2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

An Implementation of Optimum-Time Firing Squad Synchronization Algorithm on 1-Bit Cellular Automaton

Jun Nishimura† and Hiroshi Umeo‡

†MegaChips System Solutions Inc. Yodogawa-ku, Miyahara, 4-1-6, Osaka, Japan
‡Univ. of Osaka Electro-Communication, Faculty of Information Science and Technology,
Neyagawa-shi, Hatsu-cho, 18-8, Osaka, Japan

Abstract—We propose a time-optimum $(2n-2)$-step firing squad synchronization algorithm for a new class of cellular automata, CA1-bit, whose inter-cell communication amount at one step is restricted to 1-bit. The number of internal states of each cell implemented is 78 and the total number of transition rules is 208.

1. Introduction

In the long history of the study of the CA, generally speaking, the number of internal states of each cell is *finite* and the local state transition rule is defined in a such way that the state of each cell depends on the previous states of itself and its neighboring cells. Thus, in the finite state description of the CA, the amount of communication bits exchanged at one step between neighboring cells is assumed to be O(1)-bit, however, such bit-information exchanged between inter-cell has been hidden behind the definition of *conventional* automata-theoretic finite state description.

In the present paper, we focus our attention to the communication bits exchanged between inter-cells and introduce a new class of cellular automata, CA1-bit, whose inter-cell communication amount at one step is restricted to 1-bit. We refer the model as 1-bit CA. The number of internal states of the CA1-bit is assumed to be *finite* in a usual sense. The next state of each cell is determined by the present state of itself and two binary 1-bit inputs from its left and right neighbor cells. Thus the 1-bit CA can be thought of as one of the most powerless and the simplest models in a variety of CAs. We study a classical firing squad synchronization problem that gives a finite-state protocol for synchronizing a large scale of cellular automata, in which it was originally proposed by J. Myhill to synchronize all parts of self-reproducing cellular automata [5]. The firing squad synchronization problem has been studied extensively for more than 40 years [1-11]. It is important and interesting to develop optimum-time algorithms on the most powerless and the simplest models in a variety of CAs. First, we introduce a cellular automaton with 1-bit inter-cell communication and define the firing squad synchronization problem on CA1-bit. Then, we give an optimum-time firing squad synchronization algorithm on CA1-bit. The algorithm is based on the classical synchronization scheme developed by Waksman [11], having O(1)-bit communication, and it will be implemented on a CA1-bit with 78 internal states and 208 transition rules.

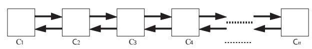

Figure 1: A one-dimensional cellular automaton having 1-bit inter-cell communication.

2. A Firing Squad Synchronization Problem on $\mathbf{CA}_{1-\mathrm{bit}}$

2.1. 1-Bit Communication Cellular Automaton

A one-dimensional 1-bit inter-cell communication cellular automaton consists of an infinite array of identical finite state automata, each located at positive integer point. See Fig. 1. A cell at point i is denoted by C_i, where $1 \leq i \leq n$. Each C_i, except C_1 and C_n, is connected with its left and right neighbor cells via a left or right one-way communication link, in which those communication links are indicated by right- and left-going arrows, as is shown in Fig. 1, respectively. Each one-way communication link can transmit only one bit at each step in each direction. The array operates in lock-step mode in such a way that the next state of each cell (except both end cells) is determined by both its own present state and the present binary inputs of its right and left neighbors. A more formal treatment can be found in Nishimura, Sogabe and Umeo [6, 7]. The CA1-bit is a special subclass of *normal* (i.e., *conventional*) cellular automata studied so far.

