Development of specialized algorithm and hardware implementation for calculation of value of truncated convolution of polynomials over Galois field

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. This scientific paper deals with an algorithm for calculation of the truncated convolution of polynomials over the Galois field GF($p^n$). Calculation of the truncated convolution of polynomials is a key part of the information frame decoding algorithm on application of the Reed-Solomon codes in the noise protected data transmission and storage systems. Within the scientific paper, the definitions of polynomial and truncated convolution of the polynomials over the Galois field GF($p^n$) are also overviewed. Finally, the author offers a specialized algorithm for calculation of the value of the truncated convolution of polynomials over the Galois Field GF($p^n$) with a given argument. Its hardware implementation for the binary Galois field GF(2$^m$) is also presented.

1. Introduction

In present days the information play a key role in life of persons, in business processes of enterprises. However, the data transmission lines and data storage media are vulnerable to external noises and physical damages [1, 2], and this causes the corruption and loss of the information. Accordingly, various technologies of the information redundancy with application of the specialized algorithms of coding on the basis of error-correcting codes are used. For example, the Reed-Solomon codes [3, 4], which allows one to detect and correct data transmission and read errors using the redundant coding. However, the algorithms of information coding and decoding with the use of Reed-Solomon codes are quite complex and based on the specialized arithmetic of the Galois fields [5, 6]. In particular, the decoding algorithm consists of the blocks for calculation of the syndrome of corruption, error locators polynomial, error locators and error values, and all the blocks deal with the polynomials defined over the Galois field. The decoder of the Reed-Solomon codes also includes a block for calculation of the error values based on the Forney method [7, 8], which uses the truncated convolution of the polynomials defined over the Galois field. Accordingly, a special approach is required to develop an effective algorithm for calculation of the value of the truncated convolution of polynomials over the Galois field with a given argument.

Within the research work in the field of fault-tolerant computing systems and networks [9, 10], there was a scientific task of development of an efficient algorithm for calculation of value of the truncated convolution of the polynomials $\Psi(x)$ and $\xi(x)$ over the Galois Field GF($p^n$) with the given argument $x = a$. And its hardware implementation for the binary Galois field GF(2$^m$) was studied by the author.
2. Polynomials over the Galois Field $\text{GF}(p^n)$ and the truncated convolution of polynomials

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

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

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

It is possible to apply the algebraic operations using the polynomials. It is also possible to substitute formal variable $x$ with a specific value, which is an element of the Galois field $\text{GF}(p^n)$, and calculate the value of the polynomial. It should be noted, that rules of arithmetic of the Galois field $\text{GF}(p^n)$ are used for recalculation of the polynomial coefficients during the execution of the operations with the polynomials or calculation of value of the polynomial.

Now, let us overview several key operations on the polynomials.

**Sum of polynomials.** Addition of polynomials is performed by adding the corresponding coefficients, standing in front of the variable $x$ at same degrees, by using the rules of arithmetic of the Galois field $\text{GF}(p^n)$. The coefficients, which are absent in the polynomials, are considered as zero.

The sum of the polynomials $\Psi(x)$ and $\xi(x)$ can be obtained by the following formula:

$$\Psi(x) + \xi(x) = \sum_{q=0}^{\max(k-1,l-1)} x^q \left(\Psi_q + \xi_q\right);$$ \hspace{1cm} (2)

$\Psi_q, \xi_q \in \text{GF}(p^n); \quad \deg(\Psi(x)) = k - 1; \quad \deg(\xi(x)) = l - 1.$

**Example.** Let us calculate the sum of the polynomials $\Psi(x) = 49x^2 + 50x + 51$ and $\xi(x) = 19x^2 + 93x + 1$ over the Galois field $\text{GF}(2^8)$ defined with the irreducible polynomial $p(z) = z^8 + z^4 + z^3 + z^2 + 1$. By using the arithmetic of the field $\text{GF}(2^8)$ for calculation of sums of the coefficients, we obtain: $\Psi(x) + \xi(x) = (49 + 19)x^2 + (50 + 93)x + (51 + 1) = 34x^2 + 111x + 50$.

**Product of polynomials.** Multiplication of polynomials is performed by calculating the sum of the results of multiplication of one polynomial by each component of another polynomial. The degree of the resultant polynomial is equal to the sum of degrees of polynomials.

