Recurrent algorithm for calculation of value of formal derivative of polynomial over Galois field and its hardware implementation

P A Rahman

Department of Automated Technological and Informational Systems, Ufa State Petroleum Technological University, 2, October Avenue, Sterlitamak, 453118, Russia

E-mail: pavelar@yandex.ru

Abstract. Within the scope of this scientific paper the formal derivative of the polynomial over the Galois field GF($p^m$) and a formula for calculation of the formal derivative are considered. Calculation of the formal derivative of the polynomial is used in the decoding algorithm of the information frames during application of the Reed-Solomon codes, which are widely used in the data transmission and storage systems with the data redundancy. The definition of the formal derivative of polynomial over the Galois field GF($p^m$) and the recurrent algorithm offered by the author for calculation of the value of the formal derivative of the polynomial over the Galois field GF($p^m$) with a given argument are also presented. Finally, the hardware implementation of the offered calculation scheme for the case of binary Galois field GF(2$^m$) is also given.

1. Introduction

Nowadays data transmission lines and data storage media are an essential part of human life. However, due to the imperfection of the wired and wireless transmission lines as well as magnetic and optical media, information can be corrupted or even completely destroyed [1, 2]. In particular, external noises affect data transmission lines as well as data storage media are sensitive to physical damages, which causes transmission failures and read errors.

To provide the tolerance of the data storage and transmission lines to the damages and noises there are various technologies of the data redundancy used with application of the special algorithms of coding on the basis of the error-control codes. It enables using of the redundant coding for errors correction. The Reed-Solomon codes [3, 4] are well-known as error-control codes.

However, it should be mentioned that the decoding algorithm of the information frames on application of the Reed-Solomon codes uses specialized procedures for calculation of the error syndrome, error-locators polynomial, error locators and error values, and all the algorithms deal with the polynomials over the Galois field [5, 6]. A specific place is held by the procedure for error values calculation based on the Forney method [7, 8], which uses the value of the formal derivative of polynomial over the Galois field with a given argument.

In last few years the author has provided scientific research in the field of reliability of computer systems and networks [9, 10]. In addition, he has obtained an algorithm for calculation of the value of the formal derivative of the polynomial over the Galois field GF($p^m$) with a given argument and hardware implementation of the offered calculation algorithm for the case of binary Galois field GF(2$^m$).
2. The formal derivative of polynomial over the Galois Field $\text{GF}(p^m)$

The polynomial given over the Galois Field $\text{GF}(p^m)$ can be presented in the following form:

$$\Psi(x) = \Psi_{k-1}x^{k-1} + \ldots + \Psi_1x + \Psi_0 = \sum_{i=0}^{k-1} \Psi_i x^i;$$

$$\Psi_i \in \text{GF}(p^m); \quad i = 0 \ldots k - 1. \quad (1)$$

It is possible to substitute the formal variable $x$ with a specific value, which is an element of the Galois field $\text{GF}(p^m)$, and calculate the value of polynomial. In the process of the calculation the rules of arithmetic of the Galois field $\text{GF}(p^m)$ are used.

As for the calculation of the derivative of the polynomial given over the Galois Field, a special approach is required because we cannot speak about infinitesimal values, considering that there are no such elements which we could use as infinitesimal in the Galois Field. However, nobody forbids us to speak about some formal infinitesimal value, which we will use only for the algebraic conversions.

Let us designate the formal infinitesimal value as $\varepsilon$.

At first we will discuss the formal derivative of polynomial given over the Galois field $\text{GF}(p^m)$ and then overview a particular case of the derivative over the binary Galois field $\text{GF}(2^m)$.

We can offer the following definition of the formal derivative of the function $f(x)$:

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

It should be noted that it is not necessary to find the explicit numerical value of the limit at the specific value of the argument $x$. The formula is given only for the analytical conversions with further exclusion of the value $\varepsilon$ from the resultant analytical expression of the derivative. Also it should be mentioned that the following properties of the formal derivative are true as they are in the traditional algebra:

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

Now let us consider the formal derivatives for the following simple functions:

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

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

Now, before obtaining the derivatives of power functions with higher orders, we should discuss in more detail the expression $(x + \varepsilon)^k, k \geq 1$.

In order to raise the expression $x + \varepsilon$ to the $k$-th degree, there is a simple formula in the traditional algebra: $(x + \varepsilon)^k = \sum_{i=0}^{k} C_k^i x^i \varepsilon^{k-i}$. The binomial coefficient $C_k^i = \frac{k!}{i!(k-i)!}$ represents the quantity of common (repeated) summands $x^i \varepsilon^{k-i}$ (for every $i = 0 \ldots k$), which are generated and then added, when we raise the expression $x + \varepsilon$ to the $k$-th degree.

