Вероятностная синхронизация в телекоммуникационных системах: разграничение каналов в мультиплексированном потоке данных

В предыдущих статьях [1 – 3] предложен новый способ установления и поддержания синхронизации между удалёнными друг от друга телекоммуникационными устройствами и рассмотрены примеры построения систем передачи данных на его основе. Предложенный способ синхронизации основан на “вероятностном” взаимодействии между устройствами, в том смысле, что моменты координации их систем отсчёта времени непредсказуемы на уровне битовых интервалов и определяются лишь “законом больших чисел”. В данной статье предлагается идея построения и пример схемного решения двухканальной системы “мультиплексор – демультиплексор”, предназначенной для объединения/разделения потоков данных при их передаче по линии связи. Как и в ранее описанных системах, применение вероятностной синхронизации позволяет сократить число избыточных битов до пренебрежимо малого уровня, исключить программные средства поддержки и уменьшить время восстановления синхронизации в случае её потери.

В традиционных системах мультиплексированной передачи данных с временным разделением каналов (Time Division Multiplexing, TDM) потоки битов от разных источников смешиваются в линии связи, а затем на удалённой стороне распределяются по соответствующим направлениям (рис. 1, *а*). При передаче суммарного потока данных по линии связи в ту или иную сторону каждому каналу отводится определённый временной интервал. Чтобы упорядочить последовательность этих интервалов, данные группируются в информационные кадры. В каждом кадре размещение канальных интервалов фиксировано. Для распознавания границ между кадрами в начале каждого из них помещается некоторый код, обычно именуемый флагом.

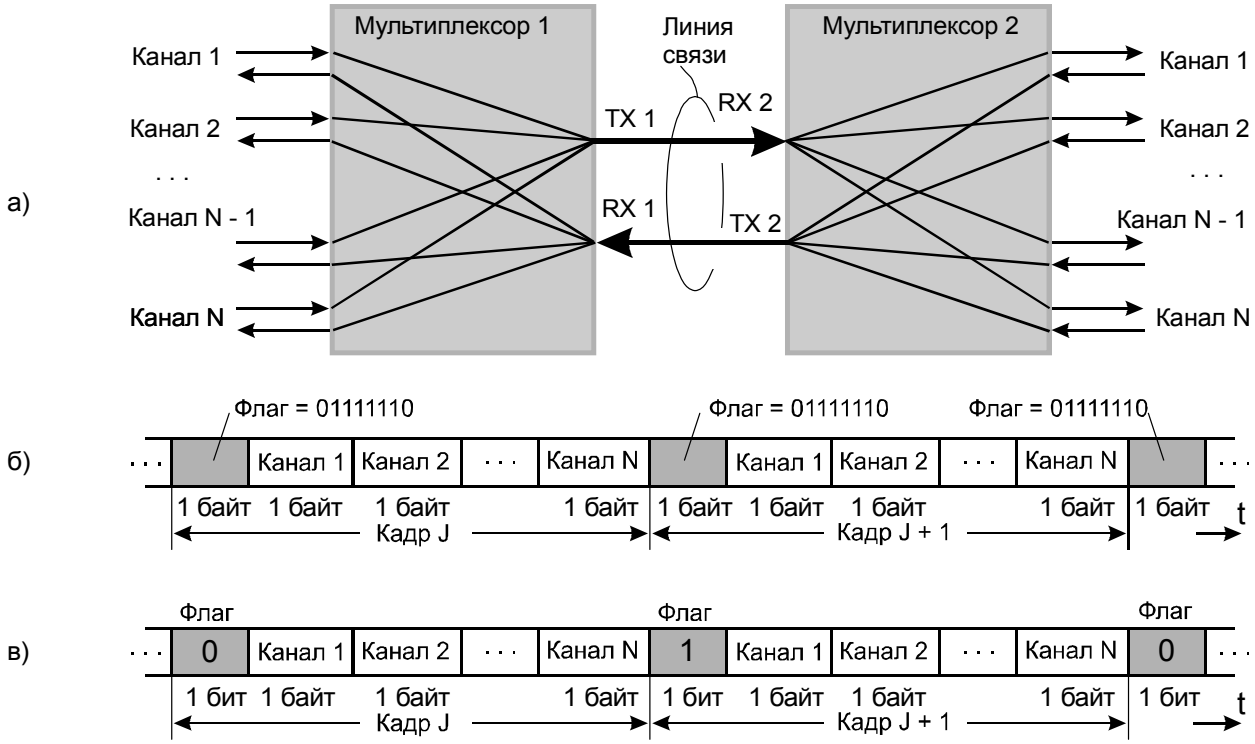

Рис. 1. Традиционная система мультиплексированной передачи данных с временным разделением каналов: *а* – структурная схема; *б, в* – структура одного из потоков данных в линии связи (TX 1 → RX 2 или TX 2 → RX 1) с использованием многоразрядного и одноразрядного флага

Многоразрядный флаг (рис. 1, *б*) может быть представлен уникальным или неуникальным кодом. Уникальность флагового кода достигается введением в тело кадра избыточных битов с помощью процедуры битстаффинга [4]. После выполнения этой процеду-

ры ложные копии флага обратимо искажаются, а истинный флаг остаётся неизменным и однозначно распознаётся приёмником как признак начала кадра. Использование многоразрядного неуникального флага также допустимо; при этом в потоке битов истинные флаги распознаются на фоне ложных в результате непрерывного сбора и анализа статистических данных о положении тех и других во времени. Это, однако, приводит к задержке установления кадровой синхронизации между передатчиком и приёмником и связано с применением относительно сложных программных средств.

Минимально возможная длина флага составляет один бит (рис. 1, *в*) [4, 5]. Идея использования такого флага заключается в следующем. В установившемся режиме после вхождения мультиплексоров в кадровый синхронизм каждый мультиплексор помечает передаваемые им в линию последовательные кадры фиксированной длины чередующимися одноразрядными флаговыми битами. Некоторый кадр с номером J (нумерация условная) содержит флаг, равный лог. 0. Следующий кадр с номером J + 1 имеет флаг, равный лог. 1. Последующие кадры содержат флаги, равные соответственно 0, 1, 0 и т. д.

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

В отсутствие ошибок наблюдается правильное чередование флаговых битов. Ошибки передачи, например связанные с проскальзываниями [4], могут привести к нарушению такого чередования. Ошибки неизбежно обнаруживаются, но, возможно, со значительной задержкой из-за случайных ложных совпадений битов данных с ожидаемыми флаговыми битами. При этом кадровая синхронизация теряется. Её восстановление проводится в несколько этапов, требует больших затрат времени и выполняется с использованием относительно сложных программных средств.

Предлагаемое далее решение, основанное на вероятностной синхронизации, снижает избыточность, вносимую в мультиплексированный битовый поток, практически до нулевого уровня, уменьшает время восстановления синхронизации и при этом не требует применения каких-либо программных средств.

Для упрощения описания далее рассмотрена двухканальная система и прослеживается только одно направление передачи (слева направо), так как противоположное направление реализуется с использованием аналогичных аппаратных средств. Рассмотренную далее идею вероятностной синхронизации можно применить и для построения многоканальных систем передачи данных.

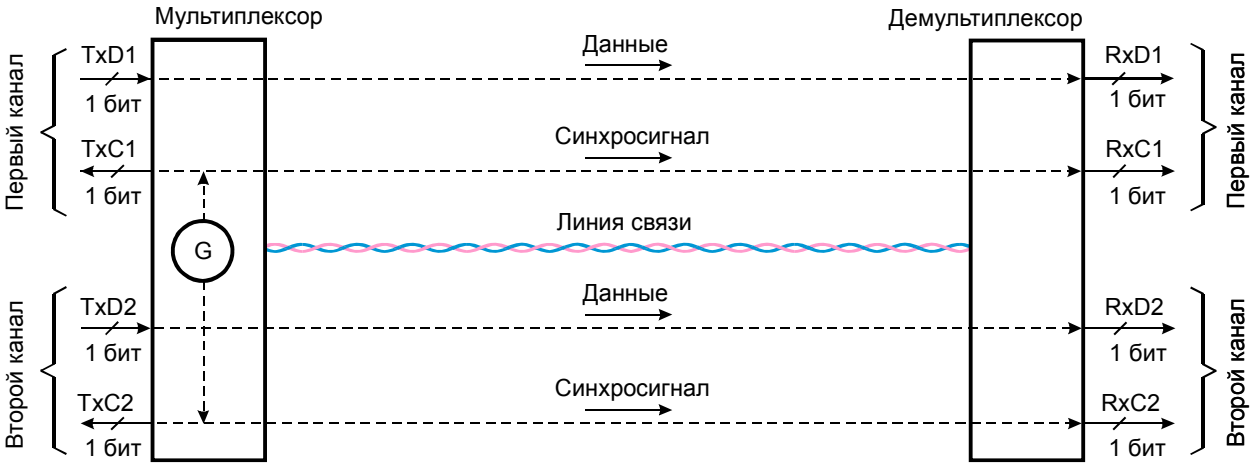

Рис. 2. Потоки данных и синхросигналов в предлагаемой системе “мультиплексор – демультиплексор”

В предлагаемой системе данные первого и второго каналов и сопровождающие их синхросигналы смешиваются мультиплексором, пересылаются по линии связи в удалён-

ный демультиплексор и распределяются по соответствующим каналам (рис. 2). Трассы сигналов, показанные штриховыми линиями, в действительности пролегают внутри линии связи – витой пары проводов.

