18. *Rosenko A.P., Bordak I.V.* Metod opredeleniya veroyatnosti nesanktsionirovannogo dos-tupa zloumyshlennika k konfidentsial'noy informatsii // Informatsionnaya bezopasnost' – aktual'naya problema sovremennosti. Sovershenstvovanie obrazovatel'nykh tekhnologiy podgotovki spetsialistov v oblasti informatsionnoy bezopasnosti: sb. trudov VIII–IX Vseros. NTK. g. Gelendzhik 2014 g. [*Informatsionnaya bezopasnost' – aktual'naya problema sovremennosti. Sovershenstvovanie obrazovatel'nykh tekhnologiy podgotovki spetsialistov v oblasti informatsionnoy bezopasnosti: sb. trudov VIII–IX Vseros. NTK. g. Gelendzhik 2014 g.* [Information security is a current problem. Improvement of educational technologies of training specialists in the field of information security: collection of works the VIII–IX All-Russian the scientific and technical conferencing. Gelendzhik 2014]. Krasnodar: FVAS, 2014, 480 p.
19. *Rosenko A.P.* Analiz i obobshchenie sushchestvuyushchikh podkhodov k klassifikatsii ugroz bezopasnosti konfidentsial'noy informatsii [Analysis and synthesis of existing approaches to the classification of threats to the security of confidential information] *Vestnik Severo-Kavkazskogo federal'nogo universiteta* [Vestnik of North-Caucasus Federal University], 2013, No. 3 (36), pp. 30-34.

Статью рекомендовал к опубликованию д.т.н., профессор В.Д. Ковалев.

Росенко Александр Петрович – ФГАОУ ВПО «Северо-Кавказский федеральный университет»; e-mail: rap.44@mail.ru, 355009, г. Ставрополь, ул. Беличенко, 2, кв. 21; тел.: 89197506556, кафедра прикладной математики и компьютерной безопасности; доцент.

Бордак Ирина Владимировна – e-mail: irinabordak@mail.ru; 355005, г. Ставрополь, ул. Гофицкого, 107г; тел.: 89188777404, кафедра прикладной математики и компьютерной безопасности; аспирант.

Rosenko Alexander Petrovich – FGAOU VPO "North-Caucasian Federaltion University"; e-mail: rap.44@mail.ru; 355009, Stavropol, St. Belichenko, 2, sq. 21; phone: +79197506556, the department of applied mathematics and computer security; associate professor.

Bordak Irina Vladimirovna – e-mail: irinabordak@mail.ru, 355005, Stavropol, St. Gorickogo, 107g; phone: +79188777404; the department of applied mathematics and computer security; post-graduate student.

УДК 004.67

П.С. Поперечный

РАЗРАБОТКА ПАРАЛЛЕЛЬНОГО КОДЕРА БЧХ С РЕГУЛИРУЕМОЙ КОРРЕКТИРУЮЩЕЙ СПОСОБНОСТЬЮ

Целью данной статьи является описание способа распараллеливания кодера БЧХ (Боуза–Чоудхури–Хоквингема) с возможностью изменять корректирующую способность кода. Представлена схема традиционного кодера, но с реконфигурируемым порождающим полиномом, хранящимся в перезаписываемых регистрах. Также предложена схема распараллеливания кодера в общем виде для любой ширины шины данных. В результате предложена схема параллельного реконфигурируемого кодера в общем виде для любой шины данных и любого полинома. Выведено аналитическое выражение для параллельной реализации кодера, с переменным количеством исправляемых ошибок. Данное выражение позволяет выполнить как программную, так и аппаратную реализацию кодера. Аппаратно реализовано устройство кодирования предложенным способом. Приведено подробное описание устройства, а также произведено сравнение с аналогичными устройствами с различной шириной входной шины данных с помощью современных САПР на базе ПЛИС. Показано, что за счет своей универсальности в применении для различных накопителей, разработанное устройство имеет преимущество по аппаратным ресурсам. Так же ввиду параллельной обработки входных данных устройство обладает большей пропускной способностью в сравнении с традиционным кодером. Такой подход распараллеливания может применяться при разработке устройств кодирования в системах на кристалле, в контроллерах памяти,

потому как данные поступают параллельно с шины. Этот же подход с небольшими изменениями может быть применим в декодере, в поиске синдромов, ввиду того, что схема поиска синдромов сильно похожа на схему кодера. Несмотря на большую сложность декодера в сравнении с кодером, реализация реконфигурируемого декодера осуществляется проще. Для изменения количества исправляемых ошибок, необходимо просто менять количество тактов работы схемы вычисления полинома локаторов ошибок соответственно.

БЧХ; корректирующая способность; регистр с линейной обратной связью (РЛОС); поле Галуа.

P.S. Poperechny

DEVELOPMENT OF PARALLEL BCH ENCODER WITH ADJUSTABLE ERROR CORRECTION CAPABILITY

This article proposes a method, based on traditional BCH (Bose–Chaudhuri–Hocquenghem) encoder for implementation with parallel input data processing, and with on–demand adjusting correcting level capability. There is original encoder scheme but with adjustable generating polynomial, stored in the rewritable registers. Also the encoder parallelization structure is proposed in-common for any data bus width. As a result, the parallel adjustable encoder scheme is proposed in-common for any data bus and any polynomial. Here is mathematical equation for parallel encoder implementation with variable error-for-correct number. The equation allows to implement both software and hardware encoder. There is hardware implementation of this correction method. There are details of the proposed implementation, and comparison examples with different input databus width by means of state-of-art FPGA CAD. The proposed encoder has hardware resources benefits due to universality for application in any memory devices. Also due to the parallel input data processing the encoder has better data throughput comparing with traditional encoder. So, the same approach of parallelization can be used in decoding, the syndrome calculation scheme, cause this scheme is very similar to the encoder scheme. In spite of much more complexity of decoding algorithm rather than encoding, an adjustable decoding is implemented simpler. To change the error correction level of decoding needs to vary the clock cycles of the error polynomial locator correspondently.

