КИБЕРНЕТИКА

А.В. АНИСИМОВ, И.А. ЗАВАДСКИЙ

УДК 519.725

ПОМЕХОУСТОЙЧИВОЕ ПРЕФИКСНОЕ КОДИРОВАНИЕ С ПОМОЩЬЮ НИЖНЕГО (2,3)-ПРЕДСТАВЛЕНИЯ ЧИСЕЛ

Аннотация. Предлагается бинарное помехоустойчивое кодирование с помощью представления чисел в двухбазисной системе с основанием 2 и 3. Исследованы свойства (2,3)-кода и возможности корректирования одной или двух ошибок. Для обеспечения помехоустойчивых свойств вводится специальная разновидность (2,3)-кода — нижний (2,3)-код.

Ключевые слова: помехоустойчивое кодирование, префиксные коды, исправление ошибок.

ВВЕДЕНИЕ

Увеличение быстродействия каналов связи, в частности переход к $5G$ мобильным сетям, предъявляет новые требования к помехоустойчивому кодированию передаваемой информации. Важнейшими критериями качества передачи информации являются скорость кодирования/декодирования и ресурсные затраты. Коды, детерминировано исправляющие ошибки (например, код Рида-Соломона), требуют привлечения сложных алгоритмов арифметики многочленов над конечными полями и являются ресурсоемкими, что существенно замедляет процессы передачи информации. Возникает проблема поиска новых высокоскоростных кодов, обладающих повышенной помехоустойчивостью.

В настоящей статье детализируется вопрос о помехоустойчивости префиксных (2,3)-кодов, введенных в [1] и подробно изученных в [2]. Рассматриваются быстрые алгоритмы (2,3)-кодирования/декодирования с исправлением одной ошибки и с высокой вероятностью (свыше 90%) исправления двух ошибок. (2,3)-коды относятся к классу самосинхронизирующихся префиксных кодов переменной длины.

1. МЕТОД (2,3)-КОДИРОВАНИЯ

В работе [1] предложен метод построения кодов переменной длины, основанный на представлении чисел натурального ряда в смешанной двоично-троичной системе. Число 2 служит основным, а 3 — вспомогательным базисом. В работе [2] изучены основные свойства (2,3)-кодов. Такие коды всегда имеют определенную помехоустойчивость, поскольку между кодовыми словами существуют разделители (свойство самосинхронизации). Кроме того, помехоустойчивость поддерживается наличием определенных числовых соотношений между группами бит внутри кодового слова. (2,3)-представление чисел является частным случаем двухбазисных систем представления чисел [3], которые, в свою очередь, обобщают традиционные M-ичные нумерационные системы, основанные на разложении чисел по степеням базиса M.

Все рассматриваемые в настоящей статье числа принадлежат множеству $\mathbb{N}$ натуральных чисел, расширенному нулем. Обозначим $\mathbb{N}_{2,3}$ подмножество натуральных чисел, взаимно простых с числами 2 и 3. Пусть $x \in \mathbb{N}_{2,3}$ — число, большее или равное числу 2; (2,3)-представление числа x образуется применением к x следующей итеративной процедуры.

Предположим, что $x_0 = x$. Выделим из x_0 делители 2 и 3 максимальной степени n_0 и k_0 соответственно, $x_0 = 2^{n_0} 3^{k_0} x_1$, $x_1 \in \mathbb{N}_{2,3}$.

Если $x_1 = 1$, то процесс заканчивается.

Если $x_1 \neq 1$, то $x_1 = 2^{n_1} + 3^{k_1} x_2$, где n_1 — максимальная степень числа 2 такая, что $x_1 \equiv 2^{n_1} \bmod 3$ и $x_2 \in \mathbb{N}_{2,3}$. Применяем к x_2 процедуру, аналогичную применяемой к x_1.

В общем случае на *i*-й стадии итеративной процедуры получаем остаточное число $x_i = 2^{n_i} + 3^{k_i} x_{i+1}$, $i = 1, \ldots, t$.

На предпоследнем шаге имеем $x_t = 2^{n_t} + 3^{k_t}$, тогда $x_{t+1} = 1$.

Из приведенной процедуры следует, что числу $x \in \mathbb{N}_{2,3}$ можно однозначно поставить в соответствие последовательность остаточных чисел

$$x_0, x_1, \ldots, x_t, x_{t+1}, \tag{1}$$

где $x_0 = 2^{n_0} 3^{k_0} x_1$, $x_i = 2^{n_i} + 3^{k_i} x_{i+1}$, $i = 1, \ldots, t$, $x_{t+1} = 1$.

Последовательность (1) позволяет записать x в следующем структурированном виде:

$$x = 2^{n_0} 3^{k_0} (2^{n_1} + 3^{k_1} (2^{n_2} + \ldots + 3^{k_{t-1}} (2^{n_t} + 3^{k_t}) \ldots)). \tag{2}$$

Рассмотрим примеры:

$$20132013 = 3^2 (2^{22} + 3(2^{18} + 3(2^{17} + 3(2^{13} + 3(2^{10} + 3(2^9 + 3(2^7 + 3(2^2 + 3^2)))))))),$$

$$20142014 = 2^2 (2^{22} + 3^2 (2^{19} + 3(2^{15} + 3^4 (2^5 + 3(2^4 + 3(2 + 3)))))),$$

$$20152015 = 2^{24} + 3(2^{19} + 3(2^{16} + 3(2^{14} + 3(2^{13} + 3(2^7 + 3(2^6 + 3(2^2 + 3^2))))))).$$

Таким образом, с числом $x \in \mathbb{N}_{2,3}$ однозначно связывается последовательность пар

$$(n_0, k_0)(n_1, k_1) \ldots (n_t, k_t), \tag{3}$$

где числа n_i и k_i однозначно определяются вышеописанной декомпозиционной процедурой $i = 0, \ldots, t$.

Представление x в виде (1), (2) или (3) называем (2,3)-представлениями числа x.

Из процедуры получения последовательности (1) следует, что выполняются неравенства $k_i \geq 1$, $n_i > n_{i+1}$, $i = 1, \ldots, t$.

Обозначим $\bar{n}_i$ максимальную степень числа 2 в традиционном бинарном представлении числа x_i, $i = 1, \ldots, t$:

$$x_i = 2^{\bar{n}_i} + a_{\bar{n}_i - 1} 2^{\bar{n}_i - 1} + \ldots + a_0, \; a_j \in \{0, 1\}, \; j = 0, \ldots, \bar{n}_i - 1.$$

Заметим, что для любого натурального s числа 2^s и 2^{s+1} дают разные остатки при делении на число 3. Поэтому в (1), (2) и (3) либо $n_i = \bar{n}_i$, либо $n_i = \bar{n}_i - 1$, $i = 1, \ldots, t$. Пары (n_i, k_i), $i = 1, \ldots, t$, назовем блоками. Если $n_i = \bar{n}_i$, то блок называем максимальным; если $n_i = \bar{n}_i - 1$, то — минимальным.

