INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 9, Number 2, Pages 304–311

REALIZATION OF A TRI-VALUED PROGRAMMABLE CELLULAR AUTOMATA WITH TERNARY OPTICAL COMPUTER

JUN-JIE PENG, LIANG TENG, AND YI JIN

Abstract. A TPCA (tri-valued programmable cellular automata) is proposed in this paper. Implemented based on TOC (Ternary Optical Computer) the TPCA has three advantages over other automata, that is the high programmability, the parallelism of computing and the tri-valued logic implementation. The programmability means that the transformation rules of each cell in CA can be modified at will and be any functions both linear and nonlinear. The parallelism comes from the advantage of optical computing and it can guarantee that CA even with very large-scale can be constructed in parallel and efficient. And the tri-valued implementation would make the CA be more flexible and complex than the counterparts in binary. Combining the characteristics of TOC, the TPCA is discussed in detail. Studied results show that the time complexity has nothing to do with the number of the cells in CA and it is just related to the complexity of the transformation functions. This means that it would be easy to construct more powerful and complicated CA and be widely used in many other fields.

Key words. Cellular Automata, Ternary Optical Computer, Parallel Computing, and Tri-valued Logic.

1. Introduction

Cellular automaton (CA) was put forward by J.von Neumann and Stanislaw Ulam in Mid-20th century, it is a discrete model that consists of a regular grid of cells, each in one of a finite number of states. Since the setup of the CA theory, much attention has been focused on it. Because of the diversity of its statuses and the simplicity of the forms of the transformation rules, CA has been widely studied and applied in considerable fields with varieties of purposes, say in parallel computing model, traffic simulation model, encryption system [1], Built-In Self-Test [2] and so on. Though all these researches and applications are interesting, there is a limitation in common, that is the transformation rules applied to the cells of CA are invariable in the whole transformation process. This would make that the application of CA just suit to be used in the relatively limited and simple areas.

In order to solve more complicated problems, it need to make the cells of CA change following with different rules and the transformation rules can be arbitrary. For example, when construct an adaptive feedback control system, it is needed to modify the transformation rule based on the result of the pre-step. The other problem is the computational speed. Because of the characteristics of CA, it is possible to compute large-scale CA in parallel.

Received by the editors November 12, 2009 and, in revised form, March 6, 2010.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.
The authors would like to thank the support from the Shanghai Leading Academic Discipline Project under Grant(No.J50103), the research project of excellent young talents in the universities in Shanghai,the innovation project of Shanghai University, the Innovation project of Shanghai Municipal Education Commission (No.11YZ09).

In 2000, Prof. Yi Jin from Shanghai University proposed the framework of TOC [3][5][6]. Since then, much research has done and many breakthroughs of TOC have been obtained which include the architecture of TOC, the theory of encoding and decoding, Principles of MSD adder in TOC[4] and decrease-radix design principle[10] and so on. Based on all these researches, especially the tri-valued logic optical computing system [7][8][9], a method that can be used to design highly controllable tri-valued cellular automata (TPCA) is put forward. Combined with the characteristics of CA and optical computing of TOC, the TPCA proposed not only can be used to do computation in parallel, but also can be reconfigured and programmable according to the users' requirements.

The paper is organized as follows: After a brief introduction of TPCA, the following section gives a brief review of the basic concepts the TOC computing platform and the theory of CA. Section 3 discusses the implementation of the computing parallelism and programmability of TPCA in detail. Section 4 is the experiment and results and section 5 is the analysis and conclusion remark.

2. Basic Concepts

2.1. Ternary Optical Computer. TOC is an opto-electronic hybrid parallel computer. Built based on the ternary number system, the TOC expresses the processed information with three states of light, that is the dark state, the vertical polarized state and horizontal polarized state. In the implementation, liquid crystal devices (LCD) together with polarizers are used as the optical encoder, processor and decoder. Where the encoder is mainly used to encode the natural light into the three operable states available in the TOC. The processor is mainly used to process the user's request and output the results. And the decoder is to try to decode the computation results from the unreadable format into more easily understandable format. They all belong to the optical operation components. Besides these components, the control of the rotation of polarizers, the synchronization of different components and the storage to the processed results are all implemented on an embedded system, and human machine interface, the monitor and display of the results are implemented on a host.

In 2008, Dr. Junyong Yan et al. discovered the decrease-radix design principle [10]. In this principle, they put forward a normative method that can be use to reconstruct any kinds of calculate unit in TOC with the 18 kinds of basic units. This is a very important discovery. Following the principle together with the optical computing characteristics of TOC, a new complex CA that can be used to do the computation in parallel is put forward which means it will be very promising in the complex applications of CAs.

