Title	Time-optimum parallel binary address setting algorithms for array processors(Mathematical Theories on Computing Schemes and Their Applications)
Author(s)	Umeo, Hiroshi
Citation	数理解析研究所講究録 (1983), 494: 185-193
Issue Date	1983-06
URL	http://hdl.handle.net/2433/103566
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
494巻 1983年 185-193

Time-optimum parallel binary address setting algorithms for array processors

Hiroshi Umeo
梅尾 博司

Department of Applied Electronic Engineering
Faculty of Engineering
Osaka Electro-Communication University
18-8, Hatsu-cho, Neyagawa-shi, 572, Japan

Abstract Two time-optimum parallel binary address setting algorithms, both suitable for VLSI implementations on array processors, are presented. One is a pipelined method and the other is a parallel binary division method which is based on the firing squad synchronization algorithm in the cellular automata theory. They are represented by the cellular logics and their time complexities for n processors are $n - 1 + [\log_2 n]$ and $2n - 1$ steps, respectively. Our algorithms are time-optimum ones in different situations, respectively.

1. Introduction

With the recent developments of VLSI technology it becomes possible to design and construct special- or general-purpose VLSI-based parallel cellular computers with reasonable costs[3], [8].

In this paper we consider an address setting problem on cellular computers, consisting of many identical processors each with its own address. The concept of processor address in the parallel computer architecture sometimes plays an important role in the design of parallel algorithms, such as for SIMD-type parallel computers. The problem considered in this paper was originally proposed in the study of conversion of parallel algorithms from the SIMD-type to the MISD[1]. But this problem will be useful for the future general-purpose VLSI-based parallel computers and is interesting in its own right from a theoretical point of view.

The organization of this paper is as follows. Section 2 defines the address setting problem. In section 3 two time-optimum parallel binary address setting schemes, both suitable for VLSI implementations on array processors, are presented. One is a pipelined scheme and the other is a parallel binary recursive division scheme which is based on the firing squad synchronization technique in the cellular automata theory[2], [4], [6]. They are represented by the cellular logics. For n processors, the former requires $n - 1 + [\log_2 n]$ steps, and the latter requires $2n - 1$ steps, respectively. Both of them are time-optimum algorithms in different situations. The conclusion is presented in the last section 4.

2 The Address Setting Problem

Consider a cellular computer M, consisting of identical n processors, where $n = 2^m$ for some integer m. See Fig.1. Each processor is referred to as a cell, denoted by $C_i (0 \leq i \leq n-1)$. As is shown in Fig.1, M is used as an attached processor of a host computer. Each cell C_i ($1 \leq i \leq n-2$) is

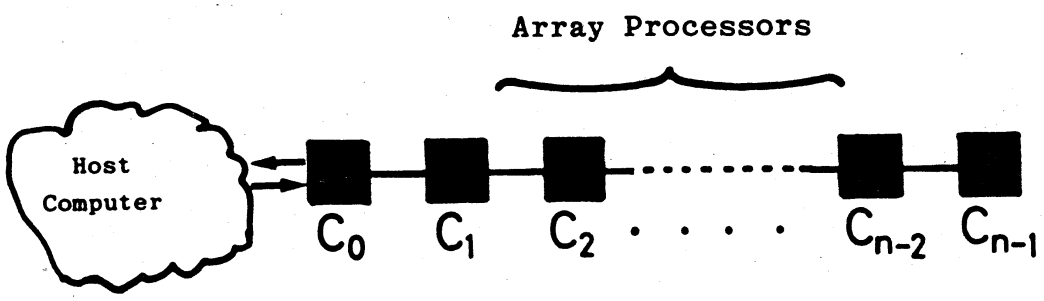

Fig.1 Illustration of an attached cellular computer.

uniformly connected to C_{i-1} and C_{i+1} and can communicate directly with each other in one step. C_0 and C_{n-1} act as end cells, each connected to C_1 and C_{n-2}, respectively.

Each cell is composed of a finite number of finite registers and an address register Ra. The internal state of C_i at step t, denoted by s_i^t, means the content of all registers in C_i at step t, where $t \geq 0$.

