

# Novel reversible logic elements for unconventional computing

Kenichi Morita<sup>1</sup>

Logic elements used in conventional computers are mostly logic gates such as AND, OR, NOT, NAND, etc. Their history is quite long, since logical operations of AND, OR and NOT were already known more than two-thousand years ago.<sup>2</sup> They were obtained by analyzing human thinking and reasoning, and thus it is easy for us to understand them. However, when we investigate future computing systems, we should not be tied to the old traditions. Reversible computing<sup>3</sup> is a paradigm of computing that reflects physical reversibility, one of the fundamental microscopic physical laws of nature. It searches for novel and unconventional methodologies that are directly related to reversible microscopic phenomena. We consider here a reversible logic element with 1-bit memory (RLEM), and examine its possibilities as a novel logical device. Though its physical realizability in the nano-level is not known at present, it gives new vistas on unconventional computing devices. We shall see that it has very different features that cannot be seen in conventional logic gates. In particular, reversible computing systems such as reversible Turing machines can be constructed out of RLEMs in a very unique way. We also see that an RLEM can be implemented in reversible environments such as a billiard ball model of computation, and a very simple reversible cellular automaton. In the following, these features are explained using many illustrations without giving technical details.

## Reversible logic element with 1-bit memory

A reversible logic element with 1-bit memory (RLEM) is a kind of reversible finite automaton. Fig. 1 shows a typical example of an RLEM with four input ports and four output ports called a *rotary element* (RE).<sup>4</sup> Conceptually, it has a rotatable bar that controls the move direction of an incoming signal (or a particle). It takes one of the two states H and V depending

on the direction of the bar (i.e., horizontal or vertical). If a signal comes from the direction parallel to the bar, it goes straight ahead and the state does not change. If a signal comes from the direction orthogonal to the bar, it turns rightward and the state changes. It is reversible in the following sense: From the state at  $t+1$  and the output, the state at  $t$  and the input are uniquely determined.

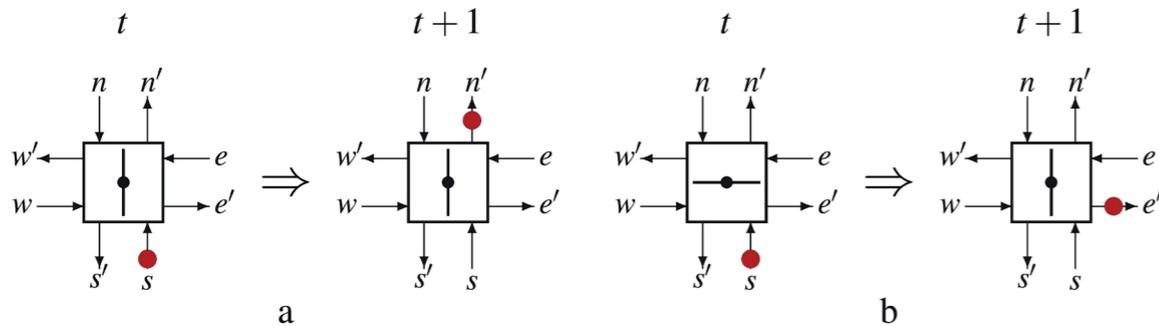


Fig. 1. Rotary element (RE) and its operations. (a) The parallel case, and (b) the orthogonal case.

