HCA101: A Chaotic Map based on Cellular Automata with Binary Synchronization Properties

Radu Dogaru
University “Politehnica” of Bucharest
ETTI, Dept. of Applied Electronics and Info. Eng., Natural Computing Laboratory
Bucharest, Romania
e-mail: radu_d@ieee.org

Abstract—This paper reviews the properties of a novel chaotic map based on hybrid cellular automata (HCA101-map) and compares them against the widely known logistic map with finite precision implementation, from the perspective of their use as PN-sequences in communication systems. It turns out that our HCA-map is superior in any respect to the logistic map and in addition posses the binary synchronization property allowing to dramatically simplify acquisition circuits needed to reconstruct the phase of the PN-sequence in the receiver.

Keywords—chaotic maps; cellular automata; pseudo noise sequences; synchronization

I. INTRODUCTION

Pseudo random sequences (PN-sequences) are widely used in communication systems. They are often generated by discrete time binary automata like the LFSR (linear feedback shift register) and must posses certain properties. Among these we mention: i) a very long period of the generated sequence; ii) good randomness properties; iii) a certain kind of correlation function that allows synchronization (called PN code acquisition in communications parlance) and iv) convenient complexity for hardware implementation. Recently, chaotic maps were considered as an alternative for generating PN-sequences, [1][2][3]. One specific motivation for this choice is the theoretically infinite state space expected to produce infinitely long periods and the existence of the chaos synchronization properties which opens a novel perspective to otherwise complex and power consuming PN-acquisition schemes based on correlating the received sequence with a local reference. Various chaotic maps were proposed in the literature but so far the “logistic map” [4] is the most popular [5][6]. Still, in dealing with “real world” digital implementations of chaotic systems one needs to consider that state variables will be implemented as binary words of finite size \( n \) with dramatic effects over such properties as the length of the PN sequence. Another approach to increase the state space as much as needed (and generate very long sequences) is to build cellular automata capable to produce PN-sequences. To date, certain hybrid cellular automata (HCA) gained interest and were proposed to replace the LFSR in communication systems. Particularly attractive for these systems was the property of generating orthogonal sequences at each site of the cellular automaton. One such widely cited HCA is the HCA90/150 using cells belonging to the rules 90 and 150 respectively [7]. In some recent works [8][9] we investigate the possibility to achieve binary synchronization in CA, i.e. the possibility to reconstruct and track the entire CA state in the receiver while sending only one part of the transmitter state (a single bit corresponding to one cell, in fact the serial bit defining the PN-sequence). The result was a hybrid cellular automata based on rule 101, called here a HCA101 map. Systems with binary synchronization property eliminate the need for a sophisticated PN-acquisition since they allow the reconstruction of the entire transmitted state solely as a nonlinear dynamics effect and without any need to calculate correlation functions. In [10] it is shown that a very low complexity image transmission system with compression and ciphering features is obtained solely as an effect of using the HCA101 map. Binary synchronization is not present in the LFSR or other previously introduced CA (like HCA90/150), nor is present in the logistic map (although there a form of synchronization can be achieved but in the case of sending the whole state resulting in a low immunity to noise).

This paper reviews the major properties of the HCA101 maps in comparison to other chaotic map (e.g. the logistic map) and concludes that in any respect HCA101 is superior providing: i) a much compact hardware implementation; ii) maximal length sequence for a given size \( n \); iii) binary synchronization property; iv) orthogonal PN-sequences from all CA cells [11]; v) no-transient time until reaching the PN-cycle.

Section II reviews the specificity of chaotic maps, while a detailed analysis of their properties follows in Section III. Concluding remarks and discussions are given in Section IV.

II. CHAOTIC MAPS AS PN-SEQUENCES GENERATORS

In what follows we consider finite precision implementations of chaotic maps, i.e the state variable \( x \) is represented using \( n \) bits. The general formula for a nonlinear map in discrete time “\( t \)” is: \( x(t+1) = F(x(t)) \), where \( F \) is a nonlinear function and \( x \) is a state variable, here considered to
be a finite size $n$ binary vector. As shown next, $F$ may be specified analytically via a relatively simple formulae (like in the case of the logistic map) but it can be also the result of applying a transform to the digital word representing the state vector. Such an approach is exemplified for the recently introduced HCA101-map.

