Title: Extended class of linear feedback shift registers

Author: Ireneusz Gościniak

Citation style: Ireneusz Gościniak. (2016). Extended class of linear feedback shift registers. W: "2016 International Conference on Signals and Electronic Systems (ICSES) : 5-7 September 2016, Krakow" (S. 23-28). Kraków : Centrum Informatyki AGH, doi 10.1109/ICSES.2016.7593813

Extended Class of Linear Feedback Shift Registers

Ireneusz Gosciniak
Institute of Computer Science, University of Silesia
Będzińska 39, 41-200 Sosnowiec, Poland
Email: ireneusz.gosciniak@us.edu.pl, Telephone: +48694366102

***Abstract*—Shift registers with linear feedback are frequently used. They owe their popularity to very well developed theoretical base. Registers with feedback of prime polynomials are of particular practical importance. They are willingly applied as test sequence generators and test response compactors. The article presents an attempt to extend the class of registers with linear feedback. Basing on the formal description of the register, the algorithms of register transformation are proposed. It allows to obtain the registers with equivalent graphs.**

Keywords*—*LFSR*; *register perturbation*; *entropy*; *equivalent form of graphs*; *algorithms of transformation

I. INTRODUCTION

Registers with linear feedback realized by the prime polynomial are of particular importance – they are widely discussed in the literature. LFSR registers are used in circuits testing as test generators (they generate pseudo-random sequences) and test response compactors. They are also widely used in cryptography and hence they have many practical applications in the field of telecommunication: pseudo-random sequence generators, cyclic coding, convolutional coding, scrambling, reed-solomon coding or reed-solomon decoding. A formal description of LFSR allows to extend the class of these registers. The operation of LFSR can be shown basing on the information theory and this approach is presented in the section II on the article. The maintenance of proper correlation between streams creating feedback is of particular importance in the implementation of transformation algorithms. A formal description of these registers allows to realize transformation algorithms – it is presented in the section III. The proposed approach allows to extend the class of registers with linear feedback. The scheme of LFSR is shown in Fig. 1.

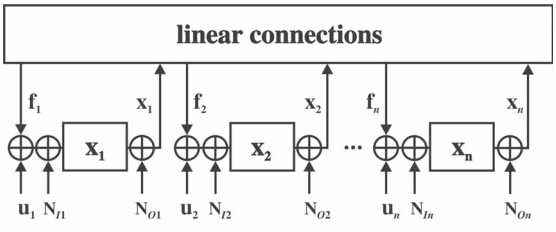


Fig. 1. LFSR model.

Registers with linear feedback (linear registers) and external input $U(t)$ – Multi-Input Shift Registers (MISR) can be described as follows:

$$X\left(t+1\right)=N_I\oplus\left(T\cdot\left(N_O\oplus X\left(t\right)\right)\oplus U\left(t\right)\right), \tag{1}$$

or

$$\begin{cases} x_1\left(t+1\right) &= \mathrm{N}_{I1}\oplus\left(\left(a_{1,1}\cdot\left(\mathrm{N}_{O1}\oplus x_1\left(t\right)\right)\oplus\ldots\right.\right.\\ &\quad\left.\left.\oplus\ a_{1,n}\cdot\left(\mathrm{N}_{On}\oplus x_n\left(t\right)\right)\right)\oplus u_1\left(t\right)\right)\\ x_2\left(t+1\right) &= \mathrm{N}_{I2}\oplus\left(\left(a_{2,1}\cdot\left(\mathrm{N}_{O2}\oplus x_2\left(t\right)\right)\oplus\ldots\right.\right.\\ &\quad\left.\left.\oplus\ a_{2,n}\cdot\left(\mathrm{N}_{On}\oplus x_n\left(t\right)\right)\right)\oplus u_2\left(t\right)\right),\\ &\vdots\\ x_n\left(t+1\right) &= \mathrm{N}_{In}\oplus\left(\left(a_{n,1}\cdot\left(\mathrm{N}_{On}\oplus x_n\left(t\right)\right)\oplus\ldots\right.\right.\\ &\quad\left.\left.\oplus\ a_{n,n}\cdot\left(\mathrm{N}_{On}\oplus x_n\left(t\right)\right)\right)\oplus u_n\left(t\right)\right)\end{cases} \tag{2}$$

where:

$X(t)$ – vector of size n that describes the state of register – initial state results from the planned assumptions for initial value.