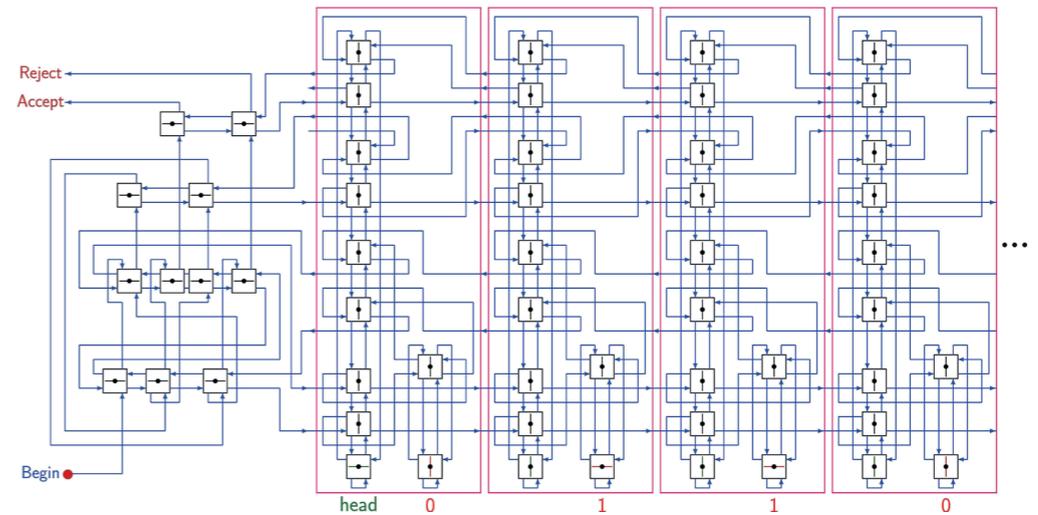


Fig. 2. An example of a reversible Turing machine composed of rotary elements.

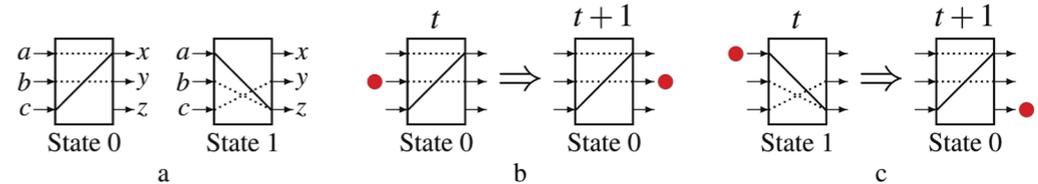


Fig. 3. (a) Two states of RLEM 3-7. (b) The case where the state does not change. (c) The case where the state changes.

## Composing reversible computers out of rotary elements

A reversible Turing machine is an abstract model of a reversible computer where every computational state has at most one predecessor. Hence, we can trace back its computing process uniquely. It is known that any (irreversible) Turing machine can be simulated by reversible one without leaving garbage information on the tape, and thus reversible Turing machines are computationally universal.<sup>5</sup> We can construct any reversible Turing machine out of rotary elements. Fig. 2 shows a circuit that simulates a simple reversible Turing machine that checks if a unary number given on its tape is even.<sup>6</sup> In this figure, a finite-state control of the reversible Turing machine is in the left part, and a tape unit is in the right part. If a particle is given to the Begin port, it starts to compute.

## Universal RLEMs

There are infinitely many RLEMs if we do not restrict the number of input/output ports. Fig. 3 shows RLEM No. 3-7, where “3” stands for 3-input and 3-output, and “7” is its serial number. Two boxes in Fig. 3(a) indicate its two states. The dotted and solid lines give input-output relation in each state. If an input signal goes through a dotted line, the state does not change (Fig. 3(b)). If it goes through a solid line, the state changes (Fig. 3(c)). Note that RE can also be represented by such a figure, but we employ Fig. 1 for ease in understanding. An RLEM is called *universal* if it can simulate any other RLEM. Remarkably, it has been proved that every RLEM (except degenerate ones) is universal if it has three or more input/output ports.<sup>7</sup> Therefore, RLEM 3-7 and RE are of course universal. Figure 4 shows how to simulate an RE by RLEM 3-7. Replacing each occurrence of REs in Fig. 2 by the circuit of Fig. 4, we obtain

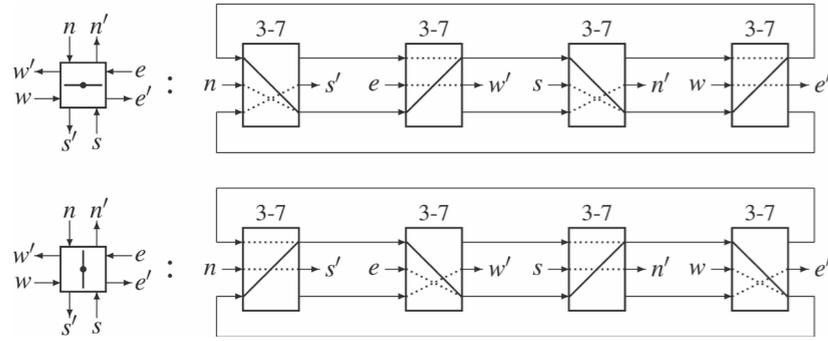


Fig. 4. Simulating a rotary element by a circuit composed of RLEM 3-7.

