Short Note

A Time-Optimum Systolic Simulation of One-Way Cellular Automata

HIROSHI UMEO*

It is shown that, for any one-way kn time-bounded cellular automaton M, there exists a systolic array which can simulate M in $kn+n+O(1)$ optimum steps, where k is any positive integer.

1. Introduction

There has been increasing interest in the study of systolic systems which overlap I/O operations and computations. In the design of systolic algorithms, speeding up the I/O operations, without sacrificing the total throughput, is an important problem. In [1] we have developed a time-efficient conversion technique from cellular algorithms which separate computations from I/O operations into systolic ones.

In this paper we consider a similar technique for the 1-dimensional cellular automata(CA's) with restricted information-flow. It is shown that a more remarkable speed up is attained in the simulation of one-way CA's than in the case of two-way CA's.

A 1-dimensional one-way CA, M, consists of an array of finite state automata, called cells $C_i(1 \leqq i \leqq n)$, which are uniformly interconnected. See Fig. 1. M is a pair $M=(Q, \delta)$, where Q is the set of cell states and δ: $Q^2 \rightarrow Q$ is the one-way local transition function. We denote the state of C_i at time t by s_i^t. At time $t=0$, CA receives a spatial input in a way such that $s_i^0=a_i(1 \leqq i \leqq n)$. A step of computation of M consists of a state transformation of each cell, that is, the simultaneous applications of δ at all cells in such a way that $s_i^{t+1}=\delta(s_i^t, s_{i+1}^t)$. Thus the information-flow on M is restricted to one-way, that is, from right to left. The configuration $s_1^{T(n)}s_2^{T(n)} \ldots s_n^{T(n)}$ is considered as the output of $T(n)$ time-bounded CA, M, for an input $a_1, a_2, \ldots, a_n$. Following the convention in the cellular automata theory, we measure the time complexity for the CA's by parallel steps required only for the computations.

For systolic arrays(SA's), various models have been proposed [1], [4], [5]. We take an array configuration shown in Fig. 2 as the systolic model. Details of the

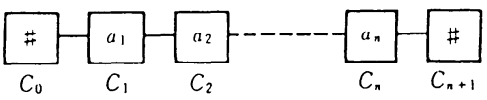

Fig. 1 Cellular automaton.

*Dept. of Applied Electronic Engineering, Faculty of Engineering, Osaka Electro-Communication Univ.

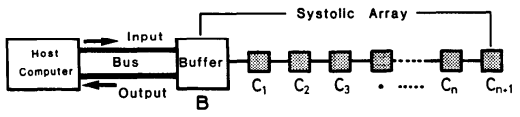

Fig. 2 Systolic array.

SA's are omitted. See [4]. Differing from the case of CA's, the time complexity of the SA is measured by the parallel steps from the beginning of the input operations to the end of the output operations.

2. Time-Optimum Systolic Simulation of One-Way CA's

In [1] we have gotten the following result for the two-way CA's.

[Theorem 1] For any two-way kn time-bounded CA, M, there exists an SA which can simulate M in $kn+3n+O(1)$ steps.

If the direction of the information-flow of the CA's is restricted to one-way, the following faster systolic simulation is possible.

[Theorem 2] For any one-way kn time-bounded CA, M, there exists an SA, A, which can simulate M in $kn+n+O(1)$ steps.

(Proof sketch) We construct an SA, A, which simulates M in $kn+n$ steps. First we prove the case $k \geqq 2$. The SA, A consists of n systolic cells, each contains four data registers R_1, R_2, R_3, and R_4. We refer to R_1 and R_2 in each cell as the first layer and R_3 and R_4 as the second layer. The initial data to A are loaded through the buffer B according to the order $a_n, a_{n-1}, \ldots, a_2, a_1$ at the rate of 1 state/1 step. The data movement on the array is as follows: A pair of states of M continue to advance in the right direction at a unit speed on the first layer, looking for empty R_3 and R_4 registers. When they are found, the data remains at R_3 and R_4 until an output signal is transmitted to that cell. A pair of data registers R_i and R_{i+1}, where $i=1$ and 3, simulate one cell of M.

