Constructing a Ternary FCSR with a Given Connection Integer

Lin Zhiqiang[1,2] and Pei Dingyi[1,2]

[1] School of Mathematics and Information Sciences, Guangzhou University, China
[2] State Key Laboratory of Information Security, Chinese Academy of Science, China

Abstract. FCSRs have been proposed as an alternative to LFSRs for the design of stream ciphers. In 2009, a new ”ring” representation of FCSRs was presented. This new representation preserves the statistical properties and circumvents the weaknesses of the Fibonacci and the Galois FCSRs. Moreover an extension of the ring FCSRs called ternary FCSRs has been proposed. They are suitable for hardware and software implementations of FCSRs. In this paper, we show a method of constructing a ternary FCSR with a given connection integer for hardware implementation. The construction is simple and convenient. And the ternary FCSRs we get are able to meet the hardware criteria.

Keywords: stream cipher, 2-adic integer, FCSRs, ring FCSRs, l-sequences

1 Introduction

Feedback with Carry Shift Registers (FCSRs) were introduced by Klapper and Goresky in [1]. They are very similar to classical Linear Feedback Shift Registers (LFSRs) used to generate pseudo-random sequences. According to their nonlinearity, FCSRs have been suggested as an alternative to LFSRs for avoiding the drawback of linear structure. The mathematical model for the generated sequences of FCSRs is the one of rational 2-adic numbers. It can be used to prove several interesting properties such as proven period, non-degenerated states, and good statistical properties [1∼4].

FCSRs have two traditional representations. They are the Fibonacci and the Galois representations [5]. However, FCSRs are usually implemented by hardware and software using the Galois representation since the Fibonacci mode is not suitable for cryptographic applications [6]. By using a filter on the cells of the Galois FCSR automaton, family of hardware stream ciphers based on FCSRs: F-FCSRs[7∼10] and family of software stream ciphers based on FCSRs: X-FCSRs[11,12] were proposed for stream cipher design. Unfortunately, these stream ciphers were exposed to a very powerful attack by LFSRization of them [13,14].

In [10,12], Arnault et al. have introduced a new FCSR representation called a ring or diversified representation for responding to the LFSRization attack. This new representation is based on the transition matrix of the automaton graph instead of the quadratic transition function in the Galois representation. Many advantages have appeared with this new representation if the transition matrix is well-chosen. For hardware implementations, the criteria are that the critical path length must be equal to 1 and the fan-out must be 2. For software applications, a particular realization suitable for software utilization has been given. This realization uses a specific circuit which acts essentially on binary words. In [15], Arnault et al. have generalized ring FCSRs to 2-adic automata. These automata have been constructed of inputs and outputs, with the entries of matrices in the set of 2-adic integers.

In this paper, we focus on the hardware stream cipher design based on ternary FCSRs, a special kind of 2-adic automata. The criteria to build these automata were presented in [10,15]. Moreover, a ternary FCSR automaton can meet the hardware criteria by well-chosen of the transition matrix. In [15], the authors have proposed an algorithm to get suitable transition matrices. But it is time-consuming because it has to compute the connection integer and test if the connection integer is primitive every time. Therefore an open problem has been presented: How can a diversified or ternary FCSR be constructed when a connection integer is specified?

We solve this problem in this paper. The method of our construction is simple and convenient. It is more efficient than the algorithm above. What's more, the cost of logic gates of a constructed ternary FCSR is less than the one of the Fibonacci or Galois FCSR with the same connection integer. After we finished our work, we found that our work was similar to the method in [16], constructing Ring LFSRs with given connection polynomials. The results of [16] have good application value in the construction of Ring LFSRs, and it seems that our method is useful to the construction of ternary FCSRs. Algorithm 2 in section 3 presents this method, and through the algorithm we also prove a conjecture in [15]: For each given q of size n, there is a transition matrix with a critical path of length 1 and a fan-out 2.

2 Ternary FCSRs

In this section, we briefly introduce some properties of 2-adic integers and ternary FCSRs. Then we present the criteria to build hardware oriented FCSRs.

2.1 2-adic Integers

First, we recall some properties of 2-adic numbers. For more details, the readers could refer to [1].