At time t=0 the host computer gives an activating signal to M via C_0. Each cell is in quiescent state until the activating signal reaches the cell. A local transition function $\delta : Q^3 \to Q$, which is common to all cells, determines the next state of C_i from the local informations such that $s_i^{t+1} = \delta (s_{i-1}^t, s_i^t, s_{i+1}^t)$, where Q is the set of internal states, $t \geq 0$, and $1 \leq i \leq n - 2$. The states s_0^{t+1} and s_{n-1}^{t+1} are determined from the host computer, s_0^t, and s_1^t and s_{n-2}^t and s_{n-1}^t, respectively.

The address setting problem on M is to design the local transition function δ of M so that the Ra in C_i contains the binary representation of i, where $0 \leq i \leq n - 1$. We place the following restrictions on this problem from an automata-theoretic consideration.

Basic Restrictions:

A1 δ is independent of n.

A2 Each Ra has at least $m = [\log_2 n]$ compartments, each capable of holding 1 bit. They are denoted by $d_0, d_1, \ldots\ldots, d_{m-1}, \ldots$ according to the lowest digit order.
The content of the j-th digit of Ra in C_i at step t is represented by $d_j^t(i)$.

A3 Initially all Ra's are set to "zero", that is, $d_j^0(i) =$ "0", for any i and j.

A4 At most one digit can be altered in one step from "zero" to "one" in each Ra.

A5 The Ra is non-erasable, except for d_{m-1}.

A6 Each cell must print explicitly the symbol "#" on the highest digit d_{m-1} to denote the completion of the address setting in the cell.

3 Optimum Address Setting Algorithms

Two time-optimum parallel address setting schemes are presented. Our algorithms are represented by the wavefront notation, which has been a useful tool for describing cellular automata algorithms[2], [4], [6] and is recently made effective use of in the design of VLSI cellular computers[7]. The readers can easily obtain the local transition function from an algorithm described by the wavefront notation.

In addition to the basic restrictions in the previous section, we consider the problem on the following assumptions.

Optional Assumptions:

- B1 The transition function is restricted to the left-to-right one-way information flow, that is, $s_i^{t+1} = \delta(s_{i-1}^t, s_i^t)$.
- B2 Two-way information flow.
- B3 Each Ra has a special marking on the exact $[\log_2 n]$ compartments.

We first propose an addressing scheme, referred to as pipelined addressing scheme(ver.1), based on Ai, i = 1, 2, .., 6, B1, and B3 assumptions.

[Pipelined Addressing Scheme Ver.1]

(Algorithm) In this scheme an address is set from the lowest digit in each cell. From time t=1, C_0 generates repeatedly m waves, denoted by w_j, j=0, 1, 2, ,m - 1, at the rate of one-wave/one-step. Each wave propagates in the right direction at one-cell/one-step speed. The j-th wave w_j sets the j-th digit of R_a to "zero" or "one" according the following rule given below.

The first wave w_0 sets d_0 to "zero" or "one", alternatively, in each cell, that is, for each i such that $0 \leq i \leq n - 1$,

$$d_0^{i+1}(i) = \begin{cases} 0, & i \text{ is even}, \\ 1, & i \text{ is odd}. \end{cases}$$

The succeeding waves set d_j in C_0 to "zero" such that $d_j^{j+1}(0) = 0$. The j-th wave sets d_j in C_i at step $t = j + i + 1$ from the following pre-set values of C_{i-1} and C_i such that

$$d_j^{t+1}(i) = \begin{cases} \overline{d_j^t(i-1)}, & \text{if } d_{j-1}^{t-1}(i-1) = 1 \text{ and } d_{j-1}^{t}(i) = 0, \\ d_j^t(i-1), & \text{otherwise} \end{cases}$$

where $1 \leq j \leq m-1$, $1 \leq i \leq n-1$, $t = i + j$, and x denotes the complement of x in $\{ 0, 1 \}$. In this setting, d_j's of the first 2^j cells are set to "zero" and those of the next 2^j cells are set to "one" by w_j. These settings are repeated alternatively to the right end.