BCH; correcting level capability; linear feedback shift register (LFSR); Galois field.

Введение. Коды БЧХ (Боуза–Чоудхури–Хоквингема) относятся к блочному кодированию и широко используются в системах хранения и передачи информации. Данные коды позволяют исправлять множественные ошибки в блоках данных от нескольких бит до нескольких килобайт (увеличение блока данных приводит к аппаратным сложностям) [1].

В настоящее время данные коды массово используются в таких системах хранения информации как флэш-память [2, 18], твердотельные накопители, в системах связи [16] и стандартах цифрового телевидения [14, 23]. Таким образом, ввиду использования разных накопителей для работы с одним устройством, необходимо применение кодов с разной корректирующей способностью соответственно. Например, для того чтобы в блоке данных используемый код позволял исправлять до 16 ошибок, необходимо применение определенного порождающего полинома определенной длины. Однако, для того чтобы в этом же блоке данных код позволял исправлять например 12 ошибок, необходимо применение другого порождающего полинома с меньшей длиной. То есть, для использования одного и того же устройства с разными накопителями необходимо применение разных порождающих полиномов и, как следствие, разных кодеров, это приводит к увеличению аппаратных ресурсов. Однако, использование кодера с регулируемой корректирующей способностью (переменный порождающий полином) может также удовлетворить различные требования к корректирующей способности [3, 18]. При этом для использования данного кодирования в системах хранения данных чаще всего возникает необходимость кодировать данные поступающие параллельно, то есть с шины данных, что

делает необходимым использовать параллельный кодер [7]. Существуют способы распараллеливания «разворачиванием» аналитического выражения [4], однако отсутствует единый подход для требуемой ширины шины данных [5].

В данной статье предложен способ построения параллельного кодера с возможностью изменять корректирующую способность кода и выведено аналитическое выражения для аппаратной реализации построения подобных устройств. Разработанное устройство к тому же обладает более высоким быстродействием по сравнению с традиционным кодером с последовательной обработкой входного потока данных.

1. Традиционное кодирование БЧХ. Для описания способа распараллеливания для начала необходимо обратиться к традиционному кодированию БЧХ, где систематическое кодирование осуществляется следующим образом [1, 6]:

$$\frac{u(x)\cdot x^{n-k}}{g(x)} = q(x) + \frac{r(x)}{g(x)}, \tag{1}$$

где $u(x)$ – входные незакодированные данные; $g(x)$ – порождающий полином; n – длина кодового слова (длина закодированных данных); k – длина незакодированных данных; $q(x)$ – частное от деления; $r(x)$ – остаток от деления на $g(x)$.

При этом, результирующее кодовое слово (закодированные данные) в систематическом виде представляются как:

$$c(x) = u(x)\cdot x^{n-k} + r(x), \tag{2}$$

где $c(x)$ – кодовое слово.

Таким образом, данные на выходе кодера остаются неизменными, однако к ним добавляется контрольные данные $r(x)$.

Аппаратная реализация выражения (2) выполняется при помощи регистра с линейной обратной связью (РЛОС), представленная на рис. 1.

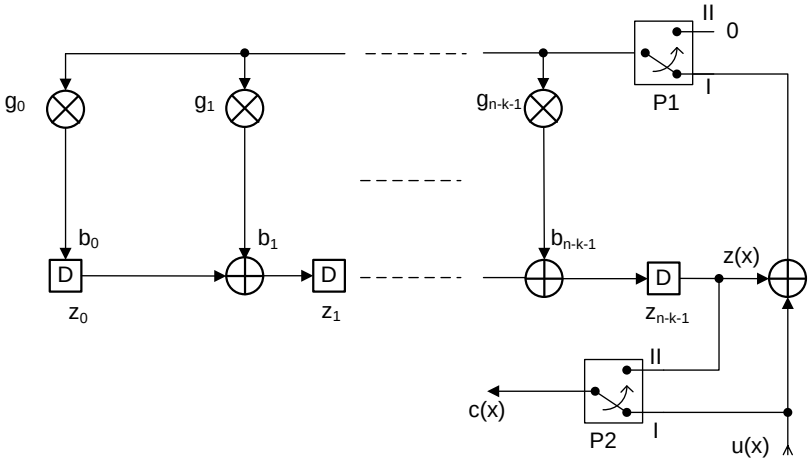

Рис. 1. Традиционная схема кодера БЧХ

2. Кодер БЧХ с регулируемой корректирующей способностью. При необходимости изменения требований к корректирующей способности кода БЧХ, необходимо изменить порождающий полином $g(x)$, что ведет к изменению схемы РЛОС. Предложенные ранее способы изменения корректирующей способности

[3, 13, 22] основаны на разбиении порождающего полинома на множители, что влечет за собой значительное усложнение традиционной схемы, а также добавление многовходового мультиплексора Предлагаемое устройство основано на способе построения схемы РЛОС с возможностью минимальными затратами изменять порождающий полином в процессе работы.

Для реализации выражения (2), но с переменной корректирующей способностью, применяется схема РЛОС с настраиваемыми коэффициентами порождающего полинома и узлами управления, представленными на рис. 2. В подобной структуре, предложенной в патенте [21], полином можно перенастраивать лишь для двух вариантов, что усложняет использование больше двух конфигураций порождающего полинома. В предлагаемой же схеме на рис. 2 порождающий полином можно задавать любым, ограничение стоит лишь в максимальной степени.

