Analysis of characteristics of phase noise generated by 90/150 cellular automata

An Di†, Cao Xin†, Ji Qingwei†, Shi Wenchuang†, Gao Xinghan††

†School of Information and Technology, Beijing Institute of Technology, Beijing, People's Republic of China
††E-mail: xnandi@126.com

Abstract: Pseudo-random sequence plays an important role in the field of the phase noise simulation. This study analyses the pseudo-random sequence generated by the method based on cellular automata (CA). Compared with the method of look-up table and linear feedback neighbourhood registers, the simulation results show that the pseudo-random sequence generated by the CA method has better adaptability in the process of phase noise simulation.

1 Introduction

Pseudo-random number generators (PRNGs) are required in various areas including Monte-Carlo simulations, on-chip self-test circuitry, and optimisation methods such as simulated annealing and genetic algorithms. In many hardware implementations, it is desirable to optimise performance of the PRNGs in terms of speed, area, and power dissipation, while producing high-quality random numbers due to the use of nearest neighbour interconnectivity and regularity in their physical layout [2]. CA has long been of interest to researchers, both for their fascinating theoretical properties and for their practical applications. In recent years, one-dimensional (1-D) linear hybrid CA has been proposed as an alternative to linear feedback shift registers, in applications such as pseudo-random number generation, cryptography, error correcting codes, and signature analysis [2].

This paper includes two parts: the first part introduces the principle based on one-dimensional 90/150 linear hybrid group CA of generating the pseudo-random number; the second part introduces the results of simulation based on Matlab, which includes the simulation of the phase noise generated by three methods.

2 Theory

The rule number of the basic CA is defined as follows: the neighbour state configuration of the basic CA is:

\[ s_{i-1}, s_i, s_{i+1} \]  

(1)

The mapping of the neighbour state configuration of the basic CA is:

\[ f_i = (111), f_0 = (110), \ldots f_2 = (001), f_3 = (000) \]  

(2)

The combination formula of the mapping is:

\[ C_f = \sum_{i=0}^{2} f_i 2^i \]  

(3)

e.g. (1),

\[ \{f_3f_2f_1f_0f_1f_0 = 10010110\} \]  

(4)

i.e. the rule number of the CA is 150, which is equal to the expression of 150 = 2^4 + 2^3 + 2^1 + 2^0;

\[ f_3f_2f_1f_0f_1f_0 = 01011010 \]  

(5)

\[ a_0 = 0 \]  

(6)

i.e. the rule number of the CA is 90, which is equal to the expression of 90 = 2^4 + 2^3 + 2^1 + 2^0.

The logical expression of rules 90 and 150 can be expressed as:

\[ s_{i+1} = s_{i-1} + a_i s_{i} + s_{i+1} \]  

(6)

Since the near function \( f_i \) of a single cell unit is a linear function, \( f \) in (3) is still a linear function, mapping the states of \( N \) cell units at time \( n \) to \( N \) cell unit states at time \( n+1 \). The linear mapping means that the mapping \( f \) can be expressed as an \( N \times N \)-dimensional matrix, and (3) is a multiplication operation of a finite element field [4]. Based on it, the matrix \( A \) is called a state transfer matrix of a CA, if the above operation is limited to adjacent between the cell units, the 90/150 regular vector \( R \) can be brought into the above formula, and the state transfer matrix \( A \) can be expressed as:

\[ S^{n+1} = f(S^n) = AS^n \]  

(8)
The state of the \( i \)th cell unit at the \( n+1 \) time is the product of the \( i \)th row in the matrix \( A \) and the state vector \( s^i_0 \) at the time \( n \) and can be expressed as:

\[
s^i_{k+1} = \begin{bmatrix} d_1 & 0 & \cdots & 0 & 0 \\ 1 & d_2 & 1 & \cdots & 0 \\ 0 & 1 & d_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 & d_{N-1} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s^i_1 \\ s^i_2 \\ \vdots \\ s^i_{k+1} \end{bmatrix} + s^i_{k+1} + s^i_{k+1}
\]

Each cellular unit in a CA is equivalent to a state machine. The working processes of all state machines are shown in Fig. 1.

In order to simulate the characteristics of the phase noise, the module needs to generate two-way parallel pseudo-random sequence, i.e. generate parity two-channel output in real time.

According to the theory of CA, the output sequence is obtained from the state at time \( n \) generated in parallel, the state of the next moment (time \( n+1 \)) can be obtained from the state at time \( n \) and the state transfer equation of the CA is calculated from the rule vector and the state vector.

In (4), we use the 29th-order cellular automaton, assuming that the rule vector of the CA is \([d_1, d_2, \ldots, d_{29}]\), \([s_1, s_2, \ldots, s_30] \) represents the state vector of the CA at time \( k \), and the \( i \)th cell unit is at the time \( k+1 \). The state is obtained from the state at time \( k \) and the regular vector by the following state transfer equation:

\[
s^i_{k+1} = s^i_{k+1} + d^i_k + s^i_{k+1}
\]

In order to generate two parity sequences in parallel, the state at time \( k+2 \) must be obtained from the state at time \( k \). By analysing the transferring state, if the state at time \( k+1 \) is substituted into the state transition equation at time \( k+2 \), the state at time \( k+2 \) can be obtained from the state at time \( k \). When two sequences were generated in parallel, the state of the next moment (time \( k+2 \)) of the \( i \)th cell unit depends on the logic values and status values of adjacent two cell units \((i-1, i-2) \) and \((i+1, i+2) \) at this moment \((k \) time). The above iteration can be expressed by the process of the following derivation:

\[
s^{i+1}_{k+1} = s^{i+1}_{k+1} + d^i_k + s^{i+1}_{k+1}
\]

