

# Emergence of Self-Replicating Loops in an Interactive, Hardware-Implemented Game-of-Life Environment

André Stauffer and Moshe Sipper\*

## Abstract

We present the design of an interactive self-replicating loop, wherein the user can physically induce the loop's creation and then control its replication and 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 fifty years ago, has produced a plethora of results over the years [1, 2]. 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 [3, 4]. The fabrication of artificial self-replicating machines could have diverse applications, ranging from nanotechnology [5], through space exploration [6], 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 [7]. His 86-cell loop is embedded in a two-dimensional, 8-state, 5-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 [8] proposed a simplified version of Langton's loop, followed by Reggia *et al.* [4] who designed yet simpler loops, the smallest being sheath-less and comprising five cells. More recently, Sayama [9] 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.

---

\*A. Stauffer is with the Logic Systems Laboratory, Swiss Federal Institute of Technology in Lausanne, CH-1015 Lausanne, Switzerland. M. Sipper is with the Department of Computer Science, Ben-Gurion University, Beer-Sheva 84105, Israel, and with the Logic Systems Laboratory, Swiss Federal Institute of Technology in Lausanne. E-mails: {andre.stauffer@epfl.ch, sipper@cs.bgu.ac.il}. This work was supported in part by the Swiss National Science Foundation under grant 20-63711.00, by the Leenaards Foundation, Lausanne, Switzerland, and by the Villa Reuge, Ste-Croix, Switzerland.

In our previous work we described the design of a simple  $2 \times 2$  self-replicating loop that can be physically activated by the user [10]. In this paper we present the emergence of this interactive self-replicating loop in a Game-of-Life (or, simply, Life) environment [11]. Contrary to the spontaneous emergence demonstrated by Chou and Reggia [12], the user can physically induce the creation of the loop and then control the loop’s replication and destruction, two mechanisms which are explained in Section 3. Section 4 presents the detection of the loop in the Life context and the switching to the loop context. Section 5 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 6.

## 2 An Interactive Reconfigurable Computing Tissue

The *BioWall*<sup>1</sup> (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 [13], 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  $80 \times 25 = 2000$  cells of its hardware implementation. Each cell is made up of a transparent touch-sensitive element, a two-color  $8 \times 8$  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.

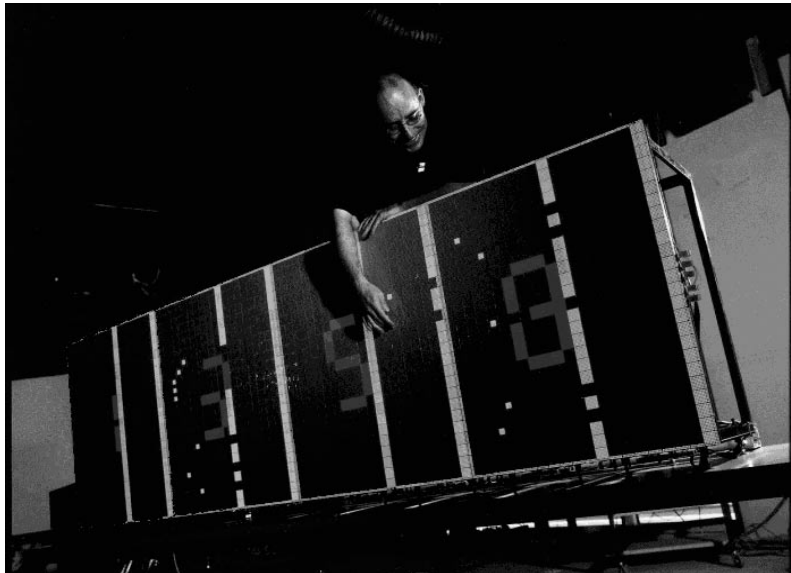


Figure 1: The BioWall, an interactive reconfigurable computing tissue of  $80 \times 25 = 2000$  cells (Photograph by A. Herzog).