The wave generation can be stopped by C_0 at time $t = [\log_2 n] + 1$, since C_0 detects the time $t = [\log_2 n]$ on the assumption B3. The last wave w_{m-1} completes the address setting of M at time $t = n - 1 + [\log_2 n]$.

The scheme presented here is a time-optimum one in the sense that the activation of C_{n-1} and its address setting require n and $[\log_2 n] - 1$ steps, respectively.

The validity of our algorithm is easily obtained by the mathematical induction on j. Fig.2 is an illustration of the pipelined addressing scheme(ver.1). Thus we get the following theorem. ∎

[Theorem 1] (Assumptions: Ai,i = 1, 2,.., 6, B1, B3)
The pipelined addressing scheme(ver.1) sets addresses of n processors in $n - 1 + [\log_2 n]$ steps, so in optimum time.

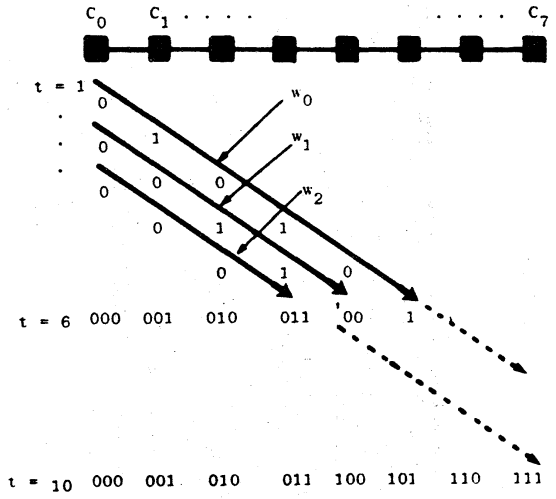

Fig.2 Pipelined addressing scheme(ver.1).

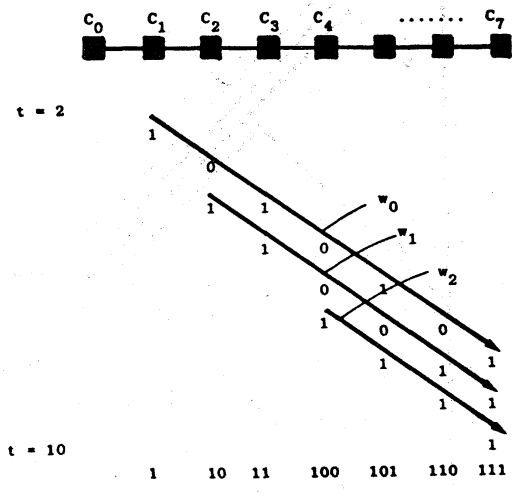

Fig.3 Pipelined addressing scheme(ver.2).

[Pipelined Addressing Scheme Ver.2]

In our above scheme, we made a somewhat unnatural assumption B3 that only C_0 had an address register with specially-marked exact $[\log_2 n]$ compartments. It is shown that this assumption can be removed if we don't require the scheme to make an explicit mark on the highest digit in each cell. It is sufficient to change the wave generation places such that w_j ($1 \leq j \leq m-1$) is generated by C_k, $k = 2^j$, which is initiated by w_{j-1}, shown in Fig.3. At time $t = n - 1 + [\log_2 n]$ the address setting of M is completed. ∎

Now the following theorem is established on the assumptions that Ai, i=1, 2,.., 5, and B1.

[Theorem 2] (Assumptions: Ai, i = 1, 2,.., 5, B1)
The pipelined addressing scheme(ver.2) sets addresses of n processors in $n - 1 + [\log_2 n]$ steps, so in optimum time.