By substituting (12), (13), and (14) into (15):

\[
s^{i+2} = s^{i+2} + (d_{i-1} \oplus d_i)s^{i+1} + d^i_k + (d_{i-1} \oplus d_i)s^{i+1} + s^{i+1}
\]

For the zero-boundary additive CA, the neighbouring cell states of the first cell unit and the 29th cell unit need to be supplemented by two zeros:

\[
s^{i+2} = 0 + (d_{i-1} \oplus d_i)s^{i+1} + d^i_k + (d_{i-1} \oplus d_i)s^{i+1} + s^{i+1}
\]

The neighbouring cell states of the second cell unit and the 28th cell unit need to be supplemented with a zero:

\[
s^{i+2} = 0 + (d_{i-1} \oplus d_i)s^{i+1} + d^i_k + (d_{i-1} \oplus d_i)s^{i+1} + s^{i+1}
\]

According to the study of phase noise spectral density study [4], power spectrum distribution of phase noise in the actual oscillator is a piecewise power law, which can be expressed as follows:

\[
S(f) = \frac{1}{f^q}(0 \leq r_i \leq 2, i = 1, 2 \cdots)
\]

\[ r_i \text{ denotes the power exponent corresponding to different ten-octave bands starting from the local oscillator frequency. In general, the more gradual the phase noise spectrum closer to the local oscillator frequency, the larger the } r_i, \text{ and the distance away from the local oscillator frequency can be approximated as the distribution of white noise, that is, } r_i \text{ is closed to zero.}
\]

According to the spectrum representation theorem and the linear system theory [5], the time-domain phase noise simulation method can be summarised as follows: Design a linear system whose amplitude modulus squared result of frequency response is consistent with the required phase noise power-law spectrum model. The signal of white noise is filtered by a linear system, which is shown in Fig. 3. The filtered output signal is a phase noise signal that satisfies the power-law distribution of the power spectrum [6].
In order to analyse the performance of the pseudo-random sequence generator in the simulation of phase noise, the phase noise which conforms to the distribution of $1/f$ in power spectrum is taken as an example. The pseudo-random sequence is generated by look-up table method, the linear feedback shift register method and the cellular automaton method pass through the filter bank. The final white noise sequence is modelled and filtered.

The comparisons of pseudo-random sequences generated by look-up table method and CA method and the linear feedback shift register method are shown in Figs. 4–6.

The comparisons of autocorrelation generated by look-up table method and CA method and the linear feedback shift register method are shown in Figs. 7–9.

The comparisons of power spectrum generated by look-up table method and CA method and the linear feedback shift register method are shown in Figs. 10–12.

The simulation parameters are set as follows: The frequency point of interest is 10 to 10 kHz, the sampling rate is assumed to be 10 MHz, the number of FFT points is selected as $1 \times 10^7$ and a filter fit is selected for each decade, taking the phase noise as the simulation model, which conforms to the distribution of $1/f$ in power spectrum (i.e. $r_i = 0$ in the $S(f) = (1/f)^i(i = 1, 2, \ldots)$).

Fig. 13 shows that the power spectrum passing through the filter of output signal does not approach the ideal curve, and the look-up table method need to store more pseudo-random sequences and more complex addressing logic design to increase the cycle length of the signal, which weakens the randomness of the pseudo-random sequences.

Fig. 14 shows that the power spectrum of the filtered output signal approaches the ideal curve, which shows that it conforms to the distribution of power-law.

Fig. 15 shows that the power spectrum of the filtered output signal approaches the ideal curve, but when the single-path implementation is implemented, the logic design is relatively simple, but its inherent feedback connection structure determines
that it cannot satisfy high-speed applications. If it is necessary to generate multiple white noise sequences in parallel, then the existence of the feedback structure will make the design logic extremely complex and seriously affect the efficiency of the implementation.

4 Conclusion

According to the power spectrum of the white noise generated by filtering the integrated filter system according to the three methods, the following conclusions can be obtained:

5 References

[1] Comer, J.M., Cerda, J.C., Martinez, C.D., et al.: ‘Random number generators using cellular automata implemented on FPGAs’. 44th IEEE Southeastern Symp. on System Theory, Jacksonville, FL, USA, 2012, pp. 67–72
[2] Wolfram, S.: ‘Random sequence generation by cellular automata’, Adv. Appl. Math., 1986, 7, pp. 123–169
[3] Catell, K.: ‘Synthesis of one-dimensional linear hybrid cellular automata’, IEEE Trans. Comput.-Aided Des., 1996, 15, (3), pp. 325–335
[4] Chuanwu, Z.: ‘Cell automata combination pseudorandom sequence generator’, J. Univ. Electron. Sci. Technol., Natural Science Edition, 2008, 37, (5), pp. 716–719
[5] Serra, M.: ‘The analysis of one-dimensional linear cellular automata and their aliasing properties’, IEEE Trans. Comput.-Aided Des., 1990, 9, (7), pp. 767–778
[6] Moon, U.-K.: ‘Spectral analysis of time-domain phase jitter measurements’, IEEE Trans. Circuits Syst.: Analog Digit. Signal Process., 2002, 49, (5), pp. 321–327