# Швидке декодування паралельних кодів CRC

Василь Семеренко, Богдан Григорчук
кафедра обчислювальної техніки,
Вінницький національний технічний університет
Вінниця, Україна
vpsemerenko@ukr.net

# Fast decoding of parallel CRC codes

Vasyl Semerenko, Bogdan Grygorchuk
Department of Computer Technique
Vinnytsia National Technical University
Vinnytsia, Ukraine,
vpsemerenko@ukr.net

***Анотація*—Запропоновано новий спосіб швидкого декодування кодів CRC (Cyclic Redundancy Code) в каналах з паралельним надходженням вхідних даних. Використання теорії паралельних ЛПС (лінійних послідовнісних схем) та математичного представлення симетрії часу дозволяє вдвічі прискорити CRC-контроль. Такий спосіб контролю може бути використано в системах зберігання та архівації даних.**

***Abstract*—A new method for fast decoding of CRC (Cyclic Redundancy Code) in channels with parallel input data is proposed. The use of the theory of linear finite state machine (LFSM) and the mathematical representation of time symmetry makes it possible to double the speed of the check with the help of CRC. This check method can be used in data storage and archiving systems.**

***Ключові слова*—коди CRC; контрольна сума; паралельні обчислення; лінійна послідовнісна схема; декодування**

***Keywords*—CRC codes; checksum; parallel processing; linear finite state machine; decoding**

## I. Вступ

Виявлення помилок за допомогою контролю CRC найчастіше використовується в різноманітних системах передачі даних, зокрема в обчислювальних мережах стандарту Ethernet [1]. CRC також є ефективним для забезпечення цілісності даних, які зберігаються на різноманітних пристроях збереження даних, зокрема на магнітних дисках та дискових масивах.

В останні роки CRC стали успішно застосовувати і в ПЛІС для безперервного контролю цілісності даних в конфігураційній пам'яті [2]. Однак, цей метод апаратного контролю віднімає значну частину ресурсів ПЛІС. Наприклад, в сімействі ПЛІС Cyclone зменшується максимальна частота на 9%, збільшується використання провідників на 21% та збільшується час трасування на 9%. Тому актуальною є задача зменшення використання ресурсів обчислювальних пристроїв, зокрема часу, при збереженні засобів контролю даних.

## II. Теоретичний базис паралельних кодів CRC

CRC має дві розшифровки його абревіатури і, відповідно, дві інтерпретації [3]. Найчастіше CRC розглядають як *Cyclic Redundancy Check* – циклічний надлишковий контроль, тобто контрольну суму. В цьому випадку CRC є ущільненим представленням заданої вхідної послідовності $I$ довільної довжини (близький аналог в криптографії – хеш-функція). Зміна значення контрольної суми з великою ймовірністю свідчить про наявність помилок в послідовності $I$.

Інтерпретація CRC як *Cyclic Redundancy Code* – циклічного надлишкового коду, дозволяє перевіряти правильність даних за відомими правилами завадостійкого кодування.

В сучасних багатоканальних системах передачі даних реалізована паралельна передача даних: біти одного байту або слова (2, 4, 8 байт) поступають одночасно. Кожний байт або слово можна інтерпретувати як один $\rho$-бітовий символ ($\rho$=8,16,32,64), тоді для передачі $w$ біт знадобиться $m$ символів ($m = w/\rho$). Такий спосіб передавання даних називається символьно-паралельним [4].

Послідовність із $m$ символів будемо розглядати як кодове слово коду CRC, яке формується кодером на стороні передавача та декодується декодером на стороні приймача. Передачу даних можна розглядати або як передачу по послідовному каналу символів, або як передачу бітів даних по $\rho$ паралельним каналам. В першому випадку знадобляться математичні перетворення в недвійкових полях Галуа $GF(2^\rho)$, а в другому випадку – математичні перетворення в двійкових полях Галуа $GF(2)$.

В цій роботі розглядається лише другий випадок, більш простий для програмно-апаратної реалізації. Такий спосіб інтерпретації даних означатиме, що код CRC складається із $\rho$ кодових

слів $z_i$ ($i = 1 \ldots \rho$), об'єднаних в кодову матрицю:

$$Z_{(\rho)} = \begin{bmatrix} z_1 \\ z_2 \\ \ldots \\ z_\rho \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \ldots & z_{1n} \\ z_{21} & z_{22} & \ldots & z_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ z_{\rho 1} & z_{\rho 2} & \ldots & z_{\rho n} \end{bmatrix}, GF(2). \quad (1)$$

Коди CPC є різновидом циклічних кодів, тому для їх опису часто використовують традиційні способи представлення циклічних кодів (матричне, поліноміальне, алгебраїчне). Найчастіше циклічний код задається породжувальним поліномом:

$$g(x) = g_0 + g_1 x + \cdots + g_{r-1} x^{r-1} + x^r, \quad GF(2) \quad (2)$$

З позицій теорії циклічного кодування коди CPC належать або до циклічних кодів Хемінга, або до кодів Абрамсона [3]. Будемо використовувати коди Хемінга, які мають найбільшу довжину. Таким чином, породжувальний поліном (2) має задовольняти двом вимогам:

– поліном має бути примітивним;

– степінь поліному має бути $r \geq log_2 m$ .