Все процессы, протекающие в системе, синхронизируются от размещённого в мультиплексоре генератора G непрерывной последовательности импульсов. Данные TxD1 и TxD2 поступают на входы системы в ответ на положительные фронты синхросигналов TxC1 и TxC2. Выходные данные RxD1 и RxD2 сопровождаются синхросигналами RxC1 и RxC2. Синхросигналы TxC1 и TxC2 (RxC1 и RxC2) не совпадают во времени; их периодичность иногда нарушается в связи с обнаружением некоторых событий (дешифрации определённых кодовых комбинаций в линии связи), имеющих вероятностный характер.

Прежде чем перейти к более детальному описанию предлагаемой системы, рассмотрим её общую структуру и идею применения вероятностной синхронизации для разграничения каналов в битовом потоке данных (рис. 3).

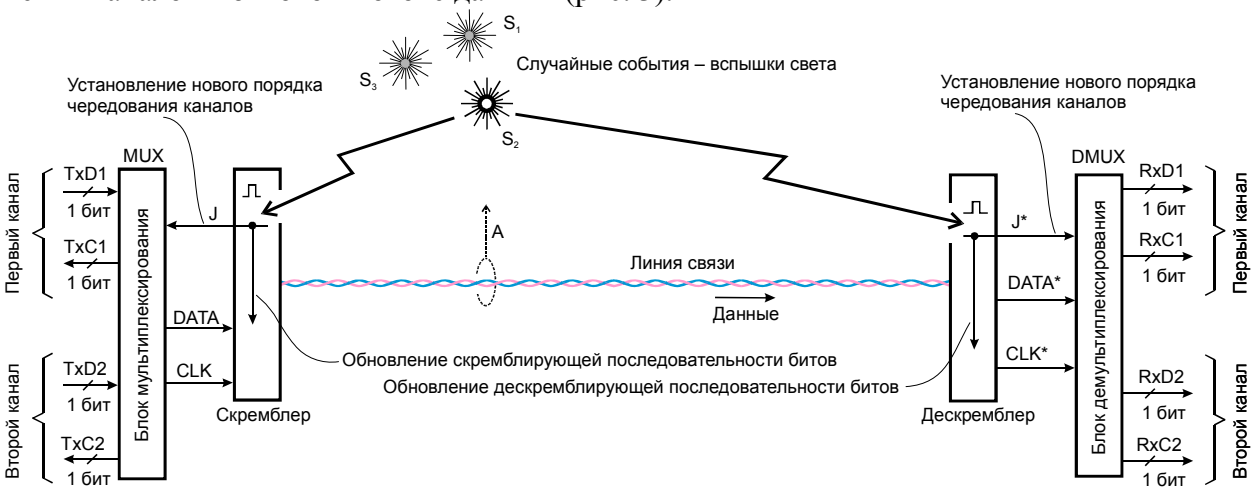

Рис. 3. Идея построения предлагаемой системы передачи данных

Передающая аппаратура (мультиплексор) содержит блок мультиплексирования и скремблер [1]. Приёмная аппаратура (демультиплексор) включает в себя дескремблер и блок демультиплексирования. Блок мультиплексирования объединяет потоки данных TxD1 и TxD2 в единый поток данных DATA, сопровождаемый непрерывным периодическим синхросигналом CLK. Скремблер преобразует поток данных к виду, при котором последовательность битов, выдаваемых в линию связи, может рассматриваться как случайная. Дескремблер восстанавливает сигналы DATA* = DATA и CLK* = CLK, а блок демультиплексирования распределяет суммарный поток данных DATA* между первым и вторым каналами.

Основная задача состоит в определении приёмной аппаратурой принадлежности каждого получаемого бита первому или второму каналам. Для её решения используется вероятностная синхронизация.

Предположим, что в некоторый момент произошло одно из трёх несовместных случайных событий S_1 – S_3. Эти события порождаются определёнными кодовыми ситуациями в линии связи (стрелка A на рис. 3), точнее, они связаны с обнаружением в потоке скремблированных данных некоторых заранее заданных кодов. На рис. 3 событие S_2 условно показано в виде вспышки света в точке, расположенной ближе к передающей аппаратуре, чем к приёмной. Вспышка регистрируется сначала передающей, а затем и приёмной аппаратурой. В результате регистрации этого события в скремблере и дескремблере формируются импульсы J и J*. Эти импульсы определяют новый порядок чередования битов в линии связи, т. е. служат опорными метками, ориентирами во времени для указания принадлежности передаваемых битов данных соответствующим каналам.

Если предыстория системы такова, что к моменту формирования импульсов J и J* синхронизация уже была достигнута и новый порядок чередования каналов совпадает со

старым, то эти импульсы игнорируются блоками мультиплексирования и демультиплексирования. Если синхронизация не была достигнута или передача данных велась правильно, но новый порядок чередования каналов противоположен старому, то в этом случае блоки мультиплексирования и демультиплексирования на один такт “сбивают шаг” и синхронно устанавливают новый единый порядок чередования каналов.

Импульсы J и J* используются также для синхронизации работы скремблера и дескремблера. Каждое из событий S_i вызывает переход к соответствующим начальным точкам скремблирующей и дескремблирующей последовательностей битов (подробное описание этого процесса приведено в [1]). Это позволяет первоначально установить кодовую синхронизацию, а в случае её потери – автоматически восстановить правильную работу подсистемы “скремблер – дескремблер” при регистрации первого же события S_i.

Преимущества вероятностной синхронизации связаны с тем, что для формирования опорных меток времени (импульсов J и J* – отдалённых аналогов флагов начала кадров) не нужно прикладывать никаких “усилий” – эти метки возникают спонтанно по мере передачи полезных данных, без введения в поток данных служебных битов. Средняя частота следования меток может регулироваться разработчиком системы или программироваться пользователем. Она определяется скоростью передачи данных, разрядностью искомых (дешифрируемых) кодов и их числом.

Рассмотрим предлагаемую систему подробнее (рис. 4). Начнём с описания её составных частей.

Сдвиговые регистры RG1 и RG2 предназначены для временного хранения фрагментов SDATA и SDATA* потока скремблированных данных и реализуют функцию “скользящего окна”, через которое просматривается битовая последовательность. В установившемся режиме эти фрагменты одинаковы (совпадают с точностью до задержки передачи). Приём очередного бита в регистр RG1 (RG2) происходит по положительному фронту сигнала на синхронизирующем входе C этого регистра. Одновременно с приёмом очередного бита с входа D ранее хранимые в регистре данные сдвигаются на один разряд вправо. В данном примере построения системы разрядность регистров RG1 и RG2 выбрана равной восьми, хотя она может быть большей или меньшей.

Генератор W псевдослучайной последовательности битов скремблера выполнен на основе сдвигового регистра RG3. Аналогичный генератор на основе сдвигового регистра RG4 содержится в дескремблере. 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 с входа D происходит по положительному фронту сигнала на синхронизирующем входе C 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. Если на управляющем входе P/S (P/S*) регистра присутствует сигнал лог. 1, то по положительному фронту сигнала на синхронизирующем входе C в регистр принимается параллельный код с группы выходов дешифратора DC1 (DC2). В данном примере построения устройства разрядность регистров RG3 и RG4 выбрана равной пяти, хотя она может быть большей или меньшей. При формировании импульсов J и J* регистры RG3 и RG4 одновременно (с точностью до задержки передачи) устанавливаются в одинаковые состояния [1].

Элементы Исключающее ИЛИ D1 и D2 формируют скремблированный SCRD и дескремблированный DIN сигналы данных. Триггеры D-типа 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 многократно изменяться из-за “гонок”. Триггер D4 в значительной степени 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.

