On the Systematic Encoding of p-ary Reed-Solomon Codes with an Arbitrary Prime p

Zhaneta Savova
Computer Systems and Technologies Department
Faculty of Artillery, Air Defense and Communication and Information Systems,
National Military University
Shumen, Bulgaria
zh.savova@mail.bg

Antoniya Tasheva
Computer Systems Department
Faculty of Computer Systems and Technologies,
Technical University of Sofia
Sofia, Bulgaria
antoniya.tasheva@yahoo.com

Rosen Bogdanov
Communication Networks and Systems Department
Faculty of Artillery, Air Defense and Communication and Information Systems,
National Military University
Shumen, Bulgaria
r61@abv.bg

***Abstract*— The Reed-Solomon (RS) codes, which are a subset of error-correcting codes, are currently employed in a number of significant applications. The most notable of these are data recovery in storage systems, barcodes in management and advertising systems, and communication systems and networks. In the present technological era, RS codes over Galois fields GF(2^m) are frequently employed in the aforementioned applications, with GF(2^8) being the most prevalent. This enables the representation of all 256 values of a byte as a polynomial with eight binary coefficients over GF(2^8). The motivation for verifying and generalizing the idea of systematic encoding of RS codes in a field with a base *p* other than 2 stems from the fact that the mathematical dependencies in arbitrary field GF(p^m) are not only valid for GF(2^m), but also have wider applicability. Consequently, the article establishes the specific features of the systematic encoding of *p*-ary Reed-Solomon codes and LFSR-based systematic encoder, conceptualizing them as a class of cyclic codes over any field GF(p^m) whose base is a prime *p* other than 2.**

Keywords—Reed-Solomon Codes, Systematic Encoding, LFSR-based pRS Encoder, Multilevel Sequences, PAM-3, PAM-5.

I. Introduction

The Reed-Solomon (RS) codes are non-binary cyclic (*n, k*) error correction codes of length *n* and size *k*, initially proposed by Irving Reed and Gustav Solomon in 1960 [1]. By incorporating additional symbols to check the data, the RS code is capable of detecting any combination of up to a maximum of $t = (n - k)/2$ erroneous symbols, correcting up to $\lfloor t/2 \rfloor$ symbols, or detecting and correcting combinations of errors and erasures. Here $\lfloor x \rfloor$ denotes the largest integer less than or equal to *x*. Furthermore, RS codes are well-suited for correcting burst errors.

The Reed-Solomon codes are a widely utilized method of error correction in practice, as evidenced by their numerous applications. The most significant of these are data recovery in storage systems [2], [3], [4] including hard drives, CDs, DVDs, BigTable, Google's GFS, and RAID 6, as well as in barcodes in management and advertising systems [5], such as QR Code, PDF-417, MaxiCode, Datamatrix, and Aztec Code.

Furthermore, RS codes are predominantly employed in communication systems. These include software-defined wireless networks [6], [7] telecommunication systems and networks [8], mobile wireless systems [9], digital video broadcasting (DVB-T) [10] and video-watermarking applications [11]. Furthermore, the applications of Reed-Solomon codes extend to code division multiple access in mobile phones [12], communication systems based on electrical power transmission lines [13], [14], ultra-wideband communications [15], reconfigurable electronic intelligent devices for smart energy [16], 802.11ad MIMO-OFDM transceivers [17], and other related fields.

The RS codes implementation can effectively enhance the characteristics of the communication system. For example, the performance of a RS-coded spectral-amplitude-encoding optical-code-division-multiplexing system [18] is evaluated in the presence of optical beating interference (OBI). It is observed that RS coding can effectively mitigate the impact of OBI, thereby enabling a doubling of the number of active users.

In the paper [19], the performance of the Satellite Downlink Transmitter that employs a Wavelet-based Filtered Multi-Tone (WFMT) modulation is examined. It is

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol2.8568

demonstrated that error correction block codes, such as RS codes, can effectively enhance the Peak-to-Average Power Ratio characteristics of the communication system.

In their study [20], the authors examined the Bit Error Rate (BER) performance of tree models utilizing Hamming, BCH, and RS codes. Their findings indicate that the Reed-Solomon model demonstrates enhanced BER performance compared to the Hamming and BCH models, which demonstrate comparable outcomes.

In practical applications, Reed-Solomon codes use message symbols typically represented using a Galois Field with a base 2, i.e. GF(2^m). For example, the authors of the paper [21] propose a reconfigurable FEC system based on the Reed-Solomon codec for DVB and WiMax networks. The FEC system is based on the parameterisation approach of the RS encoder-decoder architecture, which is a key solution for software defined radio systems. The FPGA implementation of the advanced RS encoder-decoder uses only polynomials over GF(2^8).

The paper [22] presents a novel method for bit level shortening of a Reed Solomon (RS) code, resulting in only shortened BCH subcodes. Using a specific basis, an RS code over GF(2^m) is mapped to a binary image containing *m* concatenated BCH sub-codewords and some glue-vector code words.

To mitigate the effects of fading and inter symbol interference when WiMAX is used in the 2-11 GHz range, the time diversity by incorporating an outer Reed-Solomon block code concatenated with an inner convolutional code is used in WiMAX physical layer. The systematic RS(255, 239) which Galois Field elements are from GF(2^8) is applied [23].

