

УДК 621.391.(075.8)

## КОНТРОЛЬ МОДУЛЬНЫХ СТИРАНИЙ КОРРЕКТИРУЮЩИМИ КОДАМИ

ФАМ ХАК ХОАН, В.К. КОНОПЕЛЬКО

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

*Поступила в редакцию 4 октября 2007*

Предложены метод и устройство декодирования для контроля модульных стираний с высоким быстродействием на основе ПЗУ и сумматоров. Приводится анализ аппаратурных и временных затрат на реализацию предложенного декодера.

*Ключевые слова:* контроль, коррекция стирания, модульные ошибки, вектор согласования, коды БЧХ, РС, декодер.

### Введение

На практике в пакетных сетях, в запоминающих устройствах и во многих других системах передачи и хранения информации появляются стертые символы — разряды с известными позициями, но не известными состояниями. Известно, что код с кодовым расстоянием  $d$  позволяет исправить  $\rho$  стираний, для которых  $\rho \leq d-1$ . Наличие дополнительной информации о местонахождении стертых разрядов позволяет в два раза эффективнее использовать корректирующие возможности кодов [1]. Как правило, исправление стираний осуществляется с помощью модифицированного итеративного алгоритма Берлекэмп–Месси для исправления ошибок БЧХ, Рида–Соломона (РС) кодами, приводящее к высокой сложности и низкому быстродействию декодера [2]. Ниже рассматриваются эффективный метод коррекции стираний и быстродействующий декодер для коррекции многократных модульных стираний.

### Метод коррекции многократных модульных стираний путем сравнения вычисленного синдрома с синдромами всевозможных ошибок в стертых модулях

Известно, что для линейных кодов ошибке в  $i$ -й позиции кодового слова соответствует синдром, образованный  $i$ -м столбцом проверочной матрицы, а при появлении многократных ошибок синдром равен сумме соответствующих синдромов, образованных теми столбцами, где произошли ошибки. Это можно использовать для разработки быстродействующего алгоритма исправления стираний. В [3] рассматривается эффективный метод исправления многократных случайных стираний, основанный на использовании вычисленного синдрома  $S$  и сумм составляющих одиночных синдромов  $S_{\Sigma}$  всевозможных стираний. По аналогии можно построить метод и для исправления модульных стираний.

Согласно символам принятого слова  $A_n^*$  вычисляется синдром:

$$S = A_n^* H^T. \quad (1)$$

В соответствии с позициями  $\rho$  стертых модулей длины  $b$  составляется подматрица стираний  $S_e$  размерности  $r \times b$ ,  $\rho$ , которая содержит столбцы проверочной матрицы кода  $H$

с номерами, равными номерам позиций стертых модулей. Обозначим вектор согласования стираний  $\omega_{bp}$  — вектор строки с  $b \cdot p$  элементами,  $i$ -й элемент соответствует  $i$ -й слева стертой позиции и равен нулю, если значение этой позиции у принятого слова совпадает с правильным значением кодового слова и равен единице в противном случае. Составляется система линейных уравнений:

$$\omega_{bp} C_e^T = S_{\Sigma} \quad (2)$$

Сущность метода коррекции модульных стираний состоит в нахождении вектора согласования  $\omega_{bp}$  в стертых модулях на основе сравнения всевозможных сумм синдромов, соответствующих одиночным модульным стираниям с вычисленным синдромом  $S$ . Декодирование осуществляется по следующему правилу:

1. Вычисляется синдром  $S = A_n^* N^T$  (блок вычисления синдрома — БВС).
2. Формируются всевозможные синдромы в каждом модуле (блок ПЗУ).
3. Формируются всевозможные суммы синдромов в стертых модулях.
4. Производится сравнение  $S_{\Sigma} = S$ .
5. По совпадению  $S_{\Sigma}$  с  $S$  формируется вектор согласования стираний  $\omega_{bp}$ .
6. Формируется вектор стираний  $E_n$ .
7. Осуществляется коррекция принятого слова  $A_n = A_n^* + E_n$ .

На рис. 1 приведен алгоритм декодирования двукратных модульных стираний.