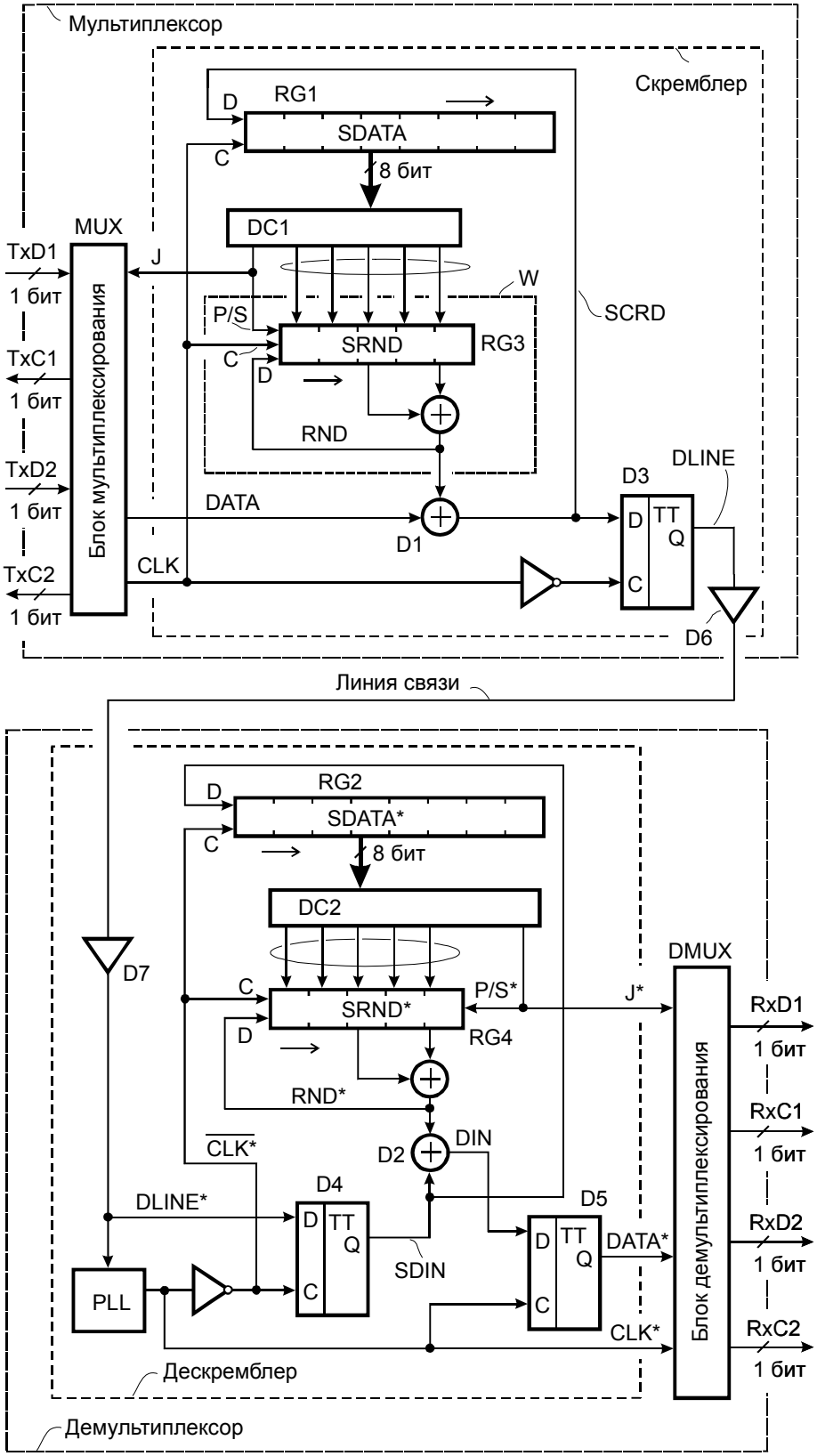

Рис. 4. Предлагаемая система передачи данных

Генератор PLL с фазовой автоподстройкой частоты может быть выполнен по одной из известных схем (см., например, [6]). Он предназначен для формирования высокостабильного синхросигнала CLK* на основе непрерывного слежения за входным сигналом DLINE*. Положительный фронт сигнала CLK* привязан к моментам изменения сигнала DLINE*, так что 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. (Такое использование генератора с фазовой автоподстройкой частоты в телекоммуникационных системах является общепринятым и далее не детализируется.)

Дешифратор DC1 (DC2) является ключевым звеном системы вероятностной синхронизации и предназначен для выделения в потоке скремблированных данных, проходящем через скользящее окно (сдвиговый регистр RG1 (RG2)), определенных кодов $CODE_1$, $CODE_2$, …, $CODE_K$. При обнаружении дешифратором DC1 (DC2) указанных кодов, на его выходах, обведённых на рис. 4 овалом, формируется соответствующий V-разрядный код $LOAD_1$, $LOAD_2$, …, $LOAD_K$ для последующей параллельной загрузки сдвигового регистра RG3 (RG4). В данном примере K = 4, V = 5 (подробнее – см. [1]). При обнаружении любого кода $CODE_1$, $CODE_2$, …, $CODE_K$ дешифратор DC1 (DC2) формирует также единичный сигнал J (J*) 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 определяются типом линии связи, которая в наиболее простом варианте может быть выполнена в виде витой пары проводов, коаксиального или оптоволоконного кабеля. Линия связи может содержать последовательно включённые ретрансляторы, в которых могут использоваться блоки буферной памяти. Поэтому задержка прохождения сигнала между аппаратурой передачи и приёма данных может быть значительной и заранее не известной (но постоянной).

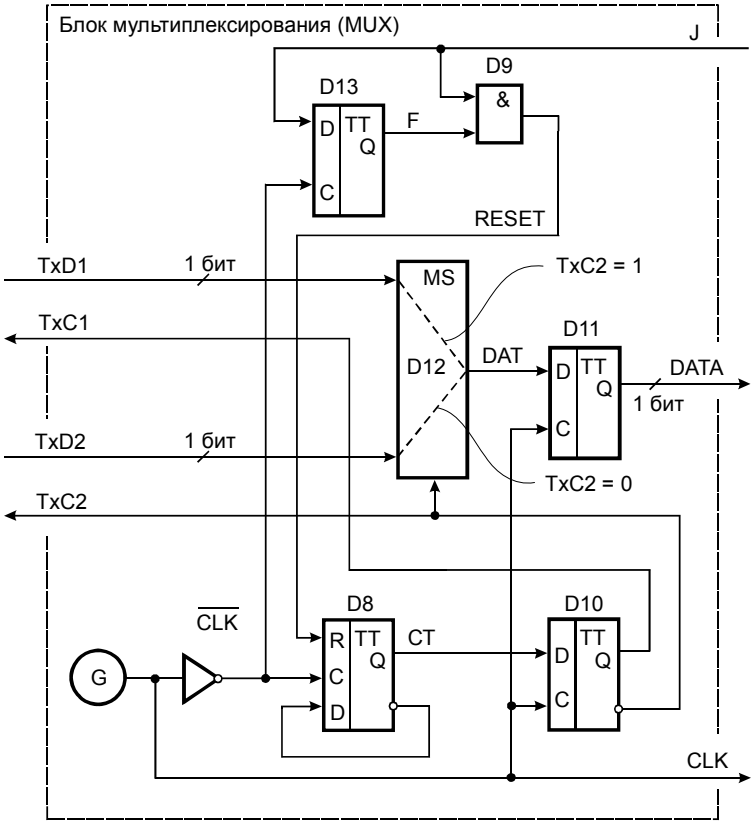

Рис. 5. Схема блока мультиплексирования

Генератор G синхросигналов, размещённый в блоке мультиплексирования данных (рис. 5), задаёт темп работы всей системы. На выходе генератора формируется непрерывная последовательность импульсов со скважностью, равной двум. Триггер D8 делит входную частоту на два, при этом фаза выходного сигнала CT в необходимых случаях корректируется импульсом RESET с выхода элемента И D9. Триггер D10 выполняет функцию повторителя сигнала CT и “очищает” этот сигнал от кратковременных ложных импульсов в моменты формирования сигнала RESET на выходе

элемента И D9 (см. диаграммы 32 и 33 на рис. 9). Сигналы TxC1 и TxC2 с выходов триггера D10 задают темп поступления данных TxD1 и TxD2 по первому и второму каналам.

Очередной бит данных TxD1 (TxD2) поступает от внешнего источника (на рисунках не показан) на вход системы в ответ на положительный фронт сигнала TxC1 (TxC2). Бит данных TxD1 (TxD2) может поступать от внешнего источника с некоторой задержкой, которая на временных диаграммах (рис. 7 – рис. 9) не учитывается для повышения их наглядности. Эта задержка (обозначенная символом T* в интервале T8 – T10 на диаграмме 10, рис. 7) может приближаться к двум периодам синхросигнала CLK при условии сохранения достаточного для устойчивой работы триггера D11 времени предварительной установки сигнала DAT на его входе D. Время предварительной установки сигнала DAT представляет собой интервал между окончанием периода T* и ближайшим положительным фронтом сигнала CLK (моментом T10).

Данные TxD1 и TxD2 первого и второго каналов смешиваются в едином потоке данных DAT с помощью мультиплексора D12. Как показано на рис. 5, при TxC2 = 1 мультиплексор D12 транслирует на выход данные TxD1 первого канала, а при TxC2 = 0 – данные TxD2 второго канала. Триггер D11 осуществляет привязку границ битовых интервалов сигнала DATA к положительным фронтам синхросигнала CLK. Триггер D13 и элемент И D9 формируют импульс RESET установки в нуль триггера D8 при получении на вход блока импульса J (диаграммы 16 – 18 и 29 – 31 на рис. 8 и рис. 9).

В блоке демультиплексирования данных (рис. 6) входной поток битов DATA* разделяется на два потока – RxD1 (данные первого канала) и RxD2 (данные второго канала). Разделение осуществляется триггерами D14 и D15, которые синхронизируются сигналами RxC1 и RxC2 с выходов триггера D16. Эти сигналы используются внешними приёмниками данных первого и второго каналов (приёмники на рисунках не показаны) для синхронного приёма битов RxD1 и RxD2.

