УДК 004.056.3

РЕКУРРЕНТНЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ ФОРМАЛЬНОЙ ПРОИЗВОДНОЙ ПОЛИНОМА НАД ПОЛЕМ ГАЛУА И ЕГО АППАРАТНАЯ РЕАЛИЗАЦИЯ

Рахман П.А.

ФГБОУ ВПО «Уфимский государственный нефтяной технический университет», филиал в г. Стерлитамаке, e-mail: pavelar@yandex.ru

В данной статье рассматривается алгоритм вычисления формальной производной полинома над полем Галуа. Вычисление формальной производной является частью алгоритма декодирования информационного кадра при применении кодов Рида-Соломона. Формальная производная необходима для вычисления значения искаженных символов в информационном кадре при использовании метода Форни. В статье дается определение полинома, формальной производной полинома и формула для ее вычисления над полем Галуа $GF(p^m)$. Также в статье рассматривается алгоритм вычисления значения формальной производной полинома над полем Галуа $GF(2^m)$ при заданном аргументе и предлагаемая автором аппаратная реализация алгоритма.

Ключевые слова: коды Рида-Соломона, поле Галуа, формальная производная, полином

RECURRENT ALGORITHM FOR COMPUTING THE FORMAL DERIVATIVE OF A POLYNOMIAL OVER GALOIS FIELD AND ITS HARDWARE IMPLEMENTATION

Rahman P.A.

Ufa State Petroleum Technological University, Sterlitamak branch, e-mail: pavelar@yandex.ru

This paper deals with recurrent algorithm for computing the formal derivative of a polynomial over Galois field. Calculation of formal derivative is a part of the decoding algorithm for information frame, encoded on application of Reed-Solomon codes. Formal derivative is necessary in the algorithm based on Forney method for computing the error-values of corrupted symbols in information frame. In this paper the definition of polynomial, formal derivative of polynomial over Galois field $GF(p^m)$ and calculation formula for formal derivative are given. Calculation algorithm for value of formal derivative at the given argument over Galois field $GF(2^m)$ and hardware implementation of the algorithm offered by author are also overviewed.

Keywords: Reed-Solomon codes, Galois field, formal derivative, polynomial

На сегодняшний день каналы передачи данных и носители информации остаются далекими от совершенства. Кабельные и беспроводные линии передачи информации подвержены воздействию внешних помех, искажающих форму передаваемых сигналов и тем самым делающих невозможным однозначное распознавание информации на стороне приемника, магнитные и оптические носители информации чувствительны к физическим повреждениям, делающим невозможным чтение информации из отдельных участков на поверхности носителя.

В настоящее время в системах хранения и передачи данных применяют различные технологии информационного резервирования с применением специальных алгоритмов кодирования на базе корректирующих кодов, в частности, кодов Рида-Соломона [1], которые за счет использования избыточной информации делают возможным исправление искажений. Однако, алгоритмы кодирования и декодирования информации с применением кодов Рида-Соломона достаточно нетривиальны, и для эффективной программной или аппаратной реализации требуется достаточно глубокая математическая проработка с применением алгебры конечных полей Галуа [2]. В частности, алгоритм декодирования состоит из блока вычисления синдрома искажений, полинома локаторов искажений, самих локаторов искажений и значений искажений, и все блоки имеют дело с полиномами, заданными над конечным полем Галуа. Особое место занимает блок вычисления значений искажений на базе метода Форни, использующего формальную производную полинома заданного над полем Галуа, и здесь требуется особый подход для построения эффективного алгоритма вычисления значения формальной производной полинома при заданном аргументе.

В рамках научных исследований автора в области надежности систем хранения, передачи и обработки данных [3–9], а также в области информационного резервирования [10] возникла научная задача разработки эффективного алгоритма вычисления формальной производной полинома $\Psi(x)$, заданного над полем Галуа $GF(p^m)$, при заданном аргументе x, и аппаратной реализации для поля Галуа $GF(2^m)$.

