UDC 681.3
DOI: 10.15587/2706-5448.2020.210272

Semerenko V.

PRESENTATION OF REED-SOLOMON CODES BASED ON AUTOMATON THEORY

The object of research is the processes of error-correcting coding in telecommunication and computer systems. The main attention is paid to Reed-Solomon (RS) codes, which belong to the very widespread error-correcting codes. Despite the 60-year existence of these codes, the complexity of their decoding still remains a problem. This problem is mainly due to the use of an algebraic approach to their description.

The article proposes to use the theory of linear finite-state machine (LFSM) for RS codes as a mathematical basis, which is a combination of the theory of digital filters and finite automaton over nonbinary Galois fields. In the course of research, 12 types of LFSMs are considered for the first time: the recursive LFSMs of 8 types and the non-recursive LFSMs of 4 types.

The recursive LFSMs are used for systematic encoding and form a circuit for dividing of polynomials, and the non-recursive LFSMs are used for non-systematic encoding and form a circuit for multiplying of polynomials. All types of LFSMs give the same result for encoding and decoding, but with different complexity, which is important for practical implementation.

The automaton representation is the most suitable for RS codes, since it takes into account the cyclicity property and other features of these codes to the maximum. In contrast to algebraic methods, automaton decoding methods have a simple software and hardware implementation and high performance. With the help of automaton-graphical models, it can accurately estimate the corrective capability of the code. Automaton representation combines known methods of representing Reed-Solomon codes (polynomial, matrix, algebraic) and provides mutual transitions between them.

The article attention is spare to the fact that automaton methods for encoding and decoding (n, k)-codes of RS using quantum computers give a gain in time n times.

Keywords: *Reed-Solomon codes, automaton theory, linear finite-state machine (LFSM), decoding, quantum computer.*

Received date: 02.04.2020
Accepted date: 27.05.2020
Published date: 31.08.2020

1. Introduction

This paper is the extended version of conference paper [1].

The Reed-Solomon (RS) codes appeared 60 years ago and until now they have included into the best error-correcting codes. This is evidenced by the scope of use of these codes: fiber-optic communication lines, mobile communications, digital television, optical discs, etc. [2].

In recent years, new codes and new principles for decoding error-correcting codes began to appear, but in RS codes, as in other subclasses of cyclic codes, the laborious and inconvenient Berlekamp-Massey method remains the main decoding method [3].

This situation has developed mainly because cyclic codes are considered only as a subclass of linear codes and, accordingly, universal algebraic methods for decoding linear codes are try adapted to these codes.

However, linear codes combine completely different codes and it is impossible to create encoding and decoding methods that would be effective for all subclasses of linear codes. Therefore, different types of linear codes should be considered separately, taking into account their properties and features.

In [4], a new representation of cyclic codes was considered and with its help, automaton methods for encoding and decoding these codes in Galois binary fields were substantiated. Therefore, it is important to ensure high validity and fault-tolerance of various technical devices and systems. Thus, *the object of research* is the processes of error correcting coding in telecommunication and computer systems. *The aim of research* is to improve the efficiency of means of transmission, storage, processing and protection of data through the development of new theoretical models of RS codes.

2. Methods of research

The general theoretical model for all subclasses of cyclic codes can be the mathematical apparatus of digital filters. The theory of error correcting coding and the theory of filtering are united by a common goal: restoration of useful input signals against a background of interference by observing the corresponding output signals [5].

It is known that the classical digital filter is a nonlinear system. Since cyclic codes belong to linear codes, therefore, the theory of filters can be applied to error-correcting coding

ISSN 2664-9969

only after eliminating the phenomenon of nonlinearity. This problem can be solved with the help of the Galois fields.

On the other hand, the theory of filters is also close to the theory of automata, since a digital filter converts input automaton words into output automaton words. Thus, for a linear filter, it is possible to give an automatic way to describe it [6]:

$$s_{i+1} = s_i T + u_i B;$$
$$w_i = s_i R + u_i Q,$$

where u_i is the input word; w_i is the output word; s_i is the state word; T, B, R, Q are the matrices which characterize the structure of the filter.

A traditional finite automaton is based on the Boolean algebra. If finite Galois fields are used as a basic mathematical apparatus, then a linear automaton (linear filter) will be obtained, the processes in which develop in time.

