An Asynchronous Cellular Automaton Implementing 2-State 2-Input 2-Output Reversed-Twin Reversible Elements

Jia Lee[1], Ferdinand Peper[2], Susumu Adachi[2],
and Kenichi Morita[3]

[1] Celartem Technology Inc., Japan
lijia@celartem.com
[2] Nano ICT Group,
National Institute of Information and Communications Technology, Japan
[3] Dept. of Information Engineering, Hiroshima University, Japan

Abstract. Reversible computers usually work in a synchronous mode, i.e., in the presence of clock signals, but in the light of the asynchronous nature of microscopic physical phenomena this may be an anomaly. The alternative, an asynchronous mode of operation, has therefore attracted attention from researchers, witness the proposal of a reversible circuit element in (Morita 2001) that works in such a mode. Simplicity of circuit elements like this is an important design objective since it correlates positively with the efficiency by which they may be realized physically. In this paper, we present two mutually inverse logic elements that compare favorably to other circuit elements in terms of their number of states and their number of input and output lines. We show that the proposed circuit elements can perform universal computation by embedding circuits made of them in asynchronous cellular automata.

1 Introduction

Reversible logic has its origins in computing schemes that achieve near-zero power consumption by preventing entropy loss in computations. It has been extensively studied [1,2,3,4,5,6], but always under the assumption that timing is synchronous, i.e., that all logic elements switch simultaneously in accordance with a central clock. For example, the Fredkin gate [4], a well-known reversible gate, fails to work correctly if all its input signals would arrive at different times. Asynchronous systems have virtually been unexplored for reversible computing, probably since the randomness by which events in them are timed appears incompatible with the backward determinism of reversible computing. This lack of interest may be hard to defend in the light of the existence of microscopic physical interactions that are both asynchronous and reversible. As with reversibility, asynchrony tends to reduce power consumption, be it for different reasons: logic elements in an asynchronous system need not be active in the absence of signals, unlike in synchronous systems, in which idle logic elements may engage in dummy switching events triggered by the continuous arrival of clock signals [7].

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 67–76, 2008.

To achieve low-power computing in practice, it makes sense to investigate the combination of asynchrony and reversibility and particularly to find asynchronous reversible logic elements to be used as the basic building blocks to construct universal circuits. Intuitively, the less complex a logic element is, the easier it can be implemented physically—a reason to look for elements with as few input lines, output lines, and internal states as possible. Unsuitable for use in an asynchronous framework are the reversible logic elements typically employed in synchronous circuits, since they lack the timing functionality to make up for the absence of a clock. A straightforward measure of a circuit element's complexity is the ease by which it can be implemented in cellular automata: for, a complex functionality usually translates in an increased number of cell states and transition rules.

Patra and Fussell [8] has studied asynchronous reversible systems in the context of *Delay-Insensitive (DI) circuits*. A delay-insensitive circuit (e.g. see [7]) is an asynchronous circuit in which signals may be subject to arbitrary delays without this being an obstacle to the circuit's correct operation. The circuits constructed in [8] are not reversible in the strict sense, since his constructions require a *Merge-element*—an irreversible element that merges two input streams of signals into a single output stream of indistinguishable signals.

Morita [9], has proposed a DI reversible logic element, called a *Rotary Element* (RE), from which computationally universal models can be constructed, including a reversible Turing machine. The RE has four input lines, four output lines, and two internal states. An improved element with three input lines, three output lines, and two internal states is proposed in [10], and more such elements are investigated in [11]. In both references [10,11] it is proven that each of the proposed elements can be used as a basis into which the RE can be decomposed, which implies that the elements are universal. In [12] an asynchronous reversible cellular automaton is proposed that implements the RE. The construction of the reversible Turing machine from REs in [9] implies the universality of this cellular automaton.

