УДК 681.32

В. П. СЕМЕРЕНКО
Вінницький національний технічний університет, Україна

КОДУВАННЯ КОДІВ РІДА-СОЛОМОНА НА ОСНОВІ АВТОМАТНИХ МОДЕЛЕЙ

Розглянута автоматна модель кодів Ріда-Соломона (РС) на основі теорії лінійної послідовнісної схеми. Дано теоретичне обґрунтування рекурентного і згорткового способів кодування кодів РС та пропонується їх почергове використання. Проведено аналіз складності виконання алгоритмів кодування при послідовній та паралельній реалізаціях.
Ключові слова: циклічні коди, коди Ріда-Соломона (РС), кодування, лінійна послідовнісна схема, згортка.

V. SEMERENKO,
Vinnytsia National Technical University, UKRAINE

ENCODING OF REED-SOLOMON CODES BASED ON AUTOMATON MODELS

Abstract – *The aim of the research – the theoretical ground of the encoding algorithms of Reed-Solomon (RS) codes with the using the finite automata in unbinary Galois fields – linear finite-state machine (LFSM). New determination of RS codes based on the automaton model is done and the peculiarities of the systematic encoding for two types recursive LFSM are considered.*

Usually the encoding of RS codes is executing by a recurrent method (step by step dividing by the generator polynomial of code) or a convolutional method (by means of the generator matrix or the checking matrix of code). The first method is very slow and the second method requires many equipment and program costs. The combined variant of encoding by the association of above-mentioned methods is suggested. It is possible to use a recurrent method of encoding of RS codes and to get an intermediate result after k *units time and further during one unit time by a convolutional method to complete work.*

Spatial complexity of known convolution is a function from the parameter k *of (* n, k *) RS code and spatial complexity of the offered convolution based on LFSR theory is a function from the parameter* r *of RS code (* $r = n - k$ *). For widely-spread in practice the case* $r << k$ *the essential reduction of the complexity of encoder can be attained.*

Keywords: cyclic codes, Reed-Solomon (RS) codes, encoding, linear finite-state machine (LFSR), convolution.

Вступ

Як показав К. Шеннон в своїй знаменитій статті [1], використання завадостійких кодів в каналах з шумами дозволяє зменшити частоту помилок до прийнятного рівня. З тих пір було розроблено велику кількість різноманітних кодів для виявлення та виправлення помилок [2].

Достойне місце серед них займають циклічні коди, зокрема, їх підклас – коди Ріда-Соломона (РС). Сфера використання кодів РС вражає: супутниковий і мобільний зв'язок, цифрове телебачення, пристрої пам’яті (оптичні диски CD і DVD, дискові масиви RAID 6) та багато іншого **[**3**]**.

Незважаючи на численні публікації за більш, ніж півстолітню історію цих кодів, ще залишається багато невирішених проблем. Однією з них є розробка ефективної процедури кодування кодів РС.

Аналіз проблеми

Завадостійке кодування реалізується через різноманітні технічні компроміси [2]. Теоретично можна виявити чи виправити будь-яку кількість помилок в даних, що передаються. Однак, підвищення коректувальної здатності коду вимагає збільшення ступеня надлишковості, тобто, зменшення частки корисної інформації в кожній порції переданих даних. В результаті знадобиться більше часу для передачі початкових даних з корисною інформацією. На практиці мінімізація часу передавання даних є важливою проблемою [4] і

виникає питання лише про те, якою ціною вона може бути вирішена. Розглянемо цю проблему на алгоритмічному рівні, тобто на рівні алгоритмів кодування кодів РС.

Процес кодування коду РС, як і будь-якого іншого циклічного (n, k)-коду, полягає в тому, що k -розрядні інформаційні слова відображаються в n -розрядні кодові слова ($n > k$), які і передаються по каналу зв'язку. З позицій структури кодове слово Z може бути систематичним або несистематичним. Обмежимось розглядом лише систематичного кодового слова $Z = z_0, z_1, z_2, \ldots, z_{n-1}$, яке формується доданням до початкового k -розрядного інформаційного слова $\mathrm{I} = \iota_0, \iota_1, \iota_2, \ldots, \iota_{k-1}$ r -розрядного контрольного слова $\Psi = \psi_0, \psi_1, \psi_2, \ldots, \psi_{r-1}$ ($r = n - k$).

Для порівняння складності алгоритмів кодування будемо аналізувати їх часову та просторову асимптотичні складності [5].

З позицій реалізації алгоритму кодування відомі два основних способи формування контрольного слова Ψ : рекурентний та згортковий.