N_O, N_I – vector of output/input negation at the size n; realization of negation is possible by means of the one XOR gate.

T – matrix describing connections between D flip-flops (DFFs) with the size of $n \times n$; n – number of bits of the register (the number of DFFs). The connection is created from $column\ index$ to $row\ index$ (from $\mathrm{DFF}_{column\ index}$ to $\mathrm{DFF}_{row\ index}$). This matrix is a formal description of the scheme of connections between flip-flops.

$U(t)$ – vector of external input; perturbing stream.

The register without any external vector $U(t)$ is an autonomous register – Linear Feedback Shift Register described by (3):

$$X\left(t+1\right)=N_I\oplus T\cdot\left(N_O\oplus X\left(t\right)\right). \tag{3}$$

(4) describes matrix of connection between flip-flops.

$$\begin{array}{ccccc} a_{n,n} & a_{n,1} & \cdots & a_{n,n} & a_{n,1}\\ a_{1,n} & a_{1,1} & \cdots & a_{1,n} & a_{1,1}\\ a_{2,n} & a_{2,1} & \cdots & a_{2,n} & a_{2,1}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ a_{n-1,n} & a_{n-1,1} & \cdots & a_{n-1,n} & a_{n-1,1}\\ a_{n,n} & a_{n,1} & \cdots & a_{n,n} & a_{n,1}\\ a_{1,n} & a_{1,1} & \cdots & a_{1,n} & a_{1,1} \end{array} \tag{4}$$

Connections between D flip-flops are not marked in fig. 1 (symbol $\oplus$ represents a function XOR). The connections between the D flip-flops (including feedback) are implemented by the the matrix T. (1) describes the MISR. Because

978-1-5090-2667-8/16/$31.00 ©2016 IEEE

the vector U is equal to zero in (3) (vector U is omitted) then (3) performs the autonomous register – the LFSR. The register with linear feedback is any register due to the complexity of the connections that can be implemented by a matrix T. Matrix T can implement complex circuits – e.g. by perturbing registers discussed in the next section of the paper. Vectors N_I and N_O implement internal negation and their value does not change during operation of the register. Values of negation vectors ($N_I(t)$ and $N_O(t)$) can be changed during operation of the register in the same way as the perturbation vector $U(t)$ – it gives interesting possibilities of implementation (e.g. in cryptography).

In a typical approach to the description of the register, predefined connection schemes are implemented by means of using the matrix ($a_{i,k} = 1$ when connection is realized $i \in \{1, 2, ..., n\}$ and $k \in \{1, 2, ..., n\}$ – TRUE is represented by 1 in equations and FALSE is represented by 0 for better presentation):

$a_{i+1,i}$, $a_{n,n}$ – they implement connections of a cyclic shift register $i \in \{1, 2, ..., n-1\}$ (see (5) matrix T_I),

$a_{i,i+k-1}$ – it involves a feedback of k flip-flops group (when $k = 1$ it transforms D to T flip-flop) and $i \in \{1, 2, ..., n-k+1\}$ (see (5) matrix T_{II}: $i = 1$ and $k = 3$; matrix T_{III}: $i = 3$ and $k = 1$),

$a_{1,i}$ – it realizes an external feedback $i \in \{1, 2, ..., n-1\}$ (see (5) matrix T_{II}),

$a_{i,n}$ – it realizes an internal feedback $i \in \{2, 3, ..., n\}$ (see (5) matrix T_{III}).

In the case of mixed feedback (internal and external) the rule $a_{i+1,i} = a_{1,i} \wedge a_{i+1,n}$ should be realized (see (5) matrix T_{IV}). The criterion of one type of feedback for one flip-flop can be defined as follows: $a_{1,i} \wedge a_{i+1,n} = 0$. In this way a generalized class of registers with linear feedback is defined [1].

$$T_I = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad T_{II} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$
$$T_{III} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad T_{IV} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5}$$

The remaining connections realize the forward connections:

$a_{i+1+k,i}$ – it involves a forward connection of k flip-flops group $i \in \{1, 2, ..., n-k-1\}$.

