Окончание. Начало — № 12/2004

Скремблирование передаваемых данных

ВЕРОЯТНОСТНАЯ СИНХРОНИЗАЦИЯ СИСТЕМЫ «СКРЕМБЛЕР-ДЕСКРЕМБЛЕР» С ИЗОЛИРОВАННЫМИ ГЕНЕРАТОРАМИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ БИТОВ

Как было показано, система «скремблер-дескремблер» с изолированными от линии связи генераторами псевдослучайных битовых последовательностей (рис. 3) обладает некоторыми преимуществами по сравнению с системой на основе неизолированных генераторов (рис. 2). 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, содержащие 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. 11) 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 ее нарушения.

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, размещенными 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 определенное 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 искомых 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 ее. Если кодовая синхронизация была ранее потеряна, то она восстанавливается при первом же обнаружении одного из заданных кодов в потоке скремблированных данных.

Рассмотрим предлагаемую систему подробнее (рис. 12).

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

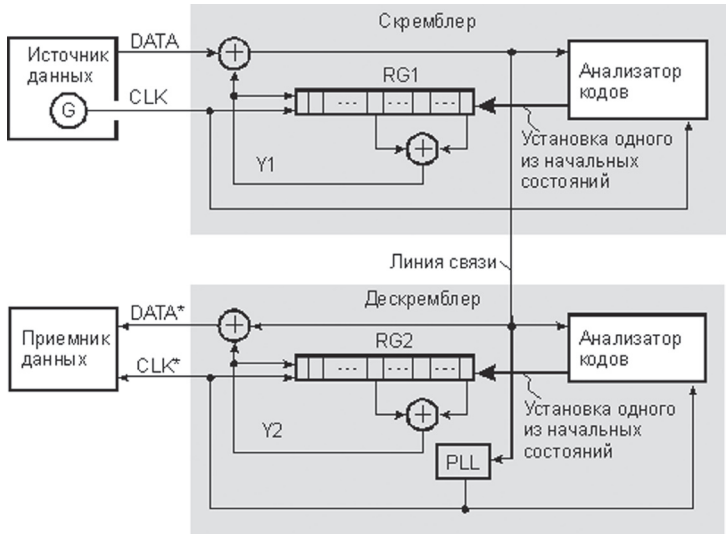

Рис. 11

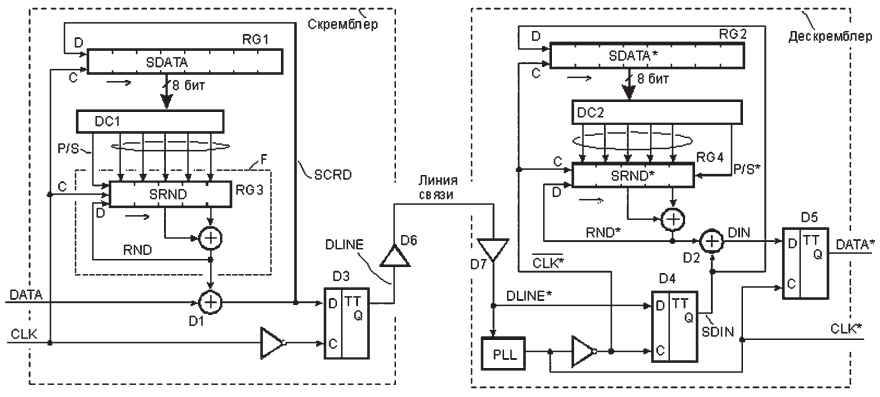

Рис. 12

скремблированных данных. В установившемся режиме эти фрагменты одинаковы (совпадают с точностью до задержки передачи). Прием очередного бита в регистр RG1 (RG2) происходит по фронту сигнала на его синхронизирующем входе C. Одновременно с приемом очередного бита с входа D ранее хранимые данные сдвигаются на один разряд вправо (по стрелке). Выходящие из регистров биты теряются.

Разрядность регистра RG1 (RG2) выбрана равной восьми, хотя она может быть больше или меньше. Динамику работы регистра RG1 можно проследить по таблице его состояний (рис. 13).

