An Interactive Self-Replicator Implemented in Hardware

André Stauffer
Logic Systems Laboratory
Swiss Federal Institute of Technology in Lausanne
1015 Lausanne, Switzerland
andre.stauffer@epfl.ch

Moshe Sipper
Department of Computer Science
Ben-Gurion University
Beer-Sheva 84105
Israel
and
Logic Systems Laboratory
Swiss Federal Institute of Technology in Lausanne
sipper@cs.bgu.ac.il

Abstract Self-replicating loops presented to date are essentially worlds unto themselves, inaccessible to the observer once the replication process is launched. In this article we present the design of an interactive self-replicating loop of arbitrary size, wherein the user can physically control the loop's replication and induce its destruction. After introducing the BioWall, a reconfigurable electronic wall for bio-inspired applications, we describe the design of our novel loop and delineate its hardware implementation in the wall.

1 Introduction: Cellular Automata and Self-Replication

The study of self-replicating machines, initiated by von Neumann over 50 years ago, has produced a plethora of results over the years [7, 8]. Much of this work is motivated by the desire to understand the fundamental information-processing principles and algorithms involved in self-replication, independent of their physical realization [5, 11]. The fabrication of artificial self-replicating machines can have diverse applications, ranging from nanotechnology [2], through space exploration [3], to reconfigurable computing tissues—the latter of which shall be introduced in Section 2.

A major milestone in the history of artificial self-replication is Langton's design of the first self-replicating loop [4]. His 86-cell loop is embedded in a two-dimensional, eight-state, five-neighbor cellular space; one of the eight states is used for so-called core cells and another state is used to implement a sheath surrounding the replicating structure. Byl [1] proposed a simplified version of Langton's loop, followed by Reggia et al. [5] who designed yet simpler loops, the smallest being sheathless and comprising five cells. More recently, Sayama [6] designed a structurally dissolvable loop, based on Langton's work, which can dissolve its own structure, as well as replicate.

All self-replicating loops presented to date are essentially worlds unto themselves: once the initial loop configuration is embedded within the cellular automaton (CA) universe (at time step 0), no further user interaction occurs, and the CA chugs along in total oblivion of the observing user.

In our previous work we described the design of a simple 2 × 2 self-replicating loop that can be physically activated by the user [10]. In this article we present an interactive self-replicating loop of arbitrary size n, $n \geq 3$, thus generalizing our preliminary result to loops of any size. The user can physically control the loop's replication and induce its destruction, two mechanisms that are explained in Section 3. Section 4 discusses the hardware implementation of the loop in the interactive reconfigurable computing tissue for bio-inspired applications, the *BioWall*. Finally, we present concluding remarks in Section 5.

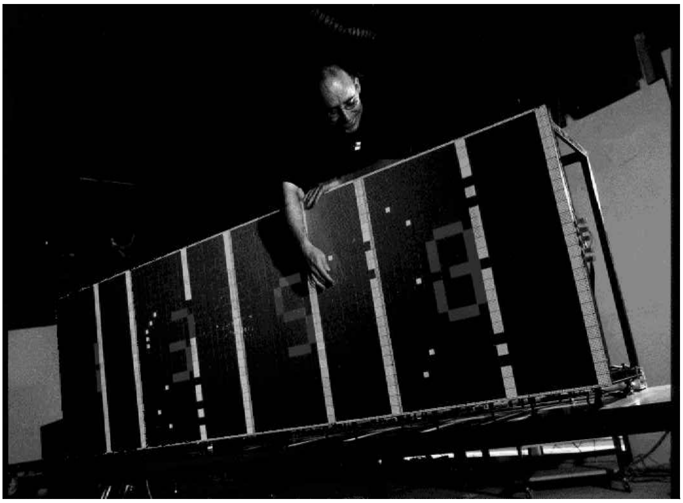

Figure 1. The BioWall, an interactive reconfigurable computing tissue of 80 × 25 = 2,000 cells (Photograph by A. Herzog).