There are corresponding cells in the matrix, indices of which can be calculated as follows: $k' - k = i - i'$ and $k + i = k' + i'$ ([2]). When a calculated index is greater than n, it performs a division $\bmod n$ (see (4)). The above presented relation- ship allows to formulate transformations of matrix T.

Transformations of the register with linear feedback are implemented by feedback transformation while the correct correlation between streams is maintained. Transformations presented in the report [3] are based on patent [4]. At this point, the importance of this subject can highlight, that it inspired the development of the patent.

Registers of equivalent graph have different states in the corresponding nodes. Paper [2] gives a transformations of LFSRs to Bottom-Top and Top-Bottom registers (the generalization of this class is shown in [1] as well as the generalization of the LFSRs class. The algorithm of LFSR transformation to canonical form (external LFSR) is presented in the work [5]. This transformation shows how to maintain a proper correlation streams during their movement. Each step of the register transformation creates the register with the equivalent graph. A related issue is the reconfiguration of LFSR realized by switching feedback [6]. The proposed transformations extend significantly the approach to the register reconfiguration.

II. REGISTER PERTURBING REGISTER

The simple connection of two registers is shown in Fig. 2. A 4-bits register of maximum cycle perturbs the 4-bits cyclic serial register. (6) shows as follows: the matrix T describing connections, the initial state, vectors of negation.

$$T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$
$$X(0) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, N_O = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, N_I = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

cycle: 1, 2, 4, 9, 3, 6, 13, 10, 5, 11, 7, 15, 14, 12, 8.

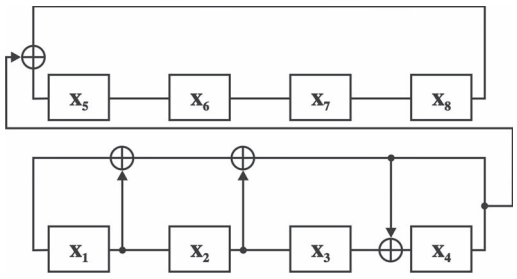


Fig. 2. The simple feedback shift register perturbed by LFSR with cycle of maximum length.

The result can be justified as follows (basing on the information theory [7]). $X(t)$ is the state of the register at the instant t and $U(t)$ is the state of the perturbing register just before the transition to the next state $X(t+1)$ ($X(0)$ is a known initial state of the register). Entropy of the register X

is not higher than the sum of conditional entropy of the input perturbing signal U. The necessary condition of the increase in entropy of the perturbing register is the lack of correlation between the perturbation and the state of the register, i.e. the entropy of the perturbation and the state of the register must not be zero.

A set of consecutive states in a graph is a path when for all $U(t)$ the function $X(t+1) = f(X(t), U(t))$ satisfies the dependency $X(t+1) \neq X(t)$. It clearly states that the path is a sub-graph. If the perturbation $U(t)$ is independent on the state of register $X(t)$ and a n-bit register is perturbed by p-bit register (where $p \leq n$ and the analyzed number of states equals $2^p - 1$ – for LFSR) then the increase in the register entropy is the same as in the perturbing register. Basing on it, the entropy of register X will not be greater than the sum of the conditional entropy of the input perturbing signal $U(t)$ and it can reach a maximum value.

If the entropy of n-bit register perturbed by p-bit register of X (where $p \leq n$ and the analyzed number of states equals $2^p - 1$ – for LFSR) reaches its maximum value then the transition graph describing the behavior of the register constitutes a path of the graph of the maximum length.

The entropy reaches the maximum value if and only if the probability of each state has the same value (property of entropy [7]). It occurs when there are all possible register states $2^p - 1$. In this case, each node of the graph has only one output – it creates a path of graph.