A 2-adic integer is formally a power series $s = \sum_{i=0}^{\infty} s_i 2^i$ with $s_i \in \{0, 1\}$. The set of 2-adic integers is a Ring denoted by $\mathbb{Z}_2$. Addition and multiplication in $\mathbb{Z}_2$ can be performed by reporting the carries to the higher order terms, i.e., $2^n + 2^n = 2^{n+1}$ for all $n \in \mathbb{N}$. s is a positive integer if there exists an integer K such that $s_n = 0$ for all $n \geq K$. Moreover, any odd integer q has an inverse in $\mathbb{Z}_2$ which can be computed by $q^{-1} = \sum_{n=0}^{\infty} q'^n$, where $q' = 1 - q$. The functions *mod* 2 and *div* 2 are defined on the set $\mathbb{Z}_2$ by:

$$s \; mod \; 2 = s_0$$
$$s \; div \; 2 = \sum_{i=0}^{\infty} s_{i+1}2^i.$$

The following theorems present the relationship between eventually periodic sequences and 2-adic integers.

Theorem 1. *Let $s = \sum_{i=0}^{\infty} s_i 2^i$ be a 2-adic integer, with $s_i \in \{0, 1\}$. Denote $S = (s_i)_{i\in\mathbb{N}}$. Then the sequence S is eventually periodic if and only if there exists two numbers p and q in $\mathbb{Z}$, q odd, such that $s = p/q$. Moreover, S is strictly periodic if and only if $pq \leq 0$ and $|p| \leq |q|$.*

Theorem 2. *Let $s = \sum_{i=0}^{\infty} s_i 2^i = p/q$ with $s_i \in \{0, 1\}$, $pq \leq 0$, $|p| \leq |q|$, q odd and $gcd(p, q) = 1$. Denote $S = (s_i)_{i\in\mathbb{N}}$. Then the sequence S is strictly periodic and the period of S is the order of 2 modulo q, i.e., the smallest integer T such that $2^T \equiv 1(mod \; q)$. The period satisfies $T \leqslant |q| - 1$.*

Definition 1. *An l-sequence is a periodic sequence $S = (s_i)_{i\in\mathbb{N}}$ such that $s = \sum_{i=0}^{\infty} s_i 2^i = p/q$ with $s_i \in \{0, 1\}$, $pq \leq 0$, $|p| \leq |q|$, q a power of an odd prime and the period of S is $\varphi(q)$ where φ denotes the Euler's phi function.*

Obviously, the period of a l-sequence can achieve its maximum possible least period $T = q - 1$ if and only if q is prime and 2 is a primitive root modulo q. In particular, l-sequences are close to m-sequences: known period, good statistical properties, fast generation, etc. [1∼4].

In this paper, we use the notations proposed in [15].

Given a sequence $a = (a(t))_{t\in\mathbb{N}}$ of elements in $\{0, 1\}$, we have

$$\sum_{t\geq t_0} a(t)2^{t-t_0} = a(t_0)2^0 + a(t_0 + 1)2^1 + \cdots$$

in $\mathbb{Z}_2$.

A time dependent vector m in $\{0, 1\}^n$ is denoted at time t by $m(t)^T = (m_0(t), \ldots, m_{n-1}(t))$ where $m(t)$ is a column vector. And we denote

$$M(t_0) = \sum_{t\geq t_0} m(t)^T 2^{t-t_0}$$

in $\mathbb{Z}_2^n$, i.e., $M(t_0) = (M_0(t_0), \ldots, M_{n-1}(t_0))$, where

$$M_i(t_0) = m_i(t_0)2^0 + m_i(t_0 + 1)2^1 + \cdots$$

$0 \leq i \leq n - 1$.

2.2 Diversified FCSRs and Ternary FCSRs

A new FCSR representation called a ring or diversified representation was first introduced in [10] for responding to the attack against the stream ciphers based on Galois FCSRs.