Обозначим $\tau_{\max}(x)$ количество максимальных блоков в (3), а $\tau_{\min}(x)$ — соответствующее количество минимальных блоков, $\tau(x)$ — количество всех блоков в (2,3)-разложении числа x.

В [2] установлены основные неравенства для параметров (2, 3)-представления чисел. В частности, получены следующие оценки:

$$2\tau(x) < \log_2 x - (\log_2 3 - 1)\tau_{\max}(x); \tag{4}$$

$$\tau_{\max}(x) < \frac{\log_2 x}{1 + \log_2 3} \approx 0{,}38 \log_2 x. \tag{5}$$

В среднестатистическом случае

$$E\left[\frac{\tau(x)}{\log_2 x}\right] \le \frac{1}{1{,}5+\log_2 3} \approx 0{,}32\,,$$

где E — математическое ожидание. Отсюда следует, что в большинстве случаев выполняется неравенство

$$\frac{\tau_{\max}(x)}{\log_2 x} \le 0{,}16.$$

Локальные свойства внутри $(2,3)$ -кодовых слов определяются следующими неравенствами:

$$0< n_i - \overline{n}_{i+1} - k_i \log_2 3\,, \tag{6}$$

если i-й блок максимален;

$$-\log_2 3< n_i - \overline{n}_{i+1} - k_i \log_2 3<1\,, \tag{7}$$

если i-й блок минимален.

Предположим, что $x \in \mathbb{N}_{2,3}$. Тогда представление (3) записывается в виде

$$x = x_1 = (n_1, k_1)(n_2 k_2)\ldots(n_t, k_t). \tag{8}$$

Обозначим Δ_i разность $n_i - \overline{n}_{i+1} - k_i$, $i=1,\ldots,t$. Из (8) легко перейти к эквивалентному дельта-представлению

$$x = (\Delta_1, k_1)(\Delta_2, k_2)\ldots(\Delta_t, k_t). \tag{9}$$

Нетрудно проверить, что в (9) выполняются неравенства $\Delta_i \ge 0$, $i=1,\ldots,t$. Обозначим 0^m последовательность из m нулей, аналогично 1^k — последовательность из k единиц, $m>0$, $k>0$. Последовательность вида $0^m 1^k$ назовем 01-группой.

Выполним возможное кодирование слова $C(x)$ для числа x в алфавите $\{0,1\}$ следующим образом. Положим $\Delta_i^1 = \Delta_i + 1$. Основное тело кода числа x задается биективной функцией $C(x) = 0^{\Delta_1} 1^{k_1} \ldots 0^{\Delta_t} 1^{k_t}$.

Из установленных выше неравенств следует, что внутри кодового слова $C(x)$ не может быть кода блока вида 01^k при $k \ge 3$. Поэтому в конце слова $C(x)$ можно приписать последовательность 0111, обозначаемую #, которая служит признаком конца кодового слова:

$(2,3)$-кодом числа x считаем слово $C_{2,3}^{+}(x) = C(x)\#$.

Если кодируются все числа x из множества $\mathbb{N}$, то дополнительно резервируются два префиксных бита, в которых записывается наименьшее число из набора $\{0,1,2,3\}$, которое необходимо добавить к x, чтобы получить число из $\mathbb{N}_{2,3}$. Код $C_{2,3}^{+}$ состоит из кодовых слов $C_{2,3}^{+}(x)\#$, $x \in \mathbb{N}$.

Восстановление числа x по его коду осуществляется в обратном порядке — справа налево: $x_t, x_{t-1}, x_{t-2}, \ldots, x_1, x_0$.

В [2] доказано, что среднестатистическая длина кодового слова $C_{2,3}^{+}(x)$ не превышает $1{,}16 \log_2 x$. Компьютерные эксперименты показывают, что эта величина стремится к $1{,}13 \log_2 x$. Для сравнения отметим, что длина кода Фибоначчи всегда приблизительно равна $1{,}44 \log_2 x$.

Неравенства (4)–(7) дают дополнительную информацию о структуре кодового слова. Они позволяют с некоторой вероятностью в процессе декодирования обнаруживать ошибки, нарушающие эти неравенства.

2. НИЖНЕЕ $(2,3)$-ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

Для получения более сильных двухсторонних неравенств, заменяющих (6) и (7), введем понятие нижнего $(2,3)$-разложения чисел.

Лемма 1. Пусть $x \in \mathbb{N}_{2,3}$, $x > 1$, $n = \lfloor \log_2 x \rfloor$. Тогда x можно представить в одной из следующих форм:

$$x = 2^{n-1} + 3^k x_1, \tag{10}$$

$$x = 2^{n-2} + 3^k x_1. \tag{11}$$

Здесь $k \in \mathbb{N}$, $x_1 \in \mathbb{N}_{2,3} \cup \{2\}$.

Доказательство. Как отмечено в разд. 1, число x имеет единственное $(2, 3)$-представление вида $x = 2^{n_1} + 3^k x_1$, где $n_1 \geq n - 1$ — максимальная степень такая, что $x \equiv 2^{n_1} \pmod 3$, $x_1 \in \mathbb{N}_{2,3}$. Если $n_1 = n - 1$, то получим представление (10). Если $n_1 = n$, то выполним преобразование

$$x = 2^n + 3^k x_1 = 2^{n-2} + (3^k x_1 + 3 \cdot 2^{n-2}) = 2^{n-2} + 3^k x'_1,\ x'_1 \in \mathbb{N}_{2,3},$$

и получим форму, аналогичную (11).

Лемма доказана.

Представление числа x в виде (10) или (11) назовем нижним $(2, 3)$-разложением x.

Базой разложения назовем число 2^{n-1}, если x представимо в виде (10), и число 2^{n-2}, если x представимо в виде (11). Число $3^k x_1$ назовем дополнением $(2, 3)$-разложения числа x.

Предположим, что $x = 2^b + 3^k x_1$, где b — база нижнего $(2, 3)$-разложения. Пару (b, k) назовем блоком. Блок (b, k) является минимальным, если x имеет вид (10), и нижним максимальным, если x имеет вид (11).

Таким образом, число x однозначно раскладывается в последовательность остаточных чисел

$$x, x_1, \ldots, x_t, x_{t+1}, \tag{12}$$

связанных соотношениями $x_i = 2^{b_i} + 3^{k_i} x_{i+1}$, $i = 1, 2, \ldots, t$, предствляющими нижние $(2, 3)$-разложения, $x_{t+1} = 1$ или $x_{t+1} = 2$.

