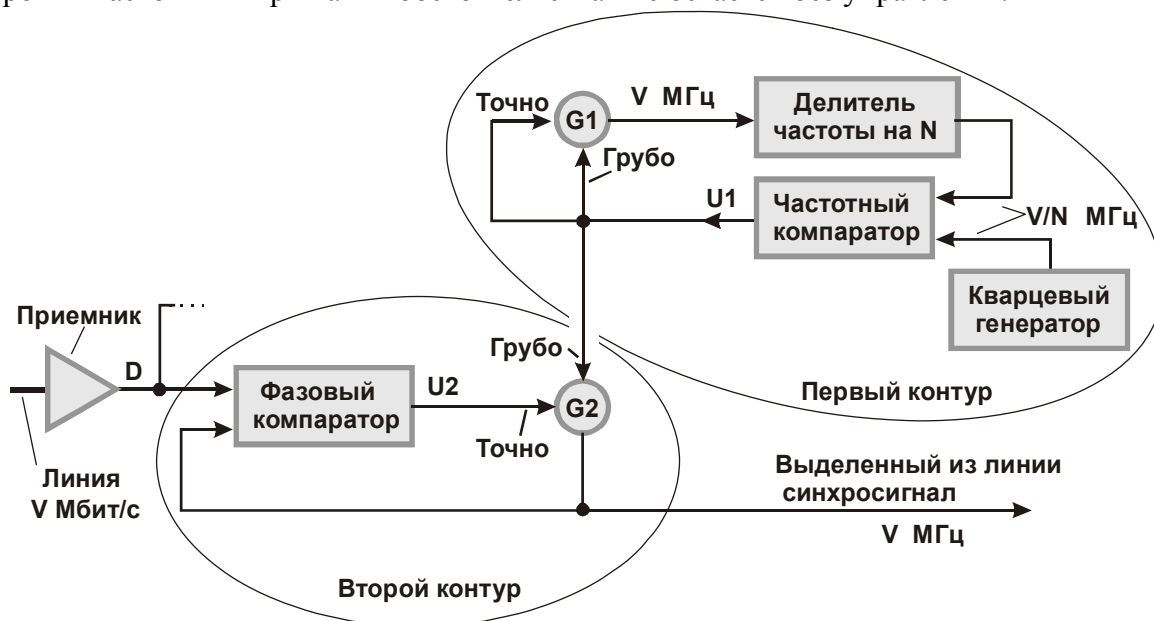


Рис. 2.97. Двухконтурная схема выделения синхросигнала из линии — второй вариант

Схема построена на основе двух нестабилизированных генераторов G1 и G2. Эти генераторы размещены в одном кристалле интегральной схемы и имеют одинаковую топологию. Поэтому они обладают очень близкими электрическими параметрами, что существенно для данного решения. Генератор G1 (G2) содержит аналоговые входы грубой и точной подстройки частоты. При нормальной работе устройства регулирующие напряжения U1 и U2 находятся примерно в середине диапазона регулировки (например близки 2,5 В при диапазоне (0,5 — 4,5) В). Входы грубой и точной подстройки генератора G1 объединены.

В первом, автономном контуре управления отслеживается частота кварцевого генератора, так что генератор G1 формирует стабилизированную частоту V, близкую скорости передачи данных в линии. Поскольку генераторы G1 и G2 имеют практически одинаковые электрические параметры, генератор G2 также настроен на эту частоту по входу грубой подстройки. В результате независимо от уровня напряжения U2 частота сигнала на выходе генератора G2 отличается от нужной всего лишь на доли герца и в любой ситуации не выходит за пределы полосы захвата фазового компаратора!

Таким образом, фазовый компаратор всегда готов к работе, даже после длительных “замираний” сигнала в линии. Исключен дрейф частоты в отсутствие сигнала в линии, уменьшено время вхождения в синхронизм после обнаружения этого сигнала.

Генераторы G1 и G2 выполнены на МДП-транзисторах по схеме симметричного мульти-вибратора (Рис. 2.98).

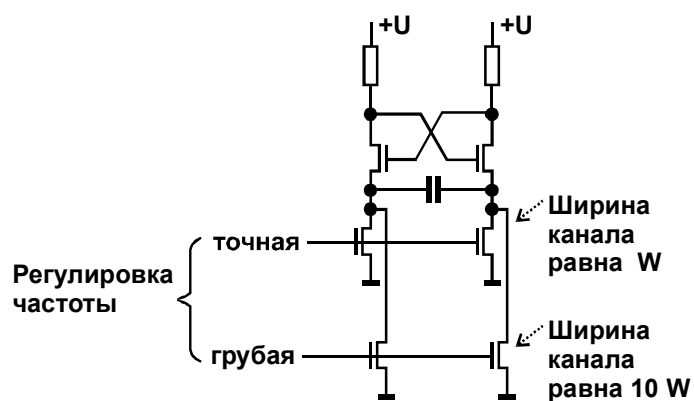


Рис. 2.98. Схема генератора

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.

## 2.22. Вопросы и задачи к разделу 2.21

2.22.1. На Рис. 2.99 показана двухконтурная схема выделения синхросигнала из принимаемого сигнала.

- Каким образом сигнал с кварцевого генератора влияет на синхросигнал, выделенный из линии?
- Одинаковы ли частоты сигналов на выходах генераторов G1 и G2?
- Для чего и какими средствами достигается близость характеристик генераторов G1 и G2?

Литература: [1], гл. 9

Ответы приведены в разделе 3 на с. 183

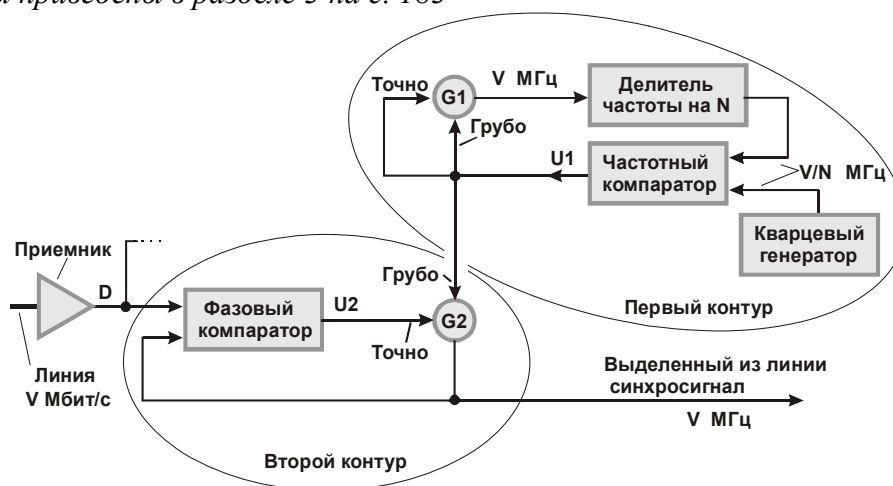


Рис. 2.99. Двухконтурная схема выделения синхросигнала из линии

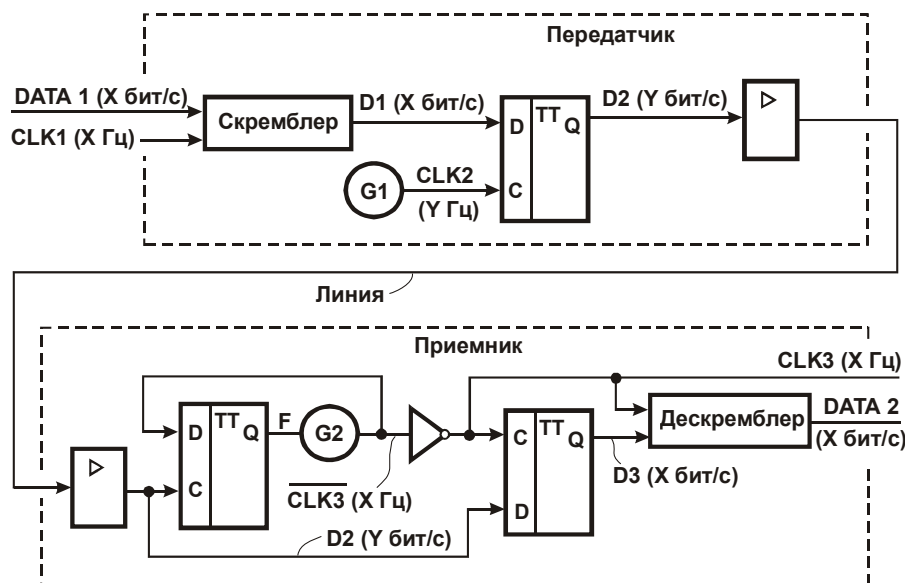
## 2.23. Сопряжение разноскоростных устройств

Литература к разделу 2.23: [1], гл. 4; [4]

### 2.23.1. Введение в раздел 2.23

Рассмотрим способ исключения проскальзываний при сопряжении разноскоростных компонентов одноканальной системы.

В системе передачи данных, показанной на Рис. 2.100, 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. 2.100. 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. Эта последовательность синхронизирована сигналом CLK1 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 (Рис. 2.101), 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. 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 С синхронизации первого триггера, а также на вход D данных второго триггера. Оба триггера срабатывают по положительному фронту сигнала на входе С. 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.

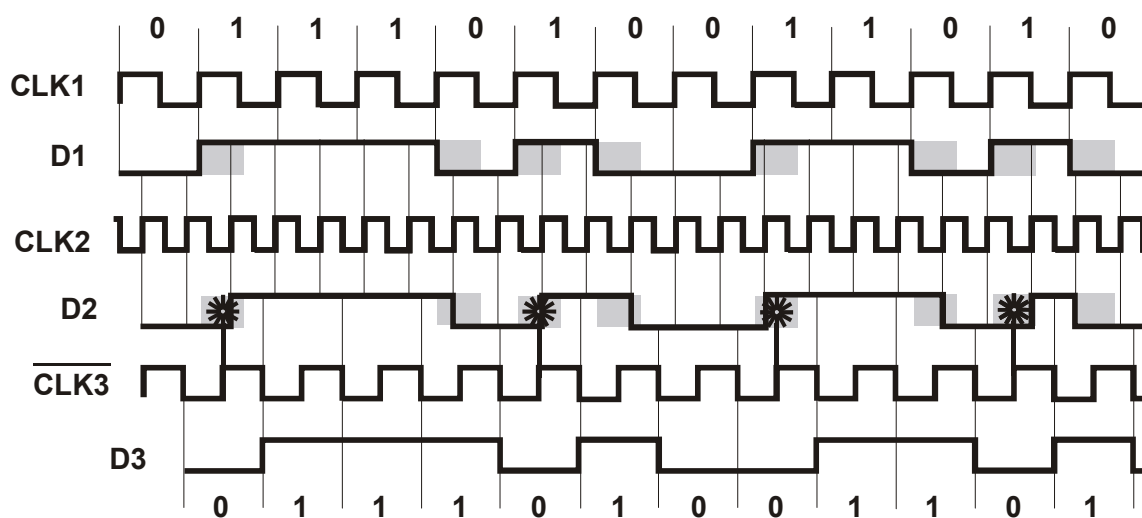


Рис. 2.101. Временные диаграммы передачи данных в системе, показанной на Рис. 2.100

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.

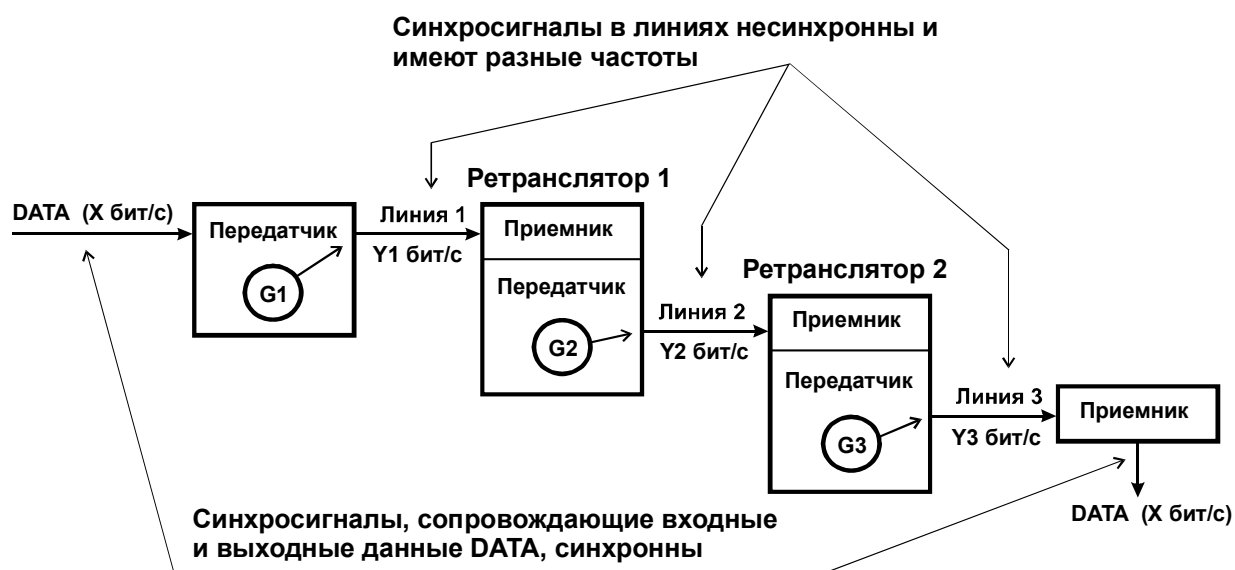
В результате постоянных незначительных колебаний около равновесного состояния осуществляется показанная на рисунке точная привязка синхросигнала NOT(CLK3) к сигналу D2 с учетом искусственно внесенного в него джиттера. Такая привязка обеспечивает стабильность данных на D-входе второго триггера в момент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

## 2.23.2. Многокаскадное сопряжение разноскоростных каналов

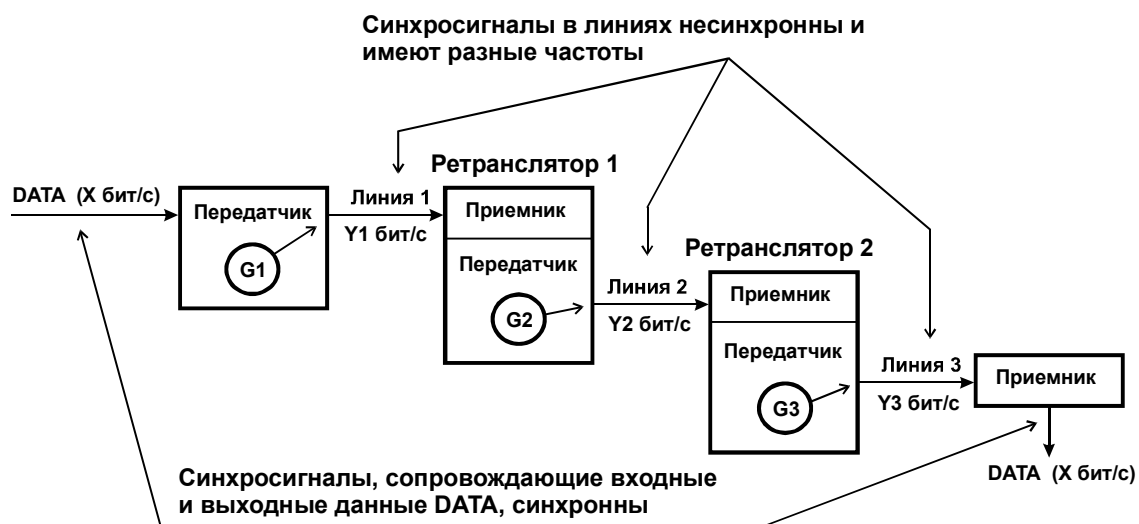
Рассмотренное решение позволяет создавать, на первый взгляд, “невозможные” каналы передачи данных; один из таких каналов показан на Рис. 2.102. Входные данные проходят по цепи “передатчик — ретранслятор 1 — ретранслятор 2 — приемник”. Передатчики и приемники выполнены по схемам, приведенным на Рис. 2.100, но скремблер и дескремблер (на рисунке не показаны) подключаются только на входе и выходе канала.



**Рис. 2.102. 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.





**Рис. 2.104. Сопряжение разноростных компонентов канала связи без использования буферов типа FIFO**

## 2.25. Канал E1

Литература к разделу 2.25: [19], [20]

### 2.25.1. Общие сведения

Канал E1 — первичный канал иерархии PDH (Plesiochronous Digital Hierarchy — плезиохронная цифровая иерархия, европейский стандарт для волоконно-оптических сетей) — является основным каналом, используемым во вторичных сетях телефонии, а также для передачи данных в сетях ISDN (Integrated Services Digital Network — цифровая сеть с интеграцией служб, ЦСИС). По сравнению с остальными каналами иерархии PDH этот канал имеет несколько особенностей, связанных с его использованием, а именно, сверхцикловую структуру и канал сигнализации, используемый во вторичных сетях цифровой телефонии и ISDN.

Остальные каналы иерархии PDH имеют только цикловую структуру. Такое отличие канала E1 обусловлено тем, что он является “пограничным” каналом между первичной и вторичными сетями. Структура систем передачи E1 включает три уровня эталонной модели OSI (Open System Interconnection): физический, канальный и сетевой.

- Физический уровень описывает электрический интерфейс и параметры сигнала E1.
- Канальный уровень описывает процедуры мультиплексирования и демultipлексирования каналов более низкого уровня, цикловую и сверхцикловую структуру потока E1, встроенные процедуры контроля ошибок.
- Сетевой уровень описывает процедуры управления каналами E1 в первичной сети, а также контроль параметров ошибок.

### 2.25.2. Физический уровень E1

Физический уровень E1 определяет электрические параметры интерфейса E1 и параметры сигналов передачи, включая структуру линейного кода, в соответствии с рекомендациями ITU-T G.703, G.704. Рассмотрим наиболее важные параметры.

*Основные характеристики интерфейса E1, тип линейного кодирования*

- Скорость передачи — 2048 кбит/с  $\pm 50$  ppm (1 ppm (point per million) =  $10^{-6}$ ), таким образом, допускается отклонение частоты передаваемого сигнала (2048 кГц) на величину  $\pm 102,4$  Гц.
- Типы кодирования: HDB3 или, в ранее разработанных устройствах, — AMI [1].

*Уровни сигналов, электрические параметры интерфейса, форма импульса*

|                                                                                         |                                  |                        |
|-----------------------------------------------------------------------------------------|----------------------------------|------------------------|
| Форма импульса электрического сигнала                                                   | В соответствии с рис. Рис. 2.106 |                        |
| Тип линии в каждом направлении                                                          | Одна коаксиальная пара           | Одна симметричная пара |
| Импеданс                                                                                | 75 Ом                            | 120 Ом                 |
| Номинальное пиковое напряжение импульса                                                 | 2,37 В                           | 3 В                    |
| Пиковое напряжение при отсутствии импульса                                              | $0 \pm 0,237$ В                  | $0 \pm 0,3$ В          |
| Номинальная ширина импульса                                                             | 244 нс                           |                        |
| Отношение амплитуд положительного и отрицательного импульсов в середине импульсного ин- | от 0,95 до 1,05                  |                        |

|                                                                           |                 |
|---------------------------------------------------------------------------|-----------------|
| тервала                                                                   |                 |
| Отношение ширины положительного импульса к ширине отрицательного импульса | от 0,95 до 1,05 |

Из таблицы следует, что существуют два стандарта на параметры физического интерфейса E1: симметричный интерфейс на основе двух витых пар с волновым сопротивлением 120 Ом и на основе двух коаксиальных (несимметричных) кабелей с волновым сопротивлением 75 Ом. Им соответствуют значения пикового напряжения 3 В и 2,37 В. Симметричный интерфейс получил наибольшее распространение в Европе и России. Несимметричный интерфейс получил широкое распространение на американско-канадском рынке. В России этот интерфейс не рекомендован к применению, тем не менее, он может встретиться при эксплуатации зарубежного оборудования.

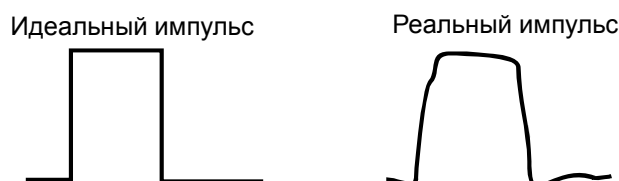


Рис. 2.105. Формы идеального и реального импульсов в линии связи

Как показано на Рис. 2.105, реальный импульс в системе передачи отличается от идеального. Форма реального импульса должна укладываться в незаштрихованную область Рис. 2.106.

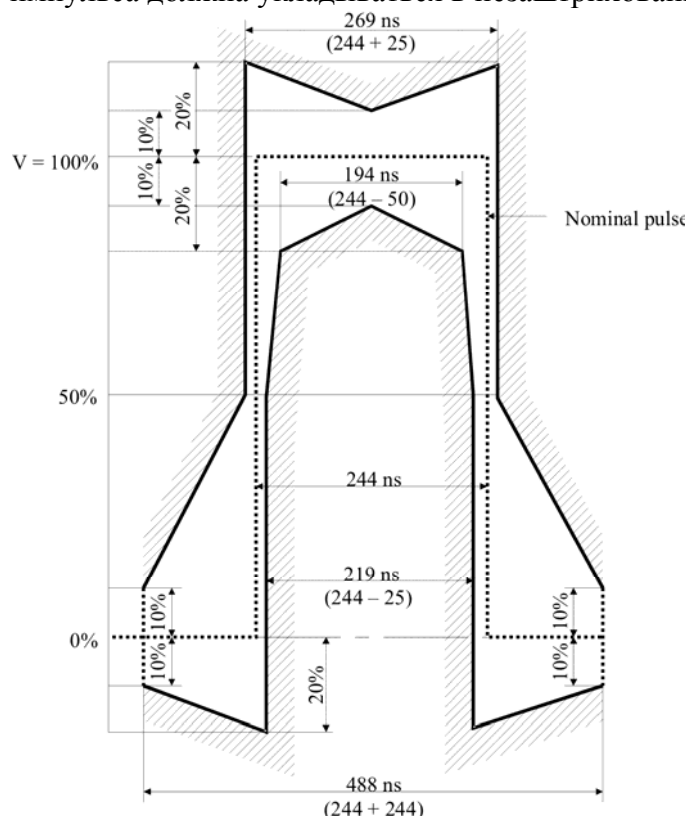


Рис. 2.106. “Маска” реального импульса (согласно рекомендации ITU-T G.703)

### 2.25.3. Канальный уровень

Канальный уровень включает в себя цикловую и сверхцикловую структуру потока, описание процедур контроля ошибок с использованием циклического избыточного кода (CRC), а также описание процедур мультиплексирования и демультиплексирования каналов тональной

частоты (ТЧ) в поток (из потока) Е1. Последние включают в себя процедуры дискретизации аналогового сигнала, которые в данной книге не рассматриваются.

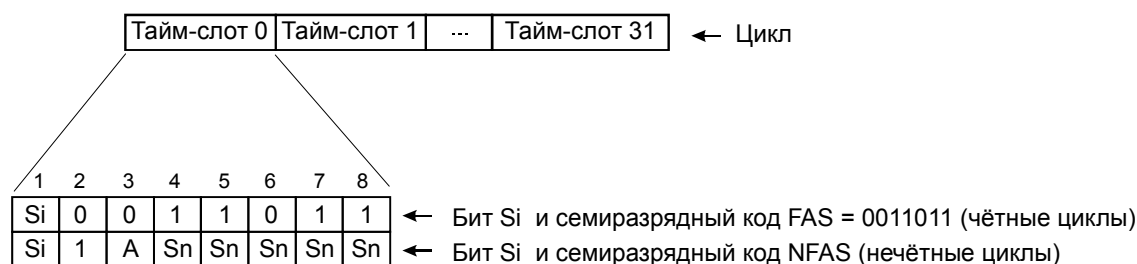
### Цикловая и сверхцикловая структура Е1

При передаче цифрового потока по первичной сети он преобразуется в последовательность циклов (они же именуются также фреймами, кадрами). Цикловая структура позволяет мультиплексировать и демultipлексировать потоки данных, передавать управляющую информацию, а также вести непрерывный контроль правильности передачи. Существуют три основных варианта структуры потока Е1: неструктурированный поток, с цикловой структурой и с цикловой и сверхцикловой структурой.

Неструктурированный поток Е1 используется в сетях передачи данных и не имеет цикловой структуры.

Поток Е1 с цикловой структурой предусматривает размещение в нём тридцати двух основных цифровых каналов (ОЦК) по 64 кбит/с с выделением каждому из них одного временно-го (канального) интервала TS (Time Slot). Эти интервалы пронумерованы от 0 до 31. В каждом временном интервале передаются 8 бит из соответствующего канала (нулевой таймслот является некоторым исключением). Длина цикла равна 256 бит, длительность его передачи составляет 125 мкс. Частота следования циклов равна 8 кГц.

Структура цикла представлена на Рис. 2.107. Различаются четные и нечетные циклы. В нулевых таймслотах TS0 четных циклов передается семиразрядный код FAS = 0011011 (Frame Alignment Signal), который служит флагом начала цикла, и один служебный бит Si, зарезервированный для международного использования. В нулевых таймслотах TS0 нечетных циклов также передаются один служебный бит Si и код NFAS, в котором размещены логическая 1 (предотвращающая ложное распознавание кода FAS) и биты A и Sn.



**Рис. 2.107. Структура цикла (кадра)**

Бит Si в нижней строке также зарезервирован для международного использования. Бит A (remote alarm indication) используется для передачи признака аварии удалённого устройства. Использование пяти служебных битов Sn строго не регламентируется. Они могут применяться, например, для построения служебных каналов связи между узлами сети. При пересечении национальных границ эти биты должны быть установлены в единичные состояния.

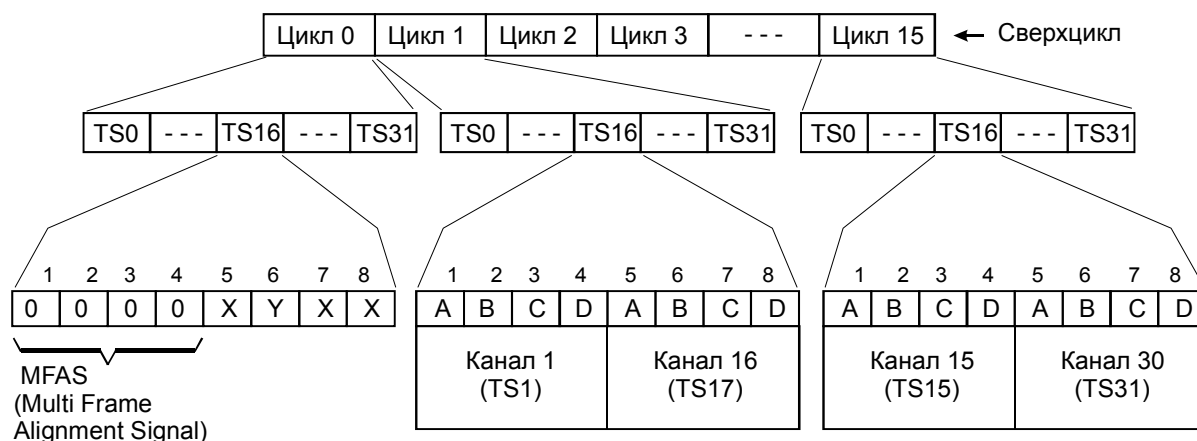
В отечественной терминологии вариант потока Е1 с цикловой структурой получил название ИКМ-31 (ИКМ — сокращение от словосочетания “импульсно-кодовая модуляция”). Он используется в ряде систем передачи данных, а также в некоторых приложениях: ОКС7, ISDN и В-ISDN.

В ряде случаев аппаратура передачи/приема Е1 использует еще и шестнадцатый канальный интервал (TS-16) для передачи информации о сигнализации, связанной с разговорным каналом (сигнализация CAS — current awareness service — служба сигнальной информации).

В этом случае поток Е1 в дополнение к цикловой имеет еще и сверхцикловую структуру. В отечественной терминологии такой вариант структуры Е1 получил название ИКМ-30. При этом 16 циклов объединяются в сверхцикл (суперкадр) размером 4096 бит и длительностью 2 мс.

Первый цикл в шестнадцатом канальном интервале (TS16) содержит информацию о сверхцикле (это своего рода флаг сверхцикла — четырёхразрядный код MFAS = 0000 (Multi Frame Alignment Signal)). Остальные 15 интервалов TS16, принадлежащие другим циклам, ис-

пользуются для передачи сигнальной информации (например, “В данном канале снята телефонная трубка”, “Линия занята” и т. п.). Структура сверхцикла показана на Рис. 2.108.



**Рис. 2.108. Структура сверхцикла**

Пояснения:

X — запасные биты, обычно установлены в состояние лог. 1

Y — distant multiframe alarm bit. Y = 1, если на удалённой стороне потеряна сверхцикловая синхронизация

A, B, C, D — биты сигнализации, отражающие состояние одного из каналов 1 — 30 (например, состояние номеронабирателя телефонного аппарата в соответствующем канале)

В структуре цикла (кадра) не предусмотрены избыточные биты для компенсации проскальзываний синхронизации из-за разности частот синхросигналов на передающей и приёмной сторонах канала Е1 (см. пп. 2.17, 2.29). Частоту проскальзываний можно уменьшить (но не исключить их полностью) применением буферной памяти типа FIFO, как показано на Рис. 2.113, в.