Definition 2. *A diversified FCSR is an automaton composed of a main shift register of n binary cells $m^T = (m_0, m_1, \ldots, m_{n-1})$ and a carry register of n integer cells $c^T = (c_0, c_1, \ldots, c_{n-1})$. It is updated using the following relations:*

$$\begin{cases} m(t+1)^T = Am(t)^T + c(t)^T \ mod\ 2 \\ c(t+1)^T = Am(t)^T + c(t)^T \ div\ 2 \end{cases}$$

where A is a $n \times n$ matrix with entries 0 or 1 in $\mathbb{Z}$, called transition matrix.

A diversified FCSR is a no-input binary 2-adic FSM presented in [15]. Moreover, using the subtracter-with-carry to compute the difference between two 2-adic integers, the authors have introduced an extension of binary 2-adic FSMs, which allows the entries of the matrices in $\{-1, 0, 1\}$. These automata are called ternary 2-adic FSMs. In particular, the no-input ternary 2-adic FSMs called ternary FCSRs are proposed to build hardware oriented FCSRs.

Definition 3. *A ternary FCSR is the same automaton as the one defined in Definition 2.4 except for the entries of the transition matrix A in $\{-1, 0, 1\}$.*

The following two properties of 2-adic FSMs presented in [15], which are also the behaviors of diversified and ternary FCSRs.

Proposition 1. *Consider a diversified or ternary FCSR composed of the main register m, the carry register c and the transition matrix A. We have*

$$M(t_0 + 1) = AM(t_0) + c(t_0)^T$$

Theorem 3. *The series $M_i(t_0)$ $(0 \leq i \leq n-1)$ observed in each cell of the main register are 2-adic expansion of p_i/q with $p_i \in \mathbb{Z}$ and with integer $q = det(I-2A)$. q is called the connection integer of the automaton.*

Theorem 2.7 implies that the transition matrix A completely defines the diversified or ternary FCSR. The Galois and Fibonacci representations are special cases of diversified FCSRs with the following transition matrices A_G and A_F respectively:

$$A_G = \begin{pmatrix} q_1 & 1 & & & \\ q_2 & 0 & 1 & (0) & \\ \vdots & & \ddots & \ddots & \\ q_{n-1} & (0) & & 0 & 1 \\ 1 & & & & 0 \end{pmatrix} \quad A_F = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & (0) & \\ & & \ddots & \ddots & \\ & (0) & & 0 & 1 \\ 1 & q_{n-1} & q_{n-2} & \cdots & q_1 \end{pmatrix}$$

where $|q| = \sum_{i=0}^{n} q_i 2^i$ ($q_0 = -1$, $q_n = 1$, $q_i \in \{0, 1\}$ for $1 \leq i \leq n-1$).

2.3 Hardware Criteria

In this subsection, we introduce the criteria to build hardware oriented FCSRs. The reader can refer to [10,15] for more details. In this paper, we focus on the problem: How to choose a good ternary FCSR for hardware implementation?

We first consider some of the characteristics of the hardware implementation of FCSRs:
Critical path— the critical path length is the maximum number of logic gates the signal has to pass though. If this number is low, the automaton can be clocked

at a higher rate.
Fan-out— the signal of a flip-flop should drive a minimal number of gates as exposed in [17]. Large fan-out makes possible differential power analysis attacks.
Cost— the number of logic gates must be as small as possible to lower consumption and cost of the automaton.

The critical path length, fan-out and cost of logic gates are the basic blocks for FCSRs. Shorter length of critical path, smaller fan-out and lower cost lead to high clock frequencies. These data can be computed from the transition matrix A of the ternary FCSR:
— the critical path length is the smallest integer j such that 2^j is greater or equal to the highest Hamming weight of the rows of A;
— the fan-out is the highest Hamming weight of the columns of A;
— the cost is the Hamming weight of A.

For hardware oriented FCSRs, the shortest of the critical path is 1 and the smallest fan-out is 2. Therefore the requirements of the transition matrix $A = (a_{ij})_{1\leq i,j\leq n}$ of size $n \times n$ are as follows:
— the over diagonal must be full of 1 and $a_{n1} = 1$ (to preserve the shifting);
— the number of nonzero entries in any row or any column must be at most two (to preserve the critical length 1 and the fan-out 2);
— $q = det(I - 2A)$ is prime, and the order of 2 modulo q is $|q| - 1$ (to preserve outputting l-sequences).