Обозначим $\tau_{\min}(x)$ и $\tau_{\max}(x)$ количество соответственно минимальных и нижних максимальных блоков в (12); $\tau(x)$ — количество всех блоков, $\tau(x) = \tau_{\min}(x) + \tau_{\max}(x)$; $\sigma(x) = \sum_{i=1}^{t} k_i$.

В дальнейшем в целях сокращения текстовых записей будем использовать обозначения: $A \Leftrightarrow B$ означает, что утверждение A эквивалентно утверждению B; $A \Rightarrow B$ означает следование из A утверждения B.

Теорема 1. Пусть $x \in \mathbb{N}_{2,3}$ и последовательность (12) определяет нижнее $(2, 3)$-разложение числа x. Справедливы следующие неравенства:

$$(2 - \log_2 3)\tau_{\min}(x) + (3 - \log_2 7)\tau_{\max}(x) + \sigma(x) \log_2 3 <$$
$$< \log_2 x < \tau_{\min}(x) + (2 - \log_2 3)\tau_{\max}(x) + \sigma(x) \log_2 3. \tag{13}$$

Доказательство. Рассмотрим представление (10). В этом случае выполняются неравенства

$$2^n < x = 2^{n-1} + 3^k x_1 < 2^{n+1}.$$

Для $x - 2^{n-1} = 3^k x_1$ выводим оценки

$$x < 2^{n+1} \Leftrightarrow x - 2^{n-1} < \frac{3}{4} x, \quad 2^n < x \Leftrightarrow \frac{x}{2} < x - 2^{n-1}.$$

Отсюда получаем ограничения на x_1:

$$\frac{x}{2 \cdot 3^k} < x_1 < \frac{3x}{4 \cdot 3^k}. \tag{14}$$

Предположим, что число x задается формой (11). В этом случае используем неравенства

$$x< 2^{n+1} \Leftrightarrow x-2^{n-2} < \frac{7}{8}x, \quad 2^n < x \Leftrightarrow \frac{3}{4}x< x-2^{n-2}.$$

Получаем оценку для x_1:

$$\frac{3x}{4\cdot 3^k}< x_1 < \frac{7x}{8\cdot 3^k}. \tag{15}$$

В логарифмическом виде (14) и (15) переписываются в виде

$$\log_2 x-k\log_2 3-1< \log_2 x_1 < \log_2 x-k\log_2 3+\log_2 3-2, \tag{16}$$

$$\log_2 x-k\log_2 3+\log_2 3-2< \log_2 x_1 < \log_2 x-k\log_2 3+\log_2 7-3. \tag{17}$$

Применяя (16) и соответственно (17) для всех остаточных числел x_i, $i=1,\ldots,t$, в (12) и суммируя все неравенства, после промежуточных сокращений получаем неравенство (13).

Следствие 1. Для числа блоков $\tau(x)$ выполняется неравенство

$$\tau(x)< \frac{\log_2 x}{3+\log_2 3-\log_2 7}\approx 0{,}56\log_2 x.$$

Доказательство. Рассмотрим левую часть неравенства (13).

Учитывая, что $\sigma(x)=\tau(x)+\sum_{i=1}^{t}(k_i-1)$, $k_i\ge 1$, получаем неравенство

$$2\tau_{\min}(x)+(3+\log_2 3-\log_2 7)\tau_{\max}(x)+\sum_{i=1}^{t}(k_i-1)\log_2 3< \log_2 x.$$

Так как $3+\log_2 3-\log_2 7< 2$, то приходим к требуемому неравенству.

Перейдем к неравенствам, которые существенно влияют на бинарную кодировку чисел с помощью нижнего (2, 3)-разложения.

Лемма 2. Пусть $3^k x_1$ — дополнение нижнего (2, 3)-разложения числа $x\in\mathbb{N}_{2,3}$; 2^b — его база и $m=\lfloor\log_2 3^k x_1\rfloor$. Тогда $m-2\le b\le m$. Причем если x представимо в виде (10), то $m-1\le b\le m$, а если x представимо в виде (11), то $m-2\le b\le m-1$.

Доказательство. Если $n=\lfloor\log_2 x\rfloor$, то $x< 2^{n+1}$. Поэтому если x представимо в виде (10), то

$$2^{n-1}+3^k x_1 < 2^{n+1} \Rightarrow 3^k x_1 < 2^{n+1}-2^{n-1} = 4\cdot 2^{n-1}-2^{n-1} = 3\cdot 2^{n-1} = 3\cdot 2^b.$$

Логарифмируя это неравенство, получаем

$$\log_2 3^k x_1 < \log_2 3+b. \tag{18}$$

Отсюда следует, что $\log_2 3^k x_1 < b+1{,}59$, т.е. $m\le b+1$.

Если x представимо в виде (11), то имеем

$$2^{n-2}+3^k x_1 < 2^{n+1} \Rightarrow 3^k x_1 < 2^{n+1}-2^{n-2} = 8\cdot 2^{n-2}-2^{n-1} = 7\cdot 2^{n-2} = 7\cdot 2^b.$$

Логарифмируя это неравенство, получаем

$$\log_2 3^k x_1 < \log_2 7+b. \tag{19}$$

Отсюда имеем $\log_2 3^k x_1 < b+2{,}81 \Rightarrow m\le b+2$.

Таким образом, неравенство $m-2\le b$ доказано.

Если x представимо в виде (10), то с учетом $x>2^n$ получим

$$2^{n-1}+3^k x_1 > 2^n \Rightarrow 3^k x_1 > 2^n-2^{n-1} = 2^{n-1} = 2^b.$$

Прологарифмируем это неравенство:

$$\log_2 3^k x_1 > b \Rightarrow m \geq b.$$

Если x представляется в виде (11), то $2^{n-2} + 3^k x_1 > 2^n$, а значит, $3^k x_1 > 3 \cdot 2^{n-2} = 3 \cdot 2^b$. Прологарифмировав это неравенство, получим

$$\log_2 3^k x_1 > \log_2 3 + b. \tag{20}$$

Отсюда следует неравенство

$$m \geq b + 1. \tag{21}$$

Таким образом, неравенство $m \geq b$ и лемма 2 в целом доказаны.

Положим $\Delta = m - b$. Лемма 2 определяет ограничения для величины Δ в зависимости от варианта нижнего $(2, 3)$ -разложения. Для дальнейшего исследования важно также определить, как величина Δ связана с величиной $\{\log_2 3^k x_1\}$.