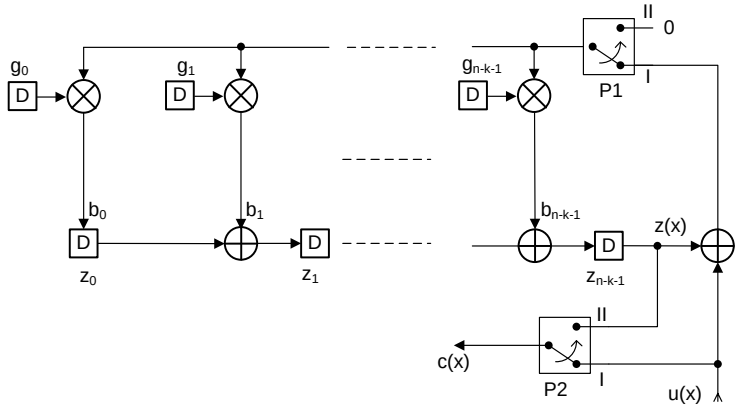

Рис. 2. Кодер БЧХ с регулируемой корректирующей способностью

Первые k тактов работы устройства данные $u(x)$ без изменений проходят на выход схемы, и одновременно поступают на вход РЛОС. Переключатели P1, P2 находятся в положении I. В течение этого этапа работы устройства происходит вычисление остатка $r(x)$. После k тактов работы, переключатели P1, P2 переходят в состояние II, отключая тем самым петлю обратной связи, и подключая РЛОС к выходу схемы. Последующие n-k тактов происходит выгрузка остатка $r(x)$ из сдвигового регистра $z_0, z_1, \ldots . z_{n-k-1}$. Перед началом кодирования очередного блока данных, регистры $z_0, z_1, \ldots . z_{n-k-1}$ сбрасываются в нулевое состояние. Тогда значения $b_0, b_1, \ldots . b_{n-k-1}$, поступающие на вход регистров с выходов умножителей, можно описать с помощью следующего итеративного выражения:

$$\begin{cases} b_j = z_{j-1} + (z_{n-k-1} + u_i) \cdot g_j, \quad j \in [0, n-k-1] \\ \} \ i \in [0, k-1], \end{cases} \tag{3}$$

где j – это позиция соответствующего умножителя g_j или регистра z_j; i – номер текущего такта работы схемы; u_i – значение символа входных данных в текущем такте i.

Далее следующие (n-k) тактов схема РЛОС работает без обратной связи, просто как сдвиговый регистр, поэтому значения сигналов $b_0, b_1, ...b_{n-k-1}$ на входе регистров можно описать следующим образом:

$$\begin{cases} b_j = z_{j-1}, & j \in [0, n-k-1] \\ \} \; i \in [k, n-1]. \end{cases} \quad (4)$$

Все операции выполняются в поле Галуа $GF(p^m)$.

Для двоичных кодов БЧХ, а именно такие коды наиболее распространены в системах хранения и передачи информации, схема на рис. 2 упрощается. А именно:

- умножители вырождаются в элементы «И»;
- сумматоры в элементы «исключающее ИЛИ» (полусумматор);
- регистры, хранящие коэффициенты $g_0, g_1, ..., g_{n-k-1}$ порождающего полинома являются одноразрядными.

Схема кодера двоичных кодов БЧХ представлена на рис. 3.

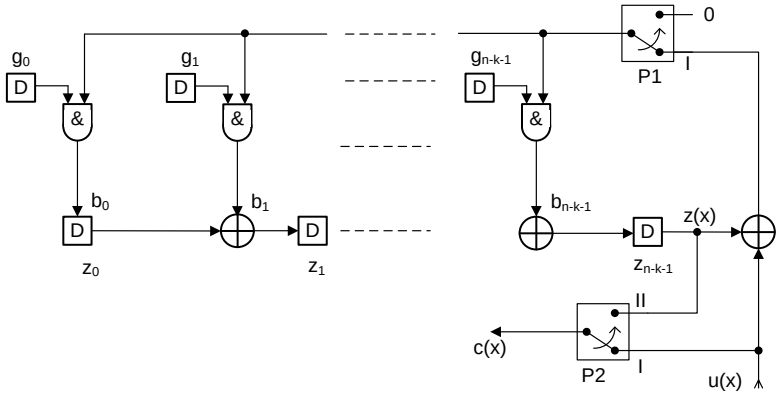

Рис. 3. Кодер двоичных кодов БЧХ с регулируемой корректирующей способностью

Арифметические операции для двоичных кодов БЧХ в выражениях (3, 4), а также в схеме на рис. 3 выполняются в расширенном (m-степень поля) поле Галуа $GF(2^m)$. Так как порождающий полином можно задавать любой степени (от максимального количества значащих коэффициентов $(n-k)_{\max}$ до m), кодер является реконфигурируемым под требуемое количество исправляемых ошибок (от максимального $t_{\max} = \frac{(n-k)_{\max}}{m}$ до 1) для выбранного кода БЧХ. Для изменения требуемой корректирующей способности кода, необходимо перед этапом кодирования перезаписать коэффициенты обновленного полинома. При этом необходимо учитывать, что значащий коэффициент при самой старшей степени полинома должен быть на месте регистра g_{n-k-1}, поэтому если требуемый полином имеет степень меньшую чем заложено в данной реализации $(n-k)_{\max}$, необходимо регистры при младших степенях обнулить. В табл. 1 показан пример, в случае если порождающие полиномы имеют степени 10, и следующей строкой со степенью 8. Коэффициент при старшей степени в кодах БЧХ всегда равен 1, поэтому в таблице он не приведен.