To choose suitable transition matrices, an algorithm has been given in [15]. In this algorithm, a matrix A has been constructed in the form of the requirements, then $q = det(I - 2A)$ is tested whether it is prime and whether 2 is the primitive root modulo q. If q can pass the test, then A is a suitable matrix. However, this algorithm is time-consuming, because every time q has to be computed and tested. Hence, at the end of the paper [15], the authors have left an open problem: How can a diversified or ternary FCSR be constructed when a connection integer is specified? Fortunately, we find a method to solve this problem. Through our method, we can immediately construct a suitable A for a given integer q, without any test and complicated computation. This method looks like the one in [16] where the authors proposed a method of constructing Ring LFSRs with transition matrix of the form

$$A = \begin{pmatrix}
 & 1 & & & & & & & \\
 & & 1 & & & & (0) & & \\
 & & & \ddots & & & & & \\
 & & & & 1 & & & & \\
(0) & & & & h_1 & 1 & & & \\
 & & & \iddots & h_2 & & \ddots & & \\
 & & h_{n-4} & \iddots & & & & \ddots & \\
 & h_{n-2} & h_{n-3} & & & & & & 1 \\
1 & h_{n-1} & & & (0) & & & &
\end{pmatrix}$$

for the given connection polynomial $X^n + h_{n-1}X^{n-1} + \cdots + h_1 + 1$ and n odd (the form is similar when n is even). However, these two results are different between the theories and applications of LFSRs and FCSRs. Moreover, the methods of proofs are also different. It seems that our proof is clearer and simpler. Our construction will be showed in the next section.

3 Constructing A with a Given q

This section gives a method of constructing suitable matrices for hardware oriented FCSRs. For a FCSR automaton, the connection integer q is often assumed to be a negative odd integer. Therefore, we consider the following problem:

Given a negative odd integer q, we will construct a matrix $A = (a_{ij})_{1\leq i,j\leq n}$ with entries in $\{-1, 0, 1\}$, meeting the requirements as follows:
— the over diagonal must be full of 1 and $a_{n1} = 1$;
— the number of nonzero elements in each row and each column must be at most two;
— $q = det(I - 2A)$.

First, we introduce a particular binary signed digit representation of a positive integer presented in [18], called the non-adjacent form (NAF).

Definition 4. *A non-adjacent form (NAF) of a positive integer k is an expression $k = \sum_{i=0}^{l-1} k_i 2^i$ where $k_i \in \{-1, 0, 1\}$, $k_{l-1} \neq 0$ and no two consecutive digits k_i are nonzero. l is called the length of the NAF.*

This form has the following properties:
— Any positive integer k has a unique NAF denoted $NAF(k)$.
— $NAF(k)$ has the fewest nonzero digits of any binary signed representation of k.
— The length of $NAF(k)$ is at most one more than the length of the binary representation of k.

$NAF(k)$ can be efficiently computed using the following algorithm proposed in [18]:

Now we present our construction by Algorithm 2. It first computes $NAF(-q)$ of a given negative odd integer q, then Algorithm 2 will output a matrix with order $n = l-1$ where l is the length of $NAF(-q)$. In some cases the $NAF(-q)$ is made by shortening the length to output a smaller matrix with order $n = l - 2$. Interestingly, the Hamming weight of the constructed matrix is less than the one of the transition matrix of the Fibonacci or Galois representation with the same connection integer q, because $NAF(-q)$ has the fewest nonzero digits of any binary signed representation of $-q$. It implies that the cost of logic gates of the constructed ternary FCSR is less.

The following example explains Algorithm 2:

Example 1. $q = -747$

1. Using Algorithm 1, compute: $NAF(747) = (1, 0, -1, 0, 0, 0, -1, 0, -1, 0, -1)$.

Algorithm 1 Computing the NAF of a positive integer

```
Input: A positive integer $k$.
Output: $NAF(k) = (k_{l-1}, k_{l-2}, \ldots, k_1, k_0)$.
$l \leftarrow 0$.
while $k \geq 1$ do
    if $k$ is odd then
        $k_l \leftarrow 2 - (k \ mod \ 4)$, $k \leftarrow k - k_l$;
    else
        $k_l \leftarrow 0$.
    end if
    $k \leftarrow k/2$, $l \leftarrow l + 1$.
end while
return $(k_{l-1}, k_{l-2}, \ldots, k_1, k_0)$.
```

2. Since $q_{10} = 1$, $q_9 = 0$ and $q_8 = -1$, change $NAF(747)$ into $(1, 1, 0, 0, 0, -1, 0, -1, 0, -1)$ and let $n = 9$.

