A Synchronization Problem on 1-Bit Communication Cellular Automata

Hiroshi Umeo, Koshi Michisaka, and Naoki Kamikawa

Osaka Electro-Communication University,
Neyagawa-shi, Hatsu-cho 18-8, Osaka, 572-8530, Japan
umeo@umeolab.osakac.ac.jp

Abstract. We study a classical firing squad synchronization problem for a large scale of one- and two-dimensional cellular automata having 1-bit inter-cell communications ($\mathrm{CA}_{1-\mathrm{bit}}$). First, it is shown that there exists a one-dimensional $\mathrm{CA}_{1-\mathrm{bit}}$ that can synchronize n cells with the general on the kth cell in $n + \max(k, n - k + 1)$ steps, where the performance is two steps larger than the optimum one that was developed for O(1)-bit communication model. Next, we give a two-dimensional $\mathrm{CA}_{1-\mathrm{bit}}$ which can synchronize any $n \times n$ square and $m \times n$ rectangular arrays in $2n - 1$ and $m + n + \max(m, n)$ steps, respectively. Lastly, we propose a generalized synchronization algorithm that operates in $m + n + max(r + s, m + n - r - s) + O(1)$ steps on two-dimensional $m \times n$ rectangular arrays with the general located at an arbitrary position (r, s) of the array, where $1 \leq r \leq m$ and $1 \leq s \leq n$. The time complexities for the first three algorithms developed are one to four steps larger than optimum ones proposed for O(1)-bit communication models. We show that there still exist several new interesting synchronization algorithms on $\mathrm{CA}_{1-\mathrm{bit}}$ although more than 40 years have passed since the development of the problem.

1 Introduction

In recent years cellular automata (CA) have been establishing increasing interests in the study of modeling real phenomena occurring in biology, chemistry, ecology, economy, geology, mechanical engineering, medicine, physics, sociology, public traffic, etc. Cellular automata are considered to be a good model of complex systems in which an infinite one-dimensional array of finite state machines (cells) updates itself in synchronous manner according to a uniform local rule.

In this paper, we study a famous firing squad synchronization problem on a newly introduced 1-bit CA model for which solution gives a finite-state protocol for synchronizing a large scale of cellular automata. The synchronization for cellular automata has been known as the firing squad synchronization problem since its development, where it was originally proposed by J. Myhill to synchronize all parts of self-reproducing cellular automata [9]. The firing squad synchronization problem has been studied extensively in more than 40 years [1-19].

An O(1)-bit communication model is a conventional CA where the amount of communication bits exchanged at one step between neighboring cells is assumed

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 492–500, 2003.

to be O(1)-bit, however, such bit-information exchanged between inter-cells has been hidden behind the definition of conventional automata-theoretic finite state descriptions. On the other hand, a 1-bit inter-cell communication model studied in this paper is a new CA whose inter-cell communication is restricted to 1-bit. We call the model 1-bit CA in short. The number of internal states of the 1-bit CA is assumed to be finite in a usual way. 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 to be one of the most powerless and simplest models in a variety of CAs.

In the next section 2, we define a (generalized) firing squad synchronization problem on the cellular automata whose inter-cell communication is restricted to 1-bit. In section 3, we propose a new generalized synchronization algorithm that operates in $n+\max(k, n-k+1)$ steps for firing n cells on 1-D $CA_{1-\mathrm{bit}}$, where the general is located on the kth cell from the left end. The algorithm is a generalized extension of Mazoyer [6] and Nishimura, Sogabe and Umeo [11]. In section 4, three 1-bit implementations of synchronization algorithms for two-dimensional square and rectangular arrays will be given. Due to the space available, we omit the details of the proofs of theorems given below.

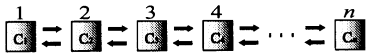

Fig. 1. One-dimensional cellular automaton having 1-bit inter-cell communication links.

2 Firing Squad Synchronization Problem on $CA_{1-\mathrm{bit}}$

The firing squad synchronization problem is formalized in terms of the model of cellular automata. Fig. 1 shows a finite one-dimensional (1-D) cellular array consisting of n cells. Each cell is an identical (except the end cells) finite state automaton.