See Fig. 3. When the data are moving or staying on A, they simulate a 1-step transition of M at every cycle. We will explain how A simulates M in the first n steps. Let t be any positive integer such that $1 \leqq t \leqq n-1$. At time t the configuration α:

$$\alpha: \underbrace{s_n^t s_n^{t-1}}_{C_1}\ \underbrace{s_{n-1}^{t-1} s_{n-1}^{t-2}}_{C_2}\ \cdots\ \underbrace{s_{n-t+2}^2 s_{n-t+2}^1}_{C_2}\ \underbrace{s_{n-t+1}^1 a_{n-t+1}}_{C_1}\ \underbrace{a_{n-t}}_{B}$$

is folded at its center and is stored on the 1st and 2nd layers of the first $t/2$ cells. At one step A is able to transform α into β:

$$\beta: \underbrace{s_n^{t+1} s_n^t}_{C_1}\ \underbrace{s_{n-1}^t s_{n-1}^{t-1}}_{C_2}\ \cdots$$

$$\cdots\ \underbrace{s_{n-t+2}^3 s_{n-t+2}^2}_{C_3}\ \underbrace{s_{n-t+1}^2 s_{n-t+1}^1}_{C_2}\ \underbrace{s_{n-t}^1 a_{n-t}}_{C_1}\ \underbrace{a_{n-t-1}}_{B},$$

since M is a one-way CA and the information necessary for the transformation can be found in either the right or the left neighbour cell. In time t ($\in \{t \mid n \leqq t \leqq kn\}$) the simulation is made similarly by generating configurations one by one. Note that at time $t=kn$ the second layer contains the following configuration:

$$\underbrace{s_n^{kn} s_n^{kn-1}}_{C_1}\ \underbrace{s_{n-1}^{kn-1} s_{n-1}^{kn-2}}_{C_2}\ \cdots\ \underbrace{s_1^{kn-n+1} s_1^{kn-n}}_{C_n}.$$

From time $t=1$ A begins to prepare the firing squad synchronization [6] which will fire at time $t=kn$[†]. The firing tells each cell to begin output operations. Each cell begins to shift its data on the second layer in the left direction at a unit speed. Additional n steps are required for the output operation. In Fig. 3 we illustrate the configurations of A in the case where $k=2$ and $n=5$.

In the case $k=1$, $n/2$ cells are sufficient for the simulation. At time $t=n$ the firing occurs and n data are output after n steps.

Thus A requires $kn+n+O(1)$[††] steps for the simulation of M.

Remark 1: Our simulation algorithm is time-optimum under the following conditions:

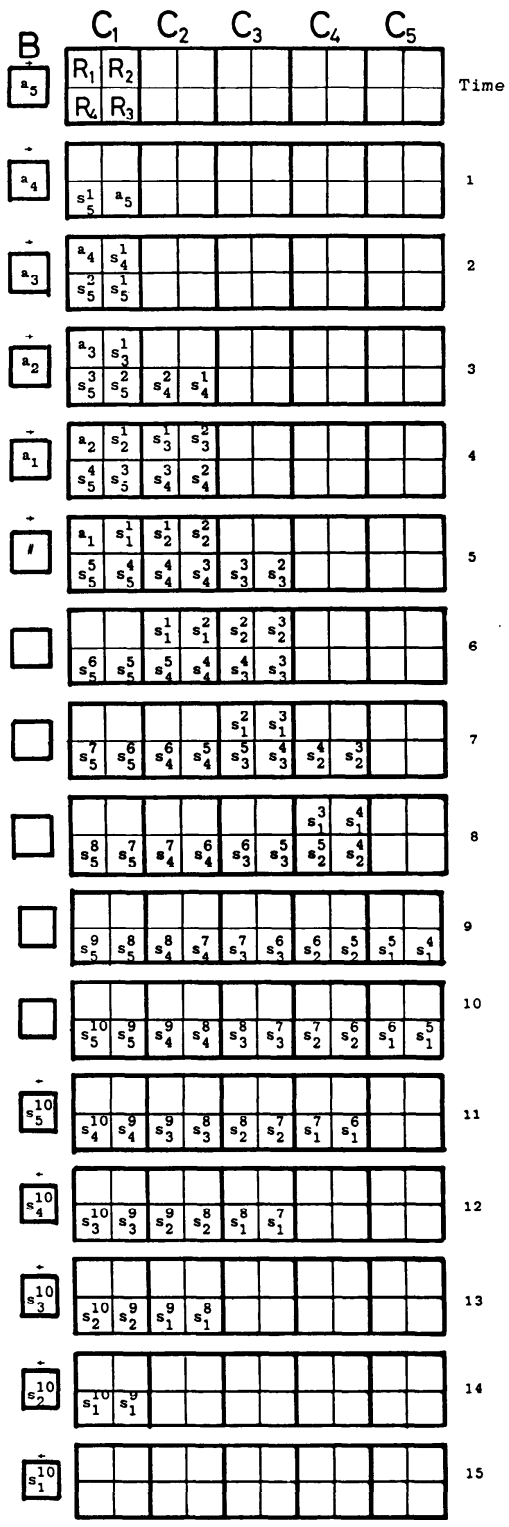

Fig. 3 Configurations of the systolic array A which simulates M.