3. Since $q_0 = -1$, construct a matrix A through step 6 to step 17 of Algorithm 2, then

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and $det(I - 2A) = -747$.

Finally, we prove that the matrices constructed by Algorithm 2 meet the requirements proposed at the beginning of this section. First, some properties of determinants should be mentioned.

Proposition 2. *Let G be a matrix over a Ring R of size $n \times n$. Let E_{ij} be the matrix with a single 1 in position (i, j) and other entries are zero. Then $det(G + \lambda E_{ij}) = det(G) + \lambda Cof_{ij}(G)$, where $Cof_{ij}(G)$ denotes the cofactor of (i, j) in the matrix G.*

Proposition 3.3 is a classic property of determinants. It gives a way to compute the determinant of a modified matrix by its original matrix and the cofactors of the original matrix.

Proposition 3. *Let $B = (b_{ij})_{1 \leq i,j \leq n}$ be the matrix*

Algorithm 2 Constructing a Ternary FCSR with a negative odd integer q

```
Input: A negative odd integer $q < -1$.
Output: A matrix $A = (a_{ij})_{1 \le i,j \le n}$.
Using Algorithm 1, compute $NAF(-q) = (q_{l-1}, q_{l-2}, \dots, q_1, q_0)$.
$n \leftarrow l - 1$
if $q_{l-1} = 1$, $q_{l-2} = 0$ and $q_{l-3} = -1$ then
    $q_{l-1} \leftarrow 0$, $q_{l-2} \leftarrow 1$, $q_{l-3} \leftarrow 1$, and $n \leftarrow l - 2$.
end if
for $1 \le i, j \le n$ do
    $a_{ij} \leftarrow \begin{cases} 1 \text{ if } j \equiv i + 1 \; mod \; n \\ 0 \text{ otherwise} \end{cases}$
end for
$m \leftarrow 0$.
$k \leftarrow n - 1$
if $q_0 = -1$ then
    while $k > 1$ do
        if $q_k \ne 0$ then
            $a_{k+m,m+1} \leftarrow q_k$, $m \leftarrow m + 1$.
        end if
        $k \leftarrow k - 1$.
    end while
else
    $a_{nn} \leftarrow 1$.
    while $k > 1$ do
        if $q_k \ne 0$ then
            $a_{k+m,m+1} \leftarrow -q_k$, $m \leftarrow m + 1$.
        end if
        $k \leftarrow k - 1$.
    end while
end if
return $(a_{ij})_{1 \le i,j \le n}$.
```

$$\begin{pmatrix} 1 & -2 & & & & \\ & 1 & -2 & & (0) & \\ (0) & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ b_{i_0 1} & \cdots & b_{i_0 j_0} & & \ddots & \ddots \\ \vdots & \cdots & \vdots & & & \ddots & -2 \\ b_{n1} & \cdots & b_{n j_0} & & (0) & & 1 \end{pmatrix}$$

where $b_{ii} = 1$ for $1 \le i \le n$, $b_{i,i+1} = -2$ for $1 \le i \le n-1$, $b_{ij} \in \mathbb{Z}$ for $i_0 \le i \le n$ and $1 \le j \le j_0$ where i_0, j_0 are two constants $(1 \le j_0 < i_0 \le n)$ and other entries are 0. We can determine some cofactors of B:

$Cof_{kl}(B) = 2^{k-l}$ *for* $1 \le j_0 \le l < k \le i_0 \le n$.