Рассмотрим случай, когда число x представимо в форме (10), и перепишем неравенство (18) в виде

$$m + \{\log_2 3^k x_1\} < \{\log_2 3\} + 1 + b,$$

что эквивалентно $\Delta < \{\log_2 3\} - \{\log_2 3^k x_1\} + 1$. Отсюда следует, что если в случае (10) выполняется неравенство $\{\log_2 3^k x_1\} > \{\log_2 3\}$, то $\Delta = 0$.

Представив величину $3^k x_1$ в виде $2^m + d$, где $d < 2^m$, получим

$$\log_2 3^k x_1 = \log_2 (2^m + d) = \log_2 2^m \cdot \frac{2^m + d}{2^m} = m + \log_2 \left(1 + \frac{d}{2^m} \right),$$

и поскольку $0 < \log_2 \left(1 + \frac{d}{2^m} \right) < 1$, имеем $\{\log_2 3^k x_1\} = \log_2 \left(1 + \frac{d}{2^m} \right)$. Поэтому неравенство $\{\log_2 3^k x_1\} > \{\log_2 3\}$ можно представить в виде $\log_2 \left(1 + \frac{d}{2^m} \right) > \log_2 3 - 1$, откуда получим $\frac{d}{2^m} > \frac{1}{2}$.

Если x представимо в виде (11), то из неравенств (19) и (20) следуют соотношения $\log_2 3 + b < m + \{\log_2 3^k x_1\} < \log_2 7 + b$, что эквивалентно $\{\log_2 3\} - \{\log_2 3^k x_1\} + 1 < \Delta < \{\log_2 7\} - \{\log_2 3^k x_1\} + 2$. Отсюда, в свою очередь, следует, что если в случае (11) выполняется неравенство $\{\log_2 3^k x_1\} > \{\log_2 7\}$, то $\Delta = 1$, а если $\{\log_2 3^k x_1\} < \{\log_2 3\}$, то $\Delta = 2$.

Представив $3^k x_1$ в виде $2^m + d$, где $d < 2^m$, и выполнив преобразования, аналогичные описанным выше, легко показать эквивалентность неравенств

$$\{\log_2 3^k x_1\} > \{\log_2 7\} \text{ и } \frac{d}{2^m} > \frac{3}{4},$$

а также

$$\{\log_2 3^k x_1\} < \{\log_2 3\} \text{ и } \frac{d}{2^m} < \frac{1}{2}.$$

Зависимость между величиной Δ, способом разложения x и величиной $\{\log_2 3^k x_1\}$ (т.е. распределением величины $3^k x_1$ на интервале $(2^m; 2^{m+1})$) представим в виде табл. 1.

Особый интерес представляет последняя строка табл. 1, так как при выполнении указанных в ней неравенств допустимы только два значения Δ, тогда как в остальных случаях таких значений может быть три. Этот факт важен для построения кодировки, поэтому сформулируем соответствующую лемму, несколько изменив запись неравенства.

Таблица 1

Величина $3^k x_1$ на интервале $(2^m; 2^{m+1})$	Значения Δ при способе разложения x	
	в виде (10)	в виде (11)
$2^m < 3^k x_1 < \frac{3}{2} 2^m$	$0 \le \Delta \le 1$	$\Delta = 2$
$\frac{3}{2} 2^m < 3^k x_1 < \frac{7}{4} 2^m$	$\Delta = 0$	$1 \le \Delta \le 2$
$\frac{7}{4} 2^m < 3^k x_1 < 2^{m+1}$	$\Delta = 0$	$\Delta = 1$

Лемма 3. Пусть $3^k x_1$ — дополнение нижнего 2,3-разложения числа $x \in \mathbb{N}_{2,3}$, 2^b — его база, $m = \lfloor \log_2 3^k x_1 \rfloor$, $\Delta = m - b$ и $\frac{7}{8} 2^{m+1} < 3^k x_1 < 2^{m+1}$. Тогда $0 \le \Delta \le 1$.

3. ОПРЕДЕЛЕНИЕ ЧАСТОТЫ РАЗЛИЧНЫХ ВАРИАНТОВ НИЖНЕГО (2,3)-РАЗЛОЖЕНИЯ

Пусть имеется нижнее разложение $x = 2^b + 3^k x_1$, $m = \lfloor \log_2 3^k x_1 \rfloor$, $\Delta = m - b$. Если известны величины x_1, k и Δ, то можно однозначно вычислить x. Согласно леммам 2 и 3 величина Δ удовлетворяет таким неравенствам:

$$0 \le \Delta \le 1, \text{ если } \frac{7}{8} 2^{m+1} < 3^k x_1 < 2^{m+1}, \tag{22}$$

$$0 \le \Delta \le 2, \text{ если } 2^m < 3^k x_1 \le \frac{7}{8} 2^{m+1}. \tag{23}$$

Будем считать, что $x \in \mathbb{N}_{2,3}$ — случайная величина, равномерно распределенная на отрезке $[2^n; 2^{n+1}]$. Определим, с какой частотой будем получать различные значения Δ в каждом из случаев (22) и (23). Заметим, что из двух соседних элементов множества $\mathbb{N}_{2,3}$ один всегда представим в виде (10), а другой — в виде (11), поэтому можно считать, что с вероятностью 0,5 число x представляется в виде (10) и с вероятностью 0,5 — в виде (11).

Сначала определим частоту для случая (23). Если x представляется в виде (10), то $b = n-1$, $0 \le \Delta \le 1$, причем

$$\Delta = 0 \Leftrightarrow b = m \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor = n - 1 \Leftrightarrow 2^{n-1} < 3^k x_1 \le \frac{7}{8} 2^n \Leftrightarrow$$

$$\Leftrightarrow 2^n < 2^{n-1} + 3^k x_1 \le \frac{11}{8} 2^n \Leftrightarrow 2^n < x \le \frac{11}{8} 2^n, \tag{24}$$

$$\Delta = 1 \Leftrightarrow b = m - 1 \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor - 1 = n - 1 \Leftrightarrow$$

$$\Leftrightarrow 2^n < 3^k x_1 \le \frac{7}{8} 2^{n+1} \Leftrightarrow \frac{3}{2} 2^n < 2^{n-1} + 3^k x_1 \le \frac{9}{8} 2^{n+1} \Leftrightarrow \frac{3}{2} 2^n < x < 2^{n+1}.$$

В последнем переходе учитывается, что $x < 2^{n+1}$.

Если x представляется в виде (11), то $b = n - 2$, $0 \le \Delta \le 2$, причем

$$\Delta = 1 \Leftrightarrow b = m - 1 \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor - 1 = n - 2 \Leftrightarrow$$

$$\Leftrightarrow 2^{n-1} < 3^k x_1 \le \frac{7}{8} 2^n \Leftrightarrow \frac{3}{2} 2^{n-1} < 2^{n-2} + 3^k x_1 \le \frac{9}{8} 2^n \Leftrightarrow 2^n < x \le \frac{9}{8} 2^n.$$

Здесь в последнем переходе учтено, что $x > 2^n$.