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. Let k be any integer such that $1 \leq k \leq n$. All cells (*soldiers*), except the kth cell C_k from the left end, 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_k is in *fire-when-ready* state that is an initiation signal to the array. The *generalized* firing squad synchronization problem [7, 13, 18] is stated as follows:

Given an array of n identical cellular automata, including a *general* on the kth 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 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 A Generalized Synchronization Algorithm on 1-D Arrays

Nishimura, Sogabe and Umeo[11] designed an optimum-step firing squad synchronization algorithm on $\mathrm{CA}_{1-\mathrm{bit}}$, where $2n-2$ steps are required for synchronizing n cells on 1-D array and the general is located at the left end of the array. The algorithm, that is referred to as NSU algorithm, is stated as follows:
[Theorem 1][11] There exists a $\mathrm{CA}_{1-\mathrm{bit}}$ which can synchronize n cells with the general on the left end in $2n-2$ steps. The $\mathrm{CA}_{1-\mathrm{bit}}$ constructed has 78 internal states and 208 transition rules.

The generalized synchronization algorithm that we are going to design is based on the NSU algorithm. In our construction additional two steps are required for transmitting a signal to the nearest end, where the signal has been kept for $\min(2k-2,\ 2n-2k-2)$ steps by the general cell.

In Fig. 2, we show snapshots of the generalized firing synchronization algorithm on 24 cells with a general on C_8. Small right and left black triangles, ▶ and ◀, shown in the figure, indicate a 1-bit signal transfer in the right or left direction between neighbor cells. A symbol in a cell shows its internal state. The total number of internal states and transition rules of the $\mathrm{CA}_{1-\mathrm{bit}}$ realized on a computer is 282 and 721, respectively. We checked the validity of the rule set for arrays of length $n=2$ to 100 at any position of the general. Thus we have:
[Theorem 2] There exists a $\mathrm{CA}_{1-\mathrm{bit}}$ which can synchronize n cells in $n+\max(k, n-k+1)$ steps, where k is any integer such that $1 \le k \le n$ and a general is located on the kth cell from the left end of the array.

4 Synchronization Algorithms on 2-D Arrays

In this section we develop some synchronization algorithms for 2-D 1-bit inter-cell communication CA models. Fig. 3 shows a finite two-dimensional cellular array consisting of $m \times n$ cells. A cell on (i, j) is denoted by $\mathrm{C}_{i,j}$. Each cell is an identical (except the border cells) finite state automaton.

The array operates in lock-step mode in such a way that the next state of each cell (except border cells) is determined by both its own present state and the present binary inputs from its north, south, east and west neighbors. All cells, except the general cell, are initially in the quiescent state with the property that the next state of a quiescent cell with quiescent neighbors is the quiescent state again. Several 2-D synchronization algorithms and their implementations have been presented in Beyer [2], Grasselli [4], Shinar [12], Szwerinski [13] and Torre, Napoli and Parente [14] for O(1)-bit communication models.

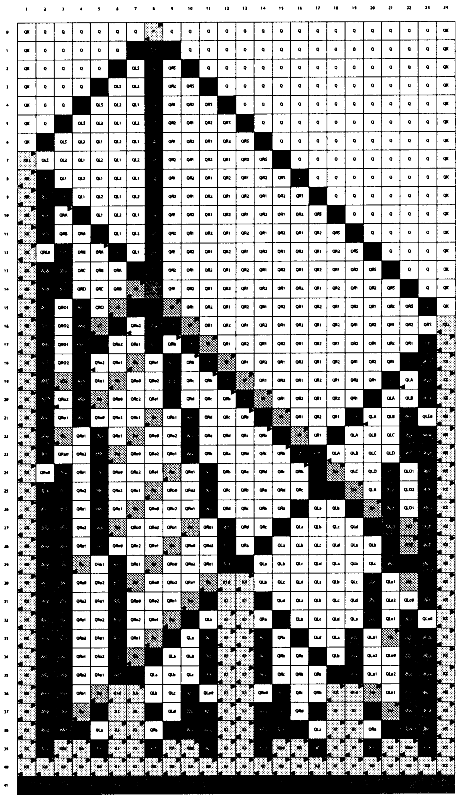

Fig. 2. Snapshots of the generalized 1-bit firing squad synchronization algorithm operating on 24 cells with a general on C_8.

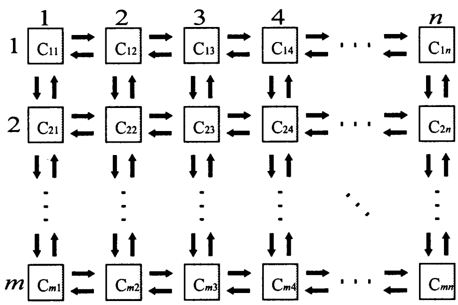

Fig. 3. Two-dimensional cellular automaton.