SCRD	$SDATA_2$	$SDATA_{16}$
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. 13

Генератор F псевдослучайной последовательности битов скремблера построен на основе сдвигового регистра RG3, генератор дескремблера — на регистре 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 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 приемом 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 фронту сигнала на синхронизирующем входе C в регистр принимается параллельный код с группы входов, обведенной на рис. 12 овалом. В данном примере построения устройства разрядность регистра RG3 (RG4) выбрана равной пяти, хотя она может быть больше или меньше. При этом точки подключения элемента «Исключающее ИЛИ» к регистру RG3 (RG4) выбираются в соответствии с таблицей, представленной на рис. 1, *в*.

В таблице (рис. 14, *а*) представлен список состояний генератора псевдослучайной последовательности битов на основе сдвигового регистра RG3. Диаграмма состояний этого генератора (рис. 14, *б*) отражает перемещение указателя A текущего состояния по кольцевому пути, линии B и C разделяют диаграмму на четыре примерно равных сектора.

а)

STATE	$SRND_2$	RND	$SRND_{16}$
S1	11111	0	1F
S2	01111	0	0F
S3	00111	0	07
S4	00011	1	03
S5	10001	1	11
S6	11000	0	18
S7	01100	1	0C
S8	10110	1	16
S9	11011	1	1B
S10	11101	0	1D
S11	01110	1	0E
S12	10111	0	17
S13	01011	1	0B
S14	10101	0	15
S15	01010	0	0A
S16	00101	0	05
S17	00010	0	02
S18	00001	1	01
S19	10000	0	10
S20	01000	0	08
S21	00100	1	04
S22	10010	0	12
S23	01001	1	09
S24	10100	1	14
S25	11010	0	1A
S26	01101	0	0D
S27	00110	1	06
S28	10011	1	13
S29	11001	1	19
S30	11100	1	1C
S31	11110	1	1E
S1	11111	0	1F

б)

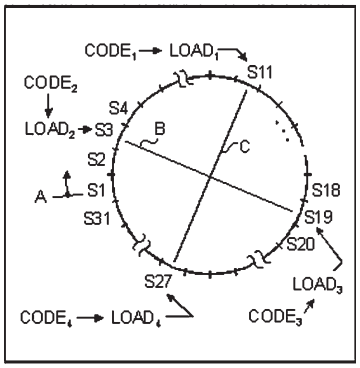

Рис. 14

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 $CODE_1$, который распознается дешифратором DC1. Если в регистре RG3 первоначально присутствовал нулевой код, то код $CODE_1$ 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 $LOAD_1$ 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 $CODE_1$ 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 «Исключающее ИЛИ» D1 и D2 формируют скремблированный SCRD и дескремблированный DIN сигналы данных.

Триггеры D3, D4 и D5 принимают биты данных с входа D по 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) бла-

годаря тому, что прием бита в этот триггер происходит в середине 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 предназначен для формирования высокостабильного синхросигнала CLK* на основе непрерывного слежения за входным сигналом DLINE*. Фронт сигнала CLK* привязан к моментам изменения сигнала DLINE*, спад 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 приему данных по положительному фронту очередного синхроимпульса на входе C.

Усилитель D6 (D7) предназначен для передачи (приема) 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.

Далее приведено описание работы системы.

На входы скремблера поступают данные DATA и сопровождающий их сигнал синхронизации CLK. Фронты сигнала CLK (моменты T0, T1, …, T18 на рис. 15) соответствуют границам между битовыми интервалами сигнала данных DATA. По фронтам сигнала CLK изменяется содержимое регистра RG1 (диаграмма 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 спада сигнала CLK бит SCRD принимается в триггер D3 (диаграмма сигнала DLINE) и через усилитель D6 передается в линию связи.

