УДК 519.725

И.А. ЗАВАДСКИЙ

ПОМЕХОУСТОЙЧИВЫЕ КОДЫ ПЕРЕМЕННОЙ ДЛИНЫ НА ОСНОВЕ КОНЕЧНЫХ АВТОМАТОВ

Аннотация. Предложен новый метод помехоустойчивого кодирования, основанный на обработке информационных сообщений конечными автоматами и использовании двухбазисной системы исчисления. Мощные помехоустойчивые свойства обеспечиваются благодаря двухуровневой структуре кодера. На первом, внутреннем, уровне входное сообщение рассматривается как двоичное число, представляемое в двухбазисной системе исчисления в виде нижнего (2,3)-кода, характеризующегося определенной избыточностью и помехоустойчивостью. Затем помехоустойчивые свойства кода усиливаются с помощью внешнего кодирования, выполняемого конечным автоматом. Код имеет переменную длину: для различных входных сообщений одинаковой длины битовая длина генерируемых кодовых слов может различаться. Однако средняя скорость кодера, т.е. отношение битовой длины сообщения на входе к длине кодового слова, составляет 1/2.

Ключевые слова: конечный автомат, помехоустойчивый код, (2,3)-код, сверточный код.

ВВЕДЕНИЕ

В теории помехоустойчивого кодирования конечные автоматы обычно используются для представления диаграмм состояний сверточных кодов. Это автоматы специального вида, удовлетворяющие жестким ограничениям на количество состояний, переходов из каждого состояния и др. Однако на самом деле помехоустойчивые коды можно строить на основе гораздо более широкого класса конечных автоматов, два из которых приведены в данной статье. Общая идея кодирования с помощью конечного автомата состоит в том, что автомат меняет состояния в зависимости от считанных символов кодируемого слова и записывает при этом некоторые символы в результирующий код. Избыточность кода на выходе автомата, а значит, и его потенциальная помехоустойчивость, обеспечивается благодаря тому, что автомат может записывать больше символов, чем считывает, а также иметь «тупиковые» состояния или «ошибочные» переходы, в которых он никогда не окажется при декодировании корректно переданного кодового слова, однако может попасть в случае инвертирования некоторых его битов.

Условия, при выполнении которых автомат осуществляет тот или иной переход, могут быть значительно сложнее, чем проверка соответствия нескольких считанных автоматом символов определенному эталонному значению. В частности, битовая последовательность на входе автомата может интерпретироваться как целое число, арифметические свойства которого определяют последовательность переходов. Такой подход использован на первом, внутреннем, уровне описанной в данной статье двухуровневой системы кодирования: входное сообщение рассматривается как двоичное число и представляется в двухбазисной системе исчисления в виде так называемого (2,3)-кода, обладающего определенной избыточностью и помехоустойчивостью. Затем специальный конечный автомат выполняет внешнее кодирование, усиливающее помехоустойчивые свойства. Высокая помехоустойчивость обеспечивается именно благодаря сочетанию на внешнем и внутреннем уровнях разнородных подходов, состоящих в генерировании символов кода на основе нескольких входных двоичных символов и результата арифметических операций над большим блоком символов входного кода как над целым числом.

Код имеет переменную длину, т.е. для различных входных информационных сообщений одинаковой длины битовая длина генерируемых кодовых слов может различаться. Однако среднее отношение битовой длины сообщения на входе к длине кодового слова составляет 1/2.

ВНУТРЕННЕЕ КОДИРОВАНИЕ

В основе внутреннего кодирования лежит нижний (2,3)-код целых чисел. Впервые (2,3)-кодирование было описано в [1], а нижний (2,3)-код, методы его построения и помехоустойчивые свойства исследованы в [2]. Приведем основные определения. Пусть $\mathbf{N}_{2,3}$ — множество натуральных чисел, взаимно простых с 2 и 3, $x \in \mathbf{N}_{2,3}$, $x > 1$, $n = \lfloor \log_2 x \rfloor$. Тогда x можно представить в форме $2^{n-1} + 3^k x_1$ или $2^{n-2} + 3^k x_1$, где $k \in \mathbf{N}$, $x_1 \in \mathbf{N}_{2,3}$, $x_1 < x$, и только в одной из этих форм. Раскладывая аналогичным образом x_1, получаем x_2 и далее будем вычислять x_{i+1} из уравнения $x_i = 2^{n_i} + 3^{k_i} x_{i+1}$, пока на некоторой итерации не получим $x_i = 1$ или $x_i = 2$. Для однозначного декодирования чисел достаточно запоминать вычисляемые на каждой итерации величины $\Delta_i = \lfloor \log_2 3^{k_i} x_i \rfloor - n_i$ и k_i.