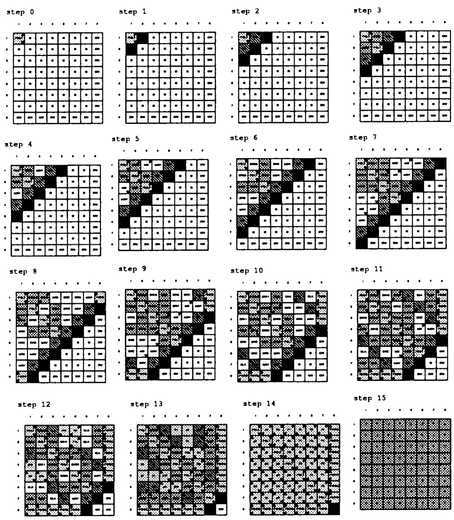

Fig. 4. Snapshots of our $(2n-1)$-step square firing squad synchronization algorithm with the general on the north west corner.

4.1 Synchronization Algorithm on Square Arrays

We present a new synchronization algorithm that runs in $(2n-1)$ steps on $n \times n$ square arrays. Our algorithm is one step slower than that of Shinahr [13] for O(1)-bit communication model and operates as follows. By dividing the entire square array into n L-shaped 1-D arrays such that the length of the ith L is $2n - 2i + 1$ $(1 \le i \le n)$, we treat the square firing as n independent 1-D firings with the general located at the center cell. On the ith L, a general is generated at $C_{i,i}$ at time $t = 2i - 1$, and the general initiates the horizontal and vertical firings on the row and column arrays. In our construction, we apply the previous NSU algorithm [12] for each row and column firing. The array fires in optimum time $t = 2i - 1 + 2(n - i + 1) - 2 = 2n - 1$.

We have tested our transition rule set on squares of size 2×2 to 1000×1000. The total number of internal states and transition rules of the $\mathrm{CA}_{1-\mathrm{bit}}$ realized on a computer is 127 and 405, respectively. Figure 4 shows snapshots of configurations of our 127-state synchronization algorithm running on a square of size 8×8. Thus we have:

[Theorem 3] There exists a 2-D $\mathrm{CA}_{1-\mathrm{bit}}$ which can synchronize $n \times n$ cells in $2n - 1$ steps.

4.2 Synchronization Algorithm on Rectangular Arrays

The generalized firing squad synchronization algorithm presented in [Theorem 2] can be applied to the problem of synchronizing rectangular arrays with the

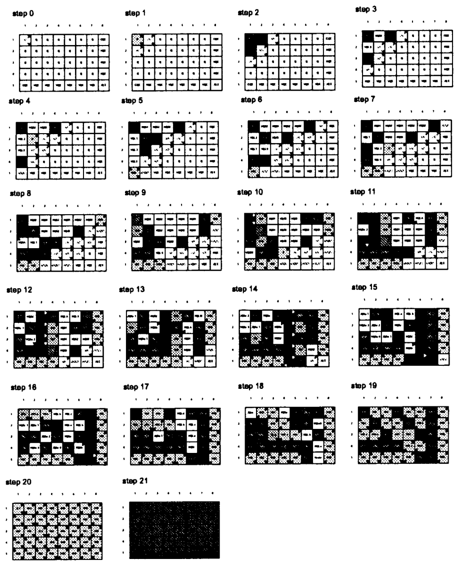

Fig. 5. Snapshots of our rectangular firing squad synchronization algorithm with the general at the north-west corner.

general at the north-west corner. The configuration of the generalized firing on 1-D arrays can be mapped on 2-D array.

The rectangular array is regarded as $\min(m, n)$ L-shaped 1-D arrays, where they are synchronized independently using the generalized firing squad synchronization algorithm. We have implemented the algorithm on a computer. In Fig. 5, we show snapshots of the synchronization process on 5×8 rectangular array. The total number of internal states and transition rules of the $\mathrm{CA}_{1-\mathrm{bit}}$ realized on a computer are 862 and 2217, respectively. Thus we have:

[Theorem 4] There exists a 2-D $\mathrm{CA}_{1-\mathrm{bit}}$ which can synchronize $m \times n$ rectangular arrays in $m + n + \max(m, n)$ steps.