---

<sup>1</sup>European patent No 01201221.7.

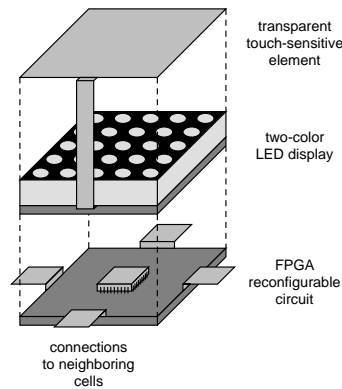


Figure 2: The basic cell of the interactive reconfigurable computing tissue.

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

### 3 Design of a Life/Loop Automaton Cell

Defined in a two-dimensional cellular space, our automaton cell is made up of a Life state machine, a loop state machine and a pseudo-random number generator.

The Life machine requires a nine-neighbor environment with two states,  $Q = 0$  (dead) and  $Q = 1$  (living) per cell. The state  $Q_+$  of the machine at the next time-step depends on the number of neighboring cells that have a current state of ‘living’:

- if the number is less than 2, then  $Q_+ = 0$  (death due to isolation);
- if the number is 2, then  $Q_+ = Q$  (survival);
- if the number is 3, then  $Q_+ = 1$  (birth);
- if the number is greater than 3, then  $Q_+ = 0$  (death due to overpopulation).

The loop machine requires a five-neighbor environment with 16 states per cell (Figure 3; the state-transition rules are given in the Appendix) and defines a minimal  $2 \times 2$  interactive loop. As long as no physical input is provided, the seven basic states, 0 to 6, lead to the continually undergoing four-time-step cycle of the idle loop (Figure 4). The additional states 7 to 15 take part in the eight-time-step self-replication process (Figure 5) or the eight-time-step self-destruction process (Figure 6), operated when the user activates one of the four cells of the idle loop.

The pseudo-random number generator supplies at each time-step one of the loop machine states 1 to 6. In order to produce randomly these six building states of the idle loop, the generator is implemented as an 8-bit linear feedback shift register working together with a decoder.

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

Figure 3: The seven basic cellular states 0 to 6 used for the idle loop and the nine additional cellular states 8 to 15 involved in the self-replication and self-destruction processes.

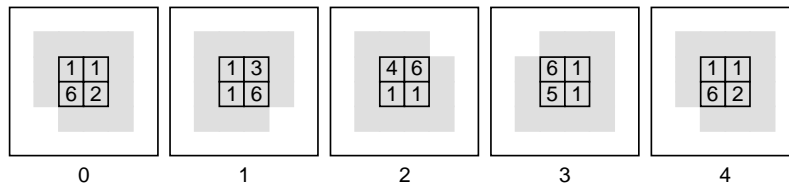


Figure 4: The four-time-step idle cycle of the inactive loop. The loop context appears in light grey.

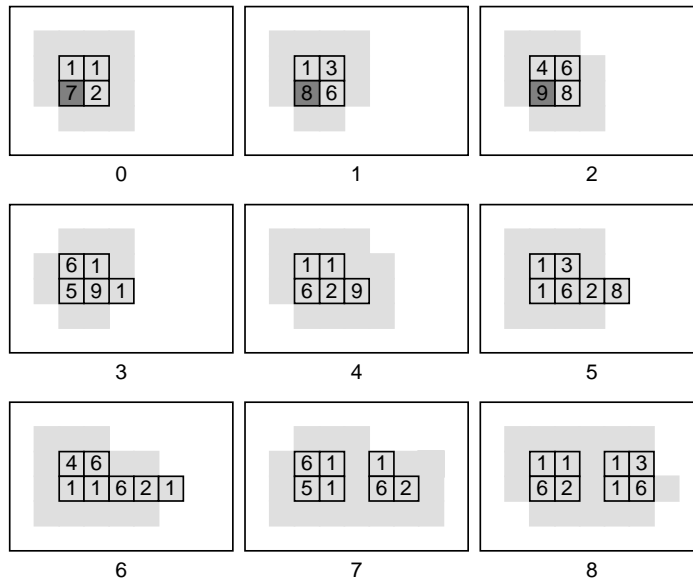


Figure 5: When the bottom-left cell is activated (touched), the loop self-replicates eastward in an eight-time-step process.