Next, let us use the arithmetic feature of the Galois Fields $\text{GF}(p^m)$: $\frac{a + \ldots + a}{n \text{ summands}} = \frac{\lambda \cdot a}{\lambda = n \mod p}$. In this case $a = x^i \varepsilon^{k-i}$ and $n = C_k^i$, therefore $\frac{x^i \varepsilon^{k-i} + \ldots + x^i \varepsilon^{k-i}}{n \text{ summands}} = x^i \varepsilon^{k-i} \cdot (C_k^i \mod p).$
Considering all the above mentioned, we can obtain the following formula for the expression
\[(x + \varepsilon)^k, k \geq 1: \ (x + \varepsilon)^k = \sum_{i=0}^{k} x^i \varepsilon^{k-i} \cdot (C_k^i \mod p).\] Accordingly, we can obtain the derivative of the function \(f(x) = \beta x^k, k \geq 1 \Rightarrow \frac{df(x)}{dx} = \beta \cdot \lim_{\varepsilon \to 0} \frac{(x + \varepsilon)^k - x^k}{\varepsilon} = \beta \cdot \lim_{\varepsilon \to 0} \left\{ \sum_{i=0}^{k} x^i \varepsilon^{k-i} \cdot (C_k^i \mod p) \right\} - x^k.\)

Next, let us separately extract the summand \(x^k\) from the summation and consider that \(C_k^1 = 1\) and \(\varepsilon^0 = 1\). Accordingly, we obtain:
\[
\frac{df(x)}{dx} = \beta \cdot \lim_{\varepsilon \to 0} \left\{ \left( x^k + \sum_{i=1}^{k-1} x^i \varepsilon^{k-i} \cdot (C_k^i \mod p) \right) - x^k \right\} = \beta \cdot \lim_{\varepsilon \to 0} \left\{ \sum_{i=0}^{k-1} x^i \varepsilon^{k-i} \cdot (C_k^i \mod p) \right\}.
\]

Now, it should be noted that in the nominator among the all summands of the summation only the summand \(x^{k-1}\varepsilon\) contains \(\varepsilon\) in the first degree, which can be cancelled by \(\varepsilon\) in the denominator. Other summands contain \(\varepsilon\) in the degree \(> 1\), so they will not be cancelled with \(\varepsilon\) in the denominator. And, therefore, after applying of the limit, these summands will be turned into zero. Accordingly, we have:
\[
\frac{d}{dx} (\beta x^k) = \beta \cdot \lim_{\varepsilon \to 0} \left\{ \left( x^{k-1} \varepsilon \cdot (C_k^{k-1} \mod p) + \left( \sum_{i=1}^{k-2} x^i \varepsilon^{k-i} \cdot (C_k^i \mod p) \right) \right) \right\} = \beta \cdot (k \mod p) \cdot x^{k-1}, k \geq 1.
\]

Finally, taking into account all aforesaid, we obtain the following formula for calculation of the formal derivative of the polynomial \(\Psi(x) = \Psi_k x^{k-1} + \ldots + \Psi_1 x + \Psi_0\) over the Galois Field \(\text{GF}(p^m)\):
\[
\frac{d}{dx} \Psi(x) = \frac{d}{dx} \left( \sum_{i=0}^{k-1} \Psi_i x^i \right) = \frac{d}{dx} \Psi_0 + \sum_{i=1}^{k-1} \Psi_i \frac{d}{dx} (x^i) = \sum_{i=1}^{k-1} \left( \Psi_i \cdot \left( (i \mod p) \cdot \text{GF}(p^m) \right) \right) \cdot x^{i-1}; \quad (3)
\]
\[
\Psi_i \in \text{GF}(p^m); \quad i = 0 \ldots k-1.
\]

It should be mentioned that the expression \(\Psi_i \cdot (i \mod p)\) is calculated as a product of the elements \(\Psi_i\) and \(\lambda\) in the Galois field \(\text{GF}(p^m)\), where \(\lambda\) is numerically equal to the \((i \mod p)\) in the arithmetic of the real numbers \(\Re\), nevertheless it is interpreted as an element of the Galois Field \(\text{GF}(p^m)\).

In a particular case of calculation of the formal derivative of polynomial \(\Psi(x)\) over the binary Galois Field \(\text{GF}(2^m)\), we obtain the following formula by substituting \(p = 2\):
\[
\frac{d}{dx} \Psi(x) = \sum_{i=1}^{k-1} \left( \Psi_i \cdot \left( (i \mod 2) \cdot \text{GF}(2^m) \right) \right) \cdot x^{i-1} = \Psi_1 \cdot x^{k-1}, \text{ if } (k-1) \mod 2 = 1;
\]
\[
\Psi_1 \in \text{GF}(2^m); \quad i = 0 \ldots k-1.
\]

It should be noted that expression \((i \mod 2)\) results in multiplying by zero of the coefficients \(\Psi_i\) with even indexes at the variable \(x^{i-1}\) of the derivative polynomial in the binary Galois field \(\text{GF}(2^m)\), and, correspondingly, summands with odd degrees of the variable \(x\) are always absent in the resultant derivative polynomial in case of the binary Galois Field \(\text{GF}(2^m)\).