### Процедуры контроля ошибок передачи. Использование избыточного кода CRC-4

Структура сверхцикла ИКМ-30 предусматривает обнаружение ошибок. Для этого используются биты  $S_i$  в составе таймслотов TS0 чётных и нечётных циклов. Как показано в табл. 2, сверхцикл (точнее, представленная его часть) разбит на две половины, обозначенные SMF#1 и SMF#2. На месте битов  $S_i$  размещаются биты C1 — C4 двух кодов CRC-4. При вычислении и проверке кода CRC-4 используется образующий полином  $X^4 + X + 1$ . Вычисление и проверка этого кода выполняются по алгоритму, аналогичному рассмотренному в примере, приведенном в [1] на рис. 7.25, с. 162. Отличие состоит в том, что в упомянутом примере используется код CRC-8 и соответствующий ему полином  $X^8 + X^2 + X + 1$ .

Каждый код CRC-4 соответствует не “своей”, а переданной ранее половине сверхцикла, т. е. обработка данных проводится в конвейерном режиме. Передатчик E1 вычисляет код CRC-4 для некоторой половины сверхцикла и включает результат вычислений не в передаваемую, а лишь в следующую половину сверхцикла, которая ещё ждёт своей очереди на обработку и передачу. Приемник принимает очередную половину сверхцикла, проводит аналогичный расчет кода CRC-4 и в дальнейшем сравнивает вычисленный код с тем, который будет принят в составе следующей половины сверхцикла. Если в вычисленном и принятом кодах имеется расхождение, то в сверхцикл помещается бит ошибки E1 = 0 или (и) E2 = 0 в зависимости от того, в какой его половине обнаружена ошибка при вычислении и проверке кода CRC-4. Помеченные битами E1 = 0 или (и) E2 = 0 сверхциклы отсылаются назад к их источнику для замены на безошибочные.

Таблица 2. Структура нулевых канальных интервалов сверхцикла

|           | Половина<br>сверхцикла<br>(канальный<br>интервал 0) | Номер<br>цикла | Канальный интервал 0<br>(биты 1 – 8) |   |   |     |     |     |     |     |
|-----------|-----------------------------------------------------|----------------|--------------------------------------|---|---|-----|-----|-----|-----|-----|
|           |                                                     |                | 1                                    | 2 | 3 | 4   | 5   | 6   | 7   | 8   |
| Сверхцикл | SMF#I                                               | 0              | C1                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 1              | 0                                    | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           |                                                     | 2              | C2                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 3              | 0                                    | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           |                                                     | 4              | C3                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 5              | 1                                    | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           |                                                     | 6              | C4                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 7              | 0                                    | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           | SMF#II                                              | 8              | C1                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 9              | 1                                    | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           |                                                     | 10             | C2                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 11             | 1                                    | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           |                                                     | 12             | C3                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 13             | E1                                   | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |
|           |                                                     | 14             | C4                                   | 0 | 0 | 1   | 1   | 0   | 1   | 1   |
|           |                                                     | 15             | E2                                   | 1 | A | Sn4 | Sn5 | Sn6 | Sn7 | Sn8 |

Пояснения:

SMF#1 и SMF#2 — Sub-multiframe — первая и вторая половины сверхцикла (канальный интервал 0).

C1 — C4 — биты кодов CRC.

001011 — код сверхцикловой синхронизации “по вертикали” (выделен фоном). Для его обнаружения просмотрите столбец “Бит 1” в направлении сверху вниз, не обращая внимания на биты C1 — C4, E1, E2.

Биты E1 и E2 идентифицируют ошибку при проверке кода CRC (E = 1 — нет ошибки, E = 0 — есть ошибка).

Sn = биты, зарезервированные для национального использования и для передачи управляющей информации сетевого уровня.

На сетевом уровне E1 осуществляются процедуры управления первичной сетью, которые широко используют сигналы о неисправностях, генерируемые в цифровых системах передачи, а также сигналы о возникновении ошибок, фиксируемые встроенными средствами диагностики. Эта информация собирается в узлах системы управления и обрабатывается. Таким образом, сетевой уровень E1 включает в себя набор определенных служебных сигналов и сообщений, используемых системой управления первичной сетью. Такие сообщения делятся на три категории:

- сообщения о возникновении ошибок в системе передачи;
- сообщения о неисправностях, возникающих в системе передачи;
- сообщения, используемые для реконфигурирования первичной сети и восстановления системы синхронизации.

Структура этих сообщений здесь не рассматривается.

## 2.26. Вопросы и задачи к разделу 2.25

### 2.26.1. Какие способы кодирования используются в канале E1?

Литература: разделы 2.3, 2.25 данной книги, [20]

*Ответ приведен в разделе 3 на с. 184*

### 2.26.2. Сколько кабелей входят в состав канала E1 и какого типа кабели могут использоваться?

Литература: раздел 2.25 данной книги, [20]

*Ответы приведены в разделе 3 на с. 184*

### 2.26.3. Уникален ли флаг начала кадра E1 (Рис. 2.107)? Все ли кадры имеют флаги?

Литература: разделы 2.17, 2.25 данной книги, [20], [1], гл. 7

*Ответы приведены в разделе 3 на с. 185*

### 2.26.4. Часть передаваемых кадров E1 не имеют флагов (Рис. 2.107). Как приёмник распознаёт эти кадры? Как устанавливается кадровая синхронизация между приёмником и передатчиком?

Литература: разделы 2.17, 2.25 данной книги, [20], [1], гл. 7

*Ответы приведены в разделе 3 на с. 185*

### 2.26.5. В структуре кадра E1 (Рис. 2.107) имеются пять битов $S_n$ для передачи служебной информации. Какова пропускная способность полученного таким образом служебного канала?

Литература: раздел 2.25 данной книги, [20], [1], гл. 7

*Решение задачи приведено в разделе 3 на с. 185*

### 2.26.6. В структуре кадра E1 (Рис. 2.107) имеется бит A для передачи признака аварии удалённого устройства. Какова пропускная способность полученного таким образом служебного канала? Какова пропускная способность канала на основе битов $S_i$ ?

Литература: раздел 2.25 данной книги, [20], [1], гл. 7

*Решение задач приведено в разделе 3 на с. 185*

### 2.26.7. Какова пропускная способность канала, образованного пятью таймслотами кадра E1?

Литература: раздел 2.25 данной книги, [20], [1], гл. 7

*Решение задачи приведено в разделе 3 на с. 185*

### 2.26.8. Какова максимальная пропускная способность канала E1 без учёта потока служебных битов $S_i$ , FAS, NFAS (Рис. 2.107)?

Литература: раздел 2.25 данной книги, [20], [1], гл. 7

*Решение задачи приведено в разделе 3 на с. 186*

### 2.26.9. Как в каналах E1 осуществляется предотвращение проскальзываний синхронизации?

Литература: разделы 2.25, 2.29 данной книги, [20], [1], гл. 4

*Ответ приведен в разделе 3 на с. 186*

## 2.26.10. Для чего в структуру потока E1 введён сверхцикл (суперкадр)?

Литература: раздел 2.25, данной книги, [20]

*Ответ приведен в разделе 3 на с. 186*

## 2.26.11. В структуре сверхцикла (суперкадра) E1 (Рис. 2.108) предусмотрена группа из 15 таймслотов сигнализации. Какова скорость передачи сигнальной информации для каждого из 30 телефонных каналов?

Литература: раздел 2.25 данной книги, [20]

*Решение задачи приведено в разделе 3 на с. 187*

## 2.26.12. В структуре сверхцикла (суперкадра) E1 (Рис. 2.108) в шестом разряде 16-го таймслота нулевого цикла предусмотрен признак Y потери синхронизации. Оцените максимальную задержку от события потери синхронизации удалённым передатчиком до момента обнаружения этого признака приёмником.

Литература: раздел 2.25 данной книги, [20]

*Решение задачи приведено в разделе 3 на с. 187*

## 2.26.13. Насколько уверенно можно судить о правильности полученной половины суперкадра по принятому коду CRC, который совпал с ожидаемым?

Литература: раздел 2.25 данной книги, [20]

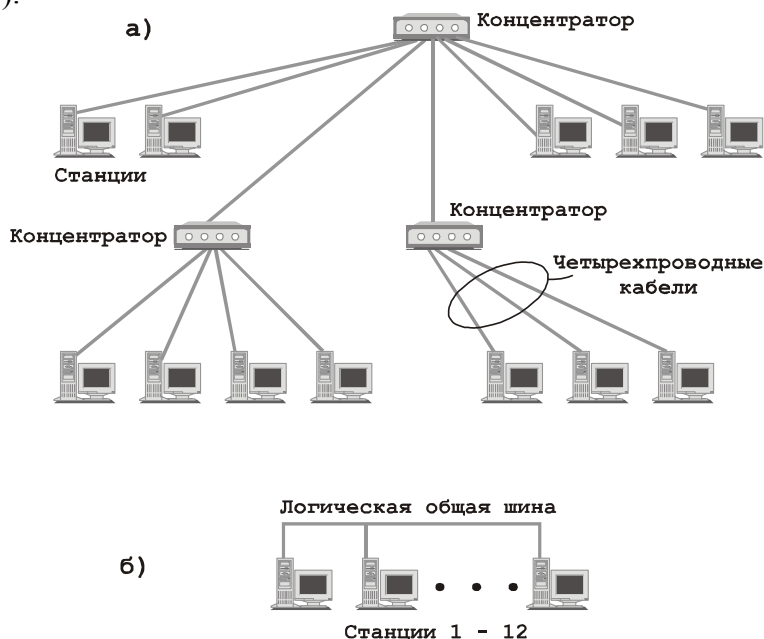
*Решение задачи приведено в разделе 3 на с. 187*

## 2.27. Объединение удалённых сегментов сети Ethernet 10 Base T

Литература к разделу 2.27: [19]

### 2.27.1. Структура сети Ethernet 10 Base T

Сеть Ethernet 10 Base T представляет собой пирамидальную схему из станций и концентраторов (Рис. 2.109).



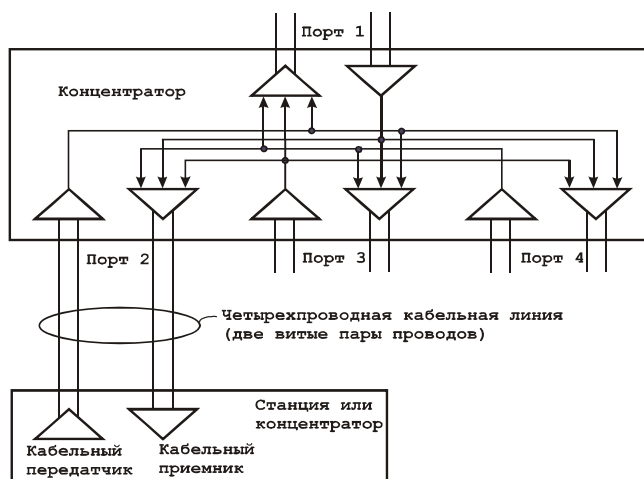
**Рис. 2.109. Пример построения сети Ethernet 10 Base T: а — топология связей между устройствами; б — логическая структура сети**

В данном примере сеть объединяет 12 станций — персональных компьютеров с сетевыми картами (платами сопряжения с четырехпроводными кабелями сети). Все станции равноправны, хотя и находятся на разных уровнях пирамидальной структуры (Рис. 2.109, а). Логически все станции подключены к общей шине; в этом смысле концентраторы можно было бы исключить из рассмотрения и считать, что структура сети соответствует приведенной на Рис. 2.109, б.

Каждая станция имеет индивидуальный физический адрес (так называемый MAC-адрес, MAC — Media Access Control — управление доступом к разделяемой среде передачи сигналов). Адрес — это уникальное 48-разрядное число, обычно присваиваемое сетевому адаптеру фирмой-изготовителем. Одна из частей адреса отражает уникальный код фирмы, другая — заводской номер адаптера из диапазона 0 — 16 млн.

Данные передаются в коде “Манчестер-II” (п. 2.3). Обмен данными по сети происходит в полудуплексном или дуплексном режиме. В любом случае одна станция является источником адреса, данных и других параметров, заключенных в единый кадр (пакет), а другая, распознавшая свой адрес, — приемником. Принимающих станций может быть несколько при передаче широковещательных и многоадресных кадров (пакетов), а может и не быть вовсе, когда адрес не распознан ни одной из станций (если нужная станция, например, просто выключена).

Простейший концентратор, обеспечивающий полудуплексный режим работы, показан на Рис. 2.110. Он содержит несколько портов. Логика работы такова: при получении сигнала на вход одного из портов концентратор транслирует его на выходы всех остальных портов, т. е. размножает сигнал. Таким образом, каждая станция передает сигнал всем остальным, как в структуре с общей шиной (см. Рис. 2.109, б).



**Рис. 2.110. Упрощенная схема одного из концентраторов (см. Рис. 2.109, а); все порты равноправны**

Прежде чем начать передачу кадра, станция должна убедиться, что общая шина (точнее, эквивалентная ей пирамидальная структура, см. Рис. 2.109, а) свободна. Об этом можно судить по отсутствию сигнала на выходе кабельного приемника станции. Если шина была свободна или только что освободилась, то станция начинает ею пользоваться.

Однако при этом не исключена возможность того, что одна или несколько станций также попытаются передать в общую шину свои кадры, так что возникает конфликт, называемый коллизией. Как видно из схемы, приведенной на Рис. 2.110, при бесконфликтной передаче кадра сигнал на выходе кабельного приемника передающей станции отсутствует, а при коллизии на этом выходе формируется логическая сумма сигналов от конкурирующих станций. При обнаружении коллизии станция сначала “усугубляет” ее выдачей специальной 32-битовой jam-последовательности (jam — загромождение, заедание, помеха), а затем освобождает общую шину. Повторная попытка передачи кадра предпринимается после небольшой паузы, имеющей случайную длительность в определенном диапазоне.

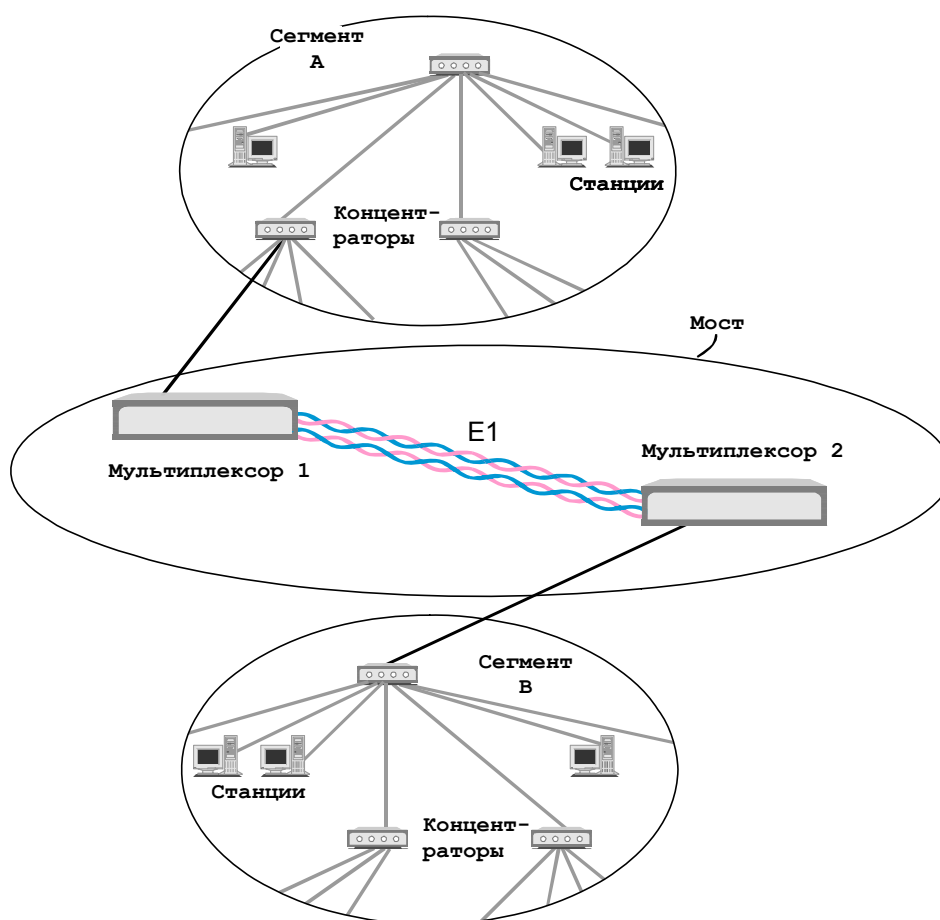
Применение концентраторов позволяет электрически разделить общую шину на отдельные фрагменты и тем самым решить ряд проблем:

- уменьшить нагрузку на кабельный передатчик станции;
- повысить качество линии благодаря исключению неоднородностей, вносимых ответвлениями от общего кабеля;
- обеспечить оперативное обнаружение и отключение отказавшей ветви концентратор — станция; для этого через каждые 16 мс каждая ветвь проверяется тестом связности (link test). (Аппаратура отключения ветвей на рисунке не показана.)

Более подробное описание принципов построения сети Ethernet 10 Base T можно найти в [19].

## 2.27.2. Мост на основе двух мультиплексоров

Рассмотрим два сегмента единой сети Ethernet 10 Base T, объединенные мостом (Рис. 2.111). Мост выполнен на основе двух мультиплексоров, подключенных к противоположным сторонам линии связи, выполненной в виде витой пары проводов. Эта пара может входить в состав стандартного кабеля городской телефонной сети. Диаметр медной жилы провода обычно равен 0,4 мм; с увеличением диаметра дальность связи между мультиплексорами возрастает.



**Рис. 2.111. Пример построения моста между двумя сегментами сети Ethernet 10 Base T с использованием двух мультиплексоров**

Мультиплексор содержит, в частности, порт Ethernet. Он предназначен для подключения к концентратору или сетевой карте станции сети Ethernet 10 Base T. Как отмечалось, все порты всех концентраторов равноправны, поэтому для подключения мультиплексора к сегменту сети можно выбрать любой незадействованный порт любого концентратора. Непосредственное подключение мультиплексора к сетевой карте станции соответствует вырожденному сегменту сети, в котором присутствует только одна эта станция. Расстояние между мультиплексором и концентратором (или сетевой картой станции) не должно превышать 100 м при использовании витых пар проводов категории не ниже третьей.

С помощью иерархической системы меню (реализованной в программном обеспечении мультиплексора) оператор может задать с лицевой панели параметры настройки обоих мультиплексоров. Один из мультиплексоров может рассматриваться как “локальный” (local), т. е. находящийся рядом с человеком-оператором. Второй называется “удаленным” (remote) по отношению к первому.

### 2.27.3. Обмен кадрами через мост

Мост позволяет передавать кадры (пакеты) из сегмента А в сегмент В (см. Рис. 2.111) и обратно, если возникает необходимость такой передачи. Благодаря упомянутой ранее уникальности адресов сетевых карт, все станции сегментов А и В различимы.

Мост оперирует адресами физического уровня и остается прозрачным для высокоуровневых протоколов, таких как TCP/IP, DECnet и IPX, а также операционных систем, таких как NetWare и MS LAN manager. С их точки зрения сегменты А и В просто объединены в единую сеть, а такие “мелочи” как сортировка адресов, принадлежащих разным сегментам, остаются незамеченными.

Мультиплексор содержит буферную память для хранения до 10000 MAC-адресов станций своего (т. е. непосредственно связанного с ним) сегмента сети Ethernet 10 Base T. При этом сам мультиплексор не имеет адреса в сети. В процессе работы мультиплексор постоянно обучается: он следит за адресами станций своего сегмента сети и ведет их оперативный учет в буферной памяти. В результате он поименно знает все свои станции, которые проявляли активность (инициировали обмен данными) в течение последних пяти минут. Адреса утративших активность станций вычеркиваются из учетного списка.

Такое обучение возможно благодаря тому, что в кадрах, передаваемых по сети, содержится, в частности, адрес отправителя. После добавления в сегмент сети новой станции она автоматически становится на учет после первого проявления активности.

Мультиплексор буферизует все передаваемые по своему сегменту сети кадры и анализирует их тип, адрес отправителя и адрес предполагаемого получателя (если такой адрес предусмотрен данным типом кадра). Кадры, не соответствующие принятым форматам (например искаженные вследствие возникновения коллизий), либо с неправильной контрольной суммой отбрасываются (уничтожаются).

Широковещательные и многоадресные кадры (типа Broadcast и Multicast, адресованные всем или определенной группе станций), а также кадры с MAC-адресами получателя, не учтенными в списке активных адресов своего сегмента сети, ставятся в очередь для пересылки в удаленный мультиплексор и далее в связанный с ним сегмент. Для пересылки кадров между мультиплексорами используется канал E1. На удаленной стороне кадры извлекаются из канала E1 и передаются в удаленный сегмент сети. Очередь на передачу кадров через мост может состоять из нескольких десятков или сотен кадров; ее переполнение приводит к потере вновь поступающих кадров.

Если адресат находится в пределах своего сегмента сети, то мультиплексор уничтожает полученный кадр. На начальном этапе обучения некоторые кадры ошибочно пересылаются из сегмента А в сегмент В; аналогичные ошибочные передачи происходят и в обратном направлении. Эти кадры теряются, не найдя адресата. То же наблюдается и в установившемся режиме, когда по истечении пяти минут утратившая активность станция принимает кадр из своего сегмента сети. Копия этого кадра пересылается в удаленный сегмент и теряется. Такие ситуации, однако, не приводят к сколько-нибудь заметному снижению пропускной способности моста, так как на фоне высокой скорости сортировки кадров единичными ложными передачами можно пренебречь.

#### **2.27.4. Преобразование кадра при его передаче между сегментами сети**

В сети Ethernet 10 Base T предусмотрены четыре типа кадров. Рассмотрим процесс передачи кадра наиболее простого типа (Raw 802.3/Novel 802.3) из сегмента А в сегмент В. Кадр (Рис. 2.112, а) содержит пять полей: DA, SA, L, D (Data) и FCS.

Передаче кадра предшествуют преамбула — Preamble, 7 байтов вида 10101010 и начальный ограничитель кадра SFD — Start of Frame Delimiter, один байт вида 10101011. Сразу за ними (без промежутка, показанного на рисунке лишь для выделения собственно кадра) следуют адрес назначения DA — Destination Address и адрес источника SA — Source Address, тот и другой по шесть байтов.

Поле L из двух байтов (Length — длина) задает длину поля данных в кадре — от 0 до 1500 байтов. Если L меньше 46, то поле данных автоматически расширяется до этой границы добавлением заполняющих (padding) байтов. Это позволяет утверждать, что минимальная длина кадра равна 64 байтам, что, в свою очередь, обеспечивает корректную работу механизма обнаружения коллизий. Кадр завершается четырьмя байтами контрольной суммы FCS — Frame Check Sequence, вычисленной по алгоритму циклического кодирования CRC-32 (CRC — Cyclic Redundancy Check).



**Рис. 2.112. Эволюция кадра Ethernet в процессе его передачи из сегмента А в сегмент В (см. Рис. 2.111). Цифрами обозначено число байтов в соответствующих полях**

При передаче кадра по сегменту А все станции и мультиплексор 1 обнаруживают преамбулу, соответствующую периодическому сигналу "Манчестер-II" частотой 5 МГц. Этот сигнал используется приемниками для вхождения в синхронизм с передатчиком. Далее, получив байт SFD, все потенциальные абоненты сегмента А отмечают начальную границу кадра и начинают его прием. В данном примере предполагаем, что адрес назначения DA соответствует одной из станций сегмента В. Поэтому абоненты сегмента А обнаруживают, что данные адресованы не им, и отбрасывают принятый кадр (или принятый фрагмент кадра).

Мультиплексор 1 в любой ситуации принимает и буферизует полный кадр (см. Рис. 2.112, а, б). Далее проверяются правильность формата, отсутствие синтаксических ошибок и совпадение вычисленной и принятой контрольных сумм FCS. Если ошибок нет (в противном случае кадр отбрасывается), то из поля данных исключаются заполняющие байты (если это поле содержит менее 46 полезных байтов), так что поле данных сжимается, а контрольная сумма уничтожается (см. Рис. 2.112, в).

Далее кадр, показанный на Рис. 2.112, в, нарезается мультиплексором 1 на фрагменты и упаковывается в кадры канала Е1. На дальней стороне канала Е1 мультиплексор 2 выполняет обратную операцию: он извлекает фрагменты кадра из канала Е1 и восстанавливает его целостность (Рис. 2.112, г, д).

После этого (см. Рис. 2.112, е) поле данных, если это необходимо, дополняется до 46 байтов байтами заполнения, кадр комплектуется вновь вычисленной контрольной суммой. Затем в соответствии с протоколом доступа после выдачи преамбулы и начального ограничителя кадр пересылается в сегмент В (см. Рис. 2.112, ж).

## 2.28. Вопросы и задачи к разделу 2.27

2.28.1. С какой целью в кадре Ethernet присутствует преамбула? Какова форма сигнала преамбулы?

Литература: [1], гл. 8; [19]

*Ответы приведены в разделе 3 на с. 188*

2.28.2. Каковы скорости передачи Ethernet-кадра по сегменту сети и через мост (Рис. 2.111)?

Литература: [19], [20]

*Ответ приведен в разделе 3 на с. 188*

2.28.3. Как мультиплексор 1 (2) определяет, нужно ли передавать кадр через мост (Рис. 2.111)?

Литература: [19], [20]

*Ответ приведен в разделе 3 на с. 188*

2.28.4. Каковы возможные причины потерь Ethernet-кадров при их передаче через мост (Рис. 2.111)?

Литература: [19], [20]

*Ответ приведен в разделе 3 на с. 188*

2.28.5. Как источник Ethernet-кадров оповещается об их потере?

Литература: [19]

*Ответ приведен в разделе 3 на с. 188*

2.28.6. Каковы MAC-адреса мультиплексоров 1 и 2 (Рис. 2.111)?

Литература: [19]

*Ответ приведен в разделе 3 на с. 188*

2.28.7. Как будет осуществляться передача данных через мост при совпадении MAC-адресов двух станций, расположенных в сегментах А и В (Рис. 2.111)?

Литература: [19]

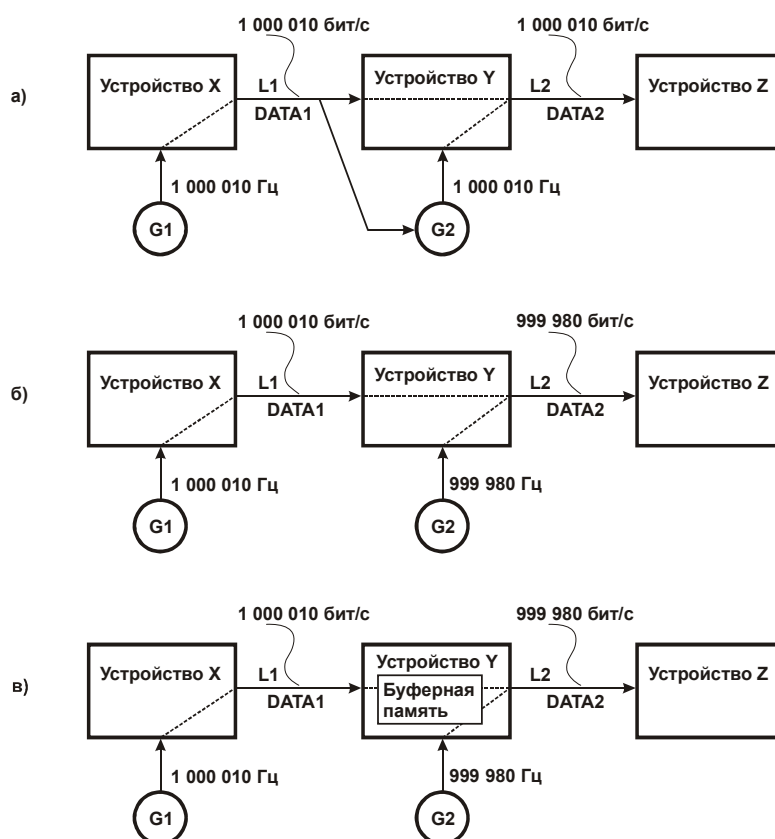
*Ответ приведен в разделе 3 на с. 188*

## 2.29. Проскальзывания синхронизации

Литература к разделу 2.29: [1], гл. 4

### 2.29.1. Введение

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. 2.113, 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. 2.113. Синхронизация передачи данных: а — правильная (без проскальзываний); б — неправильная (с проскальзываниями, когда буферной памяти нет); в — допустимый вариант (желательно 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. 2.113, 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 “втекают” 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. 2.113, б, либо сохранены в этой памяти, если она имеется и еще не переполнена (Рис. 2.113, 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.

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 удается 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. Однако из сказанного не следует, что ситуация безнадежна; проскальзывания, как показано далее, можно предотвратить.

Влияние проскальзываний на качество передачи различного рода данных отражено в табл. 3.

**Таблица 3. Проявления проскальзываний синхронизации при передаче разных типов данных или использовании разных технологий их передачи**

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

## 2.29.2. Предотвращение проскальзываний

Далее показана возможность предотвращения проскальзываний введением избыточных битов в потоки данных. 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 буферной памяти, вызываемое неточным равенством частот синхросигналов от генераторов  $G1$  и  $G2$  (см. Рис. 2.113, в).

Если буферная память (см. Рис. 2.113, 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. 2.114). 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 ...BBB... 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.