Таблица 1

Пример заданных полиномов со степенями 10 и 8

коэффициенты полинома в соответствующих регистрах																		степень
g[0]	g[1]	g[2]	g[3]	g[4]	g[5]	....	....	g[n-k-10]	g[n-k-9]	g[n-k-8]	g[n-k-7]	g[n-k-6]	g[n-k-5]	g[n-k-4]	g[n-k-3]	g[n-k-2]	g[n-k-1]	(n-k)max
0	0	0	0	0	0	0	0	g[0]	g[1]	g[2]	g[3]	g[4]	g[5]	g[6]	g[7]	g[8]	g[9]	10
0	0	0	0	0	0	0	0	0	0	g[0]	g[1]	g[2]	g[3]	g[4]	g[5]	g[6]	g[7]	8

3. Параллельная реализация кодера БЧХ с регулируемой корректирующей способностью. Существуют способы распараллеливания «разворачиванием» аналитического выражения [4, 10, 11, 12, 13, 19, 20], однако отсутствует единый подход для требуемой ширины шины данных [7, 17]. Для распараллеливания некоторых этапов декодирования БЧХ в источнике [8] схема построена из нескольких стадий, количество которых равно порядку распараллеливания (то есть ширине шины данных).

Подобную структуру можно применить и для кодирования. А именно, каждая стадия соответствует представлению традиционного кодера (см. рис. 1), за исключением того что значения с регистров подменяются значениями от предыдущих стадий. Реализация кодера с параллельным потоком данных представлена на рис. 4.

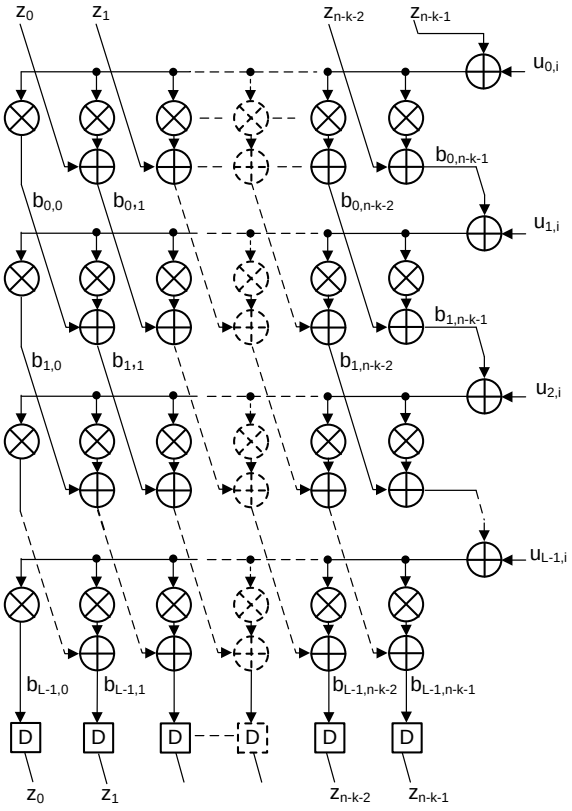

Рис. 4. Параллельный кодер кодов БЧХ

Значения $b_{h,0}, b_{h,1}, \ldots b_{h,n-k-1}$ с выходов умножителей на каждой стадии, а именно на каждой параллельной ступени h потока данных $u_{h,i}$ можно представить следующим образом:

$$
\begin{aligned}
& b_{0,j} = z_j, \quad j \in [0, n-k-1] \\
& \{ \\
& b_{h,j} = b_{h-1,j-1} + (b_{h-1,n-k-1} + u_{h,i}) \cdot g_j, \qquad j \in [0, n-k-1] \\
& \} h \in [0, L-1] \\
& i \in [0, \frac{k}{L} - 1],
\end{aligned}
\tag{5}
$$

где $u_{h,i}$ – значение h-го символа слова данных поступивших в i-й такт.

Выражение (5) получается из выражения (4) с заменой значений с регистров на значения предыдущих стадий, как было сказано выше. Каждый такт на вход схемы подается L бит данных, и каждому из этих бит соответствует своя стадия от 0 до L-1.

Количество тактов работы схемы уменьшилось в L (размерность шины данных) раз с k до k/L, так как данные поступают параллельно. К моменту k/L такта данные заканчиваются, и вычисленные контрольные символы хранятся в регистрах $z_0, z_1, \ldots z_{n-k-1}$.

Данная схема значительно упрощается в случае кодирования двоичных кодов БЧХ. Для двоичных кодов БЧХ умножители в поле Галуа $GF(p^m)$ в схеме рис. 2 заменяются простым элементом «И», сумматоры выполняют сложение по модулю 2, так как работают в поле Галуа $GF(2^m)$ как показано в схеме на рис. 5.

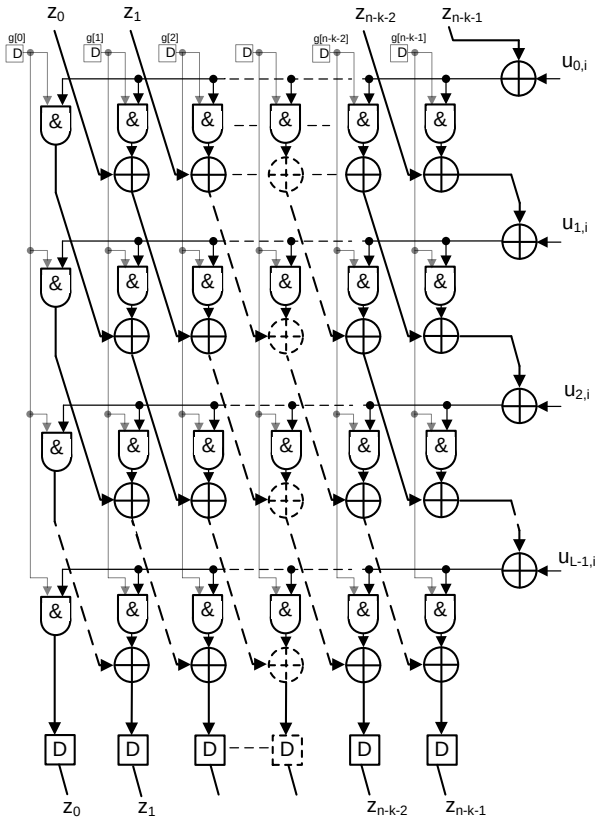

Рис. 5. Параллельный кодер кодов БЧХ с регулируемой корректирующей способностью

Регистры, хранящие коэффициенты $g_0, g_1, ..., g_{n-k-1}$ порождающего полинома являются одноразрядными. Алгоритм работы аналогичен для схемы, описанной для рис. 2, однако кодер к тому же стал реконфигурируемым. Простое изменение состояния регистров $g_0, g_1, ..., g_{n-k-1}$ обеспечивает перенастройку корректирующей способности. Критический путь немногим больше пути кодера без возможности реконфигурирования:

$$\tau = 2 \cdot L \cdot \tau_{\oplus} + L \cdot \tau_{\otimes}, \quad (6)$$

где $\tau_{\oplus}$ – задержка на элементе сумматора (исключающего «ИЛИ»); $\tau_{\otimes}$ – задержка на элементе «И».

Критический путь в L раз больше пути в кодере последовательного потока данных (рис. 3), однако следует учитывать, что каждый такт на входе параллельного кодера (рис. 5) поступает L бит с шины данных.

4. Процедура декодирования БЧХ кодов. Декодирование кодов БЧХ структурно показано на рис. 6. Принятые данные (с возможными ошибками) v(x) поступают в схему декодера, одновременно происходит запись этих данных в буфер FIFO. Декодирование поделено на три основных этапа [9]. Сначала данные поступают в схему вычисления синдромов (признаки ошибок), первые n/L тактов, потому как кодовое слово стало длины n. Дальнейшие вычисления декодер проводит с вычисленными синдромами. Следующий этап – вычисление полинома локаторов ошибок. Например, в алгоритме Берлекэмпа–Мэсси (BMA) без инверсии для данного этапа требуется t тактов (где t – количество исправляемых ошибок, с которым было закодировано переданное кодовое слово). Далее вычисленные коэффициенты уравнения поступают в схему поиска позиций ошибок, в этот же момент происходит считывание данных с буфера, а схема поиска позиций ошибок выдает маску, при сложении с которой искаженные данные исправляются и поступают на выход схемы. При фиксированном параметре m поля Галуа, декодер, реконфигурируемый по количеству исправляемых ошибок, реализуется изменением количества тактов необходимых для работы алгоритма BMA. Таким образом, добавив к схеме декодера управление в зависимости от параметра t (максимальное количество исправляемых ошибок), декодер становится реконфигурируемым в зависимости от корректирующей способности, с которой было закодировано кодовое слово.

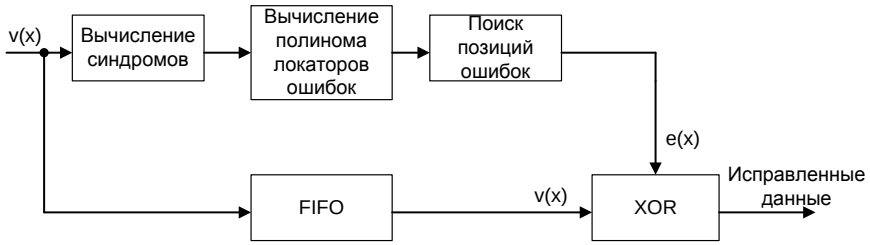

Рис. 6. Структурная схема декодирования БЧХ кодов

5. Экспериментальные результаты. Проведены сравнения по занимаемой площади (количество логических вентилей), а также по задержкам на критических путях для ПЛИС семейства Altera (Arria II). Ниже представлена таблица полученных характеристик для кодеров с максимальной степенью полинома R=208. А именно, степень полинома:

$$R = n - k = m \cdot t, \quad (7)$$

т.е. для поля $GF(2^m)$ при m=13, длина закодированных данных не будет превышать $n < 2^{13}$, а количество исправляемых ошибок исходя из выражения (7) будет не более $t_{\max} = \frac{R}{m} = \frac{208}{13} = 16$. Данный размер полинома позволит кодировать наиболее часто используемыми кодами для хранения информации во flash-памяти, с количеством исправляемых ошибок 16, 15, 14 и т.д. согласно используемой flash-памяти [2].

Таблица 2

Ресурсы ПЛИС, задержка на критическом пути в зависимости от ширины шины данных кодеров (R=208) с постоянным многочленом и с переменным

L	кол-во ячеек, LC		задержка, нс	
	const	var	const	var
1	120	224	1,15	1,37
2	163	226	1,25	1,76
4	202	453	1,56	2,66
8	246	1128	2,80	3,60
16	392	2250	5,59	7,07
24	642	3139	8,77	9,80
32	837	4037	11,21	12,27
40	1017	4932	15,15	15,18
48	1123	5829	18,65	18,78
56	1305	6500	21,23	20,83
64	1500	7395	24,60	24,50

Результаты синтеза логической схемы представлены также на графике рис. 7.