Суть рекурентного способу, який найчастіше використовується в каналах зв'язку [6], полягає в покроковому обчисленні слова Ψ з використанням одного із рекурсивних алгоритмів. Наприклад, при поліноміальному представленні циклічного коду алгоритм систематичного кодування полягає в помноженні інформаційного поліному $\iota(x) = \iota_0 + \iota_1 x + \iota_2 x^2 + \ldots + \iota_{k-1} x^{k-1}$ степеня k на одночлен x^r ($r = n - k$) і далі в покроковому діленні його на породжувальний поліном коду $g(x) = g_0 + g_1 x + \cdots + g_r x^r$ степеня r .

Отриманий в результаті ділення поліном остачі $\psi(x)$ займає молодші r розрядів кодового полінома $z(x)$ степеня n :

$$z(x) = \psi(x) + \iota(x), \quad GF(q) .$$

Таким чином, процедура систематичного кодування (n, k)-коду при його поліноміальному представленні завжди вимагає n тактів часу, тому часову складність такого способу кодування можна вважати лінійною $O(n)$. Для зберігання проміжних значень обчислень при систематичному кодуванні використовується r -розрядний регістр зсуву з лінійними оберненими зв'язками – РЗЛОЗ). Отже, просторова складність рекурентного способу кодування становить $O(r)$.

Згортковий спосіб кодування може бути реалізовано при представленні коду РС за допомогою породжувальної ($k \times n$)-матриці або перевіряльної $(r \times n$)-матриці.

При використанні систематичної породжувальної матриці G_S i -а компонента z_i систематичного кодового слова Z обчислюється як результат одновимірної згортки

$$z_i = \sum_{j=0}^{k-1} w_{i,j} \times \iota_j , \quad GF(q), \quad \iota_j \in \mathrm{I}, \quad w_{i,j} \in G_S , \quad i = 0 \ldots n-1 . \tag{1}$$

При використанні систематичної перевіряльної матриці H_S i -а компонента ψ_i контрольного слова Ψ обчислюється як результат одновимірної згортки

$$\psi_i = \sum_{j=0}^{k-1} h_{i,j} \times \iota_j , \quad GF(q), \quad \iota_j \in \mathrm{I}, \quad h_{i,j} \in H_S , \quad i = 0 \ldots r-1 . \tag{2}$$

Останній спосіб обчислень використовується в стандартах кодування DAT і CIRC (Cross Interlived Reed-Colomon Code – код РС з перемежуванням) для оптичних дисків [3].

Якщо всі компоненти інформаційного слова $\mathbf{I}$ та значення самих матриць G_S та H_S наперед відомі, тоді операції згортки (1) і (2) можуть бути виконані протягом одного такту часу (одного кроку), отже, часова складність обчислення одновимірної згортки буде константною і становитиме $O(1)$.

Для обчислення i-го компонента ($i = 1 \ldots n$) слова Z необхідно k вузлів множення та ($k - 1$) суматорів, тобто просторова складність обчислення всього слова Z згортковим способом за формулою (1) способом становитиме $O(n k^2)$.

Для обчислення i-го компонента ($i = 1 \ldots r$) слова Ψ необхідно k вузлів множення та ($k - 1$) суматорів, тобто просторова складність обчислення всього слова Ψ згортковим способом за формулою (2) способом становитиме $O(r k^2)$.

Відзначимо, що відомі різні оцінки складності обчислення згортки над полем $GF(q)$ [7]. Наприклад, швидкі способи обчислення згортки (1) і (2) мають просторову складність відповідно $O(n k \log k)$ і $O(r k \log k)$.

Таким чином, зменшення часової складності алгоритму кодування компенсується суттєвим підвищенням його просторової складності. В цьому проявляється один з фундаментальних принципів схемотехнічного проектування, і його не можна обійти. Задача полягає лише в знаходженні оптимального компромісу між перерозподілом складностей для різних технічних характеристик, який може бути реалізований на практиці.

Мета і задачі досліджень

Метою даної роботи є розробка методів кодування кодів РС на основі математичного апарату лінійних послідовнісних схем (ЛПС). Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Дати теоретичне обґрунтування процедури кодування кодів РС на основі їх автоматного представлення.

2. Дослідити компроміси між часовою та апаратною складністю різних процедур кодування в недвійкових полях Галуа та реалізувати швидке кодування кодів РС з прийнятною на практиці складністю кодера.