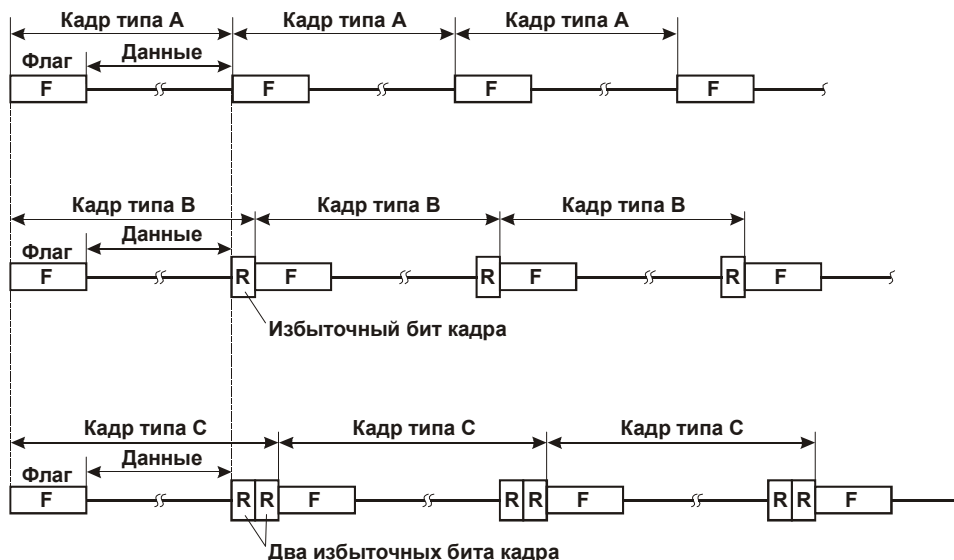


Рис. 2.114. Структура кадров, используемых для предотвращения проскальзываний

Прежде чем пояснить, каким образом используются избыточные биты, рассмотрим модель прохождения последовательности кадров через буферную память типа FIFO (Рис. 2.115). 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.

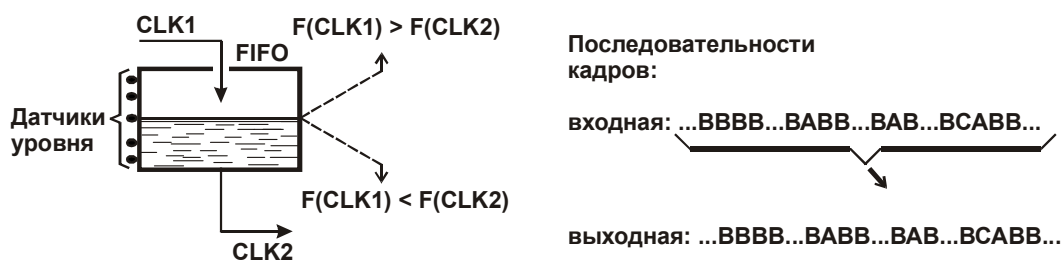
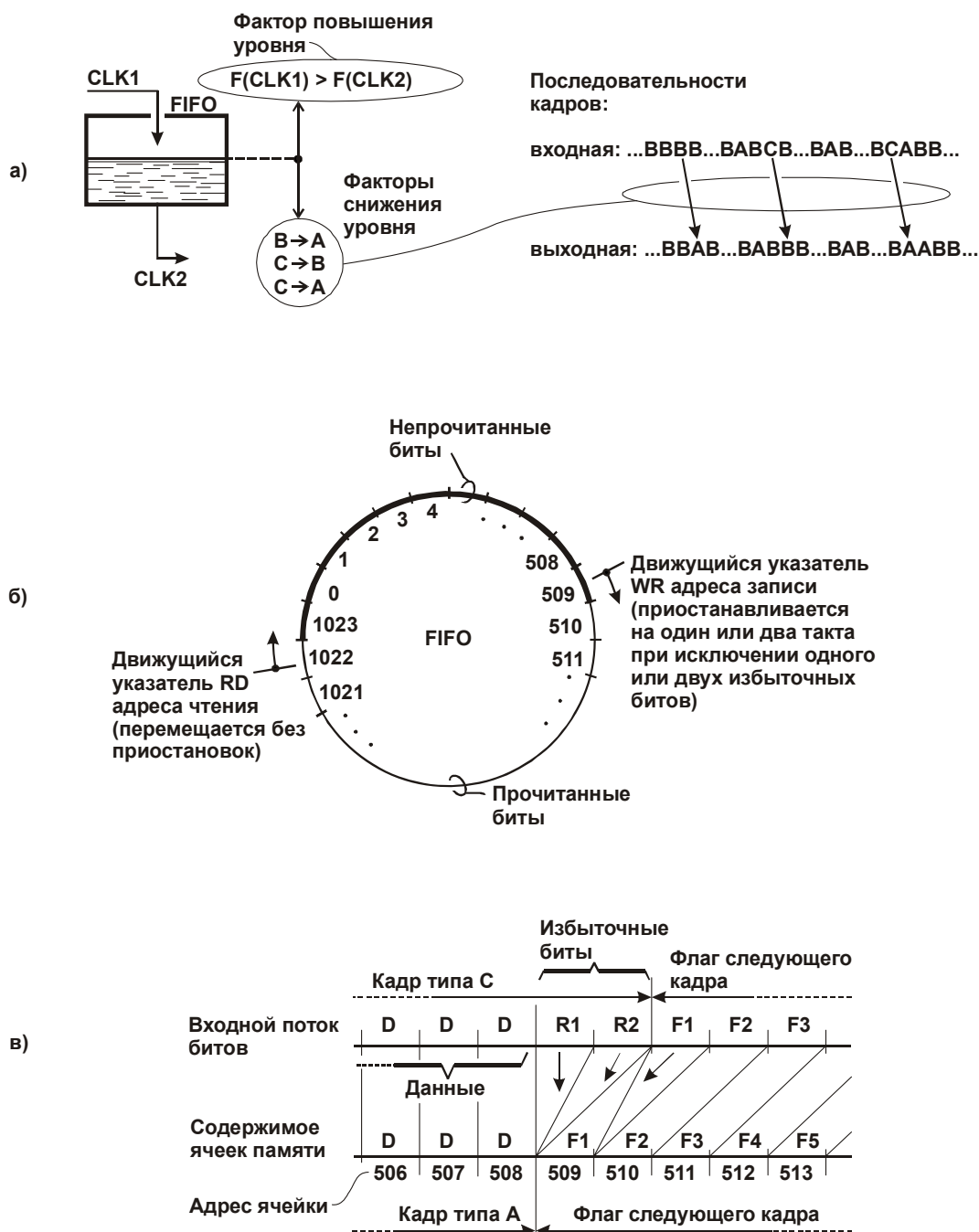


Рис. 2.115. Прохождение последовательности кадров через буферную память типа FIFO в отсутствие коррекции уровня ее заполнения

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. 2.116, Рис. 2.117).



**Рис. 2.116. 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. 2.116, 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 разрешима, 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. 2.116, б. На этом рисунке приведена уточненная модель буферной памяти, позволяющая более детально описать протекающие в ней процессы.

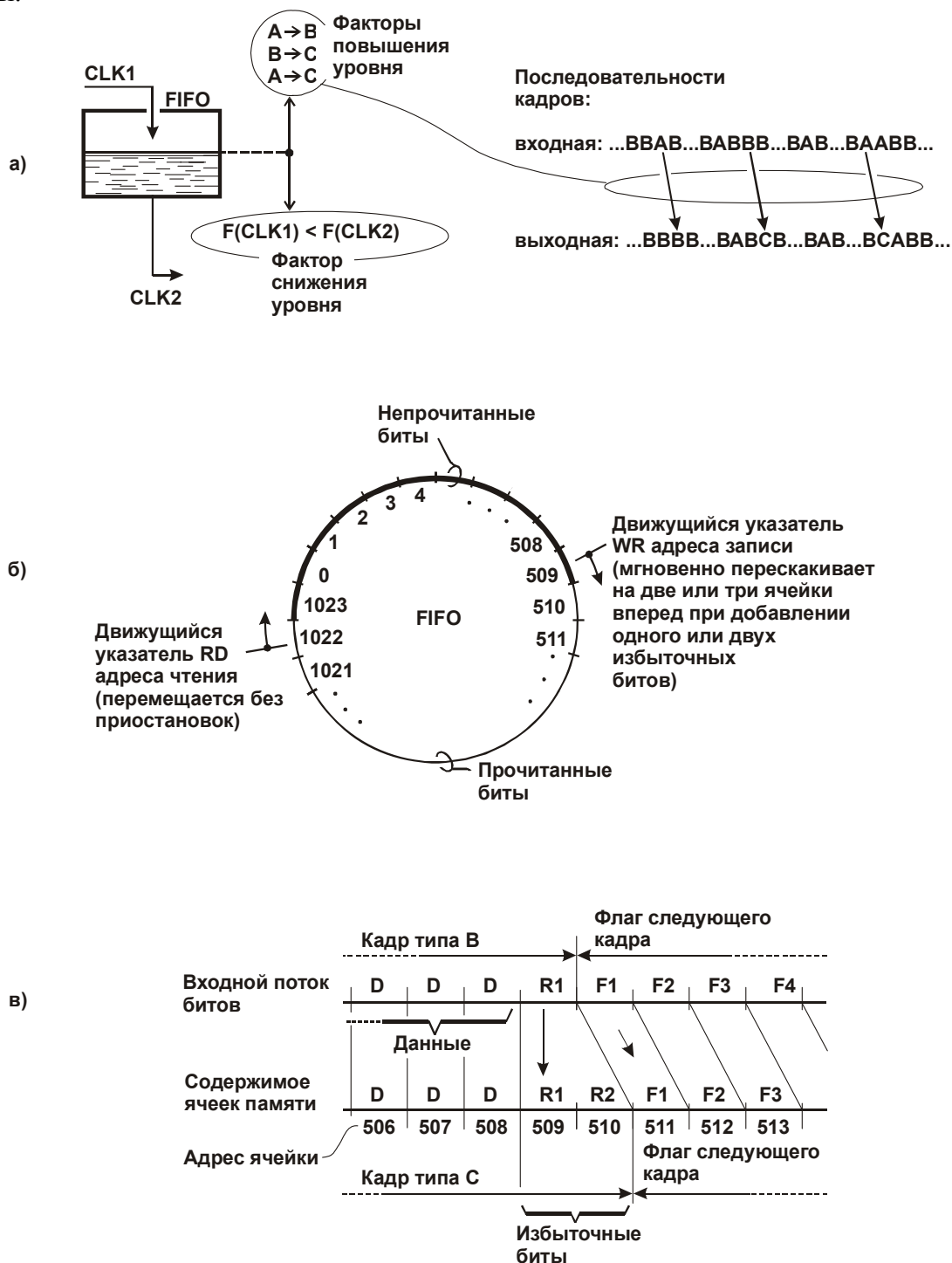


Рис. 2.117. Стабилизация уровня заполнения буферной памяти при отрицательной разности скоростей входного и выходного битовых потоков ( $F(\text{CLK1}) < F(\text{CLK2})$ ): а — общая модель; б — модель динамики заполнения буферной памяти; в — модель процесса вставки одного избыточного бита

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. 2.116, *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. 2.117, *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. 2.1.2). 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.

### 2.29.3. Пример структуры буфера типа FIFO

Буфер типа FIFO может быть выполнен на основе микросхемы оперативного запоминающего устройства (RAM) и счётчиков адресов записи и чтения (Рис. 2.118). При нормальной работе (Рис. 2.119, *а*) счётчики адресов периодически прибавляют единицу к текущему содержимому по сигналам +1 А WR, +1 А RD и формируют циклические последовательности

N-разрядных кодов (см. Рис. 2.116, б, Рис. 2.117, б). Последовательность одноразрядных входных данных поступает на вход DIN микросхемы памяти. Последовательность одноразрядных выходных данных считывается с выхода DOUT этой микросхемы.

Выходы счётчиков адреса записи и чтения соединены соответственно с адресными входами A WR записи и A RD чтения микросхемы памяти. Импульсы синхронизации записи и чтения (SYN WR и SYN RD) подаются на соответствующие управляющие входы микросхемы памяти и задают моменты выполнения этих операций, когда переходные процессы на выходах счётчиков завершаются.

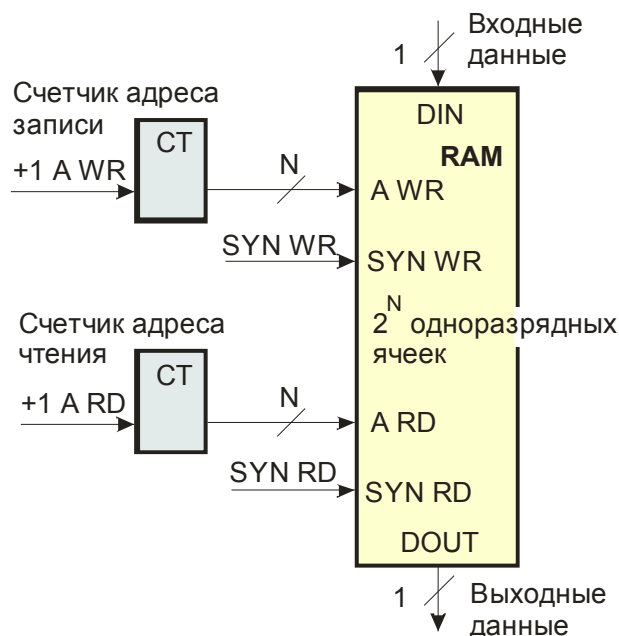
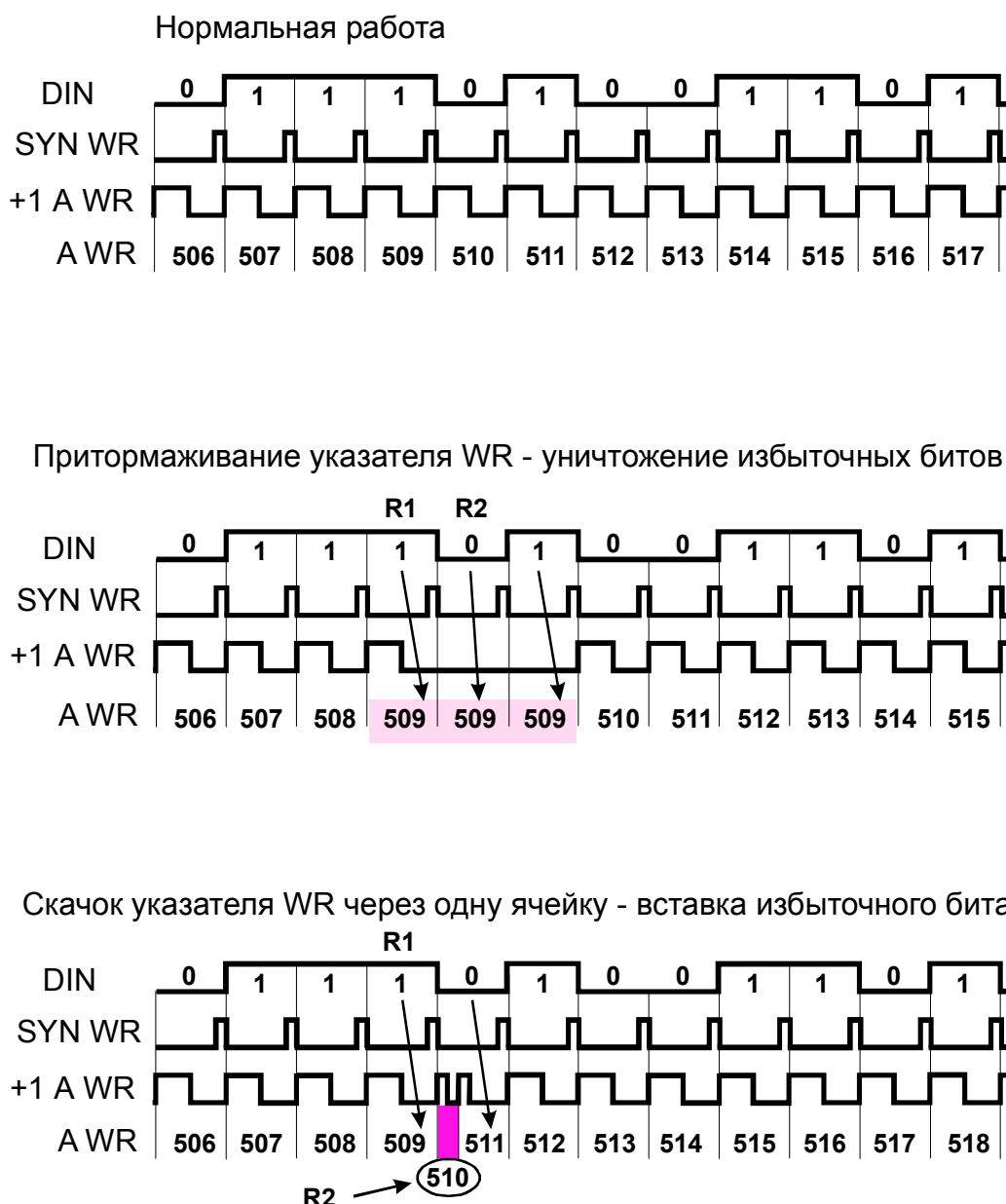


Рис. 2.118. Пример реализации буферной памяти типа FIFO



**Рис. 2.119. Временные диаграммы работы буферной памяти типа FIFO (Рис. 2.118)**

Операции записи и чтения выполняются независимо. Не исключена возможность одновременной записи и чтения бита данных из одной и той же ячейки. Конфликт в данном случае устраняется внутренним приоритетным устройством микросхемы памяти. Напомним, что такие ситуации связаны с переполнением или опустошением буферной памяти (см. Рис. 2.116, Рис. 2.117)

Как отмечалось, для предотвращения переполнения или опустошения буферной памяти необходимо соблюдать достаточную “диаметральную” дистанцию между движущимися указателями адресов записи и чтения. Для этого счётчик адреса записи должен притормаживаться или перескакивать через одно или два состояния в зависимости от соотношения частот синхросигналов передатчика и приёмника данных. Счётчик адреса чтения работает с неизменным темпом.

В соответствии с временными диаграммами, приведенными на Рис. 2.119, б, коррекция состояния счётчика записи в сторону замедления осуществляется кратковременной приостановкой счёта (в данном примере — на два такта) благодаря удалению двух импульсов +1 A WR. Аналогично в данном примере для ускорения счёта формируется лишний импульс +1 A WR (Рис. 2.119, в).

## 2.30. Вопросы и задачи к разделу 2.29

2.30.1. Оцените объём буферной памяти (типа FIFO), предназначенной для уменьшения частоты проскальзываний, при следующих условиях:

- данные, не содержащие избыточных битов, поступают на вход узла сети со скоростью 1 Гбит/с плюс 500 бит/с;
- эти же данные передаются из узла сети со скоростью 1 Гбит/с минус 500 бит/с;
- проскальзывание должно наблюдаться не чаще одного раза в 10 минут.

Каким образом полностью исключить проскальзывания?

Литература: [1], гл. 4

*Решение задачи приведено в разделе 3 на с. 189*

## 2.31. Мультиплексирование разнородных потоков данных (E1, Ethernet, V.35)

Литература к разделу 2.31: [1], [19], [20]

Рассмотрим модель системы передачи данных на основе четырёх мультиплексоров (Рис. 2.120). Два из них (крайние по схеме) имитируют две автоматические телефонные станции — АТС 1 и АТС 2. Два других мультиплексора (1 и 2) объединяют и разделяют данные, обеспечивая двунаправленную передачу данных по следующим маршрутам (выделены на Рис. 2.120, а утолщёнными линиями):

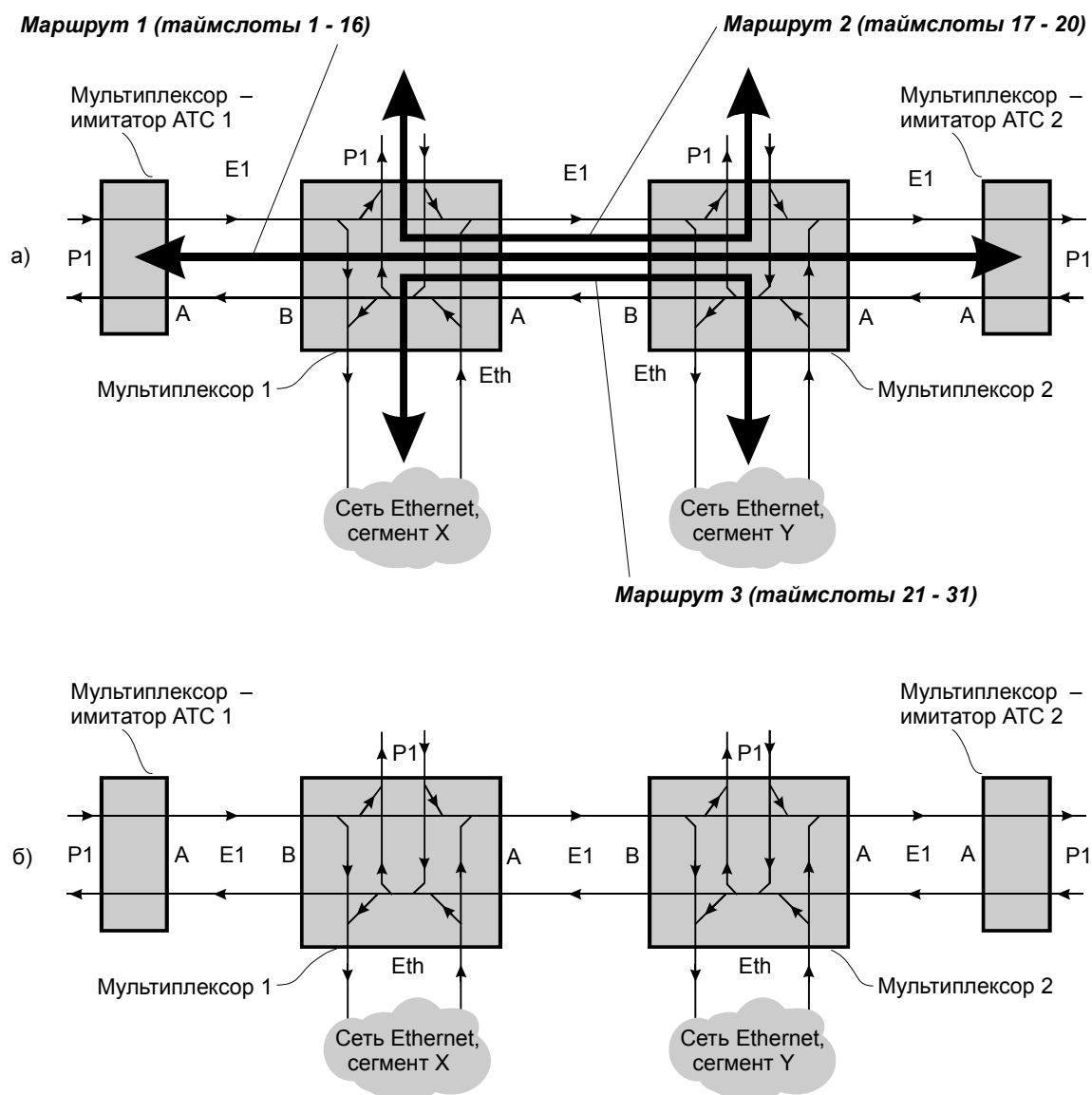


Рис. 2.120. Логическая структура моделируемой системы передачи данных: а — с указанием трёх возможных маршрутов передачи данных; б — то же без указания маршрутов

- Маршрут 1.

Порт А (с интерфейсом E1) автоматической телефонной станции АТС1 ↔ порт А (с интерфейсом E1) автоматической телефонной станции АТС2.

По этому маршруту передаются непрерывные встречные потоки кадров E1 с не полностью задействованным множеством таймслотов (1 — 16). Не задействованные таймслоты (17 — 31) используется для передачи данных по маршрутам 2 и 3.

- Маршрут 2.

*Порт P1 (с интерфейсом V.35) мультиплексора 1 ↔ порт P1 (с интерфейсом V.35) мультиплексора 2.*

По этому маршруту передаются непрерывные встречные потоки битов со скоростями, кратными 64 кбит/с в зависимости от числа задействованных для такой передачи тайм-слотов канала E1 между мультиплексорами 1 и 2. В данном примере использованы четыре таймслота (17 — 20), каждый из них переносит данные со скоростью 64 кбит/с, поэтому скорости прямого и обратного потоков равны  $64 \times 4 = 256$  кбит/с.

- Маршрут 3.

*Порт Eth (с интерфейсом Ethernet) мультиплексора 1 ↔ порт Eth (с интерфейсом Ethernet) мультиплексора 2.*

По этому маршруту по мере необходимости обмена данными между станциями, размещёнными в разных сегментах сети, в ту или иную сторону передаются потоки кадров Ethernet. Для передачи на удалённую сторону эти кадры упаковываются в таймслоты канала E1 между мультиплексорами 1 и 2. Для передачи Ethernet-кадров выделены 10 таймслотов (21 — 31), что соответствует полосе  $64 \times 10 = 640$  кбит/с.

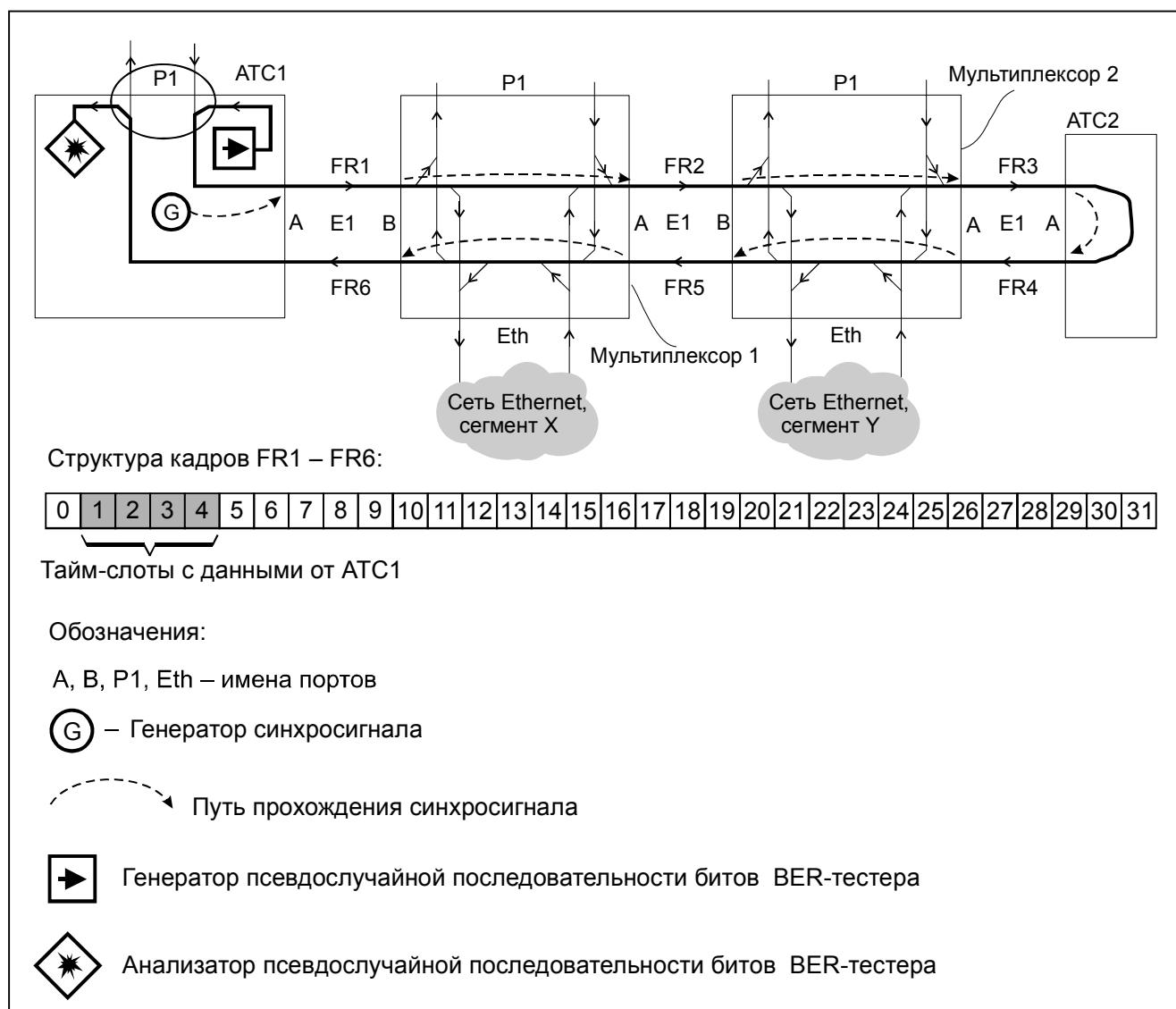
Каждый мультиплексор Рис. 2.120, б содержит порт P1, настраиваемый пользователем на один из интерфейсов семейства “RS” (обычно на интерфейс V.35), порт A (B), подключаемый к каналу E1, и порт Ethernet (Eth). Кроме того, в каждом мультиплексоре имеется порт P2 с интерфейсом RS-232, через который осуществляется его настройка с помощью персонального компьютера (на Рис. 2.120 порты P2 не показаны).