При декодировании последовательность значений x_i восстанавливается в обратном порядке: $x_t, \ldots, x_0$, где $x_t = 1$ или $x_t = 2$. Выбор одного из двух возможных начальных значений x_t определяется тем фактом, что если $x_{t-1} = 7 = 2^0 + 3^1 \cdot 2$, то $x_t = 2$, $b_i = 0$, $k_{t-1} = 1$, $m = \lfloor \log_2 3^k x_1 \rfloor = 2$, $\Delta_{t-1} = m - b = 2$. Легко показать, что $x_{t-1} = 7$ — единственный случай, когда $k_{t-1} = 1$, $\Delta_{t-1} = 2$, и единственный случай, когда $x_t = 2$. Поэтому, если $k_{t-1} = 1$ и $\Delta_{t-1} = 2$, полагаем $x_t = 2$, иначе $x_t = 1$.

Пусть $2^b + 3^k x_1$ — нижнее разложение числа x, $\Delta = \lfloor \log_2 3^k x_1 \rfloor - b$. Покажем, что Δ может принимать только значения 0, 1 или 2, если

$$2^m < 3^k x_1 \le \frac{7}{8} 2^{m+1}, \tag{1}$$

и только значения 0 или 1, если

$$\frac{7}{8} 2^{m+1} < 3^k x_1 < 2^{m+1}. \tag{2}$$

Нижний (2,3)-код числа x — последовательность битовых групп вида $0\ldots01\ldots1$, первая из которых кодирует (2,3)-разложение числа x, а последующие — разложения x_i на дальнейших итерациях. Количество единиц в группе равно k_i, а количество нулей определяется величиной Δ_i. В случае (1) значение $\Delta_i = 0$ кодируется тремя нулями, $\Delta_i = 1$ — двумя, $\Delta_i = 2$ — одним нулем, а в случае (2) $\Delta_i = 0$ кодируется двумя нулями, $\Delta_i = 1$ — одним нулем. Такой способ кодирования выбран в целях сокращения средней длины кода.

Заметим, что уровень помехоустойчивости нижнего (2,3)-кода недостаточен для практического применения без сочетания с другими методами кодирования. Однако помехоустойчивость можно существенно повысить, преобразовав нижний (2,3)-код с помощью изображенного на рис. 1 конечного автомата. Автомат имеет три состояния, обозначенных кружками. Первая цифра в кружке — номер состояния, а под ней в скобках записаны символы, расположенные перед головкой автомата, когда он пребывает в этом состоянии.