Fig. 5. A rotary element realized in the billiard ball model, an idealized mechanical computing model.

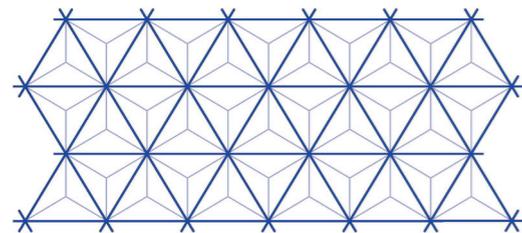
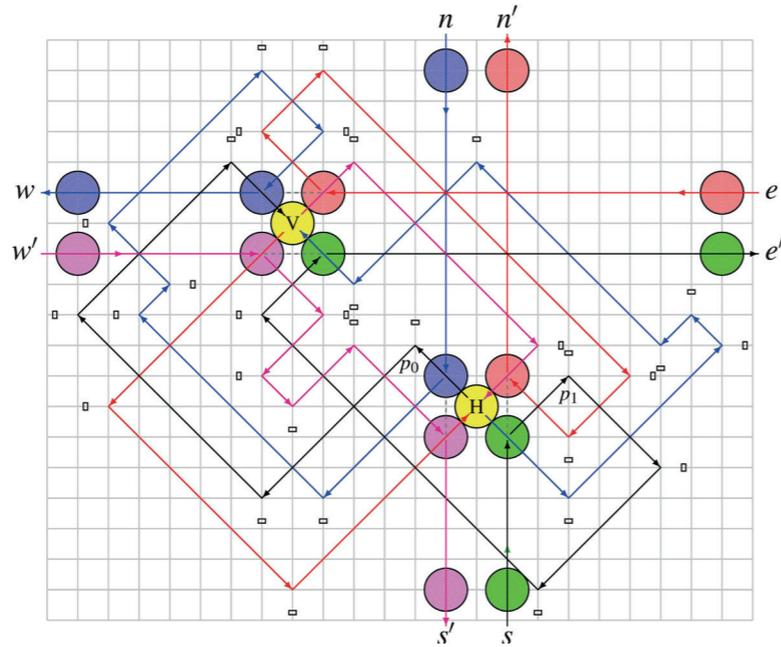


Fig. 6. Cellular space of an elementary triangular partitioned cellular automaton (ETPCA).

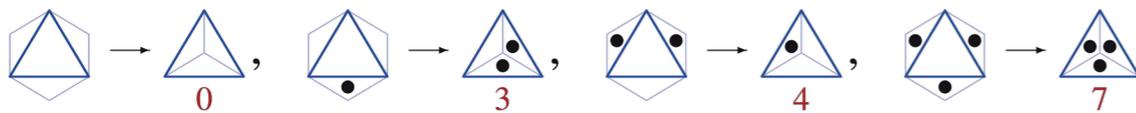


Fig. 7. Local transition rules of ETPCA 0347.

a circuit made of RLEM 3-7 that simulates the reversible Turing machine.