The product of the polynomials $\Psi(x)$ and $\xi(x)$ can be obtained by the following formula:

$$\Psi(x) \cdot \xi(x) = \left(\sum_{i=0}^{k-1} \Psi_i x^i\right) \cdot \left(\sum_{j=0}^{l-1} \xi_j x^j\right) = \sum_{i=0}^{k-2} x^i \left(\sum_{j=0}^{l-1} \Psi_i \cdot \xi_j x^j\right)$$ \hspace{1cm} (3)

$\Psi_i, \xi_j \in \text{GF}(p^n); \quad \deg(\Psi(x)) = k - 1; \quad \deg(\xi(x)) = l - 1.$

**Example.** Let us calculate the product of the polynomials $\Psi(x) = 49x^2 + 50x + 51$ and $\xi(x) = 19x^2 + 93x + 1$ over the Galois field $\text{GF}(2^8)$ defined with the irreducible polynomial $p(z) = z^8 + z^4 + z^3 + z^2 + 1$. By using the arithmetic of the field $\text{GF}(2^8)$ for addition and multiplication of the polynomial coefficients, we obtain the following resultant polynomial: $\Psi(x) \cdot \xi(x) = (49 \cdot 19)x^4 + (49 \cdot 93 + 50 \cdot 19)x^3 + (49 \cdot 1 + 50 \cdot 93 + 51 \cdot 19)x^2 + (50 \cdot 1 + 51 \cdot 93 + 51 \cdot 1) = 100x^4 + 218x^3 + 31x^2 + 3x + 51$.

Now, it should be noted that if we imply that any «absent» coefficients in polynomials, including the coefficients with indices, exceeding the degree of polynomials, are zero: $\forall i > k-1: \Psi_i = 0$ and $\forall j > l-1: \xi_j = 0$, then it is possible to convert internal summation with two indices $i$ and $j$ into the summation with the single index. To do this, at first, we should decrease the upper bounds for the both indices as follows: $i \leq q$ and $j \leq q$. It should be noted that in this case we will not lose anything, because
\(i \geq 0,\ j \geq 0\) and \(i + j = q\). The cases when \(i > q\) (at \(k - 1 > q\)) or \(j > q\) (at \(l - 1 > q\)) will never pass the condition \(i + j = q\), as far as \(i \geq 0\) and \(j \geq 0\). Moreover, the coefficients \(\Psi_i = 0\) for \(i > k - 1\) (when \(q > k - 1\)) and the coefficients \(\xi_j = 0\) for \(j > l - 1\) (when \(q > l - 1\)). Thus, we obtain three following conditions for the indices \(i\) and \(j\): \(i + j = q, 0 \leq i \leq q\) and \(0 \leq j \leq q\).

Now, because the both indices \(i\) and \(j\) are non-negative and bounded by the same upper limit \(q\), and moreover, the sum of the indices is also equal to the same upper limit \(q\), we can express one of the indices through another one as: \(j = q - i\) and use summation with the single index \(i\), \(0 \leq i \leq q\).

Finally, we obtain the following «optimized» formula for the product of the polynomials:

\[
\Psi(x) \cdot \xi(x) = \sum_{q=0}^{k-1} x^q \Psi_i \cdot \xi_{q-i};
\]

\(\Psi, \xi_j \in \text{GF}(p^m);\) \(\deg(\Psi(x)) = k - 1;\) \(\deg(\xi(x)) = l - 1.\)

**Truncated convolution of polynomials.** The truncated convolution \(\Psi(x) \circ \xi(x)\) of the polynomials \(\Psi(x)\) and \(\xi(x)\) is defined as the polynomial remainder of the product of these polynomials modulo with the given monomial \(x^r\). In other words, \(\Psi(x) \circ \xi(x) = (\Psi(x) \xi(x)) \mod x^r\).