Автоматні методи представлення кодів РС

Традиційно код РС, який дозволяє виправити $\tau_{\min}$ помилок, описується над полем $GF(q)$ породжувальним поліномом

$$g(x) = (x - \alpha)(x - \alpha^2) \ldots (x - \alpha^{2\tau_{\min}}), \; GF(q). \qquad (3)$$

Для розв'язання поставленої задачі будемо використовувати автоматно-аналітичний спосіб представлення кодів РС на основі теорії ЛПС [8]. В цьому випадку ЛПС над полем $GF(q)$ (назвемо таку ЛПС символьною) задається функцією станів (переходів):

$$S(t+1) = A \times S(t) + B \times U(t), \quad GF(q) \qquad (4)$$

і функцією виходів

$$Y(t) = C \times S(t) + D \times U(t), \quad GF(q),$$

де t – дискретний час; $A = \lfloor a_{ij} \rfloor_{r \times r}$, $B = \lfloor b_{ij} \rfloor_{r \times l}$, $C = \lfloor c_{ij} \rfloor_{m \times r}$, $D = \lfloor d_{ij} \rfloor_{m \times l}$ – характеристичні матриці; $S(t) = \lfloor s_i \rfloor_r$ – слово стану, $U(t) = \lfloor u_i \rfloor_l$ – вхідне слово, $Y(t) = \lfloor y_i \rfloor_m$ – вихідне слово.

Будемо використовувати ЛПС з одним входом і одним виходом ($l = 1, w = 1$), для якої функція виходу збігається з функцією стану: $Y(t) = S(t)$.

Будемо розрізняти автоматно-аналітичну і автоматно-графову моделі коду РС [9]. Оскільки символьна ЛПС, є кінцевим автоматом, тому як автоматно-графову модель коду РС можна вибрати граф переходів-виходів G_{FA} цього автомата. Автоматно-графова модель коду РС описана в [10].

До речі, формула для обчислення циклів на РЗЛОЗ в [11] безпосередньо випливає із функції (4) ЛПС, оскільки РЗЛОЗ – це найпростіший випадок ЛПС.

Проведемо детальний аналіз автоматно-аналітичної моделі коду РС, яка базується на характеристичних матрицях ЛПС.

Якщо породжувальний поліном (3) перетворити до вигляду

$$g(x)=\alpha_0^i+\alpha_1^i x+\alpha_2^i x^2+\ldots+\alpha_{2\tau_{\min}-1}^i x^{2\tau_{\min}-1}+x^{2\tau_{\min}},\ GF(q), \qquad (5)$$

тоді можна отримати чотири типи символьних ЛПС: рекурсивні ЛПС типу Галуа, рекурсивні ЛПС типу Фібоначчі, нерекурсивні ЛПС типу Галуа та нерекурсивні ЛПС типу Фібоначчі.

Розглянемо лише перші два типи ЛПС, які використовуються для систематичного кодування кодів РС.

Рекурсивні ЛПС типу Галуа – це ЛПС, у яких сигнали на входи поступають з їх виходів, а характеристичні матриці мають вигляд:

$$A=\begin{vmatrix} 0 & 0 & \cdots & 0 & \alpha_0^i \\ \alpha^0 & 0 & \cdots & 0 & \alpha_1^i \\ 0 & \alpha^0 & \cdots & 0 & \alpha_2^i \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha^0 & \alpha_{d_{\min}-1}^i \end{vmatrix};\quad B=\begin{vmatrix} \alpha^0 \\ 0 \\ 0 \\ \cdots \\ 0 \end{vmatrix},\ C=\begin{vmatrix} 0 & \ldots & 0 & \alpha_0 \end{vmatrix};\ D=|0|.$$

Рекурсивні ЛПС типу Фібоначчі – це ЛПС, у яких сигнали на входи поступають з їх виходів, а характеристичні матриці мають вигляд:

$$A=\begin{vmatrix} 0 & \alpha^0 & 0 & \ldots & 0 \\ 0 & 0 & \alpha^0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_0^i & \alpha_1^i & \alpha_2^i & \ldots & \alpha_{2\tau_{\min}-1}^i \end{vmatrix};\quad B=\begin{vmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ \alpha^0 \end{vmatrix};\ C=\begin{vmatrix} \alpha_0 & 0 & \ldots & 0 \end{vmatrix};\quad D=|0|.$$

Для кожного типу ЛПС структура всіх характеристичних матриць є стандартною і незмінною, за винятком одного рядка або стовпця, елементи яких ($\alpha_0^i, \alpha_1^i, \alpha_2^i, \ldots, \alpha_{2\tau_{\min}-1}^i$) визначаються коефіцієнтами породжувального полінома (5).

Кодування кодів РС за допомогою рекурсивних ЛПС

На основі автоматно-аналітичної моделі можна дати означення коду РС.