As a result, a new mathematical model will be obtained, which can be called a linear finite-state machine (LFSM).

According to [7], an LFSM with one input, one output and memory elements is a linear finite automaton, which at discrete time steps t over the Galois field $GF(q)$ is described by a transition function:

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

and output function:

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

where $U(t) = |u_i|_1$ is the input word; $Y(t) = |y_i|_1$ is the output word; $S(t) = |s_i|_r$ is the state word.

Let's consider the classification of LFSM depending on the structure of their characteristic matrices. Let's use the terminology of digital filters and algebraic polynomials in non-binary Galois fields.

From these positions, it is convenient to separate of all LFSMs into two groups: recursive (are used to divide of polynomials) and non-recursive (are used to multiply of polynomials).

In most publications, only LFSMs of the Galois type and Fibonacci type [8] are distinguished in consequence of a difference of structure of characteristic matrices these LFSMs.

If to consider the different structures of matrices A and B, and also distinguish between alternative directions of information transfer, then it is possible to get eight basic types of recursive LFSMs [1]:

– left-sided LFSM of type 1 (Galois type) with matrices:

$$A = \begin{vmatrix} 0 & 0 & \cdots & 0 & g_0 \\ \alpha^0 & 0 & \cdots & 0 & g_1 \\ 0 & \alpha^0 & \cdots & 0 & g_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha^0 & g_{2\tau-1} \end{vmatrix},\ B = \begin{bmatrix} \alpha^0 \\ 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix}; \quad (3)$$

– left-sided LFSM of type 2 with matrices:

$$A = \begin{vmatrix} 0 & 0 & 0 & \cdots & g_0 \\ \alpha^0 & 0 & 0 & \cdots & g_1 \\ 0 & \alpha^0 & 0 & \cdots & g_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \alpha^0 & g_{r-1} \end{vmatrix},\ B = \begin{vmatrix} g_0 \\ g_1 \\ g_2 \\ \cdots \\ g_{r-1} \end{vmatrix}; \quad (4)$$

– left-sided LFSM of type 3 (Fibonacci type) with matrices:

$$A = \begin{vmatrix} g_{r-1} & g_{r-2} & \cdots & g_1 & g_0 \\ \alpha^0 & 0 & \cdots & 0 & 0 \\ 0 & \alpha^0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha^0 & 0 \end{vmatrix},\ B = \begin{vmatrix} \alpha^0 \\ 0 \\ \cdots \\ 0 \\ 0 \end{vmatrix}; \quad (5)$$

– left-sided LFSM of type 4 with matrices:

$$A = \begin{vmatrix} g_{r-1} & g_{r-2} & \cdots & g_1 & g_0 \\ \alpha^0 & 0 & \cdots & 0 & 0 \\ 0 & \alpha^0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha^0 & 0 \end{vmatrix},\ B = \begin{vmatrix} g_0 \\ g_1 \\ g_2 \\ \cdots \\ g_{r-1} \end{vmatrix}; \quad (6)$$

– right-sided LFSM of type 1 (Galois type) with matrices:

$$A = \begin{vmatrix} g_{r-1} & \alpha^0 & 0 & \cdots & 0 \\ g_{r-2} & 0 & \alpha^0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_1 & 0 & 0 & \cdots & \alpha^0 \\ g_0 & 0 & 0 & \cdots & 0 \end{vmatrix},\ B = \begin{vmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ \alpha^0 \end{vmatrix}; \quad (7)$$

– right-sided LFSM of type 2 with matrices:

$$A = \begin{vmatrix} g_{r-1} & \alpha^0 & 0 & \cdots & 0 \\ g_{r-2} & 0 & \alpha^0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_1 & 0 & 0 & \cdots & \alpha^0 \\ g_0 & 0 & 0 & \cdots & 0 \end{vmatrix},\ B = \begin{vmatrix} g_{r-1} \\ g_{r-2} \\ \cdots \\ g_1 \\ g_0 \end{vmatrix}; \quad (8)$$

– right-sided LFSM of type 3 (Fibonacci type) with matrices:

$$A = \begin{vmatrix} 0 & \alpha^0 & 0 & \cdots & 0 \\ 0 & 0 & \alpha^0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \alpha^0 \\ g_0 & g_1 & g_2 & \cdots & g_{r-1} \end{vmatrix},\ B = \begin{vmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ \alpha^0 \end{vmatrix}; \quad (9)$$