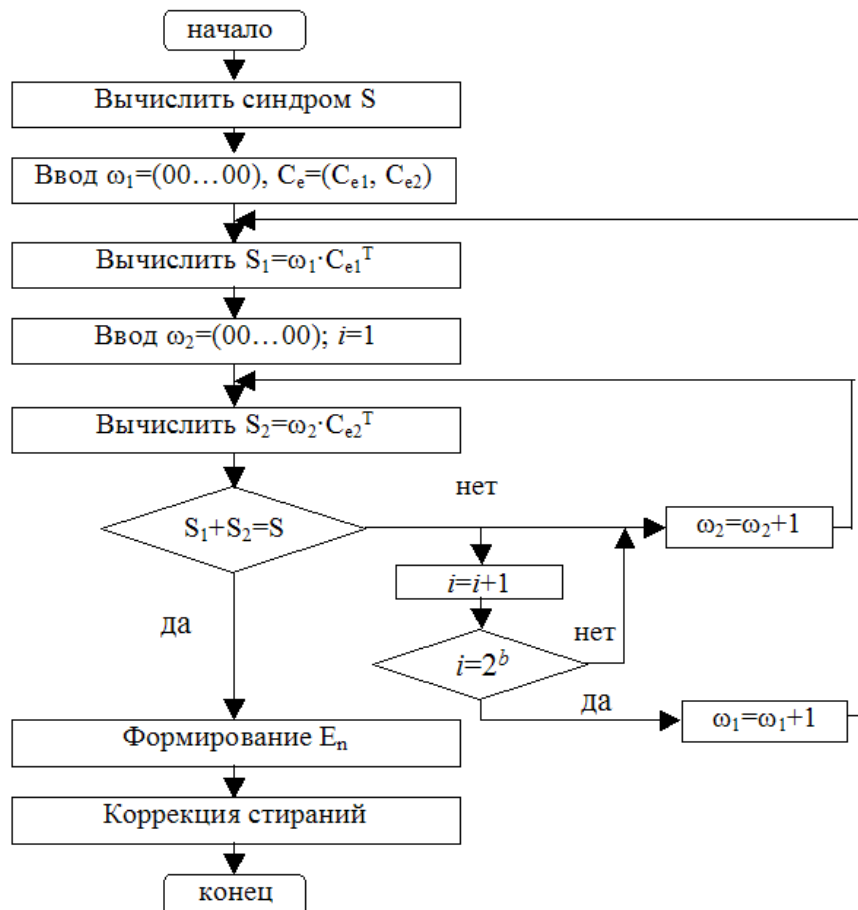


Рис. 1. Алгоритм декодирования двукратных модульных стираний

Рассмотрим в качестве примера исправление модульных стираний кодом Рида–Соломона (27, 21) с  $b=3$ , задаваемым проверочной матрицей:



мых модулей возрастает экспоненциально сложность декодера. Уменьшение сложности можно достичь путем блочного построения декодера.

На рис. 3 приведена функциональная схема блочного декодера для коррекции двукратных модульных стираний, позволяющего уменьшить вычисленные затраты.