Полином над полем Галуа. Пусть задано поле Галуа $GF(p^m)$. Тогда полиномом,

заданным над полем Галуа, будем называть функцию:

$$\Psi(x) = \Psi_{k-1}x^{k-1} + \ldots + \Psi_1 x + \Psi_0 = \sum_{i=0}^{k-1} \Psi_i x^i;$$

$$\Psi_i \in GF(p^m);\ i = 0 \ldots k-1. \qquad (1)$$

Над полиномами можно производить алгебраические операции, также на место переменной x можно подставлять конкретное значение, являющееся элементом поля Галуа $GF(p^m)$, и вычислять значение функции. При пересчете коэффициентов полинома при выполнении операции или вычислении значении функции, строго соблюдаются правила арифметики для поля $GF(p^m)$.

Формальная производная полинома. Вычисление производной от полинома, заданного в поле Галуа, требует особого подхода, так как мы не можем говорить о бесконечно малых величинах, поскольку в поле Галуа нет таких элементов, которые бы мы могли использовать в качестве бесконечно малого. Однако, никто нам не запрещает говорить о некоторой формальной бесконечно малой величине, обозначим ее ε, которая будет нами использоваться исключительно для выполнения математических преобразований.

Кроме того, ради общности рассуждений мы рассмотрим формальную производную полинома, заданного над полем Галуа $GF(p^m)$, а затем рассмотрим частный случай производной над полем Галуа $GF(2^m)$.

Тогда, мы можем дать определение формальной производной функции $f(x)$:

$$\frac{d}{dx} f(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon}. \qquad (2)$$

Особо отметим, что не следует «в лоб» пытаться искать численное значение предела при конкретных значениях аргумента x. Формула задана исключительно для аналитических преобразований, подразумевающих исключение величины ε из итогового аналитического выражения для производной функции. Заметим, что так же, как и в традиционной алгебре, здесь справедливы следующие свойства формальной производной функции:

$$\frac{d(\beta f(x))}{dx} = \beta \frac{df(x)}{dx};$$

$$\frac{d(f(x) \pm g(x))}{dx} = \frac{df(x)}{dx} \pm \frac{dg(x)}{dx};$$

$$\frac{d(f(x)g(x))}{dx} = \frac{df(x)}{dx} g(x) + \frac{dg(x)}{dx} f(x).$$

Теперь же выведем формальные производные для простейших функций:

● Константная функция $f(x) = \beta \Rightarrow$

$$\frac{d}{dx} f(x) = \lim_{\varepsilon \to 0} \frac{\beta - \beta}{\varepsilon} = \lim_{\varepsilon \to 0} \frac{0}{\varepsilon} = 0.$$

● Функция $f(x) = \beta x \Rightarrow$

$$\frac{d}{dx} f(x) = \beta \lim_{\varepsilon \to 0} \frac{x + \varepsilon - x}{\varepsilon} = \beta \lim_{\varepsilon \to 0} \frac{\varepsilon}{\varepsilon} = \beta.$$

Однако, прежде чем продолжить выводить производные степенных функций более высокого порядка, рассмотрим подробнее выражение $(x+\varepsilon)^k$, $k \geq 1$. В традиционной алгебре для возведения в степень k выражения $x + \varepsilon$ существует простая общая формула:

$$(x+\varepsilon)^k = \sum_{i=0}^{k} C_k^i x^i \varepsilon^{k-i}.$$

Биномиальный коэффициент $C_k^i = \frac{k!}{i!(k-i)!}$ представляет собою количество одинаковых (повторяющихся) слагаемых $x^i\varepsilon^{k-i}$ (для каждого $i = 0 \ldots k$), образуемых и затем складывающихся при возведении в степень k выражения $x + \varepsilon$. Учтем арифметическое свойство полей Галуа $GF(p^m)$:

$$\overbrace{\underbrace{a + \ldots + a}_{n \text{ слагаемых}}}^{GF(p^m)} = \overbrace{\lambda a}^{GF(p^m)},$$

где $\lambda = \overbrace{n \bmod p}^{<R,\{+,\cdot\}>}$.