В мультиплексорах — имитаторах ATC1 и ATC2 использованы порты P1 и A. Чтобы не усложнять аппаратуру модели системы передачи, т. е. не вводить в неё внешние источники данных, предполагаем, что доступ к порту P1 осуществляется только “изнутри” мультиплексора, при активизации встроенного в него BER-тестера. Это же относится и к мультиплексорам 1 и 2. В этих мультиплексорах задействованы также порты B и Ethernet. Порты Ethernet, так же как и порты 1, в данном случае доступны только “изнутри” соответствующих мультиплексоров при активизации BER-тестеров.

На Рис. 2.120, б стрелками показаны все возможные направления распространения потоков данных. Разумеется, при каждой конкретной настройке мультиплексоров выбираются (программируются) только вполне определённые направления, которые показаны на поясняющих задачи рисунках (Рис. 2.121 — Рис. 2.126) утолщёнными линиями.

## 2.32. Вопросы и задачи к разделу 2.31

2.32.1. Рассмотрим первый вариант настройки описанной ранее (Рис. 2.120) системы передачи данных (Рис. 2.121).



**Рис. 2.121. Система передачи данных (первый вариант настройки)**

В данном примере передача данных между всеми её компонентами происходит под управлением генератора G синхросигналов, размещённого в мультиплексоре, имитирующем ATC 1. Генераторы G присутствуют в каждом мультиплексоре, но при данной настройке системы активизирован только тот генератор, который находится в ATC 1.

То же относится и к BER-тестеру — он в данном случае активизирован только в ATC 1. Обратите внимание на принятый здесь способ отображения BER-тестера на Рис. 2.121 — Рис. 2.126. Передающая часть BER-тестера (квадрат со стрелкой внутри) в данном случае настроена на выдачу тестовых данных в порт P1. Этот порт, в свою очередь, передаёт данные в порт A. Для такой передачи использованы таймслоты 1 — 4 кадров FR1. Скорость выдачи тестовых данных BER-тестером определяется числом N выделенных для них таймслотов:  $64 \times N$  кбит/с.

Приёмная часть BER-тестера (ромб с “каплей чернил” внутри) настроена на приём тестовых данных из порта P1. Этот порт, в свою очередь, принимает данные из порта A. Эти данные также размещены в таймслотах 1 — 4 принимаемых кадров FR6. Как показано утолщёнными линиями на рисунке, тестовая последовательность битов проходит через мультиплексоры 1, 2,

через петлю возврата данных, установленную в АТС 2, и возвращается в анализатор последовательности битов BER-тестера. При правильной работе системы анализатор не должен регистрировать ошибки.

Оператор может нажатиями кнопки на лицевой панели мультиплексора вводить в поток исходящих из BER-тестера тестовых данных ошибки, нарушающие закономерность построения псевдослучайной последовательности битов. Анализатор должен регистрировать такие ошибки.

Мультиплексоры 1 и 2 настроены на передачу данных из порта В в порт А по “верхней” половине канала Е1 и в обратном направлении по его “нижней” половине. Передаваемые кадры при этом остаются неизменными:  $FR2 = FR1$ ,  $FR3 = FR2$ ,  $FR5 = FR4$ ,  $FR6 = FR5$ . Порты Р1 и Ethernet этих мультиплексоров в данном случае не участвуют в передаче данных.

Мультиплексор, имитирующий АТС 2 настроен так, что выход порта А логически замкнут с его входами ( $FR4 = FR3$ ) через петлю возврата данных. Петля может включаться дистанционно, например, с мультиплексора, имитирующего АТС 1, либо с использованием органов управления, расположенных на мультиплексоре, имитирующем АТС2. Другие порты этого мультиплексора при данной его настройке не используются.

Пути распространения синхросигналов в системе показаны штриховыми стрелками. Так, выходные сигналы портов А мультиплексоров 1 и 2 синхронизируются сигналами, выделенными из входных потоков данных портов В этих же мультиплексоров. Аналогично выходные сигналы портов В мультиплексоров 1 и 2 синхронизируются сигналами, выделенными из входных потоков данных портов А этих мультиплексоров. Выходной сигнал порта А мультиплексора АТС 2 синхронизируется входным сигналом этого же порта. Выходной сигнал порта А мультиплексора АТС 1, как уже отмечалось, синхронизируется сигналом от внутреннего генератора G.

Вопросы по схеме, приведенной на Рис. 2.121:

1. Передаваемые и принимаемые BER-тестером данные не совпадают во времени из-за задержки передачи тестового сигнала по цепи из мультиплексоров. Сказывается ли это на работе анализатора BER-тестера? Существует ли максимально допустимая задержка?
2. Какова скорость передачи тестовых данных в каналах Е1?
3. Как нужно изменить настройки мультиплексоров (показать общую идею без её воплощения в настройки), чтобы увеличить (уменьшить) скорость передачи тестовых данных? Каковы пределы изменения этой скорости?

Литература: разделы 2.9, 2.25, 2.27, 2.29, 2.31 настоящей книги, [1], [19], [20].

*Ответы приведены в разделе 3 на с. 189*

2.32.2. Вопрос по схеме, приведенной на Рис. 2.122:

В отличие от предыдущей схемы (Рис. 2.121), в данном случае в системе передачи данных активизированы BER-тестеры, расположенные в мультиплексорах — имитаторах АТС.

В чём заключаются преимущества и недостатки схемы, представленной на Рис. 2.122, по сравнению со схемой, представленной на Рис. 2.121, в части полноты тестирования и удобства использования? Предполагаем, что ошибки возникают только в линиях связи; их интенсивность невелика и не приводит к нарушению кадровой синхронизации.

Литература: разделы 2.9, 2.25, 2.27, 2.29, 2.31 настоящей книги, [1], [19], [20].

*Ответ приведен в разделе 3 на с. 189*

2.32.3. Вопрос по схеме, приведенной на Рис. 2.123

В отличие от предыдущей схемы (Рис. 2.122), каждое направление передачи данных синхронизируется отдельным генератором синхросигналов (G1, G2).

В чём заключается преимущество схемы, представленной на Рис. 2.123, по сравнению со схемой, представленной на Рис. 2.122, в части локализации ошибок? Предполагаем, что ошибки возникают только в линиях связи; их интенсивность невелика и не приводит к нарушению кадровой синхронизации.

Литература: разделы 2.9, 2.25, 2.27, 2.29, 2.31 настоящей книги, [1], [19], [20].

*Ответ приведен в разделе 3 на с. 189*

#### 2.32.4. Вопрос по схеме, приведенной на Рис. 2.124

Система передачи данных (Рис. 2.124) проверяется одновременно тремя BER-тестерами. Потоки тестовых данных суммируются по мере перехода от кадра FR1 к кадру FR3, который без изменения переходит в кадр FR4. При переходе от кадра FR4 к кадру FR6 из него извлекаются тестовые данные, адресованные анализаторам BER-тестеров мультиплексоров 1 и 2. Кадр FR6 содержит тестовые данные, адресованные анализатору BER-тестера мультиплексора — имитатора АТС1. Предполагаем, что ошибки возникают только в линиях связи; их интенсивность невелика и не приводит к нарушению кадровой синхронизации.

Имеются ли преимущества схемы (Рис. 2.124) по сравнению с предыдущими (Рис. 2.121 — Рис. 2.123) в отношении полноты и удобства тестирования?

Литература: разделы 2.9, 2.25, 2.27, 2.29, 2.31 настоящей книги, [1], [19], [20].

*Ответ приведен в разделе 3 на с. 189*

#### 2.32.5. Вопросы по схеме, приведенной на Рис. 2.125

В схеме, показанной на Рис. 2.125, задействованы все имеющиеся BER-тестеры. Кадр FR1 содержит тестовые данные от АТС 1, кадр FR2, в дополнение к этим данным, тестовые данные от Ethernet-порта мультиплексора 1. В кадре FR3 содержатся только те данные, которые поступают от АТС 1. Кадр FR4 содержит тестовые данные от АТС 2, кадр FR5, в дополнение к этим данным, тестовые данные от Ethernet-порта мультиплексора 2. В кадре FR6 содержатся только те данные, которые поступают от АТС 2. Предполагаем, что ошибки возникают только в линиях связи; их интенсивность невелика и не приводит к нарушению кадровой синхронизации.

Вопросы:

1. В чём состоит преимущество этой схемы тестирования по сравнению с предыдущими (Рис. 2.121 — Рис. 2.124)?
2. Можно ли совместить передачу тестовых данных с обменом данными между сегментами сети Ethernet?
3. Можно ли совместить передачу тестовых данных с обменом данными между портами Р1 АТС 1 и АТС 2?
4. Можно ли совместить передачу тестовых данных с обменом данными между портами Р1 мультиплексоров 1 и 2?

Литература: разделы 2.9, 2.25, 2.27, 2.29, 2.31 настоящей книги, [1], [19], [20].

*Ответы приведены в разделе 3 на с. 190*

#### 2.32.6. Вопросы по схеме, приведенной на Рис. 2.126

В схеме, показанной на Рис. 2.126, активизированы BER-тестеры мультиплексоров — имитаторов АТС. Передача данных из АТС1 в АТС2 происходит под управлением разных генераторов G, поэтому из-за неидеального совпадения частот синхросигналов автоматически включается эластичная память (буфер

типа FIFO), до некоторой степени снижающая частоту проскальзываний.

Предполагаем, что ошибки, не связанные с проскальзываниями, могут возникать только в линиях связи; их интенсивность невелика и не приводит к нарушению кадровой синхронизации.

Вопросы:

1. Как работает эластичная память?
2. Удаётся ли полностью устранить проскальзывания в данной системе передачи?
3. На каком участке трассы передачи данных наблюдаются проскальзывания?
4. Возможны ли проскальзывания при обмене данными между сегментами сети Ethernet?

Литература: разделы 2.9, 2.25, 2.27, 2.29, 2.31 настоящей книги, [1], [19], [20].

Ответы приведены в разделе 3 на с. 190

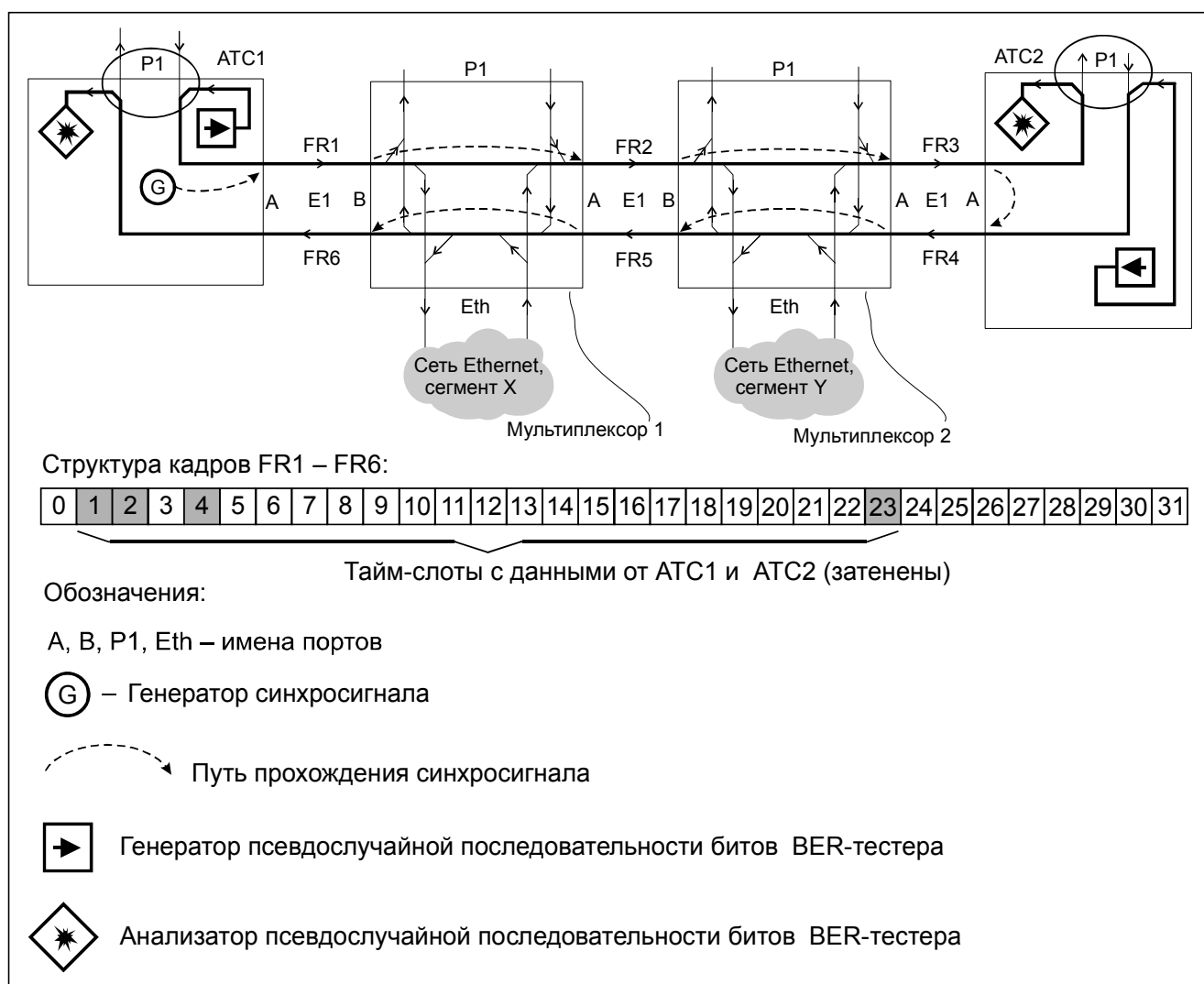


Рис. 2.122. Система передачи данных (второй вариант настройки)

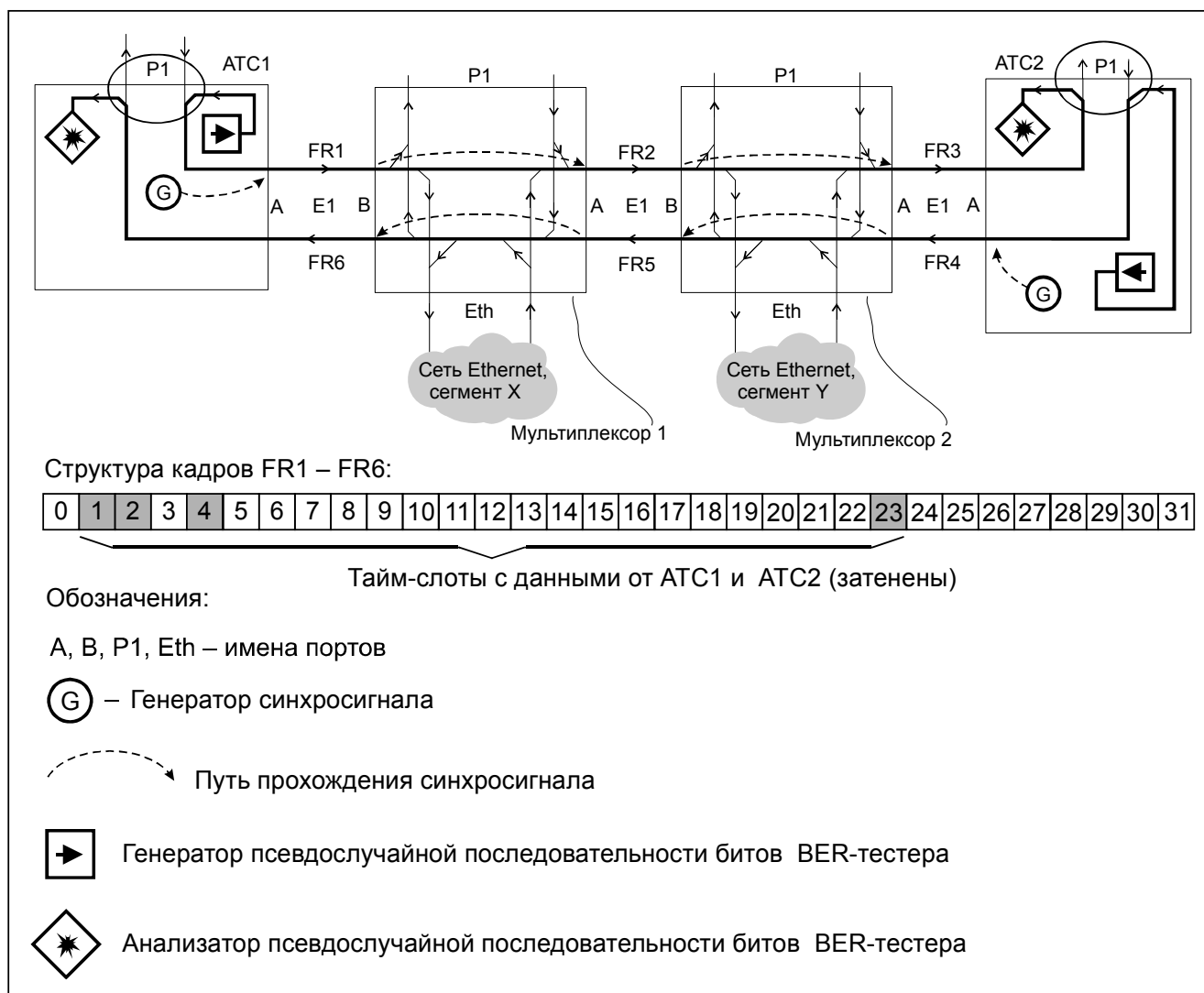


Рис. 2.123. Система передачи данных (третий вариант настройки)

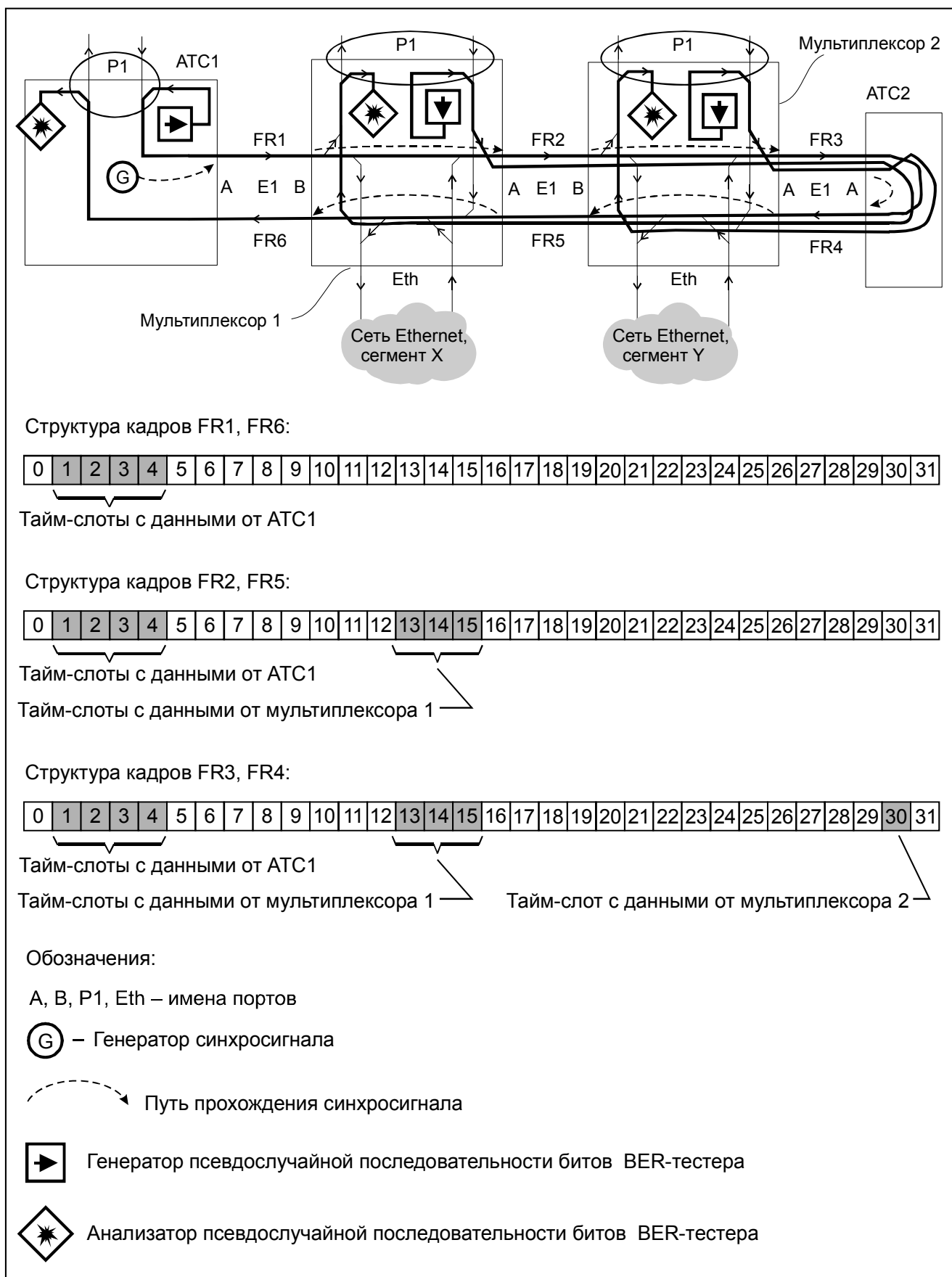


Рис. 2.124. Система передачи данных (четвёртый вариант настройки)

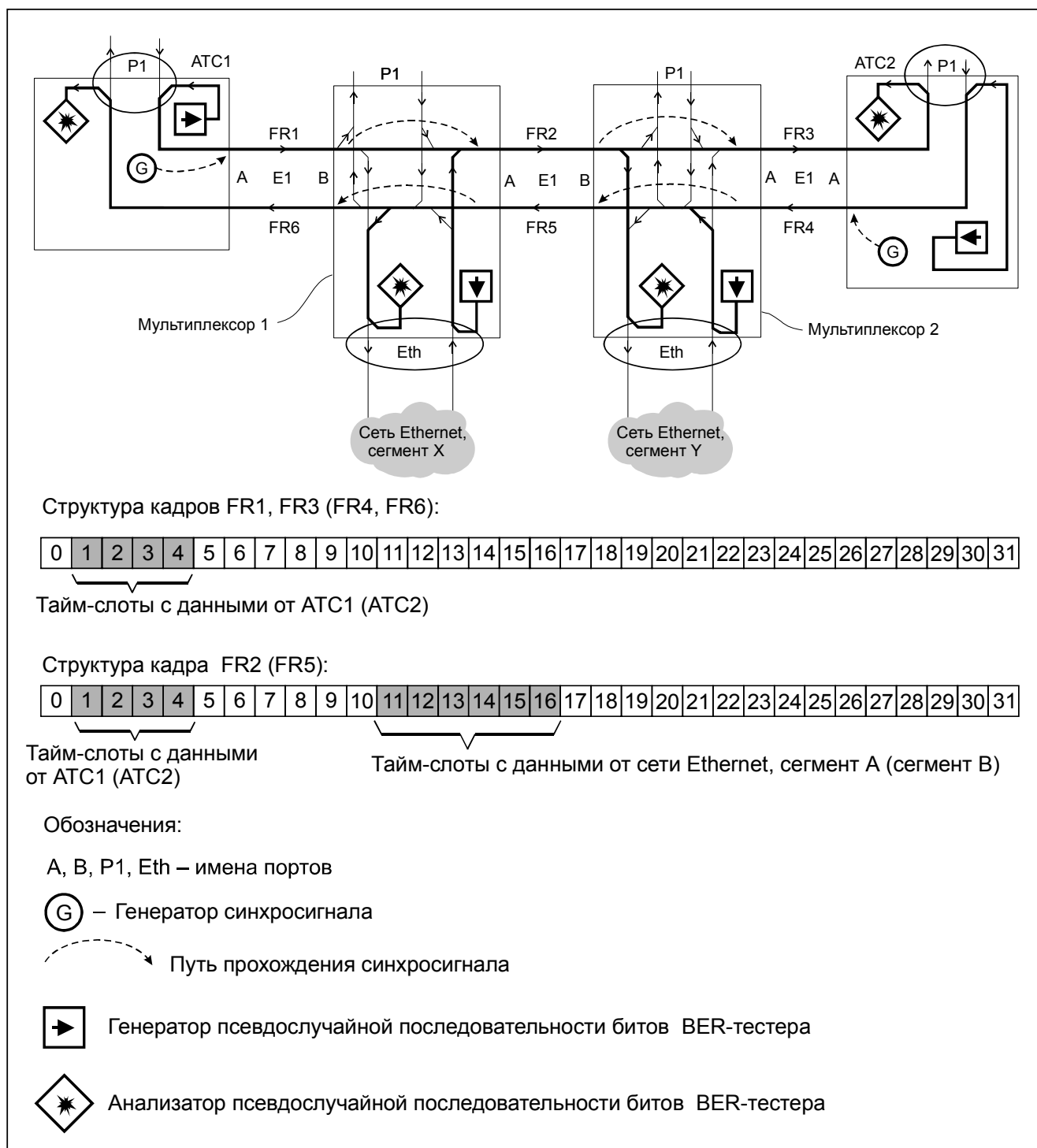


Рис. 2.125. Система передачи данных (пятый вариант настройки)

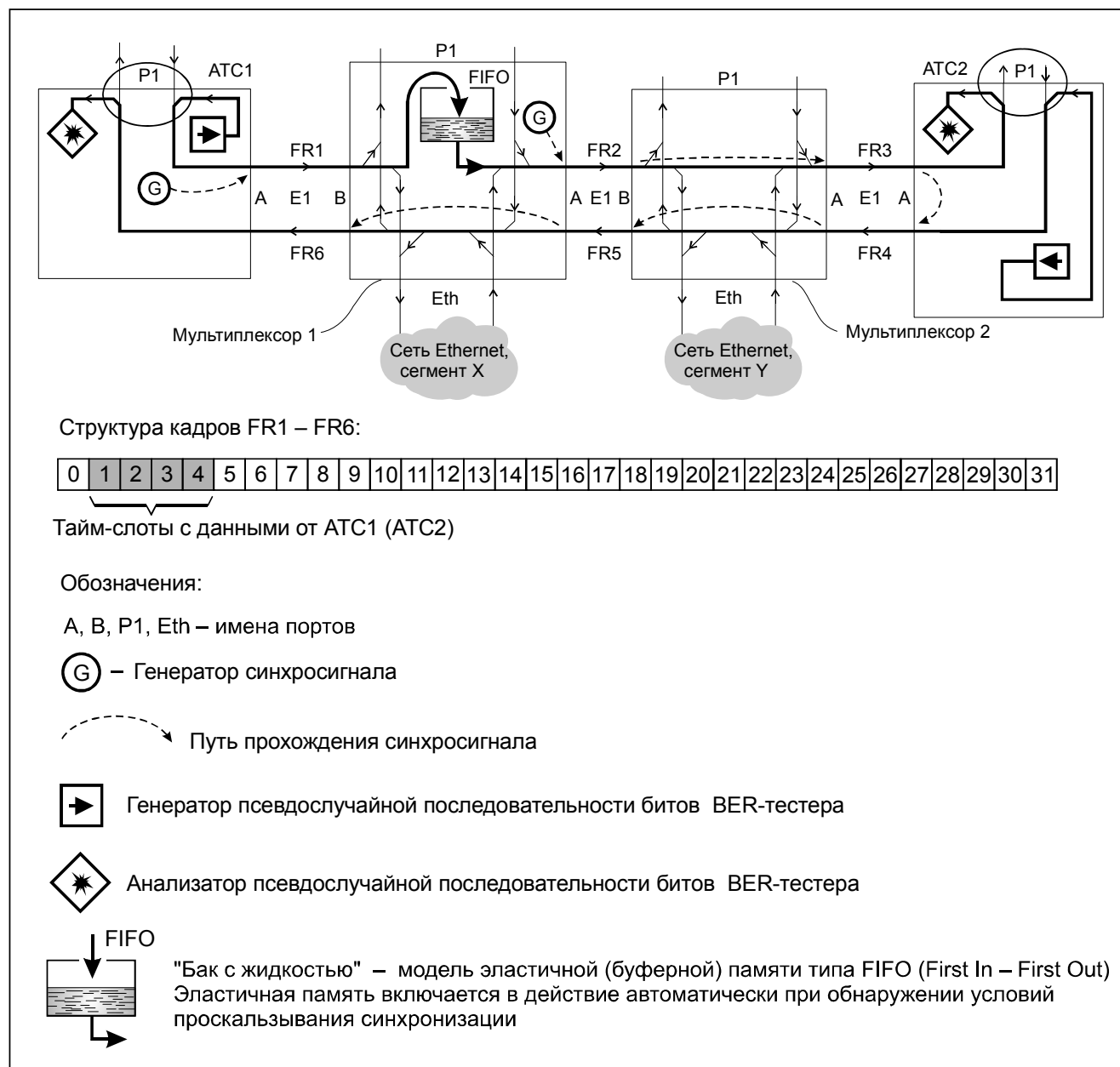
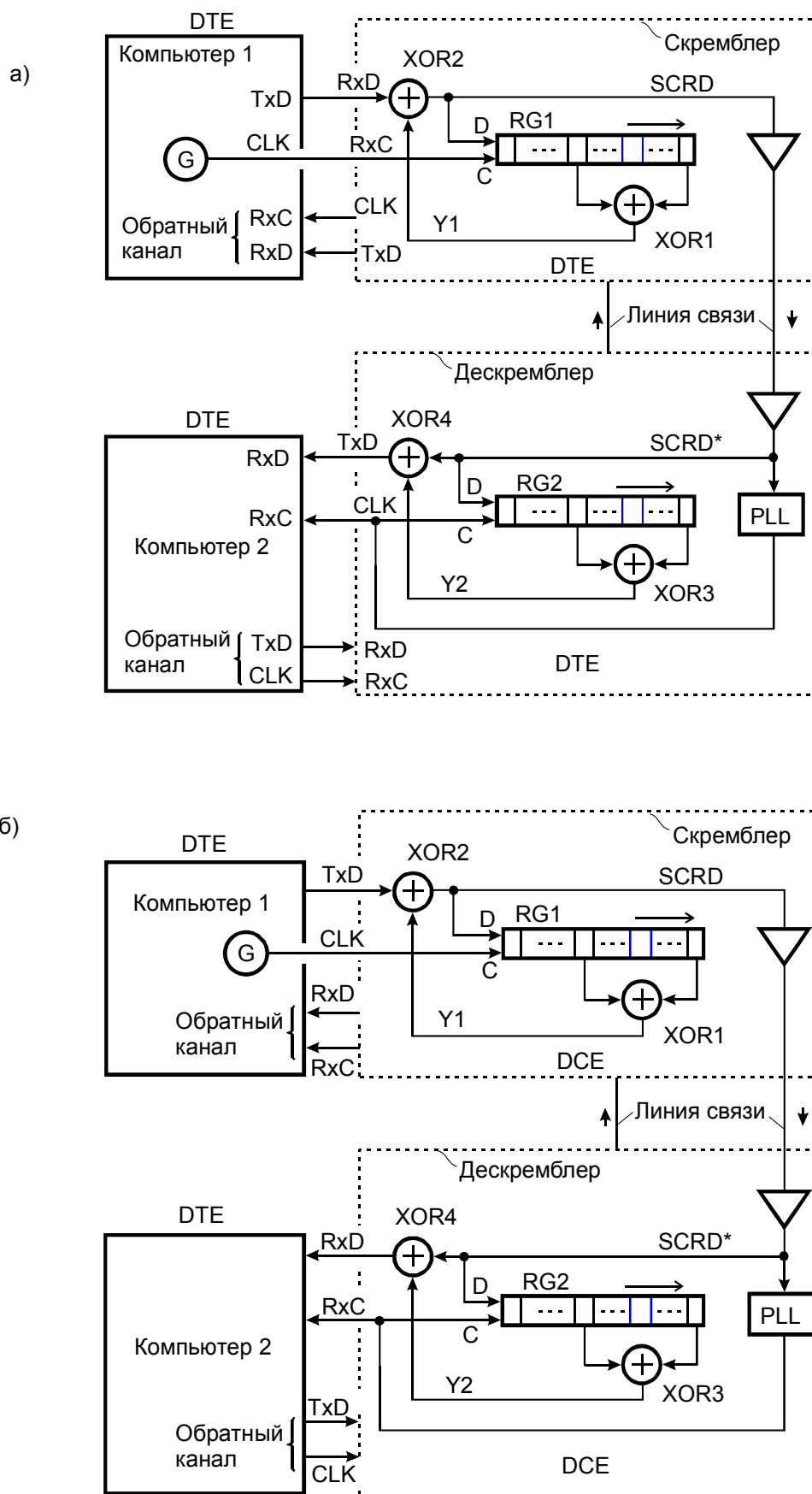


Рис. 2.126. Система передачи данных (шестой вариант настройки)

### 3. Ответы и решения

Решение задачи, приведенной в п. 2.2.1 (с. 22)

Варианты построения системы передачи данных показаны на Рис. 3.1, а — г.