Moreover, if we let $b_{nn} = -1$ and other entries of B be fixed, then:

$Cof_{kl}(B) = -2^{k-l}$ *for* $1 \leq j_0 \leq l < k \leq i_0 \leq n-1.$

Proof. Consider $Cof_{kl}(B)$ $(1 \leq j_0 \leq l < k \leq i_0 \leq n)$, we have

$$Cof_{kl}(B) = (-1)^{k+l} det(B^*_{kl})$$

where B^*_{kl} is a matrix of size $(n-1) \times (n-1)$ by deleting the kth row and the lth column of B, i.e.,

$$B^*_{kl} = \begin{pmatrix} b_{11} & \cdots & b_{1,l-1} & b_{1,l+1} & \cdots & b_{1k} & b_{1,k+1} & \cdots & b_{1n} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ b_{l-1,1} & \cdots & b_{l-1,l-1} & b_{l-1,l+1} & \cdots & b_{l-1,k} & b_{l-1,k+1} & \cdots & b_{l-1,n} \\ b_{l1} & \cdots & b_{l,l-1} & b_{l,l+1} & \cdots & b_{lk} & b_{l,k+1} & \cdots & b_{ln} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ b_{k-1,1} & \cdots & b_{k-1,l-1} & b_{k-1,l+1} & \cdots & b_{k-1,k} & b_{k-1,k+1} & \cdots & b_{k-1,n} \\ b_{k+1,1} & \cdots & b_{k+1,l-1} & b_{k+1,l+1} & \cdots & b_{k+1,k} & b_{k+1,k+1} & \cdots & b_{k+1,n} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ b_{n1} & \cdots & b_{n,l-1} & b_{n,l+1} & \cdots & b_{nk} & b_{n,k+1} & \cdots & b_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} D & & (0) \\ & E & \\ (*) & & F \end{pmatrix}$$

where

$$D = \begin{pmatrix} b_{11} & \cdots & b_{1,l-1} \\ \vdots & \ddots & \vdots \\ b_{l-1,1} & \cdots & b_{l-1,l-1} \end{pmatrix} = \begin{pmatrix} 1 & -2 & & & \\ & 1 & -2 & (0) & \\ & & \ddots & \ddots & \\ & (0) & & 1 & -2 \\ & & & & 1 \end{pmatrix}$$

$$E = \begin{pmatrix} b_{l,l+1} & b_{l,l+2} & \cdots & b_{lk} \\ b_{l+1,l+1} & b_{l+1,l+2} & \cdots & b_{l+1,k} \\ \vdots & \vdots & \dots & \vdots \\ b_{k-1,l+1} & b_{k-1,l+2} & \cdots & b_{k-1,k} \end{pmatrix} = \begin{pmatrix} -2 & & & & \\ & -2 & & (0) & \\ & & \ddots & & \\ & (*) & & \ddots & \\ & & & & -2 \end{pmatrix}$$

$$F = \begin{pmatrix} b_{k+1,k+1} & \cdots & b_{k+1,n} \\ \vdots & \dots & \vdots \\ b_{n,k+1} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} 1 & -2 & & & \\ & 1 & -2 & (0) & \\ & & \ddots & \ddots & \\ & (0) & & 1 & -2 \\ & & & & 1 \end{pmatrix}$$

Then b_{ij} ($i_0 \leq i \leq n$ and $1 \leq j \leq j_0$) are all in the area $(*)$ of B^*_{kl} and E since $1 \leq j_0 \leq l < k \leq i_0 \leq n$. Hence $det(B^*_{kl}) = det(D)det(E)det(F) = (-2)^{k-l}$, and $Cof_{kl}(B) = (-1)^{k+l} \cdot (-2)^{k-l} = 2^{k-l}$ for $1 \leq j_0 \leq l < k \leq i_0 \leq n$.

Moreover, if we let $b_{nn} = -1$ and other entries of B be fixed, then for $1 \leq j_0 \leq l < k \leq i_0 \leq n-1$, $F' = \begin{pmatrix} 1 & -2 & & & \\ & 1 & -2 & (0) & \\ & & \ddots & \ddots & \\ & (0) & & 1 & -2 \\ & & & & -1 \end{pmatrix}$ will be substituted for F in the above proof, and the remaining proof is similar. Therefore, the result is valid.

Proposition 3.4 proposes that some cofactors of B are of the form of 2^k, which is useful for the proof of the following theorem.