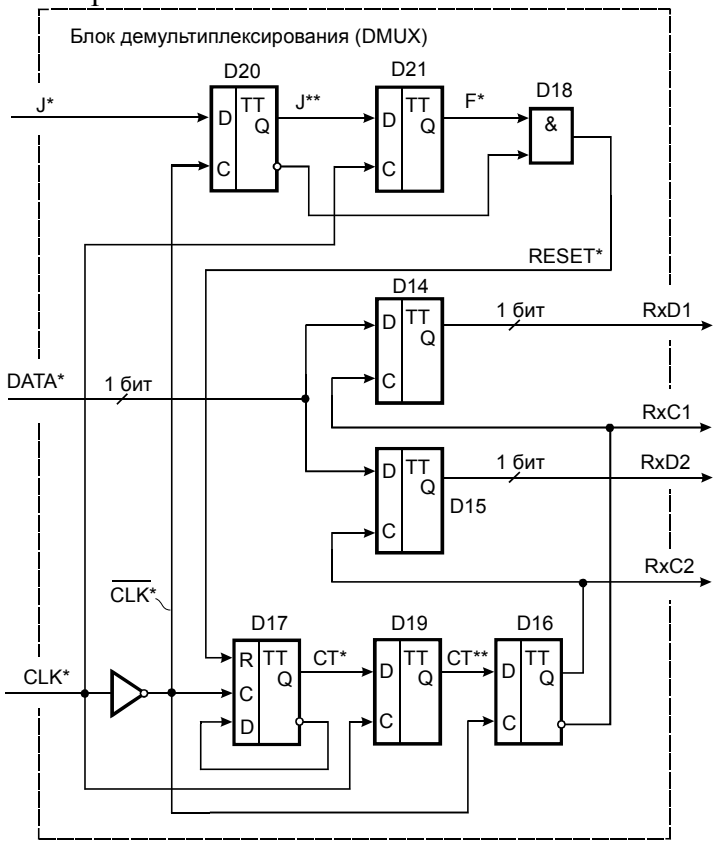

Рис. 6. Схема блока демультиплексирования

На входе битовой синхронизации блока демультиплексирования присутствует непрерывная последовательность импульсов CLK* (копия сигнала CLK от генератора G, см. рис. 5) со скважностью, равной двум. Триггер D17 делит входную частоту на два, при этом фаза выходного сигнала CT* в необходимых случаях корректируется импульсом RESET* с выхода элемента И D18. Триггер D19 выполняет функцию повторителя сигнала CT* и “очищает” этот сигнал от кратковременных ложных импульсов в моменты формирования сигнала RESET* на выходе элемента И D18 (см. диаграммы 110 и 111 на рис. 13, диаграммы 129 и 130 на рис. 14). Триггер D16 осуществляет привязку выходных сигналов блока демультиплексирования к отрицательным фронтам сигнала CLK*. Триггеры D20, D21 и элемент И D18 формируют импульс RESET* установки в нуль триггера D17 при получении на вход блока

импульса J* коррекции (диаграммы 82 – 85, 101 – 104, 120 – 123 и 139 – 142 на рис. 12 – 15).

Далее рассмотрена работа системы в целом.

При вероятностной синхронизации на передающей и приёмной сторонах линии связи одновременно (с точностью до задержки передачи) регистрируются случайные события, связанные с “самопроизвольным” возникновением в потоке скремблированных данных некоторых заранее заданных кодов. Эти моменты используются для коррекции состояния общей системы синхронизации. Сказанное поясняется рассмотренными далее вариантами функционирования системы в разных условиях, зависящих от предыстории и от моментов обнаружения упомянутых случайных событий.

При работе системы возможны шесть кодовых ситуаций, описанных в табл. 1.

Таблица 1. Возможные кодовые ситуации при передаче данных через систему

Наличие импульса J коррекции и реакция на него со стороны мультиплексора	Состояние демультиплексора		Номер кодовой ситуации
	Ранее установленная синхронизация	Правильность данных на выходах RxD1 и RxD2	
Импульс J корреции отсутствует (рис. 7)	Правильная	Данные правильные, темп выдачи постоянный. Импульс J* коррекции отсутствует (рис. 10)	1
	Неправильная, не восстанавливается	Данные неправильные. Импульс J* коррекции отсутствует (рис. 11)	2
Импульс J корреции порождён битом синхронизации из канала 2. Импульс J игнорируется (рис. 7, 8)	Правильная	Данные правильные, темп выдачи постоянный. Импульс J* коррекции игнорируется (рис. 10, 12)	3
	Неправильная, затем восстанавливается импульсом J*	Данные исходно неправильные. Становятся правильными и выдаются с постоянным темпом после получения импульса J* коррекции (рис. 11, 13)	4
Импульс J корреции порождён битом синхронизации из канала 1. Следующий за ним бит из канала 1 дублируется (рис 7, 9)	Правильная	Данные правильные, темп их выдачи однократно замедляется на один такт в связи с получением импульса J* коррекции (рис. 10, 14)	5
	Неправильная, затем восстанавливается импульсом J*	Данные исходно неправильные. Становятся правильными и выдаются с постоянным темпом после получения импульса J* коррекции (рис. 11, 15)	6

Кодовая ситуация № 1 (рис. 7, 10)

В этой ситуации в предыстории установлена правильная синхронизация между мультиплексором и демультиплексором. Блок мультиплексирования данных периодически запрашивает данные первого и второго каналов и объединяет получаемые в ответ биты в единый поток DATA, после чего скремблированные данные DLINE выдаются в линию связи, как показано на диаграммах 1 – 13 (рис. 7). На этом и последующих рисунках цифрами 1 и 2 обозначены биты, принадлежащие первому и второму каналам. Нумерация моментов времени на диаграммах, описывающих работу мультиплексора (рис. 7 – 9), не свя-

зана с нумерацией моментов времени на диаграммах, описывающих работу демультиплексора (рис. 10 – 15). Напомним, что задержка передачи сигналов по линии связи заранее не известна, так что системы отсчёта времени в мультиплексоре и демультиплексоре разные.

Дескремблер принимает скремблированные данные и восстанавливает их (см. диаграммы 40 – 52 на рис. 10). Блок демультиплексирования распределяет данные DATA* между первым и вторым каналами (см. диаграммы 53 – 58 на рис. 10).