The situation is complicated when both registers perturb each other. These are two registers MISR, the operation of each of them is disturbed by stream $U(t)$ – the physical connections and the model implemented by the formal description are presented. Such an example is shown in Fig. 3. The matrix of connections and the initial state are shown in (7). This autonomous register has a transition graph with maximum cycle – 255 states:

cycle: 17, 163, 215, 47, 199, 15, 135, 159, 182, 124, 241, 98, 213, 42, 76, 145, 178, 116, 248, 232, 200, 136, 24, 41, 202, 141, 147, 183, 255, 102, 221, 35, 198, 140, 16, 32, 64, 144, 49, 227, 71, 30, 36, 72, 153, 187, 254, 229, 74, 156, 48, 96, 208, 161, 210, 164, 89, 58, 108, 209, 34, 69, 27, 175, 214, 172, 80, 176, 113, 115, 118, 253, 99, 86, 189, 243, 103, 94, 180, 121, 122, 252, 224, 193, 2, 5, 139, 158, 53, 235, 78, 148, 57, 234, 205, 3, 134, 28, 33, 195, 7, 142, 21, 171, 222, 165, 218, 173, 211, 39, 206, 133, 154, 61, 226, 196, 137, 155, 190, 117, 123, 127, 119, 126, 244, 233, 75, 31, 167, 223, 38, 77, 18, 37, 203, 14, 4, 8, 9, 138, 29, 162, 84, 184, 120, 249, 107, 95, 55, 238, 197, 10, 12, 1, 131, 151, 191, 246, 236, 192, 129, 146, 52, 104, 217, 43, 207, 6, 13, 130, 20, 40, 73, 26, 44, 65, 19, 166, 92, 177, 242, 228, 201, 11, 143, 150, 60, 97, 83, 54, 109, 82, 181, 250, 237, 67, 22, 45, 194, 132, 25, 170, 93, 50, 101, 91, 63, 231, 79, 23, 174, 85, 59, 239, 70, 157, 179, 247, 111, 87, 62, 100, 216, 168, 88, 185, 251, 110, 212, 169, 219, 46, 68, 152, 56, 105, 90, 188, 112, 240, 225, 66, 149, 186, 125, 114, 245, 106, 220, 160, 81, 51, 230, 204, 128.

In this case, the entropy can reach the value H(X, U) ≤ H(X)+H(U). The equality occurs if and only if X and U are independent.

The above presented discussion can be generalized to the shift registers with the non-linear feedback. The similar discussion is carried out in the work [8] for a complex testing structure. Schemes 2 and 3 are the connections of two MISR registers, their description is realized by (3) with matrices of connections 6 and 7 – they play a role of LFSRs.

$$T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, X(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (7)$$

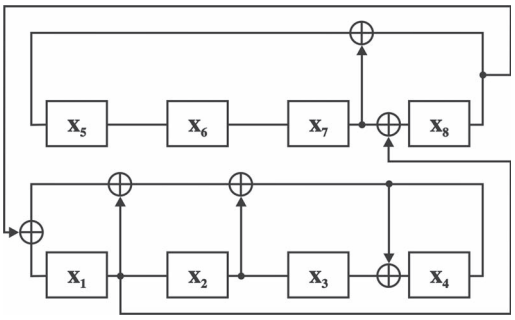


Fig. 3. LFSRs perturbed each over.

The extended class of registers with linear feedback includes all possible registers with linear feedback. 2688 matrices with different connections are determined for 4-bit register with maximum cycle in the extended class of registers with linear feedback. The internal negation greatly expands the possibilities to obtain equivalent graphs [9]. Taking into account the negation vectors (3) 43008 registers for maximum cycle were obtained.

III. TRANSFORMATIONS OF REGISTERS

The simple algorithm (1) of registers transformation that allows to obtain equivalent graphs is described below. Such a subset of registers can be selected in the extended class of registers with linear feedback. The example of such a transformation is shown in Fig. 4 and (8).

$cycle_A$: 1, 3, 6 13, 2, 5, 11, 15, 7, 14, 4, 8, 9, 10, 12;

$cycle_B$: 1, 3, 5, 9, 2, 6, 12, 11, 4, 10, 7, 15, 14, 13, 8;