In today's high-speed communication systems, multi-level signals and sequences are becoming a distinctive feature. New methods are being used for in-vehicle networking, such as the 4D PAM-7 [24], PWAM signaling scheme [25], and Automotive Ethernet [26]. These methods use four-dimensional five-level pulse amplitude modulation (4D-PAM-5), PAM-7 and PAM-3 symbols to improve data transfer rates compared to wired infrastructures.

In numerous research papers, the theoretical encoding of Reed-Solomon codes is presented over an arbitrary field GF(q) of q elements, where q is a power of a prime number. Despite this theoretical representation, the real applications and examples elucidating Reed-Solomon codes are over Galois fields GF(2^m) of base 2.

It is therefore evident that in order to guarantee the error correction features of new multi-level sequences, it is necessary to employ advanced methods of RS encoding that are capable of handling not only binary but also non-binary symbols over GF(p^n) of base which is arbitrary prime p.

The objective of this article is to present a comprehensive analysis of systematic p-ary Reed-Solomon (pRS) encoding as a family of codes over an arbitrary Galois field GF(p^m). Additionally, it elucidates the specific features of the encoding process and the encoding scheme, based on Linear Feedback Shift Register (LFSR), when a pRS code is employed over a finite field of base prime p, other than 2.

II. Materials and methods

This section provides a concise overview of the problem formulation and the preliminary concepts employed in the solution.

In order to guarantee the error-correcting capabilities of multi-level sequences, such as PAM signals comprising 3, 5 or 7 levels, it is vital to determine the specific characteristics of the systematic pRS encoding and the circuits responsible for pRS encoding in instances where the message symbols are situated within a Galois field with an odd prime base.

First, we provide a concise overview of the two primary methods for representing Reed-Solomon codes and their special properties. We then present a brief introduction to the systematic RS encoding procedure. In the subsequent subsection, we examine the specific features of GF(p^m) calculations with an odd prime base p.

A. A Representation of Reed-Solomon Codes and Their Properties

There are two methods of representing Reed-Solomon codes. The first views the code word as a sequence of values, which was proposed in the original work of Reed and Solomon in 1960 [1]. Reed and Solomon propose a code E by which each k-tuple (a_0, a_1, …, a_{k-1}) of the field K of degree n over a field Z_2 of 2 elements is matched by a 2^n-tuple ($m(0)$, $m(\alpha)$, $m(\alpha^2)$, …, $m(1)$) of K. Here $m(x)$ is a polynomial of degree $k - 1$

$$m(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}, \qquad (1)$$

where a_i, $\in K$, $k < 2^n$, and α is the primitive n–th unit root in K. The k-tuple is the input message and the 2^n-tuple is the transmitted message. The authors prove that, depending on whether k is an even or odd number, this code corrects either $(2^n - k)/2$ or $(2^n - k - 1)/2$ symbols.

The second method considers Reed-Solomon codes to be Bose-Chaudhuri-Hocquenghem (BCH) codes [27], in which the code word is represented as a sequence of coefficients. In this approach, instead of sending all the values of the message polynomial $m(x)$ (1), the transmitter calculates another polynomial $s(x)$

$$s(x) = m(x)g(x). \qquad (2)$$

of degree at most n - 1 and sends the n coefficients of this polynomial.

The generating polynomial $g(x)$ is defined as the polynomial whose roots are the elements α, α^2, …, α^{n-k} of the field K

$$\begin{aligned} g(x) = (x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{n-k}) = \\ = g_0 + g_1x + \cdots \\ + g_{n-k-1}x^{n-k-1} + x^{n-k}. \end{aligned} \qquad (3)$$

Reed-Solomon codes are Maximum Distance Separable (MDS) codes because their minimum Hamming distance is

$$d_{min} = n - k + 1. \tag{4}$$

Because RS codes are MDS codes, they have special properties. For example, any *k* positions in the block can be used as information and its weight distribution can be easily determined.

The RS code can correct up to a maximum of $t = (n - k)/2$ erroneous symbols or can correct up to $2t$ erasures if the error locations are known in advance, as defined by the erasure term. Moreover, the RS code is able to correct any combination of errors and erasures, provided they fall within its correction capability

$$2e + er \leq n - k, \tag{5}$$

where *e* is the number of errors and *er* is the number of erasures.

Given that GF(2^m) is a vector space of size *m* over GF(2), it follows that each element of GF(2^m) can be represented by *m* bits, with the element 0 being represented by a zero binary m-tuple (0, 0, 0, ..., 0). In this representation, the (*n, k*) RS code over GF(2^m) is transformed into a binary (*mn, mk*) code with a minimum distance d'_{min} that is at least as large as the minimum distance d_{min} of the RS code. Binary (*mn, mk*) codes with equivalent properties are useful for their ability to correct bursts of errors. Each burst of errors, defined as $(t - 1)\, m + 1$ or fewer consecutive bits, will appear as at most *t* errors in the GF(2^m) symbols. As a result, RS codes are ideal for correcting bursts of errors [28] due to their largest possible d_{min}.

B. Systematic Encoding of Reed-Solomon Codes

The RS encoding (3), which considers RS codes as a subset of BCH codes, is a well-established approach for all linear codes. However, it does not satisfy the requirement of the systematic encoding procedure that the message is contained unchanged in the transmitted codeword.