This paper proposes a pair of DI reversible logic elements each of which has two input lines, two output lines, and two internal states. These elements are each other's functional reverses, i.e., running signals backwards through one element produces the equivalent of the other element. We prove that the two elements as a set are universal by constructing an RE from them. The asynchronicity of these two elements combined with their mutually reversed functionalities enable efficient implementation in a special type of asynchronous cellular automata, called *Self-Timed Cellular Automata* (STCA) [13], such that merely five transition rules are required.

Section 2 defines the proposed reversible elements in detail. In section 3 an RE is constructed from the proposed elements; this result implies the universality of the elements, since a universal reversible Turing machine can be constructed from RE elements [9]. Implementation of these two elements and circuits based on them in terms of an asynchronous cellular automaton are described in section 4. The paper finishes with conclusions and a short discussion.

2 Reversible Logic Elements

A *reversible sequential machine* [9] is a system defined as $N = (Q, \Sigma, \Delta, \delta)$, where Q is a finite set of states ($Q \neq \emptyset$), Σ and Δ are sets of input and output symbols, respectively. The transition function $\delta : \; Q \times \Sigma \rightarrow Q \times \Delta$ is bijective. A reversible sequential machine is a special type of Mealy machine [14].

A *reversible serial module* is a system defined as $M = (I, O, N)$, where I and O are two sets of input and output lines, respectively ($I \cap O = \emptyset$). $N = (Q, \Sigma, \Delta, \delta)$ is a reversible sequential machine with I in one-to-one correspondence with Σ, and O in one-to-one correspondence with Δ.

Signals used for inputs and outputs of a reversible serial module are considered particles. The binary signals 1 and 0 are encoded by the presence or absence, respectively, of a particle on a line. Let $\mu : \; I \rightarrow \Sigma$ be the bijective function between I and Σ, and $\nu : \; O \rightarrow \Delta$ be the bijective function between O and Δ. A reversible serial module M is said to be in state $q (\in Q)$ if the reversible sequential machine N of M is in state q. Assume $a \in I, \; b \in O$ and $q, q' \in Q$ such that $\delta(q, \mu(a)) = (q', \nu^{-1}(b))$. Then if a particle arriving on input line a is received by M in state q, M operates on this particle such as to transfer it from a to output line b, and to change M's state from q to q'. The operation of M is undefined for simultaneous input signals on its input lines. In other words, a reversible serial module can only process at most one input particle at any time. The operation of M is reversible, in that from the current state and output, the previous state and input can be uniquely determined due to the bijective transition function δ.

We present two reversible serial modules, which have inverse functionalities. One of the modules, called the *Reading Toggle* (RT) element, is defined as $(\{S, T\}, \; \{T_A, T_B\}, \; N_{RT})$, where $N_{RT} = (\{A, B\}, \; \Sigma_{RT}, \; \Delta_{RT}, \; \delta_{RT})$ (see Fig. 1). Let $\mu : \; \{S, T\} \rightarrow \Sigma_{RT}$ be the bijective function between $\{S, T\}$ and Σ_{RT}, and $\nu : \; \{T_A, T_B\} \rightarrow \Delta_{RT}$ be the bijective function between $\{T_A, T_B\}$ and Δ_{RT}. The RT element operates such that a particle arriving on input line T is transferred to output line $T_A (T_B)$ if the RT is in state A(B); in this case, the state changes to B(A) (upper row of Fig. 2(a)). A particle arriving on input line S is merely transferred to output line $T_A (T_B)$ if the RT is in state A(B), without the state being changed (lower row of Fig. 2(a)).

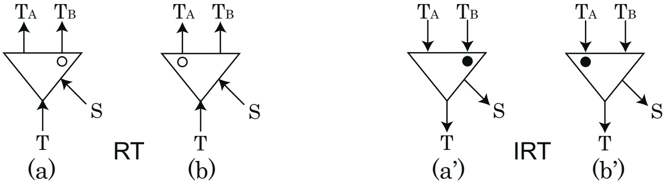

Fig. 1. RT element in (a) state A, and (b) state B. IRT element in (a') state A, and (b') state B.