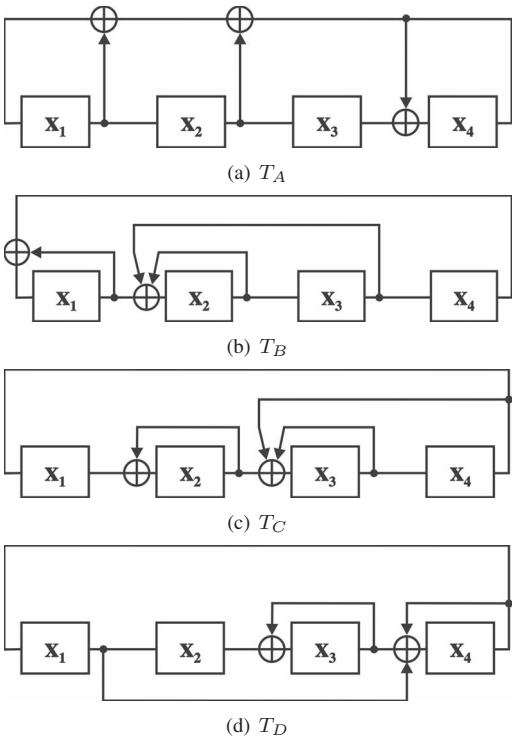
$cycle_C$: 1, 2, 6, 10, 3, 4, 12, 9, 7, 8, 5, 14, 15, 13, 11;

$cycle_D$: 1, 10, 13, 15, 11, 7, 2, 4, 12, 5, 6, 8, 9, 3, 14.

Algorithm 1: Equivalent form of matrix T.

Input: T – matrix of connections; n – dimension.
Data: c – column vector; r – row vector; cbind() binds vector of column and matrix; rbind() binds vector of row and matrix; array() creates matrix (or vector) of given dimensions and initial value.
Output: Equivalent form of matrix T.

$c \longleftarrow \text{array}(FALSE, dim = n)$;
$r \longleftarrow \text{array}(FALSE, dim = n+1)$;
$T \longleftarrow \text{cbind}(T, c)$;
$T \longleftarrow \text{rbind}(T, r)$;
for i in $n:1$ **do**
  for j in $1:n$ **do**
    $T[i+1, j+1] \longleftarrow T[i, j]$;
$T[1:n+1, 1] \longleftarrow T[1:n+1, n+1]$;
$T[1, 1:n] \longleftarrow T[n+1, 1:n]$;
$T \longleftarrow T[1:n, 1:n]$;
return T;



(a) T_A (b) T_B (c) T_C (d) T_D

Fig. 4. LFSRs transformation.

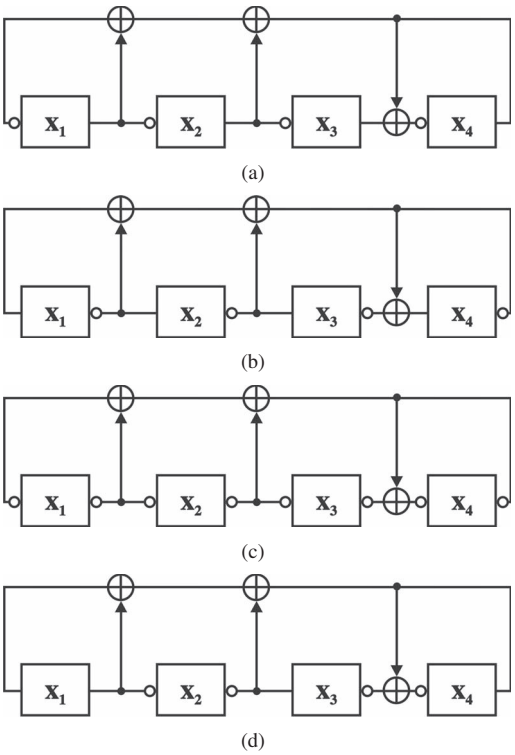
$$T_A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad T_B = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$
$$T_C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad T_D = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}. \tag{8}$$

By implementing the negation of inputs and outputs to the register described by the matrix T_A the following equivalent transition graphs are obtained:

$cycle_a$: 1, 4, 15, 0, 7, 9, 13, 5, 12, 6, 10, 11, 8, 14, 3;
$cycle_b$: 1, 12, 14, 11, 0, 15, 8, 6, 2, 10, 3, 9, 5, 4, 7;
$cycle_c$: 1, 11, 7, 6, 5, 3, 14, 12, 9, 2, 13, 10, 4, 0, 8;
$cycle_d$: 1, 9, 0, 10, 6, 7, 4, 2, 15, 13, 8, 3, 12, 11, 5;

Fig. 5 presents schemes of registers with internal negation (symbol $\circ$ represents the negation). (9) shows the vectors of negation of (3) describing the operation of registers.

$$N_{Ia} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, N_{Oa} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad N_{Ib} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, N_{Ob} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$
$$N_{Ic} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, N_{Oc} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad N_{Id} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, N_{Od} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \tag{9}$$



(a) (b) (c) (d)

Fig. 5. LFSRs with internal negation.