To satisfy the requirement of systematic encoding, the message polynomial *m*(*x*) is relocated to the *k* most significant positions of the codeword register. The parity polynomial *t*(*x*) is then concatenated to the right of *m*(*x*) in the *n - k* least significant positions. This rule, which is applicable to any RS code with base 2, can be found in any work on the subject and any textbook on data transmission or information theory.

The mathematical description of these operations is as follows [29]. The rightmost shift of the polynomial *m*(*x*) is represented by multiplying *m*(*x*) by x^{n-k}. The product $m(x).x^{n-k}$ is then divided by the generating polynomial *g*(*x*), yielding the quotient *q*(*x*) and the remainder *t*(*x*).

$$x^{n-k}.m(x) = \mathrm{g}(x).q(x) + t(x). \tag{6}$$

In a manner analogous to the classical binary encoding procedure, the remainder is the parity polynomial, which, as indicated in (5), can be represented as follows:

$$t(x) = x^{n-k}.m(x) \bmod \mathrm{g}(x). \tag{7}$$

The resulting polynomial of the codeword in binary case is

$$c(x) = x^{n-k}.m(x) + t(x). \tag{8}$$

For an example of systematic encoding for a (7, 3) RS code one can see [29].

C. Some Specific Features of GF(p^n) Calculations with Odd Base p

The field GF(*q*) can be considered an extension of the Galois field GF(*p*) with a power of *m*, provided that the order of GF(*q*) can be expressed as a power of the prime *p* ($q = p^m$), where *m* is a positive integer with $m \geq 2$. In such cases, the notation Extended Galois Field GF(p^m) is used.

The elements of the field GF(p^m) are polynomials of degree $m - 1$, which belong to the ring GF(*p*)[*x*] and have coefficients in GF(*p*)

$$GF(p^m) = \{a_{m-1}\alpha^{m-1} + \cdots + a_1\alpha + a_0 \mid a_i \in GF(p)\}. \tag{9}$$

The arithmetic in GF(*p*) is computed using integer arithmetic modulo *p* and the arithmetic in GF(p^m) involves polynomial arithmetic modulo the irreducible polynomial *p*(*x*) over GF(*p*) with degree *m* [30].

In their analysis, the authors in [31] identify two specific features for p-ary Reed-Solomon code construction by considering them as a family of codes over any Galois field GF(p^m) whose base is a prime *p* other than 2. These features are as follows:

Specific Feature 1. Every nonzero element a has an additive inverse element $-a = p - a$ in a Galois Field GF(*p*).

Specific Feature 2. The subtraction operations must be performed due to the difference between addition and subtraction operations for fields GF(p^n) with base $p > 2$.

III. Results and discussion

This section looks at the specific features of p-ary Reed-Solomon encoding and an LFSR-based circuit for encoding non-binary symbols in GF(p^m) with an arbitrary prime p. Examples of systematic encoding and polynomial division are also given, as well as an LFSR-based encoder for an (8, 4) 3RS code over the field GF(3^2).

A. Systematic Encoding of p-ary Reed-Solomon Codes

As demonstrated in Section 2.2, in the systematic encoding, the codeword is constituted of *k* message symbols, followed by $n - k$ parity symbols.

In this section, the codeword polynomial will be considered from a slightly different perspective, taking into account the differences in addition and subtraction operations in GF(p^m). Let $f(x)$ be a polynomial in which the leading k coefficients represent the message symbols, and the coefficients in front of powers less than $n - k$ are set to zero, i.e.

$$\begin{aligned} f(x) = a_{k-1}x^{n-1} + \cdots a_1 x^{n-k+1} + a_0 x^{n-k} \\ = x^{n-k}.m(x). \end{aligned} \tag{10}$$

In accordance with Euclidian's division algorithm

$$f(x) = g(x).q(x) + t(x), \tag{11}$$

where the degree of the remainder $t(x)$ is less than the degree $n - k$ of the generating polynomial $g(x)$.

Hence

$$f(x) - t(x) = g(x).q(x), \tag{12}$$

and thus, the codeword is $c(x) = f(x) - t(x)$.

In the aforementioned context, in order to obtain the resulting polynomial of the codeword $c(x)$ from the encoding procedure with pRS codes with a base $p \neq 2$, it is necessary to subtract the remainder $t(x)$ from the product $x^{n-k}.m(x)$, i.e.

$$c(x) = x^{n-k}.m(x) - t(x). \tag{13}$$

This is the third significant distinction in encoding a pRS code with an arbitrary base $p \neq 2$. Here, the check symbols in the codeword $c(x)$ are the additive inverse of the coefficients of the remainder $t(x)$. The following specific feature will present the fundamental rule that governs systematic encoding procedure in pRS codes.

Specific Feature 3. In systematic encoding of pRS codes, the message polynomial $m(x)$ is shifted to the k most significant positions of the codeword register and then the additive inverse of each coefficient of the parity polynomial $t(x)$ is written to the $n - k$ lowest positions of the register.

Example 1. Consider an (8, 4) 3RS code over the field GF(3^2) with the generating polynomial $g(x)$

$g(x) = 12 + 20x + 12x^2 + 11x^3 + 10x^4 =$

$= \alpha^2 + \alpha^4 x + \alpha^2 x^2 + \alpha^7 x^3 + \alpha^0 x^4$.