2 An Interactive Reconfigurable Computing Tissue

The BioWall[1] (bio-inspired electronic wall) is an ongoing project in our laboratory. This wall, first designed to implement an electronic watch with self-repair and self-healing capabilities [9], constitutes a reconfigurable computing tissue capable of interacting with its environment by means of a large number of touch-sensitive elements coupled with two-color light-emitting diode (LED) displays. Figure 1 shows the 80 × 25 = 2,000 cells of its hardware implementation. Each cell is made up of a transparent touch-sensitive element, a two-color 8 × 8 dot-matrix LED display, and a reconfigurable Xilinx Spartan XCS10XL FPGA circuit (Figure 2). Within the cell, the transparent touch-sensitive element and the LED display are physically joined by an adhesive film. As each of the cells provides the same connections to its four neighbors, the BioWall is homogeneous and fully scalable.

In the electronic watch application (Figure 1), the touch-sensitive element of the BioWall cell acts as a push button used to render the cell faulty or healthy again. The

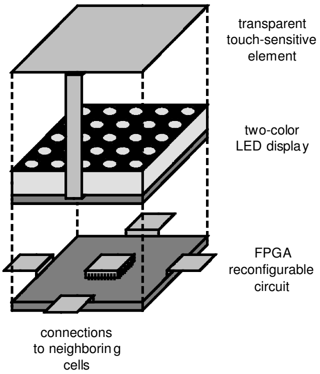
transparent touch-sensitive element
two-color LED display
FPGA reconfigurable circuit
connections to neighboring cells

Figure 2. The basic cell of the interactive reconfigurable computing tissue.

[1] European patent No. 01201221.7.

0:	empty component
1:	building component
2:	east-moving growth signal
3:	north-moving growth signal
4:	west-moving growth signal
5:	south-moving growth signal
6:	left-turn signal
7:	first east-branching signal
8:	second east-branching signal
9:	first north-branching signal
10:	second north-branching signal
11:	first west-branching signal
12:	second west-branching signal
13:	first south-branching signal
14:	second south-branching signal
15:	cut-off signal

Figure 3. The seven basic cellular states 0 to 6 used for the idle loop and the nine additional cellular states 7 to 15 involved in the self-replication and self-destruction processes.

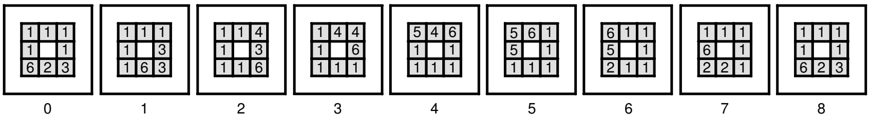

Figure 4. The eight-time-step idle cycle of the inactive loop.

LED display shows the boundaries, the spare cell columns, and the current time of each of its four digits, as well as the faulty or healthy state of the cell.

3 Loop Design and Operation

Contrary to previous loops, which self-replicate continually, the novel one presented in Figure 3 is idle *unless physically activated.* This $n \times n$ loop, with $n \geq 3$, is therefore an interactive self-replicator.

We consider a two-dimensional, five-neighbor cellular space, with seven basic states 0 to 6 per cell (Figure 3; the CA's state-transition rules for loops of any size $n \geq 3$ are given in the Appendix). Our minimal 3×3 loop, based on one of Reggia's loops called UL10W8V [5], is made up of eight cells. As long as no physical input is provided, the loop is inert, continually undergoing an eight-time-step cycle (Figure 4).

The user activates the idle loop by providing a physical input on one of its eight cells. This activation occurs by pressing the touch-sensitive element of the BioWall cell and leads to the appearance of an activated left-turn signal (in Figure 5 the physically activated cells adopt shadowed representations). The activated loop is now ready to self-replicate (if space is available) or self-destruct (if surrounding loops prevent replication).

When surrounded by empty space, the activated loop self-replicates; for example, if the bottom-left cell or the bottom-middle cell is activated, the loop replicates eastward within 48 time steps (Figure 6). Depending on the initially activated cell, the replication

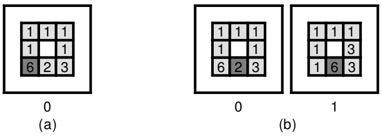

Figure 5. Activating the idle loop by touching one of its two bottom-left and bottom-middle cells. Depending upon the cell touched, the appearance of an activated state 6 (activated left-turn signal) takes zero (a) or one (b) time step. In this representation, the externally activated cell is shadowed.

