

УДК 621.391.14

## ДВУМЕРНАЯ КОРРЕКЦИЯ СТИРАНИЙ С ИДЕНТИФИКАЦИЕЙ КРАТНОСТИ ОШИБОК В ПРЯМОМ И ИНВЕРСНОМ КАНАЛАХ ДЕКОДИРОВАНИЯ

Н.А. САЛАС, В.К. КОНОПЕЛЬКО, А.И. КОРОЛЕВ

*Белорусский государственный университет информатики и радиоэлектроники  
П. Бровка, 6, Минск, 220013, Беларусь*

*Поступила в редакцию 25 сентября 2013*

Предложен метод коррекции стираний на основе идентификации кратности ошибок в двух каналах (прямом и инверсном) при двумерном кодировании информации. Установлено, что предложенный метод декодирования стираний позволяет полностью использовать корректирующую способность двумерного кода за счет использования идентификации ошибок в прямом и инверсном каналах, определять (фиксировать) канал с меньшим числом ошибок и осуществлять их коррекцию двумерным кодом, исправлявшим ошибки малой кратности.

*Ключевые слова:* ошибки, стирания, коррекция, идентификация, декодер, синдром, двумерный код.

### Введение

В современных телекоммуникационных системах одним из способов обойти так называемую «проблему селектора», которая возникает при коррекции многократных ошибок во время передачи данных по каналам связи, является применение двумерного кодирования. Данное кодирование заключается в организации информационных таблиц и их кодировании двумерным кодом, при котором информация по строкам кодируется кодом  $C_1$  и по столбцам кодом  $C_2$  (рис. 1).

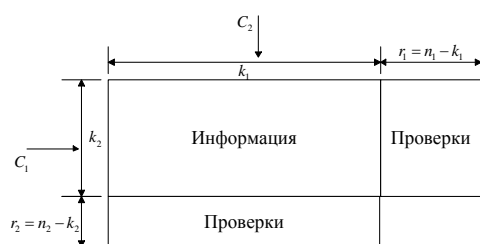


Рис. 1. Структурная схема двумерного кодера

Коррекция ошибок при двумерном кодировании информации происходит, как правило, с использованием двухэтапного метода декодирования, сущность которого состоит в декодировании кода первой ступени  $C_1$  по строкам, а затем кода второй ступени  $C_2$  по столбцам. На первом этапе декодер исправляет небольшое число ошибок малой кратности и обнаруживает неисправляемые комбинации ошибок большой кратности. В случае обнаружения кодом  $C_1$  неисправляемой комбинации ошибок все символы кодовых слов ошибочных строк помечаются как стирания. На втором этапе выполняется исправление стираний кодом  $C_2$  [1]. К недостаткам описанного выше способа коррекции стираний можно отнести: стирание всех символов строк при обнаружении кодом  $C_1$  неисправляемой комбинации ошибок; код  $C_2$  должен выполнить исправление стираний во всех столбцах

таблицы кодирования, которые требуют коррекции. Очевидно, что последнее приводит к большим вычислительным и временным затратам: это происходит за счет повторения процесса коррекции стираний кодом  $C_2$  во всех столбцах таблицы кодирования информации. Однако при использовании данного метода коррекции ошибок некоторые комбинации расположения ошибок невозможно исправить, что приводит к неполному использованию корректирующей способности двумерного способа кодирования [1, 2]. Если использовать коррекцию стираний вместо ошибок, то в большей степени вышеописанные недостатки двумерного декодирования с исправлением ошибок могут быть преодолены.

Известно, что применение метода коррекции стираний позволяет в два раза увеличить корректирующую способность кодов по сравнению с методом исправления ошибок [3]. Таким образом, если использовать коррекцию стираний при двумерном кодировании, то можно использовать коды с меньшим кодовым расстоянием (и, следовательно, с меньшей избыточностью) по сравнению с кодами, исправляющими ошибки той же кратности. При двумерном декодировании информационных таблиц известно местоположение стертых символов. Поэтому на предварительном этапе разработки декодера можно проанализировать всевозможные комбинации размещений стертых символов и сопоставить им некоторые идентификационные параметры, позволяющие реализовать достаточно простые правила их коррекции.

### **Коррекция стираний как ошибок**

В [4] предложен метод коррекции стираний кодами, исправляющими ошибки малой кратности, основанный на использовании двух каналов декодирования (прямого и инверсного). Таким образом, в прямом канале сохраняется полученное из канала двумерное кодовое слово со стираниями, а в инверсном канале – инверсное кодовое слово кроме значений стертых позиций, которые хранятся без изменения. Далее выполняется идентификация канала с меньшим числом несогласованных стираний для его последовательного выбора при коррекции ошибок. При коррекции стираний двумерными кодами, можно использовать аналогичную процедуру. Первоначально принятое из канала связи двумерное слово и его инверсия хранятся в двух запоминающих устройствах (ЗУ), а в дополнительном ЗУ хранится местоположение стертых кодовых символов. При записи значения двумерного слова в прямом и инверсном ЗУ, позиции стирания не инвертируются в инверсном канале, а записываются, как были получены. Таким образом, появляется некая комбинация несогласованных символов в этих ЗУ. Задача состоит в определении двумерного кодового слова для декодирования по некоторым параметрам регистра с меньшим числом несогласованных стираний.