Let the input be 10 11 12 22. It is therefore represented by the message polynomial $m(x)$

$m(x) = 10 + 11x + 12x^2 + 22x^3 =$

$= \alpha^0 + \alpha^7 x + \alpha^2 x^2 + \alpha^3 x^3$.

The first calculation in systematic encoding is

$x^{n-k}.m(x) = x^4.m(x) =$

$= 10x^4 + 11x^5 + 12x^6 + 22x^7 =$

$= \alpha^0 x^4 + \alpha^7 x^5 + \alpha^2 x^6 + \alpha^3 x^7$.

Then dividing the polynomial $x^4.m(x)$ by the generating polynomial $g(x)$ gives a quotient $q(x)$ and a remainder $t(x)$:

$q(x) = 01 + 12x + 22x^3 = \alpha^1 + \alpha^2 x + \alpha^3 x^3$;

$t(x) = 11 + 11x + 20x^2 + 01x^3 =$

$= \alpha^7 + \alpha^7 x + \alpha^4 x^2 + \alpha^1 x^3$.

The inverse additive polynomial is found as follows:

$- t(x) = - (11 + 11x + 20x^2 + 01x^3) =$

$= 22 + 22x + 10x^2 + 02x^3 =$

$= \alpha^3 + \alpha^3 x + \alpha^0 x^2 + \alpha^5 x^3$.

The codeword is obtained according to (13)

$c(x) = x^4.m(x) + (- t(x)) = 22 + 22x +$

$= + 10x^2 + 02x^3 + 10x^4 + 11x^5 + 12x^6 + 22x^7 =$

$= \alpha^3 + \alpha^3 x + \alpha^0 x^2 + \alpha^5 x^3 + \alpha^0 x^4 + \alpha^7 x^5 + \alpha^2 x^6 + \alpha^3 x^7$.

If the code word is represented only by the coefficients of the polynomial $c(x)$, it is obtained by concatenating the inverse additive elements of the coefficients of the remainder $t(x)$ to the right of the input symbols of the message polynomial $m(x)$

$$c(x) = \underset{m(x)}{(10\ 11\ 12\ 22)} \quad \underset{-t(x)}{(22\ 22\ 10\ 02)}.$$

B. LFSR Encoder for pRS Codes

As well as obtaining an output p-ary pseudorandom sequence with a pLFSR register [32], the systematic encoding of $(n, k) = (p^m - 1, k)$ pRS codes can be achieved through a linear feedback shift register. As demonstrated in the preceding paragraphs, the elements of the input sequence, $m(x)$, and the code sequence, $c(x)$, respectively, are the elements of a Galois field extension, GF(p^m). This will determine the characteristics of the elements of the pLFSR register operating in the Galois field GF(p^m). Each of its elements must be a non-binary register comprising m elements, each of which can store integers from 0 to $p - 1$. The elements $\oplus$ implement the addition operation in the field GF(p^m), and $\otimes$ implements the multiplication operation in the field GF(p^m). The feedback coefficients g_1, g_2, …, g_L are determined by the coefficients of the generator polynomial $g(x)$ of the pRS code (3).

It should be noted that the generating polynomial $g(x)$ is not always a primitive polynomial so that some transformations can be applied to determine the feedback coefficients. Therefore, it is necessary to implement a hardware scheme for the polynomial division operation. Such a scheme was proposed by Peterson and Weldon in their book [33].

This section presents the specific characteristics of the aforementioned scheme when applied to an arbitrary field GF(p^m) with a base p that differs from 2. To illustrate this process, we will initially consider the example of performing polynomial division on the input sequence $x^{n-k}.m(x) = x^4.m(x)$ with the generator polynomial $g(x)$.

Example 2. Polynomial division of the input message sequence with the generator polynomial $g(x)$.

In the case of the (8, 4) 3RS code with generator polynomial $g(x) = 12 + 20x + 12x^2 + 11x^3 + 10x^4 = \alpha^2 + \alpha^4 x$

+ $\alpha^2x^2 + \alpha^7x^3 + \alpha^0x^4$ and input message 10 11 12 22, it is necessary to perform the division of $x^4.m(x) = 10x^4 + 11x^5 + 12x^6 + 22x^7 = \alpha^0x^4 + \alpha^7x^5 + \alpha^2x^6 + \alpha^3x^7$ by $g(x)$.

The steps of the actual polynomial division are shown in Fig. 1. For a more compact representation of the polynomial division, the powers of x in the dividend and divisor are omitted, and only the coefficients of the polynomials are recorded. Here, $q(x)$ is a quotient and $t(x)$ – remainder.

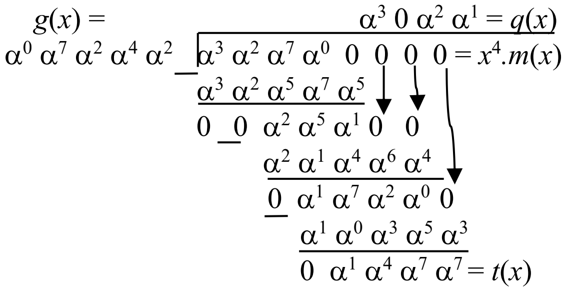
$g(x) =$
$\alpha^3\ 0\ \alpha^2\ \alpha^1 = q(x)$
$\alpha^0\ \alpha^7\ \alpha^2\ \alpha^4\ \alpha^2$
$\alpha^3\ \alpha^2\ \alpha^7\ \alpha^0\ 0\ 0\ 0\ 0 = x^4.m(x)$
$\alpha^3\ \alpha^2\ \alpha^5\ \alpha^7\ \alpha^5$
$0\ 0\ \alpha^2\ \alpha^5\ \alpha^1\ 0\ 0$
$\alpha^2\ \alpha^1\ \alpha^4\ \alpha^6\ \alpha^4$
$0\ \alpha^1\ \alpha^7\ \alpha^2\ \alpha^0\ 0$
$\alpha^1\ \alpha^0\ \alpha^3\ \alpha^5\ \alpha^3$
$0\ \alpha^1\ \alpha^4\ \alpha^7\ \alpha^7 = t(x)$

Fig. 1. Example of polynomial division.