– right-sided LFSM of type 4 with matrices:

$$A = \begin{vmatrix} 0 & \alpha^0 & 0 & \cdots & 0 \\ 0 & 0 & \alpha^0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \alpha^0 \\ g_0 & g_1 & g_2 & \cdots & g_{r-1} \end{vmatrix},\ B = \begin{vmatrix} g_{r-1} \\ g_{r-2} \\ \cdots \\ g_1 \\ g_0 \end{vmatrix}. \quad (10)$$

The entries of the last column of the matrix A from (3) and (4), the entries of the first row of the matrix A from (5) and (6), the entries of the first column of the matrix A from (7) and (8), the entries of the last row of the matrix A from (9) and (10), the entries of the matrix B from (4), (6), (8) and (10) are the constant coefficients of the generator polynomial:

$$g(x) = g_0 + g_1 x + \cdots + g_{r-1} x^{r-1} + g_r x^r, \quad (11)$$

where degree r ($r=n-k$) in the Galois field $GF(q)$. The coefficient g_i in (11) is equal to the j-th degree of the primitive element of the field $GF(q)$ ($j=0\div r-1$). For other matrices in (2), it is possible to choose constant values:

$$C=\left[\alpha^0\right],\ D=[0].$$

Let's now consider the classification of non-recursive LFSMs. In contrast to the LFSMs considered above, signals to the inputs of non-recursive LFSMs never come from their outputs. As a result, it is possible to get four basic types of non-recursive LFSMs:

– non-recursive LFSM of type 1 with matrices:

$$A=\begin{vmatrix}0 & 0 & \cdots & 0 & 0\\ \alpha^0 & 0 & \cdots & 0 & 0\\ 0 & \alpha^0 & \cdots & 0 & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & \cdots & \alpha^0 & 0\end{vmatrix},\ B=\begin{bmatrix}g_0\\ g_1\\ g_2\\ \cdots\\ g_{r-1}\end{bmatrix};\qquad(12)$$

– non-recursive LFSM of type 2 with matrices:

$$A=\begin{vmatrix}0 & 0 & 0 & \ldots & 0\\ \alpha^0 & 0 & 0 & \ldots & 0\\ 0 & \alpha^0 & 0 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & \alpha^0 & 0\end{vmatrix},\ B=\begin{vmatrix}\alpha^0\\ 0\\ 0\\ \ldots\\ 0\end{vmatrix};\qquad(13)$$

– non-recursive LFSM of type 3 with matrices:

$$A=\begin{vmatrix}0 & \alpha^0 & 0 & \ldots & 0\\ 0 & 0 & \alpha^0 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & \ldots & \alpha^0\\ 0 & 0 & 0 & \ldots & 0\end{vmatrix},\ B=\begin{bmatrix}g_{r-1}\\ g_{r-2}\\ \ldots\\ g_1\\ g_0\end{bmatrix};\qquad(14)$$

– non-recursive LFSM of type 4 with matrices:

$$A=\begin{vmatrix}0 & \alpha^0 & 0 & \ldots & 0\\ 0 & 0 & \alpha^0 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & \ldots & \alpha^0\\ 0 & 0 & 0 & \ldots & 0\end{vmatrix},\ B=\begin{bmatrix}0\\ 0\\ \ldots\\ 0\\ \alpha^0\end{bmatrix}.\qquad(15)$$

Another feature of non-recursive LFSMs is that all their characteristic matrices must be specified to explain of its functioning:

– non-recursive LFSM of type 1 has the matrices C and D:

$$C=\left|0\ \ 0\ \ \ldots\ \ 0\ \ \alpha^0\right|,\ D=\left[\alpha^0\right];\qquad(16)$$

– non-recursive LFSM of type 2 has the matrices C and D:

$$C=\left|g_{r-1}\ \ \ldots\ \ g_2\ \ g_1\ \ g_0\right|,\ D=\left[\alpha^0\right];\qquad(17)$$

– non-recursive LFSM of type 3 has the matrices C and D:

$$C=\left|\alpha^0\ \ 0\ \ \ldots\ \ 0\ \ 0\right|,\ D=\left[\alpha^0\right];\qquad(18)$$

– non-recursive LFSM of type 4 has the matrices C and D:

$$C=\left|g_0\ \ g_1\ \ g_2\ \ \ldots\ \ g_{r-1}\right|,\ D=\left[\alpha^0\right].\qquad(19)$$

The matrix elements in (17) and (19) correspond to the coefficients of the polynomial (11).

