ELSEVIER

Signal Processing 56 (1997) 199–203

SIGNAL PROCESSING

Fast communication

Maximum length cellular automaton sequences and its application

Taejoo Chang, Iickho Song*, Jinsoo Bae, Kwang Soon Kim

Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, South Korea

Received 23 October 1996

Abstract

In this paper, the maximum length linear binary cellular automaton (CA) sequence is considered. It is shown that all possible maximum length linear binary CA sequences, which are equivalent to maximum length linear binary feedback shift register (LFSR) sequences, can be constructed using linear simple CAs. A table of configurations of the n-cell maximum length simple CAs with its characteristic polynomials is obtained for $76 \leqslant n \leqslant 120$. An application of the CAs to stream ciphers is indicated. In other applications, a maximum length LFSR may be replaced by a maximum length linear binary CA. © 1997 Elsevier Science B.V.

Keywords: Cellular automata; Maximum length sequences; Linear feedback shift register

1. Introduction

The cellular automaton (CA) is useful in many application areas including error-correcting codes and cryptography [9] due to its simplicity, modularity, and regularity [6]. Other application examples are well compiled in [9]. Linear feedback shift register (LFSR) sequences also have many useful properties and wide areas of applications [5,7]. The CA can be implemented efficiently on VLSI with fewer gate delays than the LFSR, especially when the number of feedback taps are large, and can be used in a variety of high-bandwidth applications.

In this paper, the maximum length linear binary CA sequence is considered. It will be shown that all possible maximum length linear binary CA sequences can be constructed using linear simple CAs, and that the maximum length linear binary CA sequences are equivalent to the maximum length binary LFSR sequences. A procedure for finding the configurations of the maximum length linear simple CA is described.

2. Preliminaries

2.1. Cellular automata

A CA is a collection of n storage elements. The elements are called the *cells*, which take on discrete values. At each clock (discrete time step) the value of each cell is set to the value of the output of a function. The function, called a *transition function* or a *rule*, takes the present values of a cell and its neighborhood cells as arguments. The arrangement of cells, one of

* Corresponding author. E-mail: isong@sejong.kaist.ac.kr.

PII S0165-1684(97)00017-0

the factors determining the size of the neighborhood, determines the dimension of a CA. Extreme cells, i.e., cells located at the boundary, may have their neighborhood missing. It is called the *null boundary conditions* when the values of the missing neighborhood cells are set to zero and the *cyclic boundary conditions* when the extreme cells are considered to be adjacent.

Let $s_i(t)$ be the state of a cell i at time t in a one-dimensional CA, where the cells form a single line. The state at time $t+1$ can be written as

$$s_i(t+1) = F[s_{i-r}(t), \ldots, s_i(t), \ldots, s_{i+r}(t)], \quad (1)$$

where F is the transition function. (For notational simplicity, we will sometimes drop the time variable in a state: for example, $s_i(t)$ may also be written as s_i.) The collection of cells $(i-r, \ldots, i, \ldots, i+r)$ about cell i will be called the $(2r+1)$-*neighborhood* of i. Assume $s_i(t)$ may take values from the set $I_q = \{0, 1, \ldots, q-1\}$. A CA is called a *binary* CA if $s_i(t)$ takes only 0 or 1, i.e., if $q = 2$. A binary CA is called *linear* if the transition function F has the form

$$F[s_{i-r}, \ldots, s_i, \ldots, s_{i+r}] = \sum_{k=-r}^{r} c_{ik} s_{i+k}, \quad (2)$$

where $c_{ik} \in \{0, 1\}$ and the arithmetic is performed on GF(2).