Если

$$\Delta = 2, \text{ то } b = m - 2 \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor - 2 = n - 2 \Leftrightarrow$$

$$\Leftrightarrow 2^n < 3^k x_1 \le \frac{7}{8} 2^{n+1} \Leftrightarrow \frac{5}{4} 2^n < 2^{n-2} + 3^k x_1 < 2^{n+1} \Leftrightarrow \frac{5}{4} 2^n < x < 2^{n+1}.$$

Исходя из этих неравенств, в случае (23) получим

$$P\{\Delta = 0 \mid (23)\} = \frac{1}{2} \cdot \frac{3}{8} : \frac{7}{8} = \frac{3}{14};$$

$$P\{\Delta = 1 \mid (23)\} = \left(\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{8} \right) : \frac{7}{8} = \frac{5}{14};$$

$$P\{\Delta = 2 \mid (23)\} = \frac{1}{2} \cdot \frac{3}{4} : \frac{7}{8} = \frac{3}{7}.$$

Далее рассмотрим случай (22). Если x представляется в виде (10), то $b = n - 1$, а также

$$\Delta = 0 \Leftrightarrow b = m \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor = n - 1 \Leftrightarrow \frac{7}{8} 2^n < 3^k x_1 < 2^n \Leftrightarrow$$

$$\Leftrightarrow \frac{11}{8} 2^n < 2^{n-1} + 3^k x_1 < \frac{3}{2} 2^n \Leftrightarrow \frac{11}{8} 2^n < x < \frac{3}{2} 2^n;$$

$$\Delta = 1 \Leftrightarrow b = m - 1 \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor - 1 =$$

$$= n - 1 \Leftrightarrow \frac{7}{8} 2^{n+1} < 3^k x_1 < 2^{n+1} \Rightarrow \frac{9}{8} 2^{n+1} < 2^{n-1} + 3^k x_1.$$

Таким образом, для $\Delta = 1$ соответствующих значений x в диапазоне $[2^n; 2^{n+1}]$ не существует.

Если x представляется в виде (11), то $b = n - 2$ и согласно (21) $\Delta = 1$, а также

$$b = m - 1 \Leftrightarrow \lfloor \log_2 3^k x_1 \rfloor - 1 = n - 2 \Leftrightarrow \frac{7}{8} 2^n < 3^k x_1 < 2^n \Rightarrow$$

$$\Rightarrow \frac{9}{8} 2^n < 3^k x_1 + 2^{n-2} < \frac{5}{4} 2^n \Rightarrow \frac{9}{8} 2^n < x < \frac{5}{4} 2^n.$$

Таким образом, $P\{\Delta = 0 \mid (22)\} = P\{\Delta = 1 \mid (22)\} = \frac{1}{2} \cdot \frac{1}{8} : \frac{1}{8} = \frac{1}{2}$.

4. МЕТОД НИЖНЕГО (2,3)-КОДИРОВАНИЯ

Рассмотрим нижнее (2, 3)-разложение числа x: $x = 2^b + 3^k x_1$, $m = \lfloor \log_2 3^k x_1 \rfloor$, $\Delta = m - b$, которое будем кодировать в виде $0\ldots01\ldots1$. Количество единиц равно k. Последовательность нулей будет кодировать Δ определенным образом. В результате, зная x, можно определять $3^k x_1$ и наоборот. Исходя из вычисленных ранее частот значений Δ, с целью минимизации длины кода определим следующую кодировку значения Δ.

Если $2^m < 3^k x_1 < \frac{7}{8} 2^{m+1}$, то значения $\Delta = 0$, $\Delta = 1$, $\Delta = 2$ кодируются соответственно символами 000; 00; 0.

Если $\frac{7}{8} 2^{m+1} < 3^k x_1 < 2^{m+1}$, то кодировать два возможных равновероятных значения Δ можно произвольным образом; например, значение $\Delta = 0$ кодируется символами 00, значение $\Delta = 1$ — символом 0.

В такой кодировке не может быть последовательности символов 0000, которую будем использовать как разделитель, обозначающий конец кода числа.

Нижним $(2,3)$-кодом числа $x \in \mathbb{N}_{2,3}$ будем называть последовательность кодировок нижних $(2,3)$-разложений чисел $x_0, x_1, \ldots, x_{t+1}$, дополненную разделителем 0000. Здесь $x_0 = x$, $x_i = 2^{b_i} + 3^{k_i} x_{i+1}$, $i = 1, \ldots, t$, $x_{t+1} = 2$ или $x_{t+1} = 1$.

Для чисел 1 и 2 нижних $(2,3)$-разложений не существует, поэтому кодировать их не будем. Заметим, что если

$$x_t = 7 = 2^0 + 3^1 \cdot 2,$$

то $x_{t+1} = 2$, $b = 0$, $k = 1$, $m = \lfloor \log_2 3^k x_t \rfloor = 2$, $\Delta = m - b = 2$.

Если $x_t = 5 = 2^1 + 3^1 \cdot 1$, то $x_{t+1} = 1$, $b = 1$, $k = 1$, $m = \lfloor \log_2 3^k x_t \rfloor = 1$, $\Delta = m - b = 0$.

Так, число 5 кодируется символами 0001, а число 7 — символами 01. Очевидно, что $x_t = 5$ и $x_t = 7$ — это единственные два случая, когда двумя последними символами перед разделителем будут 01. Также несложно заметить, что $x_{t+1} = 2$ только тогда, когда $x_t = 7$. Таким образом, случаи, когда $x_{t+1} = 2$ и $x_{t+1} = 1$, различимы. Для сокращения длины кода можно заменить кодировку числа 5 на 001 (поскольку при стандартной кодировке окончание кода 001 не имеет места). Коды чисел без разделителя 0000 из множества $\mathbb{N}_{2,3}$ в диапазоне от 11 до 67 приведены в табл. 2.

Чтобы закодировать произвольное натуральное число x (а не только числа из множества $\mathbb{N}_{2,3}$), применим подход, описанный в [1]. В начале каждой кодировки будем записывать двухразрядное двоичное число, которое необходимо прибавить к x, чтобы получить число из множества $\mathbb{N}_{2,3}$. Число 1 можно кодировать символом 1 с добавкой 00. Полные коды первых десяти натуральных чисел без разделителя 0000 приведены в табл. 3.