2.2. Firing Squad Synchronization Problem on $\mathbf{CA}_{1-\mathrm{bit}}$

The firing squad synchronization problem is formalized in terms of the model of $\mathrm{CA}_{1-\mathrm{bit}}$. All cells (*soldiers*), except the left and right end cells, are initially in the quiescent state at time $t = 0$ with the property that the next state of a quiescent cell with quiescent neighbors is the quiescent state again. At time $t = 0$ the *general* cell C_1 is in *fire-when-ready* state that is an initiation signal to the array. The firing squad synchronization problem [1, 5, 11] is stated as follows: Given an array of n identical cellular automata, including a *general* on the left end cell which is activated at time $t = 0$, we want to give the description (state set and next-state function) of the automata so that, *at some future time*, all the cells will *simultaneously* and, *for the first time*, enter a special *firing* state. The set of states and the next-state function must be independent of n. The tricky part of the problem is that the same kind of soldier with a fixed number of states is required to synchronize, regardless of the length n of the array.

3. Waksman's Optimum-Time Algorithm

3.1. Outline of Waksman's Algorithm

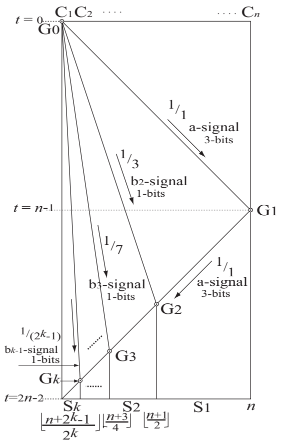

Figure 2: Time-space diagram for Waksman's synchronization algorithm.

Waksman's algorithm is constructed on the conventional O(1)-bit communication CA, and it can synchronize any cellular array consisting of n cells at exactly $2n - 2$ steps. Figure 2 shows its time-space diagram for the synchronization. At time $t = 0$, a general G_0, located at C_1, generates an a-signal and $k - 1$ b_k-signals, where $2 \leq k \leq \lceil \log_2 n \rceil - 1$. The a-signal propagates right at the sloop of $\frac{1}{1}$ (one cell per step). The b_k-signal propagates in the right direction at the sloop of $\frac{1}{2^k-1}$ (one cell per $2^k - 1$ step). The a-signal reaches C_n at $t = n-1$, and generates G_1 there and then it reflects and proceeds in the left direction at the same speed. The reflected signal meets b_2-signal, b_3-signal, …, b_k-signal, and generates $k-1$ generals $G_2, G_3, \ldots, G_k$ at each crossing point, respectively. Let S be any cellular space and $|S|$ be size of S. Let S_0 be the initial cellular space consisting of n cells and S_i be a cellular space betwen G_{i+1} and G_i. The general G_i is responsible for synchronizing S_i for any i such that $1 \leq i \leq k$.

3.2. Generation of Generals

G_i is generated at the position which divides exactly $|S_{i-2}|$. Therefore the parity of $|S_{i-2}|$ is an important factor for generating G_i. If the parity of $|S_{i-2}|$ is an odd number , G_i consists of a cell. And if the parity of $|S_{i-2}|$ is an even number, G_i consists of two cells. This parity information is determined by the cell on G_{i-1}, and it is transmitted to the cell of G_i by the a-signal. The signal can mark the general state G_i on the cell. Figure 2 is a time-space diagram for generating G_2 in case of that n is an even number. G_2 is on C_m, and consists of two cells. One of the cells includes the left part of the cellular array that is denoted C_1C_m, and another includes the right part of the cellular array that is denoted $C_{m+1}C_n$. In this case, G_2 of the right part is generated at $t = \frac{3n-4}{2}$, and G_2 of the left part is generated at $t = \frac{3n-2}{2}$. Therefore the right part has been synchronized earlier than the left part. In this case, Waksman' solution is that the right part starts to synchronize with delayed 1 step. The technique is referred to as delaying. In the case that the parity of $|S_{i-1}|$ is an even number, G_i uses the delaying. When the parity of $|S_{i-1}|$ is an odd number , G_i doesn't use it.

4. Time-Optimum Firing Squad Synchronization Algorithm on $CA_{1-\mathrm{bit}}$

We design a firing squad synchronization algorithm on $CA_{1\text{-}bit}$ based on Waksman's algorithm.

4.1. Generation of infinite signals

The key idea is the following construction of an infinite set of 1-bit signals which propagate at $\frac{1}{2^{k+1}-1}$-speed in one-way direction on a $CA_{1\text{-}bit}$.