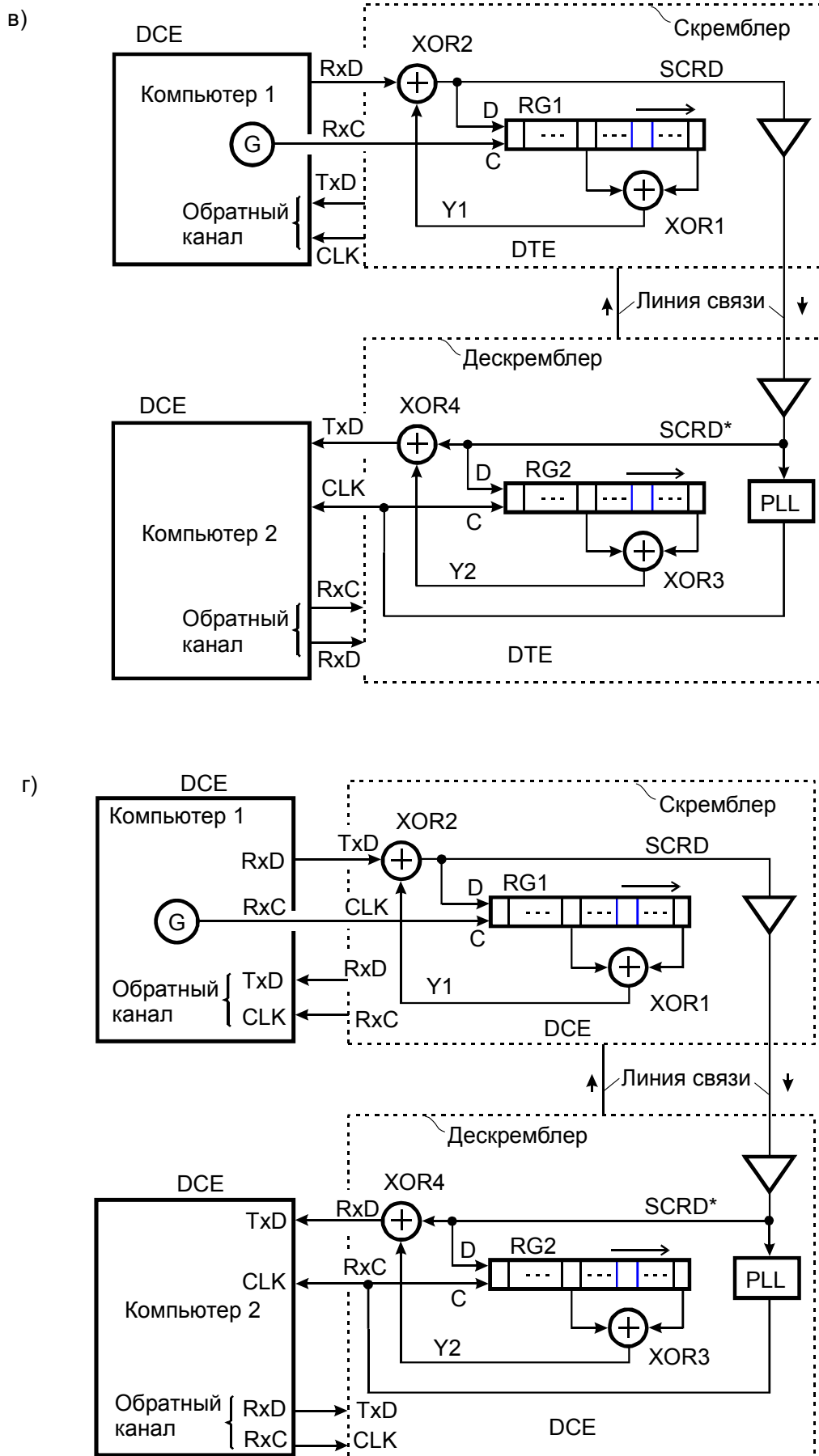
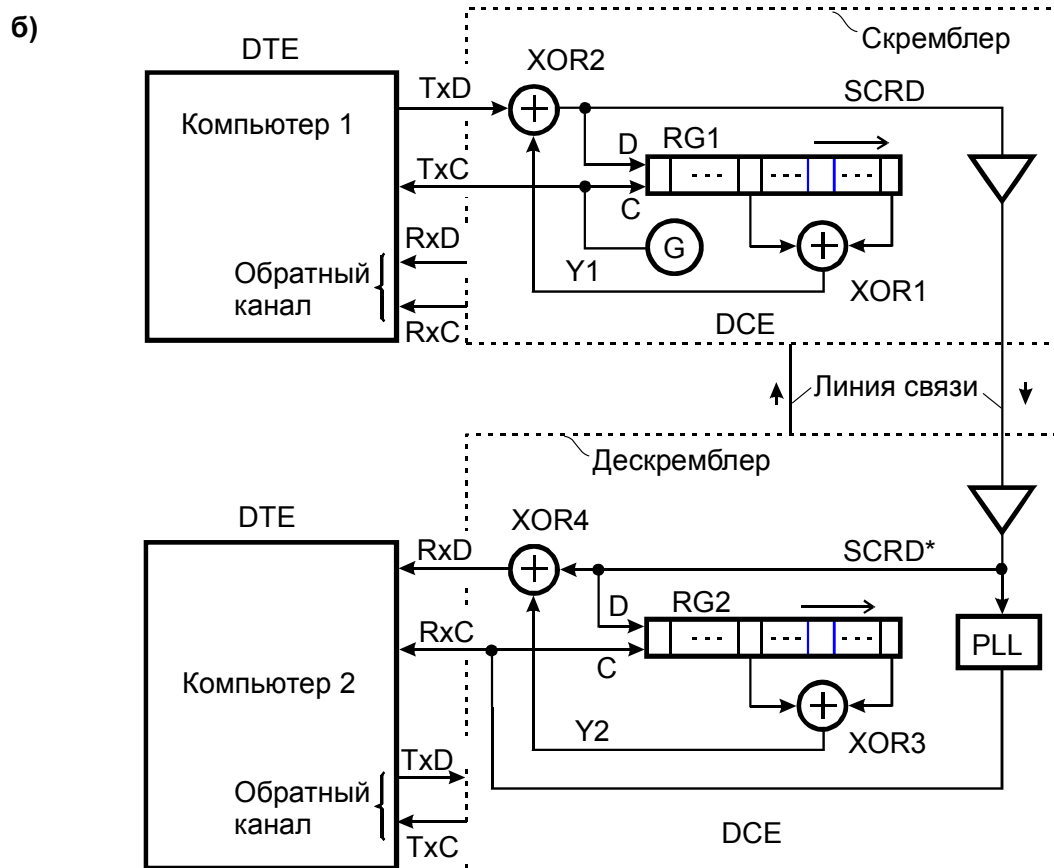


Рис. 3.1. Варианты а — г построения системы передачи данных. Показаны типы (DTE, DCE) составных частей системы и приведены наименования сигналов

Решение задачи, приведенной в п. 2.2.2 (с. 22)

Варианты построения системы передачи данных описаны на Рис. 3.2, а — з.

а) Этот вариант не реализуется из-за отсутствия возможности встречной синхронизации



в) Этот вариант не реализуется из-за отсутствия возможности встречной синхронизации

г) Этот вариант не реализуется из-за отсутствия возможности встречной синхронизации

**Рис. 3.2. Оценка вариантов построения системы передачи данных**

Ответы на вопросы, приведенные в п. 2.2.3 (с. 24)

**Варианты а и б** правильные, остальные — нет:

**вариант в** не содержит стабилизированного генератора синхросигнала; синхронизация неустойчива, так как синхросигнал формируется в результате совместной работы двух не стабилизированных генераторов PLL с фазовой автоподстройкой частоты, которые постоянно пытаются подстроиться друг к другу;

**вариант з** не обеспечивает устойчивого приёма данных компьютером 1, так как соотношение фаз сигналов RxС и RxD на его входах зависит от длины линии и других неуправляемых параметров системы;

**вариант д** неработоспособен из-за того, что передача данных из компьютера 1 и их приём компьютером 2 синхронизируются от разных генераторов G.

### Решение задачи, приведенной в п. 2.2.4 (с. 29)

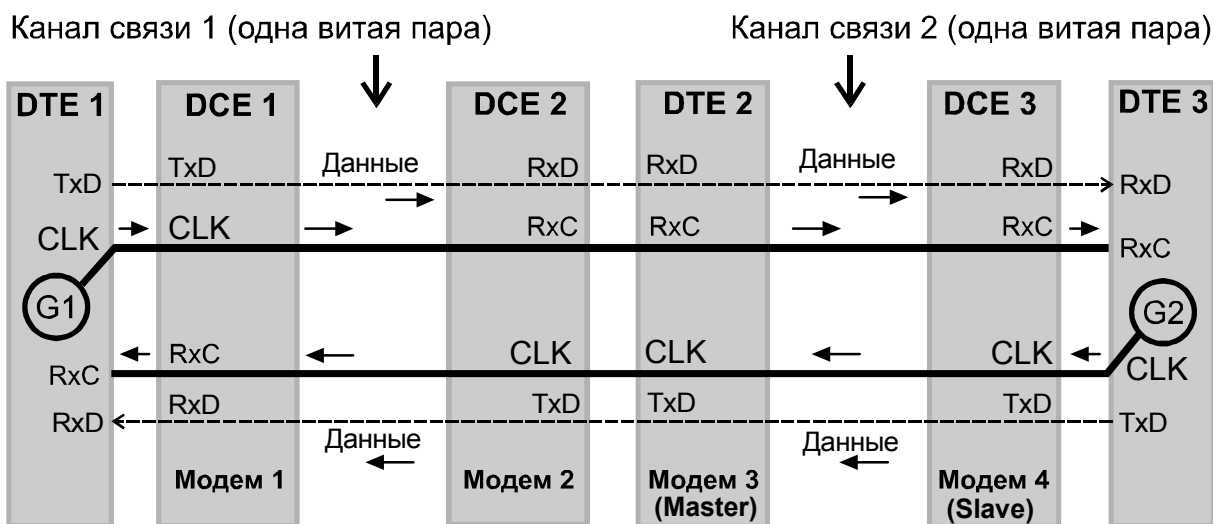
В первом режиме 9-битовая информационная посылка содержит стартовый бит, семь битов данных и стоп-бит. Доля полезных данных:  $7/9 \approx 0,78$ . Доля служебных данных:  $2/9 \approx 0,22$ . Максимальная скорость передачи полезных данных в первом режиме равна  $38,4 \times 0,78 \approx 30$  кбит/с.

Во втором режиме 12-битовая информационная посылка содержит стартовый бит, восемь битов данных, два стоп-бита, а также бит контроля по чётности (нечётности). Доля полезных данных:  $8/12 \approx 0,67$ . Доля служебных данных:  $4/12 \approx 0,33$ . Максимальная скорость передачи полезных данных во втором режиме равна  $38,4 \times 0,67 \approx 25,6$  кбит/с.

Таким образом, максимальная скорость передачи пользовательских данных в первом режиме более высокая, чем во втором.

### Решение задачи, приведенной в п. 2.2.5 (с. 29)

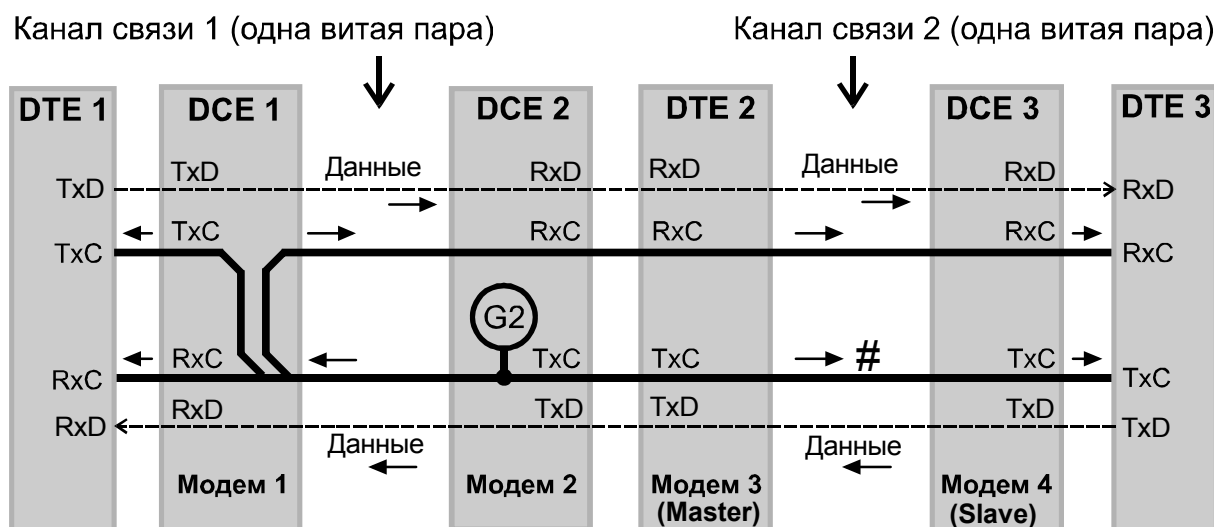
В соответствии с определениями сигналов и правилами соединения устройств DTE и DCE ([1], гл. 1 — 3) после заданного в условиях задачи преобразования получим систему, показанную на Рис. 3.3. Способы передачи синхросигнала против течения потока данных также описаны в гл. 1 — 3 указанной книги [1]. Один из способов поясняется Рис. 3.21.



**Рис. 3.3. Система передачи данных с двумя последовательно включёнными каналами связи после преобразования в соответствии с условиями задачи 2.2.5**

### Решение задачи, приведенной в п. 2.2.6 (с. 30)

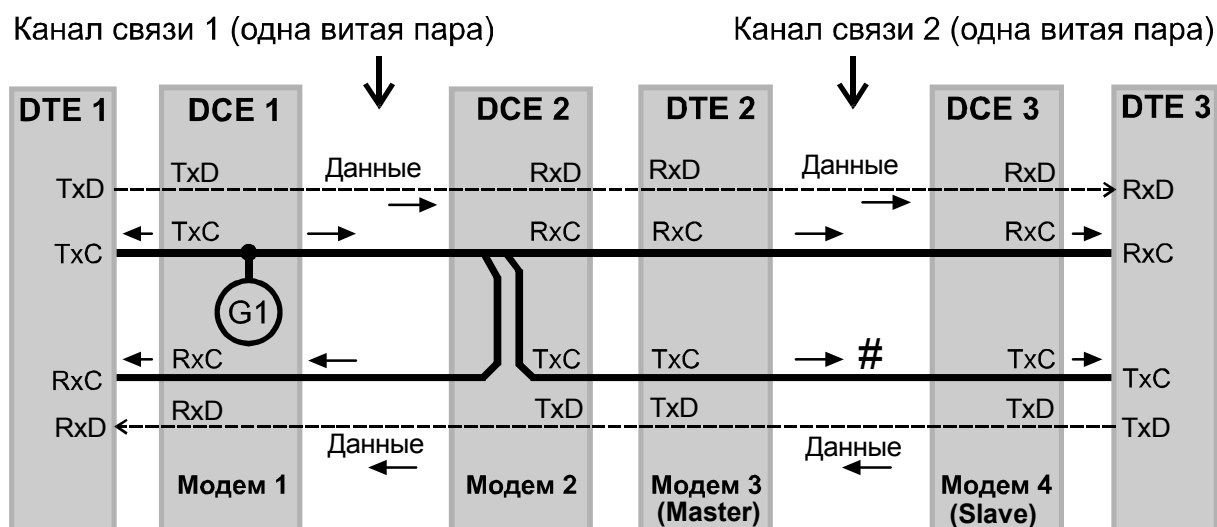
В соответствии с определениями сигналов и правилами соединения устройств DTE и DCE ([1], гл. 1 — 3) после заданного в условиях задачи преобразования получим систему, показанную на Рис. 3.4. Способы передачи синхросигнала против течения потока данных также описаны в гл. 1 — 3 указанной книги [1]. Один из способов поясняется Рис. 3.21.



**Рис. 3.4. Система передачи данных с двумя последовательно включёнными каналами связи после преобразования в соответствии с условиями задачи 2.2.6**

Решение задачи, приведенной в п. 2.2.7 (с. 30)

В соответствии с определениями сигналов и правилами соединения устройств DTE и DCE ([1], гл. 1 — 3) после заданного в условиях задачи преобразования получим систему, показанную на Рис. 3.5. Способы передачи синхросигнала против течения потока данных также описаны в гл. 1 — 3 указанной книги [1]. Один из способов поясняется Рис. 3.21.



**Рис. 3.5. Система передачи данных с двумя последовательно включёнными каналами связи после преобразования в соответствии с условиями задачи 2.2.7**

Решение задачи, приведенной в п. 2.2.8 (с. 31)

В соответствии с определениями сигналов и правилами соединения устройств DTE и DCE ([1], гл. 1 — 3) после заданного в условиях задачи преобразования получим систему, показанную на Рис. 3.6. Способы передачи синхросигнала против течения потока данных также описаны в гл. 1 — 3 указанной книги [1]. Один из способов поясняется Рис. 3.21.

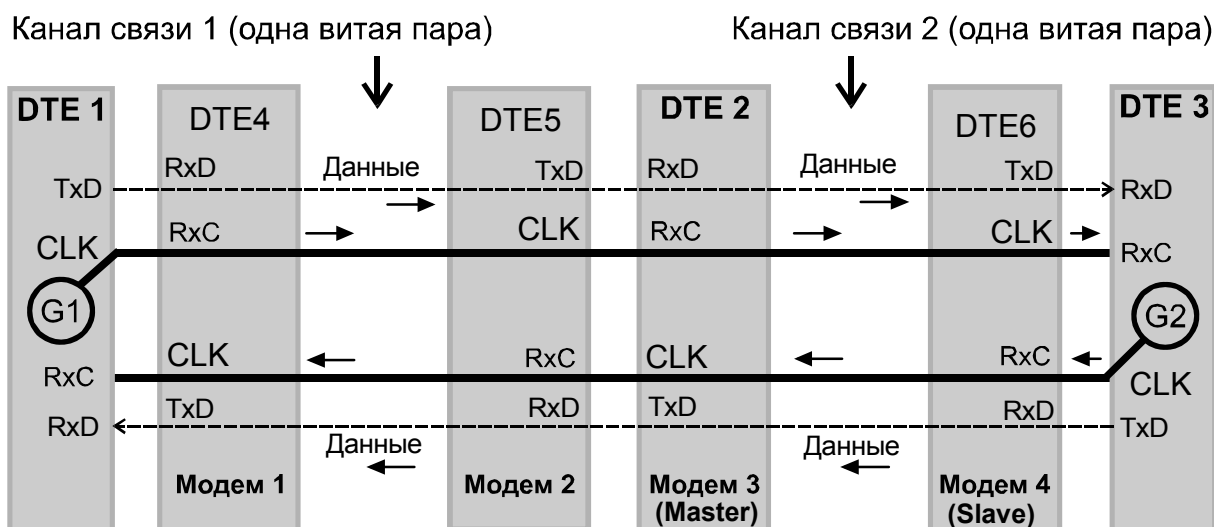


Рис. 3.6. Система передачи данных с двумя последовательно включёнными каналами связи после преобразования в соответствии с условиями задачи 2.2.8

Решение задачи, приведенной в п. 2.2.9 (с. 32)

В соответствии с определениями сигналов и правилами соединения устройств DTE и DCE ([1], гл. 1 — 3) после заданного в условиях задачи преобразования получим систему, показанную на Рис. 3.7. Способы передачи синхросигнала против течения потока данных также описаны в гл. 1 — 3 указанной книги [1]. Один из способов поясняется Рис. 3.21.

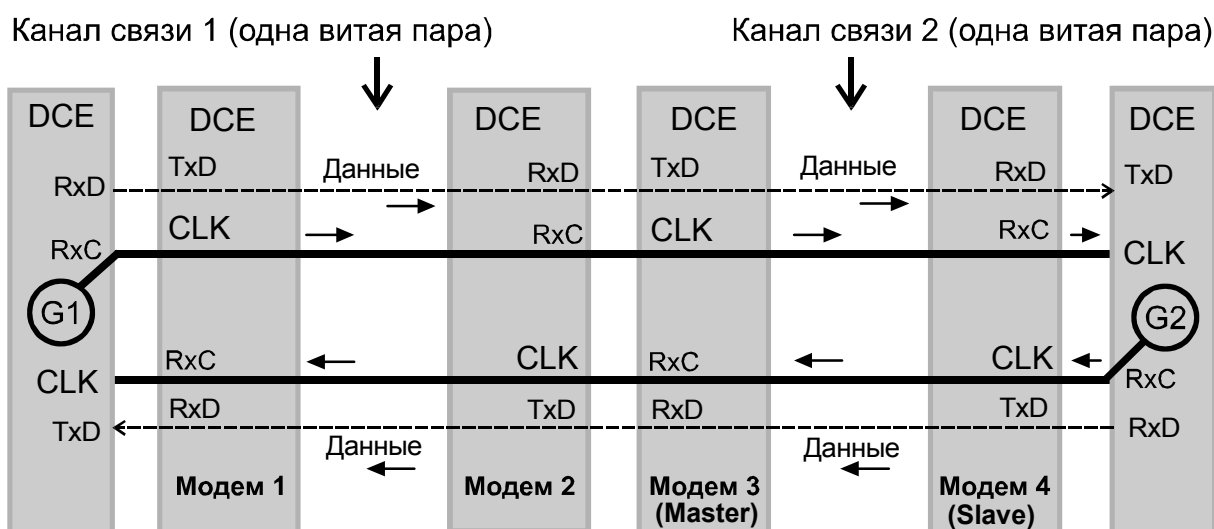


Рис. 3.7. Система передачи данных с двумя последовательно включёнными каналами связи после преобразования в соответствии с условиями задачи 2.2.9

Решение задачи, приведенной в п. 2.2.10 (с. 32)

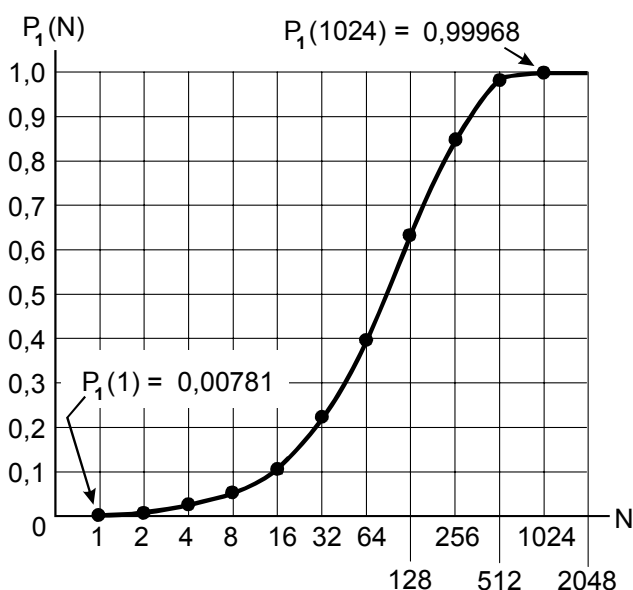
Если длина цепи равна одной посылке ( $N = 1$ ), то вероятность  $P_1(N)$  гарантированного установления синхронизации равна вероятности обнаружения в этой посылке нулевого кода ( $1/128$ ). Вероятность  $P_2(N)$  не установления синхронизации равна  $1 - 1/128 \approx 0,992$ .

Если длина цепи равна двум посылкам ( $N = 2$ ), то вероятность не установления синхронизации после её прохождения по линии равна  $P_2(2) = (1 - 1/128)^2 \approx 0,984$ . Эта величина убывает с возрастанием длины  $N$  цепи:  $P_2(N) = (1 - 1/128)^N$ . В свою очередь, вероятность  $P_1(N)$  установления синхронизации  $P_1(N) = 1 - P_2(N)$  растёт с увеличением  $N$ . Результаты расчётов приведены в табл. 4 и на Рис. 3.8.

**Таблица 4. Результаты расчёта вероятностей неустановления  $P_2(N)$  и установления  $P_1(N)$  синхронизации за  $N$  шагов**

| $N$  | $P_2(N)$ | $P_1(N)$ |
|------|----------|----------|
| 1    | 0,99219  | 0,00781  |
| 2    | 0,98444  | 0,01556  |
| 4    | 0,96911  | 0,03089  |
| 8    | 0,93918  | 0,06082  |
| 16   | 0,88206  | 0,11794  |
| 32   | 0,77804  | 0,22196  |
| 64   | 0,60534  | 0,39466  |
| 128  | 0,36644  | 0,63356  |
| 256  | 0,13428  | 0,86572  |
| 512  | 0,01803  | 0,98197  |
| 1024 | 0,00032  | 0,99968  |

Таким образом, после прохождения цепи из 1024 посылок синхронизация будет достигнута с вероятностью 0,99968. Это пессимистическая оценка, так как она не учитывает, что посылки, содержащие код 1111111, также обеспечивают вхождение в синхронизацию в случае её потери ([1], гл. 3).



**Рис. 3.8. Зависимость вероятности  $P_1(N)$  гарантированного установления синхронизации от длины  $N$  случайной последовательности 7-битовых слов данных**

Решение задачи, приведенной в п. 2.4.1 (с. 38)

Рассмотрим код NRZ как наиболее простой. Если в обоих случаях перепад уровней сигнала составляет  $U$  [В], то при передаче случайного потока данных и однополярном кодировании (при котором логическая единица соответствует напряжению  $+U$ , а логический нуль — отсутствию напряжения) средняя мощность, выделяемая на нагрузочном резисторе  $R$ , равна  $P_1 = U^2/2R$ . То же при биполярном кодировании (когда лог. 0 и лог.1 представлены напряжениями  $U/2$  разной полярности):  $P_2 = (U/2)^2/R = U^2/4R$ . Выигрыш в два раза ( $P_1 = 2P_2$ ).

Решение задачи, приведенной в п. 2.4.2 (с. 38)

Сигнал “Манчестер-II” характеризуется тем, что в центре каждого битового интервала должен наблюдаться перепад его уровня, положительный (лог. 0) или отрицательный (лог. 1). Этому условию удовлетворяет разделение сигнала на битовые интервалы в соответствии с Рис. 2.32, г. Передаваемый код: 010011111011111010.