A. The logistic map

This map resulted from a refined model of reproduction in species [4] where the state variable $x(t)$ represents the number of individuals in a population and $\lambda$ is a “prolific” coefficient no larger than 4. It is known that for large valued of $\lambda$ (i.e. $\lambda > 3.6$) the dynamics of the logistic map: $F(x) = \lambda x(1-x)$ is chaotic. For a finite word implementation the same as above arithmetic is considered but the operators (such as multiplication, addition/subtraction) are implemented with $n$-bit width operators. An FPGA implementation is reported in [12]. Due to the existence of a multiplier the hardware complexity is $O(n^2)$.

B. The HCA101 map

The HCA map results from the implementation of a chaotic counter. Such chaotic counters were already used to replace normal counters in addressing images with surprising positive effects in performing compression in image transmission [10]. In the following we will briefly compare chaotic counters such as the Linear Feedback Shift Register (LFSR) and the novel Hybrid Cellular Automaton (HCA) [8][9] The schematic diagram of both systems is provided in Fig.1.

\[
\begin{align*}
\text{CELLULAR AUTOMATA} & \quad A = \text{chaotic} \\
\text{LFSR (Linear Feedback Shift Register)} & \quad \text{fixed point,}\end{align*}
\]

\[
\begin{align*}
x_i(t+1) &= m \oplus \text{Cell}(x_{i-1}(t), x_i(t), x_{i+1}(t), ID) \quad (1)
\end{align*}
\]

Figure 1. Two types of chaotic counters and their properties: While both have similar properties of the generated pseudo-random sequence, they have different synchronization properties. The autonomous ($A$) mode is used in the sensor (transmitter) while the synchronous ($S$) mode is used in the receiver.

The discrete-time dynamics of the HCA101 is given by the next equation, which applies synchronously to all cells (a cell is identified by an index $i \in [1,2,..,n]$: \[ x_i(t+1) = m \oplus \text{Cell}(x_{i-1}(t), x_i(t), x_{i+1}(t), ID) \quad (1) \]

where $\oplus$ is the logical XOR operator and $\text{Cell}(u1,u2,u3,ID)$ is a Boolean function with 3 binary inputs $(u1,u2,$ and $u3)$. For the HCA101 map, the Boolean function corresponds to rule ID=$101$ in Wolfram’s notation [13]. In its binary representation, the most significant bit of ID corresponds to the cell output when $(u3,u2,u1) = [1,1,1]$.  

A periodic boundary condition is assumed i.e. the leftmost cell ($i=1$) is connected to the rightmost one ($i=n$). Assuming a binary representation (fixed point, $0 \leq x < 1$) such that $x_n$ is the most significant bit of the state variable $x$ and $x_1$ is the least significant bit of the same state variable, equation (1) may be rewritten in the “standard” map formulation with $x(t+1) = F_{\text{HCA}101}(x(t))$. A similar idea was exploited in [14] to simplify the FPGA implementation of tent maps. Two plots of the HCA101 and logistic map respectively (for $n=13$ and \[ m = [001110110110101] \]) are given in Fig.2. It is interesting to note the complex nonlinear mapping $F_{\text{HCA}101}$ is obtained with a very simple arithmetic (defined by the set of $n$ feedback gates with 3 inputs each, as shown in Fig.1) while a simple mapping (in the case of logistic function) require some hardware inefficient arithmetic.

III. PROPERTIES OF THE HCA MAP VERSUS LOGISTIC MAP

A. Period of the PN-sequence

Any finite automaton (including the logistic map in a finite precision $n$) has a finite number of states. From this perspective complex behaviors are associated with the existence of attractors with very long periods (Fig.3). In addition to long periods an additional measure for the “degree of chaos” is needed and was introduced in [8] to differentiate between traditional counters (also generating maximal length sequences, but with no chaotic behaviors) and chaotic counters to be used as PN-sequence generators.