Obviously, the truncated convolution can be calculated as the product of the polynomials \(\Psi(x)\) and \(\xi(x)\) with subsequent deletion of the components, for which the degree of the variable \(x\) is greater than or equal to \(r\). Therefore, the truncated convolution \(\Psi(x) \circ \xi(x)\) of the polynomials \(\Psi(x)\) and \(\xi(x)\) can be calculated by the following «optimized» formula:

\[
\Psi(x) \circ \xi(x) = (\Psi(x) \cdot \xi(x)) \mod x^r = \sum_{q=0}^{r-1} x^q \sum_{i=0}^{q} \Psi_i \cdot \xi_{q-i};
\]

\(\Psi, \xi_j \in \text{GF}(p^m);\) \(\deg(\Psi(x)) = k - 1;\) \(\deg(\xi(x)) = l - 1.\)

**Example.** Let us calculate the truncated convolution of the polynomials \(\Psi(x) = 49x^2 + 50x + 51\) and \(\xi(x) = 19x^2 + 93x + 1\) modulo the given monomial \(x^5\) over the field \(\text{GF}(2^8)\) defined with the irreducible polynomial \(p(z) = z^8 + z^4 + z^3 + z^2 + 1\). According to the formula 5 we obtain

\[
\Psi(x) \circ \xi(x) = \sum_{q=0}^{1} x^q \sum_{i=0}^{q} \Psi_i \cdot \xi_{q-i} = \Psi_0 \cdot \xi_0 + (\Psi_1 \cdot \xi_1 + \Psi_1 \cdot \xi_5) x = 51 \cdot 1 + (51 \cdot 93 + 50 \cdot 1) x = 3x + 51.
\]

3. Calculation of value of the truncated convolution of polynomials over the Galois Field \(\text{GF}(p^m)\) with the given argument

As we discussed above, the formula for calculation of the truncated convolution of the polynomials \(\Psi(x)\) and \(\xi(x)\) over the Galois field \(\text{GF}(p^m)\) modulo with the given monomial \(x^r\) is the following:

\[
\Omega(x) = \Psi(x) \circ \xi(x) = \sum_{q=0}^{r-1} x^q \sum_{i=0}^{q} \Psi_i \cdot \xi_{q-i};
\]

It is easy to see that the calculation of the convolution polynomial using the formula obtained above requires \(r(r + 1)/2 - r^2/2\) iterations of calculations in double summation.

Therefore, let us perform the following conversion of the calculation formula:
\[ \Omega(x) = \sum_{q=0}^{r-1} x^q \sum_{i=0}^{q} \Psi_i x^{-\sigma} = \Psi_0 x^0 + (\Psi_1 x^1 + \Psi_2 x^0) x + \ldots + (\Psi_{r-1} x + \Psi_r x^0) x^{r-1} = \\
= \Psi_0 (\xi_0 x + \ldots + \xi_{r-1} x^{r-1}) + \Psi_1 x (\xi_0 x + \ldots + \xi_{r-2} x^{r-2}) + \ldots + \Psi_{r-1} x^{r-1} x_0 = \sum_{i=0}^{r-1} \Psi_i x^i \left( \sum_{j=0}^{r-i} \xi_j x^j \right). \]

Next, we obtain the following recurrent form of the calculation formula:

\[ \Omega(x) = \left( \ldots (\Psi_{r-2} x^2 \xi_0 x + \Psi_{r-1} (\xi_0 x + \xi_1 x)) x + \ldots + \Psi_1 \left( \sum_{j=0}^{r-2} \xi_j x^j \right) \right) x + \Psi_0 \left( \sum_{j=0}^{r-1} \xi_j x^j \right). \]

Now, we can introduce the following recurrent scheme for calculation of the value of the convolution polynomial with the given argument \( x = a \):

\[ \begin{cases} 
\Omega^{(0)}(a) = 0; \\
\Omega^{(s)}(a) = a \cdot \Omega^{(s-1)}(a) + \sum_{j=0}^{s-1} a^j \cdot \xi_j. 
\end{cases} \]

Although the recurrent scheme requires \( r \) iterations of calculations, obviously, with each iteration it is necessary to calculate the sum \( \sum_{j=0}^{r-1} a^j \cdot \xi_j \), and the number of summands is increased with each iteration \( s = 1 \ldots r \). On the other hand, we can calculate the sum for each iteration \( s \) just by adding the next summand to the sum, calculated at the previous iteration \( s - 1 \). Accordingly, let us designate the sum, obtained on the iteration \( s \), as \( \xi^{(s)}(a) = \sum_{j=0}^{s-1} a^j \cdot \xi_j \). After that, we can introduce the following recurrent formula \( \xi^{(s)}(a) = \xi^{(s-1)}(a) + a^{s-1} \cdot \xi_{s-1} \), where \( \xi^{(0)}(a) = 0 \). Moreover, in each iteration we can, first, calculate \( \xi^{(s)}(a) \) by using \( \xi^{(s-1)}(a) \) and then calculate \( \Omega^{(s)}(a) \) by using the \( \Omega^{(s-1)}(a) \) and \( \xi^{(s)}(a) \). At last, the \( s \)-th power \( a^s \) of the given argument \( x = a \) can also be calculated for each iteration by using the recurrent formula \( a^s = a^{s-1} \cdot a \), where \( a^0 = 1 \).

