Vol. 3, No. 1, 2021

DOI: https://doi.org/10.23939/csn2021.01.001

УДК 519.6:004.77

ПРОГРАМНА МОДЕЛЬ КОДІВ РІДА-СОЛОМОНА

Є. Я. Ваврук[1], Б. Р. Попович[2], Р. Б. Попович[3]

Національний університет Львівська політехніка,
[1]кафедра електронних-обчислювальних машин,
[3]кафедра спеціалізованих комп'ютерних систем,
[2]Національний медичний університет, кафедра медичної інформатики/
E-mail: *yevhenii.y.vavruk@lpnu.ua*

Розроблена програма для моделювання завадостійких кодів Ріда-Соломона на основі об'єктно-орієнтованої технології. Вхідними даними для системи є блоки байтів для передачі через канал зв'язку, де в цих блоках можуть статися помилки. Створена програма реалізує коди типу (255,239) та (255,223) для скінченого поля з 256 елементів GF(2^8) зі стандартними породжуючими багаточленами $x^8+x^4+x^3+x^2+1$ та $x^8+x^7+x^2+x+1$. Крім того, передбачена можливість у випадку необхідності додати інші типи кодів та багаточлени, які породжують скінченне поле.

Ключові слова – завадостійкий код, скінченне поле, процедура кодування, процедура декодування.

Вступ

Потреба використання лише достовірної інформації зумовила використання різних засобів (криптологія, завадостійке кодування, стеганографія) [1] для захисту даних від завад. Серед великої кількості кодів, які забезпечують завадостійкість повідомлення, виділяється код Ріда-Соломона [6]. Код має багато застосувань, серед яких найпопулярнішими є споживчі технології (MiniDiscs, компакт-диски, DVD-диски, диски Blu-ray, QR-коди), технології передачі даних (DSL, WiMAX), системи мовлення (супутниковий зв язок, DVB, ATSC), системи зберігання даних (RAID 6).

З математичного погляду – це потужний код з високою коректуючою здатністю. Водночас алгебраїчні процедури кодування та декодування для даного коду є простими, ефективними та прозорими [3, 5, 7].

Окреслення проблеми

Зростання сфер використання кодів Ріда-Соломона зумовлює розробку та впровадження нових методів та пристроїв їх кодування та декодування.

Хоча існують доступні програми, які моделюють роботу кодів Ріда-Соломона, проте:

– їх важко використати через відсутність належних їх описів та проблему отримання їх належно працюючих версій;

– вони не реалізують усіх потрібних варіантів опрацювання даних.

Цим диктується потреба в розробці програмної моделі кодів Ріда-Соломона.

Завдання роботи

Розробити з використанням об'єктно-орієнтованої технології програму, яка моделює роботу кодів Ріда-Соломона. Передбачити можливість легкої її модифікації на різні типи кодів та стандартні багаточлени.

Отримані результати

При розробці використано середовище візуального програмування Delphi версії 7.0.

Скінченне поле з q елементів (q – натуральний степінь простого числа) позначаємо через GF(q). У розробленій програмі процедури кодування та декодування, а також операції додавання та множення для цих процедур, реалізовано в скінченому полі GF(2^8) з 256 елементів. Це поле й виконання операцій у ньому залежать від багаточлена, що породжує поле. Реалізовано стандарт DVB (багаточлен $x^8+x^4+x^3+x^2+1$) та стандарт Intelsat (багаточлен $x^8+x^7+x^2+x+1$).

Якщо ж треба було б моделювати завадостійкі коди Ріда-Соломона над скінченним полем з кількістю елементів більшою, ніж 256, тобто працювати з полями GF(2^9), GF(2^{10}), GF(2^{11}), GF(2^{12}), GF(2^{13}), GF(2^{14}), GF(2^{15}), GF(2^{16}), то треба елементи цих полів зображати не як байти, а як слова. Для цього слід внести відповідні зміни в функцію множення елементів скінченного поля. Зауважимо, що коди Ріда-Соломона над зазначеними скінченними полями також починають входити в ужиток.

Блок-схема розробленої програми зображена на рис. 1. Для опису кодування, виникнення помилок у каналі зв'язку та декодування програма працює з трьома файлами: data.txt, code.txt та decode.txt. Згадані файли описані в програмі як файли типу txt.