## 4 Emergence of the Loop

The two-dimensional cellular space is initialized as an empty Life environment where the user can induce the birth of cells by physically touching them. In this environment, each

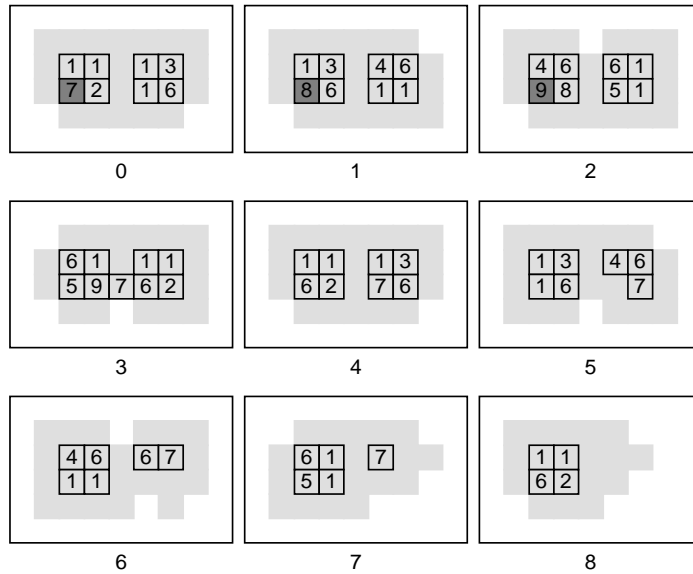


Figure 6: Activating the bottom-left cell results in an eight-time-step destruction process, since the loop is blocked by another loop and thus cannot self-replicate eastward.

living cell presents randomly one of the six idle loop states at each time-step. The context switch between Life and loop happens when a square block of four adjacent cells detects one of the four configurations of the idle loop (Figure 4). Without external activation, this newly created loop then remains idle. Depending on the absence or presence of surrounding loops, the physical activation of the idle loop induces a self-replication or self-destruction process. While performing these processes, the cellular space colonized by replication (Figure 5) shifts to the loop context and the cellular space freed by destruction (Figure 6) returns to the Life context.

## 5 Hardware Implementation

We have implemented the nine/five environment of the Life/loop CA in our interactive re-configurable computing tissue as an application of the BioWall (Figure 7). 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 either birth in the Life context or self-replication and self-destruction in the loop context (Figure 8).

The field-programmable gate array (FPGA) forming the cell's internal digital circuit is configured so as to implement: (1) the data processing of the external (touch) input, (2) the Life state machine, the loop state machine, the pseudo-random number generator, and the context-switching unit of the CA cell, and (3) 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 in both contexts and whether the loop cell is in activated or deactivated mode.