В нашем случае, $a = x^i\varepsilon^{k-i}$ и $n = C_k^i$, и тогда:

$$\overbrace{\underbrace{x^i\varepsilon^{k-i} + \ldots + x^i\varepsilon^{k-i}}_{n \text{ слагаемых}}}^{GF(p^m)} = \overbrace{x^i\varepsilon^{k-i}(C_k^i \bmod p)}^{GF(p^m)}.$$

Тогда, учитывая все сказанное, можно вывести общую формулу для $(x+\varepsilon)^k$, $k \geq 1$:

$$(x+\varepsilon)^k = \sum_{i=0}^{k} x^i \varepsilon^{k-i} (C_k^i \bmod p).$$

Теперь, наконец, вычислим производную от функции

$$f(x) = \beta x^k, k \geq 1 \Rightarrow \frac{df(x)}{dx} = \beta \lim_{\varepsilon \to 0} \frac{(x+\varepsilon)^k - x^k}{\varepsilon} = \beta \lim_{\varepsilon \to 0} \frac{\left(\sum_{i=0}^{k} x^i \varepsilon^{k-i} (C_k^i \bmod p) \right) - x^k}{\varepsilon}.$$

Выделим из суммирования слагаемое x^k при $i = k$, и учтем, что $C_k^k = 1$ и $\varepsilon^0 = 1$. Тогда в итоге имеем:

$$\frac{df(x)}{dx} = \beta \lim_{\varepsilon \to 0} \frac{\left(x^k + \sum_{i=0}^{k-1} x^i \varepsilon^{k-i} (C_k^i \bmod p)\right) - x^k}{\varepsilon} = \beta \lim_{\varepsilon \to 0} \frac{\left(\sum_{i=0}^{k-1} x^i \varepsilon^{k-i} (C_k^i \bmod p)\right)}{\varepsilon}.$$

Заметим, что в числителе среди оставшихся слагаемых суммирования, только при $i = k - 1$, слагаемое $x^{k-1}\varepsilon$ содержит ε в первой степени, которое может сократиться с ε в знаменателе, а при $i \le k - 2$, слагаемые содержат ε в степени большей, чем 1, и оно не уничтожится вместе с ε в знаменателе, и после взятия предела слагаемые превратятся в нуль, поскольку будут содержать ε в ненулевой степени.

Тогда, $$\frac{df(x)}{dx} = \beta \lim_{\varepsilon \to 0} \frac{x^{k-1}\varepsilon(C_k^{k-1} \bmod p) + \left(\sum_{i=0}^{k-2} x^i \varepsilon^{k-i} (C_k^i \bmod p)\right)}{\varepsilon} = \beta(k \bmod p) x^{k-1}.$$

Таким образом, окончательно: $\frac{d}{dx}(\beta x^k) = \beta(k \bmod p) x^{k-1}$ при $k \ge 1$.

Теперь мы можем, вывести формулу для вычисления формальной производной от полинома $\Psi(x) = \Psi_{k-1} x^{k-1} + \ldots + \Psi_1 x + \Psi_0$, заданного над полем Галуа $GF(p^m)$:

$$\frac{d}{dx}\Psi(x) = \frac{d}{dx}\left(\sum_{i=0}^{k-1} \Psi_i x^i\right) = \frac{d}{dx}(\Psi_0) + \sum_{i=1}^{k-1} \Psi_i \frac{d}{dx}(x^i) = \sum_{i=1}^{k-1}\left(\left(\underbrace{\Psi_i \overbrace{(i \bmod p)}^{<R,\{+,\cdot\}>}}_{GF(p^m)}\right) \cdot x^{i-1}\right);$$

$$\Psi_i \in GF(p^m);\ i = 0 \ldots k-1. \qquad (3)$$

Особо отметим, что выражение $\Psi_i(i \bmod p)$ вычисляется как произведение элементов Ψ_i и λ в поле $GF(p^m)$, где элемент λ численно равен $i \bmod p$ в традиционной арифметике действительных чисел, но интерпретируется именно как элемент поля $GF(p^m)$.