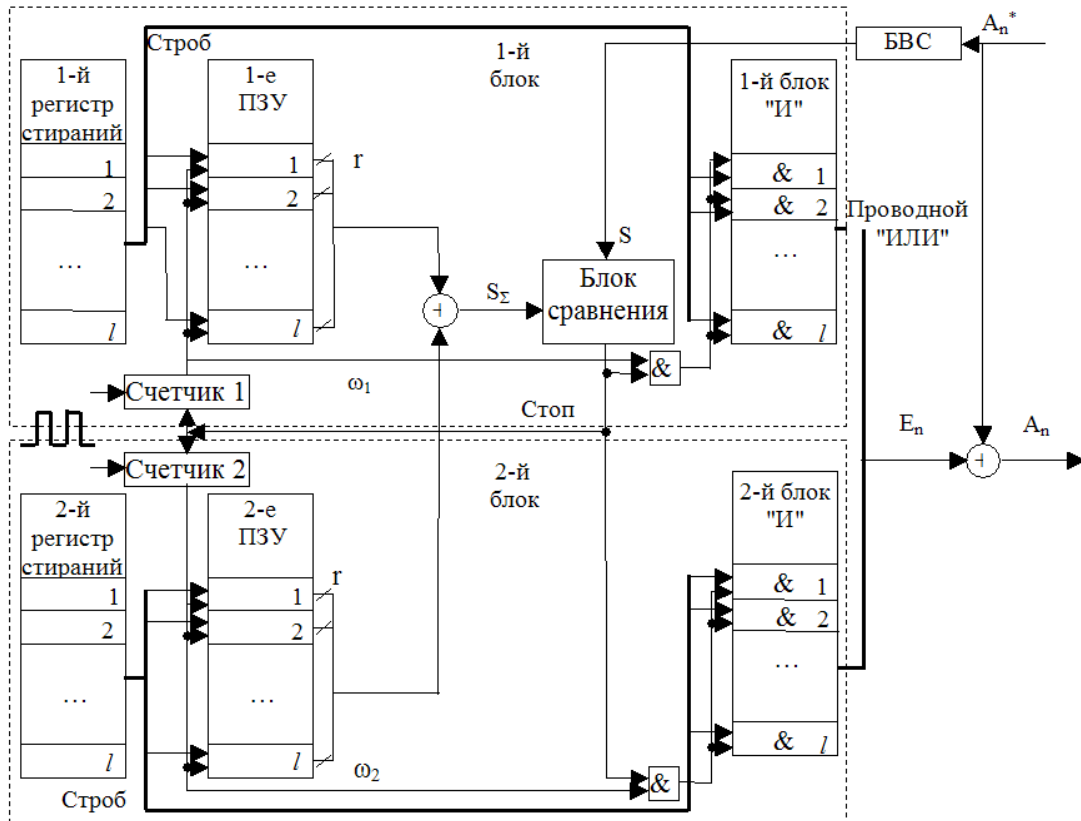


Рис. 3. Блочный декодер для коррекции двукратных модульных стираний

Декодер работает следующим образом. Информация о местоположениях стертых модулей записывается в 1-м и 2-м регистрах адресов стираний. Сигналы из выходов двух регистров стираний поступают на входы соответствующих ПЗУ в качестве стробирующих сигналов. В 1-м, 2-м ПЗУ хранятся всевозможные синдромы одиночных модульных стираний соответственно. В зависимости от сигнала  $\omega_1$  из выхода первого счетчика на одном из выходов 1-го ПЗУ появляется им соответствующий синдром, который поступает на первый вход многоразрядного сумматора по модулю два. По аналогии в зависимости сигнала  $\omega_2$  соответствующий синдром поступает на второй вход сумматора. Сигнал  $S_\Sigma$  из выхода сумматора сравнивается с вычисленным синдромом  $S$ . При совпадении входов блока сравнения на его выходе появляется сигнал лог. "1", который закрывает одновременно первый и второй счетчики. При этом соответствующие сигналы  $\omega_1$  и  $\omega_2$  поступают в 1-й, 2-й блоки элементов "И", на вторые входы которых поступают сигналы из выходов 1-го, 2-го регистров адресов стираний. В результате этого на выходах блоков элементов "И" образуется сигнал вектора стираний  $E_n$ , который скорректирует искаженные разряды в модулях принятого слова  $A_n^*$ .

### Оценка эффективности предложенного метода

Анализ сложности декодера для коррекции двукратных модульных стираний показывает, что основные затраты связаны с реализацией ПЗУ синдромов. В зависимости от длины кода  $n=b \cdot l$  и длины модуля  $b$  аппаратные затраты на реализацию ПЗУ синдромов равны  $Q=2 \cdot l \cdot 2^b \cdot 2^b = 2 \cdot l \cdot 2^{2b} = 4 \cdot n \cdot 2^b$  ячеек памяти.

При росте кратности корректируемых модулей емкость ПЗУ увеличивается. В таблице приведена сложность ПЗУ декодеров для коррекции модульных стираний для различных длин кода  $n$  и длин модуля  $b$ .

**Сложность ПЗУ синдромов при коррекции модульных стираний**

	$n=64, b=4$	$n=2048, b=8$
$\rho=2$	4 кбит	2 Мбит
$\rho=3$	9 кбит	4,5 Мбит
$\rho=4$	16 кбит	8 Мбит

Анализ данных в таблице показывает, что при коррекции модульных стираний одной и той же кратности сложность декодера, определяемая сложностью ПЗУ, растет экспоненциально длине модуля  $b$ , а при одной и той же длине модуля  $b$  — пропорционально квадрату кратности корректируемых модулей  $\rho$ .

Для подбора всевозможных векторов  $\omega=(\omega_1, \omega_2)$  в наихудшем случае требуется  $2^{2b}$  тактов. В рассмотренном примере при использовании кода (27, 21) для коррекции двух модульных стираний потребуется 64 такта (отметим, что при декодировании циклических РС-кодов, число тактов как минимум равно  $2n$ ).

Предложенный метод можно использовать и для коррекции модульных стираний большей кратности. Декодер для коррекции  $\rho$  модульных стираний аналогичен декодеру на рис. 3, но требуется  $\rho$  регистров стираний и  $\rho$  счетчиков для формирования  $\omega_1, \omega_2, \dots, \omega_\rho$ ;  $\rho$  ПЗУ;  $\rho$ -разрядный сумматор для вычисления сумм  $S_\Sigma$ ;  $\rho$  блоков элементов "И". При этом сложность ПЗУ декодера равна:

$$Q=l \cdot 2^b \cdot r \cdot \rho = n \cdot 2^b \cdot \rho^2. \quad (5)$$

Из этого следует, что при одинаковой длине кода  $n$  и длине модуля  $b$  сложность декодера растет примерно пропорционально квадрату кратности стираний. На рис. 4 приведены зависимости сложности декодера для исправления многократных модульных стираний от  $n, b$  и  $\rho$ .