4.3 Generalized Synchronization Algorithm on 2-D Rectangular Arrays

In this subsection, we study the generalized synchronization algorithm on rectangular arrays. Let r, s be any integer such that $1 \le r \le m$, $1 \le s \le n$. At time $t = 0$ the general cell $\mathrm{C}_{r,s}$ is in *fire-when-ready* state that is an initiation signal to the array. Before presenting the 1-bit algorithm, we show a simple and efficient mapping scheme developed for O(1)-bit CA model that embeds any generalized one-dimensional synchronization algorithms onto two-dimensional arrays [16].

Now we consider a 2-D array of size $m \times n$. We divide mn cells into $m+n-1$ groups g_k, $1 \le k \le m+n-1$, defined as follows;

$$g_k = \{C_{i,j} | (i-1) + (j-1) = k-1\}.$$

That is,
$g_1 = \{C_{1,1}\}$, $g_2 = \{C_{1,2}, C_{2,1}\}$, $g_3 = \{C_{1,3}, C_{2,2}, C_{3,1}\}$, . . . , $g_{m+n-1} = \{C_{m,n}\}$.

Let M be any one-dimensional $\mathrm{CA}_{1-\mathrm{bit}}$ that fires ℓ cells in $T(\ell, k)$ steps, where the general is on C_k. We assume that M has $m+n-1$ cells. We consider the one-to-one correspondence between the ith group g_i and the ith cell C_i on M such that $g_i \leftrightarrow C_i$, where $1 \leq i \leq m+n-1$. We can construct a 2-D $\mathrm{CA}_{1-\mathrm{bit}}$ N so that all cells in g_i simulates the ith cell C_i in real-time and N can fire any $m \times n$ arrays with the general $\mathrm{C}_{r,s}$ at time $t = T(m+n-1, r+s-1)$ if and only if M fires 1-D arrays of length $m+n-1$ with the general on C_{r+s-1} at time $t = T(m+n-1, r+s-1)$.

Based on the generalized 1-D algorithm given in [Theorem 2], we get the following 2-D generalized synchronization algorithm that fires in $T(m, n, r, s)$ steps given below. The total number of internal states and transition rules of the $\mathrm{CA}_{1-\mathrm{bit}}$ realized on a computer is 300 and 2333, respectively. In Fig. 6 we show snapshots of the 300-state generalized synchronization algorithm running on rectangular array of size 5×8 with the general on $\mathrm{C}_{3,4}$. Thus we have:

[Theorem 5] There exists a 2-D 1-bit communication $\mathrm{CA}_{1-\mathrm{bit}}$ that can synchronize any $m \times n$ rectangular arrays in $T(m, n, r, s)$ steps, where (r, s) is an arbitrary initial position of the general and $\mathrm{T}(m, n, r, s)$ is defined as follows:

$$T(m,n,r,s) = \begin{cases} m+n-2+\max(r+s, m+n-r-s+2) & \text{if } s=1, r=1 \text{ or } s=n, r=m \text{ or } 2 \leq r \leq n-1, 2 \leq s \leq m-1 \\ m+n+\max(r+s, m+n-r-s) & \text{if } 2 \leq s \leq n-1, r=1 \text{ or } s=1, 2 \leq r \leq m-1 \\ m+n-2+\max(r+s, m+n-r-s+4) & \text{if } 2 \leq s \leq n-1, r=m \text{ or } s=n, 2 \leq r \leq m-1 \\ m+n+\max(r+s, m+n-r-s+2) & \text{if } s=n, r=1 \text{ or } s=1, r=m \end{cases}$$

Szwerinski [14] proposed an optimum-time generalized 2-D firing algorithm with 25600 internal states that fires any $m \times n$ array in $m+n+\max(m, n)-\min(r, m-r+1)-\min(s, n-s+1)-1$ steps. Our 2-D generalized synchronization algorithm is relatively larger than the optimum one proposed by Szwerinski [14], however, the number of internal states required for the firing is the smallest known at present.

5 Conclusion

We have proposed several new generalized synchronization algorithms for one- and two-dimensional cellular arrays having 1-bit inter-cell communication and implemented them on a computer. Most of the algorithms proposed are one to four steps larger than optimum ones proposed for O(1)-bit communication model. We are convinced that there still exist interesting new synchronization algorithms, although more than 40 years have passed since the development of the problem.