В частном случае, если мы вычисляем формальную производную от полинома $\Psi(x) = \Psi_{k-1} x^{k-1} + \ldots + \Psi_1 x + \Psi_0$, заданного над полем Галуа $GF(2^m)$, мы имеем формулу:

$$\frac{d}{dx}\Psi(x) = \sum_{i=1}^{k-1}\left(\left(\underbrace{\Psi_i \overbrace{(i \bmod 2)}^{<R,\{+,\cdot\}>}}_{GF(2^m)}\right) x^{i-1}\right); \quad (4)$$

$$\Psi_i \in GF(2^m); \quad i = 0 \ldots k-1.$$

Заметим, что выражение $(i \bmod 2)$ приводит к тому, что в полях $GF(2^m)$, коэффициенты Ψ_i с четными индексами, стоящие при переменной x^{i-1} у полинома-производной, умножаются на нуль, и, соответственно, в полиноме-производной всегда отсутствуют слагаемые с нечетными степенями переменной x.

Пример. Формальная производная $\Psi(x) = 100x^4 + 218x^3 + 31x^2 + 3x + 51$, заданного над полем $GF(2^8)$:

$$\frac{d\Psi(x)}{dx} = \sum_{i=1}^{4} (\Psi_i (i \bmod 2) x^{i-1}) = \Psi_3 x^2 + \Psi_1 = 218x^2 + 3.$$

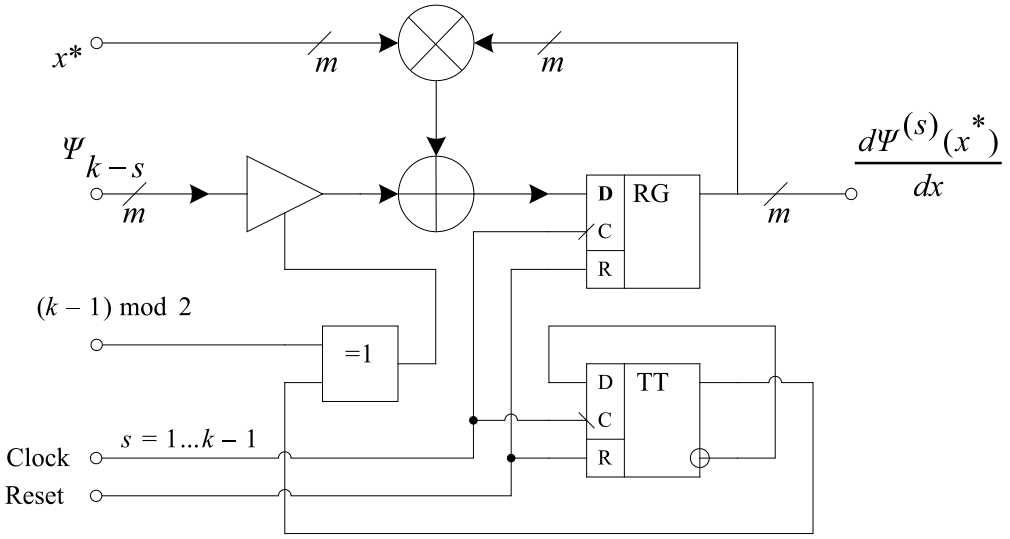
Вычисление значения формальной производной полинома при заданном аргументе над полем Галуа $GF(2^m)$. Выше мы выяснили, что формальная производная полинома $\Psi(x)$ над полем $GF(2^m)$ определяется как:

$$\frac{d\Psi(x)}{dx} = \sum_{i=1}^{k-1} (\Psi_i (i \bmod 2) x^{i-1}) = \Psi_1 + \Psi_3 x^2 + \ldots + \begin{cases} \Psi_{k-1} x^{k-2}, (k-1) \bmod 2 = 1 \\ \Psi_{k-2} x^{k-3}, (k-1) \bmod 2 = 0 \end{cases}.$$