Файл data.txt є блоком даних, які надсилаються в канал зв'язку, де вони, можливо, будуть спотворені через виникнення фізичних завад. Він складається з рядків, кожен з яких містить напис i-й байт (i=0,1,…,238 для кодів типу (255,239) та i=0,1,…,222 для кодів типу (255,223)) й значення відповідного байта в шістнадцятковій системі числення.

З метою захисту корисних (інформаційних) байтів від спотворень у каналі зв'язку виконується кодування цих даних і створюється файл code.txt. У цьому файлі є більше байтів, ніж у початковому файлі data.txt. У нашому конкретному випадку кількість байт дорівнює 255.

Файл decode.txt є протоколом декодування даних, які отримані після проходження закодованих даних через канал зв'язку. Він містить повідомлення про номери байтів файлу code.txt, в яких сталися помилки в каналі зв'язку, та вірні (відтворені) значення цих спотворених байтів. Крім того, у цьому файлі наводяться дані, які отримані на етапах декодування – конкретні обчислені значення синдрому помилок, багаточлена місць (локаторів) помилок та багаточлена значень помилок.

Далі описано процедури кодування та декодування для кодів Ріда-Соломона [2, 5, 7], які використано при розробці програмної моделі.

Кодування для кодів Ріда-Соломона

Нехай $(d_{k-1}, d_{k-2}, ..., d_1, d_0)$ позначає k m-бітових символів даних (у нашому випадку байтів), які треба передати через канал зв'язку (або зберігати в пам'яті). Ці символи розглядаємо як елементи скінченого поля GF(2^m) та кодуємо кодовим словом $(c_{n-1}, c_{n-2}, ..., c_1, c_0)$ з $n > k$ символів. Дані кодові символи передаємо через канал зв'язку.

Для коду Ріда-Соломона над GF(2^m), $n = 2^m - 1$, k – непарне, код може виправляти до $t = (n-k)/2$ спотворених символів включно. Процес кодування описуємо як перетворення багаточлена даних $D(z) = d_{k-1}z^{k-1} + d_{k-2}z^{k-2} + ... + d_1 z + d_0$ в кодовий багаточлен $C(z) = c_{n-1}z^{n-1} + c_{n-2}z^{n-2} + ... + c_1 z + c_0$. $C(z)$ ділиться на твірний багаточлен коду $G(z) = \prod_{i=0}^{2t-1} (z - \alpha^{i+1})$, де α – примітивний елемент скінченного поля.