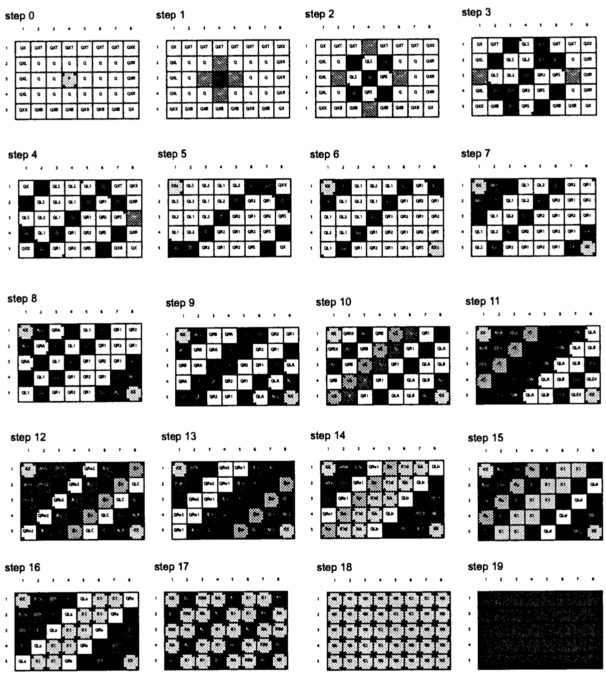

Fig. 6. Snapshots of our generalized rectangular firing squad synchronization algorithm operating for an array of size 5×8 with the general on $C_{3,4}$.

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. W. T. Beyer: Recognition of topological invariants by iterative arrays. Ph.D. Thesis, MIT, (1969), pp. 144.
3. E. Goto: A minimal time solution of the firing squad problem. Dittoed course notes for Applied Mathematics 298, Harvard University, (1962), pp. 52-59.
4. A. Grasselli: Synchronization of cellular arrays: The firing squad problem in two dimensions. *Information and Control*, vol. 28(1975), pp. 113-124.
5. J. Mazoyer: A six-state minimal time solution to the firing squad synchronization problem. *Theoretical Computer Science*, vol. 50(1987), pp. 183-238.
6. J. Mazoyer: On optimal solutions to the firing squad synchronization problem. *Theoretical Computer Science*, vol. 168(1996), pp. 367-404.

7. 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.
8. M. Minsky: *Computation: Finite and infinite machines.* Prentice Hall, (1967), pp. 28-29.
9. 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.
10. F. R. Moore and G. G. Langdon: A generalized firing squad problem. *Information and Control*, vol. 12(1968), pp. 212-220.
11. J. Nishimura, T. Sogabe and H. Umeo: A Design of Optimum-Time Firing Squad Synchronization Algorithm on 1-Bit Cellular Automaton. Tech. Rep. of IPSJ, (2000).
12. I. Shinahr: Two- and three-dimensional firing squad synchronization problems. *Information and Control*, vol. 24(1974), pp. 163-180.
13. H. Szwerinski: Time-optimum solution of the firing-squad-synchronization-problem for n-dimensional rectangles with the general at an arbitrary position. *Theoretical Computer Science*, vol. 19(1982), pp. 305-320.
14. S. L. Torre, M. Napoli and M. Parente: A compositional approach to synchronize two dimensional networks of processors. *Theoretical Informatics and Applications*, 34 (2000) pp.549-564.
15. H. Umeo: Linear-time recognition of connectivity of binary images on 1-bit inter-cell communication cellular automaton. *Parallel Computing*, 27 (2001), pp.587-599.
16. H. Umeo, M. Maeda and N. Fujiwara: An efficient mapping scheme for embedding any one-dimensional firing squad synchronization algorithm onto two-dimensional arrays. *Proc. of the 5th International Conference on Cellular Automata for Research and Industry*, LNCS 2493, Springer-Verlag, pp.69-81(2000).
17. V. I. Varshavsky: Synchronization of a collection of automata with random pairwise interaction. *Autom. and Remote Control*, vol. 29(1969), pp. 224-228.
18. V. I. Varshavsky, V. B. Marakhovsky and V. A. Peschansky: Synchronization of interacting automata. *Mathematical Systems Theory*, vol. 4, no. 3(1970), pp. 212-230.
19. A. Waksman: An optimum solution to the firing squad synchronization problem. *Information and Control*, vol. 9(1966), pp. 66-78.