It is important to note the relationship of the characteristic matrices of LFSM. For tasks of error-correcting coding, the matrices A and B must be chosen so that the recursive and non-recursive LFSM are r-controllable. The LFSM will be r-controllable if it is equal to the rank r of the $(r\times r)$-matrix:

$$L_r=\left[A^{r-1}\times B,\ \ A^{r-2}\times B,\ \ \ldots\ ,\ \ A\times B,\ \ B\right].\qquad(20)$$

To calculate (20), the values of the matrices A and B are taken from formulas (3)–(10) or (12)–(15).

3. Research results and discussion

3.1. Definition of Reed-Solomon codes based on automaton models. Based on the above types of LFSM, it is possible to define the RS codes. Let's name such LFSMs as the symbolic LFSM.

Let the LFSM be in some initial state $S_{beg}(t)$, for example, in zero state. Let's feed of n-symbol sequence L at its inputs so that the LFSM returns to the state $S_{beg}(t)$ again after n time clocks.

Definition 1. The set of all m-bit length sequences L that transferred the LFSM from any initial state $S_{beg}(t)$ back to the state $S_{beg}(t)$ creates the RS (n, k)-code over the Galois field $GF(q)$. Each such sequence is a codeword Z of RS (n, k)-code.

Polynomial (11), which is included in the description of LFSM of all types, is the generator polynomial of the RS code.

Since LFSM is a finite automaton, therefore, in addition to the automaton-analytical model of the RS code, there is an automaton-graphical model of the RS code.

Let's suppose that, in strongly connected graph G_{FA} (state graph and output graph of LFSM), i-th edge e_i corresponds to the symbol z_i of the codeword Z over the field $GF(q)$ ($z_i\in Z$, $i=1\div n$). Then a sequence L of unidirectional edges creates in graph G_{FA} a code path η of length n that will correspond to the codeword Z.

It is in automaton models of the RS code that the main property of this cyclic code is fully used – the cyclic property.

Let's note that for calculation of the elements over $GF(q)$ it is possible to use with LFSM over $GF(2)$ that being described by own matrices A_b, B_b, C_b and D_b. Let's name it as the binary LFSM). This LFSM is based on the primitive polynomial of degree m over $GF(2)$.

For example, RS (15,11)-code, to which the generator polynomial (2):

$$g(x)=\alpha^{10}+\alpha^3x+\alpha^6x^2+\alpha^{13}x^3+x^4,\ GF(8),$$

corresponds such characteristic matrices of a recursive left-sided symbolic LFSM of type 2:

$$A=\begin{vmatrix}0 & 0 & 0 & \alpha^{10}\\ \alpha^0 & 0 & 0 & \alpha^3\\ 0 & \alpha^0 & 0 & \alpha^6\\ 0 & 0 & \alpha^0 & \alpha^{13}\end{vmatrix};\quad B=\begin{vmatrix}\alpha^{10}\\ \alpha^3\\ \alpha^6\\ \alpha^{13}\end{vmatrix}.$$

For calculation of elements over $GF(8)$ can to use the binary LFSM of type 1 with matrices over $GF(2)$:

ISSN 2664-9969

$$A_b = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix}; \quad B_b = \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \end{vmatrix},$$

which is based on primitive generator polynomial:

$$g(x) = 1 + x + x^4,\ GF(2).$$

3.2. The encoding of Reed-Solomon codes based on automaton models. RS codes allow both systematic and unsystematic encoding. Let's consider systematic encoding in more detail.

From the position of the automaton representation of RS (n, k)-code, the systematic encoding procedure consists of two stages.

At the first stage, a k-symbol information word is fed to the input of the recursive LFSM, as a result of which the LFSM will move from the initial zero state $S(0)$ to the state $S(k)$ during the k time clocks according to the formula that follows from (2):

$$S(k) = A^k \times S(0) + L_k \times I,\ GF(q),$$

where $L_k = \left[A^{k-1} \times B,\quad A^{k-2} \times B,\quad \ldots\ ,\quad A \times B,\quad B\right]$.