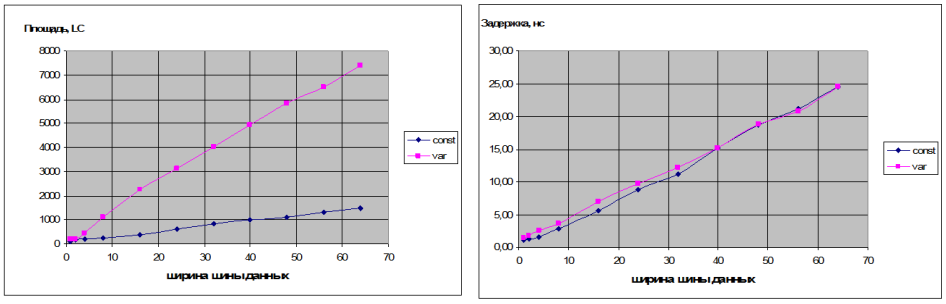

Рис. 7. Ресурсы ПЛИС и задержка на критическом пути в зависимости от ширины шины данных кодеров (R=208) с постоянным многочленом и с переменным

Ниже представлена таблица полученных характеристик для кодеров с максимальной степенью полинома 384. То есть для поля $GF(2^m)$ при m=16, длина закодированных данных не будет превышать $n < 2^{16}$, а количество исправляемых ошибок исходя из выражения (7) будет не более $t_{\max} = \frac{R}{m} = \frac{384}{16} = 24$. Данный размер полинома позволит кодировать большие блоки данных, с количеством исправляемых ошибок 24, 23, 22 и т.д. согласно применению.

Таблица 3

Ресурсы ПЛИС, задержка на критическом пути в зависимости от ширины шины данных кодеров (R=384) с постоянным многочленом и с переменным

L	кол-во ячеек		задержка, нс	
	const	var	const	var
1	164	384	1,24	1,49
2	239	386	1,29	1,84
4	335	773	1,67	2,68
8	392	1928	2,69	3,89
16	568	3845	5,18	6,59
24	1095	5378	8,15	9,65
32	1279	6917	11,93	12,37
40	1473	8453	15,31	15,13
48	1716	9989	17,68	18,89
64	2096	12676	26,11	24,94

Результаты синтеза логической схемы представлены также на графике рис. 8.

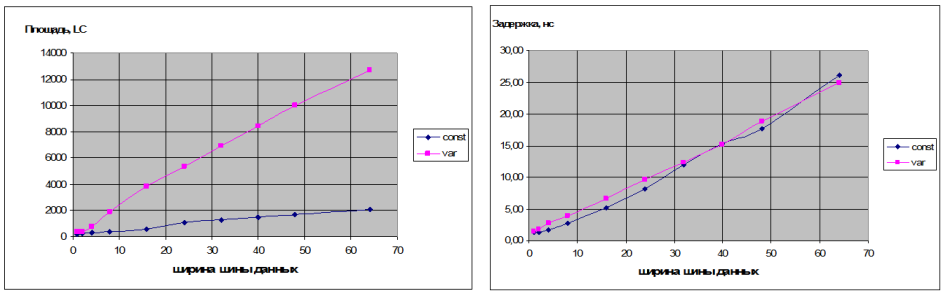

Рис. 8. Ресурсы ПЛИС и задержка на критическом пути в зависимости от ширины шины данных кодеров (R=384) с постоянным многочленом и с переменным

Результаты экспериментов по синтезу логической схемы показали следующее:

1. Задержка на критическом пути не зависит от степени используемого полинома, а зависит лишь от ширины шины данных, что и подтверждает выражение (6).
2. Для конкретного семейства ПЛИС (Arria II) в среднем занимаемая площадь (в единицах логических ячеек) реконфигурируемого кодера в 5–5.5 раз больше кодера с постоянным многочленом.
3. Задержка на критическом пути в конкретном семействе ПЛИС мало зависит от возможности реконфигурирования кодеров, а зависит только от ширины шины данных.

Заключение. В статье предложен способ построения параллельного реконфигурируемого кодера, основанный на традиционной схеме кодера, представленной на рис. 1. Именно она фигурирует во многих источниках как базовая при объяснении кодов БЧХ [1, 3, 4, 5, 17, 20], с ней же и приводится сравнение по быстродействию и площади. Подробно описан переход к реконфигурируемой, и далее к параллельной реализации кодера. В экспериментальной части показано, что реконфигурируемость не влияет на быстродействие, однако обладает универсальностью. При этом именно параллельность приводит к выигрышу по быстродействию. Из табл. 2, задержка тра-

диционного (то есть с постоянным многочленом и шиной данных L=1) равна 1.15 нс, а, к примеру, задержка параллельного кодера с переменным многочленом (шина данных L=32) равна 12.27 нс, то есть в 10.6 раз больше. Однако, за один такт работы параллельный кодер обрабатывает 32 бита данных, в то время как традиционный лишь один бит. Поэтому даже при большем критическом пути в 10.6 раз, общая пропускная способность кодера будет выше в 32/10.6=3 раза. Таким образом, пропускная способность с учетом максимальной тактовой частоты (исходя из критического пути) равна $\frac{1}{12.27 \cdot 10^{-9}} \cdot 32 = 2.6$ Gb/s, что соизмеримо с кодеками [2, 24]. Можно рассмотреть для любой ширины шины данных и увидеть, что реконфигурируемость не влияет на быстродействие, но параллельность приводит к большей пропускной способности. Так, при увеличении ширины шины данных (то есть количества бит обрабатываемых за такт) в L раз критический путь растет менее чем в L раз.