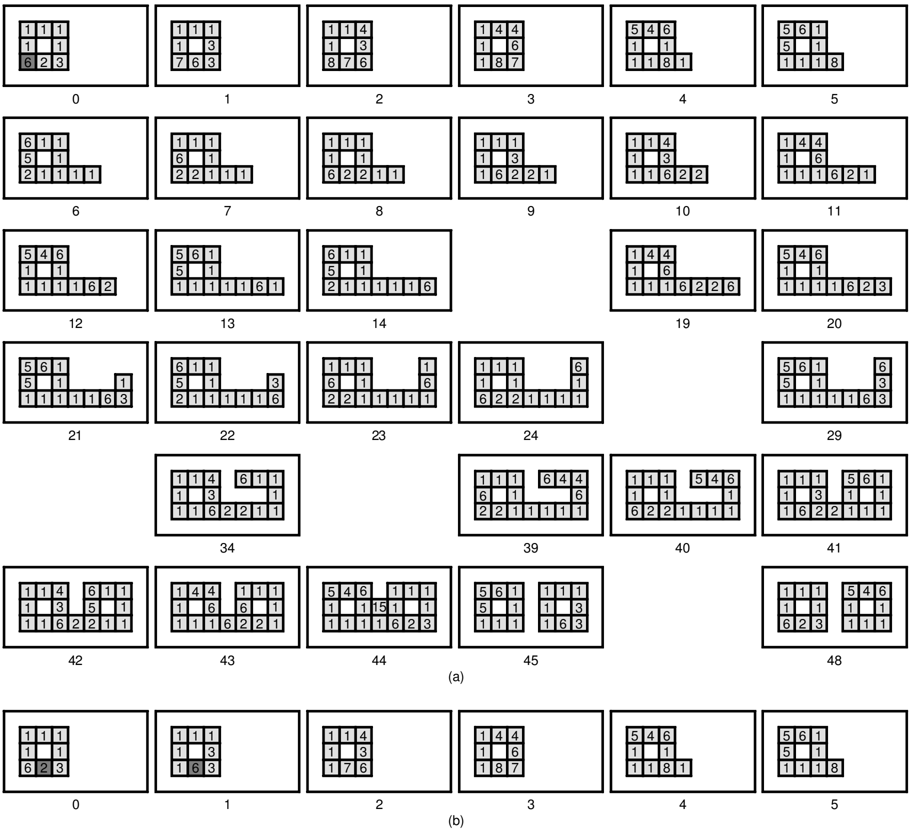

Figure 6. When the bottom-left cell (a) or the bottom-middle cell (b) is activated (touched), the loop self-replicates eastward within 48 time steps. (Note: (b) is identical to (a) from the third step onward.)

process can occur in all of the four cardinal directions and requires the nine additional cellular states 7 to 15 to be performed (Figure 3).

When the loop's replication is blocked by another loop, the former loop initiates a 16-time-step destruction process of the latter loop (Figure 7); this process can occur in one of four directions, depending on the initially activated cell. Whereas the mother loop can always destroy the daughter loop, the contrary depends on the loop's size, as shown below; for example, when a 4 × 4 daughter loop tries to replicate into the mother's space, both loops become idle within 12 time steps (Figure 8).

Considering the general case of an $n \times n$ loop, the respective time step values i of its idle cycle, r of its self-replication process, and d of its self-destruction process are given by the following expressions:

$$i = 4(n-1) \tag{1}$$

$$r = 6i \tag{2}$$

$$d = 2i \tag{3}$$

Table 1 shows the respective values of i (idle time steps), r (replication time steps), and d (destruction time steps) for n between 3 and 5.

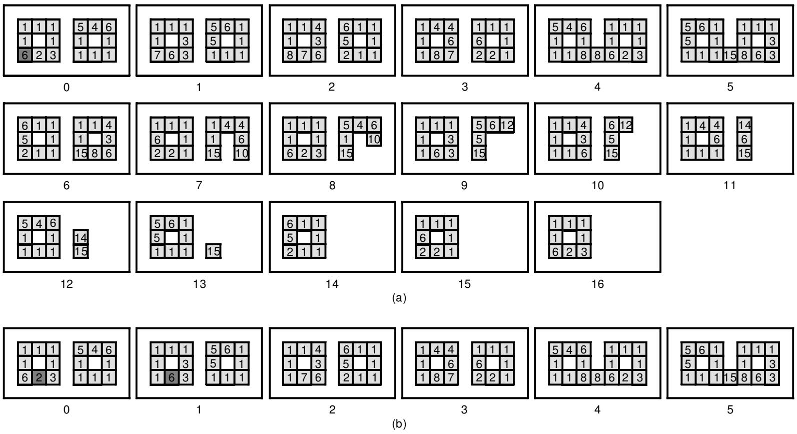

Figure 7. Activating the bottom-left cell (a) or the bottom-central cell (b) results in a destruction process, since the loop is blocked by another loop and thus cannot self-replicate eastward. (Note: (b) is identical to (a) from the third step onward.)