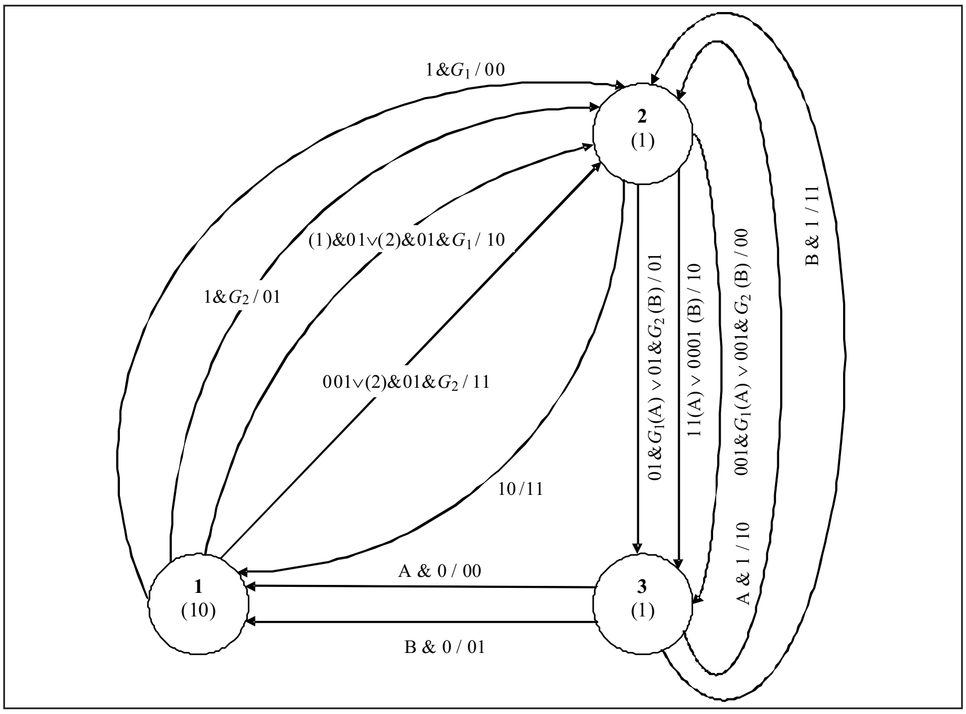

Рис. 1. Схема конечного автомата, генерирующего помехоустойчивый нижний (2,3)-код

Первый символ кода, всегда равный 0, автомат пропускает и начинает кодирование в состоянии 1 со второго символа и со значения $x_t = 1$ или $x_t = 2$, принцип выбора которого описан выше. Автомат переходит в то или иное состояние в соответствии с рис. 1 в зависимости от того, какие символы считывает головка, а также от некоторых других условий; эти символы и условия записаны возле стрелок переходов перед косой чертой. После косой черты указаны символы, которые автомат записывает в код на выходе при каждом переходе. Так как из каждого состояния есть четыре перехода, для идентификации перехода в результирующий код достаточно записать два бита.

Упомянутые выше условия переходов могут быть двух типов. Во-первых, это условия, связанные с тем, что некоторые переходы автомат выполняет в зависимости от значения хеш-функции, аргументом которой является вычисляемая на каждой итерации величина x_i. Такой хеш-функцией может быть, например, $G(x_i) = x_i \bmod 20$. Поскольку x_i — нечетное число, эта функция может принимать десять значений. Если распределить их по двум множествам G_1 и G_2 так, что $G_1 = \{1, 3, 5, 7, 11\}$, $G_2 = \{9, 13, 15, 17, 19\}$, то вероятности попадания значений $G(x_i)$ в каждое из этих множеств будут приблизительно одинаковыми. Символ G_1 или G_2 возле стрелки перехода означает, что переход осуществляется, только если текущее значение $G(x_i)$ принадлежит множеству G_1 или G_2 соответственно (и/или выполняются другие условия перехода). Во-вторых, условием может быть выполнение неравенств (1) или (2) для величины $3^k x_i$. Эти условия на рис. 1 обозначены как (1) и (2).

Рассмотрим, например, переход из состояния 1 в состояние 2, обозначенный как $(1)\&01\vee(2)\&01\& G_1 / 10$. Это означает, что автомат выполнит переход, если: в случае (1) считаны символы 01 или в случае (2) считаны символы 01 и величина $G(x_i)$ принадлежит множеству G_1.

Таким образом, при обработке автоматом нижнего (2,3)-кода должны вычисляться также и значения x_i (или могут использоваться значения, полученные при построении нижнего (2,3)-кода). Каждое следующее такое значение будет вычисляться после обработки автоматом очередной группы символов $0\ldots01\ldots1$, кодирующей пару значений (Δ_i, k_i), — назовем ее (Δ, k)-группой. По причинам, которые понятны из описания метода декодирования, (Δ, k)-группы нижнего (2,3)-кода должны обрабатываться изображенным на рис. 1 автоматом справа налево (однако порядок обработки битов внутри групп изменять не нужно).

Состояние 3 — особое. Каждый переход в это состояние может выполняться при считывании не одной определенной группы символов, а двух различных групп, обозначенных на рис. 1 как А и В. Способ выхода из состояния 3 зависит от того, в соответствии с каким условием был осуществлен переход в это состояние: А или В. Например, если автомат перешел в состояние 3 из состояния 2, считав символы 11 (условие А), а затем головка считала 0, автомат перейдет из состояния 3 в состояние 1 по стрелке А&0 и запишет символы 00.