**Решение задачи, приведенной в п. 2.4.3 (с. 38)**

Сигнал “Манчестер-II” характеризуется тем, что в центре каждого битового интервала должен наблюдаться перепад его уровня, положительный (лог. 0) или отрицательный (лог. 1).

После разделения сигнала на битовые интервалы (Рис. 3.9) выясняется, что в одном из них (выделенном серым фоном) сигнал не изменяется, что неправильно:

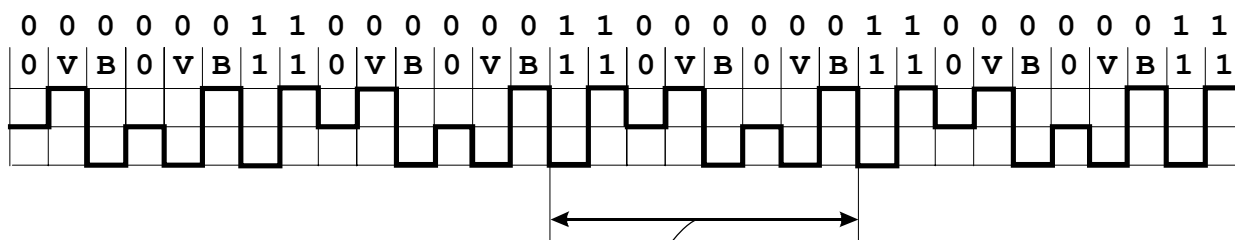


**Рис. 3.9. Сигнал “Манчестер-II” после выделения в нём битовых интервалов**

Передаваемый код: 010011111011111x010, символ “x” отображает неопределённый бит (0 или 1).

**Решение задачи, приведенной в п. 2.4.4 (с. 38)**

В коде B6ZS каждые шесть последовательных нулей подменяются комбинацией 0VB0VB. Согласно этому правилу, сигнал имеет форму, показанную на Рис. 3.10.



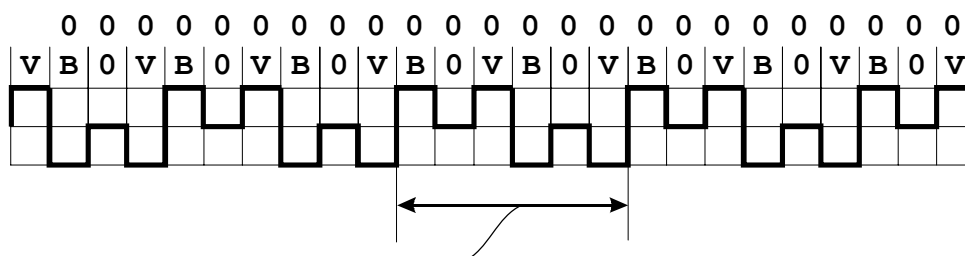
Период повторения сигнала – 8 битовых интервалов

**Рис. 3.10. Сигнал B6ZS**

**Решение задачи, приведенной в п. 2.4.5 (с. 38)**

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

В соответствии с этими правилами в данном случае выбирается “заготовка” B0V, как показано на Рис. 3.11.



Период повторения сигнала – 6 битовых интервалов

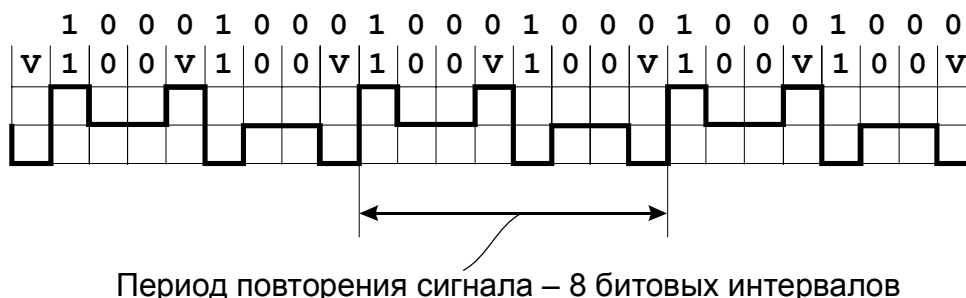
**Рис. 3.11. Сигнал B3ZS**

Диаграмма сигнала B6ZS, соответствующая передаче длинной цепочки лог. 0, совпадает с диаграммой, приведенной на Рис. 3.11.

#### Решение задачи, приведенной в п. 2.4.6 (с. 39)

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

В соответствии с этими правилами в данном случае выбирается “заготовка” 00V, как показано на Рис. 3.12.

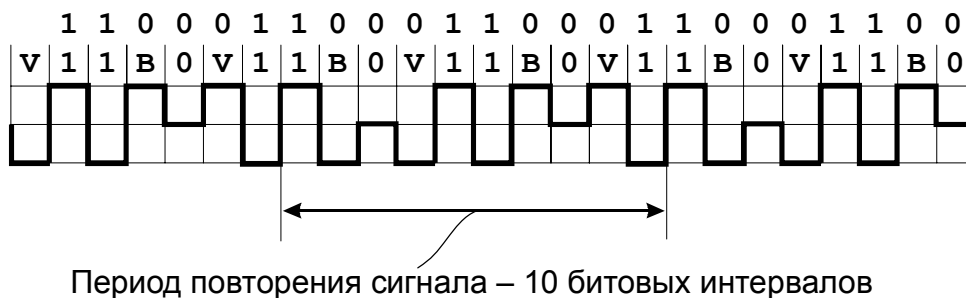


**Рис. 3.12. Сигнал B3ZS**

#### Решение задачи, приведенной в п. 2.4.7 (с. 39)

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

В соответствии с этими правилами в данном случае выбирается “заготовка” B0V, как показано на Рис. 3.13.



**Рис. 3.13. Сигнал B3ZS**

#### Решение задачи, приведенной в п. 2.4.8 (с. 39)

В коде HDB3 каждые четыре последовательных нуля подменяются либо комбинацией B00V, либо 000V. Символ B обозначает импульс, который отвечает правилам кодирования АМІ, символ V — импульс, который нарушает правила кодирования АМІ (совпадает по полярности с предыдущим). Выбор одной из этих двух “заготовок” проводится так, чтобы, во-первых, число импульсов B между двумя последовательно расположенными импульсами V было нечетным, и, во-вторых, чтобы полярность импульсов V чередовалась.

В соответствии с этими правилами в данном случае выбирается “заготовка” B00V, как показано на Рис. 3.14.

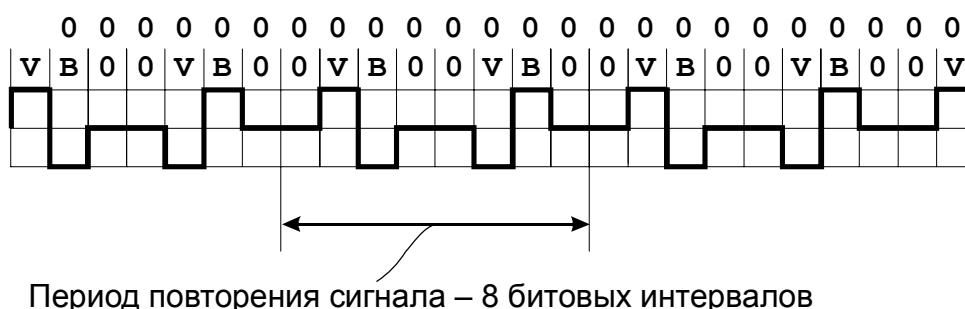


Рис. 3.14. Сигнал HDB3

Решение задачи, приведенной в п. 2.4.9 (с. 39)

В коде HDB3 каждые четыре последовательных нуля подменяются либо комбинацией B00V, либо 000V. 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 соответствии с этими правилами в данном случае выбирается “заготовка” 000V, как показано на Рис. 3.15.

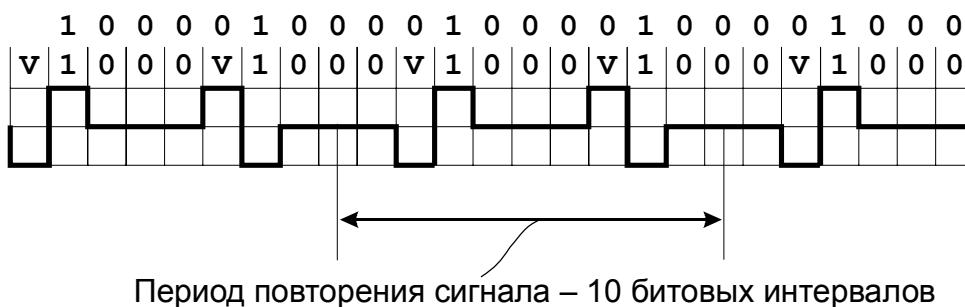


Рис. 3.15. Сигнал HDB3

Решение задачи, приведенной в п. 2.4.10 (с. 39)

В коде HDB3 каждые четыре последовательных нуля подменяются либо комбинацией B00V, либо 000V. 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 соответствии с этими правилами в данном случае выбирается “заготовка” B00V, как показано на Рис. 3.16.

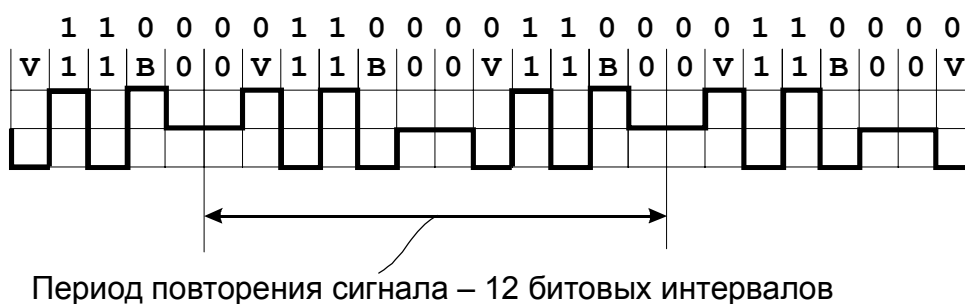
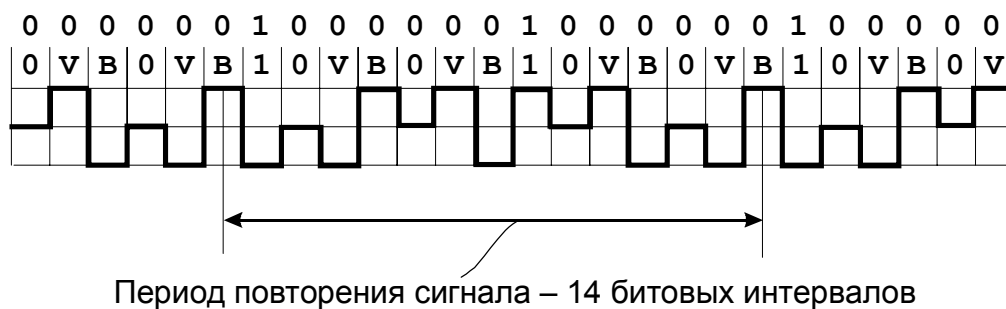


Рис. 3.16. Сигнал HDB3

### Решение задачи, приведенной в п. 2.4.11 (с. 39)

В коде B6ZS каждые шесть последовательных нулей подменяются комбинацией 0VB0VB. Согласно этому правилу, сигнал имеет форму, показанную на Рис. 3.17.



**Рис. 3.17. Сигнал B6ZS**

### Решение задачи, приведенной в п. 2.4.12 (с. 39)

Задача имеет три варианта решения в зависимости от начального состояния сигнала (–1, 0, +1). Эти варианты показаны, соответственно, на Рис. 3.18, а — в. Обратите внимание, что сумма трёх диаграмм нулевая в каждом такте.



**Рис. 3.18. Варианты решения задачи 2.4.12**

### Решение задачи, приведенной в п. 2.6.1 (с. 42)

Вероятность обнаружения в некотором такте каждого из трёх заданных 20-разрядных кодов равна  $2^{-20}$ . Иными словами, в среднем на каждые переданные  $2^{20}$  бит  $\approx 1$  Мбит приходится три командные вставки по четыре бита в каждой, всего 12 бит. При скорости передачи 1 Гбит/с в течение одной секунды будет выполнено 1000 вставок по 12 бит, всего 12000 бит. Таким образом, скорость передачи командных битов равна 12000 бит/с.

### Решение задачи, приведенной в п. 2.6.2 (с. 43)

При заданных условиях в скремблированном потоке данных в среднем в каждой цепи из  $2^{30} \approx 10^9$  бит = 1 Гбит будет встречаться один искомый код. При скорости передачи данных, равной 10 Гбит/с, средняя частота следования моментов вероятностной синхронизации составляет примерно 10 Гц. Среднее время между этими моментами составляет 100 мс. Вероятностный подход позволяет упростить схемные решения, исключить средства программной поддержки вхождения и поддержания синхронизации между удалёнными друг от друга устройствами (Сравните данное решение с приведенным в [1]).

### Решение задачи, приведенной в п. 2.8.1 (с. 51)

Как следует из эквивалентной схемы (Рис. 3.19), при одновременном использовании всех трёх телефонных аппаратов разговорные токи складываются на общем резисторе  $R1$  — эквивалентном сопротивлении параллельно включённых проводов линии 1 (Рис. 2.46). В результате в общей точке В формируется нежелательное суммарное напряжение  $U_B$ , отличное от нуля. Поэтому напряжения на резисторах  $RZ$ , имитирующих телефонные аппараты, содержат нежелательные составляющие, формируемые от двух соседних цепей. Каждая составляющая, если она достаточно велика, будет прослушиваться как посторонний разговор. При  $R1/RZ \rightarrow 0$  взаимное влияние телефонных каналов друг на друга уменьшается.

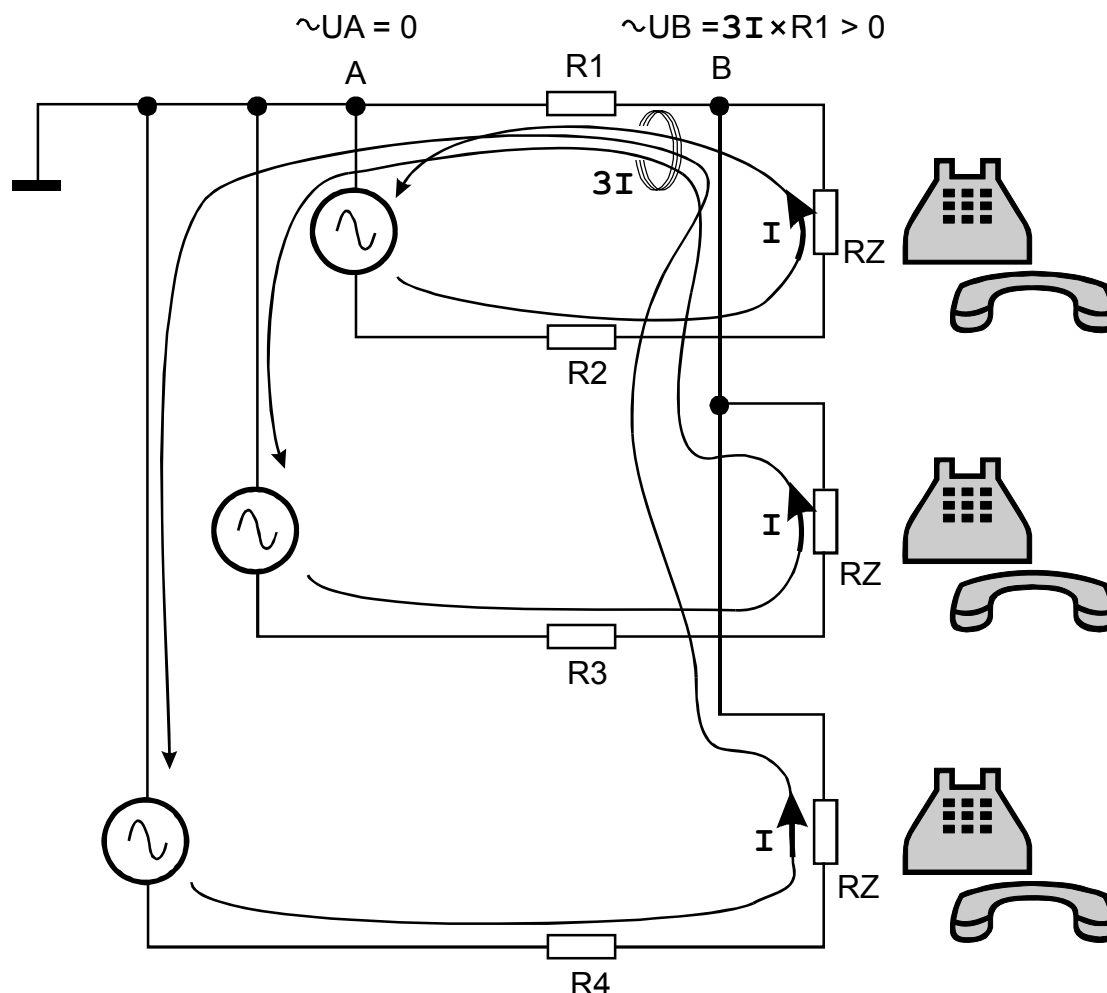


Рис. 3.19. Эквивалентная схема фрагмента сети (к задаче 2.8.1)

Решение задачи, приведенной в п. 2.10.1 (с. 56)

Источником данных для регистров А и В (Рис. 2.51) является один и тот же сигнал Q. Поэтому в установившемся режиме через эти регистры проходит одна и та же последовательность битов.

По положительному фронту сигнала CLK в регистре А формируется очередной псевдослучайный код. Вычисляется очередной бит Q. Этот бит “стремится” попасть в оба регистра, но сначала (по отрицательному фронту сигнала CLK) он попадает в регистр В и только затем (по положительному фронту сигнала CLK) — в регистр А.

Таким образом, приёмник переходит в новое состояние раньше, чем передатчик; создаётся эффект “отрицательной задержки”.

В схеме, приведенной на Рис. 3.20, генератор G перемещён в устройство DCE. Применена встречная синхронизация, вместо линии CLK использована линия TxС. Эффект “отрицательной задержки” наблюдается и в этой схеме, так как, по сути, она осталась той же самой, что и в условиях задачи (Рис. 2.51).

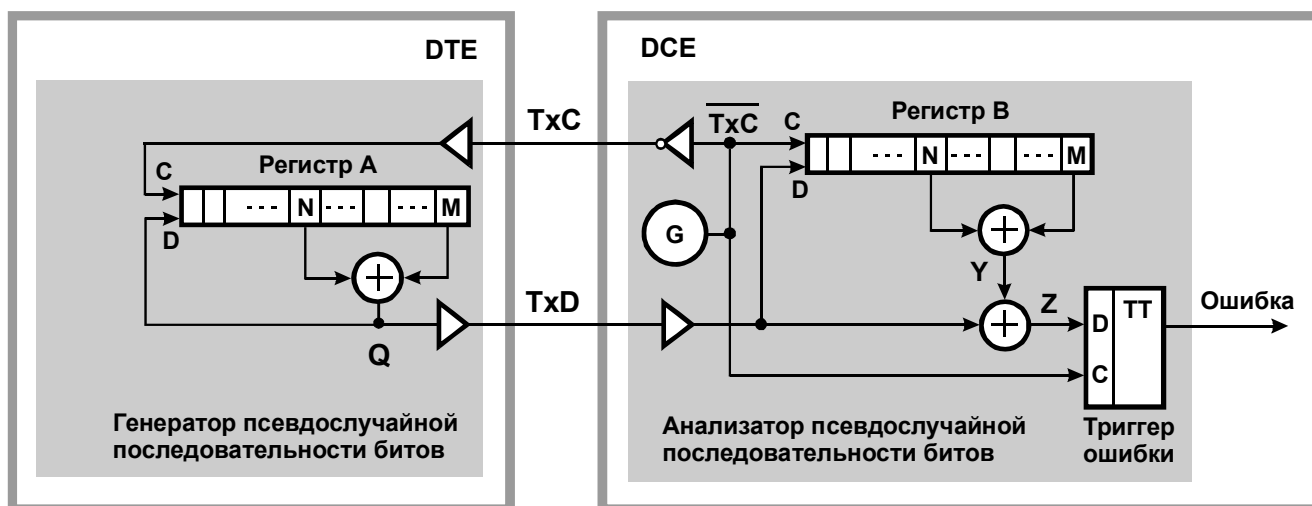


Рис. 3.20. Система контроля передачи данных и синхросигнала между устройствами типа DTE и DCE (вариант)

Ответы на вопросы, приведенные в п. 2.10.2 (с. 56)

Из описания аналогичной системы, показанной на Рис. 2.49, следует, что:

- 1) Временные соотношения между сигналами RxС и RxD удовлетворяют требованиям интерфейса V.35, в частности, положительные фронты сигнала RxС задают границы между битовыми интервалами сигнала RxD.
- 2), 3) Временные соотношения между сигналами RxС и RxD и работа BER-тестера в целом не зависят от длины линии связи (в пределах длины, при которой обеспечивается работоспособность системы).
- 4) Задержка тестового сигнала, вызванная его двойным пробегом по линии связи, не учитывается в работе BER-тестера.

Ответ на вопрос, приведенный в п. 2.10.3 (с. 57)

Действительно, в схеме, приведенной в условиях задачи (Рис. 2.53), сигнал TxС с генератора G1 проходит на вход синхронизации BER-генератора модема 3 против течения потока тестовых данных.

Идеи такой передачи рассмотрены в [1]. Одна из них, реализованная в несколько ином контексте, поясняется Рис. 3.21.

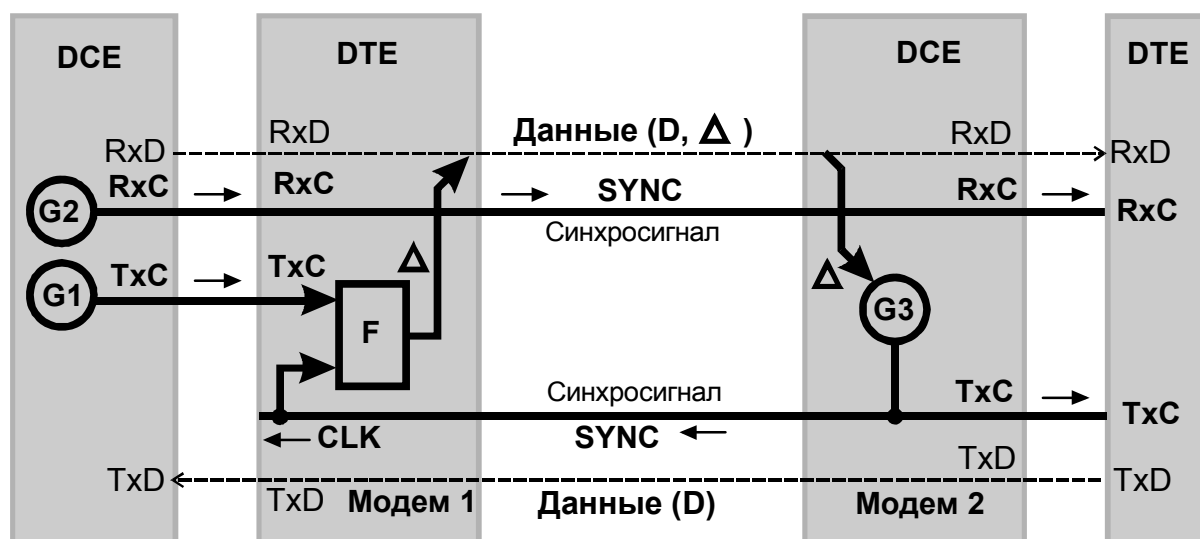


Рис. 3.21. Пояснение идеи передачи синхросигнала против течения потока данных

В данном случае сигналы TxС и RxС с генераторов G1 и G2 устройства DCE проходят на удалённую сторону системы передачи данных и под теми же наименованиями поступают на входы синхронизации оконечного устройства DTE. Один из этих сигналов (RxС), как обычно, в результате кодирования (см. п. 2.3) смешивается со “своими” данными (RxD), т. е. передаётся с ними в едином потоке (по течению). Второй сигнал (TxС) распространяется в направлении, противоположном течению “своих” данных (TxD), что, собственно, и может вызвать сомнения в работоспособности системы.

Для правильной работы системы (Рис. 3.21) необходимо совместить фазу сигнала TxС с выходом генератора G1 с фазой сигнала CLK, сопровождающего передачу данных TxD в оконечное устройство, расположенное слева. Фазовый компаратор F формирует цифровой сигнал  $\Delta$  разбаланса фаз, под действием которого генератор G3, изменяет фазу формируемого им синхросигнала в сторону уменьшения разбаланса. Генератор G3 имеет ту же номинальную частоту, что и генератор G1. Сигнал  $\Delta$  разбаланса фаз может быть представлен целым числом со знаком и разрядностью, равной, например, трём. Это число периодически передаётся из модема 1 в модем 2 по служебному каналу в составе формируемых кадров. Пример построения служебного канала в составе суперкадра показан на Рис. 3.30.

Применительно к схеме, показанной на Рис. 2.53, такое решение позволяет подстроить фазу синхросигнала на входе С BER-генератора модема 3 таким образом, чтобы сигналы TxС и TxD на входах С и D BER-анализатора модема 1 находились в правильных временных соотношениях.

#### Решение задачи, приведенной в п. 2.12.1 (с. 62)

В соответствии с определениями сигналов и правилами адаптации сети к отказу ([1], гл. 6), преобразование осуществляется за один шаг, как показано на Рис. 3.22.

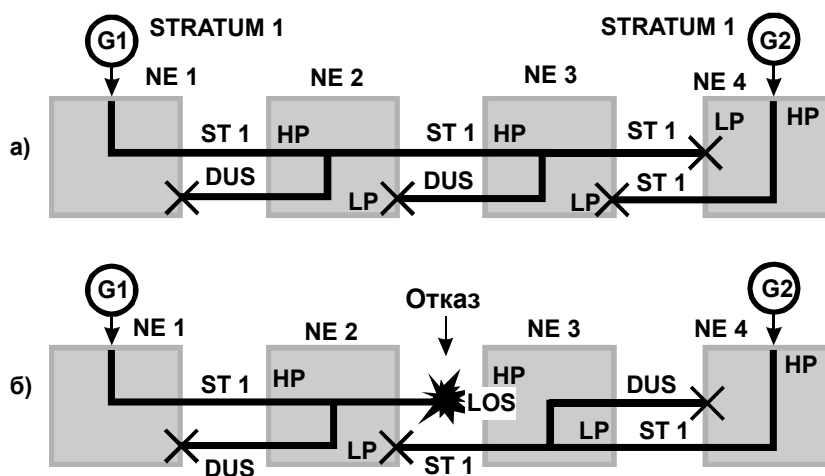


Рис. 3.22. Фрагмент компьютерной сети: а — исходный; б — после адаптации к отказу (к задаче 2.12.1)

#### Решение задачи, приведенной в п. 2.12.2 (с. 62)

В соответствии с определениями сигналов и правилами адаптации сети к отказу ([1], гл. 6), преобразование осуществляется за несколько шагов, как показано на Рис. 3.23.

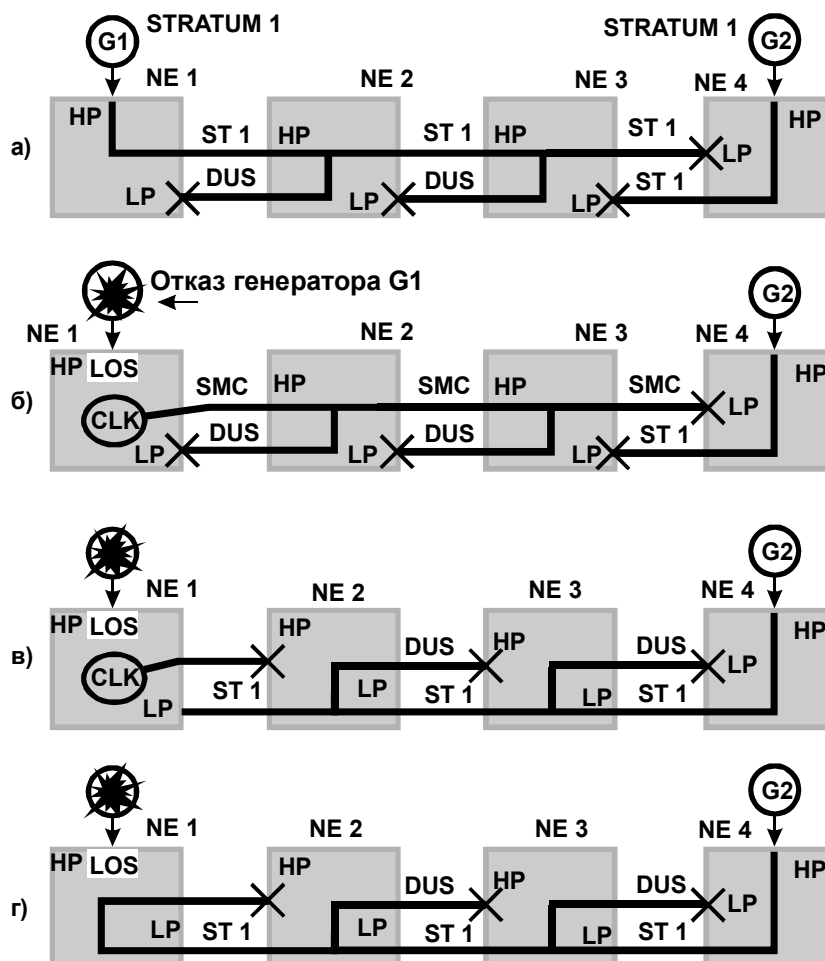


Рис. 3.23. Фрагмент компьютерной сети: а — исходный; б — г — в процессе и после адаптации к отказу (к задаче 2.12.2)

### Решение задачи, приведенной в п. 2.12.3 (с. 62)

В соответствии с определениями сигналов и правилами адаптации сети к отказу ([1], гл. 6), преобразование осуществляется за два шага, как показано на Рис. 3.24.