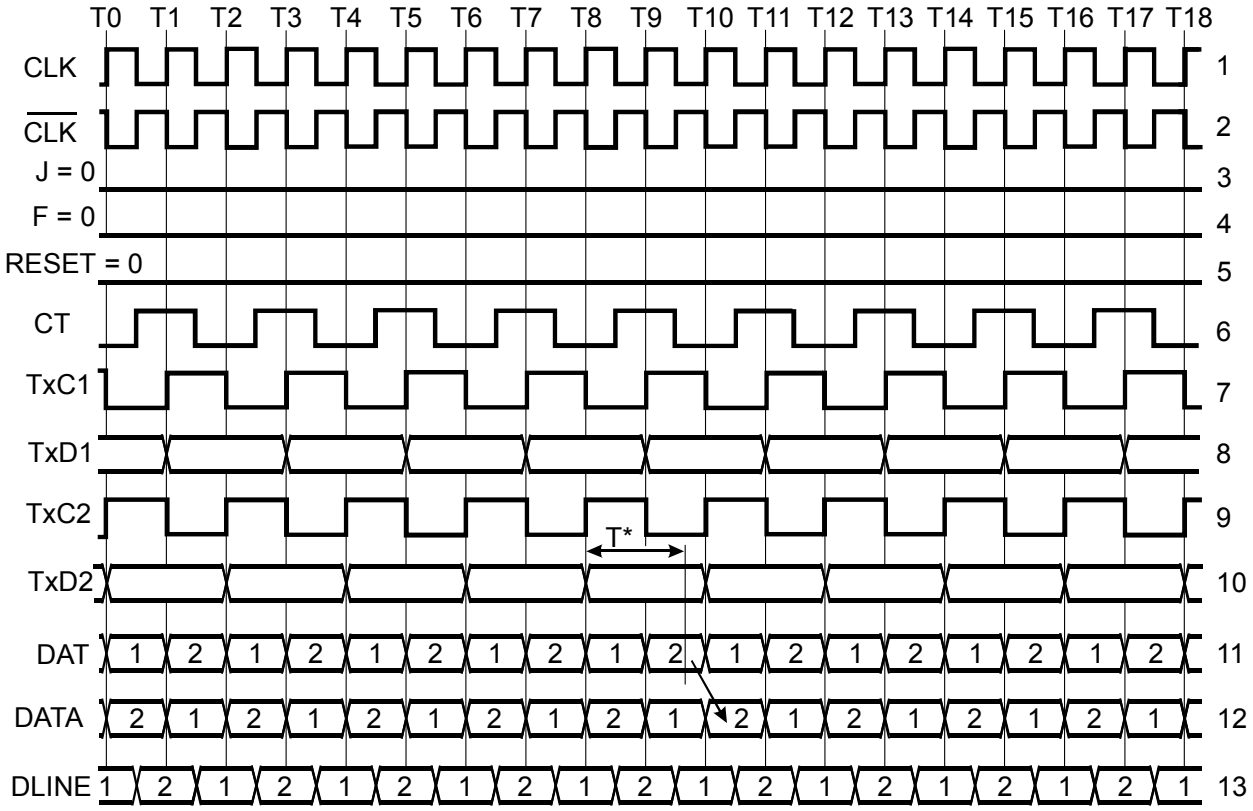

Рис. 7. Временные диаграммы работы мультиплексора в отсутствие сигнала J

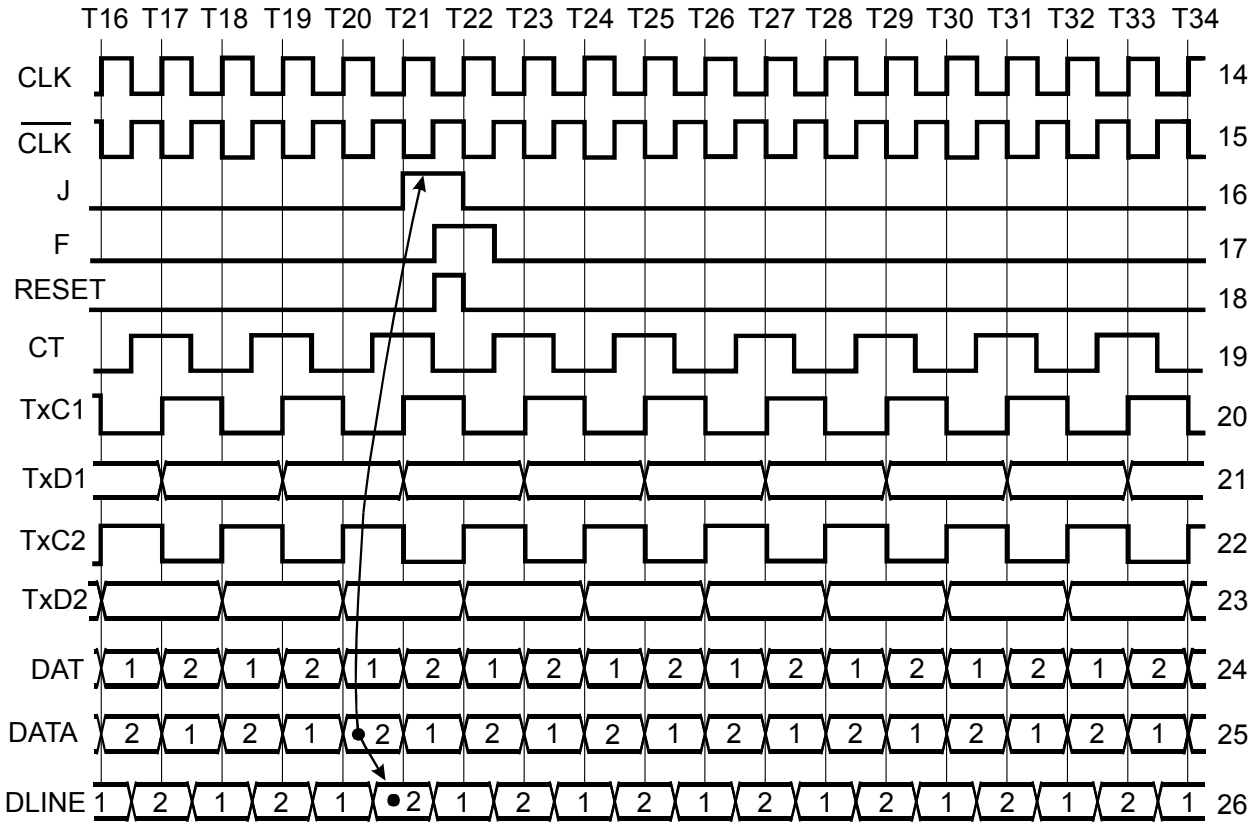

Рис. 8. Временные диаграммы работы мультиплексора при наличии сигнала J, порождённого битом синхронизации из канала 2

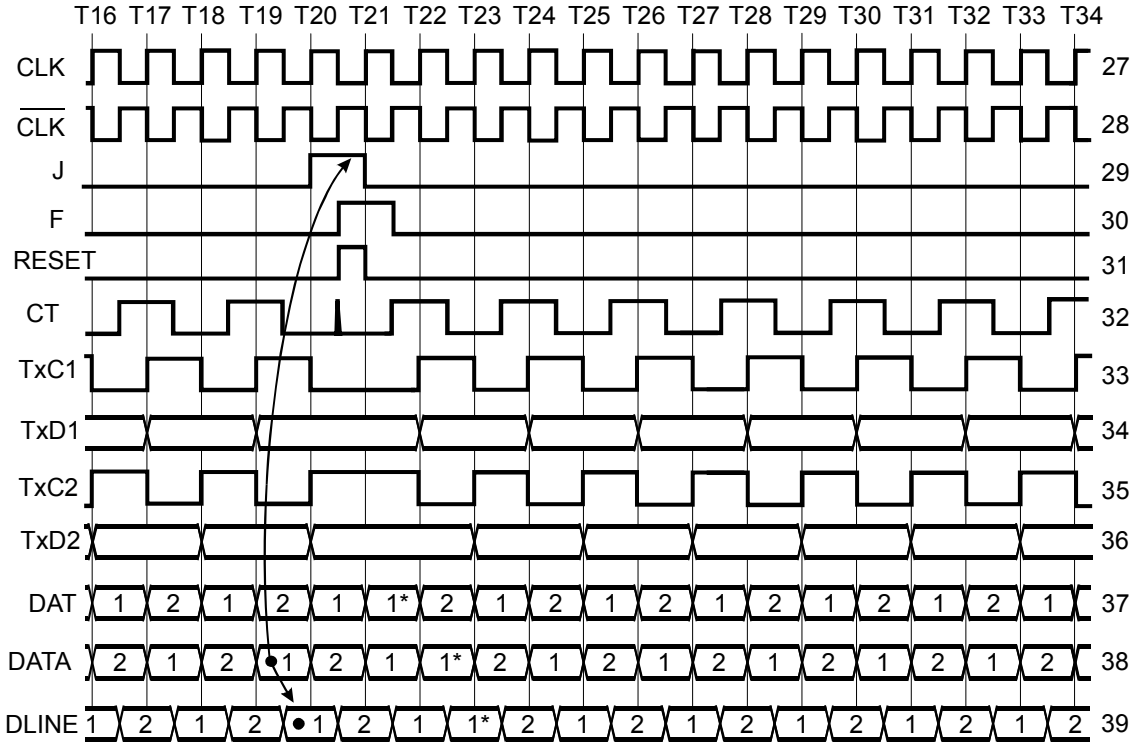

Рис. 9. Временные диаграммы работы мультиплексора при наличии сигнала J, порождённого битом синхронизации из канала 1

Кодовая ситуация № 2 (рис. 7, 11)

В этой ситуации, как и в предыдущей, блок мультиплексирования периодически запрашивает данные первого и второго каналов и чередует выдачу получаемых в ответ битов, после чего скремблированные данные DLINE выдаются в линию связи (см. диаграммы 1 – 13 на рис. 7). В данном случае демультиплексор исходно не синхронизирован с мультиплексором. Это в свою очередь означает, что по крайней мере одно из устройств приёмной стороны (дескремблер или блок демультиплексирования данных) функционирует неправильно.

Если неправильно работает дескремблер (отсутствует кодовая синхронизация со скремблером), то данные DATA* и, следовательно, сигналы RxD1 и RxD2 неправильны независимо от того, верно или неверно они распределяются между выходными каналами. Как отмечалось, синхронизация дескремблера с скремблером устанавливается всякий раз при формировании импульсов J и J*. С учётом этого работа подсистемы скремблер – дескремблер далее не рассматривается (подробности – см. [1]).

Если неправильно работает только блок демультиплексирования данных, то правильные данные DATA* распределяются по выходам неверно, а именно, данные первого канала выдаются на выход второго канала, и наоборот (см. рис. 11, диаграммы 59 – 77). Это происходит из-за того, что делитель частоты на триггере D17 (рис. 6) работает в неправильной фазе, а импульса J*, который мог бы скорректировать фазу, нет.

Кодовая ситуация № 3 (рис. 7, 8, 10, 12)

В данном случае в предыстории достигнута правильная синхронизация; работа системы соответствует временным диаграммам, приведенным на рис. 7, 10. Как показано на рис. 8, в процессе передачи заранее не известных данных случилось так, что в такте T20 – T21 сформирован бит данных DATA, который в дальнейшем порождает формирование импульсов J и J*. Этот бит помечен на диаграмме 25 точкой и далее для краткости назван *битом синхронизации*. Следует помнить, что этот бит не служебный – он принадлежит потоку пользовательских или иных “полезных” данных, в данном случае передаваемых по второму каналу.

Бит синхронизации в этом же такте (T20 – T21, рис. 8) скремблируется элементом Исключающее ИЛИ D1, а в следующем такте (T21 – T22) загружается в крайний левый разряд сдвигового регистра RG1, при этом ранее загруженные в этот регистр биты перемещаются в нём на одну позицию вправо. Полученный в такте T21 – T22 код SDATA анализируется дешифратором DC1, в результате на его выходе формируется сигнал J. Из этого сигнала формируется импульс RESET, который подаётся на вход установки нуля триггера D8. Однако этот триггер уже находится в нулевом состоянии, поэтому импульс RESET не оказывает на него воздействия. Фактически это означает, что импульс J в блоке мультиплексирования данных игнорируется.

Аналогичные процессы протекают в демультиплексоре (см. рис. 12). Бит синхронизации, принадлежащий второму каналу, вызывает формирование импульса RESET*, который также не оказывает воздействия на триггер D17, так как этот триггер работает в правильной фазе (см. диаграммы 78 – 96). Таким образом, в данной ситуации импульс J* в блоке демультиплексирования игнорируется.