As can be seen from Fig. 1, at each step the value of the product of $g(x)$ with the current input symbol is subtracted from the state of the current dividend. This operation can be implemented with a linear feedback shift register. Because it operates in an extension of a Galois field, it will be called GF(p^m) LFSR

Table captions and titles should always be right aligned and placed above the tables. Tables are numbered consecutively with Roman numerals and have reference in the main text.

The Galois architecture of the GF(p^m) LFSR will be considered because it directly corresponds to the real polynomial division algorithm shown in Fig. 1. The scheme of a GF(p^m) LFSR register with Galois architecture (Fig. 2) implementing systematic encoding of p-ary Reed-Solomon codes consists of a register ***R*** with $n - k$ number of elements, each of which can store one element of the extension GF(p^m), and with linear feedbacks defined by the generator polynomial $g(x)$, an inverter ***I*** that outputs additive inverse element $-a_0$ in the GF(p^m) field, and two keys ***K*₁** and ***K*₂**.

The operation of the scheme is as follows:

1. The information symbols *ms*, which are *k* in number, arrive sequentially in the register ***R*** and appear at the output ***c***. For this purpose, the key ***K*₁** is in the upper position and the key ***K*₂** is closed.
2. The register ***R*** continues to clock for $n - k$ more clock cycles, with no new symbols *ms* fed to its input. This computes the remainder $t(x)$ of the polynomial division, which resides in register ***R***. No symbols are output to the output ***c***. For this purpose, key ***K*₁** is in the middle position and key ***K*₂** is closed.
3. The register ***R*** continues to clock for $n - k$ more clock cycles, with no new symbols *ms* fed to its input. At this, the already obtained remainder $t(x)$ is output through the inverter ***I*** to the output ***c***. Key ***K*₁** is in the down position and key ***K*₂** is open.

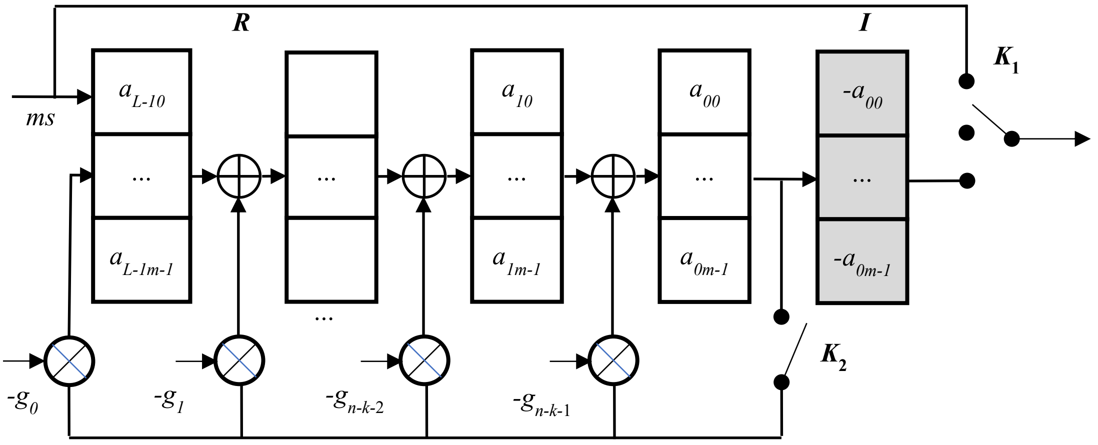

Fig. 2. Galois architecture of the GF(p^m) LFSR

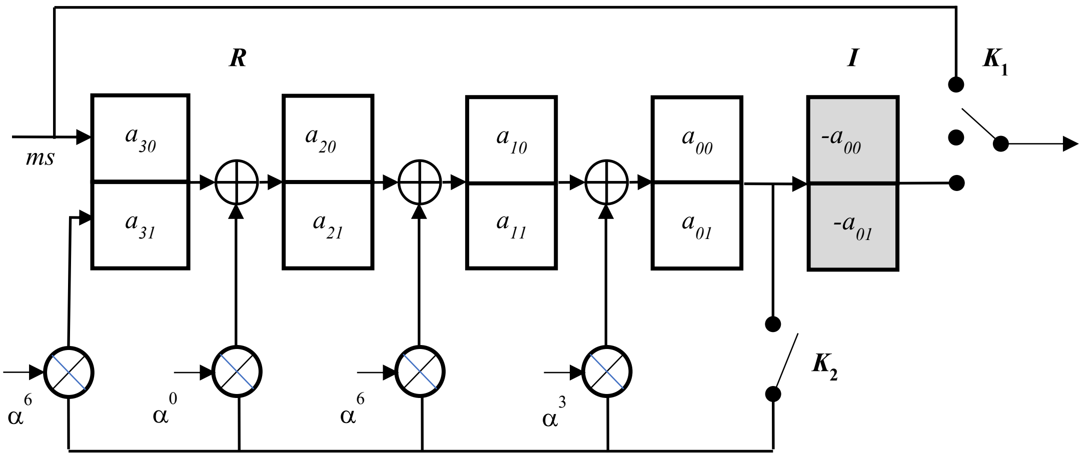

Fig. 3. GF(3^2) LFSR with generator polynomial $g(x) = \alpha^2 + \alpha^4x + \alpha^2x^2 + \alpha^7x^3 + \alpha^0x^4$