2.2. Linear feedback shift register sequences

An LFSR is a collection of n binary storage elements $x_0, x_1, \ldots, x_{n-1}$ which are called the *stages*. At each clock, the value of x_{i+1} is transferred to x_i, for $i = 0, 1, \ldots, n-2$, and x_{n-1} is set to the value $f_L(x_0, x_1, \ldots, x_{n-1}) = c_0 x_0 \oplus c_1 x_1 \oplus \cdots \oplus c_{n-1} x_{n-1}$ computed before the transition, where $c_i \in \{0, 1\}$ and $\oplus$ denotes the exclusive-or operation (modulo-2 addition). A general form of an n-stage binary LFSR is shown in Fig. 1. The output sequence $\{a_k\}$ of an LFSR, which can be taken from any of the n stages, satisfies the recurrence relation

$$a_{k+n} = \sum_{i=0}^{n-1} c_i a_{k+i}, \quad k = 0, 1, \ldots, \quad (3)$$

where the arithmetic is performed on GF(2). The polynomial

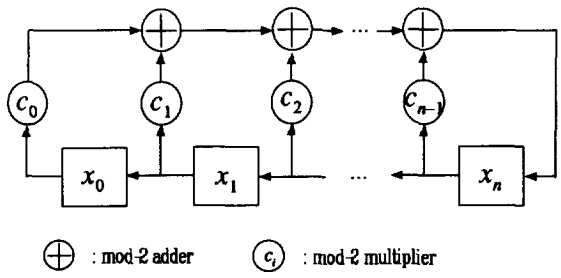

Fig. 1. A general form of LFSR.

$$f(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} + x^n \quad (4)$$

of degree n with the coefficients in GF(2) is referred to as the *characteristic polynomial*[1] of the LFSR defined with f_L.

A maximum length LFSR sequence has the maximum period $2^n - 1$, and this maximum possible period is obtained if and only if the characteristic polynomial is *primitive*, i.e., if $f(x)$ is an irreducible polynomial and divides $x^k - 1$ for $k = 2^n - 1$ and for no smaller k [4]. The number of binary primitive polynomials of degree n is given [5] by $\phi(2^n - 1)/n$, where ϕ is Euler's ϕ-function. The algorithms for searching primitive polynomials and some binary primitive polynomials may be found in [12,13].

3. Cellular automaton sequences

In the rest of this paper, only the linear binary CA with the null boundary condition and 3-neighborhood ($r = 1$) will be considered, unless otherwise specified. A linear binary 4-cell CA with the null boundary condition and the transition function $F(s_{i-1}, s_i, s_{i+1}) = c_1 s_{i-1} + c_2 s_i + c_3 s_{i+1}$ is shown in Fig. 2.

Definition 1. A linear binary CA with the null boundary condition and 3-neighborhood is called a *simple* CA.

Let $(s_1(t), s_2(t), \ldots, s_n(t))$ be the state of an n-cell simple CA at time t. The next state can be de-

[1] In [5], the characteristic polynomial is defined as the reciprocal of $f(x)$, i.e., $x^n f(1/x)$.

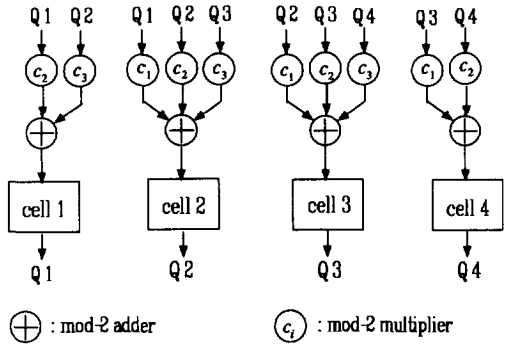

Fig. 2. An example of the 4-cell linear binary CA with 3-neighborhood and null boundary condition.

scribed by the matrix operation [1]:

$$s_{t+1} = Ts_t, \tag{5}$$

where $s_t = [s_1(t), s_2(t), \ldots, s_n(t)]'$ ($'$ denotes the transpose), and T is called the *state-transition matrix*. A state s_0 is called a *cycle state* if there exists an integer p such that

$$s_0 = T^p s_0. \tag{6}$$

The smallest integer p that satisfies (6) is called the *cycle length*, *length*, or *period* of the CA. If all of the states are cycle states, then the state-transition matrix T must be nonsingular. If the period of an n-cell simple CA is $2^n - 1$, then it will be called a *maximum length simple CA* (MLSCA). A simple CA has a maximum length if and only if the characteristic polynomial of the transition matrix is primitive [1]. A sequence $\{s_j(t)\}$, $t = 0, 1, \ldots$, will be called a *CA sequence*.