Theorem 4. *Given a negative odd integer $q < -1$, the matrix constructed by Algorithm 2 meets the requirements proposed at the beginning of this section.*

Proof. Given a negative odd integer $q < -1$, we get the $NAF(-q)$ or the $NAF(-q)$ with small modification after step 5. Then $q = -2^n - q_{n-1}2^{n-1} - \cdots - q_2 2^2 - q_1 2 - q_0$, and no two consecutive digits q_i $(0 \leq i \leq n-1)$ are nonzero by Definition 3.1.

IF $q_0 = -1$, we initially construct

$$A = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & (0) \\ & & \ddots & \ddots & \\ & (0) & & 0 & 1 \\ 1 & & & & 0 \end{pmatrix}$$

Let $B = (b_{ij})_{1 \leq i,j \leq n} = I - 2A$, then

$$B = \begin{pmatrix} 1 & -2 & & & \\ & 1 & -2 & & (0) \\ & & \ddots & \ddots & \\ & (0) & & 1 & -2 \\ -2 & & & & 1 \end{pmatrix}$$

and $det(B) = -2^n + 1 = -2^n - q_0$.

Now we prove that $det(B) = q$ after step 17 by induction. Suppose $1 \leq k_0 \leq n-1$, $q_{n-1} = q_{n-2} = \cdots = q_{k_0+1} = 0$ and $q_{k_0} \neq 0$. In step 14, A is modified by $a_{k_0 1} \leftarrow q_{k_0}$, then $b_{k_0 1} \leftarrow -2q_{k_0}$. Denote the modified matrix by A_{new}, and $B_{new} = I - 2A_{new}$. By Proposition 3.3, $det(B_{new}) = det(B) - 2q_{k_0}Cof_{k_0 1}(B)$. And by Proposition 3.4, $Cof_{k_0 1}(B) = 2^{k_0 - 1}$. Hence $det(B_{new}) = -2^n - q_{k_0}2^{k_0} - q_0$. Suppose it has got $1 < k < n-1$, $m \geq 0$ and $det(B) = -2^n - q_{n-1}2^{n-1} - \cdots - q_{k+1}2^{k+1} - q_0$ after step 16. Then the algorithm return to step 13 to check if $q_k \neq 0$. If $q_k = 0$, A is not changed. Hence $det(B) =$

$-2^n - q_{n-1}2^{n-1} - \cdots - q_k 2^k - q_0$. If $q_k \neq 0$, In step 14, A is modified by $a_{k+m,m+1} \leftarrow q_k$, then $b_{k+m,m+1} \leftarrow -2q_k$. By Proposition 3.3, after modification we have $det(B_{new}) = det(B) - 2q_k Cof_{k+m,m+1}(B)$. Suppose k' is the least integer such that $k' > k$ and $q_{k'} \neq 0$. Since no two consecutive digits q_i $(0 \leq i \leq n-1)$ are nonzero, we have $k \leq k' - 2$. Then $k + m < k' - 2 + m + 1 = k' + m - 1$ and $m > m - 1$. Hence $Cof_{k+m,m+1}(B) = 2^{k-1}$ by Proposition 3.4 (here $i_0 = k' + m - 1$ and $j_0 = m - 1$), and $det(B_{new}) = -2^n - q_{n-1}2^{n-1} - \cdots - q_k 2^k - q_0$. Therefore, $det(I - 2A) = q$ after step 17 and A has the required form by the construction process.

IF $q_0 = 1$, we initially construct

$$A = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & (0) \\ & & \ddots & \ddots & \\ & (0) & & 0 & 1 \\ 1 & & & & 1 \end{pmatrix}$$

and the remaining proof is similar to the case $q_0 = -1$.

4 Conclusions

In this paper, we have given a method of constructing a ternary FCSR for hardware implementation with a given connection integer q. The cost of logic gates of the constructed ternary FCSR is less than that of the Galois or Fibonacci case. This construction is simple, convenient and useful for hardware oriented FCSRs. It is more efficient than the algorithm presented in [15]. The results of this paper have solved an open problem and a conjecture came up in [15].