In the case of the HCA101 map the binary mask vector \[ m = [m_1,m_2,..,m_n] \] (see Fig.1 and eq. (1)) is optimized [9] for any odd size up to $n = 29$ such that $r = L/2^{n-1} \to 1$ (maximal cycle length). For instance, in the case of $n=13$ the length of the man cycle is $L = 8199 = 2^{13} - 1$, i.e. very close to the
maximal length cycle. Moreover, the mask is designed such that the state 0 (all bits in 0) is included in the maximal cycle generating the PN-sequence thus ensuring a simple reset of the PN-sequence generator. In HCA101 maps with odd \( n \) transients (see Fig.3) are reduced to 0 (i.e. in order to enter the cycle generating the PN-sequence, there is no additional transient state – since all states belong to the PN-cycle). A detailed study of the chaotic properties is given in [8] which shows “maximal degree of chaos” for the sequence of states generated by the HCA101 maps. The degree of chaos was found better than for CA with rule 30, previously known as the most representative example of a CA based PN-sequence generator.

**State space, dynamic behaviors, attractors, basins of attraction**

![State vector x (represented here on n bits, as in any computer)](image)

In the case of the logistic map, although the measures in [8] indicate a high degree of chaos for long cycles, simulations for different precisions \( n \) indicate that the lengths of the longest cycles \( L \) are very small compared to the maximal length (i.e. \( L \ll 2^n \)). These results are summarized in two tables in Fig.4.

![Figure 3. State profile and attractors for the dynamics of a finite precision nonlinear map](image)

For instance, the largest cycle for \( n=13 \) is obtained for \( \lambda = 3.9 \) but its length \( L=83 \) is almost 100 times smaller than the maximal cycle length (8192). Instead, the HCA101 map provides the longest cycle with \( L=8191 \) at the highest degree of chaos. Consequently we conclude that although both systems are chaotic, the HCA101 map provides the best (near to maximal) PN-sequence length for a given \( n \). The LFSR also provides near maximal length sequence but unlike HCA101 it is not capable of binary synchronization. The same stands for other known CA PN-sequence generators.

**B. Transients**

As seen from Fig. 4 (upper table), in the case of logistic map the total number of “states in cycles” (i.e. belonging to a loop as represented in Fig.3) is much smaller than the entire number of possible states. It follows that all these states belong to “transients”, with a negative impact over the functioning of the logistic map as a PN-sequence generator. It means that a number of cycles should be wasted until we are sure that the logistic maps enters the useful long cycle providing the PN-sequence. Moreover, the transient length is difficult to anticipate and it depends on the initial state. Instead, the HCA has no transients, and all states are included in one ore more cycles. HCA101 systems are designed such that the state 0 is always included in the longest cycle thus providing a sure functioning as PN-sequence via a simple reset of all cells before running.

**C. FPGA implementation (hardware complexity)**

In [12] FPGA implementations are provided for different chaotic maps. The logistic map although having a simple and elegant formulae requires a multiplier and other arithmetic operators which are rather complex. Instead, cellular automata with their regular structure are well suited for FPGA implementations. In a companion paper [15] we provide an FPGA implementation for the HCA101 map. While [12] reports 923 slices and 1317 LUTs for a logistic map with \( n=32 \) bits resolution, the implementation of a HCA101 map with the same size (\( n=32 \) CA cells) require almost 50 times less resources i.e. 18 slices and 32 LUTs.

**B. Binary synchronization property**

Chaos synchronization was since its discovery in the late 1980’s considered a phenomenon with a lot of potential. The general diagram of a transmission system employing chaos synchronization is provided in Fig. 5.

![Figure 4. Attractor profile for the logistic map. Note the very short “largest” cycle (\( L \ll 2^n \)). Also most of the states are not “useful” (i.e. included in cycles generating the PN-sequence) but are rather transient states.](image)

For instance, the largest cycle for \( n=13 \) is obtained for \( \lambda = 3.9 \) but its length \( L=83 \) is almost 100 times smaller than the maximal cycle length (8192). Instead, the HCA101 map provides the longest cycle with \( L=8191 \) at the highest degree of chaos. Consequently we conclude that although both systems are chaotic, the HCA101 map provides the best (near to maximal) PN-sequence length for a given \( n \). The LFSR also provides near maximal length sequence but unlike HCA101 it is not capable of binary synchronization. The same stands for other known CA PN-sequence generators.