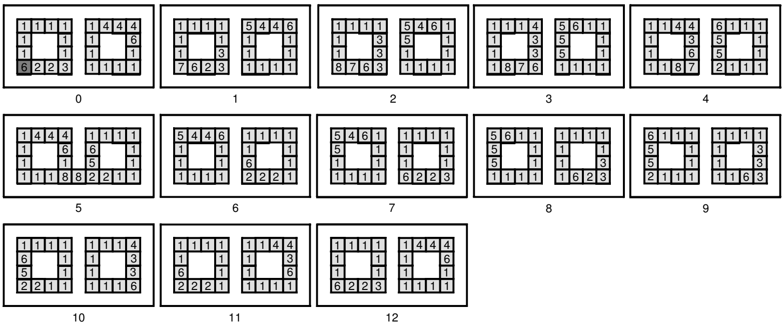

Figure 8. Daughter deactivation: When the activated 4 × 4 daughter loop (left) tries to replicate into the mother loop's (right) space both loops become idle.

According to Equation 2, six idle cycles of the mother loop are always needed to perform the self-replication process. For a mother loop of any size, this value comprises one cycle for the mother-loop arm extension, one cycle for the building of each of the four sides of the daughter loop, and one cycle for the daughter-loop detachment.

Inside an $n \times n$ loop, the n-character string comprising $n - 1$ growth signals (2, 3, 4, or 5) followed by a single left-turn signal (6) constitutes its genome. For such a loop, we define a *generation* as an entire replication cycle, that is, from the time step at which replication begins up until the time step at which a daughter loop is produced and the genome of the mother loop recovers its starting position. Keeping the genome of the original mother loop in its starting position, the genomes of the successive daughter loops are shifted counterclockwise by a constant number s of cells

Table 1. Time steps of an $n \times n$ loop: i (idle cycle), r (replication process), and d (destruction process) for n between 3 and 5.

n	i	r	d
3	8	48	16
4	12	72	24
5	16	96	32

Table 2. Generation-related data of an $n \times n$ loop: s (shifts per generation), p (phase), and g (generations per period) for n between 3 and 5.

n	s	p	g
3	4	1/2	2
4	7	7/12	12
5	10	5/8	8

after each generation:

$$s = 1 + 3(n - 2) \tag{4}$$

The phase p of the loop can then be defined as the ratio between the shifts per generation s and the time steps of the idle cycle i:

$$p = s/i \tag{5}$$

The generations per period g is the number of successive generations needed to recover the genome in the starting position. This number is defined such that $g \times p$ corresponds to the smallest integer. Table 2 shows the values of s (shifts per generation), p (phase), and g (generations per period) for n between 3 and 5.

When the generations per period g equals 2 (or the phase p equals 1/2), the daughter loop can always destroy the mother loop. This happens because the relative positions of the mother loop and the daughter loop are perfectly symmetrical after the self-replication process. Using Equations 1 and 4 to solve Equation 5 for $p = 1/2$ results in $n = 3$. The 3×3 daughter loop is thus the only one that can destroy its mother loop. For the other loops, where g is greater than two, although the mother loop can destroy the daughter loop, the contrary is impossible: when the daughter tries to replicate into the mother's space, both loops become idle within i time steps. This deactivation process also occurs in i time steps between the original mother loop and most of its successive-generation daughter loops. With the generations per period being g for a given $n \times n$ loop, only the $(g - 1)$-generation daughter loop can destroy the original mother loop.

4 Hardware Implementation

We have implemented the five-neighbor CA of the self-replicating and self-destroying loops in our interactive reconfigurable computing tissue as an application of the BioWall (Figure 9). In this hardware implementation, each CA cell corresponds to a cell in the wall. The touch-sensitive element covering the cell's outer surface acts like a digital switch, enabling the user to click on the cell and thereby activate self-replication or self-destruction (Figure 10).