Таблица 2

Число	Код числа	Число	Код числа
11	011	41	01011
13	0011	43	000111
17	00011	47	010011
19	001001	49	001011
23	01001	53	011001
25	0101	55	0010011
29	00101	59	0100011
31	0111	61	0011001
35	00111	65	0001011
37	000101	67	00100011

Таблица 3

Число	Код числа	Число	Код числа
1	001	6	0101
2	11001	7	0001
3	10001	8	11011
4	01001	9	10011
5	00001	10	01011

Оценим сверху соотношение длины нижнего $(2,3)$-кода и двоичного кода числа. Предположим, что $x = 2^b + 3^k x_1$ удовлетворяет неравенствам (23), причем $\Delta = 0$, $k = 1$, и оценим снизу отношение x / x_1. Если $\Delta = 0$, то $b = n - 1$ и в соответствии с неравенством (24) имеем $2^n < x < \frac{11}{8} 2^n$. Отсюда следует $2^{b+1} < x < \frac{11}{8} 2^{b+1}$. Поскольку $x_1 = (x - 2^b) / 3$, легко заметить, что $x / x_1 > 3 \cdot \frac{11}{8} \cdot 2^{b+1} \Big/ \left(\frac{11}{8} 2^{b+1} - 2^b \right) = \frac{33}{7}$. Код разложения x в случае $\Delta = 0$, $k = 1$ имеет вид 0001, т.е. длину 4 бит, в то время как при двоичном кодировании для уменьшения x в $\frac{33}{7}$ раза потребовалось бы $\log_2 \frac{33}{7}$ бит. Таким образом, в случае выполнения равенств $\Delta = 0$, $k = 1$ для всех блоков длина нижнего $(2,3)$-кода была бы в худшем случае в $4 / \log_2 \frac{33}{7} \approx 1{,}788$ раз больше длины двоичного кода. Для всех других комбинаций значений Δ и k соответствующее соотношение длин будет меньше. Поэтому величина 1,788 — максимально возможное отношение длины нижнего (2, 3)-кода (без учета добавки для приведения числа к $\mathbb{N}_{2,3}$) к длине двоичного кода соответствующего числа.

Оценить среднюю длину нижнего (2, 3)-кода довольно сложно. Заметим лишь, что согласно экспериментальным данным при росте x отношение длины его нижнего (2, 3)-кода к длине двоичного кода стремится к величине 1,167, что несколько больше оригинального (2, 3)-кода [2], для которого аналогичная величина составляет приблизительно 1,133.

5. ПОМЕХОУСТОЙЧИВОСТЬ НИЖНЕГО (2,3)-КОДА

Нижний (2, 3)-код числа является более избыточным, чем, например, γ-код. Покажем, как использовать эту избыточность для повышения вероятности исправления ошибок в кодах. Рассмотрим два простых случая: когда код числа имеет одну ошибку и когда имеет две ошибки. Будем предполагать, что ошибки вносятся не добавлением или удалением битов, а изменением их значений.

Пусть x — некоторое число, представленное в виде нижнего (2, 3)-кода длины n, а x' — тот же код, в который внесены одна или две ошибки. Введем две характеристические функции:

$\operatorname{par}(x) = x_1 \oplus \ldots \oplus x_n$ — четность количества единичных битов в коде;

$\operatorname{pos}(x) = \sum\limits_{i : x_i = 1} i$ — сумма номеров единичных битов в коде (предположим, что биты нумеруются от единицы до n, причем самый старший бит имеет номер 1).

Если ошибок в коде x' не больше двух, то очевидно, что $\operatorname{par}(x') \neq \operatorname{par}(x)$ тогда и только тогда, когда в коде x' есть ровно одна ошибка. Очевидно также, что если ошибка одна, то номер ошибочного бита равен величине $|\operatorname{pos}(x') - \operatorname{pos}(x)|$. Несложно заметить, что если ошибок в коде две, то $\operatorname{pos}(x') \neq \operatorname{pos}(x)$. Изложим первые шаги алгоритма по выявлению и устранению одной или двух ошибок в коде с помощью рассмотренных характеристических функций.

1. Если $\operatorname{pos}(x') = \operatorname{pos}(x)$, то в коде либо нет ошибок, либо более двух ошибок. Иначе переходим к шагу 2.

2. Если $\operatorname{par}(x') \neq \operatorname{par}(x)$, то в коде одна ошибка, причем номер ошибочного бита равен величине $|\operatorname{pos}(x') - \operatorname{pos}(x)|$.

Эти действия позволяют с вероятностью 100% устранить ошибку в коде, если она одна, если в коде две ошибки, то установить факт наличия в точности двух ошибок. Итак, задача свелась к устранению ошибок в x', если их две. Именно для решения этой задачи целесообразно использовать нижнюю (2,3)-кодировку (заметим, что свойства нижней (2,3)-кодировки пока не использовались; указанные выше проверки можно применить к любой кодировке).

Пусть B — множество двоичных строк, $f : B \to B$ — преобразование двоичного числа x в нижний (2, 3)-код $x_{2,3}$, выполняемое в соответствии с описанным в разд. 2 алгоритмом, $f^{-1} : B \to B$ — обратное преобразование, на каждом шаге которого по величинам x_t, k и Δ вычисляется x_{t-1}. Алгоритмы, по которым вычисляются $f(x)$ и $f^{-1}(x_{2,3})$, детерминированы, поэтому любому числу x соответствует только одно значение: $x_{2,3}$, и наоборот. Легко также заметить, что если $x \neq y$ и величины $f(x)$, $f(y)$ вычислимы, то $f(x) \neq f(y)$, т.е. преобразование f является инъективным.

Пусть $U = \{x_{2,3} \mid \exists x : f(x) = x_{2,3}\}$ — множество нижних (2, 3)-кодов, которыми может быть закодировано какое-либо число по описанному выше алгоритму. Пусть $z \in U$, $y \in U$ и $z \neq y$. Тогда предположение о том, что $f^{-1}(z) = f^{-1}(y) = x$, противоречит лемме 1, так как для выполнения неравенства $z \neq y$ на некоторой итерации при вычислении $f(x)$ число x_t должно было бы представляться и в виде (10), и в виде (11). Таким образом, преобразование f^{-1} также является инъективным. Из этого следует, что если $y \notin U$, то величина $f^{-1}(y)$ не может быть вычислена в силу возникновения некоторой ошибки во время вычисления.

Рассмотрим значение $x_{2,3} \in U_n$, а также значение $x'_{2,3}$, образованное из $x_{2,3}$ путем внесения двух ошибок. В силу вышесказанного возможны следующие варианты:

1) $x'_{2,3} \in U_n$;

2) $x'_{2,3} \notin U_n$ и преобразование $f^{-1}(x'_{2,3})$ не может быть выполнено, так как в процессе вычисления этой величины возникает ошибка.