Теперь формальную производную полинома можно переписать в следующем виде:

$$(\ldots(\Psi_{k-1}((k-1) \bmod 2)x + \Psi_{k-2}((k-2) \bmod 2))x + \ldots + \Psi_2 \cdot 0)x + \Psi_1 \cdot 1.$$

После этого мы можем применить следующую рекуррентную схему вычисления значения формальной производной полинома $\Psi(x)$ при заданном аргументе x^*:



*Функциональная схема аппаратной реализации последовательного вычислителя значения формальной производной dΨ(x)/dx при заданном аргументе x**

$$\begin{cases} d\Psi^{(s)}(x^*)/dx = (x^*)\cdot(d\Psi^{(s-1)}(x^*)/dx) + \Psi_{k-s}((k-s)\bmod 2) \\ d\Psi^{(0)}(x^*)/dx = 0; \quad s = 1\ldots k-1; \quad k \geq 0. \end{cases} \qquad (5)$$

После $k - 1$ итераций, результат вычислений на последней итерации $d\Psi^{(k-1)}(x^*)/dx$ и будет являться искомым значением формальной производной $d\Psi(x)/dx$ при заданном аргументе x^*. Заметим, что если $k = 0$ или $k = 1$, то в качестве результата возвращается нуль, а если $k = 2$ или $k = 3$, то в качестве результата возвращается Ψ_1.

Пример. Вычислим значение формальной производной полинома $\Psi(x) = 222x^5 + 29x^4 + 34x^3 + 183x^2 + 232x + 1$ заданного над полем $GF(2^8)$ при $x = 64$.

С одной стороны, формальная производная полинома имеет следующий вид: $d\Psi(x)/dx = 222x^4 + 34x^2 + 232$. Подставляя в нее $x = 64$, получаем значение формальной производной $222\cdot64^4 + 34\cdot64^2 + 232 = 70$.

С другой стороны, можно получить тот же результат, не прибегая к расчету полинома-производной $d\Psi(x)/dx$, используя лишь только исходный полином $\Psi(x)$ и применив к нему рекуррентную схему вычисления значения формальной производной: $((((222\cdot1)\cdot64 + 29\cdot0)\cdot64 + 34\cdot1)\cdot64 + + 183\cdot0)\cdot64 + 232\cdot1 = 70$.

Аппаратная реализация для поля Галуа $GF(2^m)$. Теперь рассмотрим аппаратную реализацию вычисления значения формальной производной $d\Psi(x)/dx$ при заданном аргументе x^* над полем Галуа $GF(2^m)$. Ниже на рисунке приведена предлагаемая автором функциональная схема последовательного вычислителя $d\Psi(x^*)/dx$.

Схема содержит, счетный D-триггер (TT), который перед началом вычислений сбрасывается, а затем с приходом каждого тактового сигнала меняет свое состояние на противоположное состояние. Выход триггера подключен к логическому элементу XOR, ко второму входу которого подается 1, если степень полинома $\Psi(x)$ является нечетной, или 0, если степень является четной. В итоге на управляющий вход коммутационной схемы $\triangleright$, поступает последовательность 101…101 при нечетной степени полинома $\Psi(x)$, и 01…101 при четной степени полинома. Таким образом, коммутационная схема всегда блокирует коэффициенты с четными индексами, коэффициенты с нечетными индексами, наоборот, всегда пропускает.