3.1. A direct search method for the maximum length CA

Let us now consider a finding method of the configuration of the MLSCA.

The next theorem is a necessary condition for a simple CA to have a maximum length.

Theorem 2. *If an n-cell simple CA has a maximum length, then the transition function has the form*[2]

$$F_i(s_{i-1}, s_i, s_{i+1}) = s_{i-1} + d_i s_i + s_{i+1}, \quad i = 1, 2, \ldots, n, \tag{7}$$

where $d_i \in \{0, 1\}$, *and consequently the transition matrix of the CA defined by* (7) *is*

$$T = \begin{bmatrix} d_1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & d_2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & d_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d_{n-1} & 1 \\ 0 & 0 & 0 & \cdots & 1 & d_n \end{bmatrix}. \tag{8}$$

Proof. Suppose that a subdiagonal element, say the $(i, i-1)$th element, is not 1. Then the transition matrix can be partitioned as

$$T = \left[\begin{array}{cccc|ccc} d_1 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ 1 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{i-1} & 1 & \cdots & 0 \\ \hline 0 & 0 & \cdots & \square & d_i & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & d_n \end{array}\right] = \begin{bmatrix} A & B \\ 0 & D \end{bmatrix}. \tag{9}$$

Thus the characteristic polynomial of T is $|xI - A|\,|xI - D|$, which is reducible. □

We shall write the matrix of the form (8) as $T = \text{tridiag}\{d_1, d_2, \ldots, d_n\}$. An n-cell MLSCA can be specified by $\{d_1, d_2, \ldots, d_n\}$, which will be called the *configuration* of the MLSCA. A straightforward observation is that if a configuration $\{d_1, d_2, \ldots, d_n\}$ has maximum length, then the configuration $\{d_n, d_{n-1}, \ldots, d_1\}$ also has maximum length, since $\text{tridiag}\{d_1, d_2, \ldots, d_n\}$ and

[2] A transition function (rule) may sometimes be specified by a rule number [3] in cellular automata. The function (7) corresponds to rule number 90 or 150.

tridiag$\{d_n, d_{n-1}, \ldots, d_1\}$ have the same characteristic polynomial.

Corollary 3. *The number of n-cell MLSCAs is* $N_t = 2\phi(2^n-1)/n$, *and all of the* N_t *n-cell MLSCA can be obtained from a CA with the transition function* (7).

To find an MLSCA, we first select a random configuration $\{d_1, d_2, \ldots, d_n\}$, next calculate the characteristic polynomial of the transition matrix tridiag$\{d_1, d_2, \ldots, d_n\}$, then test if the characteristic polynomial is primitive. Due to a special structure of the transition matrix, the characteristic polynomial can be obtained easily by a recursive formula as follows.

Lemma 4. *Let* A *be a symmetric tridiagonal matrix over* GF(2). *Then the characteristic polynomial* $f_n(x) = |A + xI_n|$ *of* A *can be obtained by the recursive formula*

$$f_n(x) = (d_n + x) f_{n-1}(x) + f_{n-2}(x), \qquad (10)$$

with $f_0(x) = 1$ and $f_1(x) = d_1 + x$.

An algorithm for testing the primitivity of a given polynomial may be found in [12,13]. In the algorithm, the factorization of $2^n - 1$ is also required, which can be obtained from the results in [2,12]. Using the procedure, we found the configurations of the n-cell MLSCAs for $76 \leqslant n \leqslant 120$ as shown in Table 1. To the best of the authors' knowledge, the only similar table may be found in [6] which lists a configuration of n-cell CA for each n, $4 \leqslant n \leqslant 53$, but without the characteristic polynomial.