Algorithm 1 shifts cells of register. It creates equivalent registers (Fig. 4). While, algorithm 2 changes the order of cells. (10) shows the collection of matrices obtained for the 4-bit register. Cycles of these registers are presented below:

$cycle_1$: 1, 2, 4, 9, 3, 6, 13, 10, 5, 11, 7, 15, 14, 12, 8;
$cycle_2$: 1, 2, 8, 5, 3, 10, 13, 6, 9, 7, 11, 15, 14, 12, 4;
$cycle_3$: 1, 4, 2, 9, 5, 6, 11, 12, 3, 13, 7, 15, 14, 10, 8;

Algorithm 2: Reconfiguration of matrix T.

Input: T – matrix of connections; n – dimension.
Data: s – vector of sequence; map – array of permutations; $listT$ – collection of reconfigured matrices; c() creates vector; $list()$ creates list; permutations() enumerate the possible permutations of vector; nrow() returns number of rows; array() see algorithm 2.
Output: Collection of transformed matrices.

```
1  s ⟵ c(1 : n);
2  map ⟵ permutations(s);
3  listT ⟵ list();
4  for k in 1 : nrow(map) do
5      T_D ⟵ array(FALSE, dim = c(n, n));
6      for i in n : 1 do
7          for j in 1 : n do
8              if T[i, j] then
9                  T_D[map[k, i], map[k, j]] ⟵ TRUE;
10     listT[k] ⟵ T_D;
11 return listT;
```

$cycle_4$: 1, 4, 8, 3, 5, 12, 11, 6, 9, 7, 13, 15, 14, 10, 2;
$cycle_5$: 1, 8, 2, 5, 9, 10, 7, 12, 3, 13, 11, 15, 14, 6, 4;
$cycle_6$: 1, 8, 4, 3, 9, 12, 7, 10, 5, 11, 13, 15, 14, 6, 2;
$cycle_7$: 1, 4, 10, 3, 5, 14, 9, 6, 11, 7, 15, 13, 12, 8, 2;
$cycle_8$: 1, 8, 6, 3, 9, 14, 5, 10, 7, 11, 15, 13, 12, 4, 2;
$cycle_9$: 1, 10, 6, 5, 11, 12, 3, 14, 7, 15, 13, 9, 8, 2, 4;
$cycle_{10}$: 1, 2, 4, 8, 3, 6, 12, 11, 5, 10, 7, 14, 15, 13, 9;
$cycle_{11}$: 1, 6, 10, 9, 7, 12, 3, 14, 11, 15, 13, 5, 4, 2, 8;
$cycle_{12}$: 1, 2, 8, 4, 3, 10, 12, 7, 9, 6, 11, 14, 15, 13, 5;
$cycle_{13}$: 1, 2, 12, 5, 3, 14, 9, 6, 13, 7, 15, 11, 10, 8, 4;
$cycle_{14}$: 1, 8, 6, 5, 9, 14, 3, 12, 7, 13, 15, 11, 10, 2, 4;
$cycle_{15}$: 1, 12, 6, 3, 13, 10, 5, 14, 7, 15, 11, 9, 8, 4, 2;
$cycle_{16}$: 1, 4, 2, 8, 5, 6, 10, 13, 3, 12, 7, 14, 15, 11, 9;
$cycle_{17}$: 1, 6, 12, 9, 7, 10, 5, 14, 13, 15, 11, 3, 2, 4, 8;
$cycle_{18}$: 1, 4, 8, 2, 5, 12, 10, 7, 9, 6, 13, 14, 15, 11, 3;
$cycle_{19}$: 1, 2, 12, 9, 3, 14, 5, 10, 13, 11, 15, 7, 6, 4, 8;
$cycle_{20}$: 1, 4, 10, 9, 5, 14, 3, 12, 11, 13, 15, 7, 6, 2, 8;
$cycle_{21}$: 1, 12, 10, 3, 13, 6, 9, 14, 11, 15, 7, 5, 4, 8, 2;
$cycle_{22}$: 1, 8, 2, 4, 9, 10, 6, 13, 3, 12, 11, 14, 15, 7, 5;
$cycle_{23}$: 1, 10, 12, 5, 11, 6, 9, 14, 13, 15, 7, 3, 2, 8, 4;
$cycle_{24}$: 1, 8, 4, 2, 9, 12, 6, 11, 5, 10, 13, 14, 15, 7, 3;

Scheme of register implemented by the matrix T_{15} is shown in Fig. 6 – connections are described by the proper indices of the matrix elements.