Когда автомат заканчивает кодировать входящую битовую последовательность, возможны следующие случаи:

1) автомат остановился в состоянии 3, тогда он должен дописать к кодовому слову справа один бит, определяющий, в соответствии с каким условием, А или В, был выполнен переход в это состояние;

2) кодовое слово оканчивается символами 11, головка размещена перед последним символом 1, автомат находится в состоянии 2 и для последнего единичного бита переход из состояния 2 не предусмотрен, тогда к кодовому слову следует дописать справа символы 11, что соответствует символам 10 в исходном коде и переходу в состояние 1; при декодировании последний нулевой бит кодового слова будет проигнорирован, так как оно состоит из $0\ldots01\ldots1$-групп и не может оканчиваться нулем.

Условия переходов между состояниями сформулированы с учетом указанных в [2] вероятностей различных значений Δ так, чтобы все переходы из каждого состояния были как можно более равновероятными. Это позволяет уменьшить длину кода, которая в результате превышает длину двоичного представления числа в 1,25 раз в среднем. Ввиду небольшого увеличения длины при существенном возрастании помехоустойчивости получаемый на выходе автомата код называется сжатым (2,3)-кодом.

При декодировании считываются пары битов из сжатого (2,3)-кода, выполняются соответствующие им переходы и вычисляются значения $x_t, \ldots, x_0$. Если некоторый бит кода вследствие помех в канале связи будет инвертирован, то на одном из переходов, выполняющихся вскоре после обработки этого бита, с большой вероятностью значение хеш-функции $G(x_i)$ может принадлежать множеству G_1, хотя согласно кодировке оно должно принадлежать множеству G_2, либо наоборот. Такая ситуация позволяет выявлять ошибки в коде, что обеспечивает определенную его помехоустойчивость, которая существенно выше помехоустойчивости описанного в [2] нижнего (2,3)-кода.

ВНЕШНЕЕ КОДИРОВАНИЕ

Если рассмотреть подробнее помехоустойчивые свойства сжатого (2,3)-кода, то окажется, что он позволяет с высокой вероятностью обнаруживать факт наличия ошибок в кодовом слове, но дает мало информации о том, в каких именно битах произошли ошибки, а значит, и возможностей для исправления ошибок. Это вполне естественно для высокоскоростного метода кодирования, в котором отношение длины двоичной последовательности к средней длине кодового слова составляет 0,8. Чтобы иметь возможность обнаруживать позиции ошибочных битов, следует модифицировать код, снизив, в частности, его скорость. Например, в изображенном на рис. 1 автомате переходы можно кодировать не двумя, а тремя битами, выбирая тройки битов из множества триад с хемминговым расстоянием 2, такого как $\{000, 011, 101, 110\}$. Скорость кода в этом случае уменьшится в полтора раза. Если в каждой триаде битов имеется не более одной ошибки, то все триады с ошибочными битами можно выявить сразу. Затем с помощью переборного алгоритма определяется, какие именно биты этих триад содержат ошибки. Подобный подход описан в [3]. Его главным недостатком является ограничение, связанное с наличием не более одной ошибки в каждой триаде. Далее опишем конечный автомат, который также обрабатывает сжатый (2,3)-код и уменьшает его скорость в полтора раза, однако лишен этого недостатка.

Автомат считывает пары битов сжатого нижнего (2,3)-кода, осуществляет переходы между состояниями и записывает при этом в результирующий код триады битов. Он имеет 16 состояний, объединенных в четыре группы по четыре состояния в каждой (рис. 2). Триады, записываемые автоматом, принадлежат к одному из множеств: $\mathrm{A} = \{000, 011, 101, 110\}$ или $\mathrm{B} = \{111, 100, 010, 001\}$, в зависимости от того, из какого состояния осуществляется переход. Эти множества будем называть также кодировками А и В. На рис. 2 множество триад, соответствующее переходу, указано возле стрелки перехода (буква А или В). В целом работа автомата на этом рисунке представлена в сокращенном виде, так как на самом деле каждой стрелке перехода соответствуют четыре перехода, ведущих во все состояния той группы, на которую указывает стрелка.