ОЗНАЧЕННЯ 1. Множина всіх двійкових послідовностей M довжини n, які переводять ЛПС із будь-якого початкового стану $S_{beg}(t)$ знову в стан $S_{beg}(t)$, утворює (n,k)-код РС Ω над полем Галуа $GF(q)$. Кожна така послідовність M є кодовим словом Z (n,k)-коду РС.

Виходячи із наведеного означення коду РС, задача кодування зводиться до знаходження такого кодового слова Z довжиною n, яке при подачі на входи ЛПС переводить її з деякого початкового стану $S_{beg}(t)$, знову в цей же стан. Як початковий стан $S_{beg}(t)$ будемо надалі розглядати нульовий стан $S(0)$.

Розглянемо систематичне кодування циклічних кодів за допомогою рекурсивних ЛПС. Суть процедури кодування в цьому випадку буде такою.

Під дією на вхід інформаційного слова I ЛПС перейде з початкового стану $S(0)$ в деякий стан

$$S(k)=\begin{bmatrix} s_0^k \\ s_1^k \\ \dots \\ s_{r-2}^k \\ s_{r-1}^k \end{bmatrix}.$$

Далі необхідно визначити контрольне слово Ψ, яке переведе ЛПС із стану $S(k)$ знову в стан $S(0)$. В підсумку стане відомим кодове слово Z. Спочатку покажемо існування такого слова Ψ.

ТЕОРЕМА 1. Для рекурсивної ЛПС типу Галуа або типу Фібоначчі існує слово Ψ довжиною r, яке переводить ЛПС із стану $S(k)$ в стан $S(0)$.

Доведення. Згідно з [8] для будь-якої пари станів $S(i)$ і $S(j)$ існує вхідна послідовність із r символів, яка переводить r-вимірну ЛПС із $S(i)$ в $S(j)$, якщо ЛПС є r-керованою. А ЛПС буде r-керованою, якщо ранг $r\times r$-матриці

$$L_r=\left|A^{r-1}\times B,\quad A^{r-2}\times B,\quad \dots\quad ,\quad A\times B,\quad B\right| \tag{6}$$

буде дорівнювати r. Підставляючи матриці A і B рекурсивної ЛПС типу Галуа в (6), отримаємо такий вигляд матриці L_r:

$$L_r^G=\begin{bmatrix} 0 & 0 & \dots & 0 & \alpha^0 \\ 0 & 0 & \dots & \alpha^0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \alpha^0 & \dots & 0 & 0 \\ \alpha^0 & 0 & \dots & 0 & 0 \end{bmatrix}. \tag{7}$$

Підставляючи матриці A і B рекурсивної ЛПС типу Фібоначчі в (6), отримаємо такий вигляд матриці L_r:

$$L_r^F=\begin{bmatrix} \alpha^0 & 0 & \dots & 0 & 0 \\ l_{2,1} & \alpha^0 & \dots & 0 & 0 \\ l_{3,1} & l_{3,2} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{r,1} & l_{r,2} & \dots & l_{r,r-1} & \alpha^0 \end{bmatrix},\quad l_{i,j}=\{\alpha^0,\alpha^1,\dots,\alpha^{n-1}\},\ i\neq j. \tag{8}$$

Ненульові значення діагональних елементів матриць (7) і (8) свідчать про те, що їх ранг дорівнює r. Отже, існує слово Ψ довжиною r, яке переводить ЛПС із стану $S(k)$ в довільний заданий стан, в тому числі і в стан $S(0)$. Теорема доведена.

$\perp$

Розглянемо кодування за допомогою рекурсивної ЛПС типу Галуа.

ТЕОРЕМА 2. Систематичне кодування (n,k)-коду РС за допомогою рекурсивної ЛПС типу Галуа може бути виконано або за ($k+1$) тактів після подачі на її вхід інформаційного слова $\mathbf{I}$, або за n тактів після подачі на її вхід інформаційного $\mathbf{I}$ і нульового O слів. В першому випадку компоненти ψ_i слова $\Psi=\{\psi_0,\psi_1,\dots,\psi_{r-2},\psi_{r-1}\}$ можуть бути знайдені в результаті обчислення згортки

$$\Psi_i = \sum_{j=0}^{r-1} a_{i,j} \times s_j^k , \quad GF(q) , \tag{9}$$

де s_j^k – j-та компонента слова стану $S(k)$; $a_{i,j}$ – компоненти r-го степеня матриці A ($s_j \in S(k)$, $a_{i,j} \in A^r$, $i = 0 \ldots r-1$, $j = 0 \ldots r-1$);

а у другому випадку дорівнюють:

$$\Psi_i = s_{r-1-j}^m , \tag{10}$$

де s_{r-1-j}^m – ($r-1-j$)-та компонента слова стану $S(m)$; ($s_{r-1-j}^m \in S(m)$, $j = 0 \ldots r-1$).