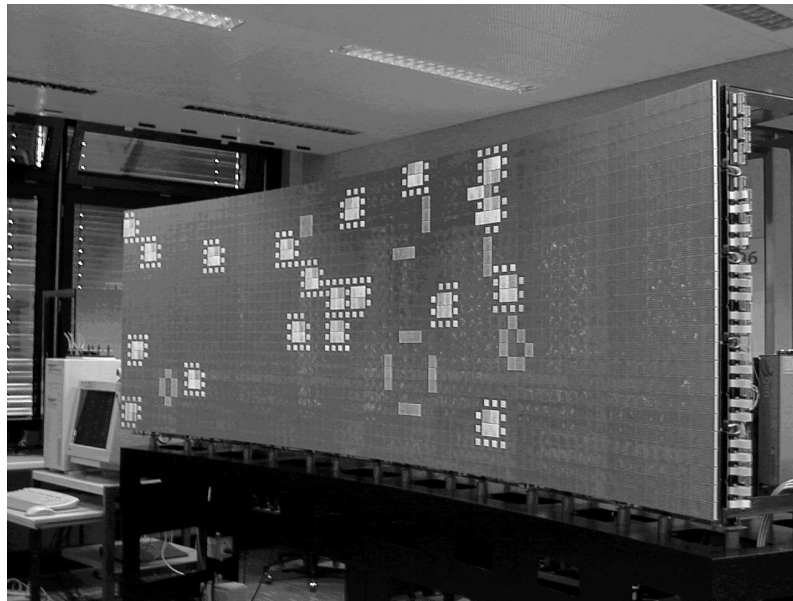


Figure 7: The BioWall used to physically implement our Life/loop CA. The dots surrounding the loops define visually the loop context (Photograph by A. Badertscher).

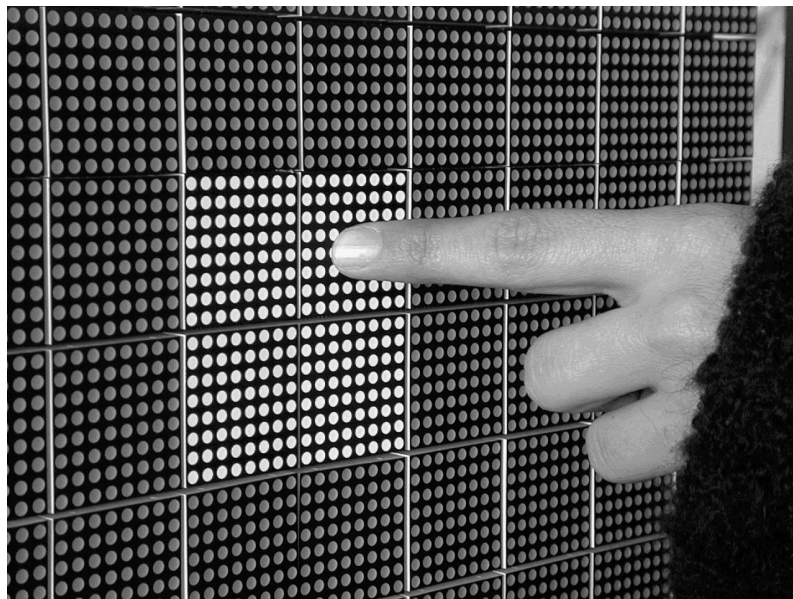


Figure 8: Touching the BioWall cells in order to physically give birth to a square block in the Life environment (Photograph by A. Badertscher).

## 6 Concluding Remarks

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 [1]. Our ongoing work concerns the design and emergence of interactive cellular replicators implemented in hardware, an issue which we believe will play an important role in the future of such devices.

## References

- [1] M. Sipper. Fifty years of research on self-replication: An overview. *Artificial Life*, 4:237–257, 1998.
- [2] M. Sipper and J. A. Reggia. Go Forth and Replicate. *Scientific American*, 285:2:26–35, August 2001.
- [3] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Illinois, 1966. Edited and completed by A. W. Burks.
- [4] J. A. Reggia, S. L. Armentrout, H.-H. Chou, and Y. Peng. Simple systems that exhibit self-directed replication. *Science*, 259:1282–1287, February 1993.
- [5] K. E. Drexler. *Nanosystems: Molecular Machinery, Manufacturing and Computation*. John Wiley, New York, 1992.
- [6] R. A. Freitas and W. P. Gilbreath (Eds.). *Advanced automation for space missions: Proceedings of the 1980 NASA/ASEE summer study, Chapter 5: Replicating systems concepts: Self-replicating lunar factory and demonstrations*, NASA, Scientific and Technical Information Branch (available from U.S. G.P.O., Publication 2255), Washington D.C., 1980.
- [7] C. Langton. Self-reproduction in cellular automata. *Physica D*, 10:135–144, 1984.
- [8] J. Byl. Self-reproduction in small cellular automata. *Physica D*, 34:295–299, 1989.
- [9] H. Sayama. Introduction of structural dissolution into Langton’s self-reproducing loop. In C. Adami, R. K. Belew, H. Kitano, and C. E. Taylor (Eds.), *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, pp. 114–122, MIT Press, Boston MA, 1998.
- [10] A. Stauffer and M. Sipper. Externally controllable and destructible self-replicating loops. In J. Kelemen and P. Sosik (Eds.), *Advances in Artificial Life: Proceedings of the 6th European Conference on Artificial Life (ECAL 2001)*. Lecture Notes in Artificial Intelligence, 2159:282–291, Springer-Verlag, Heidelberg, 2001.
- [11] M. Gardner. The fantastic combinations of John Conway’s new solitaire game “life”. *Scientific American*, 223:4:120–123, October 1970.
- [12] H. Chou and J. Reggia. Emergence of Self-Replicating Structures in a Cellular Automata Space. *Physica D*, 110:3–4:252–272, December 1997.