3.2. An equivalence between the maximum length CA and LFSR sequences

Let us consider in this section an equivalence between the MLSCA and maximum length LFSR sequences. It will be shown that the two sequences have the same recurrence relation.

Lemma 5. *Let* $v(x) = x^n + \sum_{i=0}^{n-1} c_i x^i$ *be the characteristic polynomial of an n-cell simple CA. Then, the state* s_t *of the CA satisfies the recurrence relation*

$$s_{t+n} = \sum_{i=0}^{n-1} c_i s_{t+i}, \quad t = 0, 1, \ldots. \qquad (11)$$

Table 1
Configurations of maximum length simple n-cell CAs with its characteristic polynomials

n	Characteristic polynomial[a]	Configuration[b]
76	179f389f179f0000179f	33
77	3e7d537d3e7d00003e7d	4a
78	6ea19fa16ea100006ea1	85
79	9a0d070d9a0d00009a0d	2b
80	172a103a072a1000172a1	39
81	373030000730300037303	1
82	5fb0b000efb0b0005fb0b	f
83	d1d0d00001d0d000d1d0d	1
84	18691b0203691b0018691b	a8
85	24e939001de9390024e939	2b
86	7303730000037300730373	1
87	f96dfd00046dfd00f96dfd	32
88	1f9a8f90000a8f901f9a8f9	10
89	37300730000007303730073	1
90	73700370000003707370037	1
91	fef50ef000050ef0fef50ef	2c
92	143e393d0003393d143e393d	9c
93	3b6a206b0001206b3b6a206b	df
94	48550d5500000d5548550d55	3f
95	d101000100000001d1010001	1
96	1f98b028b0000028af98b028b	20
97	3ab5b005b00000058ab5b005b	15
98	4b707000700000003b7070007	5
99	faf1f001f00000010af1f001f	1a
100	10b9e900e9000000f9b9e900e9	42
101	21b343004300000062b3430043	66
102	7481510051000000258151005 1	6d
103	e693e300e30000000593e300e3	29
104	1859c8d018d000000089c8d018d	54
105	37300730373000000000730373	1
106	6dc27f106f10000002d27f106f1	1cf
107	e3f013f0e3f000000000013f0e3f	7
108	10af0faf10af000000000faf10af	12
109	2cbd01bd2cbd0000000001bd2cbd	9
110	6fa19ea16fa1000000009ea16fa1	91
111	c1954495c195000000004495c195	bc
112	1b5250024b525000000000024b525	b
113	37303000073030000000000007303	1
114	646f523f346f50000000023f346f5	10a
115	e8d1d00138d1d0000000000138d1d	19
116	19e3310228e331000000000228e331	b0
117	3e0f3f00010f3f0000000000010f3f	e
118	47f7470000f747000000000000f747	a
119	d101d1000001d100000000000001d1	1
120	1b304950026049500000000000260495	83

[a]The table entry is $\sum_{i=0}^{n} a_i 2^i$ written to the base 16, when the characteristic polynomial is $f(x) = \sum_{i=0}^{n} a_i x^i$. For example, when $n = 4$, $f(x) = x^4 + x + 1$ is represented by 13 in base 16.

[b]The table entry is $\sum_{i=0}^{n-1} c_i 2^i$ written to the base 16, when the MLSCA is trigdiag$\{c_{n-1}, \ldots, c_0\}$. For example, when $n = 109$, the number 9 represents tridiag$\{0, \ldots, 0$ (105 times), $1, 0, 0, 1\}$. In the table, only one configuration is shown.