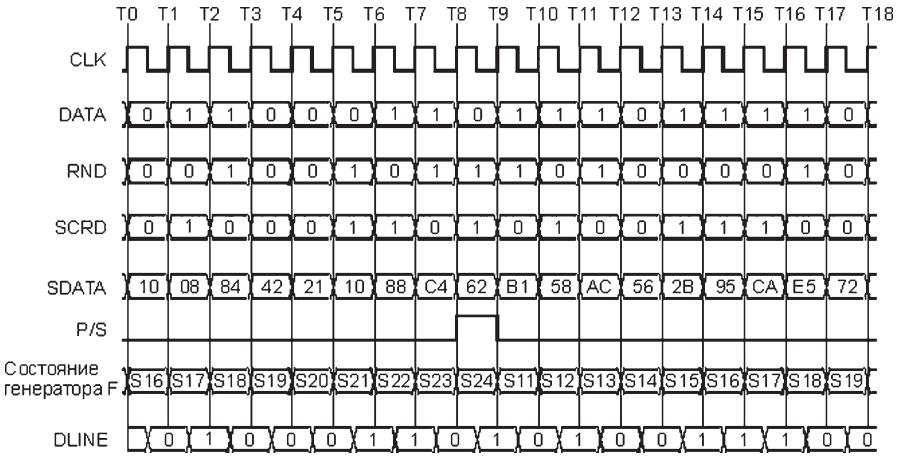

Рис. 15

В интервале времени T8—T9 дешифратор DC1 формирует сигнал лог. 1 на входе P/S управления режимом работы регистра RG3, подготавливая его к приему параллельных данных в момент T9.

В отсутствие параллельной загрузки генератор F псевдослучайной последовательности битов циклически проходит через ряд состояний S1, S2, S3, …, S31, S1, S2 и т. д., как показано в таблице и на диаграмме (рис. 14, *а, б*). В состоянии S1 (первая строка 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 S2, 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 S3, S11, S19 или S27. Эти состояния предпочтительно выбираются так, чтобы на диаграмме (рис. 14, *б*) дуги S3—S11, S11—S19, S19—S27 и S27—S3 имели примерно равную длину (линии B и C, 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 (рис. 1) 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.

Как показано на диаграммах сигналов SDATA и P/S (рис. 15), одним из кодов, вызывающих 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 приему параллельного кода $LOAD_1$. В данном примере этот код равен $0E_{16} = 01110_2$ и соответствует состоянию S11 генератора F (таблица на рис. 14, *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, в результате на его выходе формируется сигнал CLK* (рис. 16). 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 «Исключающее 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. 16) на входе P/S* регистра RG4 формируется сигнал лог. 1, в момент T9 в регистр принимается параллельный код $0E_{16}$, соответствующий состоянию S11.

Независимо от предыстории состояния генератора псевдослучайной последовательности битов дескремблера начиная с момента T9 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. 16) символами X.

Начиная с момента T9 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 $CODE_1$, $CODE_2$, …, $CODE_J$, 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 $CODE_1$, $CODE_2$, …, $CODE_J$ 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.

Борис Шевкопляс,
г. Москва

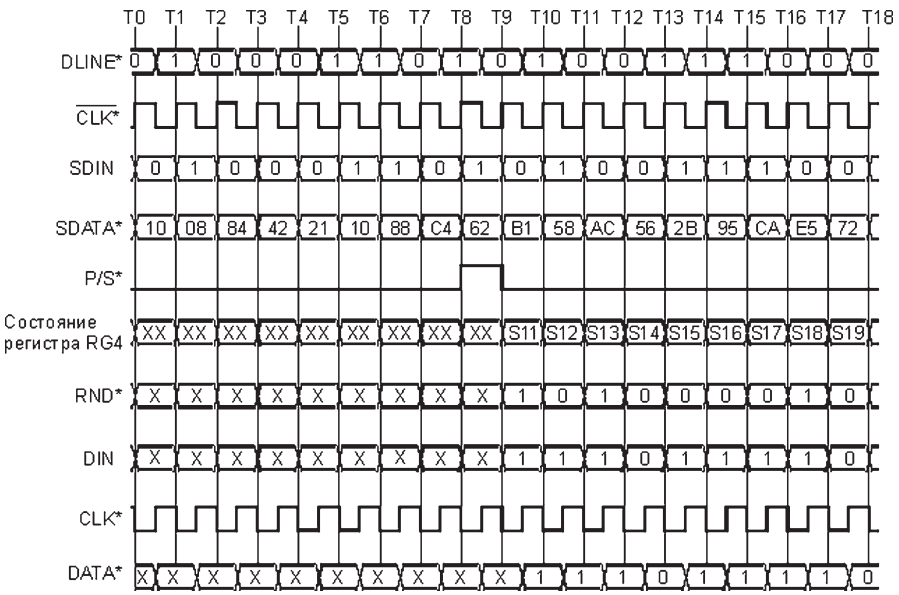

Рис. 16