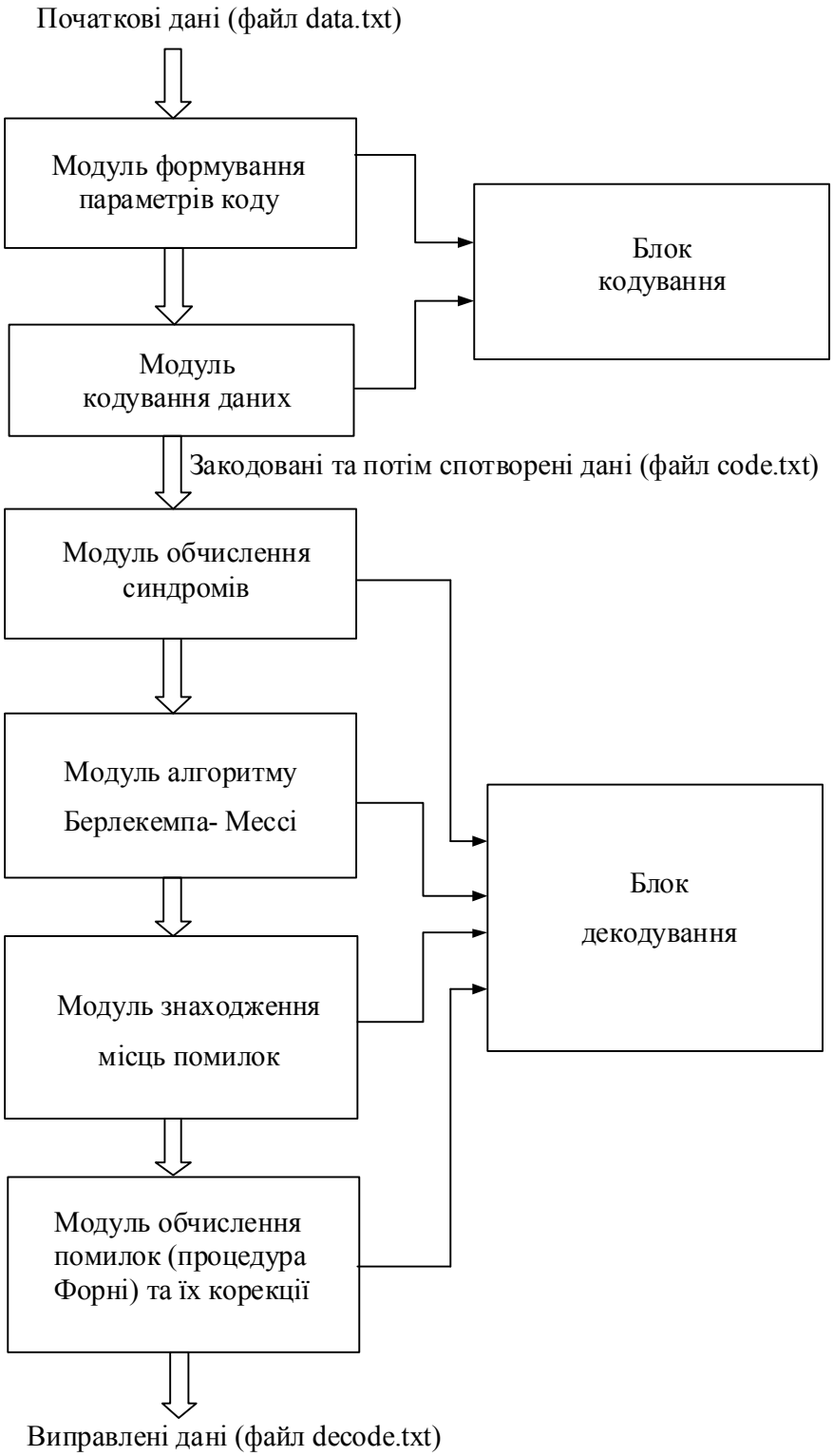

Рис. 1. Блок схема програми.

Систематичний процес кодування утворює кодові слова, що складаються з символів даних, за якими йдуть контрольні (перевіркові) символи. Кодові слова утворюємо так. Багаточлен $z^{n-k}D(z)$ степеня $n-1$ ділимо на багаточлен $G(z)$ степеня $2t=n-k$. Залишок $P(z)$ від цього ділення є багаточленом степеня щонайбільше $n-k-1$ і його коефіцієнти дають контрольні символи.

Декодування для кодів Ріда-Соломона

$C(z)$ позначає багаточлен кодового слова, яке було надіслане, а $R(z)$ – багаточлен прийнятого кодового слова. Входом декодера є $R(z)$, він задовольняє співвідношення

$$R(z) = C(z) + E(z).$$

Якщо степінь e багаточлена значень помилок $E(z)$ не рівний нулю, то це означає, що під час передачі трапились помилки. Багаточлен $E(z)$ можна записати у вигляді

$$E(z) = Y_1 z^{i_1} + Y_2 z^{i_2} + ... + Y_e z^{i_t}$$

в припущенні, що значення помилок $Y_1, Y_2, ..., Y_e$ сталися на місцях помилок $X_1 = \alpha^{i_1}, X_2 = \alpha^{i_2}, ..., X_e = \alpha^{i_e}$.

Завданням декодера є визначити $E(z)$, маючи на вході $R(z)$. Тоді декодер виправляє помилки, віднімаючи $E(z)$ від $R(z)$. Якщо $e \le t$, то це можливо, тобто можна виправити до t помилок включно.

Перший етап декодування полягає в обчисленні значень синдромів

$$s_i = R(\alpha^{i+1}) = E(\alpha^{i+1}), 0 \le i \le 2t-1.$$

Якщо всі $2t$ значення синдромів рівні нулю, значить помилок не сталося. В іншому випадку $e > 0$ і використовуємо багаточлен синдромів $S(z) = s_0 + s_1 z + ... + s_{2t-1} z^{2t-1}$, щоб знайти місця помилок і значення помилок.