Как видно из рис. 2, номер группы состояний на следующей итерации равен номеру состояния внутри группы на текущей итерации, а считываемые автоматом пары символов сжатого (2,3)-кода определяют, во-первых, какой именно элемент множества А или В будет записан в код на выходе, во-вторых, в какое состояние внутри группы состояний перейдет автомат. Эта взаимосвязь отражена в табл. 1. Номер состояния внутри группы равен значению одной из функций — $(x+1) \bmod 4$ либо $(x+3) \bmod 4$, где x — двухразрядное двоичное число, биты которого считаны автоматом. Если применяется первая из этих функций, то стрелка соответствующего перехода на рис. 2 обозначена цифрой 1 после символа А или В, если вторая — то цифрой 3.

Т а б л и ц а 1. Переходы при внешнем кодировании

Тип кодировки	**Считанные символы**	**Записываемые символы**	**Новое состояние (x — считанное двухбитовое число)**
A1	00 01 10 11	000 011 101 110	$(x+1) \bmod 4$
A3	00 01 10 11	000 011 101 110	$(x+3) \bmod 4$
B1	00 01 10 11	111 100 010 001	$(x+1) \bmod 4$
B3	00 01 10 11	111 100 010 001	$(x+3) \bmod 4$

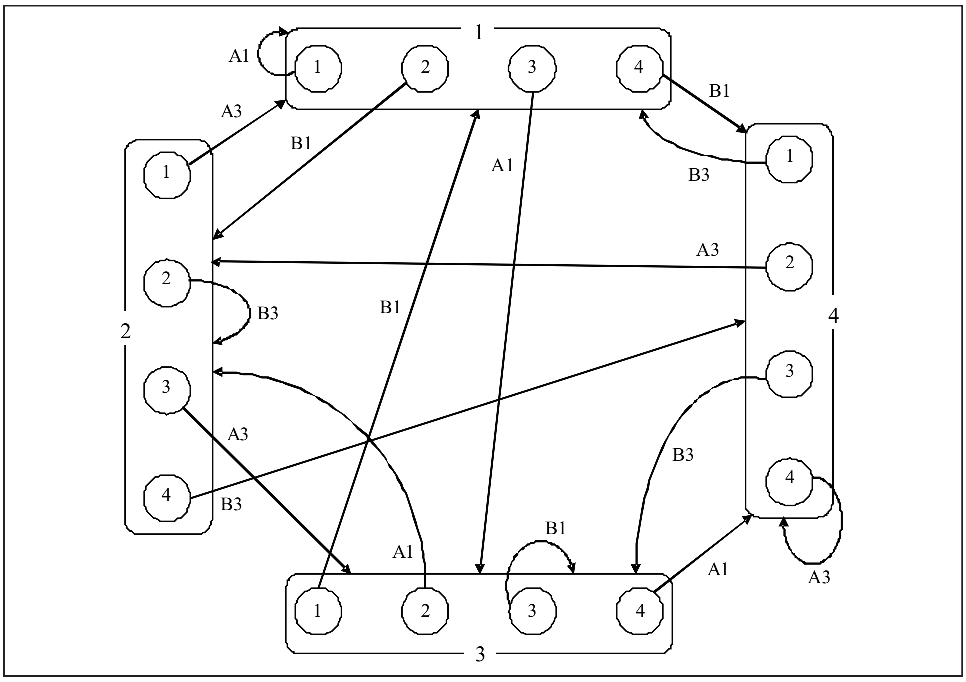

Рис. 2. Общая схема автомата, выполняющего внешнее кодирование

Как кодирование, так и декодирование автомат начинает в первом состоянии первой группы и заканчивает с окончанием обработки подающегося на вход информационного сообщения. Докажем некоторые свойства автомата, существенные с точки зрения исследования его помехоустойчивости.

Заметим, что есть два способа распределения кодировок в группах состояний: в группах 1 и 2 состояниям 1, 2, 3, 4 соответствуют кодировки A, B, A, B, а в группах 3, 4 — кодировки B, A, B, A. Поэтому группы 1 и 2 назовем объединением ABAB, а группы 3 и 4 — объединением BABA. Проанализировав рис. 2, отметим следующее свойство рассматриваемого автомата.

Свойство 1. Если двум различным состояниям одной группы соответствуют одинаковые кодировки, то из одного такого состояния переходы осуществляются в группу объединения BABA, а из другого — в группу объединения ABAB.