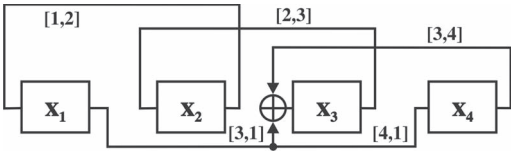


Fig. 6. The implementation of T_{15} matrix of connections.

$$
T_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix},
$$

$$
T_3 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad T_4 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},
$$

$$
T_5 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad T_6 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},
$$

$$
T_7 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad T_8 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},
$$

$$
T_9 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad T_{10} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},
$$

$$
T_{11} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad T_{12} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix},
$$

$$
T_{13} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad T_{14} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},
$$

$$
T_{15} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad T_{16} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix},
$$

$$
T_{17} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad T_{18} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},
$$

$$
T_{19} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad T_{20} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},
$$

$$
T_{21} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad T_{22} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix},
$$

$$
T_{23} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad T_{24} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \tag{10}
$$

Transformations of matrix are useful for the creating registers of equivalent transition graphs. It can be of practical importance for long registers. Thanks to the reconfigurable hardware technology used in the FPGA just such algorithms become meaningful.

IV. CONCLUSIONS

The physical connections and the model implemented by the formal description is presented in the article by means of schemes and matrices. The foregoing discussion allows to extend the class of LFSRs to registers described by (1). The condition necessary to get the register with maximum cycle is the obtaining the proper correlation of streams in feedback. In this subgroup, there are registers with feedback of the prime polynomial. The transformations of the registers can be particularly useful in the field of testing and in communication devices. The proposed solution can be used in reconfigurable h ardware. T ransformations presented in the article do not exhaust all possibilities of implementing transformations. The discussion presented in the article extends the possibilities of LFSRs use in the circuit testing and cryptography.

REFERENCES

[1] I. Gosciniak, "Linear Registers with Mixed Feedback, in Polish; Rejestry liniowe z mieszanym sprzężeniem zwrotnym," *Pomiary Automatyka Kontrola*, no. 1, pp. 4–6, 1996.

[2] K. Iwasaki, "Analysis and proposal of signature circuits for LSI testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 1, pp. 84–90, 1988.

[3] L.-T. Wang, N. Touba, R. Brent, H. Xu, and H. Wang, "On Designing Transformed Linear Feedback Shift Registers with Minimum Hardware Cost – Technical Report," Computer Engineering Research Center Department of Electrical & Computer Engineering The University of Texas at Austin, 2011.

[4] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Method for Synthesizing Linear Finite State Machines," U.S. Patent, No. 6,353,842, 2002.

[5] I. Gosciniak, "Equivalent Form of Linear Feedback Shift Registers," in *XIXth National Conference Circuit Theory and Eletronic Networks*, 1996, pp. 115–120.

[6] L. Alaus, D. Noguet, and J. Palicot, "A Reconfigurable LFSR for Tri-standard SDR Transceiver, Architecture and Complexity Analysis," in *Digital System Design Architectures, Methods and Tools, 2008. DSD '08. 11th EUROMICRO Conference on*. IEEE Computer Society, 2008, pp. 61–67.

[7] R. Ash, *Information Theory*. John Wiley & Sons, 1967.

[8] M. Kopec, "Can Nonlinear Compactors Be Better than Linear Ones?" *IEEE Trans. Comput.*, no. 11, pp. 1275–1282, 1995.

[9] A. Gucha and L. Kinney, "Relating the Cyclic Behaviour of Linear Intrainverted Feedback shift Registers," *IEEE Transactions on Computers*, vol. 41, no. 9, pp. 1088–1100, 1992.