**Example.** The formal derivative of the polynomial \(\Psi(x) = 100x^4 + 218x^3 + 31x^2 + 3x + 51\) over the field \(\text{GF}(2^m)\) defined with the irreducible polynomial \(p(z) = z^8 + z^4 + z^3 + z^2 + 1\) is the following:
\[
\frac{d\Psi(x)}{dx} = \sum_{i=1}^{4} \left( \Psi_i \cdot (i \mod 2) \cdot x^{i-1} \right) = \Psi_3 x^2 + \Psi_1 = 218x^2 + 3.
\]
3. Recurrent scheme for calculation of the formal derivative value of polynomial over the Galois Field GF($p^m$) with a given argument

As we discussed above, the formula for calculation of the formal derivative of the polynomial $\Psi(x) = \Psi_{k-1}x^{k-1} + \ldots + \Psi_1x + \Psi_0$ over the Galois field GF($p^m$) is as follow:

$$\frac{d\Psi(x)}{dx} = \sum_{i=1}^{k-1} \Psi_i \cdot (i \text{ mod } p) \cdot x^{i-1} = \Psi_1 \cdot (1 \text{ mod } p) + \Psi_2 \cdot (2 \text{ mod } p) \cdot x + \ldots + \Psi_{k-1} \cdot ((k-1) \text{ mod } p) \cdot x^{k-2}.$$  

Now, we can convert the formula of the formal derivative of polynomial to the following form:

$$(\ldots(\Psi_{k-1} \cdot ((k-1) \text{ mod } p) \cdot x + \Psi_{k-2} \cdot ((k-2) \text{ mod } p)) \cdot x + \ldots + \Psi_2 \cdot (2 \text{ mod } p)) \cdot x + \Psi_1 \cdot (1 \text{ mod } p).$$

After that we can apply the following recurrent scheme offered by the author for calculation of the value of the formal derivative of the polynomial $\Psi(x)$ with given argument $x = a$:

$$\left\{ \begin{array}{l}
    k - 1 \geq 0; \quad d\Psi^{(0)}(a)/dx = 0; \\
    s = 1...k - 1; \quad d\Psi^{(s)}(a)/dx = a \cdot (d\Psi^{(s-1)}(a)/dx) + \Psi_{k-s} \cdot ((k - s) \text{ mod } p). \\
\end{array} \right.$$ (5)

After the $k - 1$ iterations, the result of calculation at the last iteration $s = k - 1$ will be $d\Psi^{(k-1)}(a)/dx$, which is equal to the value of the formal derivative $d\Psi(x)/dx$ with given argument $x = a$.

It should be noted that if $k - 1 = 0$, then the formal derivative is equal to zero.

Example. Let us calculate the value of the formal derivative of the polynomial $\Psi(x) = 222x^5 + 29x^4 + 34x^3 + 183x^2 + 232x + 1$ over the Galois field GF(29) defined with the irreducible polynomial $p(z) = z^8 + z^4 + z^3 + z^2 + 1$ with given argument $x = 64$.

By using the recurrent scheme for calculation of the value of the derivative at $x = 64$, we obtain: $(((222 \cdot (5 \text{ mod } 2)) \cdot 64 + 29 \cdot (4 \text{ mod } 2)) \cdot 64 + 34 \cdot (3 \text{ mod } 2)) \cdot 64 + 183 \cdot (2 \text{ mod } 2)) \cdot 64 + 232 \cdot (1 \text{ mod } 2) =$ $= (((222 \cdot 1) \cdot 64 + 29 \cdot 0) \cdot 64 + 34 \cdot 1) \cdot 64 + 183 \cdot 0) \cdot 64 + 232 \cdot 1 = 70$.

To check the result, obtained by the recurrent scheme, we can also derive the function of the formal derivative by using the formula (4): $d\Psi(x)/dx = 222x^4 + 34x^2 + 232$. Next, by substituting in the found function given argument $x = 64$, we obtain the value of the formal derivative: $222 \cdot 64^4 + 34 \cdot 64^2 + 232 = 70$. As we can see, the obtained value is the same as the value calculated by the recurrent scheme.

4. Hardware implementation of calculation of the value of the formal derivative of the polynomial over the binary Galois field GF($2^m$) with a given argument.