Чтобы доказать еще одно важное свойство автомата, введем функцию $S(v, s, g)$, значение которой равно номеру состояния автомата внутри группы на следующей итерации, если на текущей итерации автомат находился в состоянии s группы g и считал двухбитовое число v. Так как всем состояниям каждой группы сопоставлена одна и та же функция перехода состояний вида $(x+c) \bmod 4$, то, если s и q — два различных состояния из одной группы g, выполняется равенство $(S(v, s, g) - S(v, q, g)) \bmod 4 = (s-q) \bmod 4$. Если при этом s и q — состояния с одинаковыми кодировками, то $(s-q) = 2 (\mathrm{mod}\, 4)$ (рис. 2), а значит, и $(S(v, s, g) - S(v, q, g)) = 2 (\mathrm{mod}\, 4)$. Состояниям, номера которых отличаются на 2, в объединениях BABA и ABAB соответствуют различные кодировки, поэтому с учетом свойства 1 получаем следующее свойство.

Свойство 2. Если s и q — различные состояния с одинаковыми кодировками, принадлежащие группам одного объединения, то для любого считанного автоматом значения v переходы из состояний $S(v, s, g)$ и $S(v, q, g)$ имеют различные кодировки.

Если в кодовом слове возникают ошибки, они приводят к сбоям: ситуациям, когда во время декодирования автомат находится в состоянии, которому соответствует множество кодовых слов A, однако считывает кодовое слово из множества B, либо наоборот. Взаимосвязь между местом возникновения ошибки и местом сбоя определяется следующими свойствами.

Свойство 3. Если триада t содержит одну или три ошибки, а при переходе к ней автомат находился в правильном состоянии, сбой произойдет при обработке триады t.

Свойство 4. Если триада t содержит две ошибки, при переходе к ней автомат находился в правильном состоянии и триады $t+1$, $t+2$ ошибок не содержат, то сбой произойдет при обработке триады $t+1$ или $t+2$.

Свойство 3 следует непосредственно из построения множеств A и B, а свойство 4 требует доказательства.

Пусть g_t, g_{t+1}, g_{t+2} и s_t, s_{t+1}, s_{t+2} — группы и внутригрупповые состояния автомата, соответствующие триадам t, $t+1$ и $t+2$ в случае отсутствия ошибок в триаде t, а g'_t, g'_{t+1}, g'_{t+2} и s'_t, s'_{t+1}, s'_{t+2} — соответствующие группы и состояния в случае двух ошибок в триаде t. Две ошибки в одной триаде преобразуют ее в другой элемент того же кодового множества (A или B). Поэтому при двух ошибках в триаде t сбой прямо в ней не возникнет и после ее обработки автомат перейдет в ту же группу, что и в случае отсутствия ошибок в этой триаде, однако в другое ее состояние, т.е. $g_{t+1} = g'_{t+1}$, но $s_{t+1} \neq s'_{t+1}$. Обозначим $E(s)$ функцию, возвращающую тип кодировки (A или B) для переходов из состояния s. Вероятность того, что $E(s'_{t+1}) \neq E(s_{t+1})$, составляет $2/3$, и в этом случае сбой произойдет при обработке триады $t+1$. Если $E(s'_{t+1}) = E(s_{t+1})$, то триада $t+1$ будет обработана без сбоя, но в соответствии со свойством 2 сбой возникнет при обработке триады $t+2$. Свойство 4 доказано.

Из свойств 3 и 4 следует простой метод определения номеров битов с ошибками, если в последовательных трех триадах имеется не более одной ошибки. А именно, если при декодировании некоторой триады t произошел сбой, предполагаем наличие ошибки в каждом бите этой триады и инвертируем этот бит. Если предположение неверно, то вследствие инвертирования создадим в триаде t вторую ошибку и сбой произойдет при обработке триад $t+1$ или $t+2$. Если предположение верно, то исправим ошибку в триаде t и тогда триады $t+1$ или $t+2$ будут обработаны без сбоя. Метод также будет действенным, если в первой из трех последовательных триад все три бита ошибочны, а в двух других триадах ошибок нет.

Чтобы определять ошибочные биты в триадах с двумя ошибками, необходимо рассмотреть еще некоторые свойства кодирующего конечного автомата. Следующее свойство вытекает непосредственно из построения функции, определяющей внутригрупповое состояние на следующей итерации (см. второй и последний столбцы табл. 1).

