# Global Synchronizations for Two-Dimensional Cellular Arrays with Local Communications

Hiroshi UMEO[1]

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

**Abstract.** The synchronization in cellular automata has been known as a firing squad synchronization problem (FSSP) since its development. The firing squad synchronization problem on cellular automata has been studied extensively for more than fifty years, and a rich variety of synchronization algorithms has been proposed not only for one-dimensional (1D) but two-dimensional (2D) arrays. In the present paper, we focus our attention to the 2D array synchronizers that can synchronize any square/rectangle arrays and construct a survey on recent developments in their designs and implementations of optimum-time and non-optimum-time synchronization algorithms for 2D arrays. The algorithms proposed can be extended to any multi-dimensional arrays without any loss of communication time complexity.

## 1 Introduction

Synchronization of large scale networks is an important and fundamental computing primitive in parallel and distributed systems. We study a synchronization problem that gives a finite-state protocol for synchronizing cellular automata. The synchronization in cellular automata has been known as a firing squad synchronization problem (FSSP) since its development, in which it was originally proposed by J. Myhill in Moore [1964] to synchronize all parts of self-reproducing cellular automata. The problem has been studied extensively for more than fifty years. See Umeo [2009] for details.
In the present paper, we focus our attention to 2D array synchronizers that can synchronize any square/rectangle arrays and construct a survey on recent developments in their designs and implementations of optimum-time and non-optimum-time synchronization algorithms for 2D arrays. Specifically, we attempt to consider the following questions:

- Is there any new 2D FSSP algorithm other than classical ones?
- What is the smallest implementation of the 2D synchronizer?
- How can we synchronize 2D arrays with the general at any position?
- How do the algorithms compare with each other?
- Can we extend those 2D synchronizers to three-dimensional arrays?

We introduce a rich variety of mapping schemes for the design of 2D array synchronizers. Any configurations of a 1D synchronization algorithm can be mapped efficiently onto a 2D array so that all of the cells on the 2D array would fall into a final synchronization state simultaneously. The mapping schemes we consider include a rotated L-shaped mapping, a zebra mapping, a diagonal mapping, and a one-sided recursive-halving marking based mapping.

## 2 Firing Squad Synchronization Problem and Algorithms

Consider a finite 2D rectangular array consisting of $m \times n$ cells arranged in $m$ rows and $n$ columns. 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 states of its north, south, east and west neighbors. Thus, we assume the von Neumann-type four nearest neighbors. The firing squad synchronization problem is to determine a finite-state local description (state set and next-state function) for cells that ensures all cells enter the *fire* state at exactly the same time and for the first time, thus realizing a global synchronization.

**Table 1.** A list of FSSP algorithms for square arrays.

| Algorithms & Implementations | # of states | # of rules | Time complexity | Communication model | Mapping |
|---|---|---|---|---|---|
| Beyer [1969] Algorithm $A_1$ | — | — | $2n-2$ | O(1)-bit | L-shaped |
| Shinahr [1974] Algorithm $A_1$ | 17 | — | $2n-2$ | O(1)-bit | L-shaped |
| Umeo, Maeda and Fujiwara [2002] Algorithm $A_1$ | 9 | 1718 | $2n-2$ | O(1)-bit | L-shaped |
| Umeo and Kubo [2010] Algorithm $A_2$ | 7 | 787 | $2n-2$ | O(1)-bit | Zebra |
| Umeo, Maeda, Hisaoka, and Teraoka [2006] Algorithm $A_3$ | 6 | 942 | $4n-4$ | O(1)-bit | Diagonal |
| Ishii et al. [2006] Algorithm $A_3$ | 15 | 1614 | $2n-2$ | O(1)-bit | Diagonal |
| Umeo, Uchino and Nomura [2011] Algorithm $A_4$ | 37 | 3271 | $2n-2$ | O(1)-bit | Recursive-Halving |
| Gruska, Torre and Parente [2007] Algorithm $A_1$ | — | — | $2n-2$ | 1-bit | L-shaped |
| Umeo and Yanagihara [2011] Algorithm $A_1$ | 49 | 237 | $2n-2$ | 1-bit | L-shaped |

In Tables 1 and 2 we present a list of implementations of square/rectangle FSSP algorithms for cellular automata with O(1)-bit or 1-bit communications. The O(1)-bit communication model is a usual cellular automaton in which the amount of communication bits exchanged in one step between neighboring cells is assumed to be O(1) bits. The 1-bit communication model is a subclass of the O(1)-bit model, in which inter-cell communication is restricted to 1-bit communication.

**Table 2.** A list of FSSP algorithms for rectangle arrays.

| Algorithms & Implementations | # of states | # of rules | Time complexity | Communication model | Mapping |
|---|---|---|---|---|---|
| Beyer [1969] Algorithm $A_1$ | — | — | $m+n+max(m,n)-3$ | O(1)-bit | L-shaped |
| Shinahr [1974] Umeo et al. [2009] Algorithm $A_1$ | 28 28 | — 12849* | $m+n+max(m,n)-3$ | O(1)-bit | L-shaped |
| Umeo and Nomura [2010] Algorithm $A_2$ | 10 | 1629 | $m+n+max(m,n)-2$ | O(1)-bit | Zebra |
| Umeo, Hisaoka and Akiguchi [2005] Algorithm $A_3$ | 12 | 1532 | $m+n+max(m,n)-3$ | O(1)-bit | Diagonal |
| Umeo, Maeda, Hisaoka, and Teraoka [2006] Algorithm $A_3$ | 6 | 942 | $2m+2n-4$ | O(1)-bit | Diagonal |
| Umeo, Nishide and Yamawaki [2011] Algorithm $A_4$ | 384 | 112690 | $m+n+max(m,n)-3$ | O(1)-bit | Recursive-Halving |

The readers can see how a rich variety of 2D FSSP algorithms exists. Some algorithms can be easily extended to 3D arrays. The embedding schemes developed in this paper would be useful for further implementations of multi-dimensional synchronization algorithms.

## References

1. 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.
2. H. Umeo: Firing squad synchronization problem in cellular automata. In *Encyclopedia of Complexity and System Science*, R. A. Meyers (Ed.), Springer, Vol.4(2009), pp.3537-3574.
3. H. Umeo, M. Hisaoka, and S. Akiguchi: Twelve-state optimum-time synchronization algorithm for two-dimensional rectangular cellular arrays. *Proc. of 4th International Conference on Unconventional Computing: UC 2005, LNCS 3699*, (2005), pp.214-223.
4. Umeo and Kubo: A seven-state time-optimum square synchronizer. *Proc. of the 9th International Conference on Cellular Automata for Research and Industry*, LNCS 6350, Springer-Verlag, pp.219-230(2010).
5. H. Umeo, K. Nishide, and K. Kubo: A simple optimum-time FSSP algorithm for multi-dimensional cellular automata. *Proc. of Automata 2012 and JAC 2012*, EPTCS, pp. 66-80, (2012).