Означимо багаточлен місць (локаторів) помилок $\Lambda(z)$ степеня e та багаточлен значень помилок $\Omega(z)$ степеня щонайбільше $e-1$:

$$\Lambda(z) = \prod_{j=1}^{e} (1 - X_j z),$$

$$\Omega(z) = \sum_{i=1}^{e} Y_i X_i \prod_{j=1, j\ne i}^{e} (1 - X_j z).$$

Ці багаточлени зв'язані з $S(z)$ таким так званим ключовим рівнянням:

$$\Lambda(z) S(z) \equiv \Omega(z) \bmod z^{2t}.$$

Другий етап декодування – розв'язання ключового рівняння для визначення $\Lambda(z)$ та $\Omega(z)$ по $S(z)$ – є найскладнішою частиною процесу декодування. Для його розв'язання використано алгоритм Берлекемпа-Мессі [2, 4] – ітеративну процедуру для рішення ключового рівняння. Нами реалізована модифікація цього алгоритму, в якій відсутні операції знаходження оберненого в скінченому полі. Вона приваблива з точки зору подальшої апаратної реалізації.

На третьому етапі знаходимо місця помилок, перевіряючи чи $\Lambda(\alpha^{-j}) = 0$ для кожного $j, 0 \le j \le n-1$. Цей процес називають пошуком Чена [5]. Якщо $\Lambda(\alpha^{-j}) = 0$, то α^j є одним з місць помилок (скажімо X_i). Іншими словами, r_j є помилкою, і її треба скоректувати.

На четвертому етапі обчислюємо значення помилки Y_i, яке потім додаємо до r_j, за допомогою такого виразу (процедура Форні):

$$Y_i = -\left. \frac{z\Omega(z)}{z\Lambda'(z)} \right|_{z = \alpha^{-j}},$$

де $\Lambda'(z)$ позначає формальну похідну для $\Lambda(z)$.

Реалізація процесів кодування й декодування залежить від того, як реалізовані операції скінченого поля. Як правило, в алгоритмах використовують такі операції: додавання, піднесення до квадрату, множення довільного елемента на константу, множення (довільних елементів), знаходження оберненого відносно множення елемента. Операції додавання, піднесення до квадрату, множення на константу є відносно простими, а операції множення та знаходження оберненого –складніші.

У результаті роботи розробленої програми користувач може бачити в одному з вікон програми блок початкових даних. Ці дані можна задавати й записувати в файл data.txt (рис. 2).

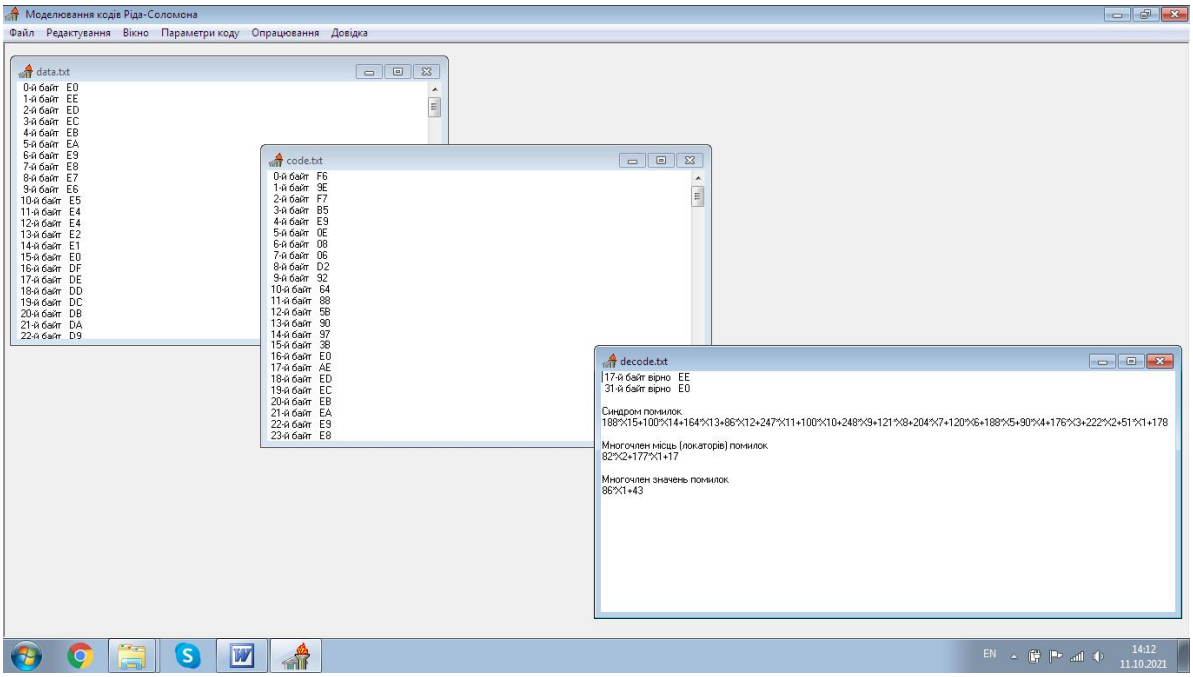

Рис. 2. Вікно програми з початковими й закодованими даними та протоколом декодування.