**B. Transients**

As seen from Fig. 4 (upper table), in the case of logistic map the total number of “states in cycles” (i.e. belonging to a loop as represented in Fig.3) is much smaller than the entire number of possible states. It follows that all these states belong to “transients”, with a negative impact over the functioning of the logistic map as a PN-sequence generator. It means that a number of cycles should be wasted until we are sure that the logistic maps enters the useful long cycle providing the PN-sequence. Moreover, the transient length is difficult to anticipate and it depends on the initial state. Instead, the HCA has no transients, and all states are included in one ore more cycles. HCA101 systems are designed such that the state 0 is always included in the longest cycle thus providing a sure functioning as PN-sequence via a simple reset of all cells before running.

**C. FPGA implementation (hardware complexity)**

In [12] FPGA implementations are provided for different chaotic maps. The logistic map although having a simple and elegant formulae requires a multiplier and other arithmetic operators which are rather complex. Instead, cellular automata with their regular structure are well suited for FPGA implementations. In a companion paper [15] we provide an FPGA implementation for the HCA101 map. While [12] reports 923 slices and 1317 LUTs for a logistic map with \( n=32 \) bits resolution, the implementation of a HCA101 map with the same size (\( n=32 \) CA cells) require almost 50 times less resources i.e. 18 slices and 32 LUTs.

**B. Binary synchronization property**

Chaos synchronization was since its discovery in the late 1980’s considered a phenomenon with a lot of potential. The general diagram of a transmission system employing chaos synchronization is provided in Fig. 5.
So far, although chaos synchronization is widely present, very few systems have been reported so far as being capable of *binary synchronization*. As already described in [8][9] the nonlinear system described by the HCA101 map has this very useful property. It means that \( m=1 \) in Fig.5. i.e. only 1 bit per clock cycle suffices to ensure that after a finite period of time (\( T_{sync} \) or PN-acquisition time) the receiving system (with an internal register of \( n \) bits) recovers entirely the state of the transmitter. Extensive simulations with many random initial conditions provided an estimate for the synchronization time in case of HCA101, as shown in Fig.6.

![Synchronization time (clock cycles) dependence on the representation precision \( n \).](image)

**Figure 6.** Synchronization time (clock cycles) dependence on the representation precision \( n \).

### IV. CONCLUDING REMARKS

The properties of a novel pseudo-noise generator based on a nonlinear map based on hybrid cellular automata (called herein HCA101 map) were analyzed in comparison with similar properties of other chaotic maps such that the widely known logistic map, for a finite precision representation \( n \). A comparison between HCA101 and some very popular PN-sequence generators is given in Table I, including issues of hardware implementation.

**TABLE I. COMPARATIVE PERFORMANCES OF DIGITAL CHAOTIC MAPS**

<table>
<thead>
<tr>
<th>Chaotic map Feature</th>
<th>HCA101</th>
<th>Logistic map</th>
<th>HCA Rules 90/150</th>
<th>LFSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle length near maximal value ( 2^n )</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>No transients</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Binary Synchronization capability</td>
<td>YES, detailed in [8][9]</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>FPGA implementation complexity (LUTs) for ( n=32 ) bits</td>
<td>32 based on ([16]_0(n))</td>
<td>1317 based on ([12]_0(n))</td>
<td>896 based on ([16]_0(n))</td>
<td>-</td>
</tr>
</tbody>
</table>

It is clear that in any respect the HCA101 map has superior properties, providing a hardware efficient pseudo-noise generator with the important additional property of binary synchronization. It can Effectively replace traditional chaotic maps while having a more compact realization but in addition may provide novel applications with an improved ratio (functional capability)/(implementation complexity). Two such successful examples are detailed in [10][11] They can be easily extended to other less conventional applications such as sonar exploration, and in building mobile and low power sensing-nodes of sensors networks or for other low power communication applications. The in-depth investigation of the cryptographic properties for this chaotic map is a subject of further research.

**REFERENCES**