Очевидно, что предложенный выше метод коррекции стираний двумерными кодами имеет большие временные и вычислительные затраты при нахождении всех комбинаций размещения стертых символов в таблице кодирования информации. Однако стертые символы могут быть согласованными или несогласованными. Если значение стертого символа совпадает с его истинным значением при передаче данных, то говорим, что данный символ принят без ошибок и является согласованным; в противном случае, т.е. если значение стертого символа не соответствует правильному значению при передаче данных, данный символ – несогласованный. Поскольку местоположение стираний известно, но неизвестно их истинное значение, то можно воспользоваться библиотекой образов ошибок при двумерном кодировании информации [5].

В данной библиотеке размещаются всевозможные расположения случайных ошибок (образов) кратностью от  $t = 2 \div 6$  в их сокращенной форме записи. В [5] было доказано, что число образов ошибок не зависит от длины кода, благодаря чему происходит уменьшение числа селектируемых комбинаций ошибок при синдромном методе декодирования информации в пределах от  $10^3$  до  $10^{14}$  раз в зависимости от  $t$ . Это обеспечивает большой выигрыш в уменьшении сложности декодера при его реализации.

На рис. 2 приведена библиотека образов случайных ошибок кратности  $t_0 = \{3; 2; 1\}$ , из которой следует, что существуют только соответственно 6, 3 и 1 разных конфигураций трехкратных, двукратных и однократных ошибок в сокращенной библиотеке образов ошибок [5].

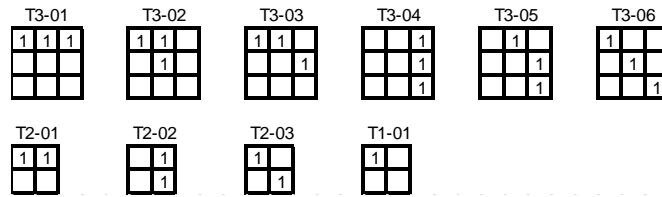


Рис. 2. Образы ошибок кратности  $t_c = \{3; 2; 1\}$

На первом этапе декодер устанавливает образы стираний по информации о местоположении стертых символов путем вычисления идентификационного вектора  $I_c = (I_1, I_2, I_3)$ . Вектор  $I_1$  определяет параметры, являющиеся значением кратности стертых позиций по строкам ( $w_1$ ) и по столбцам ( $w_2$ ), вектор  $I_2$  – количество строк ( $w_3$ ) и столбцов ( $w_4$ ) с нечетным числом стираний, а вектор  $I_3$  – количество строк ( $w_5$ ) и столбцов ( $w_6$ ), в которых имеются одиночные стирания. Например, для образа стирания Т3-02, представленного на рис. 3, вектор  $I_c = (I_1, I_2, I_3)$  равен  $I_c = ((2, 2), (1, 1), (1, 1))$ .

Для всех образов стираний кратности  $t_c = \{3; 2; 1\}$  значения идентификационных векторов  $I_c = (I_1, I_2, I_3)$  различны. Следовательно, можно определить вид стертых символов. Однако, это не дает информации о том, какие стертые символы согласованные. Для этого можно использовать идентификационные параметры двумерного кода, путем вычисления вектора идентификации  $I_o = (I_1, I_2)$  в прямом и инверсном регистрах. Вектор  $I_1$  определяет параметры, являющиеся значением кратности строк ( $w_1$ ) и столбцов ( $w_2$ ), содержащих ошибочные символы, обнаруживаемые кодами С1 и С2 соответственно в прямом кодовом слове; вектор  $I_2$  – количество строк ( $w_3$ ) и столбцов ( $w_4$ ), содержащих ошибочные символы, обнаруживаемые кодами С1 и С2 соответственно в инверсном кодовом слове.

Если все стертые символы несогласованы, т. е. значения стертого символа не совпадают с его истинным состоянием, то образ стираний будет соответствовать одному из образов ошибок рис. 2. В противном случае не все стертые символы – несогласованные, т. е. являются неискаженными. Например, для образа трехкратных ошибок Т3-01 (рис. 2) все возможные комбинации стертых символов приведены на рис. 4, где единичный «1» символ означает несогласованный стертый символ, а нулевой символ «0» – согласованный.