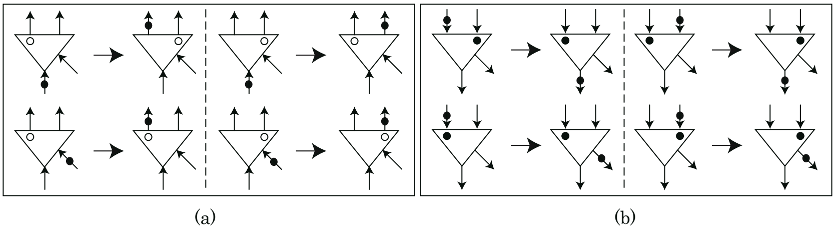

Fig. 2. (a) RT and (b) IRT operating on a particle arriving on one of its input lines. The particle is denoted by a token on a line. The two transition rules of (a) in the upper resp. lower row describe the RT's operation in case of a particle arriving on line T resp. S. In addition, the two transition rules of (b) in the upper resp. lower row describe the IRT's operation in case of a particle being output to line T resp. S.

The other reversible serial module, the *Inverse Reading Toggle* (IRT) element, is defined as $(\{T_A, T_B\}, \{S, T\}, N_{IRT})$ where N_{IRT} is defined as $N_{IRT} = (\{A, B\}, \Sigma_{IRT}, \Delta_{IRT}, \delta_{IRT})$. Let $\mu' : \{S, T\} \rightarrow \Sigma_{IRT}$ be the bijective function between $\{T_A, T_B\}$ and Σ_{IRT}, and $\nu' : \{T_A, T_B\} \rightarrow \Delta_{RT}$ be the bijective function between $\{S, T\}$ and Δ_{IRT}. The IRT element operates such that a particle arriving on input line $T_A(T_B)$ is transferred to output line T if the IRT is in state B(A); in this case, the state changes to A(B) (upper row of Fig. 2(b)). A particle arriving on input line $T_A(T_B)$ is merely transferred to output line S if the IRT is in state A(B), without the state being changed (lower row of Fig. 2(b)). Simultaneous particles on the input lines of RT or IRT are not allowed. Obviously, both RT and IRT are reversible, and they are each other's inverse.

3 Construction of RE by the Reversible Elements

Any reversible Turing machine (for more details on such machines see [1,15]) can be constructed by using a network of REs, in which at most one particle moves around at any time [9]. Since delays in any of the REs or lines do not affect the correctness of the computing process in the circuit, this circuit is DI. Such reversible computers consisting of REs need no central clock signal to drive the operations of each RE [9], i.e., they are asynchronous.

We construct an RE from RT and IRT elements to show the universality of the RT and IRT elements in an asynchronous mode of operation. An RE is a reversible serial module that is defined as $(\{n, e, s, w\}, \{n', e', s', w'\}, N_{RE})$ with $N_{RE} = (\{H, V\}, \Sigma_{RE}, \Delta_{RE}, \delta_{RE})$ (see Fig. 3). Let $\hat{\mu} : \{n, e, s, w\} \rightarrow \Sigma_{RE}$ be the bijective function between $\{n, e, s, w\}$ and Σ_{RE}, and $\hat{\nu} : \{n', e', s', w'\} \rightarrow \Delta_{RE}$ be the bijective function between $\{n', e', s', w'\}$ and Δ_{RE}. The RE operates such that if a particle comes from a direction parallel to the rotating bar of an RE, it passes straight through to the opposite output line, without changing the direction of the bar (the state of the RE), as in Fig. 4(a); if the particle comes from a direction orthogonal to the rotating bar, it is deflected to the right, and the bar rotates by 90 degrees (Fig. 4(b)). An RE remains in its state if no particle

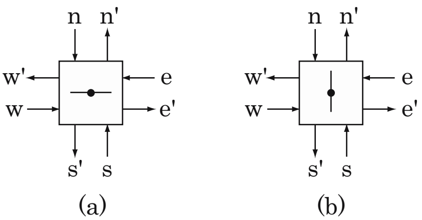

Fig. 3. An RE in (a) the H-state, and (b) the V-state, displayed as respectively horizontal and vertical bars in the RE