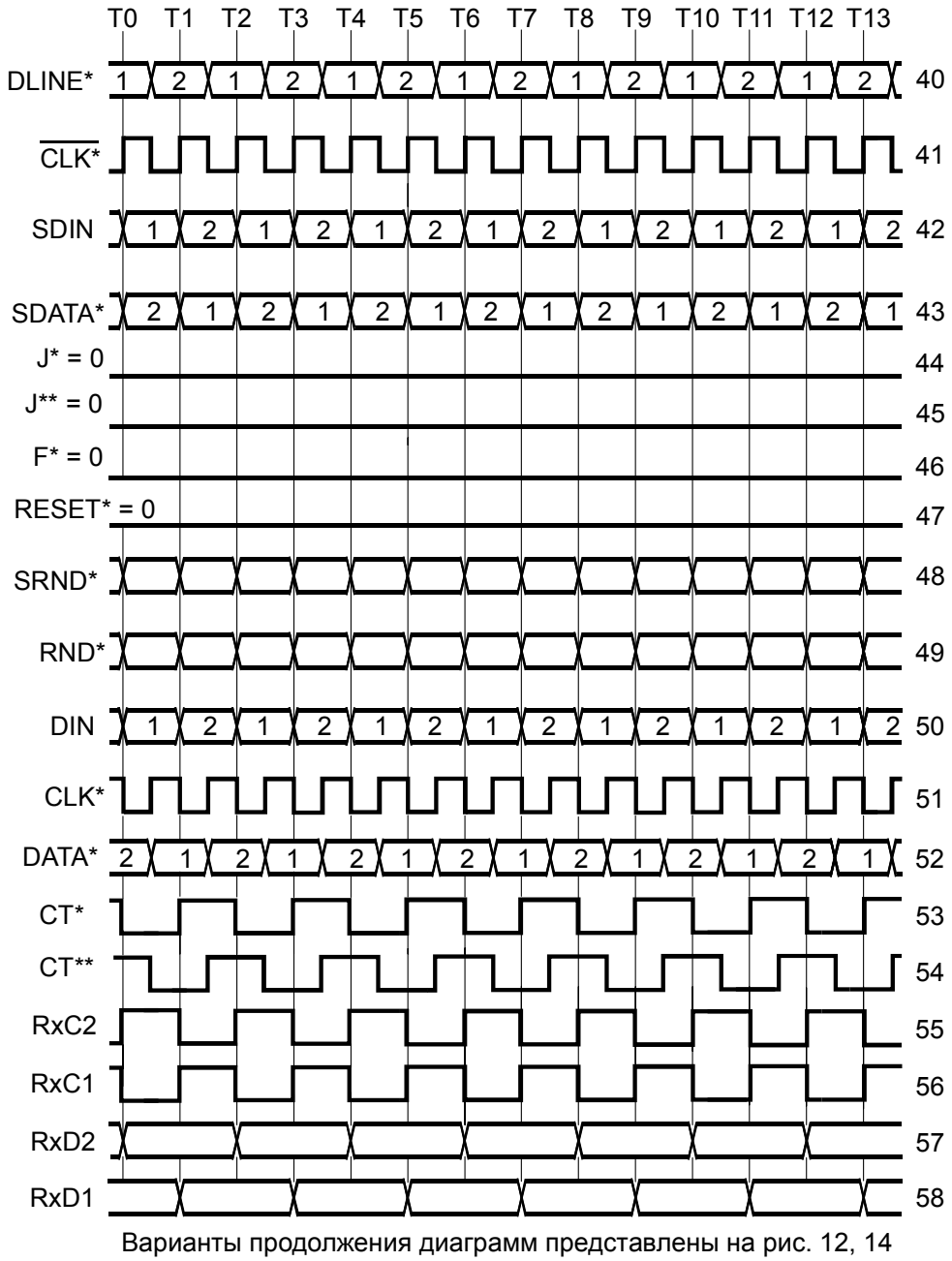

Рис. 10. Временные диаграммы правильной работы демультиплексора в отсутствие сигнала J*

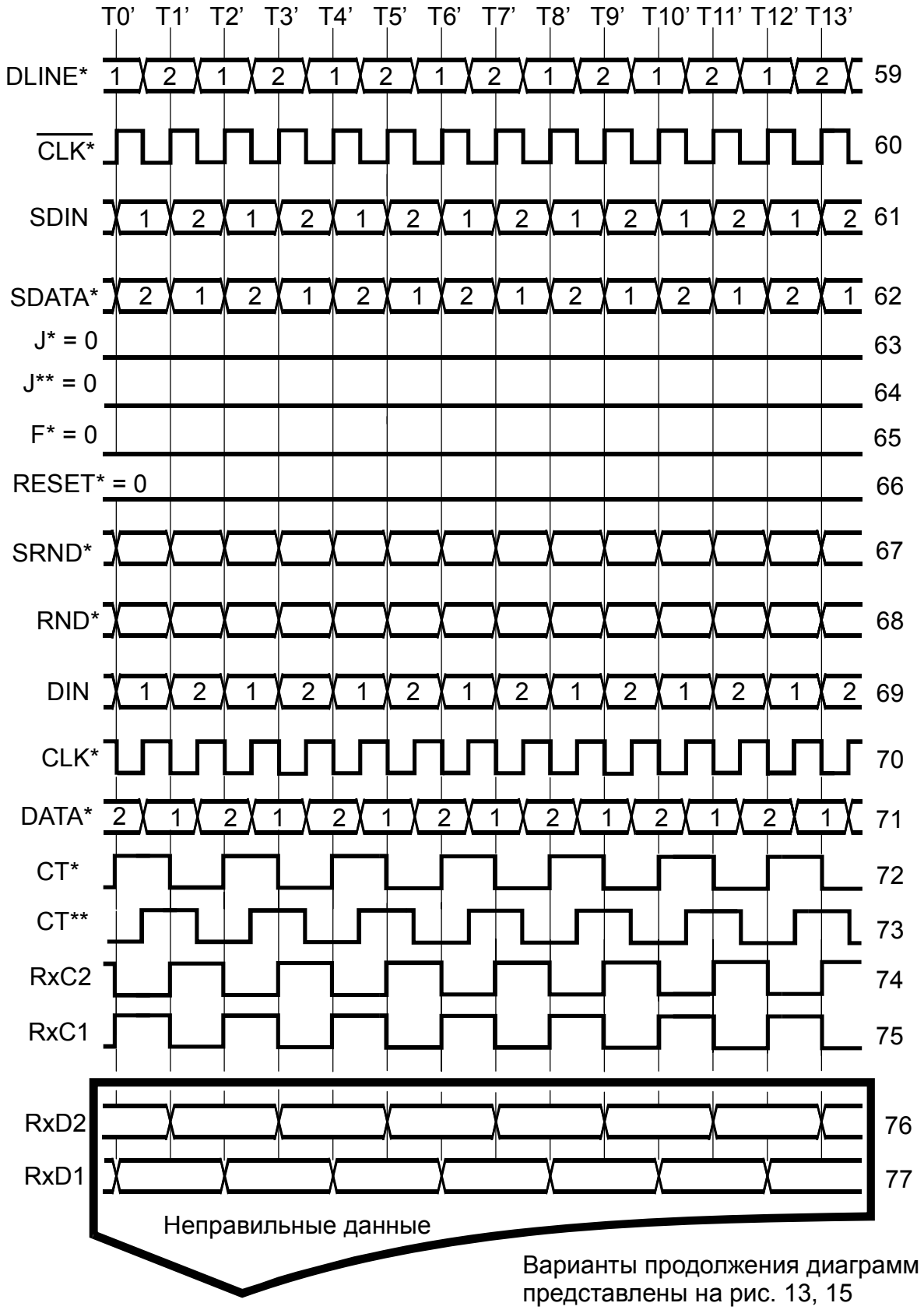

Рис. 11. Временные диаграммы неправильной работы демультиплексора в отсутствие сигнала J* (правильный поток данных DATA* распределяется по “чужим” выходным каналам)