Свойство 5. Каковы бы ни были различные состояния s и q автомата (т.е. различающиеся либо группой, либо внутригрупповым номером), если автомат считывает в этих состояниях одинаковые символы, то в результате либо в одном из состояний происходит сбой, либо автомат переходит на следующей итерации в состояния с одинаковыми или отличающимися на 2 (mod 4) внутригрупповыми номерами.

Теперь докажем более общее важное свойство.

Свойство 6. Если два экземпляра автомата на некоторой итерации t находятся в различных (т.е. различающихся либо группой, либо внутригрупповым номером) состояниях s_t и s'_t и на итерациях $t - t+3$ считывают одинаковые символы, то на одной из итераций $t - t+3$ в работе одного из автоматов произойдет сбой.

Действительно, если $E(s) \neq E(s')$, то сбой происходит непосредственно в триаде t. Если $E(s) = E(s')$, то в соответствии со свойством 5 $s_{t+1} = s'_{t+1} \pm 2 (\mathrm{mod}\, 4)$ либо $s_{t+1} = s'_{t+1}$. Если g_{t+1} и g'_{t+1} — группы различных объединений, то $E(s_{t+1}) \neq E(s'_{t+1})$ и сбой происходит в триаде $t+1$. Если g_{t+1} и g'_{t+1} — группы одного объединения и $s_{t+1} = s'_{t+1} \pm 2 (\mathrm{mod}\, 4)$, то g_{t+2} и g'_{t+2} будут группами различных объединений и сбой произойдет в триаде $t+2$. Наконец, если g_{t+1} и g'_{t+1} — группы одного объединения и $s_{t+1} = s'_{t+1}$, то $g_{t+1} \neq g'_{t+1}$, $g_{t+2} = g'_{t+2}$ и $c(s_{t+1}) \neq c(s'_{t+1})$, где $c(s)$ — константа в формуле $(x+c) \bmod 4$, по которой вычисляется новое состояние автомата (см. последний столбец табл. 1). Это означает, что s_{t+2} и s'_{t+2} — состояния одной группы, причем $s_{t+2} = s'_{t+2} \pm 2 (\mathrm{mod}\, 4)$. Поэтому в соответствии со свойством 1 g_{t+3}, g'_{t+3} будут группами различных объединений и сбой произойдет в триаде $t+3$. Свойство 6 доказано.

Далее обозначим s и g внутригрупповые состояния и группы, в которых находится автомат в случае отсутствия ошибок, а s' и g' — при их наличии.

Свойство 7. Если в триадах t и $t+1$ содержится хотя бы одна ошибка, при переходе к триаде t автомат находился в правильном состоянии, а в триадах t и $t+1$ не происходит сбоя, то $s_{t+2} \neq s'_{t+2}$ или $g_{t+2} \neq g'_{t+2}$.

Поскольку при переходе к триаде t автомат находился в правильном состоянии, $s_t = s'_t$. Заметим, что ошибка в некоторой триаде p приводит к изменению значения s_{p+1}, но не g_{p+1}. Поэтому если в триаде t содержатся ошибки, то независимо от их наличия в триаде $t+1$ будет выполняться неравенство $s_{t+1} \neq s'_{t+1}$, а значит, $g_{t+2} \neq g'_{t+2}$. Если в триаде t нет ошибок, но они есть в триаде $t+1$, то очевидно, $s_{t+2} \neq s'_{t+2}$. Свойство 7 доказано.

Свойство 8. Если триады t и $t+2$ содержат ошибки, а триада $t+1$ не содержит, при переходе к триаде t автомат находился в правильном состоянии и в триадах t, $t+1$ и $t+2$ не происходит сбоя, то $g_{t+3} \neq g'_{t+3}$.

Так как $s_t = s'_t$, то $g_{t+1} = g'_{t+1}$ и, поскольку оба автомата считывают на итерации $t+1$ одинаковые символы, будет выполняться равенство $s_{t+2} = s'_{t+2}$. Поэтому ошибки в триаде $t+2$ приведут если не к сбою в ней, то к выполнению неравенства $g_{t+3} \neq g'_{t+3}$. Свойство 8 доказано.