Let us overview the hardware implementation, offered by the author, of calculation of the value of the formal derivative $d\Psi(x)/dx$ over the binary Galois field GF($2^m$) with given argument $x = a$.

Figure 1 given below shows the offered functional diagram of the sequential calculator of the formal derivative value.

The calculator contains a counting trigger $TT$, which is reset before starting the calculation process. Further, on each clock impulse, trigger changes its state to the opposite one. The trigger output is connected to the «XOR» logic gate, to the second input of which is applied either «1», if the degree of the polynomial $\Psi(x)$ is odd, or «0», if the degree is even. Accordingly, during the calculation process the control input of the gate circuit $\triangleright$ receives either the binary sequence «101…101», if the degree of polynomial $\Psi(x)$ is odd, or «01…101», if the degree is even. Thus, gate circuit $\triangleright$ always blocks coefficients of the polynomial $\Psi(x)$ with even indexes, and passes coefficients with odd indexes. The gate circuit $\triangleright$ can be easily implemented with the use of the $m$ dual-input «AND» logic gates.
The sequential calculator also contains $m$-bit register RG to store the result of calculations of the formal derivative value $d\Psi^{(s)}(a)/dx$ on each iteration. Besides, the calculator contains $m$-bit adder $\oplus$ for elements of the Galois field GF($2^m$), which can also be implemented with the use of the $m$ dual-input «XOR» logic gates. Finally, the calculator contains $m$-bit multiplier $\otimes$ of the Galois Field GF($2^m$) elements, which can also be implemented with the use of the $m^2$ dual-input «AND» logic gates and $m$ multi-input adders modulo 2. The hardware implementation of the adders and multipliers for the elements of the binary Galois field GF($2^m$) is well covered in the literature [3, 4].

![Figure 1](image.jpg)

**Figure 1.** The functional diagram of the hardware implementation of the sequential calculator of the value of the formal derivative $d\Psi(x)/dx$ with given argument $x = a$ over the binary Galois field GF($2^m$).

Initially, the circuit is reset by the impulse given to the Reset input, thus both register RG and trigger TT are reset to zero. When the next clock impulses $s = 1\ldots k–1$ arrive, the first input of the adder receives the result of multiplication of the value of the register, stored at the previous clock impulse by the value of given argument $x = a$. If the $k–s$ is odd, then the second input of the adder receives the next coefficient $\Psi_{k–s}$, which is added to the result of multiplication, and the sum is stored into the register. If the $k–s$ is even, then the second input of the adder receives zero, and finally the result of multiplication is stored into the register without any changes.

As a result, after $k–1$ clock impulses at the output of the register RG, we obtain the resultant value of the formal derivative $d\Psi(x)/dx$ with given argument $x = a$.

It should be noted that the register RG stores information from its inputs in the front of the clock impulses, and trigger TT switches its state during the fall of the impulses. Such dual-tact synchronization let us to avoid switching of the gate circuit $\triangleright$ at the moment of storing the value into the register.

5. Conclusion

Thus, it is possible to calculate the formal derivative of the polynomial over the Galois field and the value of the formal derivative with the given argument. Within the scope of this scientific paper, the author offered an effective recurrent scheme for calculation of the value of the formal derivative of the polynomial over the Galois field GF($p^m$). Moreover, the hardware implementation of the recurrent calculation scheme for the case of binary Galois field GF($2^m$) is also presented by the author.

The obtained calculation scheme was used by the author for development of the specialized software for the scientific research in the field of the information coding technologies during application of the Reed-Solomon codes.
Acknowledgements
The author is grateful to professor I. I. Ladygin from Moscow Power Engineering Institute for scientific support and theoretical base in the field of error correcting coding.

References
[1] Olifer V G and Olifer N A 2016 Computer Networks (Saint-Petersburg: Piter)
[2] Thomasian A and Blaum M 2009 ACM Transactions on Storage 5(3) 7
[3] Wicker S B and Bhargava V K 1999 Reed-Solomon codes and their application (Wiley-IEEE Press)
[4] Clark G C and Cain J B 1981 Error-correction coding for digital communications (New York: Plenum Press)
[5] Lidl R and Niederreiter H 1983 Finite fields (Reading, MA: Addison-Wesley)
[6] Mullen G L and Panario D 2013 Handbook of Finite Fields (CRC Press)
[7] Forney G D 1966 Concatenated Codes (Cambridge, MA: MIT Press)
[8] Blahut R E 1983 Theory and Practice of Error Control Codes (Reading, MA: Addison-Wesley)
[9] Rahman P A and Bobkova E Yu 2017 J. Phys.: Conf. Ser. 803 012124
[10] Rahman P A and Bobkova E Yu 2017 J. Phys.: Conf. Ser. 803 012125