C. Example of Systematic Encoding with GF(p^m) LFSR register

The operation of the scheme of Fig. 2 will be explained with the example which actual polynomial division is shown in Fig. 1.

Example 3. The cyclic code under consideration is an (8, 4) 3RS code. The register ***R***, therefore, consists of four elements, each of length $m = 2$. The feedback coefficients are calculated from the generator polynomial $g(x) = 12 + 20x + 12x^2 + 11x^3 + 10x^4 = \alpha^2 + \alpha^4x + \alpha^2x^2 + \alpha^7x^3 + \alpha^0x^4$ as shown below

$$\begin{aligned} -g_0 &= -\alpha^2 = -12 = 21 = \alpha^6 \\ -g_1 &= -\alpha^4 = -20 = 10 = \alpha^0 \\ -g_2 &= -\alpha^2 = -12 = 21 = \alpha^6 \\ -g_3 &= -\alpha^7 = -11 = 22 = \alpha^3 \end{aligned} \qquad (14)$$

Therefore, the GF(3^2) LFSR register scheme with Galois architecture has the form shown in Fig. 3. The sequential operation of the scheme is shown in Table 1.

In the initial state, all four elements of the GF(3^2) LFSR register, which encodes an (8, 4) 3RS code, are in a zero state. For the next four clocks, key $\boldsymbol{K_1}$ is in the upper position and key $\boldsymbol{K_2}$ is closed. Therefore, the information symbols *ms* arrive sequentially in register ***R*** and simultaneously appear at output ***c***.

For the next four clocks, key $\boldsymbol{K_1}$ is in the middle position and key $\boldsymbol{K_2}$ is closed. No more symbols *ms* are passed to its input and no symbols are produced at the output ***c***. The computation of the states of the elements of the register **R** when the linear feedbacks are operating is performed as follows:

Clock 5. $a_0 = \alpha^3.\ \alpha^3 + \alpha^2 = \alpha^6 + \alpha^2 = 21 + 12 =$

$= 00 = 0$

$a_1 = \alpha^3.\ \alpha^6 + \alpha^7 = \alpha^1 + \alpha^7 = 01 + 11 = 12 = \alpha^2$

$a_2 = \alpha^3.\ \alpha^0 + \alpha^0 = \alpha^3 + \alpha^0 = 22 + 10 = 02 = \alpha^5$

$a_3 = \alpha^3.\ \alpha^6 = \alpha^1$

Table 1 Sequential operation of the GF(32) LFSR register from Example 3

Clock	Input ms	Register R				Output c
		a_3	a_2	a_1	a_0	
0	-	0	0	0	0	
1	α^3	α^3	0	0	0	α^3
2	α^2	α^2	α^3	0	0	α^2
3	α^7	α^7	α^2	α^3	0	α^7
4	α^0	α^0	α^7	α^2	α^3	α^0
5	0	α^1	α^5	α^2	0	-
6	0	0	α^1	α^5	α^2	-
7	0	α^0	α^2	α^7	α^1	-
8	0	α^7	α^7	α^4	α^1	-
9	0	0	α^7	α^7	α^4	$-\alpha^1 = \alpha^5$
10	0	0	0	α^7	α^7	$-\alpha^4 = \alpha^0$
11	0	0	0	0	α^7	$-\alpha^7 = \alpha^3$
12	0	0	0	0	0	$-\alpha^7 = \alpha^3$

Clock 6. $a_0 = 0.\ \alpha^3 + \alpha^2 = \alpha^2$

$a_1 = 0.\ \alpha^6 + \alpha^5 = \alpha^5$

$a_2 = 0.\ \alpha^0 + \alpha^1 = \alpha^1$

$a_3 = 0.\ \alpha^6 = 0$

Clock 7. $a_0 = \alpha^2.\ \alpha^3 + \alpha^5 = \alpha^5 + \alpha^5 = 02 + 02 =$

$= 01 = \alpha^1$

$a_1 = \alpha^2.\ \alpha^6 + \alpha^1 = \alpha^0 + \alpha^1 = 10 + 01 = 11 = \alpha^7$

$a_2 = \alpha^2.\ \alpha^0 + 0 = \alpha^2$

$a_3 = \alpha^2.\ \alpha^6 = \alpha^0$

Clock 8. $a_0 = \alpha^1.\ \alpha^3 + \alpha^7 = \alpha^4 + \alpha^7 = 20 + 11 =$

$= 01 = \alpha^1$

$a_1 = \alpha^1.\ \alpha^6 + \alpha^2 = \alpha^7 + \alpha^2 = 11 + 12 = 20 = \alpha^4$

$a_2 = \alpha^1.\ \alpha^0 + \alpha^0 = \alpha^1 + \alpha^0 = 01 + 10 = 11 = \alpha^7$

$a_3 = \alpha^1.\ \alpha^6 = \alpha^7$

In the last four cocks, key $\boldsymbol{K_1}$ is in the down position and key $\boldsymbol{K_2}$ is open. The remainder is sent through the inverter ***I*** to the output ***c***.