[Lemma 1] There exists a $CA_{1\text{-}bit}$ that can generate an infinite set of signals which are used efficiently in Waksman's algorithm [11]. Precisely, for any $n \geq 2$, the initial left-end *General* G generates k signals $w_1, w_2, \ldots, w_k$ propagating at speed $1/(2^{k+1} - 1)$ on n cells, where $k = \lfloor \log_2(2n - 2) \rfloor - 1$.

We need two bits that constitute the parity of $|S_{i-2}|$ and $|S_{i-1}|$ so that we can generate G_i on right cell in real time. On the $CA_{1\text{-}bit}$, both two bits can't be carried on a single 1-bit a-signal. Thus we have to develop a new technique for the real-time generation of generals.

4.2. Parity of $|S_{i-2}|$

A a-signal, which propagates from G_i to G_{i+1}, defines a_{G_i}-signal. The a_{G_i}-signal communicates a 1-bit information that is the parity of $|S_{i-1}|$. This information is determined on G_i, so that a_{G_i}-signal on $CA_{1\text{-}bit}$ can't communicate it like one on CA. And this information isn't communicated by a_{G_i}-signal, but it is

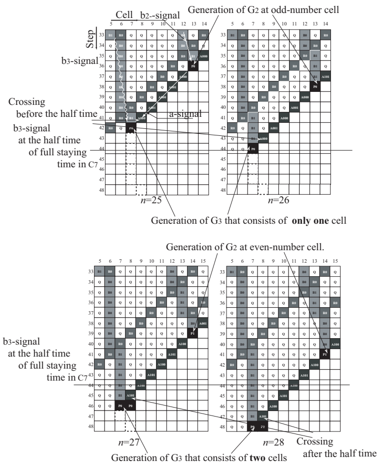

Figure 3: Generation of G_3.

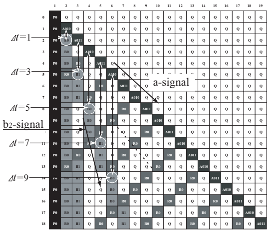

Figure 4: Relation of a-signal and b_2-signal

determined by the cell on which G_{i+1} is generated. Therefore a-signal can't include this information. we need to find an other technique.

Here, we have found a rule, when G_i is generated by the crossing of b_i-signal and a-signal. Figure 3 shows all cases in which G_3 is generated on C_7. In this figure, n means $|S_0|$. G_3 is generated by the crossing of b_3-signal and a-signal. We have investigated a parity of a cell on which G_2, which has been the previous gerenal, has been located. As a result, when G_2 has been located at odd-numbered cell, which had been counted from G_0, G_3 has always consisted of only one cell. And when G_2 has been located at even-numbered cell, which had been counted from G_0, G_3 has always consisted of two cells. And the parity of a cell, on which G_2 stays, is equal to the parity of $|S_1|$. If based on G_3, $|S_1|$ becomes $|S_{i-2}|$.

Next, we must investigate how to get the parity of $|S_{i-2}|$ at C_7. The b_3-signal stays on C_7 between 7 steps. In this figure, a horizontal line is pulled to a half time of b_3-signal full staying time. If the crossing happens before the half time, G_2 always positioned at odd-number cell. And if the crossing happens after the half time, G_2 always positioned at even-number cell.

The b_i-signal stays on C_{m_i} from $t = (2^i - 1)m_i - 2^i$ through $t = (2^i - 1)m_i - 2$. We define α_i, however $1 \leq \alpha_i \leq 2^i - 1$ as a variable that means the offset time. We can express the staying time of b_i-signal by using this equation.

$$t = (2^i - 1)m_i - 2^i - 1 + \alpha_i \tag{1}$$

And, a-signal, which crosses b_i-signal on C_{m_i}, arrives at C_{m_i+1} at time=t, where t is expressed by the following equation.