### Simulating a rotary element using billiard balls

The *billiard ball model* (BBM) of computation was proposed by Edward Fredkin and Tommaso Toffoli<sup>8</sup> to show that the Fredkin gate, a universal reversible logic gate, is realizable in the BBM. It is a kind of reversible mechanical system consisting of ideal elastic balls and reflectors. We can see that an RE is also realized in the BBM as shown in Fig. 5, where small rectangles are reflectors.<sup>9</sup> Here, two kinds of balls, *i.e.*, a *state ball* and a *signal ball*, are used. One of the key points of the construction is that the state ball (yellow) is put stationarily at the position H or V in a resting mode. Fig. 5 shows the cases where the directions of the bar and the incoming signal are orthogonal as in Fig. 1(b). Consider the case that the state ball is put at the position H, and the signal ball (green) comes from the input port  $s$ . The signal ball collides with the state ball at the position H. Then, the state ball and the signal ball move along the paths  $p_0$  and  $p_1$ , respectively. When the state ball comes to the position V, these balls collide again. Then, the state ball stops at V, while the signal ball moves eastward and goes out from the output port  $e'$ . By this the operation of Fig. 1(b) is realized. The case that the state ball is at the position V, and the signal ball comes from  $s$  is trivial. In this case, the signal ball simply moves northward without interacting with the state ball and the reflectors, and thus Fig. 1(a) is realized. In this way, the whole circuit that simulates a reversible Turing machine given in Fig. 2 is also embeddable in the BBM.

### Realizing RLEMs in a simple reversible cellular space

There is yet another spatiotemporal model of a reversible environment called a *reversible cellular automaton*. A cellular automaton (CA) consists of an infinite number of finite automata called *cells* that are placed and connected uniformly in a space. A cell changes its state depending on the states of its neighboring cells. We use a special type of a CA called an

*elementary triangular partitioned cellular automaton* (ETPCA) since it is very simple. Each cell of ETPCA is triangular, and it is further divided into three parts (Fig. 6). Each part has two states 0 and 1, which are indicated by a blank and a dot. Here, we consider a particular local transition function defined by the four local transition rules shown in Fig. 7, which is identified by the number 0347.<sup>10</sup> We assume that these rules are rotation-symmetric, *i.e.*, for each rule in Fig. 7 there exist rules obtained by rotating both sides of it by a multiple of 60 degrees. As seen from Fig. 7, the next state of a cell is determined by the three parts of its neighboring cells.

Consider the configuration given at time  $t=0$  in Fig. 8. Applying the local function to all the cells in parallel, we have a configuration at  $t=1$ . Configurations at  $t=2,3,\dots$  are obtained likewise. Note that, the dot patterns at  $t=0$  and  $t=6$  are the same except that the latter is shifted rightward. Hence, it is a space-moving pattern called a *glider*, which can be used as a signal. The local transition function defined by the rules in Fig. 7 is injective, since there is no pair of distinct rules whose right-hand sides are the same. By this, for each configuration we can find a previous configuration uniquely. Such an ETPCA is called *reversible*.

The moving direction of a glider is controlled by a stable pattern called a *block*. Figure 9 shows the process of the backward turn by a single block. At  $t=0$  a glider (left) is about to collide a block (right). At  $t=38$  the glider is split into a *rotator* (left) and a *fin* (right). The fin moves around the block. At  $t=97$  the rotator and the fin meet, and a glider is reconstructed. Then the resulting glider moves leftward. Appropriately placing several blocks, we can also realize the right turn by 120 degrees, the left turn by 120 degrees, and the U-turn of a glider.

Combining several useful phenomena found in the reversible cellular space of ETPCA 0347, we can construct a pattern that simulates RLEM 3-7 (Fig. 3) as shown in Fig. 10. A glider is given to one of the three input ports as a signal. After changing the state, the glider

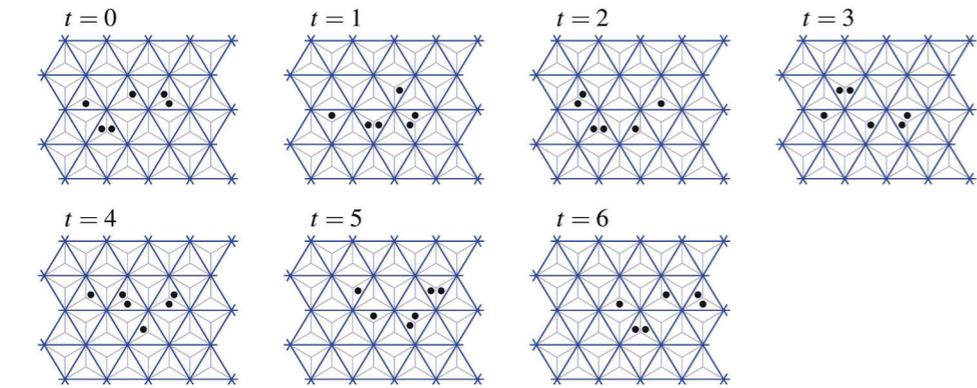


Fig. 8. A space-moving pattern called a glider in ETPCA 0347.