Для опису паралельних кодів найбільш придатним математичним апаратом є теорія лінійних послідовнісних схем (ЛПС) [5].

Традиційна ЛПС з одним входом і одним виходом є кінцевим автоматом лінійного типу (лінійним автоматом), який над полем Галуа $GF(2)$ описується функцією станів (переходів)

$$S(t+1) = A \times S(t) + B \times U(t), \quad GF(2) \quad (3)$$

і функцією виходів

$$Y(t) = C \times S(t) + D \times U(t), \quad GF(2),$$

де $t$ – дискретний час; $A, B, C, D$ – характеристичні матриці ЛПС; $S(t)$ – слово стану; $U(t)$ – вхідне слово; $Y(t)$ – вихідне слово.

В подальшому виходами ЛПС будемо вважати значення її відповідних станів, тому для такої ЛПС достатньо використати лише функцію станів (переходів) ЛПС (3). Апаратною реалізацією такої ЛПС є звичайний регістр зсуву з лінійними оберненими зв'язками.

Розрізняють два типи паралельних кодів CRC: складені та інтегровані [6]. В подальшому будемо говорити лише про інтегровані коди CRC.

Вхідні дані паралельного коду CRC поступають по паралельних каналах, тому необхідно використати багатовходову ЛПС. Для опису структури входів ЛПС використовується характеристична матриця B, тому у випадку $\rho$-входової ($\rho = 1 \div r$) паралельної ЛПС над двійковим полем Галуа має бути така одинична ($r \times r$)-матриця B:

$$B_{(\rho)} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Матриця $A$ визначає внутрішню структуру ЛПС. Багатовходова ЛПС має таку ж внутрішню структуру, що і традиційна одновходова ЛПС, тому матриця $A$ залишиться без змін. Серед різних типів ЛПС найбільш поширеною є рекурсивна ЛПС типу 1 з матрицею

$$A = \begin{bmatrix} 0 & 0 & 0 & \ldots & g_0 \\ 1 & 0 & 0 & \ldots & g_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 1 & \ldots & g_{r-2} \\ 0 & 0 & 0 & 1 & g_{r-1} \end{bmatrix}. \quad (4)$$

Елементи останнього стовпця матриці $A$ із (4) представляють собою коефіцієнти породжувального поліному (2).

Таким чином, теоретичною основою паралельних кодів CRC при декодуванні кодової матриці (1) може бути паралельна ЛПС, функціонування якої описується функцією станів (переходів)

$$S(t+1) = A \times S(t) + B_{(\rho)} \times Z_{(\rho)}(t), \quad GF(2) \quad (5)$$

## III. Декодування паралельного коду CRC

Традиційний спосіб перевірки даних полягає у формування для них контрольної суми $\Sigma_f$, наприклад, CRC-суми і порівнянні її з наперед обчисленої (еталонною) контрольною сумою $\Sigma_v$: їх рівність вважається доказом відсутності спотворень у даних. Контрольні суми, як правило, обчислюються послідовно, що потребує багато часу. Якщо дані поступають також послідовно, тоді затримка з обчисленням суми $\Sigma_f$ буде практично відсутньою. Якщо ж дані можна отримати швидше, наприклад, використовуючи додаткові канали зв'язку, тоді бажано обчислити швидше і контрольні дані.

Будемо вважати, що дані поступають посимвольно, тобто всі можливості по прискоренню обчислень за рахунок використання паралельної обробки даних вже використані. Окрім традиційного каналу зв'язку, по якому дані передаються від початку набору даних, є також можливість зчитувати дані з кінця набору даних. В результаті дані будуть передані вдвічі швидше. Чи можна в даному випадку прискорити вдвічі процес обчислення контрольної суми $\Sigma_f$ ?

Рекурсивне обчислення CRC-суми (як і CRC-коду) здійснюється згідно формули (5), тобто в

порядку надходження $t$-го вектора $Z_{(\rho)}(t)$. Іншими словами, рекурсивні обчислення традиційно здійснюються при зміні часу від "теперішнього" в "майбутнє".