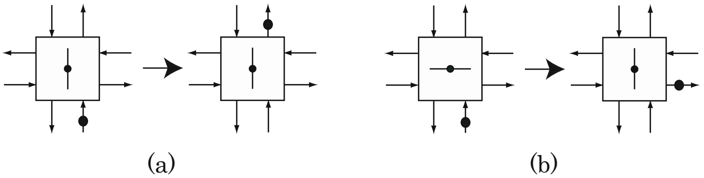

Fig. 4. REs operating on an input particle in (a) the parallel case, and (b) the orthogonal case

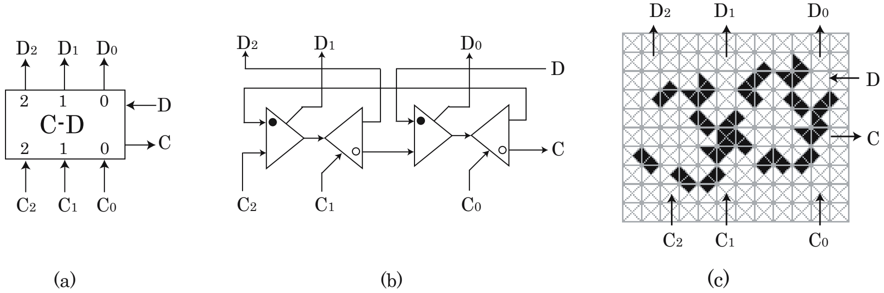

Fig. 5. (a) A C-D module. (b) The realization of a C-D module in state 0 from RT and IRT elements. (c) Construction of C-D module in the STCA (see Section 4 for details).

arrives on any of its input lines. Simultaneous particles on any pair of input lines of an RE are not allowed.

In [9] is was shown that the RE element is universal by composing a circuit of RE modules that can simulate a universal Turing machine. To show that any reversible Turing machine can be realized from RT and IRT elements, it suffices to construct an RE out of these elements. We first construct an intermediate module from the RT and IRT elements to simplify the construction of the RE. This module, called a *Coding-Decoding* (C-D) module [10] (see Fig. 5), has four input lines $\{C_0, C_1, C_2, D\}$, four output lines $\{D_0, D_1, D_2, C\}$, and three states $\{0, 1, 2\}$. If the C-D module is in state 0, an input particle arriving on input line C_i $(i \in \{0, 1, 2\})$ changes the state of the C-D module from 0 to i, and the particle is transferred to output line C. Then, a subsequent particle coming

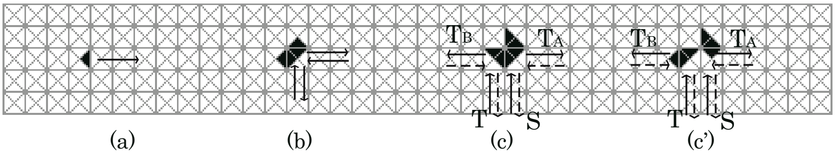

Fig. 6. (a) A signal. A subcell in state 1 is denoted by a filled triangle, while a subcell in state 0 is denoted by a blank. (b) A left or right turn element. (c) The configuration representing an RT or IRT element in state A. The direction of the input and output signals can be both ways: one way, indicated by the solid arrows, makes the configuration work as an RT element, whereas the other way, indicated by dashed arrows, corresponds to an IRT element. (c') Configuration representing an RT or IRT element in state B. The arrows indicate the directions of signal propagation for the RT and the IRT elements.