The field-programmable gate array (FPGA) forming the cell's internal digital circuit is configured so as to implement (a) the data processing of the external (touch) input, (b) the execution of the CA state-transition rules necessary for loop replication or destruction, and (c) the control of the output display. This latter is a two-color LED display that allows the user to view the cell's current state (among the 16 possible ones) and whether the cell is in activated or deactivated mode.

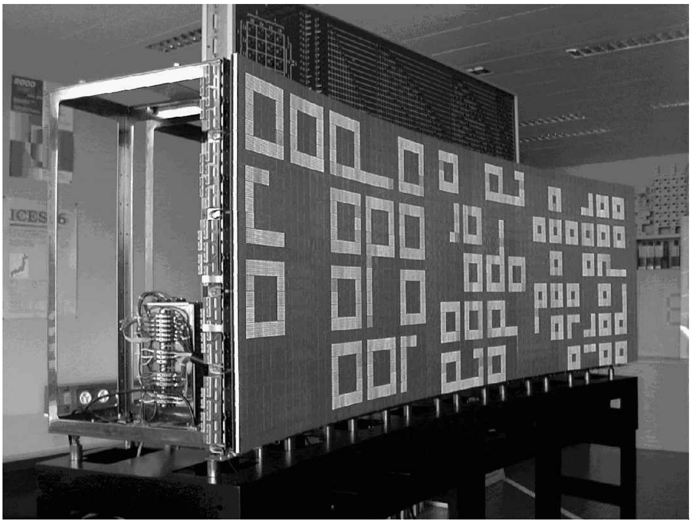

Figure 9. The BioWall used to physically implement our loops (Photograph by A. Badertscher).

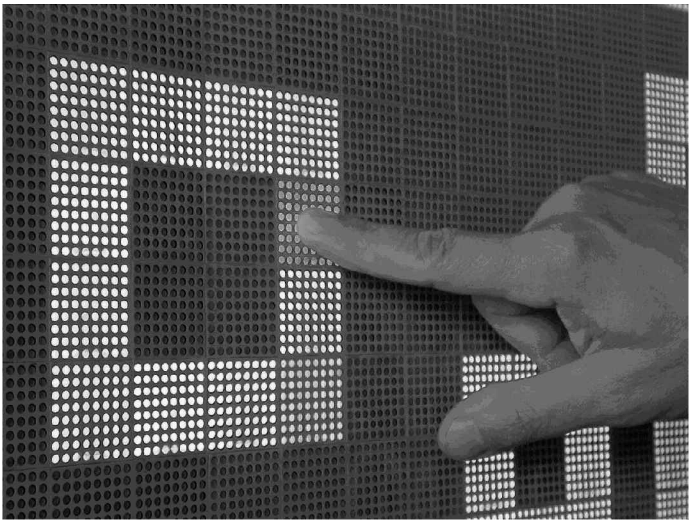

Figure 10. Touching a cell to physically activate the loop (Photograph by A. Badertscher).

5 Concluding Remarks

We presented an interactive self-replicating and self-destructing loop of arbitrary size, wherein the user can control the loop's replication and destruction (though we have shown examples of $n = 3, 4, 5$, the design supports loops of any size n, $n \geq 3$—see Appendix). We then described the loop's hardware implementation within our interactive reconfigurable tissue for a bio-inspired application, the BioWall.