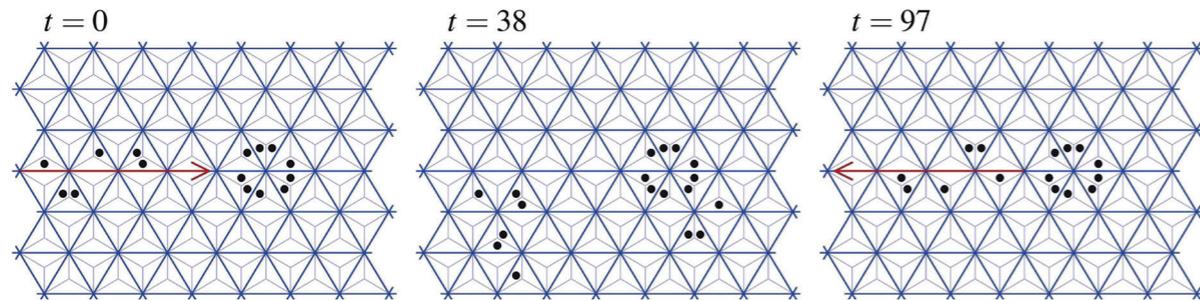


Fig. 9. Backward turn of a glider is realized by a collision with a block.

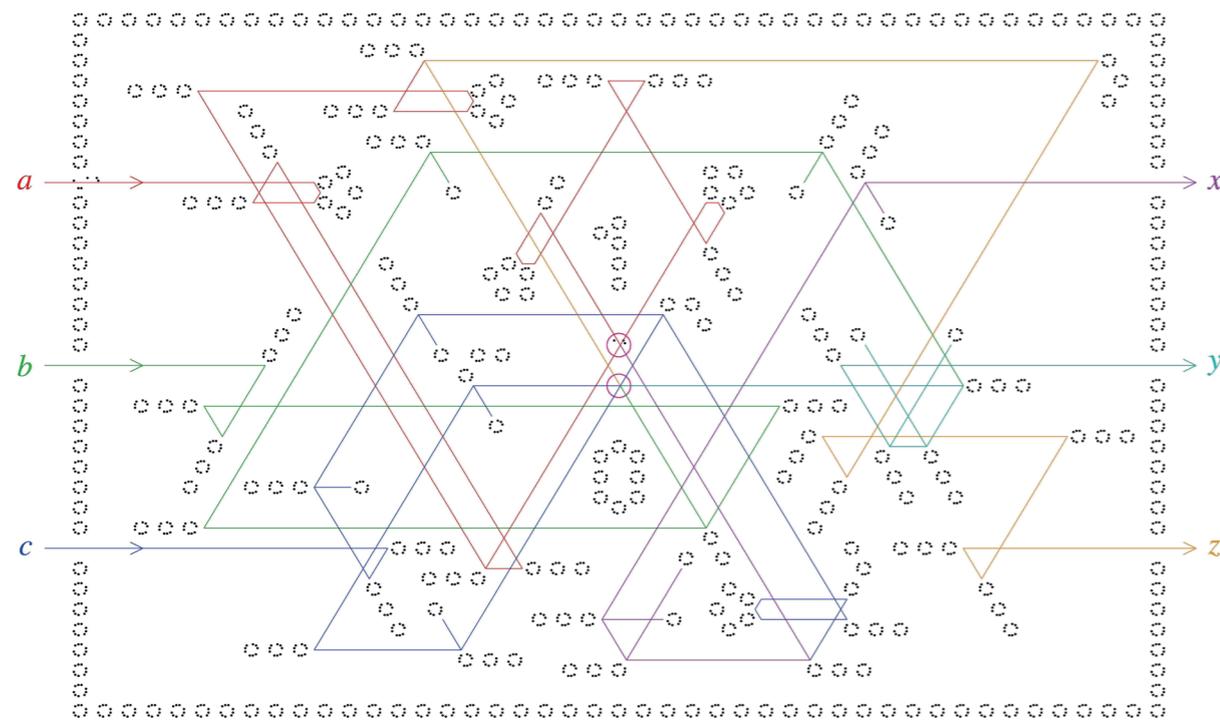


Fig. 10. RLEM 3-7 implemented in the reversible cellular space of ETPCA 0347.

comes out from one of the output ports. The pattern contains many backward turn, right turn, and U-turn modules. To keep the state of RLEM 3-7, a fin is put around the center of the pattern. A fin is a periodic pattern of period 6 consisting of three dots. Two small circles in the figure show possible positions of a fin. If a fin is at the lower (upper, respectively) position, then we consider the state is 0 (1). Testing whether the fin is at the position 0 or 1, and shifting it between these two positions are performed by collisions of the glider. In this way, RLEM 3-7 is simulated in the cellular space of ETPCA 0347 having an extremely simple local function. Since a rotary element can be composed of RLEM 3-7, circuits that simulate reversible Turing machines like the one in Fig. 2 are also simulated in this cellular space. An emulator for ETPCA 0347 has been given,<sup>11</sup> which works on the general purpose cellular automaton simulator *Golly*.<sup>12</sup> There, computing processes of reversible Turing machines composed of RLEM 4-31 can be seen.

<sup>1</sup> Professor Emeritus, Hiroshima University, Higashi-Hiroshima, Japan.  
<sup>2</sup> J. M. BOCHENSKI, *Ancient Formal Logic*, Amsterdam, North-Holland, 1951.  
<sup>3</sup> K. MORITA, *Theory of Reversible Computing*, Tokyo, Springer, 2017.  
<sup>4</sup> K. MORITA, "A simple reversible logic element and cellular automata for reversible computing," in M. MARGENSTERN & Y. ROGOZHIN (eds), *Proceedings of MCU (International Conference on Machines,*