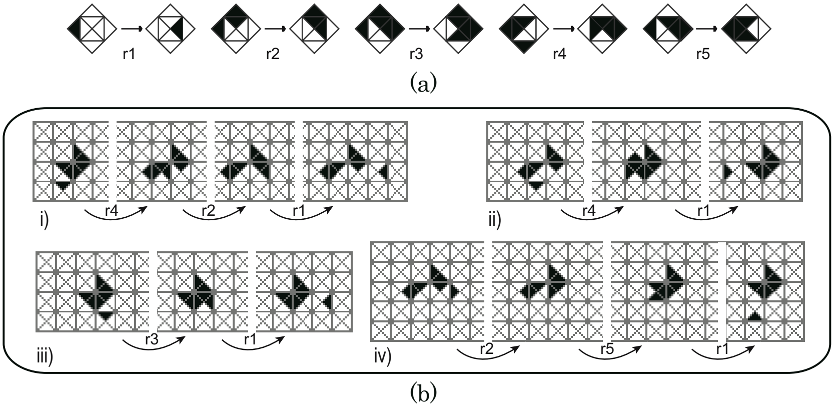

Fig. 7. (a) Transitions rules of STCA. The rotational and reflective equivalents of the rules are omitted. (b) RT and IRT elements operating on input signals: An RT element receiving a signal from input path T when it is in state i) A or ii) B. iii) The RT receiving an input signal from path S when it is in state A. iv) An IRT receiving a signal from input path T_A when it is in state B. Each arrow indicates one transition step of cells, whereby its label refers to the corresponding transition rule. It can be verified that the RT (or IRT) element here will fail to work on a signal arriving on its input path S (resp. T_B) when it is in state B. Implementation of the full functionalities of RT and IRT elements is possible, but this tends to increase the number of rules and result in more complicated cellular configurations as compared to those in Fig. 6.

from input line D is transferred to output line D_i if the C-D module is in state i, and the state is reset to 0. The C-D module is unable to receive input from lines C_0, C_1, or C_2 if it is in states 1 or 2. We apply the C-D module in the construction of the RE such that an input particle arriving on an input line of the C-D module is always followed by an output particle on an output line, before a new particle is input to an input line.