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

## Appendix: Specification of the Loop Machine State-Transition Rules

The state-transition rules of the machine implementing the physically controllable self-replicating and self-destructing loops are given in Figure 9. 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. 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 9: (1) 108 rules that are performed independently of the activation of the cell ( $A = -$ ), (2) 4 rules performed when the cell is inactive ( $A = 0$ ), and (3) 4 rules performed when the cell is active ( $A = 1$ ).

$C, N, E, S, W$	$\rightarrow C+$	$C, N, E, S, W$	$\rightarrow C+$	$C, N, E, S, W$	$\rightarrow C+$	$C, N, E, S, W$	$\rightarrow C+$
<b>A=-</b>							
0,0,0,2,0	$\rightarrow 1$	1,0,3,8,0	$\rightarrow 4$	5,6,0,1,0	$\rightarrow 6$	7,3,0,0,1	$\rightarrow 10$
0,0,0,2,1	$\rightarrow 3$	1,4,1,0,0	$\rightarrow 5$	5,6,9,0,0	$\rightarrow 6$	7,0,0,1,4	$\rightarrow 12$
0,0,3,0,0	$\rightarrow 1$	1,4,10,0,0	$\rightarrow 5$	5,6,11,0,0	$\rightarrow 6$	7,0,1,5,0	$\rightarrow 14$
0,0,3,1,0	$\rightarrow 4$	1,1,0,0,5	$\rightarrow 2$	5,6,0,14,0	$\rightarrow 6$	8,1,6,0,0	$\rightarrow 9$
0,4,0,0,0	$\rightarrow 1$	1,12,0,0,5	$\rightarrow 2$	5,6,1,15,0	$\rightarrow 6$	8,6,0,0,9	$\rightarrow 9$
0,4,1,0,0	$\rightarrow 5$	2,0,0,0,6	$\rightarrow 6$	5,7,1,0,0	$\rightarrow 6$	8,0,0,0,2	$\rightarrow 2$
0,0,0,0,5	$\rightarrow 1$	2,0,1,0,6	$\rightarrow 6$	6,1,2,0,0	$\rightarrow 1$	9,4,8,0,0	$\rightarrow 5$
0,1,0,0,5	$\rightarrow 2$	2,0,8,0,6	$\rightarrow 6$	6,3,0,0,1	$\rightarrow 1$	9,1,1,0,5	$\rightarrow 2$
0,0,0,0,8	$\rightarrow 1$	2,1,9,0,6	$\rightarrow 6$	6,0,0,1,4	$\rightarrow 1$	9,0,0,0,2	$\rightarrow 2$
0,0,0,0,9	$\rightarrow 8$	2,11,0,0,6	$\rightarrow 6$	6,0,1,5,0	$\rightarrow 1$	10,6,0,0,1	$\rightarrow 11$
0,0,0,10,0	$\rightarrow 1$	2,13,0,0,6	$\rightarrow 6$	6,3,2,0,1	$\rightarrow 1$	10,0,0,11,6	$\rightarrow 11$