Вариант продолжения диаграмм, представленных на рис.10

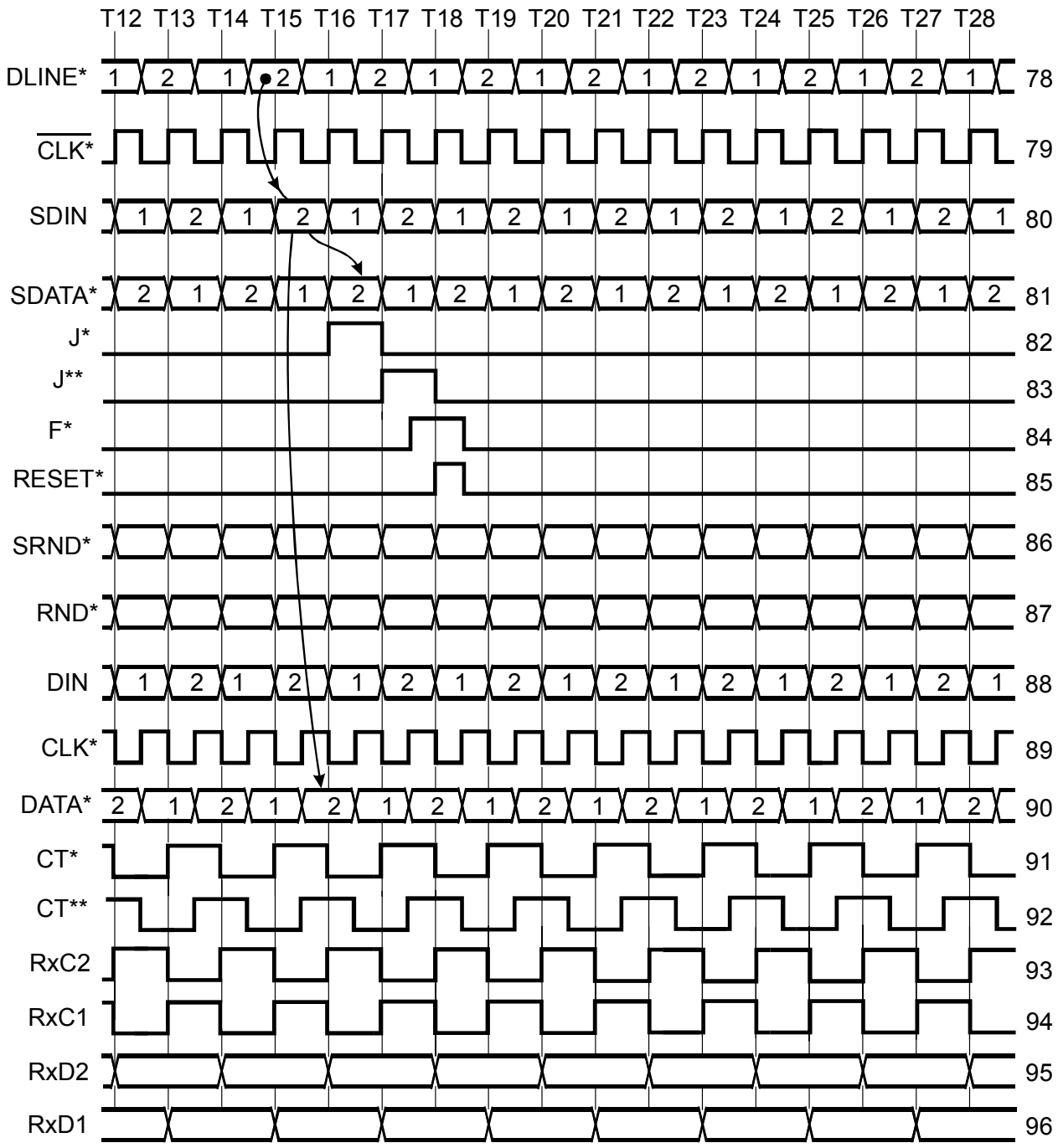

Рис. 12. Временные диаграммы правильной работы демультиплексора при наличии сигнала J*, порождённого битом синхронизации из канала 2

Кодовая ситуация № 4 (рис. 7, 8, 11, 13)

В предыстории синхронизация неправильная; работа устройства соответствует временным диаграммам, приведенным на рис. 7, 11. Как следует из диаграмм 14 – 26, приведенных на рис. 8, бит синхронизации, принятый по второму каналу, не нарушает ритма работы мультиплексора. Этот же бит проходит в демультиплексор и вызывает формирование импульса RESET*, который корректирует (изменяет на 180 градусов) фазу сигнала CT* на выходе триггера D17 (см. диаграммы 97 – 115 на рис. 13), в результате восстанавливается правильная синхронизация. Начиная с момента T21’ выходные данные становятся правильными.

Вариант продолжения диаграмм, представленных на рис. 11

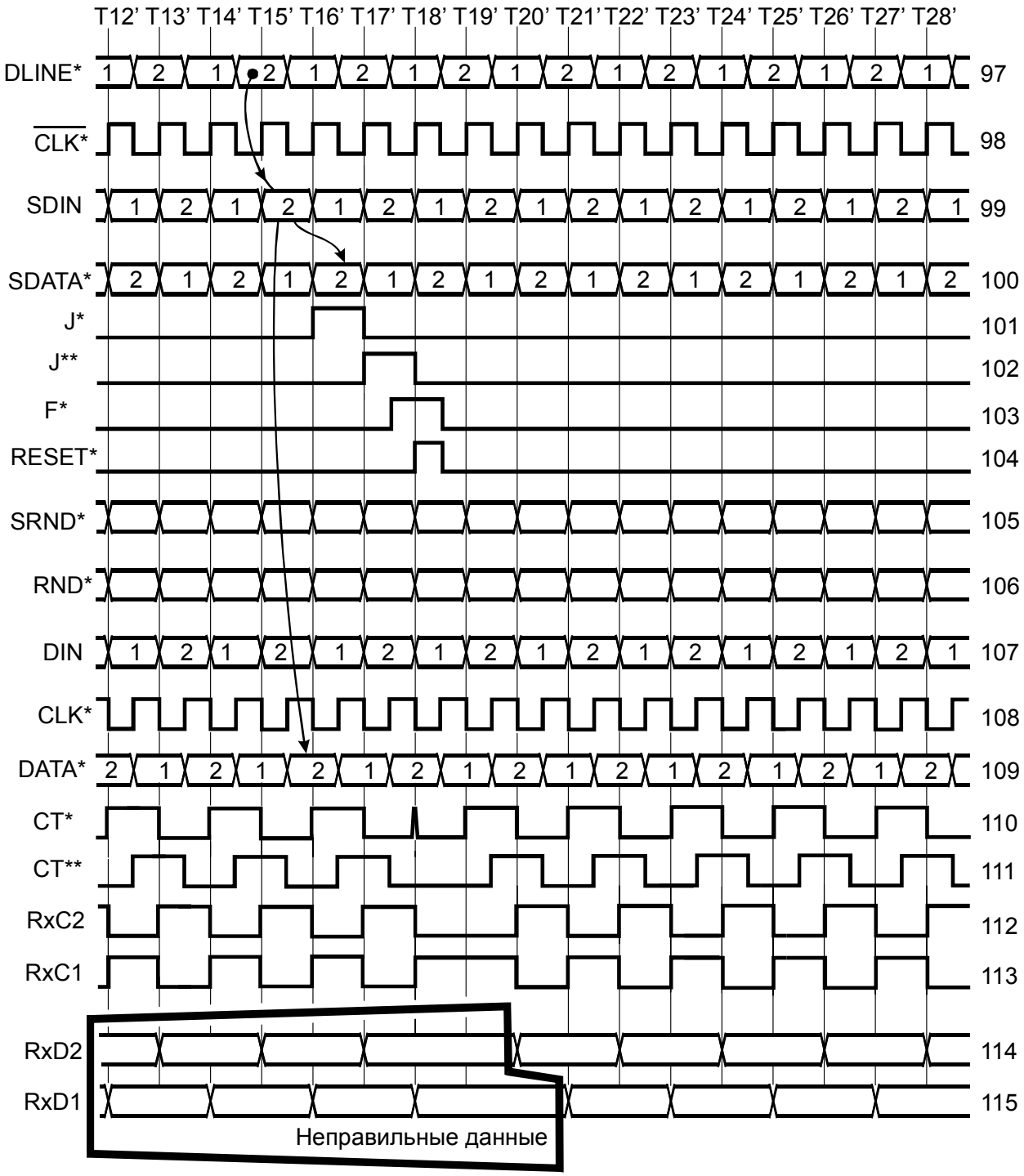

Рис. 13. Временные диаграммы восстановления правильной работы демультиплексора при формировании сигнала J*, порождённого битом синхронизации из канала 2

Вариант продолжения диаграмм, представленных на рис. 10

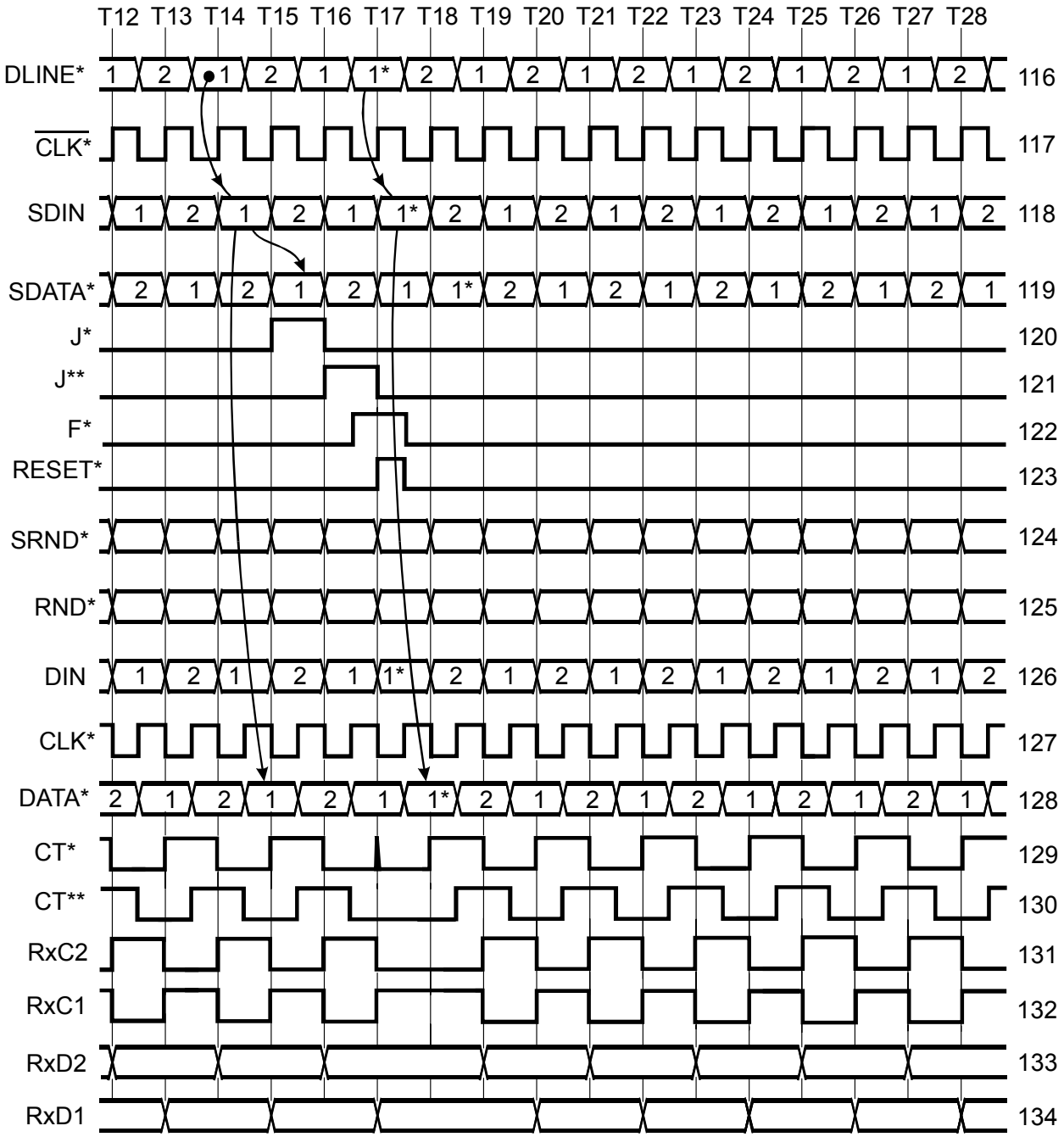

Рис. 14. Временные диаграммы, подтверждающие сохранение правильной работы демультиплексора при формировании сигнала J*, порождённого битом синхронизации из канала 1