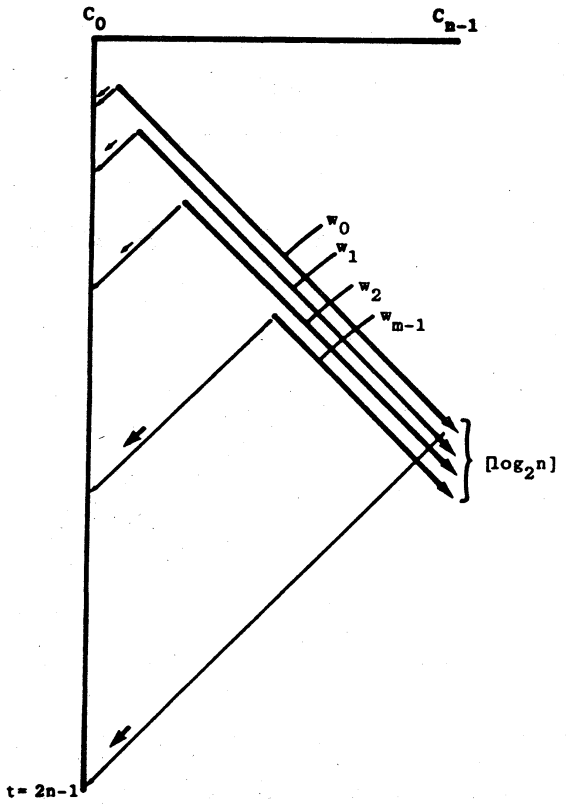

Fig.4 Pipelined addressing scheme(ver.3).

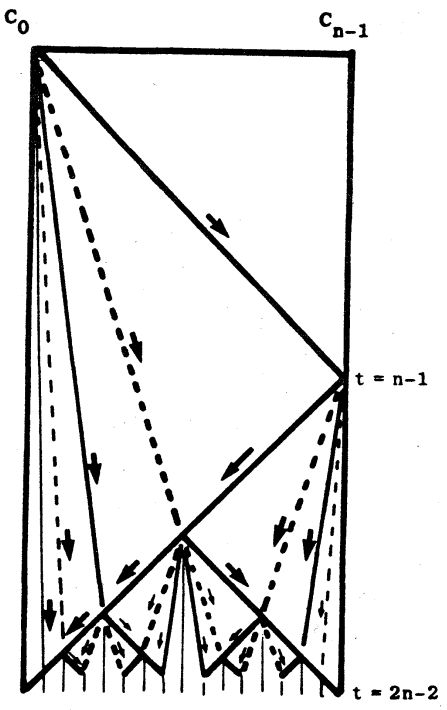

Fig.5 Wave propagation scheme in optimum firing squad synchronization.

[Pipelined Addressing Scheme Ver.3]

Next we consider the problem on the base of Ai, i = 1, 2,.., 6, and B2. If we allow two-way information flows on the processors, a somewhat modified method of ver.2 can set the addresses of M, with satisfying A6. Note that we don't require B3.

The places of the wave generations are the same as those of ver.2. When w_0 gets to C_{n-1}, C_{n-1} sends out a left-propagating signal with velocity one-cell/ one-step. This signal collides with the last wave w_{m-1} at time $t = n - 1 + [\log_2 n]/2$ on C_k, where $k = n - 1 - [\log_2 n]/2$. After the collision the left-going signal sets the highest digit for the cells C_i, $0 \leq i \leq n - 1 - [\log_2]/2$. For other cells, w_{m-1} sets the highest digit. See Fig.4. This addressing scheme requires $2n - 1$ steps. From our assumptions it is clear that this setting scheme is time-optimum. ∎

The following theorem is derived.

[Theorem 3] (Assumptions: Ai, i = 1, 2,..., 6, B2)
The pipelined addressing scheme(ver.3) sets addresses of n processors in $2n - 1$ steps, so in optimum time.

In our schemes discussed above, the setting was begun from the lowest digit in order in each cell. Next we present the second addressing scheme with two-way information flows, referred to as parallel binary division addressing scheme, whose setting is begun from the highest digit in order in each cell. This method is based on the famous firing squad synchronization algorithm known as a method for synchronizing cellular automata[2], [4], [6]. We discuss the problem on the assumptions that Ai, i = 1, 2,.., 6, and B2.

[Parallel Binary Division Addressing Scheme]

(Algorithm) This scheme is based on the time-optimum firing squad synchronization algorithm proposed by A. Wakaman[4] and R. Balzer[6]. Fig.5 shows the wave propagation in the cellular space which fires in optimum time. The addresses are set by these waves from the uppermost digits in each cell. At $t = 1$ C_0 becomes the "general" cell and begins to

prepare firing. Our setting rule is the following simple one: Left-to-right propagating waves, except for the first propagating wave with velocity one-cell/one-step generated by C_0 at t=1, set the uppermost un-preset digits to "zero". Right-to-left propagating waves set the uppermost un-preset digits to "one" in each cell.