The ability to interact physically with a CA universe—a little-studied issue—is of fundamental import where cellular devices are concerned: One must be able to enter input and to view the output if any practical application is envisaged [7]. Our work herein is but a first step in the domain of interactive cellular replicators implemented in hardware, an issue that we believe will play an important role in the future of such devices.

```
C,N,E,S,W  ->C+  C,N,E,S,W  ->C+  C,N,E,S,W  ->C+  C,N,E,S,W  ->C+

A=-
0,X,X,X,2  -> 1  0,X,X,3,X  -> 1  0,X,4,X,X  -> 1  0,5,X,X,X  -> 1
0,X,0,X,7  -> 1  0,0,X,9,X  -> 1  0,X,11,X,0 -> 1  0,13,X,0,X -> 1
0,X,0,X,8  -> 1  0,0,X,10,X -> 1  0,X,12,X,0 -> 1  0,14,X,0,X -> 1
0,X,2,X,7  -> 8  0,3,X,9,X  ->10  0,X,11,X,4 ->12  0,13,X,5,X ->14
0,X,6,6,6  ->15  0,6,6,6,X  ->15  0,6,6,X,6  ->15  0,6,X,6,6  ->15
1,X,0,1,15 -> 0  1,0,1,15,X -> 0  1,1,15,X,0 -> 0  1,15,X,0,1 -> 0
1,0,X,X,2  -> 2  1,X,X,3,0  -> 3  1,X,4,0,X  -> 4  1,5,0,X,X  -> 5
1,1,1,X,2  -> 2  1,1,X,3,1  -> 3  1,X,4,1,1  -> 4  1,5,1,1,X  -> 5
1,1,0,X,2  -> 3  1,0,X,3,1  -> 4  1,X,4,1,0  -> 5  1,5,1,0,X  -> 2
1,1,X,2,X  -> 3  1,X,3,X,1  -> 4  1,4,X,1,X  -> 5  1,X,1,X,5  -> 2
1,X,0,0,6  -> 6  1,0,0,6,X  -> 6  1,0,6,X,0  -> 6  1,6,X,0,0  -> 6
1,X,15,X,6 -> 6  1,15,X,6,X -> 6  1,X,6,X,15 -> 6  1,6,X,15,X -> 6
1,X,X,X,8  -> 8  1,X,X,10,X ->10  1,X,12,X,X ->12  1,14,X,X,X ->14
2,X,X,X,6  -> 6  3,X,X,6,X  -> 6  4,X,6,X,X  -> 6  5,6,X,X,X  -> 6
2,6,X,X,X  -> 6  3,X,X,X,6  -> 6  4,X,X,6,X  -> 6  5,X,6,X,X  -> 6
6,0,1,0,1  -> 1  6,1,0,1,0  -> 1
6,1,2,0,1  -> 1  6,3,0,1,1  -> 1  6,0,1,1,4  -> 1  6,1,1,5,0  -> 1
6,3,2,0,1  -> 1  6,3,0,1,4  -> 1  6,0,1,5,4  -> 1  6,1,2,5,0  -> 1
6,0,2,0,13 -> 1  6,3,0,7,0  -> 1  6,0,9,0,4  -> 1  6,11,0,5,0 -> 1
6,0,3,0,13 -> 1  6,4,0,7,0  -> 1  6,0,9,0,5  -> 1  6,11,0,2,0 -> 1
6,X,X,X,2  -> 3  6,X,X,3,X  -> 4  6,X,4,X,X  -> 5  6,5,X,X,X  -> 2
6,X,X,X,7  -> 7  6,X,X,9,X  -> 9  6,X,11,X,X ->11  6,13,X,X,X ->13
6,0,X,X,8  -> 8  6,X,X,10,0 ->10  6,X,12,0,X ->12  6,14,0,X,X ->14
6,X,2,X,8  -> 8  6,3,X,10,X ->10  6,X,12,X,4 ->12  6,14,X,5,X ->14
6,14,X,0,X -> 8  6,X,0,X,8  ->10  6,0,X,10,X ->12  6,X,12,X,0 ->14
7,X,X,X,X  -> 8  9,X,X,X,X  ->10  11,X,X,X,X ->12  13,X,X,X,X ->14
8,X,2,X,X  -> 0  10,3,X,X,X -> 0  12,X,X,X,4 -> 0  14,X,X,5,X -> 0
8,0,X,X,0  -> 0  10,X,X,0,0 -> 0  12,X,0,0,X -> 0  14,0,0,X,X -> 0
8,0,X,X,15 -> 0  10,X,X,15,0-> 0  12,X,15,0,X-> 0  14,15,0,X,X-> 0
8,X,X,X,1  -> 1  10,X,X,1,X -> 1  12,X,1,X,X -> 1  14,1,X,X,X -> 1
8,X,7,X,X  -> 1  10,9,X,X,X -> 1  12,X,X,X,11-> 1  14,X,X,13,X-> 1
8,X,6,X,8  ->15  10,6,X,10,X->15  12,X,12,X,6->15  14,14,X,6,X->15
8,1,X,X,15 ->15  10,X,X,15,1->15  12,X,15,1,X->15  14,15,1,X,X->15
15,0,0,0,0 -> 0
15,X,X,1,1 -> 0  15,1,1,X,X -> 0
15,X,8,X,1 -> 0  15,10,X,1,X-> 0  15,X,1,X,12-> 0  15,1,X,14,X-> 0

A=0
6,1,2,0,0  -> 1  6,3,0,0,1  -> 1  6,0,0,1,4  -> 1  6,0,1,5,0  -> 1
6,0,2,0,1  -> 1  6,3,0,1,0  -> 1  6,0,1,0,4  -> 1  6,1,0,5,0  -> 1
6,0,3,0,1  -> 1  6,4,0,1,0  -> 1  6,0,1,0,5  -> 1  6,1,0,2,0  -> 1

A=1
6,1,2,0,0  -> 7  6,3,0,0,1  -> 9  6,0,0,1,4  ->11  6,0,1,5,0  ->13
6,0,2,0,1  -> 7  6,3,0,1,0  -> 9  6,0,1,0,4  ->11  6,1,0,5,0  ->13
6,0,3,0,1  -> 7  6,4,0,1,0  -> 9  6,0,1,0,5  ->11  6,1,0,2,0  ->13
```