Відомо, що фундаментальні закони класичної та квантової динаміки обернені в часі, з чого випливає математична еквівалентність "минулого" і "майбутнього" [7]. Для динамічних систем з одним ступенем свободи, прикладом яких може служити ЛПС, рекурсивні обчислення можуть здійснюватись також і при зміні часу від "теперішнього" в "минуле".

Таким чином, якщо дані можна одночасно передавати в двох напрямках: від початку набору даних (файлу) та з кінця набору даних (файлу), тоді можна одночасно використати дві ЛПС для обчислення CRC-суми. Перша ЛПС з матрицею $A$ (назвемо її прямою) визначає перехід із початкового нульового стану $S(0)$ в наступні такти часу $t>0$ згідно формули (5). Друга ЛПС з матрицею $A_{inv}$ (назвемо її оберненою) визначає перехід із кінцевого стану $S(n)$ в попередні такти часу $t<n$ згідно формули:

$$S(t)=A_{inv}\times(S(t+1)+B_{(\rho)}\times Z_{(\rho)}(t)),\quad GF(2).$$

Правила переходу між матрицями $A$ і $A_{inv}$ наведено в [6].

В такт часу $t=n/2$ пряма ЛПС перейде в стан $S_f(n/2)$. В цей же такт часу обернена ЛПС перейде в стан $S_v(n/2)$. При відсутності помилок в наборі даних (файлі) зазначені стани мають бути однаковими:

$$S_f(n/2)=S_v(n/2) \text{ при непарному } n,$$
$$S_f(n/2)=S_v(n+1/2) \text{ при парному } n.$$

Відзначимо, що тут проявляється невелика відмінність між двома інтерпретаціями CRC. Якщо розглядати CRC як контрольну суму, тоді кінцевий стан $S(n)$ і є еталонною CRC-сумою, як правило, ненульовою.

Якщо ж розглядати CRC як циклічний код, тоді довжина послідовності $I$ дорівнює $(m+r)$, а кінцевий стан $S(n)$ є синдромом помилки коду (обов'язково нульовим, оскільки помилок перед початком передачі даних ще немає). В цьому випадку роль CRC відіграє ненульове $r$-розрядне контрольне слово коду.

Отже, при наявності одночасного доступу до початку та кінця масиву даних, який контролюється, перевірка коректності даних буде виконана вдвічі швидше. Це правило є справедливим не тільки при бітовій передачі даних [8], але також і при символьній передачі даних.

## IV. Висновки

Основним методом прискорення обчислень є використання паралельної обробки даних. При використанні контролю даних на основі CRC ефективним є паралелізм по протилежним осям часу. В роботі показано як можна вдвічі прискорити обчислення CRC лише на основі тих даних, що вже використовуються в системах передачі даних. Якщо будуть відомі еталонні проміжні контрольні суми, тоді можна ще в декілька разів прискорити обчислення CRC.

Такий спосіб контролю може бути використано в системах зберігання та архівації даних при умові одночасного доступу до початку та кінця масиву даних.

## Література References


[1]. В. Столлингс, Компьютерные системы передачи даннях. Изд. 6-е, пер. с англ., М., Издательский дом «Вильямс», 928 с., 2002.

[2]. Д. Иоффе, "Обнаружение и исправление ошибок с использованием CRC в устройствах FPGA фирмы Altera," *Компоненты и технологии*, no. 8, 2006.

[3]. В. П. Семеренко, "Теория и практика CRC кодов: новые результаты на основе автоматных моделей," *Східно-Європейський журнал передових технологій*, том. 4, випуск 9 (76), с. 38–48, 2015.

[4]. V.P. Semerenko, "The Theory of Parallel CRC Codes Based on Automaton Models," *Eastern-European Journal of Enterprise Technologies*, vol. 6, issue 9 (84), pp. 45–55, 2016.

[5]. А. Гилл, *Линейные последовательностные машины*. Пер. с англ., М., Наука, 288 с., 1974.

[6]. В. П. Семеренко, *Теорія циклічних кодів на основі автоматних моделей*. Монографія, Вінниця, ВНТУ, 444 с., 2015.

[7]. И. Пригожин, И. Стенгерс, *Время, хаос, квант*. Пер. с англ., М., издат. группа Прогресс, 272 с., 1994.

[8]. В. П. Семеренко, Б. О. Григорчук, "Швидке декодування кодів CRC на основі симетрії часу," Науково-технічна конференція Вінницького національного технічного університету (ВНТУ), 15-16 березеня 2017, Вінниця, 2017.