The address setting operations are completely finished at the time when the firing occurs. Thus 2n - 1 steps are required. Our algorithm is a time-optimum one, since C_0 needs to receive the signal from C_{n-1} which is activated at time t = n - 1. Note that our algorithm is valid for $n = 2^m$, m = 0, 1, 2,..... . Fig.6 shows the setting scheme in the case n = 8. ∎

Then, we get the following theorem.

[Theorem 4] (Assumptions: Ai, i = 1, 2,..., 6, B2)
The parallel binary division addressing scheme sets addresses of n processors in 2n - 1 steps, so in optimum time.

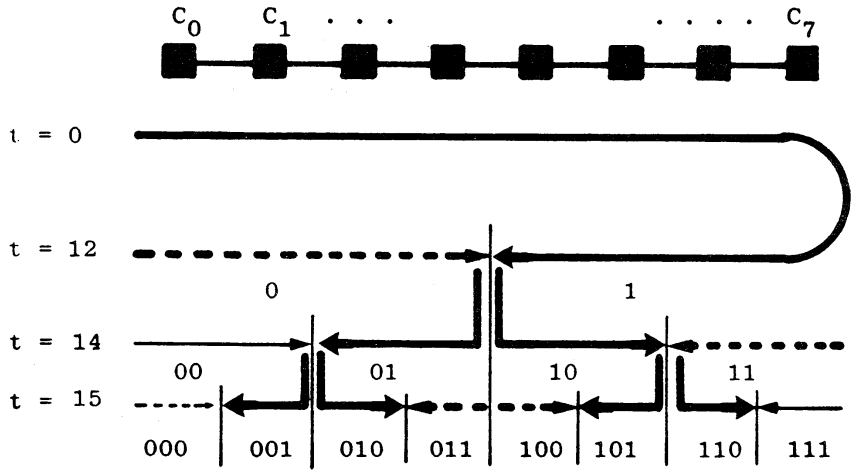

Fig.6 Parallel binary division addressing scheme.

4. Conclusion

Two time-optimum parallel binary address setting schemes, both suitable for VLSI implementations on array processors, are presented. One is a pipelined method and the other is a parallel binary division method which is based on the firing squad synchronization algorithm in the

cellular automata theory. Their time complexities for n processors are n - 1 + $[\log_2 n]$ and 2n - 1 steps, respectively. Several extensions of our discussions, such as to multi-dimensional arrays, are easily obtained.

Acknowledgement The author wishes to thank Dr. Ken'ichi Morita of Osaka Univ. and Associate Professor Hirotomo Aso of Nagoya Univ. for their very variable comments.

References

[1] Umeo, H. and Sugata, K.;"Systolic simulation of synchronous SIMD parallel Computers", Technical Report of IECE of Japan, Vol.82, AL82-21, pp.51-60, (1982).
[2] Kobayashi, K.;"Automata theory and games, Firing squad synchronization problem", Mathematical Sciences, Puzzle I, (1976).
[3] Kung, H.T.;"Why systolic architecture?", IEEE Computer, Jan., pp.37-46, (1982).
[4] Waksman, A.;"An optimal solution to the firing squad synchronization problem", Inf. and Contr., 9, pp.66-78, (1966).
[5] Rosenfeld, A. and Wu, A.Y.;"Reconfigurable cellular computers", Inf. and Contr., 50, pp.64-84, (1981).
[6] Balzer, R.;"An 8-state minimal time solution to the firing squad synchronization problem", Inf. and Contr., 10, pp22-42, (1967).
[7] Kung, S.-Y., Arun, K.S., Gal-Ezer, R.J., and Bhaskar Rao, D.V.;"Wavefront Array Processor; Language, Architecture and Application", IEEE Trans. on Comptrs., c-31, 11, (1982).
[8] Mead, C. and Conway, L;"Introduction to VLSI systems", Addison Wesley, (1980).