При применении реконфигурируемого кодера для работы с flash-памятью различных типов (более 5) делает это устройство выгодным по занимаемой площади по сравнению с набором нескольких (т.е. более 5) кодеров с постоянными многочленами без перенастройки корректирующей способности. Это видно из табл. 2, 3, количество ячеек кодера с переменным многочленом примерно в 5–6 раз больше количества ячеек кодера с постоянным многочленом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. – М.: Мир, 1986. – 447 с.
2. *Wang Z., Karpovsky M., Joshi A.* Reliable MLC NAND Flash Memories Based on Nonlinear t-Error-Correcting Codes // IEEE/IFIP International Conference on Dependable Systems & Networks. – 2010. – P. 41-50.
3. *Hoyoung Yoo, Youngjioo Lee*. 7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers // IEEE 978-1-4799-2816-3. – 2014. – P. 37-38.
4. *Keshab K.* Parhi Eliminating the fanout bottleneck in parallel long BCH encoders // IEEE Communication Society. – 2004. – P. 2611-2615.
5. *Zhang Jun, Wang Zhi-Gong, Hu Qing-Sheng, Xiao Jie.* Optimized design for high-speed parallel BCH encoder // IEEE Int.Workshop VLSI Design&Video Tech. – 2005. – P. 97-100.
6. *Вернер М.* Основы кодирования. – М.: Техносфера, 2004. – 288 с.
7. *Chuan-Sheng Lin, Kuang-Yuan Chen, Yu-Hsian Wang.* A NAND Flash Memory Controller for SD/MMC Flash Memory Card // IEEE 1-4244-0395. – 2006. – P. 1284-1287.
8. *Song-Chul Jang, Je-Hoon Lee, Won-Chul Lee, Kyoung-Rok Cho.* Design of Parallel BCH Decoder for MLC Memory // IEEE International SoC Design Conference. – 2008. – P. III-46-47.
9. *Морелос-Сарагоса Р.* Искусство помехоустойчивого кодирования. Методы алгоритмы, применение. – М.: Техносфера, 2005. – 320 с.
10. Патент CN101068113. Circuit, coder and device for parallel BCH coding.
11. Патент CN102761340. Broadcast channel (BCH) parallel coding circuit.
12. Патент CN101227194. Circuit, encoder and method for encoding parallel BCH.
13. Патент CN102820892. Circuit for parallel BCH (broadcast channel) coding, encoder and method.
14. *Gomes M., Falcao G., Silva V.* Scalable and Parallel Codec Architectures for the DVB-S2 FEC System // IEEE 978-1-4244-2342-2/08. – 2008. – P 1506-1509.
15. *Cargnini L.V., Fagundes R.D., Bezerra E.A., Almeida G.M*. Parallel Algebraic Approach of BCH coding in VHDL // Proceedings of International Multi-Conference on Computing in the Global Information Technology, 2007.
16. *Kihoon Lee, Han-Gil Kang, Jeong-In Park, Hanho Lee*. A High-Speed Low-Complexity Concatenated BCH Decoder Architecture for 100 Gb/s Optical Communications // Springer Science+ Business Media, 2010. – P.43-55.
17. *Koorapati S., Prakash S.* Design of Any Codeword Length Parallel Long BCH Encoder with the help of An Efficient C-Utility // International Conference on VLSI Systems, Architecture, Technology and Application, 2015.

18. *Yu-Peng Hu, Nong Xiao.* An elastic Error Correction Code Technique for NAND Flash-based Consumer Electronic Devices // IEEE 0098 3063/13. 2013. – P. 1-8.
19. *Aiswariya R., Parameshwaran R.* Loop Unrolling for Second Order Recursive Digital Filter to Achieve High Throughput // Contemporary Engineering Sciences. – 2014. – Vol. 7, No. 8. – P. 357-362.
20. *Zhang X., Parhi K.K.* High-speed Architecture for Parallel Long BCH Encoder // GLSVLSI. – 2004. – P. 26-28.
21. Патент CN101567696B. Encoder and decoder of Code BCH with changeable parameters.
22. Патент US8812940 B2. Programmable Error Correction Capability for BCH Codes.
23. *Yi-Min Lin, Jau-Yet Wu, Chien-Ching Lin, Hsie-Chia Chang.* A Long Block Length BCH Decoder for DVB-S2 Application // ISIC. – 2009. – P. 171-174.
24. *Lee Y., Yoo H., Jung J., Jo J., Park I.* A 2.74-pJ/bit, 17.7-Gb/s Iterative Concatenated-BCH Decoder in 65 nm CMOS for NAND Flash Memory // IEEE Journal of Solid-State Circuits. – 2013. – Vol. 48, No. 10. – P. 2531-2540.