Taking into account all of the previously mentioned, finally, we obtain an «optimized» recurrent scheme for calculation of the value of the convolution polynomial \( \Omega(x) \) with the given argument \( x = a \):

\[ \begin{cases} 
r \geq 0; \\
\Omega^{(0)}(a) = 0; \\
\xi^{(0)}(a) = 0; \\
\Omega^{(s)}(a) = a \cdot \Omega^{(s-1)}(a) + \sum_{j=0}^{s-1} a^j \cdot \xi_j; \\
\xi^{(s)}(a) = \xi^{(s-1)}(a) + a^{s-1} \cdot \xi_{s-1}. 
\end{cases} \]

After the \( r \) iterations the result of calculation on the last iteration \( s = r \) will be \( \Omega^{(r)}(a) \), equal to the value of the truncated convolution \( \Omega(x) \) with the given argument \( x = a \).

**Example.** Let us calculate the value of the truncated convolution of the polynomials \( \Psi(x) = 49x^2 + 50x + 51 \) and \( \xi(x) = 19x^2 + 93x + 1 \) over the Galois field GF(\( 2^8 \)), defined with the
irreducible polynomial $p(z) = z^8 + z^4 + z^3 + z^2 + 1$, modulo polynomial $x^2$ (accordingly, $r = 2$) with the given argument $x = 64$ by using the obtained optimized recurrent scheme.

The initial values for the variables used in the recurrent calculation scheme are the following:

$$
\Omega^{(0)}(a) = 0; \quad \xi^{(0)}(a) = 0; \quad a^0 = 1.
$$

Using the iteration $s = 1$, we perform the following calculations:

$$
\begin{align*}
\xi^{(1)}(a) &= \xi^{(0)}(a) + a^0 \cdot \xi_0 = 0 + 1 \cdot 1 = 1; \\
\Omega^{(1)}(a) &= a \cdot \Omega^{(0)}(a) + \Psi_1 \cdot \xi^{(1)}(a) = 64 \cdot 0 + 50 \cdot 1 = 50; \\
a^1 &= a^0 \cdot a = 1 \cdot 64 = 64.
\end{align*}
$$

Finally, using the iteration $s = 2$ (last iteration), we obtain:

$$
\begin{align*}
\xi^{(2)}(a) &= \xi^{(1)}(a) + a^1 \cdot \xi_1 = 1 + 64 \cdot 93 = 223; \\
\Omega^{(2)}(a) &= a \cdot \Omega^{(1)}(a) + \Psi_0 \cdot \xi^{(2)}(a) = 64 \cdot 50 + 51 \cdot 223 = 243; \\
a^2 &= a^1 \cdot a = 64 \cdot 64 = 205.
\end{align*}
$$

Accordingly, the resultant value of the truncated convolution of polynomials is $\Omega^{(2)}(a) = 243$.

To check the obtained value, we can substitute the given argument $x = 64$ into the convolution $(\Psi(x) \cdot \xi(x)) \mod x^2 = 3x + 51$ of the polynomials $\Psi(x) = 49x^2 + 50x + 51$ and $\xi(x) = 19x^2 + 93x + 1$, obtained in the previously discussed example. As the result we obtain the value: $3 \cdot 64 + 51 = 243$. Thus, obviously, the result of the calculations by the recurrent scheme is correct.

4. Hardware implementation of calculation of the value of the truncated convolution of polynomials over the binary Galois Field GF($2^m$) with the given argument

Let us overview the author’s hardware implementation of calculation of the value of the convolution polynomial $\Omega(x)$ over the binary Galois field GF($2^m$) with the given argument $x = a$.

Figure 1 given below shows the offered functional diagram of the sequential calculator of the value of the convolution polynomial.

The diagram contains three $m$-bit registers RG1, RG2 and RG3, one logic inverter, two adders $\oplus$ of elements of the Galois field GF($2^m$) and four multipliers $\otimes$ of elements of the Galois field GF($2^m$).