From Table 1, it can be seen that after the 12th clock, the scheme is in the initial zero state and it can start encoding the next k number of information symbols from the field GF(p^m).

IV. Conclusion and Feature Work

The article proposes an approach to implement a systematic p-ary Reed-Solomon encoding with input information symbols in a Galois field GF(p^m) with an arbitrary prime p.

This article presents a detailed derivation of the special features involved in determining the coefficients of the parity polynomial $t(x)$. It is demonstrated that the additive inverse of each coefficient of the parity polynomial $t(x)$ must be written to the $n - k$ lowest positions of the codeword. Furthermore, the article examines the process of encoding non-binary symbols in GF(p^m) with an arbitrary prime p using a LFSR-based circuit. It also gives some examples of systematic encoding and polynomial division, as well as an LFSR-based encoder for an (8, 4) 3RS code over the field GF(3^2).

Nevertheless, the proposed approach presents certain theoretical and practical issues that require further examination. From a theoretical standpoint, a comprehensive analysis is necessary to elucidate the distinctive characteristics that emerge when employing a GF(p^m) field comprising an odd prime base p in the p-ary Reed-Solomon codes decoding process.

From a practical standpoint, it is necessary to design the Field Programmable Gate Arrays (FPGAs) [34], [35] or Application Specific Integrated Circuits (ASICs) [36], [37] hardware implementation of the proposed approach, which will facilitate the acceleration of algebraic operations in the Galois field GF(p) and Galois Extension GF(p^m) for $p = 3$, 5 or 7. Such an approach is well-suited to the analysis of multi-level sequences, including PAM-3, PAM-5, and PAM-7 signals.

Acknowledgments:

The article was prepared with the financial support of the National Scientific Program “Security and Defence”, funded by the Ministry of Education and Science of the Republic of Bulgaria, in implementation of Decision № 731 of 21.10.2021 of the Council of Ministers of the Republic of Bulgaria.

The authors would like to thank the reviewers for their helpful comments.

References

[1] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” Journal of the Society for Industrial & Applied Mathematics 8, No. 2, 1960, pp. 300-304.

[2] X. Liu, H. Jia and C. Ma, “Error-Correction codes For Optical Disc Storage, ” Advances in Optical Data Storage Technology, Proceedings of SPIE Vol. 5643, 2005, pp. 342-347.

[3] T. N. Hewage, M. N. Halgamuge, A. Syed, and G. Ekici, “Big data techniques of Google, Amazon, Facebook and Twitter, ” Journal of Communications, Vol. 13, No. 2, 2018, pp. 94-100.

[4] A. Chiniah and A. Mungur, “On the Adoption of Erasure Code for Cloud Storage by Major Distributed Storage Systems, ” EAI Endorsed Transactions on Cloud Systems, 7(21), e1-e11, 2022.

[5] J. A. Lin and C. S. Fuh, “2D Barcode Image Decoding, ” Mathematical Problems in Engineering, Article ID 848276, 10 pages, 2013. https://doi.org/10.1155/2013/848276.

[6] S. K. Moorthy, N. Mastronarde, E. S. Bentley, M. Medley, and Z. Guan, “OSWireless: Hiding specification complexity for zero-touch software-defined wireless networks, ” Computer Networks, *237*: 110076, 2023.

[7] S. K. Moorthy, Z. Guan, N. Mastronarde, E. S. Bentley, and M. Medley, OSWireless: Enhancing automation for optimizing intent-driven software-defined wireless networks, 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS), IEEE, 2022, pp. 202-210.

[8] M. Almaz and B. Sevinj, “Estimation of the Noise Immunity Characteristics of Telecommunication Network, ” *WSEAS Transactions on Communications*, 22, 2023, pp. 192-198.

[9] J. Malhotra, “Investigation of channel coding techniques for high data rate mobile wireless systems, ” *International Journal of Computer Applications*, 115:3, 2015.

[10] E. Mohamed, W. Azeddine, M. Omar, and A. Hadjoudja, Development and Validation of an optimized syndromes block for Reed Solomon decoder. ITM Web of Conferences, Vol. 52, EDP Sciences. 2023, p. 03008.

[11] J. Wassermann and A. Dziech, “Multidimensional Enhanced Hadamard Error Correcting Code in Comparison with Reed-Solomon Code in Video-Watermarking Applications, ” WSEAS transactions on signal processing, Vol. 13, 2017, pp. 196-207.

[12] V. Riznyk, “Designs of Electronic Devices using Combinatorial Optimization, ” WSEAS Transactions on Electronics, Vol. 14, 2023, pp. 122-128.

[13] A. Ndolo and İ. H. Çavdar, “Current state of communication systems based on electrical power transmission lines, ” Journal of Electrical Systems and Information Technology, Vol. 8, 2021, pp. 1-10.