Figure 11. CA state-transition rules.

Acknowledgments

This work was supported in part by the Swiss National Science Foundation under grant 21-54113.98, by the Leenaards Foundation, Lausanne, Switzerland, and by the Villa Reuge, Ste-Croix, Switzerland.

References

1. Byl, J. (1989). Self-reproduction in small cellular automata. *Physica D, 34*, 295–299.

2. Drexler, K. E. (1992). *Nanosystems: Molecular machinery, manufacturing and computation*. New York: Wiley.

3. Freitas, R. A., & Gilbreath, W. P. (Eds.). (1980). Advanced automation for space missions: Proceedings of the 1980 NASA/ASEE summer study, Chapter 5: Replicating systems concepts: Self-replicating lunar factory and demonstrations (pp. 189–269). Washington DC: NASA, Scientific and Technical Information Branch (available from U.S. G.P.O., Publication 2255).

4. Langton, C. (1984). Self-reproduction in cellular automata. *Physica D, 10*, 135–144.

5. Reggia, J. A., Armentrout, S. L., Chou, H.-H., & Peng, Y. (1993). Simple systems that exhibit self-directed replication. *Science, 259*, 1282–1287.

6. Sayama, H. (1998). Introduction of structural dissolution into Langton's self-reproducing loop. In C. Adami, R. K. Belew, H. Kitano, & C. E. Taylor (Eds.), *Artificial Life VI:*

Proceedings of the Sixth International Conference on Artificial Life (pp. 114–122). Cambridge, MA: MIT Press.

7. Sipper, M. (1998). Fifty years of research on self-replication: An overview. *Artificial Life, 4*, 237–257.

8. Sipper, M., & Reggia, J. A. (2001). Go forth and replicate. *Scientific American, 285*, 2:26–35.

9. Stauffer, A., Mange, D., Tempesti, G., & Teuscher, C. (2001). BioWatch: A giant electronic bio-inspired watch. In D. Keymeulen, A. Stoica, J. Lohn, & R. S. Zebulum (Eds.), *Proceedings of the Third NASA/DOD Workshop on Evolvable Hardware (EH-2001)* (pp. 185–192). Pasadena, CA: IEEE Computer Society.

10. Stauffer, A., & Sipper, M. (2001). Externally controllable and destructible self-replicating loops. In J. Kelemen & P. Sosik (Eds.), *Advances in Artificial Life: Proceedings of the 6th European Conference on Artificial Life (ECAL 2001)* (pp. 282–291). (Lecture Notes in Artificial Intelligence, 2159). Heidelberg: Springer.

11. Von Neumann, J. (1966). *Theory of self-reproducing automata*, A. W. Burks, Ed. Urbana: University of Illinois Press.

Appendix: Specification of the CA State-Transition Rules

The state-transition rules of the CA implementing the physically controllable self-replicating and self-destructing loops of any size $n \geq 3$ are given in Figure 11. In this figure, where only the rules implying a state change are shown, C, N, E, S, and W correspond to the current states of the cell and of its neighbors to the north, east, south, and west, respectively; $C+$ is the state of the cell at the next time step. The symbol $-$ represents a don't-care condition for a binary variable and X represents a wild card for a decimal state. The binary variable A denotes whether the cell is activated ($A = 1$) or not ($A = 0$). Depending on the value of this variable, there are thus three sets of state-transition rules in Figure 11: (a) 137 rules that are performed independently of the activation of the cell ($A = -$), (b) 12 rules performed when the cell is inactive ($A = 0$), and (c) 12 rules performed when the cell is active ($A = 1$).