REFERENCES

1. *Bleykhut R.* Teoriya i praktika kodov, kontroliruyushchikh oshibki [Theory and practice codes, controlling errors]. Moscow: Mir, 1986, 447 p.
2. *Wang Z., Karpovsky M., Joshi A.* Reliable MLC NAND Flash Memories Based on Nonlinear t-Error-Correcting Codes, *IEEE/IFIP International Conference on Dependable Systems & Networks*, 2010, pp. 41-50.
3. *Hoyoung Yoo, Youngjioo Lee*. 7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers, *IEEE 978-1-4799-2816-3*, 2014, pp. 37-38.
4. *Keshab K.* Parhi Eliminating the fanout bottleneck in parallel long BCH encoders, *IEEE Communication Society*, 2004, pp. 2611-2615.
5. *Zhang Jun, Wang Zhi-Gong, Hu Qing-Sheng, Xiao Jie.* Optimized design for high-speed parallel BCH encoder, *IEEE Int.Workshop VLSI Design&Video Tech*., 2005, pp. 97-100.
6. *Verner M.* Osnovy kodirovaniya [The basics of coding]. Moscow: Tekhnosfera, 2004, 288 p.
7. *Chuan-Sheng Lin, Kuang-Yuan Chen, Yu-Hsian Wang.* A NAND Flash Memory Controller for SD/MMC Flash Memory Card, *IEEE 1-4244-0395*, 2006, pp. 1284-1287.
8. *Song-Chul Jang, Je-Hoon Lee, Won-Chul Lee, Kyoung-Rok Cho.* Design of Parallel BCH Decoder for MLC Memory, *IEEE International SoC Design Conference*, 2008, pp. III-46-47.
9. *Morelos-Saragosa R.* Iskusstvo pomekhoustoychivogo kodirovaniya. Metody algoritmy, primenenie [The art of error-correcting coding. Methods, algorithms, application]. Moscow: Tekhnosfera, 2005, 320 p.
10. Patent CN101068113. Circuit, coder and device for parallel BCH coding.
11. Patent CN102761340. Broadcast channel (BCH) parallel coding circuit.
12. Patent CN101227194. Circuit, encoder and method for encoding parallel BCH.
13. Patent CN102820892. Circuit for parallel BCH (broadcast channel) coding, encoder and method.
14. *Gomes M., Falcao G., Silva V.* Scalable and Parallel Codec Architectures for the DVB-S2 FEC System, *IEEE 978-1-4244-2342-2/08*, 2008, pp 1506-1509.
15. *Cargnini L.V., Fagundes R.D., Bezerra E.A., Almeida G.M*. Parallel Algebraic Approach of BCH coding in VHDL, *Proceedings of International Multi-Conference on Computing in the Global Information Technology*, 2007.
16. *Kihoon Lee, Han-Gil Kang, Jeong-In Park, Hanho Lee*. A High-Speed Low-Complexity Concatenated BCH Decoder Architecture for 100 Gb/s Optical Communications, *Springer Science+ Business Media*, 2010, pp.43-55.
17. *Koorapati S., Prakash S.* Design of Any Codeword Length Parallel Long BCH Encoder with the help of An Efficient C-Utility, *International Conference on VLSI Systems, Architecture, Technology and Application*, 2015.
18. *Yu-Peng Hu, Nong Xiao.* An elastic Error Correction Code Technique for NAND Flash-based Consumer Electronic Devices, *IEEE 0098 3063/13*, 2013, pp. 1-8.
19. *Aiswariya R., Parameshwaran R.* Loop Unrolling for Second Order Recursive Digital Filter to Achieve High Throughput, *Contemporary Engineering Sciences*, 2014, Vol. 7, No. 8, pp. 357-362.
20. *Zhang X., Parhi K.K.* High-speed Architecture for Parallel Long BCH Encoder, *GLSVLSI*, 2004, pp. 26-28.
21. Patent CN101567696B. Encoder and decoder of Code BCH with changeable parameters.
22. Patent US8812940 B2. Programmable Error Correction Capability for BCH Codes.

23. *Yi-Min Lin, Jau-Yet Wu, Chien-Ching Lin, Hsie-Chia Chang*. A Long Block Length BCH Decoder for DVB-S2 Application, *ISIC*, 2009, pp. 171-174.
24. *Lee Y., Yoo H., Jung J., Jo J., Park I.* A 2.74-pJ/bit, 17.7-Gb/s Iterative Concatenated-BCH Decoder in 65 nm CMOS for NAND Flash Memory, *IEEE Journal of Solid-State Circuits*, 2013, Vol. 48, No. 10, pp. 2531-2540.

Статью рекомендовал к опубликованию к.т.н. С.В. Николаев.

Поперечный Павел Сергеевич – Институт проблем проектирования в микроэлектронике Российской академии наук (ИППМ РАН); e-mail: ppoperechny@elvees.com; 124365, Москва, Зеленоград, ул. Советская, 3; аспирант.

Poperechny Pavel Sergeevich – Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IDPM RAS); e-mail: ppoperechny@elvees.com; 3, Sovetskaya street, Zelenograd, Moscow, 124365, Russia; postgraduate student.

УДК 004.934

И.И. Иванов

РАЗРАБОТКА МЕТОДА ОПРЕДЕЛЕНИЯ ВЕКТОРОВ ПРИЗНАКОВ ПО ПОРОЖДАЕМОМУ РЕЧЕВОМУ СИГНАЛУ ДЛЯ ПРОЦЕДУРЫ АУТЕНТИФИКАЦИИ

Целью статьи является разработка нового метода биометрической аутентификации личности по порождаемому речевому сигналу на основе реконструированных компонент векторов состояния модели речевого процесса. Приведены недостатки существующих систем аутентификации, в которых для извлечения идентифицирующих признаков речевого сигнала применяют спектральные характеристики в частотно-временной области: метода мел-частотных кепстральных коэффициентов (mel frequiency cepstral coefficients), метода кепстральных коэффициентов линейного предсказания (linear prediction cepstral coefficients). Задачей исследования является использование реконструированных компонент векторов состояния в качестве идентифицирующих коэффициентов для систем биометрической аутентификации. Для решения поставленной задачи был разработан метод, который учитывает нестационарность и внутреннюю нелинейную динамику речеобразующего аппарата на основе реконструированных компонент векторов состояния модели речевого процесса. В основе предложенного метода лежат алгоритмы нелинейной динамики и оценки параметров динамических систем по хаотическим временным рядам. Для подтверждения гипотезы о возможности использования реконструированных компонент векторов состояния в качестве идентификационного признака для систем биометрической аутентификации в системах контроля и управления доступом проведен численный эксперимент с использованием описанной модели. Целью эксперимента является подтверждение единообразия полученных из реконструированных компонент векторов состояния идентифицирующих коэффициентов. Численный эксперимент проводился на основе акустического материала, полученного из базы произношений Forvo. В качестве исходных речевых сигналов на вход системы формирования идентифицирующих признаков подавались несжатые 16-битные .wav-файлы (PCM-signed) с частотой дискретизации 44,1 кГц. Результаты эксперимента показали, что скорость реализации предложенного метода значительно выше, чем, основанного на спектральных характеристиках в частотно-временной области – метода мел-частотных кепстральных коэффициентов, поэтому можно с уверенностью утверждать, что применение метода определения векторов признаков по порождаемому речевому сигналу может являться альтернативным подходом для аутентификации пользователей высоконадежных систем. Такой подход позволяет снизить результаты негативного воздействия внешних факторов и избежать необходимости предварительной фильтрации.

Голосовая идентификация личности; вектора состояния; метод мел-частотных кепстральных коэффициентов; метод кепстральных коэффициентов линейного предсказания; компоненты векторов-состояния модели речевого процесса.