$$t = -m_i + 2n - 2 \tag{2}$$

From equations (1) and (2) we get:

$$2^i m_i = 2n + 2^i - 1 - \alpha_i, \quad 1 \leq \alpha_i \leq 2^i - 1. \tag{3}$$

m_i is always an integer that leads to the equation(3), because α_i must be an odd number. When α_i is an even number, b_i-signal doesn't cross the a-signal on C_{m_i}. In addition, we are also examined that the crossings of b_{i-1}-signal and a-signal on $C_{m_{i-1}}$. We can also express m_{i-1} by this equation such that:

$$2^{i-1} m_{i-1} = 2n + 2^{i-1} - 1 - \alpha_{i-1}, \quad 1 \leq \alpha_{i-1} \leq 2^{i-1} - 1 \tag{4}$$

Then we have equation(5).

$$2^i \cdot m_i - 2^{i-1} \cdot m_{i-1} = 2^i - 2^{i-1} - \alpha_i + \alpha_{i-1} \tag{5}$$

In this case, α_i and α_{i-1} are expressed as follows.
(i)If $C_{m_{i-1}}$ is located on odd-number cell,

$$\alpha_i = \alpha_{i-1} \tag{6}$$

(ii)If $C_{m_{i-1}}$ is located on even-number cell,

$$\alpha_i = \alpha_{i-1} + 2^{i-1} \tag{7}$$

Therefore we get equations (8) and (9).
When m_{i-1} is an odd number,

$$m_i = \frac{m_{i-1} + 1}{2} \tag{8}$$

And when m_{i-1} is an even number,

$$m_i = \frac{m_{i-1}}{2} \tag{9}$$

The b_i-signal on CA1-bit needs a trigger signal that will arrive after a half time of b_i-signal full staying time. Thus we have:

[Lemma 2] There exist a 1-bit signal that can determine the parity of $|S_{i-2}|$ in time.

4.3. Parity of $|\mathrm{S}_{i-1}|$

The $\mathrm{a}_{\mathrm{G}_i}$-signal counts the parity of $|\mathrm{S}_i|$ from G_i to G_{i+1}. The $\mathrm{a}_{\mathrm{G}_i}$-signal on CA1-bit can't count it. And this information is also determined by the cell on which G_{i+1} is generated. Therefore a-signal can't also include this information. we need to find an other technique. G_i stays on the center of S_{i-2}. Here we define $\mathrm{C}_1\mathrm{C}_{\mathrm{G}_i}$, which is the left part of S_{i-2}. Because $\mathrm{C}_1\mathrm{C}_{\mathrm{G}_i}$ is the same size as S_{i-1}, we can get the parity of $|\mathrm{S}_{i-1}|$ at $\mathrm{C}_{\mathrm{G}_i}$, when G_i is generated. Figure 4 shows a propagation of the a-signal and the b_i-signal in Waksman's algorithm. In this figure, we focus an offset time after a-signal passes until b_2-signal arrives. On any C_m, where m is a positive integer and it is bigger than 1, the offset time can lead to this equation.

$$\Delta t = 2p - 3 \tag{10}$$

In this case, we define Δt that is the remainder which divided the offset time by 4, If $p = 2x$,

$$\Delta t \;\; mod \;\; 4 = 4x - 3 = 4(x-1) + 1 \tag{11}$$

If $p = 2x + 1$,

$$\Delta t \;\; mod \;\; 4 = 4x - 1 = 4(x-1) + 3 \tag{12}$$

[**Lemma 3**] There exist 1-bit signals that can determine the parity of $|\mathrm{S}_{i-1}|$.

Based on lemmas above, our main theorem is stated as follows:

[**Theorem 4**] There exists a CA1-bit which can synchronize any n cells in $2n-2$ steps. A CA1-bit implemented has 78 internal states and 208 transition rules.

In Fig. 5 we show snapshots of the synchronization processes. Small right and left black triangles, shown ► and ◄ in the figure, indicate a 1-bit transfer in the right or left direction between neighboring cells. A symbol in a cell shows its internal state.