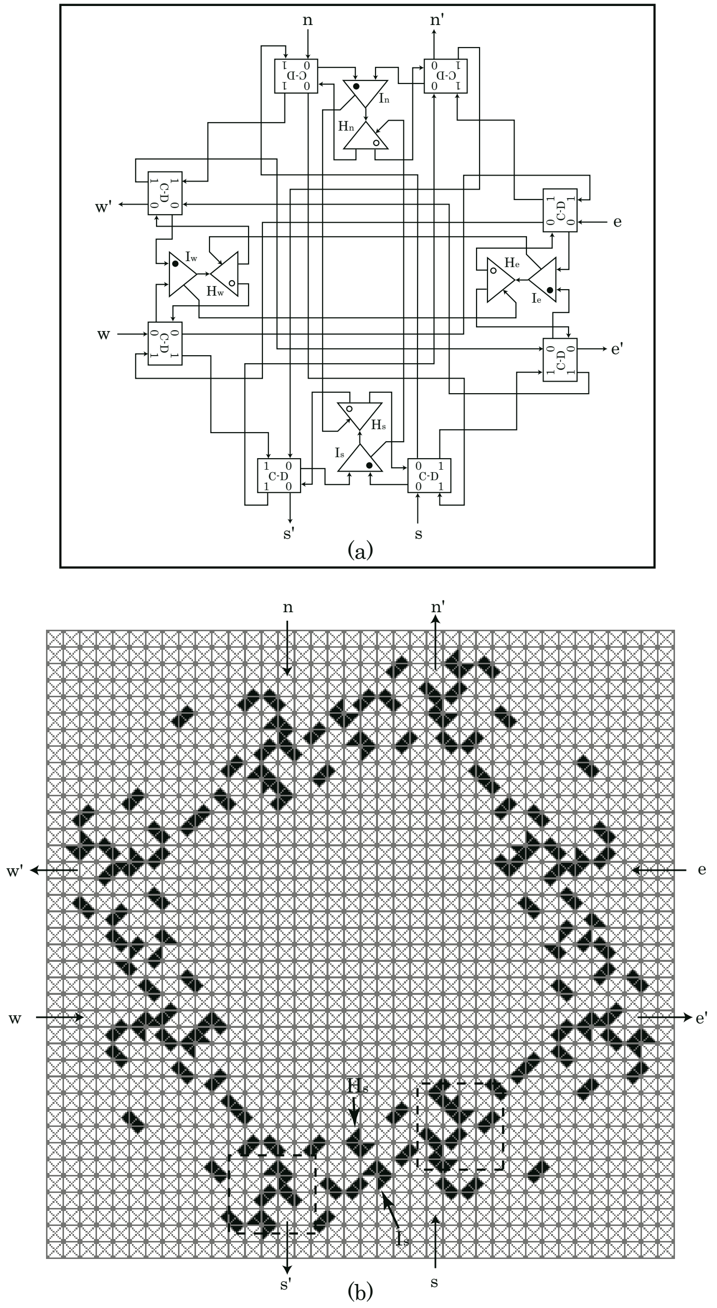

Fig. 8. (a) Realization of an RE in state V from RT and IRT elements, in which all C-D modules are in state 0 initially. (b) Construction of this RE in the STCA. Dashed boxes are put around the areas in (b) in which C-D modules adjacent to the elements H_s and I_s are placed according to the construction in (a).

Figure 8(a) shows in detail the realization of an RE from RT and IRT elements, whereby the four subcircuits of RT and IRT elements, indicated by (H_s, I_s), (H_e, I_e), (H_n, I_n), and (H_w, I_w), are in states (A, A), (B, B), (A, A), and (B, B), respectively; this represents the RE being in the V-state. The same subcircuits being in states (B, B), (A, A), (B, B), and (A, A), respectively, represent the RE being in the H-state. This result implies that we can construct a circuit from RT and IRT elements that simulates a reversible Turing machine, in which at most one particle moves around the entire circuit [9]. Thus, the RT and IRT elements are logically universal, and can work in asynchronous mode, i.e. without their operations having to be driven by a central clock.

Finally, from the constructions in Figs. 5(b) and 8(a), it can be observed that each RT (or IRT) element never receives a signal from input line S (resp. T_B) when it is in state B. This implies that the functionalities of the RT and IRT can be further simplified, which tends to benefit their implementations, of these two elements on asynchronous cellular automata, as the next section shows.

4 Embedding Reversible Elements in STCA

A Self-Timed Cellular Automaton is a two-dimensional array of identical cells. Each cell is partitioned into four subcells in one-to-one correspondence with its four nearest neighboring cells, and each subcell takes only one of two states: 0 or 1. Each cell undergoes state transitions via transition rules that operate on the cell itself along with the nearest subcells of each of its four neighbors. Moreover, the update of cells are timed randomly and independently of each other, and hence, are asynchronous.

Figure 6 shows some fundamental patterns used in the STCA. In particular, the pattern in Fig. 6(a) represents a signal that will be transferred to the right; whereas the pattern in Fig. 6(b) is used to change the direction of a signal to the left or right. Moreover, both the local configurations in Fig. 6(c) and (c') represent an RT or an IRT element. Their difference corresponds to the two internal states: A and B, of an RT or IRT element, respectively. The update of all the patterns in Fig. 6 are controlled by the five transition rules given in Fig. 7(a), for example, as demonstrated in Fig. 7(b).

Following the construction in Fig. 5(b), we are able to lay out an C-D module in our asynchronous cellular automaton by the configuration illustrated in Fig. 5(c). Furthermore, in accordance with the circuit scheme in Fig. 8(a), we lay out the configuration of an RE on the STCA in Fig. 8(b). This implies that a Turing machine can be constructed on the cellular automaton, provided the cellular space is sufficiently large.

5 Conclusions and Discussion

This paper proposes two asynchronous reversible logic elements that are each others' mutually reverse. Called RT and IRT, the elements have two input lines, two output lines, and two states. The elements are universal, because they can be used to construct Morita's RE element [9], which is universal. Moreover, the elements are less complex than the RE. Following the line of thought in [9], we can construct a universal reversible computer from RTs and IRTs, in which at most one particle moves around at any time. This allows the RT and IRT elements to conduct their computational tasks asynchronously without needing a central clock signal to drive their operations.