[†]Generally, the optimum firing squad synchronization algorithm [6] fires n cells in $(2n-2)$-step later after the general issues an initial "ready-for-fire" signal. But the firing squad synchronization algorithm can be easily modified so that the firing occurs in kn-step later for any integer $k(\geqq 2)$.

In this paper we use the modified kn-step version in cases where $k \geqq 2$. On the basis of [6] it is designed as follows: In the case where $k=2k'$ for some fixed integer $k' \geqq 1$: The cellular automaton repeats the Waksman's synchronization scheme [6] k' times. We think of the k'-th firing as the genuine firing. In the case where $k=2k'-1$ for some constant $k' \geqq 2$: After generating a general at the right end (requiring n steps), we use $(2k'-2)n=2(k'-1)n$-step version given above.

[††]To give more generality we add a term O(1).

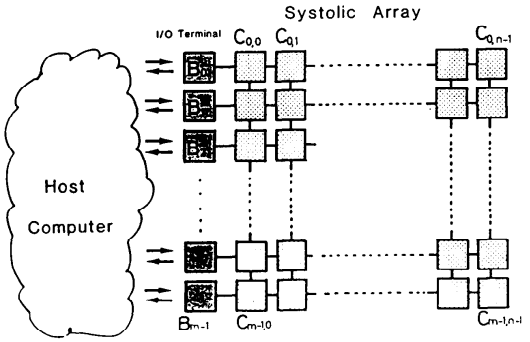

Fig. 4 a 2-D systolic array.

(1) The I/O operations are performed at the rate of 1 data/1 step, respectively.

(2) The compaction of the initial data is not allowed. Each data s_i^t $(0 \leqq t \leqq kn,\ 1 \leqq i \leqq n)$ must not stay on A for less than kn steps. So the time when the last data is output is $t=kn+n$. Thus our time complexity is optimum.

Remark 2: If we allow the compaction of the initial data, we can show that: Let ε be any positive constant. For any one-way kn time-bounded CA, M, there exists an SA, A, which can simulate M in $(3+\varepsilon)n$ steps. This simulation technique is not realistic, since the number of internal states of the systolic cell increases in exponential order.

Remark 3: In a hardware realization a one-step of A seems to be more complex than that of M, however, its complexity is bounded by a constant factor.

Theorem 2 can be easily extended to the two-dimensional (2-D) case. A 2-D systolic array is shown in Fig. 4. The following corollary is obtained. In the Corollary two-way (three-way) means that $C_{i,j}$ can communicate with only $C_{i+1,j}$ and $C_{i,j+1}$ ($C_{i+1,j}$, $C_{i,j+1}$, and $C_{i-1,j}$, respectively). The proof is omitted since a similar technique, as above, is employed.

[Corollary] For any 2-D two-way (even three-way) $km+ln$ time-bounded CA of size $m \times n$, M, there exists a 2-D SA, A, which simulates M in $km+(l+1)n$ steps.

Ackowledgements

The author would like to thank the referee who contributed to the improvements of this paper.

References

1. UMEO, H. Systolic simulation of linear-time-bounded cellular automata, *IECE of Japan*, J66-D, 6, (1983), 715-721.
2. DYER, C. R. One-way bounded cellular automata, *Inform. and Control*, 44, (1980), 261-281.
3. ULLMAN, J. D. Computational aspects of VLSI, *Computer Science Press*, (1984), 495.
4. KUNG, H. T. Why systolic architecture?, *Computer* **15**, 4, (1982), 37-46.
5. KUNG, H. T. and MONICA, S. Lam. Wafer-scale integration and two level pipelined implementation of systolic arrays, *J. of Parallel and Distributed Computing*, **1**, 1, (1984), 32-63.
6. WAKSMAN, A. An optimal solution to the firing squad synchronization problem, *Information and Control*, **9**, 1, (1966), 66-78.

(Received May 13, 1985; revised September 19, 1985)