Proof. Let T be the transition matrix of the CA. From Caley–Hamilton theorem, $\phi(T) = T^n + c_{n-1}T^{n-1} + \cdots + c_1T + c_0I = 0$. Multiplying by s_t, we get $T^n s_t = c_{n-1}T^{n-1}s_t + \cdots + c_1Ts_t + c_0Is_t$, or $s_{t+n} = c_{n-1}s_{t+n-1} + \cdots + c_1s_{t+1} + c_0s_t$. □

Comparing (11) with (3), the MLSCA and maximum length LFSR sequences have the same recurrence relation. Thus, an MLSCA sequence is a delayed version of a maximum length LFSR sequence, or vice versa, when they have the same characteristic polynomial. Another view [3] is that a maximum length LFSR may be considered as a special case of a general linear CA. The transition matrix of the LFSR is the companion matrix of the characteristic polynomial. Thus the states of LFSR also satisfy (11).

3.3. An application to stream ciphers

Maximum length LFSR is useful for constructing stream ciphers [10]. In general, several LFSR sequences are combined with a non-linear function to produce a key stream in stream ciphers [11]. The maximum length LFSRs used in this class of stream ciphers may be required to have a large number of feedback taps (equivalently, a large number of terms in the primitive polynomial) to resist a correlation attack [8]. In VLSI implementation, an n-cell simple CA is superior to an n-stage LFSR especially when n and the number of terms of the characteristic polynomial are large, because the transition function requires only the 3-neighborhood in a simple CA regardless of the number of taps. For a software implementation, a similar merit may be stated.

4. Concluding remarks

In this paper, we considered MLSCA sequences. It was shown that all possible MLSCA sequences equivalent to the maximum length LFSR sequences could be constructed using a simple CA. A table of the configurations of the n-cell MLSCA with its characteristic polynomial were obtained for $76 \leqslant n \leqslant 120$. An application to stream ciphers was indicated. A maximum length LFSR might be replaced by an MLSCA in other applications. Only the binary case was considered because in practical applications binary sequences were mainly used.

Acknowledgements

This research was supported by the Ministry of Information and Communications under a Grant from the University Basic Research Fund, for which the authors would like to express their thanks.

References

[1] T.L. Booth, *Sequential Machines and Automata Theory*, Wiley, London, 1967.

[2] J. Brillhart, D.H. Lehmer, J.L. Selfridge, B. Tuckerman and S.S. Wagstaff, "Factorization of $b^n \pm 1$, $b = 2, 3, 5, 6, 10, 11, 12$ up to high powers", *Contemporary Mathematics*, American Mathematical Society, Providence, RI, 1983.

[3] A.K. Das, A. Ganguly, A. Dasgupta, S. Bhawmik and P.P. Chaudhuri, "Efficient characterisation of cellular automata", *IEE Proc., Part E*, Vol. 37, January 1990, pp. 81–87.

[4] W.J. Gilbert, *Modern Algebra with Applications*, Wiley, New York, 1976.

[5] S.W. Golomb, *Shift Register Sequences*, Aegean Park Press, Laguna Hills, CA, 1982.

[6] P.D. Hortensius, R.D. Mcleod, W. Pries, D.M. Miller and H.C. Card, "Cellular automata-based pseudorandom number generators for built-in self-test", *IEEE Trans. Computer-Aided Design*, Vol. 8, August 1989, pp. 842–859.

[7] F.J. MacWilliams and N.J.A. Sloane, "Pseudo-random sequences and arrays", *IEEE Proc.*, Vol. 64, December 1976, pp. 1715–1728.

[8] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers", *J. Cryptology*, Vol. 1, January 1989, pp. 159–176.

[9] S. Nandi, B.K. Kar and P.P. Chaudhuri, "Theory and applications of cellular automata in cryptography", *IEEE Trans. Comput.*, Vol. 43, December 1994, pp. 1346–1357.

[10] R.A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer, Berlin, 1986.

[11] T. Siegerthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications", *IEEE Trans. Inform. Theory*, Vol. IT-30, September 1984, pp. 776–779.

[12] M.K. Simon, J.K. Omura, R.A. Scholtz and B.K. Levitt, *Spread-Spectrum Communications, Vol. 1*, Computer Science Press, New York, 1985.

[13] W. Stahnke, "Primitive binary polynomials", *Math. Comput.*, Vol. 27, October 1973, pp. 977–980.