The implementation of the proposed circuit elements on the cellular automaton requires five transition rules, which is one rule more than the implementation of the RE [12]. This indicates that the circuit elements may be slightly more complex in functionality than the RE, even though they require less input and output lines. A reason for the greater complexity could be the symmetry of the RE, as opposed of the lack thereof of the proposed elements, as well as the fact that all functionality of the RE is concentrated in one module, as opposed to the two modules required in this paper. Still, the number of five rules required here lies closely to the four rules for the RE model, implying that both models are on par with each other. Other implementations—possibly physical—may lead to a different outcome, favoring the proposed elements over the RE.

We have seen that the circuits constructed from the proposed elements allow merely one particle to be present at a time. To realize circuits with multiple particles, we need to combine the elements with a so-called *Join* element [8]—an element with two input lines and one output line, which requires input particles to be present on both input lines in order to produce output, whereas a single input to the Join is just kept pending until a second input arrives. Further research is needed, however, to rigorously define such multiple-signal asynchronous reversible circuits, as pointed out in [10]. The implementation of the Join on an STCA is likely to require at least one more transition rule.

References

1. Bennett, C.: Logical reversibility of computation. IBM Journal of Research and Development 17(6), 525–532 (1973)
2. Vos, A.D., Raa, B., Storme, L.: Generating the group of reversible logic gates. Journal of Physics A: Mathematical and General 35(33), 7063–7078 (2002)
3. Frank, M., Vieri, C., Ammer, M., Love, N., Margolus, N., Knight Jr., T.F.: A scalable reversible computer in silicon. In: Calude, C.S., Casti, J., Dinneen, M.J. (eds.) Unconventional Models of Computation, pp. 183–200. Springer, Singapore (1998)
4. Fredkin, E., Toffoli, T.: Conservative logic. International Journal of Theoretical Physics 21(3-4), 219–253 (1982)
5. Margolus, N.: Physics-like models of computation. Physica D 10(1/2), 81–95 (1984)
6. Merkle, R.: Reversible electronic logic using switches. Nanotechnology 4(1), 21–40 (1993)
7. Hauck, S.: Asynchronous design methodologies: an overview. Proc. IEEE 83(1), 69–93 (1995)
8. Patra, P., Fussell, D.: Conservative delay-insensitive circuits. In: Workshop on Physics and Computation (PhysComp 1996) (1996)

9. Morita, K.: A simple universal logic element and cellular automata for reversible computing. In: Margenstern, M., Rogozhin, Y. (eds.) MCU 2001. LNCS, vol. 2055, pp. 102–113. Springer, Heidelberg (2001)
10. Lee, J., Peper, F., Adachi, S., Mashiko, S.: On reversible computation in asynchronous systems. In: Quantum Information and Complexity, pp. 296–320. World Scientific, Singapore (2004)
11. Tanaka, K.: Universal reversible logic elements with 3 inputs, 3 outputs, and 2 states. Master Thesis, Hiroshima University (in Japanese) (2003)
12. Lee, J., Peper, F., Adachi, S., Morita, K., Mashiko, S.: Reversible computation in asynchronous cellular automata. In: Calude, C.S., Dinneen, M.J., Peper, F. (eds.) UMC 2002. LNCS, vol. 2509, pp. 220–229. Springer, Heidelberg (2002)
13. Peper, F., Isokawa, T., Kouda, N., Matsui, N.: Self-timed cellular automata and their computational ability. Future Generation Computer Systems 18(7), 893–904 (2002)
14. Mealy, G.: A method for synthesizing sequential circuits. Bell System Technical Journal 34(5), 1045–1079 (1955)
15. Morita, K., Shirasaki, A., Gono, Y.: A 1-tape 2-symbol reversible Turing machine. Trans. IEICE Japan E-72, 223–228 (1989)