At the second stage, the checkword Ψ must be sent to the input of the recursive LFSM so that the LFSM will transferred from state $S(k)$ to final state $S(n)$ during r time clocks:

$$S(n) = A^r \times S(k) + L_r \times \Psi,\ GF(q). \qquad (21)$$

Since after completing the encoding procedure, the LFSM should again return to its original state (i. e. $S(n) = S(0)$), therefore, equality (21) can be written as:

$$L_r \times \Psi = A^r \times S(k),\ GF(q).$$

From a mathematical point of view, all LFSMs are equivalent, i. e. give the same encoding and decoding results. However, they can have different software and hardware implementations and, accordingly, different complexity of the encoding procedures.

The fastest LFSM is of type 2 (both left-sided and right-sided). Using this LFSM, the encoding result will be obtained already at the k-th step, but not at the n-th step, as usual.

The LFSM of type 3 and type 4 require the most time costs for encoding.

3.3. The decoding of Reed-Solomon codes based on automaton models. The decoding process of RS codes can also be splited into two stages:

1) establishing the fact of the presence or absence of the errors;

2) determination of error parameters, if any.

From the positions of the automaton representation of RS codes, the first stage consists in calculating the state $S(n)$ into which the LFSM will go after feeding a n-symbol codeword Z to its input according to the recursive formula, which follows from (2):

$$S(j+1) = A \times S(j) + B \times z_j,\ GF(2),\ z_j \in Z,\ j = 1 \div n.$$

The state $S(n)$ is the error syndrome: a zero value of this state indicates the absence of errors in the transmitted codeword within the corrective capacity of the code. If there is a multiplicity error τ in the codeword $Z_{err}^{(\tau)}$, a nonzero error syndrome will be obtained $S_{err}^{(\tau)}$.

It is with such calculations that the main property of the cyclic code (including the RS codes) – the cyclic property will be used to the maximum extent.

In [9], specific methods of encoding and decoding based on automaton models of cyclic and RS codes are considered.

3.4. Hardware implementation of Reed-Solomon codes based on automaton models. Let's consider now the features of the hardware implementation of the LFSM. The operations of automatic encoding and decoding are based on transition functions (1) and output functions (2).

The directly implementation of these functions can be replaced with the shift operation and several arithmetic operations. Consequently, a shift register with linear feedback can be chosen as the hardware implementation of the LFSM.

The number of adders and multipliers in the field $GF(q)$ is determined by the number of corresponding elementary operations in the mathematical LFSM model. The simplest hardware implementation has LFSM of type 1.

3.5. The modelling of Reed-Solomon codes by help the quantum computer. RS codes are enough complex codes and the complexity of computations will increase in proportion to the length n of the code and the number of errors τ to be corrected.

It is possible to significantly reduce the complexity and duration of encoding and decoding operations using parallel data processing. It is interesting to explore the impact of parallelism depending on the architecture of the computer.

It is known that the basis of a quantum computer (more precisely, the existing models of a quantum computer) is a r-symbol quantum register [10, 11]. LFSM of any type is also implemented in hardware on the r-symbol register. Both registers store the state word $S(t)$ (Fig. 1).

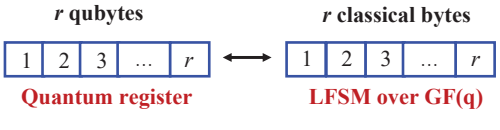

Fig. 1. The conformity between quantum register and linear finite-state machine (LFSM)

The fundamental difference between the two computation methods is as follows.

In classical computations, only one LFSM state is recursively computed at each time step, which leads to an exponential increase in computational complexity even when using traditional parallelism.

Thanks to principle of quantum superposition and using linear algebra, the quantum register can be located in all 2^r basic states at once. Such conditions are provided by the automaton model of codes. As a result of quantum parallelism, it is possible to get the result of calculations in one clock step.

The practical benefits of quantum parallelism can be obtained even now, in the absence of a full-fledged quantum computer. Consider the following task. Let the non-zero error syndrome $S_{err}^{(\tau)}$ be obtained with the help of a classical

computer. Further, in order to find errors, it is possible to the software modelling based on a quantum computer.

Let's note that quantum computations are usually considered based on individual qubits, which corresponds to binary cyclic codes in Galois fields $GF(2)$. RS codes operate in non-binary fields $GF(2^m)$, which requires the grouping of qubits, for example, into qubytes [12].