0,0,0,11,0	$\rightarrow 10$	2,1,0,0,7	$\rightarrow 6$	6,3,0,1,4	$\rightarrow 1$	10,0,0,3,0	$\rightarrow 3$
0,0,12,0,0	$\rightarrow 1$	3,0,0,6,0	$\rightarrow 6$	6,0,1,5,4	$\rightarrow 1$	11,10,0,0,5	$\rightarrow 2$
0,0,13,0,0	$\rightarrow 12$	3,1,0,6,0	$\rightarrow 6$	6,1,2,5,0	$\rightarrow 1$	11,1,0,2,1	$\rightarrow 3$
0,14,0,0,0	$\rightarrow 1$	3,10,0,6,0	$\rightarrow 6$	6,12,2,0,0	$\rightarrow 1$	11,0,0,3,0	$\rightarrow 3$
0,15,0,0,0	$\rightarrow 14$	3,11,0,6,1	$\rightarrow 6$	6,3,0,0,14	$\rightarrow 1$	12,0,0,1,6	$\rightarrow 13$
1,0,0,0,2	$\rightarrow 2$	3,0,0,6,13	$\rightarrow 6$	6,0,0,8,4	$\rightarrow 1$	12,0,13,6,0	$\rightarrow 13$
1,0,0,3,0	$\rightarrow 3$	3,0,0,6,15	$\rightarrow 6$	6,0,10,5,0	$\rightarrow 1$	12,0,4,0,0	$\rightarrow 4$
1,0,4,0,0	$\rightarrow 4$	3,0,0,7,1	$\rightarrow 6$	6,3,0,0,8	$\rightarrow 8$	13,0,0,2,12	$\rightarrow 3$
1,5,0,0,0	$\rightarrow 5$	4,0,6,0,0	$\rightarrow 6$	6,0,0,10,4	$\rightarrow 10$	13,0,3,1,1	$\rightarrow 4$
1,0,0,0,9	$\rightarrow 9$	4,0,6,0,1	$\rightarrow 6$	6,0,12,5,0	$\rightarrow 12$	13,0,4,0,0	$\rightarrow 4$
1,0,0,11,0	$\rightarrow 11$	4,0,6,9,0	$\rightarrow 6$	6,14,2,0,0	$\rightarrow 14$	14,0,1,6,0	$\rightarrow 15$
1,0,13,0,0	$\rightarrow 13$	4,0,6,0,12	$\rightarrow 6$	6,0,2,0,1	$\rightarrow 0$	14,15,6,0,0	$\rightarrow 15$
1,15,0,0,0	$\rightarrow 15$	4,0,6,1,13	$\rightarrow 6$	6,3,0,1,0	$\rightarrow 0$	14,5,0,0,0	$\rightarrow 5$
1,0,0,2,1	$\rightarrow 3$	4,0,6,15,0	$\rightarrow 6$	6,0,1,0,4	$\rightarrow 0$	15,0,3,14,0	$\rightarrow 4$
1,0,0,2,14	$\rightarrow 3$	4,0,7,1,0	$\rightarrow 6$	6,1,0,5,0	$\rightarrow 0$	15,4,1,1,0	$\rightarrow 5$
1,0,3,1,0	$\rightarrow 4$	5,6,0,0,0	$\rightarrow 6$	7,1,2,0,0	$\rightarrow 8$	15,5,0,0,0	$\rightarrow 5$
<b>A=0</b>							
2,1,0,0,6	$\rightarrow 6$	3,0,0,6,1	$\rightarrow 6$	4,0,6,1,0	$\rightarrow 6$	5,6,1,0,0	$\rightarrow 6$
<b>A=1</b>							
2,1,0,0,6	$\rightarrow 7$	3,0,0,6,1	$\rightarrow 7$	4,0,6,1,0	$\rightarrow 7$	5,6,1,0,0	$\rightarrow 7$

Figure 9: Loop machine state-transition rules.