Сравнение сложности коррекции многократных модульных ошибок с коррекцией многократных модульных стираний показывает, что например, РС-код длиной  $n=30, b=5$  и  $d=5$ , позволяет корректировать двукратные модульные ошибки ( $\rho_0=2$ ), необходимости селектирования  $2^r=2^{20}$  различных синдромов ошибок. При использовании норменного метода для коррекции двукратных ошибок реализация селектора декодера требует шесть вентильных матриц емкостью 450 бит ( $0,5 n^2$ ) каждая для вычисления норм синдрома, пять вентильных матриц емкостью 450 бит каждая для совпадения норм и выходной вентильной матрицы емкостью 900 бит, т.е., суммарная сложность декодера примерно равна 6 кбит [4]. Предложенный же декодер для коррекции двукратных модульных стираний (при  $\rho=2$  и  $b=5$ ) требует ПЗУ емкостью  $Q=3,75$  кбит, а при коррекции четырех модульных стираний (при  $\rho=4$  и  $b=5$ ) — ПЗУ емкостью  $Q=15$  Кбит. Это говорит о том, что рассмотренный декодер для коррекции четырехкратных модульных стираний требует аппаратных затрат, соизмеримых с затратами на реализацию декодера для коррекции двукратных модульных ошибок.

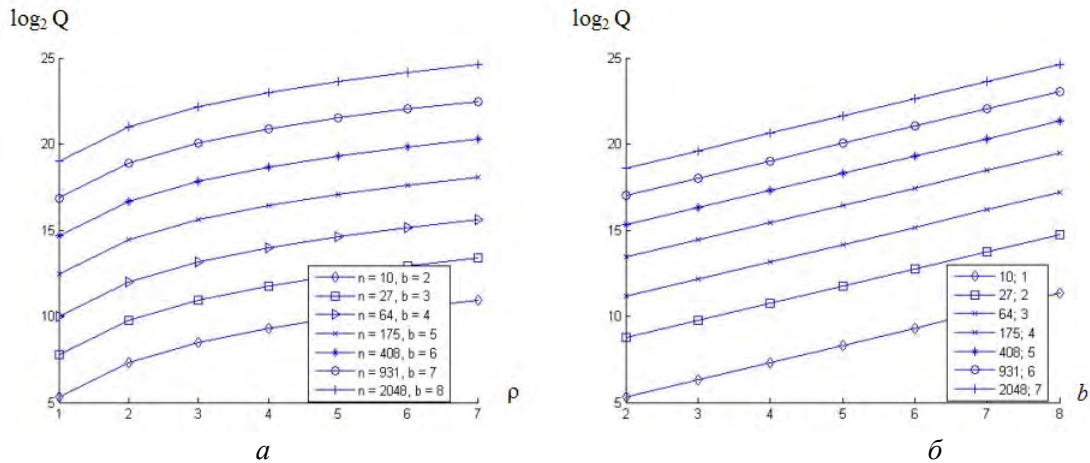


Рис. 4. Зависимости сложности декодера:  $a$  — от кратности стираний  $\rho$  и  $b$  — от длины модуля  $b$

С увеличением длины кода  $n$  сложность декодера для коррекции модульных ошибок растет пропорционально  $n^2$  и резко увеличивается при повышении кратности корректируемых модульных ошибок. Сложность же предложенного декодера растет примерно пропорционально квадрату кратности модульных стираний.

### Заключение

Проведенное исследование показало, что исправление модульных стираний большей кратности  $\rho$  и длины модуля  $b$  требует значительных временных затрат, но не зависит от длины кода, как у циклических РС-кодов. Метод исправления модульных стираний на основании нахождения вектора согласования стираний в модулях позволяет разработать быстродействующий декодер с приемлемой сложностью. При увеличении кратности корректируемых модулей стираний предложенный декодер может наращиваться путем увеличения числа однотипных функциональных блоков.

## CONTROL MODULAR ERASURES BY ERROR-CORRECTION CODES

PHAM KHAC HOAN, V.K. KONOPELKO

### Abstract

The method and decoding device for control modular erasures with high speed processing based on ROM and adder are proposed in this article. The hardware and time costs for realization of proposed decoder are analyzed.

### Литература

1. Теория прикладного кодирования. / Под ред. В.К. Конопелько. Мн., 2004, Т. 2.
2. Блейхут Р. Теория и практика кодов, контролирующих ошибок. М., 1986.
3. Фам Хак Хоан, Конопелько В.К. // Докл. БГУИР. 2006. № 6. С. 19–22.
4. В.К. Конопелько, В.А. Липницкий. Теория норм синдромов и перестановочное декодирование помехоустойчивых кодов. М., 2004.