В іншому вікні можна переглядати закодовані початкові дані, спотворювати їх (імітуючи помилки в каналі зв'язку) і записувати в файл code.txt.

У третьому вікні можна переглядати протокол декодування закодованих (і потім спотворених) початкових даних. У цьому протоколі вказуються номери спотворених байтів та правильні значення цих байтів (тобто ті, які були до спотворення). Крім цього, наводяться обчислені в процесі декодування проміжні результати.

Пункт меню "Параметри коду" дозволяє вибирати тип коду Ріда-Соломона ((255,239) або (255,223)) та стандартний багаточлен скінченого поля (стандарт DVB або стандарт Intelsat), згідно з якими виконуються обчислення при кодуванні та декодуванні. Крім того, передбачена можливість у випадку необхідності легко додати інші типи кодів та багаточлени, які породжують скінченне поле.

Пункт меню "Опрацювання" дозволяє запускати процес кодування чи декодування.

Тестування роботи програмного продукту, проведене на різних блоках даних, показало ефективність виправлення помилок. При перевірці програмного засобу на надійність програма запускалася з неправильними або суперечливими даними. Кожного разу це було виявлено і програма давала відповідне повідомлення.

Висновки

Розроблено програму для моделювання завадостійких кодів Ріда-Соломона на основі об'єктно-орієнтованої технології. Вхідними даними для неї є блоки байтів для передачі через канал зв'язку, де в цих блоках можуть статися помилки. Створена програма реалізує коди типу (255,239)

та (255,223) для породжуючих багаточленів скінченого поля з 256 елементів GF(2^8) $x^8+x^4+x^3+x^2+1$ (стандарт DVB) та $x^8+x^7+x^2+x+1$ (стандарт Intelsat). Крім того, передбачена можливість у випадку необхідності легко додати інші типи кодів та багаточлени, які породжують скінченне поле. Програмний продукт може бути рекомендований для використання при проектуванні апаратних та програмних засобів для різноманітних систем передачі інформації.

1. Emets V., Melnyk A., Popovych R. Suchasna kryptografiia: osnovni poniattia.- Lviv: Vydavnytsvo BaK, 2003. -144 P. (In Ukrainian).

2. Berlekamp E. R. Algebraic Coding Theory.– Singapore: World Scientific Publishing Co, 2015. – 501 P.

3. Lin S., Costello D. J. Error Control Coding.– Pirson: Prentice Hall, 2004.–1272 P.

4. Massey J. L. Shift-register synthesis and BCH decoding // IEEE Transactions on Information Theory, vol. 15, no. 1, 1969, p. 122–127.

5. Reed I. S., Chen X. Error-Control Coding for Data Networks.– Boston: Kluwer Academic Publishers, 1999.–549 P.

6. Reed I. S., Solomon G. Polynomial Codes over Certain Finite Fields // Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, 1960, p. 300–304.

7. Tomlinson M., Tjhai C. J., Ambroze M. A., Ahmed M., Jibril M. Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications. –Springer, 2017.–522 P.

PROGRAM MODEL OF REED-SOLOMON CODES

E. Vavruk[1], B. Popovych[2], R. Popovych[3]

Lviv Polytechnic National University,
[1] Department of Electronic Computing Machines
[3]Department of Specialized Computer Systems
[2] National Medical University, Department of Medical Informatics

Software is designed for modeling of Reed-Solomon codes on a base of object-oriented technology. Input data for system are blocks of bytes for transmitting through communication channel, where errors can occur in the blocks. Designed program realizes codes of (255,239) and (255,223) type for finite field GF(2^8) with standard generating polynomials $x^8+x^4+x^3+x^2+1$ and $x^8+x^7+x^2+x+1$. Moreover, a possibility is provided to add other types of codes and generating polynomials.

Key words – error correcting code, finite field, encoding, decoding.