References

1. Klapper, A., Goresky, M., 2-adic shift registers, in FSE. Lecture Notes in Computer Science (eds. Anderson, R.J.), New York: Springer, 1993, vol. 809: 174-178.
2. Goresky, M., Klapper, A., Arithmetic cross-correlations of feedback with carry shift register sequences, IEEE Trans. Inf. Theory, 1997, 43(4): 1342-1345.
3. Goresky, M., Klapper, A., Periodicity and distribution properties of combined FCSR sequences, in ETA. Lecture Notes in Computer Science (eds. Gong, G., Helleseth, T., Song, H.Y., Yang, K.), New York: Springer, 2006, vol. 4086: 334-341.
4. Klapper, A., Goresky, M., Large period nearly deBrujin FCSR sequences, in Advances in Cryptology, Lecture Notes in Computer Science (eds. Guillou, Quisquater J.J.), Berlin: Springer, 1995, vol. 921: 263-273.
5. Goresky, M., Klapper, A., Fibonacci and Galois representations of feedback-with-carry shift registers, IEEE Trans. Inf. Theory, 2002, 48(11): 2826-2836.
6. Fischer, S., Meier, W., Stegemann, D., Equivalent representations of the F-FCSR keystream generator. The State of the Art of Stream Ciphers, Workshop Record, 2008.

7. Arnault, F., Berger, T.P., F-FCSR: design of a new class of stream ciphers, in FSE. Lecture Notes in Computer Science (eds. Gilbert, H., Handschuh, H.), New York: Springer, 2005, vol. 3557: 83-97.
8. Arnault, F., Berger, T.P., Lauradoux, C., The FCSR: primitive specification and supporting documentation, ECRYPT - Network of Excellence in Cryptology, 2005. http://www.ecrypt.eu.org/stream/
9. Arnault, F., Berger, T.P., Lauradoux, C., Update on F-FCSR stream cipher. ECRYPT - Network of Excellence in Cryptology, 2006. http://www.ecrypt.eu.org/stream/
10. Arnault, F., Berger, T.P., Lauradoux, C. *et al.*, A new approach for FCSRs, in Selected Areas in Cryptography. Lecture Notes in Computer Science (eds. M.J.J. Jr., Rijmen, V., Safavi-Naini, R.), New York: Springer, 2009, vol. 5867: 433-448.
11. Arnault, F., Berger, T.P., Lauradoux, C. *et al.*, X-FCSR: a new software oriented stream cipher based upon FCSRs, in INDOCRYPT. Lecture Notes in Computer Science (eds. Srinathan, K., Rangan, C.P., Yung, M.), New York: Springer, 2007, vol. 4859: 341-350.
12. Berger, T.P., Minier, M., Pousse, B., Software oriented stream ciphers based upon FCSRs in diversified mode, in INDOCRYPT. Lecture Notes in Computer Science (eds. Roy, B.K., Sendrier, N.), New York: Springer, 2009, vol. 5922: 119-135.
13. Hell, M., Johansson, T., Breaking the F-FCSR-H Stream Cipher in Real Time, in ASIACRYPT. Lecture Notes in Computer Science (eds. Pieprzyk, J.), New York: Springer, 2008, vol. 5350: 557-569.
14. Stankovski, P., Hell, M., Johansson, T., An efficient state recovery attack on X-FCSR-256, in FSE. Lecture notes in computer science (eds. Dunkelman, O.), New York: Springer, 2009, vol. 5665: 23-37.
15. Arnault, F., Berger, T.P., Pousse, B., A matrix approach for FCSR automata, Cryptography and Communications, 2010, vol. 3, Num. 2: 109-139.
16. Mruglaski G., Rajski J., Tyszer J., Ring Generators-New Devices for Embedded Test Applications, IEEE Trans. on Computer-Aided Design, 2004, vol. 23, No. 9: 1306-1320.
17. Joux, A., Delaunay, P., Galois LFSR, embedded devices and side channel weaknesses, in Progress in Cryptology-INDOCRYPT 2006. Lecture Notes in Computer Science (eds. Barua, R., Lange, T.), New York: Springer, 2006, vol. 4329: 436-451.
18. Hankerson, D., Vanstone, S., Menezes, A., Guide to Elliptic Curve Cryptography, New York: Springer, 2004.