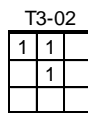


Рис. 3. Образ стираний Т3-02

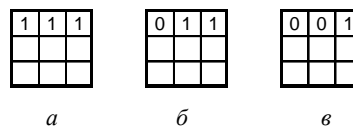


Рис. 4. Образы стираний для образа ошибок Т3-01

Схожим образом необходимо анализировать также все возможные сочетания числа несогласованных стираний кратности меньше, чем  $t_c$  ( $t_c = (2; 1)$ ).

На втором этапе декодирования, зная образ стираний который произошел, необходимо определить число несогласованных стираний как в прямом, так и в инверсном словах, для того чтобы выбрать для декодирования слова с меньшим числом несогласованных стираний. Все идентификационные параметры могут быть найдены на предварительном этапе в виде соответствующих библиотек, а во время декодирования необходимо только вычислить реальные идентификационные параметры по прямому и инверсному словам и сравнить с вычисленными на предварительном этапе идентификационными параметрами, определяя слова с меньшим числом ошибок; данное слово выбирается для декодирования двумерным кодом, исправляющим ошибки. Например, для исправления трехкратных стираний двумерным кодом с параметрами  $d_1 = 2$  и  $d_2 = 2$  имеется всего шесть образов стираний (рис. 2). Рассмотрим структуру размещения несогласованных стираний для каждого образа трехкратных стираний.

На рис. 5 приведены все размещения кратности согласованных и несогласованных стираний для всех образов. Для этих образов в таблице приведены значения идентификационных параметров при кодировании кодами C1 и C2 с  $d_1 = d_2 = 2$ . Значения идентификационных параметров  $I_o = ((I_1), (I_2))$  приведем для прямого и инверсного слов.

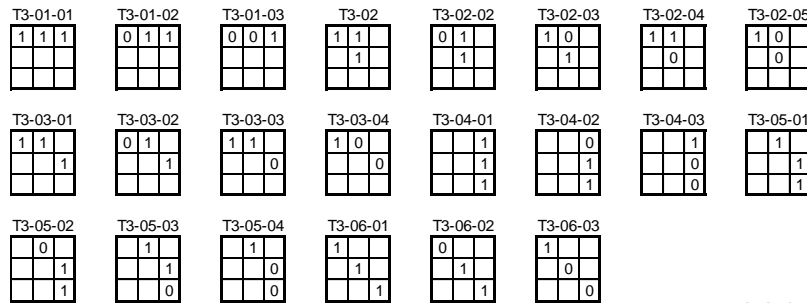


Рис. 5. Размещения кратности согласованных и несогласованных стираний кратности  $t_c = 3$

**Идентификационные параметры образов стираний кратности  $t_c = 3$**

| Образ    | $I_o = ((I_1), (I_2))$ | Образ    | $I_o = ((I_1), (I_2))$ | Образ    | $I_o = ((I_1), (I_2))$ |
|----------|------------------------|----------|------------------------|----------|------------------------|
| T3-01-01 | (1,3),(0,0)            | T3-01-02 | (0,2),(1,1)            | T3-01-03 | (1,1),(0,2)            |
| T3-02-01 | (1,1),(0,0)            | T3-02-02 | (2,0),(1,1)            | T3-02-03 | (2,2),(1,1)            |
| T3-02-04 | (0,2),(1,1)            | T3-02-05 | (1,1),(0,2)            | T3-03-01 | (1,3),(0,0)            |
| T3-03-02 | (2,2),(1,1)            | T3-03-03 | (0,2),(1,1)            | T3-03-04 | (1,1),(2,2)            |
| T3-04-01 | (3,1),(0,0)            | T3-04-02 | (2,0),(1,1)            | T3-04-03 | (1,1),(2,0)            |
| T3-05-01 | (3,1),(0,0)            | T3-05-02 | (2,0),(1,1)            | T3-05-03 | (2,2),(1,1)            |
| T3-05-04 | (1,1),(2,0)            | T3-06-01 | (3,3),(0,0)            | T3-06-02 | (2,2),(1,1)            |
| T3-06-03 | (1,1),(2,2)            | -        | -                      | -        | -                      |

Если все стирания – несогласованные и находятся в прямом регистре, то значение второй части вектора идентификации  $I_o = ((I_1), (I_2))$  равно  $I_o = ((I_1), (0,0))$ , а в противном случае  $I_o = ((0,0), (I_1))$ ; в таких случаях коррекция ошибок не выполняется и для декодирования декодером выбирается слово, где ошибки отсутствуют ( $I_i = (0,0)$ ). Для всех других случаев, т.е. когда не все согласованные стирания находятся в одном из регистров, всегда будет иметься случай, когда одно несогласованное стирание находится либо в прямом, либо в инверсном регистрах. Это можно идентифицировать по значению вектора  $I_o = ((I_1), (I_2))$  (при одном несогласованном стирании  $I_i = (1,1)$ , а в другом слове – ( $I_i \neq (1,1)$ ) и не равно нулю).