Вариант продолжения диаграмм, представленных на рис. 11

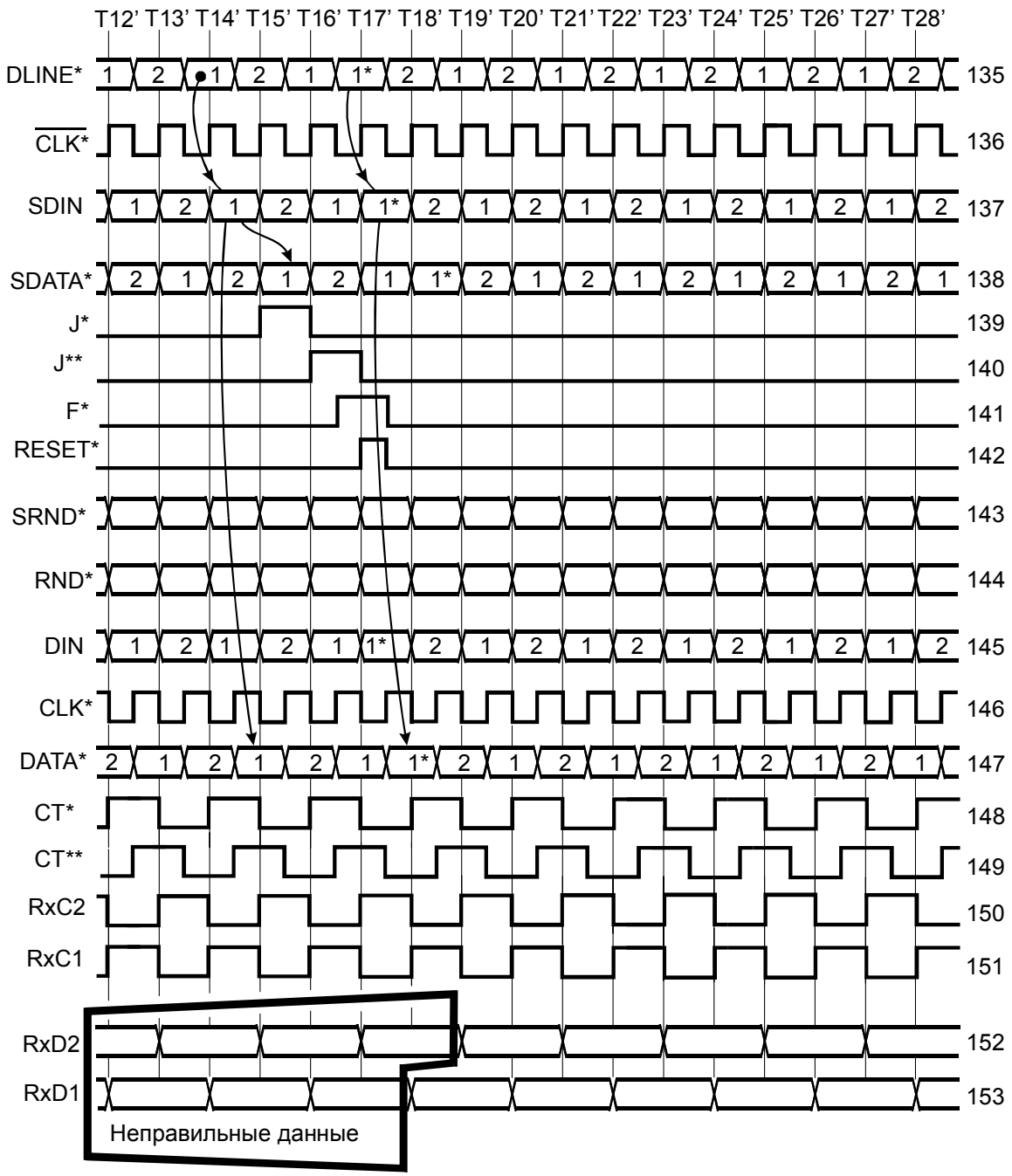

Рис. 15. Временные диаграммы восстановления правильной работы демультиплексора при формировании сигнала J*, порождённого битом синхронизации из канала 1

Кодовая ситуация № 5 (рис. 7, 9, 10, 14)

В предыстории синхронизация правильная; работа устройства соответствует временным диаграммам, приведенным на рис. 7, 10. В такте T19 – T20 (рис. 9) в потоке данных DATA обнаруживается принадлежащий первому каналу бит синхронизации (помечен точкой на диаграмме 38). В следующем такте (T20 – T21) этот бит после скремблирования записывается в регистр RG1, что порождает импульс J. В начале второй половины такта формируется импульс RESET (диаграмма 31). Этот импульс устанавливает в нуль триггер D8, который к этому моменту только что успел установиться в единицу. В результате на выходе триггера формируется кратковременный импульс (диаграмма 32). Он устраняется при перезаписи сигнала из триггера D8 в триггер D10. Воздействие импульса RESET на

триггер D8 приводит к нарушению периодичности сигналов TxC1 и TxC2 (диаграммы 33, 35). При этом бит из первого канала, следующий за битом синхронизации, дублируется при его выдаче в линию (диаграмма 39). Дублированный бит на диаграммах обозначен символом 1*.

Аналогичные процессы протекают в демультиплексоре (рис. 14). Темп выдачи данных однократно замедляется на один такт в связи с получением импульса J* коррекции. Данные передаются правильно.

Кодовая ситуация № 6 (рис. 7, 9, 11, 15)

В предыстории синхронизация неправильная; работа устройства соответствует временным диаграммам, приведенным на рис. 7, 11. В такте T19 – T20 (рис. 9) в потоке данных DATA обнаруживается принадлежащий первому каналу бит синхронизации (помечен точкой на диаграмме 38). В следующем такте (T20 – T21) этот бит после скремблирования записывается в регистр RG1, что порождает импульс J. В результате, как и в ситуации № 5, бит из первого канала, следующий за битом синхронизации, дублируется при его выдаче в линию (диаграмма 39).

В блоке демультиплексирования из импульса J* формируется импульс RESET*, который не оказывает воздействия на неправильную фазу сигнала CT* (диаграммы 135 –153 на рис. 15). Но воздействия в данном случае и не требуется, так как в потоке принимаемых данных присутствует “лишний” бит 1*, который нарушает порядок чередования каналов, так что ранее неправильная фаза сигнала CT* автоматически (без её коррекции) становится правильной. Начиная с момента T19’ на выходы устройства поступают правильные данные.

Таким образом, в любых возможных кодовых ситуациях при обнаружении условий формирования корректирующих импульсов J и J* гарантируется восстановление правильной синхронизации, если она была нарушена. Если исходно синхронизация не была нарушена, то импульсы J и J* лишь подтверждают её.

Подводя итоги, можно утверждать, что применение вероятностной синхронизации позволяет повысить скорость передачи данных через систему благодаря двум факторам. Первый фактор состоит в исключении из потока данных относительно большого объёма служебной информации, предназначенной для синхронизации работы дескремблера со скремблером, а также в исключении из протоколов обмена соответствующих программных средств. Второй фактор – исключение из потока данных служебной информации, обозначающей принадлежность данных первому или второму каналу.

Литература

1. *Б.В.Шевкопляс. Скремблирование передаваемых данных — Схемотехника, 2004, №12, с. 24 — 27, 2005, №1, с. 29 — 32, 2005, №2, с. 32 — 35, 2005, №3, с. 30 - 33. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
2. *Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: вставка команд в поток данных без использования избыточных битов – Схемотехника, 2005, № 5, с. 23 – 25, № 6, с. 23 – 26. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
3. *Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: разграничение байтов в битовом потоке данных – Схемотехника, 2005, № 8, с. 23 – 26, № 9, с. 23 – 25. Электронная версия статьи*: *http://lit.lib.ru/s/shewkopljas_b_w/*
4. *С.М.Сухман, А.В.Бернов, Б.В.Шевкопляс. Синхронизация в телекоммуникационных системах. Анализ инженерных решений. – М.: Эко-Трендз, 2003. Электронная версия книги*: *http://lit.lib.ru/s/shewkopljas_b_w/*
5. *Пат. США № 5.822.328 http://www.uspto.gov*
6. *Пат. США № 6.215.835 B1 http://www.uspto.gov*

Б. Шевкопляс