*Computations, and Universality*) 2001, LNCS 2055, Berlin, Heidelberg, Springer, 2001, 102-113.

<sup>5</sup> C. H. BENNETT, "Logical reversibility of computation," *IBM J. Res. Dev.* 17, 1973, 525-532.

<sup>6</sup> K. MORITA, "Constructing reversible Turing machines by reversible logic element with memory," in A. ADAMATZKY (ed.), *Automata, Universality, Computation*, Cham, Springer, 2015, 127-138. Hiroshima University Institutional Repository, <http://ir.lib.hiroshima-u.ac.jp/00029224> (2010).

<sup>7</sup> K. MORITA, T. OGIRO, A. ALHAZOV, and T. TANIZAWA, "Non-degenerate 2-state reversible logic elements with three or more symbols are all universal," *Journal of Multiple-Valued Logic and Soft Computing* 18, 2010, 37-54.

<sup>8</sup> E. FREDKIN & T. TOFFOLI, "Conservative logic," *International Journal of Theoretical Physics* 21, 1982, 219-253.

<sup>9</sup> K. MORITA, *Theory of Reversible Computing*, op. cit.

<sup>10</sup> K. MORITA, "A universal non-conservative reversible elementary triangular partitioned cellular automaton that shows complex behavior," *Natural Computing* 18(3), 2019, 413-428.

<sup>11</sup> K. MORITA, "Reversible world: Data set for simulating a reversible elementary triangular partitioned cellular automaton on Golly," Hiroshima University Institutional Repository, <http://ir.lib.hiroshima-u.ac.jp/00042655> (2017).

<sup>12</sup> A. TREVORROW, T. ROKICKI, T. HUTTON *et al*, "Golly: an open source, cross-platform application for exploring Conway's Game of Life and other cellular automata," <http://golly.sourceforge.net/> (2005).