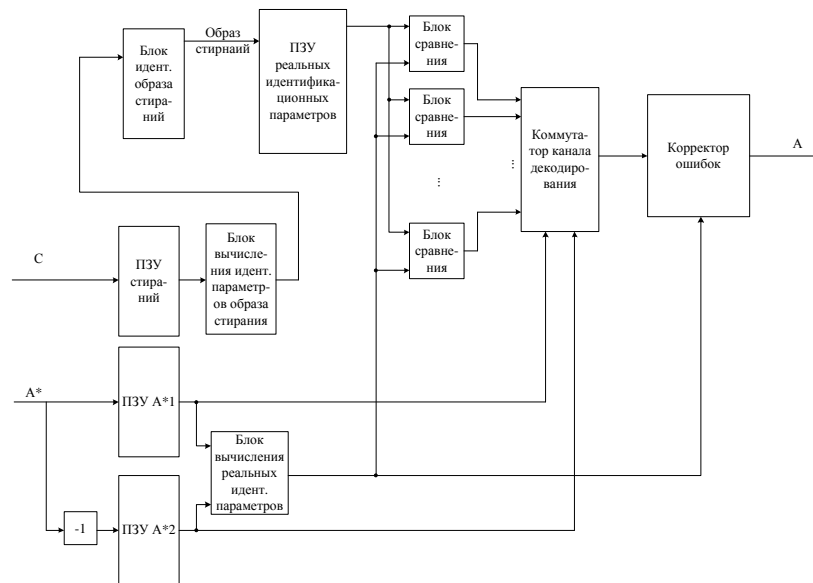


Рис. 6. Структурная схема декодера, корректирующего стирания в двух каналах декодирования при двумерном кодировании

Приведенная на рис. 6 структурная схема декодера, реализующего вышеописанный метод коррекции стираний с использованием двух каналов декодирования и идентификации канала с меньшим числом несогласованных стираний при двумерном кодировании.

При подобном декодировании исправляются трехкратные стирания двумерным кодом с суммарным кодовым расстоянием  $d_{\Sigma} = d_1 \cdot d_2 = 2 \cdot 2 = 4$ , с высоким быстродействием и с минимальной вычислительной сложностью. При этом полностью используется корректирующая способность двумерного кода, по сравнению с известным двухэтапным методом коррекции ошибок при двумерном кодировании, предложенным в [2].

### Заключение

Для коррекции стираний двумерными кодами предложено использовать два канала (прямой и инверсный), с идентификацией образов стираний, формируемых на основе информации о их местоположении. Это позволяет идентифицировать канал с меньшим числом несогласованных стираний путем сравнения вычисленных идентификационных параметров полученного кодового слова и его инверсии с разработанной на предварительном этапе библиотекой образов стираний. Установлено, данный метод декодирования позволяет полностью использовать корректирующую способность двумерного кода по сравнению с известным двухэтапным методом коррекции ошибок при двумерном кодировании.

## TWO-DIMENSIONAL CORRECTION OF ERASURES BY IDENTIFICATION OF MULTIPLICITY OF ERROR IN FORWARD AND INVERSE DECODING CHANNELS

N.A. SALAS, B.K. KONOPELKO, A.E. KOROLEV

### Abstract

A method of correcting erasures based on the identification of the multiplicity of errors in two channels (forward and reverse) decoding in two-dimensional coding is proposed. Found that the proposed method makes full use of two-dimensional coding correction capability, by the use of identification errors in the forward and reverse channels, establishing a channel with a smaller number of errors and their correction by two-dimensional code.

### Список литературы

1. *Фам Хак Хоан, Конопелько В.К., Тиволович А.Д.* // Матер. докл. Междунар. науч.-техн. семинара «Телекоммуникации: сети и технологии, алгебраическое кодирование и безопасность данных». Минск 1 января–31 октября 2007 г. С. 3–8.
2. *Фам Хак Хоан, Конопелько В.К.* // Инж.вест. 2006. № 1 (21). С. 102–105.
3. *Конопелько В.К., Липницкий В.А., Дворников В.Д.* Теория прикладного кодирования. Минск, 2004.
4. *Салас Н.А., Конопелько В.К., Королев А.И.* // Докл. БГУИР. 2013, № 1. С. 19–22.
5. *Конопелько В.К., Смолякова О.Г., Аль-Алем Ахмед Саид.* // Тез. докл. Междунар. науч.-техн. конф., посвящ. 45-летию МРТИ-БГУИР. Минск, 19 марта 2009 го. С. 71–72.