Let's clarify that classical RS codes only are considered in the paper. To remove the errors in the quantum computer itself, quantum RS codes are needed.

4. Conclusion

The article shows that the automaton representation of the RS codes is the most suitable for these codes, since it takes into account the cyclicity property and other features of the RS codes as much as possible. The generator polynomial provides a close relationship between the automaton and other methods of describing cyclic codes, which makes it easy to switch from one method to another.

The recursive LFSMs of eight types and the non-recursive LFSMs of four types have been proposed, their mathematical properties have been analysed from the standpoint of error-correcting coding. The LFSMs of all types give the same result in encoding and decoding, but with different complexity. The fastest is LFSM type 2, and LFSM type 1 has the smallest hardware implementation.

The future of error-correcting codes is determined by their adaptability to various computer architectures. Thanks to the automaton representation of RS codes, it is easy to implement the encoding and decoding of these codes on a quantum computer. With the help of quantum parallelism, it is possible to radically speed up the decoding process of classical RS codes.

At the decision of dilemma old/new code, it is not to reject the well-known codes immediately. The history of technology has already given many examples of re-using technical solutions in new conditions, for example, when a new circuitry base appears. One more argument: economy of resources, the possibility of using the existing communication infrastructure based on traditional error correcting codes. The above considerations touch the RS codes to a full degree. These codes and today included in three of the most widespread error correcting codes, although not all of their reserves have been exhausted. RS codes will be in demand for future generations of computers.

References

1. Luzhetskyi, V. Semerenko, V. (2019). Automaton Presentations of Reed-Solomon Codes. *Advanced Information and Communication Technologies*. Lviv, 50–53. doi: http://doi.org/10.1109/aiact.2019.8847892
2. Sklar, B. (2001). *Digital Communications. Fundamentals and Applications*. Los Angeles: Prentice Hall PTR.
3. Morelos-Zaragosa, R. H. (2002). *The Art of Error Correcting Coding*. Jon Wiley & Sons. doi: http://doi.org/10.1002/0470847824
4. Semerenko, V. (2018). Automaton Presentations of Cyclic Codes. *Herald of Vinnytsia Polytechnical Institute, 2 (137)*, 89–100.
5. Ifeachor, E. C., Jervis, B. W. (2002). *Digital Signal Proccesing*. A practical Approach. Los Angeles: Prentice Hall, 925.
6. Friedland, B. (1959). Linear Modular Sequential Circuits. *IRE Transactions on Circuit Theory, 6 (1),* 61–68. doi: http://doi.org/10.1109/tct.1959.1086529
7. Gill, A. (1967). *Linear Sequential Circuits. Analysis, Synthesis and Application*. New York, London: McGraw-Hill Book Company, 215.
8. Milovanovic, E., Stojcev, M., Milovanovic, I., Nikolic, T. (2015). Concurrent Generation of Pseudo Random Numbers with LFSR of Fibonacci and Galois Type. *Computing and Informatics, 34,* 941–958.
9. Semerenko, V. P. (2011). Dekodirovanie kodov Rida-Solomona na osnove grafovoi i avtomatnoi modelei. *Elektronnoe modelirovanie, 1,* 57–72.
10. Solovyev, V. M. (2015). Quantum Computers and Quantum Algorithms. Part 1. Quantum Computers. *Izvestiya of Saratov University. New Series. Series: Mathematics. Mechanics. Informatics, 15 (4),* 462–477. doi: http://doi.org/10.18500/1816-9791-2015-15-4-462-477
11. Calderbank, A. R., Rains, E. M., Shor, P. M., Sloane, N. J. A. (1998). Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory, 44 (4),* 1369–1387. doi: http://doi.org/10.1109/18.681315
12. Häffner, H., Hänsel, W., Roos, C. F., Benhelm, J., Chek-al-kar, D., Chwalla, M. et. al. (2005). Scalable multiparticle entanglement of trapped ions. *Nature, 438 (7068),* 643–646. doi: http://doi.org/10.1038/nature04279

Semerenko Vasyl, *PhD, Associate Professor, Department of Computer Technique, Vinnytsia National Technical University, Ukraine, e-mail: VPSemerenko@ukr.net, ORCID: http://orcid.org/0000-0001-8809-1848*