Основываясь на свойствах 4, 6–8, опишем метод, который при возникновении сбоя в работе автомата позволяет узнать, не привела ли к сбою двойная ошибка в некоторой триаде, и, если это так, определить два ошибочных бита. Предполагаем, что сбой произошел в триаде t, а триады $t+1-t+4$ ошибок не содержат. Согласно свойству 4 сбой в триаде t может свидетельствовать о наличии двойной ошибки в триаде $t-1$ либо $t-2$. Поэтому следует рассмотреть все шесть возможных комбинаций ошибочных битов, соответствующих одной двойной ошибке в триаде $t-1$ либо $t-2$. Если инвертирование битов в соответствии с одной из комбинаций позволит автомату обработать без сбоев триады t–$t+4$, то это и есть искомая комбинация ошибочных битов, иначе следует сделать вывод, что сбой в триаде t вызван единичной или тройной ошибкой непосредственно в этой триаде.

Чтобы показать достоверность приведенного метода, рассмотрим работу автомата при всех возможных ложных предположениях о том, какие два бита ошибочны.

• Если предполагаем наличие двойной ошибки в той триаде, где она действительно имеет место, но в других битах, то после инвертирования битов, которые посчитали ошибочными, двойная ошибка в этой триаде останется, а в следующих двух триадах согласно свойству 4 произойдет сбой.

• Если предполагаем наличие двойной ошибки в триаде $t-1$, а на самом деле она имеет место в триаде $t-2$ либо наоборот, то после инвертирования битов обе триады, $t-1$ и $t-2$, будут содержать ошибки, а значит, согласно свойству 7 состояние автомата в триаде t будет отличаться от того, в котором бы он пребывал, если бы ошибок не было. Из этого, в соответствии со свойством 6, следует, что при обработке триад t–$t+3$ произойдет сбой.

• Если предполагаем наличие двойной ошибки в триаде $t-1$, а на самом деле сбой вызван единичной или тройной ошибкой в триаде t, то после инвертирования битов ошибки будет содержать как триада $t-1$, так и триада t. Поэтому согласно свойству 7 состояние автомата в триаде $t+1$ будет отличаться от того, в котором бы он пребывал, если бы ошибок не было. Из этого, в соответствии со свойством 6, следует, что при обработке триад $t+1$–$t+4$ произойдет сбой.

• Если предполагаем наличие двойной ошибки в триаде $t-2$, а на самом деле сбой вызван единичной или тройной ошибкой в триаде t, то после инвертирования битов ошибки будут содержать триады $t-2$ и t, но не триада $t-1$. Тогда согласно свойству 8 состояние автомата в триаде $t+1$ будет отличаться от того, в котором бы он пребывал, если бы ошибок не было. Из этого, в соответствии со свойством 6, следует, что при обработке триад $t+1$–$t+4$ произойдет сбой.

ЗАКЛЮЧЕНИЕ

Описанная двухуровневая структура кодирующего устройства предполагает и двухуровневое декодирование, выполняющееся сначала с помощью внешнего автомата, а затем — внутреннего. Для внешнего автомата должен быть применен переборный алгоритм поиска позиций ошибочных битов. Как видно из изложенного выше, такой алгоритм позволит определять позиции ошибочных битов в тех случаях, когда триада битов t содержит один или три ошибочных бита, а в триадах $t+1$ и $t+2$ ошибок нет, либо когда триада битов t содержит два ошибочных бита, а в триадах $t+1$–$t+4$ ошибок нет. При более сложных комбинациях ошибочных битов ложные предположения об их позициях будут отсеиваться с помощью внутреннего автомата. Исследование подобного метода декодирования продолжается.

СПИСОК ЛИТЕРАТУРЫ

1. Anisimov A.V. Prefix encoding by means of the 2,3-representation of numbers // IEEE Trans. Inform. Theory. — 2013. — **59**, N 4. — P. 2359–2374.
2. Анисимов А.В., Завадский И.А. Помехоустойчивое префиксное кодирование на основе нижнего (2,3)-представления чисел // Кибернетика и системный анализ. — 2014. — **50**, № 2. — С. 3–14.
3. Anisimov A.V., Zavadskyi I.O. Forward error correcting codes by means of the two-base (2,3)-numeration system // IEEE Intern. Black Sea Conf. on Communications and Networking. — Chisinau, 2014. — P. 107–111.

Поступила 23.04.2014