Также схема содержит m-разрядный регистр (RG) для хранения текущего результата вычислений. Кроме того, схема содержит m-разрядный сумматор $\oplus$ элементов Галуа $GF(2^m)$, который реализуется при помощи m двухвходовых логических элементов XOR. Наконец, схема также содержит m-разрядный умножитель $\otimes$ элементов Галуа $GF(2^m)$, который также может быть реализован при помощи m^2 двухвходовых элементов «И» и m многовходовых сумматоров по модулю 2. Аппаратная реализация быстрых сумматоров и умножителей элементов поля Галуа $GF(2^m)$ хорошо освещены в литературе [1, 2].

Изначально схема сбрасывается сигналом, подаваемым на вход *Reset*, тем самым и регистр и триггер сбрасываются в нулевое значение. При поступлении очередного тактового сигнала $s = 1 \ldots k - 1$ на один вход сумматора поступает результат умножения содержимого регистра на аргумент x^*, а на второй вход сумматора поступает очередной коэффициент Ψ_{k-s}, если $k - s$ является нечетным, и он складывается с результатом умножения, и сумма записывается в регистр, если же $k - s$ является четным, то на второй вход сумматора поступает нуль, и в итоге в регистр записывается результат умножения без изменений. В итоге после $k - 1$ тактов на выходе регистра мы получаем искомое значение формальной производной $d\Psi(x^*)/dx$ при заданном аргументе x^*.

Следует особо отметить, что регистр (RG) записывает информацию со своих входов по фронту тактовых импульсов, а триггер (TT) переключается по спаду тактовых импульсов. Такой двухтактный подход позволяет избегать переключения коммутационной схемы в момент записи информации в регистр.

Заключение

Таким образом, в рамках данной статьи рассматривается алгоритм вычисления формальной производной полинома над полем Галуа. В статье приведено определение полинома, формальной производной полинома и формула для ее вычисления над полем Галуа $GF(p^m)$.

Также в статье рассматривается алгоритм вычисления значения формальной производной полинома над полем Галуа $GF(2^m)$ при заданном аргументе и предлагаемая автором аппаратная реализация алгоритма.

Полученные результаты были использованы автором для разработки обучающей программы и лабораторных стендов для изучения студентами технических специальностей технологии кодирования информации при применении кодов Рида-Соломона.

Список литературы

1. Todd K. Moon. Error correcting coding: mathematical methods and algorithms. – Hoboken, New Jersey: John Wiley & Sons Inc., 2005.

2. Рахман П.А., Григорьева Т.В. Кодирование информации с применением кодов Рида-Соломона. – Уфа: Изд-во УГНТУ, 2015.

3. Рахман П.А., Шарипов М.И. Модель надежности двухузлового кластера приложений высокой готовности в системах управления предприятием // Экономика и менеджмент систем управления. – 2015. – Т. 17, № 3. – С. 85–102.

4. Рахман П.А., Шарипов М.И. Модели надежности каскадных дисковых массивов в системах управления предприятием // Экономика и менеджмент систем управления. – 2015. – Т. 17. № 3.1. – С. 155–168.

5. Рахман П.А. Коэффициент готовности трехуровневых локальных сетей передачи данных // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–3. – С. 463–466.

6. Рахман П.А. Показатели надежности восстанавливаемых систем с заданным порогом аварийного отключения // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–3. – С. 467–470.

7. Рахман П.А. Среднее время до потери данных двухдискового массива // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–4. – С. 603–607.

8. Рахман П.А. Коэффициент готовности системы обработки данных с основным и резервным узлами // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–4. – С. 608–611.

9. Рахман П.А. Модель надежности мажоритарной вычислительной системы на базе элементов с тремя состояниями // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 10–1. – С. 33–37.

10. Рахман П.А. Алгоритм выбора кратности исправляемых искажений для кодирования информации с применением кодов Рида-Соломона // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 10–2. – С. 208–212.