The register RG1 is used to store the calculated using each iteration value $a^s$ (s-th power of the given argument $a$), the register RG2 – to store the value $\xi^{(s)}(a)$, and, finally, the register RG3 – to store the value $\Omega^{(s)}(a)$, which are used in the optimized recurrent calculation scheme discussed above.

The adder $\oplus$ of the elements of the Galois Field GF($2^m$) can be implemented by using the $m$ dual-input «XOR» logic gates. The multiplier $\otimes$ of the elements of the Galois Field GF ($2^m$) can be implemented by using the $m^2$ dual-input «AND» gates and $m$ multi-input adders modulo 2. The hardware implementation of the adders and multipliers of the elements of the Galois Field GF($2^m$) is well covered in the literature [3, 4].

Now, let us briefly overview the calculation process in the circuit.

Initially, the Reset input receives an impulse, and the register RG1 is set to the value (0...01), which corresponds to the element «1» of the Galois Field GF($2^m$), and the registers RG2 and RG3 are reset to the value (0...00), which corresponds to the element «0» of the Galois Field GF($2^m$).

After that the Clock input receives impulses, and the circuit provides the calculations in accordance with the iterations $s = 1...r$ of the recurrent calculation scheme.
Figure 1. The functional diagram of the hardware implementation of the sequential calculator of the value of the convolution polynomial $\Omega(x)$ for the given argument $x = a$ over the binary Galois field $GF(2^m)$.

It should be noted that the synchronization input (C) of the register RG2 is connected directly to the Clock input, and, accordingly, the register RG2 stores information from its inputs (D) on the front of the clock impulse. The registers RG1 and RG3 store information from their inputs (D) during the fall of the clock impulse, as far as their synchronization inputs (C) are connected to the Clock via the inverter. Such dual-tact synchronization allows us on each iteration $s = 1 \ldots r$, first, calculate and store $\xi^{(s)}(a) = \xi^{(s-1)}(a) + a^{s-1} \cdot \xi_{r-1}$ value into the register RG2, and only after that - calculate and store $\Omega^{(s)}(a) = a \cdot \Omega^{(s-1)}(a) + \Psi_{r-s} \cdot \xi^{(s)}(a)$ value into the register RG3. Also it is possible to calculate and store the value of the next power of the argument $a^r = a^{r-1} \cdot a$ into the register RG1. Thus, dual-tact synchronization provides a correct sequence of the calculations in each iteration.

As a result, after $r$ clock impulses at the output of the register RG3, we obtain the resultant value of the convolution polynomial $\Omega(x) = (\Psi(x) \cdot \xi(x)) \mod(x')$ with the given argument $x = a$.

5. Conclusion

Thus, within this scientific paper the algorithm for calculation of the truncated convolution of polynomials over the Galois field $GF(p^m)$ is discussed.

In the work the definitions of the polynomial and truncated convolution of polynomials over the Galois field $GF(p^m)$ are also overviewed.

Finally, the author offered specialized recurrent scheme for calculation of value of the truncated convolution of polynomials over the Galois Field $GF(p^m)$ with the given argument. Its hardware implementation for the binary Galois field $GF(2^m)$ is also presented.

The author used the obtained results for development of the specialized training software and laboratory works to study the information coding technologies during application of the Reed-Solomon codes for the students of technical specialties.
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] Khurshidov A S 2001 The Essential Guide to Computer Data Storage (Prentice Hall)
[2] Tanenbaum A S and Wetherall D J 2010 Computer Networks (New Jersey: Prentice Hall PTR)
[3] Moon T K 2005 Error correcting coding: mathematical methods and algorithms (New Jersey: John Wiley & Sons Inc.)
[4] Hamming R W 1986 Coding and Information Theory (Englewood Cliffs, NJ: Prentice-Hall)
[5] StichTenoth H 1986 Algebraic Function Fields and Codes (Berlin: Springer-Verlag)
[6] Lidl R and Niederreiter H 1983 Finite fields (Reading, MA: Addison-Wesley)
[7] Blahut R E 1983 Theory and Practice of Error Control Codes (Reading, MA: Addison-Wesley)
[8] Forney G D 1966 Concatenated Codes (Cambridge, MA: MIT Press)
[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