Доведення. Із теорії ЛПС [8] відомо, що при подачі на вхід ЛПС, яка знаходиться в деякому початковому стані $S(0)$, інформаційного слова $\mathbf{I}$ довжиною k ЛПС перейде в стан $S(k)$, що визначається з рівності

$$S(k) = A^k \times S(0) + L_k \times \mathbf{I}, \quad GF(q) . \tag{11}$$

Якщо далі на вхід ЛПС подати контрольне слово Ψ довжиною r, тоді ЛПС перейде в стан $S(n)$, який визначається співвідношенням

$$S(n) = A^r \times S(k) + L_r \times \Psi, \quad GF(q) . \tag{12}$$

Оскільки $S(n) = S(0)$, тому рівність (12) можна записати як

$$L_r \times \Psi = A^r \times S(k) . \tag{13}$$

Підставляючи значення матриці L_r із (7) в (13), отримаємо згортку (9). Оскільки стан $S(k)$ буде отримано на k-му такті функціонування ЛПС, отже, контрольне слово Ψ може бути обчислено на ($k+1$)-му такті, і весь процес кодування виконується за ($k+1$) тактів.

Продовжимо аналіз рівності (12). Якщо в цю рівність підставити значення $S(k)$ із (13), тоді отримаємо:

$$S(n) = A^r \times (A^k \times S(0) + L_k \times \mathbf{I}) + L_r \times \Psi =$$

$$= A^{k+r} \times S(0) + A^r \times L_k \times I + L_r \times \Psi = S(0) + A^r \times L_k \times I + L_r \times \Psi, \quad GF(q) .$$

Добуток $L_k \times I$ визначає стан, в який перейде ЛПС після подачі на її вхід слова I, а добуток $A^r \times L_k \times I$ визначає деякий стан $S(m)$, в який перейде ЛПС після наступної подачі на її вхід нульового слова O довжиною r. Таким чином, значення слова стану $S(n)$ буде таким:

$$S(n) = S(0) + S(m) + L_r \times \Psi, \quad GF(q) . \tag{14}$$

Оскільки метою кодування є перехід із початкового стану $S(0)$ знову в цей же стан, тому $S(n) = S(0)$, і тоді із виразу (14) випливає, що повинна виконуватися рівність $L_r \times \Psi = S(m)$.

Підставивши значення матриці L_r із (9), отримаємо:

$$\Psi = -S(m) . \tag{15}$$

Знак «–» в (15) означає операцію векторної інверсії, тобто взаємної перестановки між молодшими і старшими компонентами слова.

В підсумку отримаємо співвідношення між компонентами слів $S(m)$ і Ψ, яке міститься в (10). Оскільки стан $S(m)$ буде отримано на n-му такті функціонування ЛПС, отже, контрольне слово Ψ може бути знайдено на n-му такті, і кодування виконується за n тактів. Теорема доведена.

$\perp$

Таким чином, процес кодування розбивається на два етапи. На першому етапі поступає інформаційне слово $\mathbf{I}$, як правило, послідовно. Тому спочатку можна використати рекурентний спосіб кодування за формулою (6). Під дією слова $\mathbf{I}$ ЛПС протягом k тактів перейде з нульового стану $S(0)$ в стан $S(k)$.

На другому етапі кодування формується контрольне слово Ψ. Якщо це слово буде передаватись в канал також послідовно, тоді обчислити компоненти ψ_i можна рекурентним способом за r тактів. Якщо ж необхідно швидко, за один такт, обчислити слово Ψ і передати його в канал паралельно (наприклад, в комп'ютерній мережі), саме тоді знадобиться згортковий спосіб кодування.