При выполнении преобразования $f^{-1}(x'_{2,3})$ могут возникнуть следующие ошибки:

$$\left.\begin{array}{l} \bullet\ \Delta \text{ представляется в виде последовательности из более чем} \\ \text{трех нулей, если } 2^m < 3^k x_1 < \frac{7}{8}2^{m+1}; \\ \bullet\ \Delta \text{ представляется в виде последовательности из более чем} \\ \text{двух нулей, если } \frac{7}{8}2^{m+1} < 3^k x_1 < 2^{m+1}; \end{array}\right\} \qquad (25)$$

• имеет место неравенство $m - \Delta < 0$.

Вычислительный эксперимент показывает, что при достаточно большой длине 2,3-кода с двумя погрешностями (в пределах 500–2000 бит) одна из трех этих ошибок при попытке вычислить $f^{-1}(x'_{2,3})$ встречается приблизительно в 70% случаев, а приблизительно в 30% случаев $x'_{2,3} \in U_n$.

Опишем метод определения ошибочных битов. Пусть i_1 и i_2 — номера ошибочных битов, т.е. битов, отличающихся в числах $x'_{2,3}$ и $x_{2,3}$; $i_1(x_{2,3})$, $i_1(x'_{2,3})$, $i_2(x_{2,3})$, $i_2(x'_{2,3})$ — значения этих битов в соответствующих числах; $\delta = \text{pos}(x'_{2,3}) - \text{pos}(x_{2,3})$. Заметим, что

$$\delta = (-1)^{i_1(x_{2,3})} i_1 + (-1)^{i_2(x_{2,3})} i_2 = (-1)^{i_1(x'_{2,3})+1} i_1 + (-1)^{i_2(x'_{2,3})+1} i_2. \qquad (26)$$

С помощью этих формул, зная δ и местонахождение одной ошибки, можно вычислить номер другого ошибочного бита.

Рассмотрим вариант 2, когда вычисление $f^{-1}(x'_{2,3})$ приводит к сбою. Этот сбой проявляется при обработке некоторого бита под номером r. Поскольку при вычислении $f^{-1}(x'_{2,3})$ биты числа $x'_{2,3}$ обрабатываются справа налево, то $r \le \max\{i_1(x'_{2,3}), i_2(x'_{2,3})\}$ (напомним, что биты (2, 3)-кода нумеруем слева направо). Логично предположить, что либо величина $i_1(x'_{2,3}) - r$, либо величина $i_2(x'_{2,3}) - r$ будет положительна и невелика, т.е. если не первый, то второй ошибочный бит вскоре приведет к сбою в вычислении $f^{-1}(x'_{2,3})$. Действительно, вычислительный эксперимент показывает, что расстояние от места сбоя в вычислении $f^{-1}(x'_{2,3})$ до ближайшего ошибочного бита, большего по номеру, в среднем составляет около 4 бит. Однако распределение этого расстояния напоминает пуассоновское и иногда оно составляет сотни битов. Более точно: мода такого расстояния равна трем битам. Расстояния в порядке убывания их вероятностей составляют $3, 4, 2, 1, 5, 6, 7, \ldots$ (вероятность расстояния 0 довольно низка и не превышает вероятности расстояния 30). Таким образом, r — хорошая отправная точка для поиска одного из ошибочных битов с последующим определением другого бита по формуле (26). Иначе говоря, если при вычислении $f^{-1}(x'_{2,3})$ в бите r произошел сбой, будем последовательно предполагать, что ошибка возникала в битах $r+3$, $r+4$, $r+2$, $r+1$, $r+5$, $r+6, \ldots$ до тех пор, пока не находилась такая позиция ошибки, для которой выполнялся следующий ниже набор условий.

1. Второй ошибочный бит может быть вычислен по формуле (26). Это возможно не всегда, так как, во-первых, формула (26) определяет не только мес-

тонахождение второго ошибочного бита, но и его значение, которое может не совпадать с реальным значением, и, во-вторых, номер второго ошибочного бита, вычисленный по формуле (26), может превышать n.

2. Выполнено условие 1, а $x''_{2,3}$ представляет собой $x'_{2,3}$ с исправленными двумя битами. Тогда вычисление $f^{-1}(x''_{2,3})$ не должно приводить к сбою.

3. Если выполнены условия 1 и 2, то должно также выполняться условие $\mathrm{deltaBlocks}\,(x_{2,3}) = \mathrm{deltaBlocks}\,(x''_{2,3})$, где

$$\mathrm{deltaBlocks}\;(x) = \left(\sum_{j=1}^{\tau} 3i_j + \Delta_j \right) \mathrm{mod}\, T,$$

τ — общее количество блоков в (2, 3)-коде x; Δ_j — величина Δ на j-й итерации; i_j — номер бита, с которого начинается последовательность нулей, кодирующая Δ_j, $T = 2^{\lfloor (\log n)/2 \rfloor + 1}$.

Вычислительный эксперимент показывает, что при длине 500 бит двоичного кода x в 70% случаев, когда вычисление $f^{-1}(x'_{2,3})$ приводит к сбою, описанный алгоритм позволяет корректно исправить две ошибки приблизительно в 94,2 % случаев. В 30% случаев, когда $f^{-1}(x'_{2,3})$ вычисляется корректно, нельзя получить r — номер бита, где произошел сбой. Поэтому можно предполагать ошибочность любых битов (например, первого, второго, третьего и т.д.), проверяя относительно их три указанных выше условия. Естественно, при таком подходе вероятность корректно исправить две ошибки будет меньше, а именно около 72 %. Заметим также, что при увеличении длины кода вероятность исправления двух ошибок, когда $f^{-1}(x'_{2,3})$ вычисляется корректно, будет уменьшаться, а в случае сбоя при вычислении $f^{-1}(x'_{2,3})$ вероятность исправления двух ошибок будет оставаться почти неизменной.

Для осуществления упомянутых выше проверок помимо числа $x_{2,3}$ необходимо хранить значения трех характеристических функций:

par — один дополнительный бит;

pos — не более $2\lfloor \log n \rfloor - 2$ дополнительных битов, где n — длина $x_{2,3}$ в битах;

deltaBlocks — $t = \lfloor (\log n)/2 \rfloor + 1$ дополнительных битов, так как эта величина вычисляется по модулю 2^t (эксперимент показывает, что старшие биты величины $\sum_{j=1}^{\tau} 3i_j + \Delta_j$ практически не влияют на вероятность корректного исправления ошибок).

Следует также заметить, что поскольку набор символов 0000 сигнализирует о сбое при вычислении $f^{-1}(x'_{2,3})$, он не может быть разделителем при кодировке последовательности чисел. Разделителем целесообразно выбрать последовательность из восьми нулей, так как тогда даже ошибка в центральном единичном бите в наборе 100010001 (т.е. когда $\Delta_j = 0$, $k_j = 1$ и $\Delta_{j+1} = 0$) не приведет к некорректной интерпретации последовательности как разделителя.