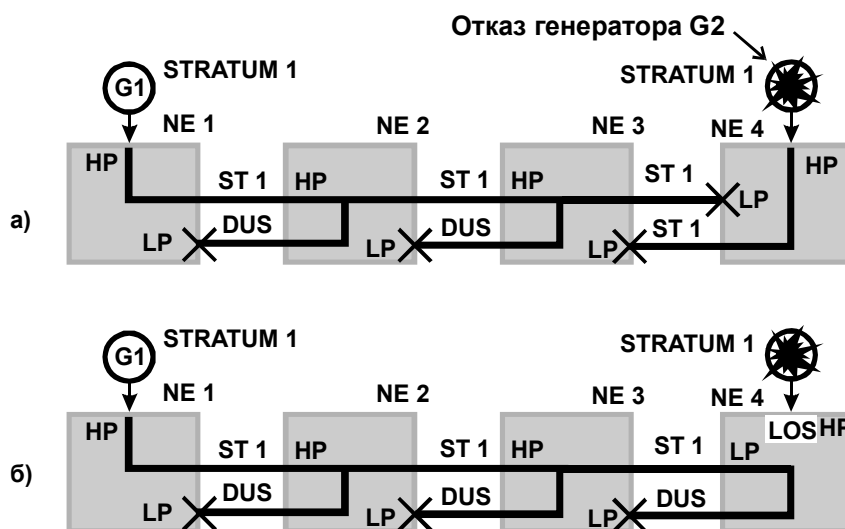


Рис. 3.24. Фрагмент компьютерной сети: а — исходный; б — после адаптации к отказу (к задаче 2.12.3)

### Решение задачи, приведенной в п. 2.14.1 (с. 90)

1, ситуация а

Получение в линии связи цепи из 10000 нулей возможно при 10000-кратном совпадении двух случайных последовательностей: DATA и Y. Вероятность совпадения первой пары битов равна 1/2, первой и второй пар — 1/4, первой, второй и третьей пар — 1/8 и т. д. Вероятность 10000-кратного совпадения пренебрежимо мала и равна  $2^{-10000}$ .

1, ситуация б

Если на вход скремблера подаётся цепь из 10000 нулевых битов, то элемент XOR2 выполняет функцию повторителя последовательности Z1 псевдослучайных битов с максимальным числом следующих подряд нулей — не более четырёх. Ответ: вероятность получения цепи из 10 тыс. нулей — нулевая.

1, ситуация в — аналогична предыдущей.

Если на вход скремблера подаётся цепь из 10000 единичных битов, то элемент XOR2 выполняет функцию повторителя-инвертора последовательности Z1 псевдослучайных битов с максимальным числом следующих подряд единиц — не более пяти. Ответ: вероятность получения цепи из 10 тыс. нулей — нулевая.

2. Битовая синхронизация между дескремблером и скремблером, вероятнее всего, будет потеряна.

3. Злоумышленник формирует последовательность длиной 10 тыс. битов, совпадающую с последовательностью Z1 и начинающуюся, например, с бита, который соответствует состоянию регистра RG1 равному 11111. Далее сформированная злоумышленником последовательность периодически передаётся на вход скремблера. В одной из ситуаций (её вероятность достаточно высока и составляет 1/31) к моменту начала передачи данных на вход скремблера в регистре RG1 присутствует желаемый код 11111, поэтому последовательности DATA и Z1 совпадают, в линии формируется цепь из 10 тыс. нулей (сигнал отсутствует). Из этого следует, что цель злоумышленника достижима простыми средствами. Одна из мер противодействия — увеличение длины регистра RG1 (RG2), например, до 64 бит.

4. Процесс установления кодовой синхронизации описан в п. 2.13.4 [4].

5. Одиночная ошибка не размножается.

6. Запрещённый код в регистре RG1 (RG2) — 00000.

#### Решение задачи, приведенной в п. 2.14.2 (с. 91)

1, ситуация а

Получение в линии связи цепи из 10000 нулей возможно при 10000-кратном совпадении двух случайных последовательностей: DATA и Y. Вероятность совпадения первой пары битов равна 1/2, первой и второй пар — 1/4, первой, второй и третьей пар — 1/8 и т. д. Вероятность 10000-кратного совпадения пренебрежимо мала и равна  $2^{-10000}$ .

1, ситуация б

Существует значительная (1/32) вероятность того, что к моменту подачи первого нулевого бита из последовательности битов DATA в регистре RG1 (RG2) сформирован нулевой код. Тогда результат суммирования последовательностей элементом XOR2 также окажется нулевым на протяжении 10000 тактов. Ответ: 1/32.

1, ситуация в — аналогична предыдущей.

Существует значительная (1/32) вероятность того, что к моменту подачи первого единичного бита из последовательности битов DATA в регистре RG1 (RG2) сформирован код 11111. Тогда результат суммирования последовательностей элементом XOR2 окажется нулевым на протяжении 10000 тактов. Ответ: 1/32.

2. Битовая синхронизация между дескремблером и скремблером, вероятнее всего, будет потеряна.

3. Первый способ — увеличение длины регистра RG1 (RG2), например, до 64 бит.

Второй способ — применение схемы защиты от длинных последовательностей нулей или единиц, приведенной на Рис. 2.67 [4].

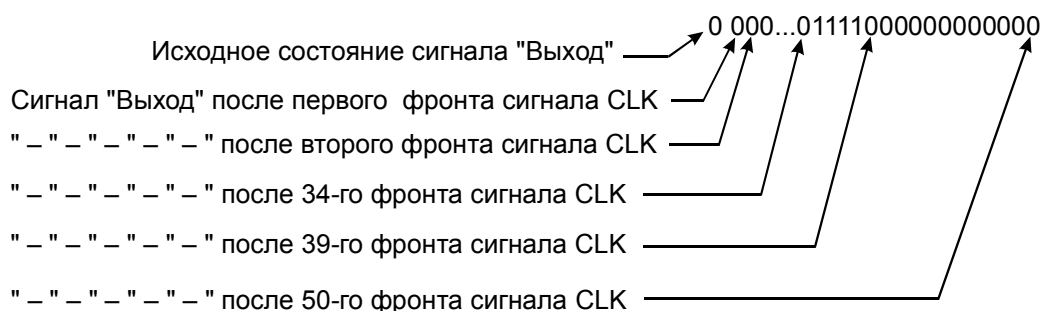
4. Время вхождения в кодовую синхронизацию составляет M тактов (M = 5).

5. Одиночная ошибка превращается в тройную.

6. Запрещённых кодов в регистре RG1 (RG2) нет.

Решение задачи, приведенной в п. 2.14.3 (с. 92)

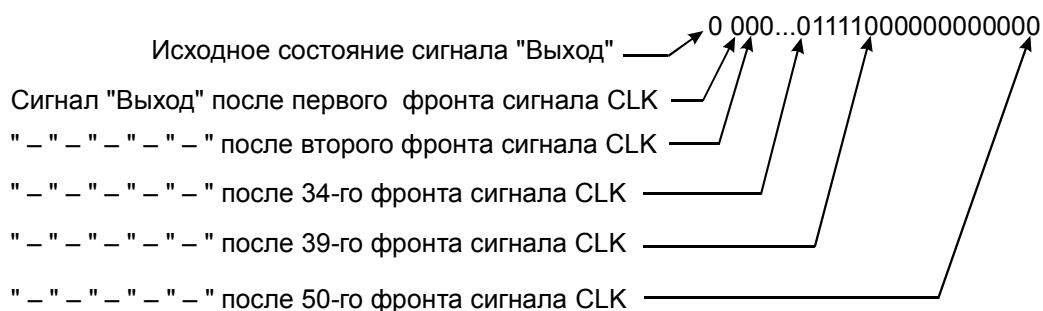
- 1) Если начальное состояние регистра RG равно 000...0, то оно остаётся неизменным на протяжении любого промежутка времени. Последовательность битов на выходе генератора: 000...0.
- 2) Если начальное состояние регистра RG равно 111...1, то сигнал на выходе формируется так:



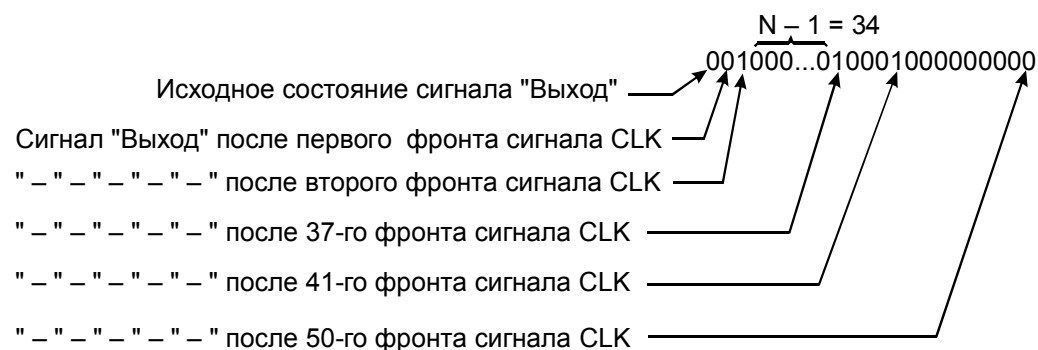
Период повторения битов в последовательности равен  $2^M - 1$  бит.

Решение задачи, приведенной в п. 2.14.4 (с. 93)

- 1) Если начальное состояние регистра RG равно 111...1, то сигнал на выходе формируется так:



- 2) Если начальное состояние регистра RG равно 000...010, то сигнал на выходе формируется так:



Период повторения битов в последовательности равен  $2^M$  бит.

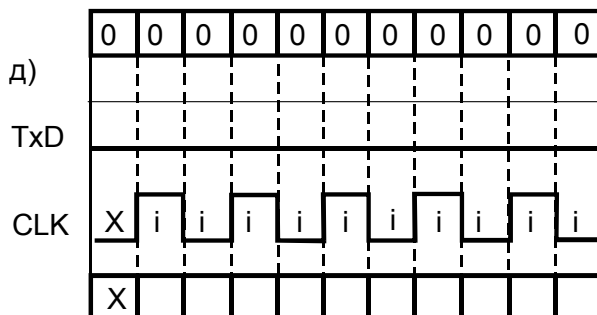
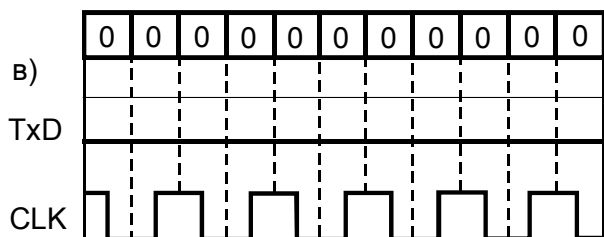
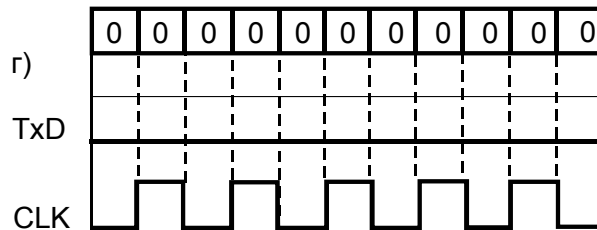
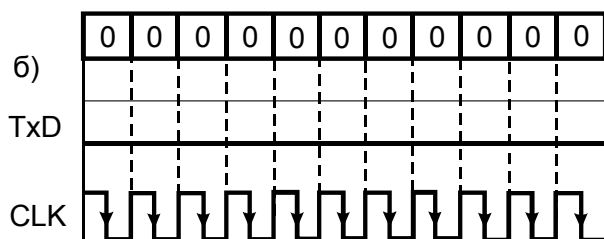
Решение задачи, приведенной в п. 2.16.1 (с. 97)

Сравнительный анализ временных диаграмм, приведенных на Рис. 2.80 (Рис. 2.79), по критериям скорости передачи, сложности реализации и уровням излучаемых помех приведен в п. 2.15.

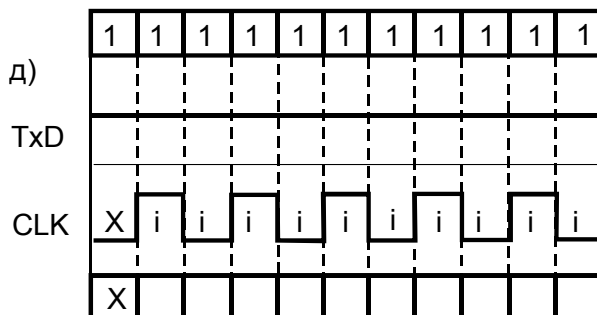
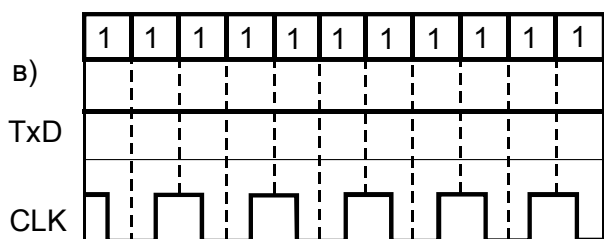
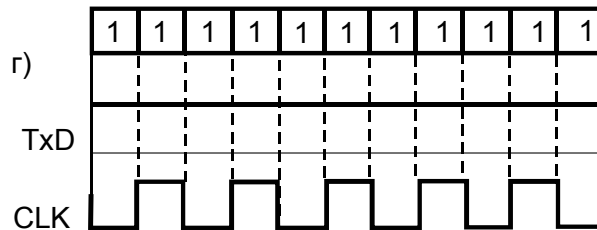
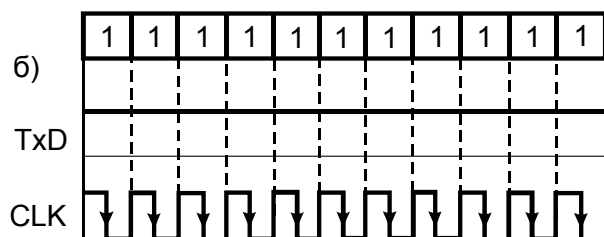
Далее приведены диаграммы, аналогичные представленным на Рис. 2.80, б — д, применительно к следующим последовательностям битов на линии TxD:

- первая ситуация: 00000000000;
- вторая ситуация: 11111111111;
- третья ситуация: 01010101010;
- четвёртая ситуация: 00110011001.

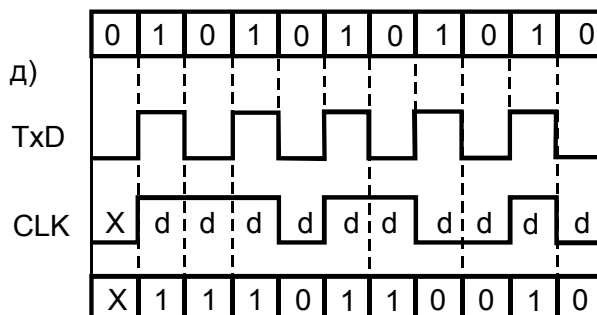
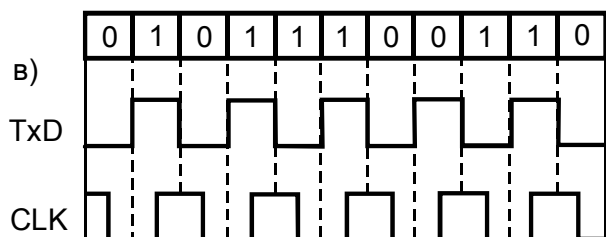
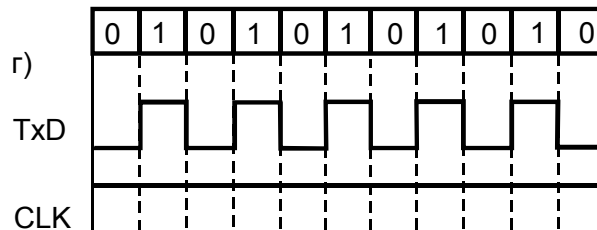
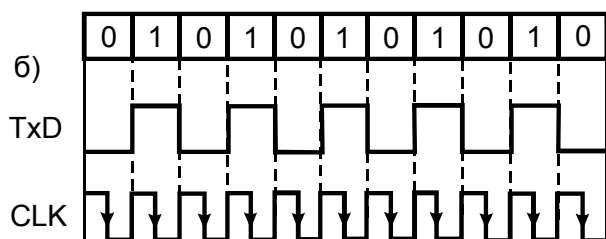
**Первая ситуация: 0000000000**



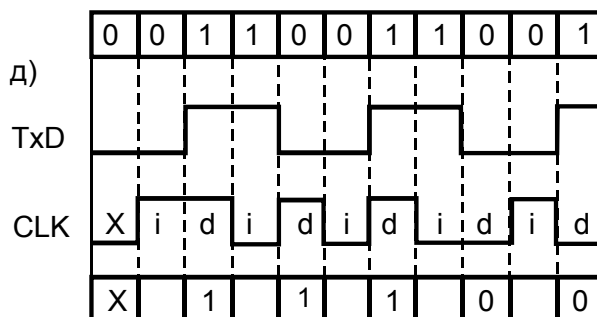
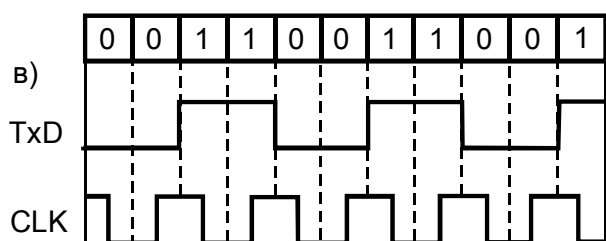
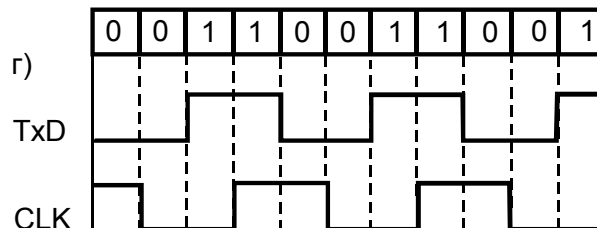
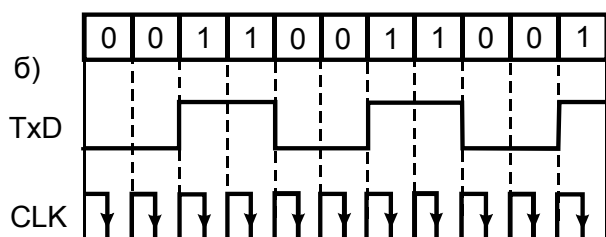
**Вторая ситуация: 1111111111**



**Третья ситуация: 01010101010**



**Четвёртая ситуация: 00110011001**



Решение задачи, приведенной в п. 2.18.1 (с. 111)

1) Тремя разрядами можно отобразить 8 чисел — от 0 до 7. Казалось бы, нумерация кадров ограничена именно этим диапазоном. Но это не совсем так, если вести нумерацию с учётом предыстории ([1], п. 7.5). Рассмотрим цепочку кадров. Предположим, что кадры с нулевого по седьмой пронумерованы числами от 0 до 7. Все номера израсходованы? Да, но для продолжения нумерации можно поступить, например, так. Восьмой — пятнадцатый кадры пронумеруем теми же числами, но в обратном порядке: 7, 6, 5, ... 0. Шестнадцатый — двадцать третий кадры пронумеруем, например, так: 00735441 (произвольное сочетание цифр). Двадцать четвёртый — тридцать первый кадры пронумеруем иначе: 13570244 (здесь также произвольное сочетание цифр) и т. д. Таким образом, получаем следующую практически не ограниченную по длине нумерацию кадров, заранее известную передатчику и приёмнику:

01234567765432100073544113570244...

В этой нумерации при её просмотре слева направо одни и те же цифры отображают разные последовательно возрастающие номера кадров. Например, цифра 0 в зависимости от того, в каком месте последовательности она присутствует, обозначает нулевой, пятнадцатый, шестнадцатый или семнадцатый кадры, цифра 1 — первый, четырнадцатый, двадцать третий или двадцать четвёртый кадры и т. д. Приёмник, зная, как построена приведенная выше последовательность цифр и отслеживая её, распознаёт номер передаваемого в данный момент кадра. Например, получив крайнюю справа цифру 4 и зная предысторию, приёмник узнаёт, что в данный момент на его вход поступил кадр с номером 31 и т. д.

2) Можно сократить число разрядов кода  $Q$  до одного. Для этого применим псевдослучайную последовательность битов длиной  $2^{10} - 1$  бит (Рис. 3.25), генерируемую программно или аппаратно с помощью сдвигового регистра с обратными связями (Рис. 2.59).

Фрагмент последовательности в 10-разрядном скользящем вправо окне представляет собой уникальный номер очередного принимаемого кадра (всего имеется  $2^{10} - 1 = 1023$  уникальных номера). Разряды  $Q$  могут служить однобитовыми флагами начала кадров [1, п. 7.2].

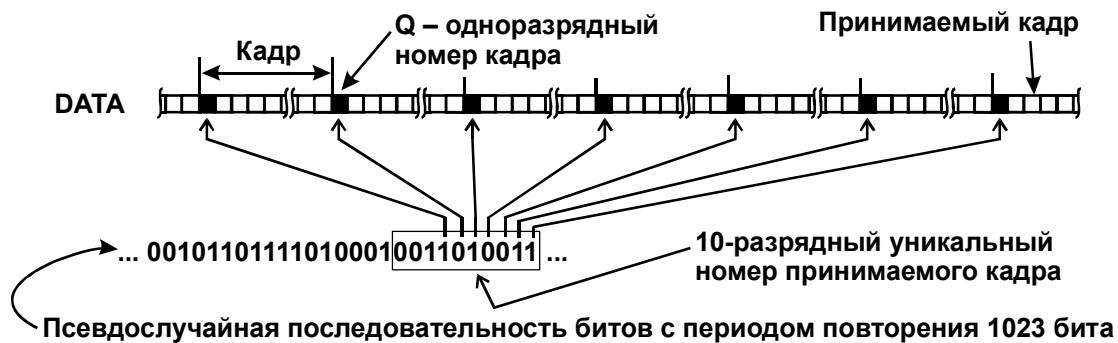


Рис. 3.25. Нумерация кадров с помощью одноразрядных кодов  $Q$

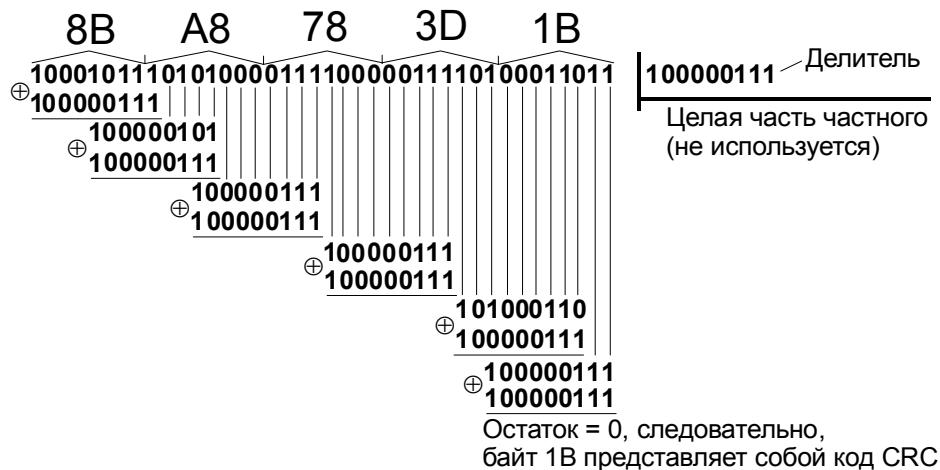
#### Решение задачи, приведенной в п. 2.18.2 (с. 111)

Обозначим стафф-бит (лог. 0) символом  $z$ . Этот бит вводится в последовательность всякий раз после обнаружения в ней кода 011111 или  $z11111$ . В результате применения битстаффинга получим искомую последовательность:

... 011111 $z$ 11111 $z$ 011111 $z$ 0101011011111 $z$ 00011111 $z$ 11111 $z$ 00000011111 $z$ 1...

#### Решение задачи, приведенной в п. 2.18.3 (с. 111)

Да, эта последовательность может “подозреваться” в качестве заголовка, так как байт 1B представляет собой код CRC — циклический избыточный код, вычисленный для четырёх предшествующих байтов с использованием образующего многочлена  $x^8 + x^2 + x + 1$ :



При периодическом (с шагом 53 байта) устойчивом повторении положительных результатов проверки предполагаемых заголовков они рассматриваются как истинные.

#### Решение задачи, приведенной в п. 2.18.4 (с. 111)

Выбранная из случайного потока цепочка данных содержит пять байтов. Она может “подозреваться” в качестве кандидата на заголовок АТМ-ячейки, если пятый байт является контрольной суммой (код CRC) четырех предыдущих. Вероятность того, что случайный 8-разрядный код примет ожидаемое значение, равна  $2^{-8} = 1/256$ . При периодическом (с шагом 53 байта) устойчивом повторении положительных результатов проверки предполагаемых заголовков они рассматриваются как истинные.

#### Решение задачи, приведенной в п. 2.18.5 (с. 111)

**Первая ситуация.** Так как поток битов данных имеет случайный характер, то любой заранее заданный 10-разрядный код, в частности, флаговый код F, будет встречаться в нем в среднем через каждые  $2^{10} = 1024$  бит  $\approx 10^3$  бит. При длине кадра, равной  $10^4$  бит, в нем, помимо истинного, будет обнаружено около  $10^4/10^3 = 10$  ложных флагов. Таким образом, среди группы обнаруженных 11 кодов F только один представляет собой истинный флаг. Вероятность выбора именно его невелика и составляет  $1/11 \approx 0,091$ .

**Вторая ситуация.** Если при той же длине кадра увеличить разрядность флага до 30 бит ( $M = 30$ ), то его случайные копии будут встречаться в потоке данных со средним периодом  $2^{30} = 10^9$  бит. Один ложный флаг будет в среднем обнаруживаться в группе из  $10^9/10^4 = 10^5$  кадров. Вероятность обнаружения ложного флага в одном кадре равна  $10^{-5}$ . Иными словами, вероятность того, что с первой попытки поиска обнаружен истинный (а не ложный) флаг, чрезвычайно высока и составляет  $1 - 10^{-5} = 0,99999$ .

#### Решение задачи, приведенной в п. 2.18.6 (с. 112)

Кадр ITU-T V.110 (Рис. 3.26, а) 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 В63 нижней строки. Биты В1 — В63 представляют собой передаваемые данные; S1 — S8 — биты синхронизации. Как показано в п. 2.17.9, вместо некоторых битов синхронизации можно передавать данные X0 — X4 дополнительного канала.

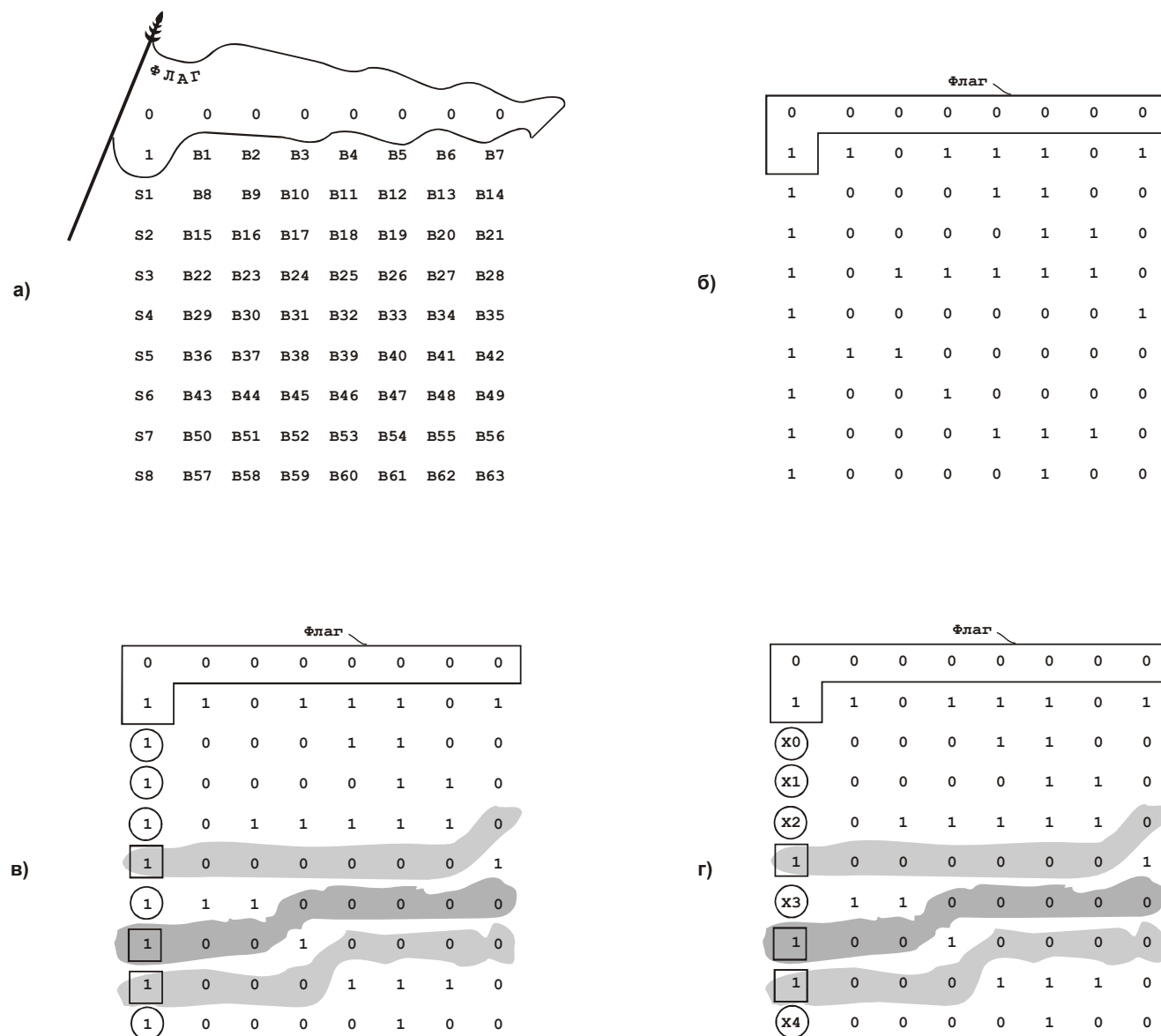


Рис. 3.26. Передача данных вместо избыточных битов синхронизации кадра: а — общая структура кадра в соответствии с рекомендацией ITU-T V.110; б — кодовый пример кадра; в — поиск битов синхронизации, которые должны безусловно принимать значения лог. 1 (выделены квадратиками); г — определение позиций для передачи данных (X0 — X4) вместо битов синхронизации (выделены кружочками)

Результат анализа кадра, приведенного в условиях задачи, таков:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

### Решение задачи, приведенной в п. 2.20.1 (с. 118)

После суммирования по модулю два с кодом 21 коды 10, 11 и 13 преобразуются в коды 31, 30 и 32. Эти коды предваряются кодом 10.