ПРИКЛАД. Нехай над полем Галуа $GF(8)$ задано інформаційне слово

$$I = \left\lfloor \alpha^1 \quad \alpha^5 \quad \alpha^3 \quad \alpha^4 \quad \alpha^0 \quad \alpha^2 \quad \alpha^5 \quad \alpha^6 \quad \alpha^0 \quad \alpha^2 \quad \alpha^{12} \right\rfloor.$$

Для систематичного кодування використаємо (15,11)-код РС, якому відповідає породжувальний поліном

$$g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4) = \alpha^{10} + \alpha^3 x + \alpha^6 x^2 + \alpha^{13} x^3 + x^4, \qquad GF(8)$$

та характеристичні матриці рекурсивної ЛПС типу Галуа:

$$A = \begin{vmatrix} 0 & 0 & 0 & \alpha^{10} \\ \alpha^0 & 0 & 0 & \alpha^3 \\ 0 & \alpha^0 & 0 & \alpha^6 \\ 0 & 0 & \alpha^0 & \alpha^{13} \end{vmatrix}; \quad B = \begin{vmatrix} \alpha^0 \\ 0 \\ 0 \\ 0 \end{vmatrix}.$$

Перший етап кодування виконаємо рекурентним способом за формулою (4). В результаті ЛПС протягом 11 тактів перейде зі стану $S(0)$ в стан $S(11)$ (з метою економії місця будемо записувати стовпці станів ЛПС у вигляді рядків):

$$S(11) = \left\lfloor \alpha^9 \quad \alpha^3 \quad \alpha^4 \quad \alpha^{13} \right\rfloor.$$

Другий етап кодування виконаємо згортковим способом за формулою (9). Для цього знадобиться четверта степінь матриці A (її можна наперед підготувати і використовувати для будь-якого інформаційного слова цього коду):

$$A^4 = \begin{vmatrix} \alpha^{10} & \alpha^8 & \alpha^{11} & \alpha^{11} \\ \alpha^3 & \alpha^8 & \alpha^5 & \alpha^{13} \\ \alpha^6 & \alpha^7 & \alpha^{11} & \alpha^{13} \\ \alpha^{13} & \alpha^1 & \alpha^1 & \alpha^{10} \end{vmatrix}.$$

Тепер можна обчислити компоненти слова $\Psi = \left\lfloor \psi_0 \; \psi_1 \; \psi_2 \; \psi_3 \right\rfloor$:

$$\psi_0 = \alpha^{10}\alpha^9 + \alpha^8\alpha^3 + \alpha^{11}\alpha^4 + \alpha^{11}\alpha^{11} = \alpha^{10}, \quad GF(8);$$

$$\psi_1 = \alpha^3\alpha^9 + \alpha^8\alpha^3 + \alpha^5\alpha^4 + \alpha^{13}\alpha^{11} = \alpha^0, \quad GF(8);$$

$$\psi_2 = \alpha^6\alpha^9 + \alpha^7\alpha^3 + \alpha^{11}\alpha^4 + \alpha^{13}\alpha^{11} = \alpha^{13}, \quad GF(8);$$

$$\psi_3 = \alpha^{13}\alpha^9 + \alpha^1\alpha^3 + \alpha^1\alpha^4 + \alpha^{10}\alpha^{11} = \alpha^1, \quad GF(8).$$

Аналогічним чином можна довести можливість згорткового кодування за допомогою рекурсивної ЛПС типу Фібоначчі.

Аналіз складності автоматно-аналітичного кодування кодів РС

На основі автоматних моделей для обчислення i-го компонента ($i = 1 \ldots r$) слова Ψ необхідно r вузлів множення та ($r-1$) суматорів, тобто просторова складність обчислення всього слова Ψ згортковим

способом за формулою (9) способом становитиме $O(r^3)$. Швидкий спосіб обчислення згортки (9) має просторову складність $O(r^2 \log r)$

Як було вже раніше показано, зменшити часову складність кодування можна тільки при згортковому способі кодування. Такий висновок базувався на припущенні, що такти часу при рекурентному та згортковому способах кодування приблизно однакові. Однак, останній спосіб вимагає великих апаратних витрат, відповідно, сумарна затримка всіх обчислювальних елементів може перевищити часові витрати рекурентного способу.

Відомі різні підходи до зменшення апаратних витрат у кодерах, наприклад, використання породжувальних поліномів з симетричними коефіцієнтами [12]. Але суттєво зменшити фізичний час виконання операції згортки можна лише при переході до паралельної обробки даних.

До речі, при рекурентному кодування паралелізм також використовується: значення окремих компонент ψ_i ($i = 1 \ldots r$) контрольного слова Ψ формуються одночасно. Для розв'язання такої ж задачі згортковим способом необхідно, щоб всі одновимірні згортки також обчислювались одночасно – це становитиме перший рівень паралелізму.

Другий рівень паралелізму має бути реалізовано при обчисленні самої згортки. Наприклад, для обчислення одновимірної згортки (9) знадобиться ($r-1$) помножувачів та ($r-1$) суматорів, які утворюють пірамідальну структуру. При реалізації помножувачів та суматорів на комбінаційній логіці [13], всі одновимірні згортки (1), (2) та (9) можуть бути обчислені протягом одного такту роботи кодера.