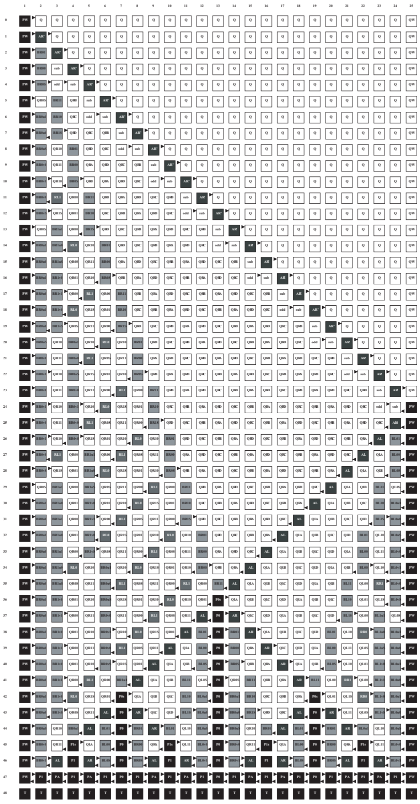

Figure 5: Snapshots of our time-optimum synchronization on $\mathrm{CA}_{\text{1-bit}}$ consisting of 25 cells.

5. Conclusion

We have designed and implemented an optimum-time firing squad synchronization algorithm on CA1-bit. Each cell has 78 internal states and 208 transition rules, and we checked its validity from $n = 2$through $n = 10000$ by computer simulation.

References

[1] R. Balzer: An 8-state minimal time solution to the firing squad synchronization problem. *Information and Control*, vol. 10 (1967), pp. 22-42.

[2] J. Mazoyer: A minimal time solution to the firing squad synchronization problem with only one bit of information exchanged. *Technical report of Ecole Normale Superieure de Lyon*, no. 89-03, April, (1989), p.51.

[3] J. Mazoyer: On optimal solutions to the firing squad synchronization problem. *Theoretical Computer Science*, vol. 168 (1996), pp. 367-404.

[4] K. Michisaka, H. Yahara, N. Kamikawa and H. Umeo: A generalization of 1-bit-communication firing squad synchronization algorithm. *Proc. of The 15th Annual Conference of Japanese Society for Artificial Intelligence*, 2C3-06, (2001), pp.1-4.

[5] E. F. Moore: The firing squad synchronization problem. in *Sequential Machines, Selected Papers* (E. F. Moore, ed.), Addison-Wesley, Reading MA., (1964), pp. 213-214.

[6] J. Nishimura, T. Sogabe and H. Umeo: A design of optimum-time firing squad synchronization algorithm on 1-bit cellular automaton. *Technical Report of IPSJ*, vol. 32-12 (2000), pp. 41-44.

[7] J. Nishimura, T. Sogabe and H. Umeo: A Realization of Optimum-Time Firing Squad Synchronization Algorithm on 1-Bit Cellular Automaton. *Technical Report of IPSJ*, vol. 87-8 (2002), pp. 59-66.

[8] H. Umeo, J. Nishimura and T. Sogabe: 1-bit inter-cell communication cellular algorithms (*invited lecture*). *Proc. of the Tenth Intern. Colloquium on Differential Equations*, held in Plovdiv in 1999, *International Journal of Differential Equations and Applications*, vol. 1A, no. 4 (2000), pp. 433-446.

[9] H. Umeo: Cellular Algorithms with 1-Bit Inter-Cell Communications. *Proc. of MFCS'98 Satellite Workshop on Cellular Automata*(Eds. T. Worsch and R. Vollmar), Interner Bericht 19/98, University of Karlsruhe, (1998), pp.93-104.

[10] H. Umeo, T. Sogabe and Y. Nomura: Correction, optimization and verification of transition rule set for Waksman's firing squad synchronization algorithm. *Proc. of the Fourth Intern. Conference on Cellular Automata for Research and Industry*, Springer, (2000), pp. 152-160.

[11] A. Waksman: An optimum solution to the firing squad synchronization problem. *Information and Control*, vol. 9 (1966), pp. 66-78.