Результат байтстаффинга:

FA, EE, 12, 10, 31, 10, 31, 10, 31, 99, 10, 32, 10, 32, 05, 10, 30, 10, 31, 88, 10, 30, 10, 30, CC, DD, DD, 10, 30

### Решение задачи, приведенной в п. 2.20.2 (с. 118)

1) Применим байтстаффинг к последовательности пользовательских данных.

После суммирования по модулю два с кодом 21 коды 10, 11 и 13 преобразуются в коды 31, 30 и 32. Эти коды предваряются кодом 10.

Результат байтстаффинга:

FA, EE, 12, 10, 31, 10, 31, 10, 31, 99, 10, 32, 10, 32, 05, 10, 30, 10, 31, 88, 10, 30, 10, 30, CC, DD, DD, 10, 30

Этот результат не содержит управляющих кодов XOFF = 13 и XON = 11.

2) Введём управляющие коды в начало и конец последовательности:

13, FA, EE, 12, 10, 31, 10, 31, 10, 31, 99, 10, 32, 10, 32, 05, 10, 30, 10, 31, 88, 10, 30, 10, 30, CC, DD, DD, 10, 30, 11.

### Ответы на вопросы, приведенные в п. 2.22.1 (с. 123)

Сигнал с кварцевого генератора не влияет на синхросигнал, выделенный из линии; последний определяется только потоком входных данных;

Частоты сигналов на выходах генераторов G1 и G2 незначительно различаются;

Близость характеристик генераторов G1 и G2 необходима для правильного выбора рабочей точки генератора G2 и компенсации дестабилизирующих факторов; близость характеристик достигается выполнением генераторов в непосредственной близости друг от друга на одном кристалле микросхемы (Рис. 3.27).

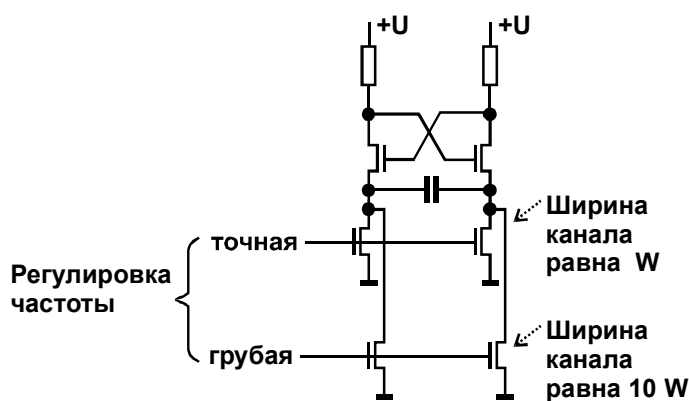


Рис. 3.27. Схема генератора G1 (G2), см. Рис. 2.99

### Ответы на вопросы, приведенные в п. 2.24.1 (с. 127)

Анализ временных диаграмм, аналогичных приведенным на Рис. 3.28 (см. п. 2.23), позволяет заключить, что:

- проскальзывания синхронизации исключаются оригинальным схемным решением, в частности, нестандартным включением триггера, обозначенного символом #, и правильным выбором соотношения частот  $Y/X$  (подробности см. в п. 2.23);
- диапазон джиттера определяется периодом сигнала CLK2.

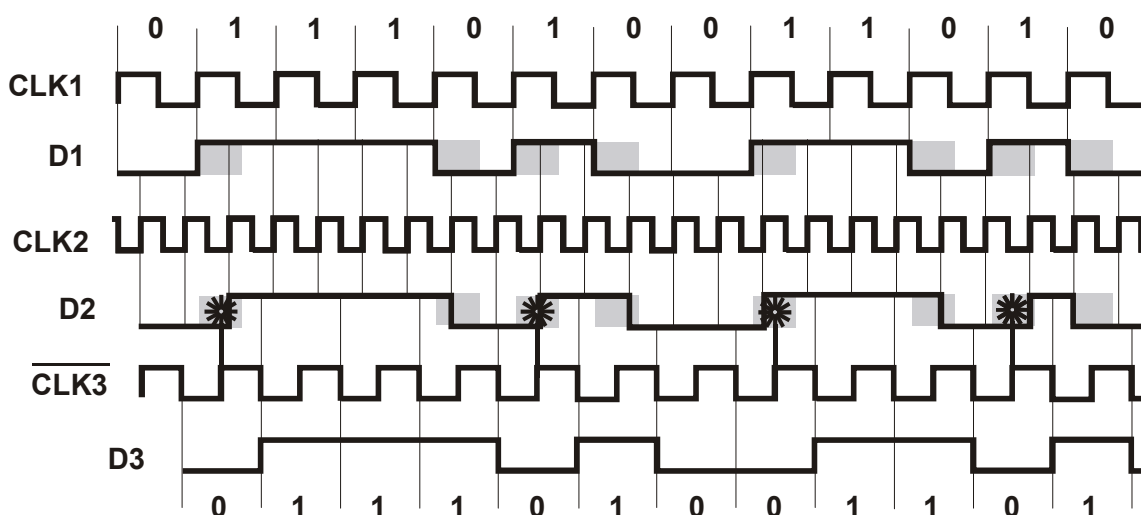


Рис. 3.28. Временные диаграммы передачи данных в системе, показанной на Рис. 2.103

### Ответы на вопросы, приведенные в п. 2.24.2 (с. 127)

Входные данные (Рис. 2.104) проходят по цепи “передатчик — ретранслятор 1 — ретранслятор 2 — приемник”. Чтобы исключить проскальзывания, передатчики и приемники можно выполнить по схемам, приведенным на, Рис. 2.103, но скремблер и дескремблер подключить только на входе и выходе канала. По существу, схема, показанная на Рис. 2.104, представляет собой расширение схемы, показанной на Рис. 2.103. Джиттер, вносимый в линию каждым генератором, уничтожается соответствующим приёмником (подробности см. в п. 2.23).

Число ретрансляторов не ограничено.

### Ответ на вопрос, приведенный в п. 2.26.1 (с. 134)

В канале E1 используются коды HDB3 и (реже) AMI (см. Рис. 2.27)

### Ответы на вопросы, приведенные в п. 2.26.2 (с. 134)

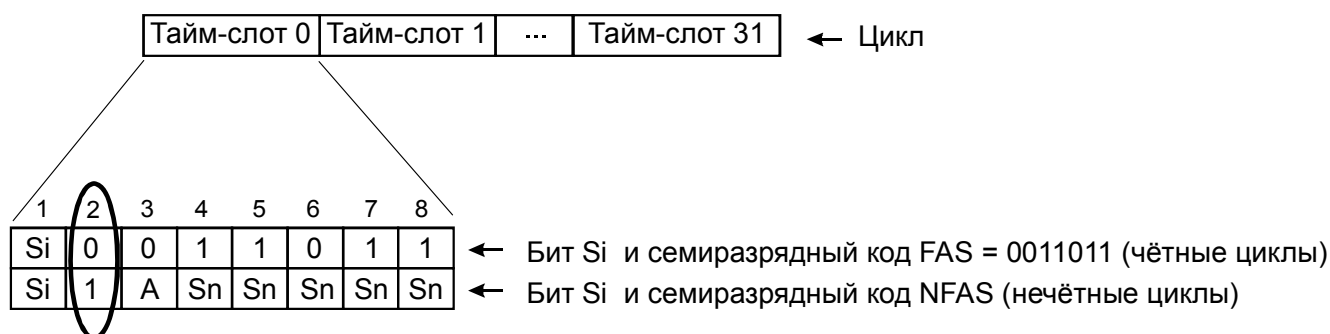
В состав канала E1 входят два кабеля для однонаправленной передачи сигналов в одном и другом направлениях. Кабели могут быть выполнены витыми парами проводов с волновым сопротивлением 120 Ом. Увеличение дальности связи достигается применением двух коаксиальных кабелей вместо витых пар. Волновое сопротивление коаксиального кабеля — 75 Ом.

### Ответы на вопросы, приведенные в п. 2.26.3 (с.134)

Флаг начала кадра не уникален. Он относится не к одному, а к двум смежным кадрам.

### Ответы на вопросы, приведенные в п.2.26.4 (с.134)

Ровно 50 % передаваемых кадров E1 не имеют флагов FAS (Рис. 3.29). Строгое чередование байтов FAS и NFAS в нулевых таймслотах, по существу, означает, что один флаговый код относится не к одному, а к двум смежным кадрам. Поэтому приёмник синхронизируется на уровне пар кадров.



**Рис. 3.29. Структура цикла (кадра) E1**

Чтобы избежать нежелательных совпадений байтов FAS и NFAS в смежных парах кадров, во втором разряде этих байтов размещены противоположные биты (выделены овалом на рисунке). Начальное установление кадровой синхронизации сопровождается набором статистических данных о найденных кодах FAS. Копии этих кодов могут встречаться в любых таймслотах 1 — 31. Приёмник отсеивает случайные коды FAS и обращает внимание только на те, которые устойчиво повторяются с периодом, равным двум кадрам. Они и являются истинными флаговыми кодами.

### Решение задачи, приведенной в п. 2.26.5 (с. 134)

Длительность передачи одного кадра E1 составляет 125 мкс. Частота следования кадров равна 8 кГц. Пары кадров передаются с частотой 4 кГц. За одну секунду из передатчика в приёмник проходят 4 тыс. пар кадров. Каждая пара кадров E1 переносит 5 служебных битов Sn (см. Рис. 3.29). Следовательно, за одну секунду передаются  $4000 \times 5 = 20000$  служебных битов. Таким образом, пропускная способность служебного канала связи составляет 20 кбит/с.

### Решение задач, приведенных в п. 2.26.6 (с. 134)

Длительность передачи одного кадра E1 составляет 125 мкс. Частота следования кадров равна 8 кГц. Пары кадров передаются с частотой 4 кГц. За одну секунду из передатчика в приёмник проходят 4 тыс. пар кадров. Каждая пара кадров E1 переносит один бит А (см. Рис. 3.29). Следовательно, за одну секунду передаются  $4000 \times 1 = 4000$  служебных битов А. Таким образом, пропускная способность служебного канала передачи признака А аварии удалённого устройства составляет 4 кбит/с.

Рассуждая аналогично и учитывая, что каждый кадр (а не пара кадров) переносит один бит Si, пропускная способность канала на основе этих битов в два раза большая и составляет 8 кбит/с.

### Решение задачи, приведенной в п. 2.26.7 (с. 134)

Длительность передачи одного кадра E1 составляет 125 мкс. Частота следования кадров равна 8 кГц. Один таймслот переносит за одну секунду  $8000 \times 8 = 64$  кбит, пять таймслотов — в

пять раз больше:  $64 \text{ кбит} \times 5 = 320 \text{ кбит}$ . Таким образом, пропускная способность канала на основе пяти таймслотов составляет 320 кбит/с.

#### Решение задачи, приведенной в п. 2.26.8 (с. 134)

Длительность передачи одного кадра E1 составляет 125 мкс. Частота следования кадров равна 8 кГц. Один таймслот переносит за одну секунду  $8000 \times 8 = 64 \text{ кбит}$ , 31 таймслот — в 31 раз больше:  $64 \text{ кбит} \times 31 = 1984 \text{ кбит}$  (нулевой таймслот исключаем из рассмотрения). Таким образом, пропускная способность канала E1 без учёта нулевого (служебного) таймслота составляет 1984 кбит/с.

#### Ответ на вопрос, приведенный в п. 2.26.9 (с. 134)

В структуре цикла (кадра) не предусмотрены избыточные биты для компенсации разности частот синхросигналов на передающей и приёмной сторонах канала E1 (см. пп. 2.17, 2.29). Частоту проскальзываний можно уменьшить (но не исключить их полностью) применением буферной памяти типа FIFO, как показано на Рис. 2.113, в.

Буферная память, исходно заполненная данными на 50%, 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 (Рис. 2.113, в). 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 проскальзывания становится невозможной, в результате происходит потеря или повтор передачи массива битов, накопленного в памяти. После этого в течение некоторого периода система передачи данных вновь работает без проскальзываний и т. д.

#### Ответ на вопрос, приведенный в п. 2.26.10 (с. 135)

Сверхцикл (суперкадр) позволяет представить передаваемые данные в виде матрицы, в которой информация располагается в строках и столбцах. Такое представление удобно при передаче данных по 30 цифровым телефонным каналам, хотя применимость суперкадров этим не ограничивается. Как показано на Рис. 3.30, в столбцах, соответствующих таймслотам 1 — 15, 17 — 31 размещены по 16 цифровых отсчётов аналогового (голосового) сигнала телефонных каналов 1 — 30. Темп опроса — 8 кГц, длительность одного кадра — 125 мкс, длительность суперкадра — 2 мс.



**Рис. 3.30. Структура суперкадра (см. также Рис. 2.107, Рис. 2.108, табл. 2)**

В столбце, соответствующем таймслоту 0, размещаются четыре сообщения; их содержание поясняется надписями на рисунке. Таймслоты 16 содержат 30 четырёхбитовых кодов сигнализации о состоянии телефонных каналов 1 — 30. Эти коды обозначены чёрными точками. Исходящие из них стрелки направлены к соответствующим столбцам матрицы. Так, верхняя пара стрелок направлена к телефонным каналам 1 и 16, следующая — к каналам 2 и 17 и т. д. Таким образом, суперкадр позволяет компактно разместить в нём данные и служебную информацию.

#### Решение задачи, приведенной в п. 2.26.11 (с. 135)

Длительность передачи одного суперкадра E1 (Рис. 3.30) составляет 2 мс. За это время передаются четыре бита сигнальной информации, относящейся к некоторому (любому) телефонному каналу. Таким образом, скорость передачи сигнальной информации составляет 4/0,002 бит/с или 2 кбит/с.

#### Решение задачи, приведенной в п. 2.26.12 (с. 135)

Длительность передачи одного суперкадра E1 (Рис. 3.30) составляет 2 мс. Если событие потери синхронизации произошло чуть позже момента формирования и отправки некоторого суперкадра, то оно будет зарегистрировано только в следующем суперкадре через 2 мс (остальными задержками пренебрегаем). Таким образом, максимальная задержка от события потери синхронизации удалённым передатчиком до момента обнаружения этого признака приёмником оценивается в 2 мс.

#### Решение задачи, приведенной в п. 2.26.13 (с. 135)

Совпадение принятого кода CRC с ожидаемым может быть и случайным. При случайных данных и случайном 4-разрядном коде CRC-4 вероятность того, что код CRC-4 принял нужное значение, составляет  $1/2^4 = 0,0625$ . Вероятность того, что совпадение неслучайно, равна  $1 - 0,0625 \approx 0,94$ .

### Ответы на вопросы, приведенные в п. 2.28.1 (с. 141)

При передаче кадра по сегменту А (Рис. 2.111) все станции и мультиплексор 1 обнаруживают преамбулу (7 байтов вида 10101010), соответствующую периодическому сигналу “Манчестер-II” частотой 5 МГц. Этот сигнал используется приемниками для вхождения в синхронизм с передатчиком.

### Ответ на вопрос, приведенный в п. 2.28.2 (с. 141)

В сети Ethernet 10 Base T скорость передачи кадра равна 10 Мбит/с. Скорость передачи данных через мост определяется числом N задействованных таймслотов в канале Е1 (от одного до 31) и составляет  $N \times 64$  кбит/с.

### Ответ на вопрос, приведенный в п. 2.28.3 (с. 141)

В процессе работы мультиплексоры 1 и 2 следят за MAC-адресами станций своих сегментов сети и ведут их оперативный учет в буферной памяти. В результате они поименно знают все свои станции, которые проявляли активность (инициировали обмен данными) в течение последних пяти минут.

Мультиплексор временно запоминает все передаваемые по своему сегменту сети кадры и анализирует их тип, адрес отправителя и адрес предполагаемого получателя (если такой адрес предусмотрен данным типом кадра).

Если адресат находится в пределах своего сегмента сети, то мультиплексор уничтожает полученный кадр (этот же кадр хранится у получателя и, разумеется, пока не уничтожается). Если адресат не числится в списке своих станций, то кадр пересылается мультиплексором на противоположную сторону моста.

### Ответ на вопрос, приведенный в п. 2.28.4 (с. 141)

Причин потерь кадров несколько. Рассмотрим ситуации, препятствующие успешной передаче кадра из сегмента А в сегмент В (Рис. 2.111).

- 1) В каждой из 16 попыток выдачи кадра в сегмент А передающая станция обнаруживает коллизию, поэтому она прекращает такие попытки и уничтожает кадр.
- 2) Мультиплексор 1 получает из сегмента А искажённый кадр (например, с неправильной контрольной суммой) и уничтожает его.
- 3) Входная буферная память мультиплексора 1 переполнена, поэтому кадр не принимается (уничтожается).
- 4) В результате ошибки в канале Е1 мультиплексор 2 не может восстановить переданный из мультиплексора 1 кадр, поэтому он уничтожается.
- 5) В каждой из 16 попыток выдачи кадра в сегмент В мультиплексор 2 обнаруживает коллизию, поэтому он уничтожает кадр.
- 6) При передаче кадра из мультиплексора 2 в сегмент В сети он искажается и вследствие этого не находит адресата или уничтожается приёмником.

### Ответ на вопрос, приведенный в п. 2.28.5 (с. 141)

На канальном уровне источник Ethernet-кадров не оповещается об их потере. Повторные запросы потерянной информации осуществляются протоколами более высоких уровней.

### Ответы на вопросы, приведенные в п. 2.28.6 (с. 141)

Мультиплексоры 1 и 2 не имеют MAC-адресов в сети.

### Ответы на вопросы, приведенные в п. 2.28.7 (с. 141)

Совпадение MAC-адресов двух станций исключено благодаря выполнению соглашения между производителями сетевого оборудования. Иными словами, в мире не существует (точнее, не должно существовать) двух устройств с одинаковыми MAC-адресами.

Имитацией такого совпадения может быть быстрый (менее чем за пять минут) перенос некоторой станции из сегмента А в сегмент В (Рис. 2.111). В течение этого времени мультип-

лексор 1 всё ещё помнит, что станция находится в его ведении, а мультиплексор 2 уже успел принять её на учёт, если она проявила активность. В этом случае временно создаётся иллюзия того, что в сегментах А и В присутствуют две станции с одинаковыми МАС-адресами. Это, однако, не приводит к каким-либо нежелательным последствиям.

#### Решение задачи, приведенной в п. 2.30.1 (с. 150)

Дисбаланс скоростей входного и выходного потоков данных приводит к накоплению в буферной памяти 1000 не переданных из узла сети битов за одну секунду или 600000 битов за 10 минут. Именно такой объём буферной памяти (600 кбит) достаточен для уменьшения частоты проскальзываний до требуемого уровня.

Для полного исключения проскальзываний необходимо наличие во входном потоке избыточных битов. В данном случае узел сети должен распознавать и уничтожать 1000 избыточных битов каждую секунду.

#### Ответы на вопросы, приведенные в п. 2.32.1 (с. 153)

Задержка передачи тестового сигнала по цепи из мультиплексоров может быть любой, разумеется, в пределах допустимых дальности связи и соотношения “сигнал/шум” в каналах. BER-тестер реагирует только на нарушение закономерности построения псевдослучайной последовательности битов и нечувствителен к задержке сигнала.

Скорость передачи тестовых данных в каналах Е1 задаётся оператором в процессе программирования карты распределения таймслотов. В данном примере тестовым данным выделены четыре таймслота (1 — 4), что соответствует скорости передачи данных  $64 \times 4 = 256$  кбит/с.

Чтобы увеличить (уменьшить) скорость передачи тестовых данных, достаточно в мультиплексоре, имитирующем АТС1, увеличить (уменьшить) число таймслотов, выделенных для передачи тестовых данных. Минимальная скорость соответствует выделению одного таймслота (64 кбит/с) максимальная — выделению 31 таймслота  $64 \times 31 = 1984$  кбит/с. Остальные мультиплексоры не требуют дополнительных настроек.

#### Ответ на вопрос, приведенные в п. 2.32.2 (с. 154.)

В отсутствие фатальных ошибок (отказов мультиплексоров, обрывов кабелей и т. п.) система передачи данных, как предполагаем, остаётся работоспособной, несмотря на наличие ошибок в каналах связи.

В отличие от предыдущей схемы (Рис. 2.121), схема, представленная на Рис. 2.122, позволяет более полно тестировать каналы связи. Точнее, в данном случае можно проследить за интенсивностью ошибок, возникающих в каждом направлении передачи тестовых данных.

В части удобства использования данная схема уступает предыдущей (Рис. 2.121), так как в проведении экспериментов должны участвовать два оператора, работающих на противоположных сторонах канала связи между АТС1 и АТС2. В предыдущей схеме петля возврата данных в мультиплексоре, имитирующем АТС2, может устанавливаться дистанционно, поэтому проверять канал связи может один оператор.

#### Ответ на вопрос, приведенный в п. 2.32.3 (с. 154)

Преимущество схемы, представленной на Рис. 2.123, по сравнению со схемой, представленной на Рис. 2.122, состоит в том, что нарушение или исчезновение синхросигнала на входе порта А мультиплексора, имитирующего АТС2, не приводит к нарушению или исчезновению сигнала на выходе этого порта. Поэтому повышается степень локализации ошибок — ошибки из верхнего канала (АТС1 → АТС2) не проходят в нижний канал (АТС2 → АТС1) и не искажают результат его тестирования.

#### Ответ на вопрос, приведенный в п. 2.32.4 (с. 155)

Преимущество схемы (Рис. 2.124) по сравнению с предыдущими (Рис. 2.121 — Рис. 2.123) состоит в повышении степени локализации ошибок. Ошибки на дистанции “Мультиплексор 2 — АТС2” регистрируются всеми тремя анализаторами BER-тестеров. Ошибки на

дистанции “Мультиплексор 1 — Мультиплексор 2” регистрируются только анализаторами BER-тестеров, размещённых в АТС1 и мультиплексоре 1. Ошибки на дистанции “АТС1 — Мультиплексор 1” регистрируются только анализатором BER-тестера, расположенного в АТС1. В зависимости от результатов тестирования можно определить дистанцию, на которой возникают ошибки. Однако остаётся неясным, какой канал (верхний или нижний по схеме) этой дистанции подвержен ошибкам.

В части удобства использования данная схема уступает предыдущим, так как в проведении экспериментов должны участвовать три оператора (а не один или два).

#### Ответы на вопросы, приведенные в п. 2.32.5 (с. 155)

Преимущество этой схемы перед предыдущими (Рис. 2.121 — Рис. 2.124) состоит в наиболее полной локализации участка канала, в котором появляются ошибки. Например, если таким участком является общий для всех маршрутов участок “мультиплексор 1 — мультиплексор 2” и внешние помехи в равной мере действуют на передачу данных в обоих направлениях, то ошибки регистрируются анализаторами всех четырёх BER-тестеров.

Если внешним помехам подвержен участок канала “АТС1 — мультиплексор 1”, или “АТС2 — мультиплексор 2”, то ошибки регистрируются одновременно анализаторами BER-тестеров, размещённых в этих АТС. При этом анализаторы BER-тестеров, расположенных в мультиплексорах 1 и 2, не обнаруживают ошибок, так как при передаче данных между портами Ethernet “зашумлённый” участок канала Е1 не используется.

Если помехи искажают данные, передаваемые только в одном направлении, то это направление также можно определить по результатам тестирования.

В части удобства использования данная схема уступает предыдущим, так как в проведении экспериментов должны участвовать четыре оператора (а не один, два или три).

#### Ответы на вопросы, приведенные в п. 2.32.6 (с. 155)

Структура и функционирование эластичной памяти (буфера типа FIFO) рассмотрены в п. 2.29.

В потоке Е1 нет избыточных битов, манипулируя которыми можно было бы поддерживать уровень заполнения буферной памяти на уровне, близком к отметке “50%”. Поэтому уровень заполнения буферной памяти неуклонно повышается или снижается в зависимости от соотношения скоростей втекающего и вытекающего потоков данных. В процессе повышения или снижения уровня проскальзываний синхронизации не возникает. Опустошение или переполнение буферной памяти приводит к повторной передаче или потере группы кадров. Однако лучше получать такого рода искажение, скажем, один раз в минуту, чем каждую секунду терять несколько кадров. В любом случае необходимо поддерживать кадровую синхронизацию, что представляет собой непростую задачу.

Проскальзывания синхронизации наблюдаются только на маршруте передачи данных из АТС1 в АТС2. Передача данных в обратном направлении не подвержена проскальзываниям, так как она осуществляется под управлением не двух, а одного генератора синхросигналов, расположенного в мультиплексоре 1. По той же причине не подвержены проскальзываниям передачи данных между сегментами X и Y сети Ethernet и передачи потоков данных через порты Р1 мультиплексоров 1 и 2. В данном примере и те, и другие передачи не контролируются.

## 4. Литература

1. С.М.Сухман, А.В.Бернов, Б.В.Шевкопляс. Синхронизация в телекоммуникационных системах. Анализ инженерных решений. — М.: Эко-Трендз, 2003. Электронная версия книги: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
2. Б.В.Шевкопляс. Флаг начала информационного кадра: каким он должен быть? — Схемотехника, 2006, № 12. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
3. Б.В.Шевкопляс. Особенности управления потоками данных в интерфейсах семейства RS. — Схемотехника, 2004, № 9, с. 34 — 36, № 10, с. 34 — 36. [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
4. Б.В.Шевкопляс. Скремблирование передаваемых данных — Схемотехника, 2004, №12, с. 24 — 27, 2005, №1, с. 29 — 32, 2005, №2, с. 32 — 35, 2005, №3, с. 30 — 33. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
5. Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: вставка команд в поток данных без использования избыточных битов — Схемотехника, 2005, №5, с. 23 — 25, №6, с. 23 — 26. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
6. Б.В.Шевкопляс. Передача данных по фантомным цепям телекоммуникационных систем — Схемотехника, 2007, №6, с. 14 — 16. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
7. Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: разграничение байтов в битовом потоке данных. — Схемотехника, 2005, №8, с. 23 — 26, №9, с. 23 — 25. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
8. Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: разграничение каналов в мультиплексированном потоке данных. — Схемотехника, 2005, №11, с. 17 — 19, №12, с. 18 — 22. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
9. Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: ускорение передачи информационных кадров. — Схемотехника, 2006, №1, с. 19 — 22, №2, с. 16 — 18. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljjas\\_b\\_w/](http://lit.lib.ru/s/shewkopljjas_b_w/).
10. П.Хоровиц, У.Хилл. “Искусство схемотехники”: В трех т. — М.: Мир, 1993. — 2 т.
11. Европейский патент EP 0.340.694 A2. <http://ep.espacenet.com/>.
12. Б.В.Шевкопляс. Микропроцессорные структуры. Инженерные решения. Дополнение первое: Справочник. — М.: Радио и связь, 1993.
13. Патент США № 5.530.959. <http://www.uspto.gov>.
14. Пат. США № 6.215.835 B1. <http://www.uspto.gov>.
15. Пат. США № 5.179.592. <http://www.uspto.gov>.
16. Пат. США № 5.835.602. <http://www.uspto.gov>.
17. Пат. США № 4.920.545. <http://www.uspto.gov>.
18. Пат. США № 6.445.719 B1. [www.uspto.gov](http://www.uspto.gov).
19. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. — СПб: Изд-во “Питер”, 2000. — 672 с.
20. Рекомендации ITU-T G.703, G. 704.
21. Пат. США № 6.185.216 B1. [www.uspto.gov](http://www.uspto.gov).