Обчислення згортки (1) або (2) за один такт вимагає попередньої наявності всього інформаційного слова $\mathbf{I}$. Кількість m компонент слова $\mathbf{I}$, яка може бути передана одночасно по каналу зв'язку, обмежена розрядністю l канальних шин ($l < k$). Це вимагатиме додаткового часу для підготовки до кодування слова $\mathbf{I}$. Якщо ж компоненти слова $\mathbf{I}$ поступають послідовно, тоді кодування раціонально виконати рекурентним способом.

На практиці, враховуючи високу просторову складність згорток (1) і (2), їх реалізація, як правило, здійснюється програмно (наприклад, для оптичних дисків). В цьому випадку просторова складність обумовлена обсягом пам'яті для збереженням породжувальної або перевіряльної матриці (32 Кбіт для методу CIRC [3]).

Варто також врахувати, що використання вкорочених (n, k_1)-кодів РС ($k_1 < k$) в методі CIRC [3]) призводить до деякого спрощення реалізації згорток (1) і (2). Складність обчислення згортки (9) залишається незмінною, але автоматне представлення вкорочених кодів РС дозволяє швидше виконати операцію декодування [14].

Висновки

Завадостійке кодування кодів РС може бути виконане рекурентним або згортковим способом. Перший спосіб характеризується мінімальною просторовою складністю, однак вимагає найбільше часу. Згортковий спосіб, навпаки, є найшвидшим при великих апаратних і програмних витратах.

Пропонується комбінований варіант кодування поєднанням зазначених вище підходів. На перших k тактах кодування (n, k)-коду РС можна використати рекурентний спосіб і отримати проміжний результат, а далі протягом одного такту згортковим способом завершити роботу. Просторова складність обчислення відомих згорток (1) і (2) є функцією від параметра k коду, а просторова складність запропонованої згортки (9) на основі теорії ЛПС є функцією від параметра r коду. Оскільки на практиці $r << k$, тому можна досягти суттєвого зменшення апаратних витрат при обчисленні згортки (9). Виграш у часі можна також отримати і при програмній реалізації кодування з використанням ЛПС.

При виборі способу кодування потрібно враховувати і спосіб введення і виведення даних в кодері. Запропонований спосіб буде оптимальним при послідовному (покомпонентному) вводі інформаційного слова і

паралельному виведенні кодового слова. Якщо ж в кодері дані вводяться і виводяться лише послідовно, тоді доцільним буде лише рекурентне кодування на всіх n тактах роботи.

Література

1. Шеннон К. Работы по теории информации и кибернетике / К. Шеннон – М. : Изд-во иностр. лит., 1963. – 829 с.

2. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр; пер. с англ.; 2-е изд., перераб. – М.: Издательский дом “Вильямс”, 2004. – 1104 с.

3. Coding Theory and Cryptography. The Essentials / [D. R. Hankerson, D. G. Hoffman, D. A. Leonard, C. C. Lindner, and others] Wall. Second Edition, Revised and Expanded. – New York : CRC Press. – 2000. – 350 p.

4. Захарченко Н. В. Компенсація надлишковості в блокових коректуючих кодах за рахунок таймерних сигналів / Н. В. Захарченко, В. Й. Кільдішев, С. В. Хомич, О. Г. Пришляк // Вісник Хмельницького національного університету. – 2011. – № 2. – С. 175–181.

5. Ахо А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. ; Пер. с англ. – М. : Мир, 1979. – 536 с.

6. Al Azad A. Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog / A. Al Azad, M. Huq, I. Rahman Rokon // International Conference on Advancements in Electronics and Power Engineering (ICAEPE'2011), Bangkok Dec., 2011. – P. 117-121.

7. Жуков И. А. Оценка сложности вычислений в конечных полях / И. А. Жуков, В. И. Кубицкий // Інформаційні технології та комп'ютерна інженерія. – 2013. – № 2. – С. 21–27.

8. Гилл, А. Линейные последовательностные машины : пер. с англ. / А. Гилл. – М. : Наука, 1974. – 288с.

9. Семеренко В. П. Теорія циклічних кодів на основі автоматних моделей : монографія [Текст] / В. П. Семеренко. – Вінниця : ВНТУ, 2015. – 444 с.

10. Семеренко В. П. Декодирование кодов Рида-Соломона на основе графовой и автоматной моделей / В. П. Семеренко // Электронное моделирование. – 2011. – № 1. – С. 57–72.

11. Лисицина Е. С. Исследование циклов генераторов на регистрах сдвига с обратными связями / Е. С. Лисицина // Вісник Хмельницького національного університету. – 2014. – № 1. – С. 121–125.