Модифицируя способ кодировки, а именно кодируя значение $\Delta = 1$ в случае $\frac{7}{8} 2^{m+1} < 3^k x_1 < 2^{m+1}$ не символом 0, а последовательностью символов 000, можно существенно повысить помехоустойчивость кода. При внесении двух ошибок в такой код вычисление $f^{-1}(x'_{2,3})$ будет приводить к сбою уже приблизительно в 87 % случаев, при которых вероятность исправления двух ошибок возрастет до 95,5 % в среднем, а в 13 % случаев, когда $f^{-1}(x'_{2,3})$ вычисляется без сбоя, вероятность исправления двух ошибок при длине 500 бит двоичного числа x возрастет до 81 % в среднем. Это объясняется тем, что ошибочному для случая $\frac{7}{8}2^{m+1} < 3^k x_1 < 2^{m+1}$ значению $\Delta = 2$ будет соответствовать недопустимый код: одиночный символ 0, в то время как при первоначальном варианте кодировки он интерпретировался как корректное значение $\Delta = 1$. Для выявления этого недопустимого кода необходимо соответствующим образом изменить условия (25). Иными словами, при выполнении преобразования $f^{-1}(x'_{2,3})$ могут возникнуть следующие ошибки:

- Δ представляется в виде последовательности из более чем трех нулей;
- Δ представляется в виде символа 0, если $\frac{7}{8}2^{m+1} < 3^k x_1 < 2^{m+1}$;
- имеет место неравенство $m - \Delta < 0$.

Разумеется, такая модификация обусловит возрастание средней длины кода, а именно отношение длины модифицированного нижнего (2, 3)-кода числа x к длине двоичного кода x в среднем будет стремиться к 1,212.

Значения характеристических функций будем записывать после разделителя, обозначающего окончание основного кодового слова. Чтобы уменьшить вероятность ошибок в характеристических функциях, можно применить к ним описанный выше подход: значения всех таких функций рассматривать как число и кодировать его описанным выше помехоустойчивым нижним (2, 3)-кодом. Основная часть такого кода (пусть ее длина составляет m бит) оканчивается разделителем из восьми нулей, после которого будут записаны характеристические функции этой части. Их длина составит $\lfloor 2{,}5 \log m \rfloor$ бит, и для определения конца такого кода разделитель можно не использовать. Если длина кода числа x и/или плотность помех невелики, то можно и под основные характеристические функции отводить $\lfloor 2{,}5 \log n \rfloor$ бит и не кодировать их нижним (2, 3)-кодом.

Для устранения ошибок в характеристических функциях можно применить известные методы помехоустойчивого кодирования. Они увеличивают длину информационного сообщения в большей степени, чем рассмотренный здесь метод, но так как непосредственно сама длина характеристических функций невелика, это не является существенным недостатком. Так, если длина первого или второго блока характеристических функций не превышает 11 бит, его можно закодировать с помощью (23,11)-кода Галоу, который исправляет три ошибки, хотя и увеличивает длину информационного сообщения более чем вдвое. Если длина первого блока характеристических функций достаточно велика и приближается к 44 бит, для гарантированного исправления в нем двух ошибок можно применить блочный (15,11)-код Рида-Соломона, преобразующий информационное сообщение длиной 44 бит в кодовое слово длиной 60 бит.

Дополнительным фактором при проверке наличия ошибок могут служить неравенства (13), динамически применяемые к остаточным числам в процессе декодирования.

ЗАКЛЮЧЕНИЕ

Сравнение (2, 3)-кодов с известными префиксными кодами, такими как коды Элиаса, Левенштейна и Фибоначчи, приведено в [2], где рассматривался вопрос о соотношении длин кодов. В частности, показано, что (2, 3)-коды асимптотически существенно короче кодов Фибоначчи. Кроме того, сочетание (2, 3)-кодирования с Δ-, γ- или ω-кодами Элиаса [4], а также кодами Левенштейна [5] позволяет получить более сжатые коды по сравнению с бинарными вариантами названных кодировок. Эти же выводы можно применить и к нижнему $(2, 3)$-кодированию, несмотря на то, что нижний $(2, 3)$-код асимптотически несколько длиннее оригинального (2,3)-кода.

Сравним описанный в настоящей статье подход к обеспечению помехоустойчивости нижнего $(2, 3)$-кодирования с другими известными методами кодирования, исходя из возможности обнаружения и исправления одной или двух ошибок. Наиболее известным кодом, исправляющим одну ошибку и позволяющим обнаружить две ошибки, является код Хэмминга. В нем используются m дополнительных битов для кодирования сообщения длиной $2^m - m - 1$ бит, т.е. длина n-битного сообщения возрастает на $\log n$. Нижний $(2, 3)$-код требует несколько большего, хотя и соизмеримого количества дополнительных битов. Однако, в отличие от кода Хэмминга, этот код не требует, чтобы длина основного кодового слова являлась степенью двойки и, что наиболее важно, с достаточно высокой вероятностью позволяет исправлять две ошибки. В соответствии с приведенными выше экспериментальными оценками при длине 500 бит двоичного представления числа x эта вероятность составляет в среднем 87,5 % для базового нижнего (2,3)-кода и около 93,5% для модифицированного, в то время как хорошо известны коды, исправляющие две ошибки с вероятностью единица. Однако увеличение длины информационного сообщения в этих кодах будет существенно выше. Так, большинство эффективных сверточных кодов требует двукратного увеличения длины, а исправляющий две ошибки блочный (15,11)-код Рида-Соломона преобразует информационное сообщение длиной 44 бит в кодовое слово длиной 60 бит, т.е. увеличение в этом случае составляет около 36%, однако кодируемое число следует дополнять нулями до границы, кратной 44 бит.

СПИСОК ЛИТЕРАТУРЫ

1. Анисимов А.В. Представление чисел в смешанном базисе $(2, 3)$ // Кибернетика и системный анализ. — 2009. — № 4. — С. 3–18.
2. 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.
3. Анисимов А.В. Представление чисел в двухбазисных системах счисления // Кибернетика и системный анализ. — 2013. — № 4. — С. 1–14.
4. Elias P. Universal codeword sets and representations of the integers // IEEE Trans. Inform. Theory. — 1976. — **21**, N 2. — P. 194–203.
5. Левенштейн В.И. Избыточность и задержка восстановительного кодирования натуральных чисел // Проблемы кибернетики. — 1968. — № 20. — С. 173–179.

Поступила 13.05.2013