[14] C. F. Fontana, C. A. Sakurai, C. L. Marte, J. R. Cardoso, and A. D. S. Andrade, “Power Line Communication as Alternative for Data Communication Channel for BRT, ” Earth sciences and human constructions, Vol. 2, 2022, pp. 60-67.

[15] X. Gao and L. Huai, “Modern ultra-wideband communications: recent overview and future prospects, ” International Journal of Ultra Wideband Communications and Systems, *4*(2), 2020, pp. 57-67.

[16] C. Sandoval-Ruiz, “LFSR-fractal ANN model applied in R-IEDs for smart energy, ” IEEE Latin America Transactions, 18(04), 2020, pp. 677-686.

[17] G. Kiokes, “Hardware Implementation of 802.11 ad MIMO-OFDM Transceiver, ” WSEAS transactions on communications, Vol. 18, 2019, pp. 71-77.

[18] A. Pham and H. Yashima, Performance analysis of Reed-Solomon coded spectral amplitude encoding OCDM system. Proceedings of the 4th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications, 2005, pp. 1-5.

[19] R. M. Vitenberg, A WFMT downlink transmitter for low earth orbit satellite. Proceedings of the 4th WSEAS Int. Conference on Electromagnetics, Wireless and Optical Communications, 2006, pp. 53-58.

[20] D. Dannan, K. Kabalan, and A. Chehab, "Performance comparison for serial concatenated block convolutional codes when sequentially and iteratively decoded, " WSEAS Transactions on Information Science and Applications, 4(6), 2007, pp. 1236-1244.

[21] L. Chaari, M. Fourati, N. Masmoudi, and L. Kamoun, "A reconfigurable FEC system based on Reed-Solomon codec for DVB and 802.16 network, " WSEAS transactions on circuits and systems, 8(8), 2009, pp. 729-744.

[22] T. H. Hu and M. H. Chang, "Decoding shortened Reed Solomon codes at bit level, " WSEAS Transactions on Communications, 9(11), 2010, pp. 695-707.

[23] A. Abderrahmane, M. Merouane, and B. Messaoud, "Diversity Techniques to combat fading in WiMAX, " WSEAS transactions on communications, 7, 2008, pp. 43-51.

[24] N. Stojanović, C. Prodaniuc, Z. Liang, J. Wei, S. Calabró, T. Rahman, C. Xie, "4D PAM-7 Trellis Coded Modulation for Data Centers, " IEEE Photonics Technology Letters, Vol. 31, No. 5, pp. 369-372, 1 March 2019, doi 10.1109/LPT.2019.2895686.

[25] H.-U. Kim and J.-K. Kang, High-speed Serial Interface using PWAM Signaling Scheme, 2022 19th International SoC Design Conference (ISOCC), Gangneungsi, Korea, Republic of, 2022, pp. 255-256, doi: 10.1109/ISOCC56007.2022.10031330.

[26] K. Matheus and T. Königseder. Automotive Ethernet. Cambridge University Press, 2021.

[27] R. C. Bose and D.K. Ray-Chaudhuri. "On a class of error correcting binary group codes, " Information and Control, Vol. 3, Issue 1, March 1960, pp. 68–79.

[28] N. G. Bardis, O. Markovskyi, and N. Doukas, Efficient burst error correction method for application in low frequency channels and data storage units, 17th International Conference on Digital Signal Processing (DSP), 2011, IEEE pp. 1-6.

[29] B. Sklar, Digital Communications: Fundamentals and Applications, Second Edition, Prentice-Hall, 2001.

[30] A. Beletsky. "An Effective Algorithm for the Synthesis of Irreducible Polynomials over a Galois Fields of Arbitrary Characteristics, " WSEAS Transactions on Mathematics, 20, 2021, pp. 508-519.

[31] Z. Savova and R. Bogdanov, Some Specific Features in the Construction of p-ary Reed-Solomon Codes for an Arbitrary Prime p. Proceedings of the 15th International Scientific and Practical Conference. Environment. Technology. Resources. Rezekne, Latvia, Vol. 4, 2024, pp. 237-243.

[32] M. Goresky, A. Klapper, "Fibonacci and Galois Representations of Feedback-With-Carry Shift Registers, " IEEE Trans. on Inform. Theory, vol. 48, pp. 2826−2836, November 2002.

[33] W. W. Peterson and E. J. Weldon, Error-correcting codes. Cambridge, MA: MIT Press 1972.

[34] P. Balasubramanian and N.E., Mastorakis, "FPGA based implementation of distributed minority and majority voting based redundancy for mission and safety-critical applications, " International Journal of Circuits and Electronics, 2016. *arXiv preprint arXiv: 1611.09446*.

[35] H. Shiyang, L. Hui, L. Qingwen, and L.Fenghua, "A Time-Area-Efficient and Compact ECSM Processor over GF (p), " Chinese Journal of Electronics, *32*(6), 2023, pp. 1355-1366.

[36] P. Balasubramanian and N.E. Mastorakis, "ASIC-based implementation of synchronous section-carry based carry lookahead adders, " Recent Advances in Circuits, Systems, Signal Processing and Communications, 2016, *arXiv preprint arXiv: 1603.07961*.

[37] N. D. Patwari, A. Srivastav, M. Kabra, P. Jonna, and M. Rao, Design and evaluation of finite field multipliers using fast XNOR cells. In Proceedings of the Great Lakes Symposium on VLSI 2023, 2023, pp. 163-166.