12. Singh A. Design and Implementation of Reed Solomon Encoder on FPGA / A. Singh, M. Kaur // Intern. Journal of Computer, Electrical, Automation, Control and Information Engineering. – 2013. – Vol. 7. – No. 9. – P. 33–39.

13. Вышенчук И. М. Алгоритмические операционные устройства и суперЭВМ / И. М. Вышенчук, Н. В. Черкасский – К. : Тэхника, 1990. – 197 с.

14. Семеренко В. П. Параллельное декодирование укороченных циклических кодов / В. П. Семеренко // Оптико-электронные информационно-энергетические технологии. – 2012. – № 1. – С. 30–41.

References

1. Shannon C. E. (1948). A mathematical theory of communication / C. E Shannon. – Bell Syst. Tech. J., 1948. – Vol. 27. – P. 379–423 (Part 1), P. 623–656 (Part 2).

2. Sklar B. (2001). Digital Communications. Fundamentals and Applications. 2nd ed. Los Angeles: Prentice Hall. (Russ. Ed.: Skljar, B. Cifrovaja svjaz'. Teoreticheskie osnovy i prakticheskoe primenenie, 2-e izd. Moscow : Izdatel'skij dom “Vil'jams”, 2004. 1104 p.)

3. Coding Theory and Cryptography. (2000). The Essentials / [D. R. Hankerson, D. G. Hoffman, D. A. Leonard, C. C. Lindner, and others] Wall. Second Edition, Revised and Expanded. – New York : CRC Press. – 350 p.

4. Zakharchenko N. V. (2001). Kompensatsiia nadlyshkovosti v blokovykh korektuiuchykh kodakh za rakhunok taimernykh syhnaliv / N. V. Zakharchenko, V. Y. Kildishev, S. V. Khomych, O. H. Pryshliak // Visnyk Khmelnytskoho natsionalnoho universytetu. – No. 2. – S. 175–181.

5. Aho A. V. (1976). The Design and Analysis of Computer Algorithms / A. V. Aho, J. E. Hopcroft, and J. D. Ullman. – Menlo Park, California : Addison-Wesley Publishing Company. – 470 p.

6. Al Azad A. (2011). Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog / A. Al Azad, M. Huq, I. Rahman Rokon // International Conference on Advancements in Electronics and Power Engineering (ICAEPE'2011), Bangkok Dec. – P. 117-121.

7. Zhukov I. A. (2013). Otsenka slozhnosti vyichisleniy v konechnyih polyah / I. A. Zhukov, V. I. Kubitskiy // Informatsiyni tekhnolohyy ta komp'yuterna inzheneriya. – No 2. – S. 21–27.

8. Gill A. (1967). Linear Sequential Circuits. Analysis, Synthesis and Application. New York, London: McGraw-Hill Book Company.

9. Semerenko V. P. (2015). Teoriia tsyklichnykh kodiv na osnovi avtomatnykh modelei: monohrafiia / V. P. Semerenko. – Vinnytsia : VNTU. – 444 s.

10. Semerenko V. P. (2011). Dekodyrovanye kodov Ryda-Solomona na osnove hrafovoi y avtomatnoi modelei / V. P. Semerenko // Elektronnoe modelyrovanie. – No. 1. – S. 57–72.

11. Lysytsyna E. S. – (2014). Issledovanye tsyklov heneratorov na rehystrakh sdvyha s obratnymi sviaziamy / E. S. Lysytsyna // Visnyk Khmelnytskoho natsionalnoho universytetu. – No. 1. – S. 121–125.

12. Singh A. (2013). Design and Implementation of Reed Solomon Encoder on FPGA / A. Singh, M. Kaur // Intern. Journal of Computer, Electrical, Automation, Control and Information Engineering. – Vol. 7. – No. 9. – P. 33–39.

13. Vyshenchuk Y. M. (1990). Alhorytmycheskye operatsyonnye ustroistva y superEVM / Y. M. Vyshenchuk, N. V. Cherkasskyi – K. : Tekhnyka, 1990. – 197 s.

14. Semerenko V. P. (2012). Parallelnoe dekodyrovanye ukorochennykh tsyklycheskykh kodov / V. P. Semerenko // Optyko-elektronnye ynformatsyonno-enerhetycheskye tekhnolohyy. – No. 1. – S. 30–41.

Рецензія/Peer review : 23-10-2015 Надрукована/Printed :

Рецензент: Завідувач кафедри менеджменту та безпеки інформаційних систем Вінницького національного технічного університету д.т.н., проф. Роїк О.М.