2.2. Cellular Automata. In the 1950s, John Von Neumann [11][12] firstly introduced some kinds of CAs which are discrete dynamical systems with simple construction and complex and variable behaviors. In order to study the CA by mathematical tools, many researchers have used the symbol vector (L_d, S, N, f) to describe CA. Where L expresses the space of cells and d is the number of the dimension of the space. All of the possible states of one cell consist of the set of S. N is an arrangement of the neighboring cells, such as the $N = (s_{p_1}, s_{p_2}, \cdots, s_{p_{|N|}})$, where the p is the index of neighboring cell,s_{p_1} means the state of the neighboring cell which is indexed by p_1,and the $|N|$ is the number of elements in N. The symbol f is the transformation function which determines the next state of some specific cells, and the usual form is $f_{(i,j)}(s_{p_1}, s_{p_2}, \cdots, s_{p_{|N|}})$, where (i,j) is used to distinguish the difference of the transformation function of the cells. As mentioned in the system of TPCA, each cell has an unique transformation function.

Generally, there are mainly two methods to implement a CA, one is to use VLSI and the other is to simulate by software. The advantage of VLSI method is that it has the high computation efficiency. However, it also has its disadvantage. Say, if f is set and the VLSI has been made, it is substantially difficult to change the f to construct any other kind of CA. The second method is a serial process, it is need to calculate the f of all cells one by one, therefore when the number of the CA is large, the computation of CA will be much inefficient.

3. TPCA on TOC

3.1. Tri-valued Logic Operations between Matrices. The set S is a tri-valued set consisted of 0,1 and 2, that is {0,1,2}. Based on the set S, there are altogether 19683 kinds of binary tri-valued logic operations. All of these operations can be implemented on TOC. To increase the index efficiency, each of these logic operations is allocated an indexical number, all in decimal.

Table(1) presents a binary tri-valued truth table. Where a_0 through a_8 are the nine values numbered following the rules as table 1 shows. With all these numbers, the tri-valued index sequence of the truth table can be obtained as $a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$.

Ψ_n	0	1	2
0	a_0	a_1	a_2
1	a_3	a_4	a_5
2	a_6	a_7	a_8

TABLE 1 *The truth table of a binary tri-valued table*

Define M as the set of all logic operations.Ψ_n in Table(1) expresses a tri-valued logic operation whose index is n. It can easily figure out that Ψ_n belongs to M, or $\Psi_n \in M$.For the convenience of discussion, the tri-valued sequence is transformed into decimal number as

$$k = \sum_{i=0}^{8} a_i 3^i.$$

Definition 3.1. *If A,B are both m×n dimension matrices, $a_{(i,j)}, b_{(i,j)}$ are the elements of matrices and $a_{(i,j)}, b_{(i,j)} \in \{0,1,2\}$,$i \in \{0,1,\cdots,m-1\}$,$j \in \{0,1,\cdots,n-1\}$,then the tri-valued logic operation between matrices A and B can be defined as the formula 1 shows:*

(1)

$$A\Theta B = \begin{pmatrix} a_{(0,0)}\Psi_{(0,0)}b_{(0,0)} & a_{(0,1)}\Psi_{(0,1)}b_{(0,1)} & \cdots & a_{(0,n)}\Psi_{(0,n)}b_{(0,n)} \\ a_{(1,0)}\Psi_{(1,0)}b_{(1,0)} & a_{(1,1)}\Psi_{(1,1)}b_{(1,1)} & \cdots & a_{(1,n)}\Psi_{(1,n)}b_{(1,n)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(m,0)}\Psi_{(m,0)}b_{(m,0)} & a_{(m,1)}\Psi_{(m,1)}b_{(m,1)} & \cdots & a_{(m,n)}\Psi_{(m,n)}b_{(m,n)} \end{pmatrix}$$

Where in formula(1),$\Psi_{(0,0)}, \Psi_{(0,1)}, \cdots, \Psi_{(m,n)} \in M$, and they all are tri-valued logic operations that may be different or the same. Θ means the operation between two matrices A and B. As TOC can process hundreds of data-bit in one clock cycle, the operation of CA can be processed in parallel.

3.2. Construction of TPCA. For the number of parameters of the CA is two, it can consider that the cells are placed on a plain as matrix. The states of the cells of the CA may be quite different from time to time. No matter how complex the states may be, the transformation of cell from time t to t+1 can be described by functions. It is assumed that the transformation function of (i,j) at time t is $f_{(i,j)}(s_{p_1}, s_{p_2}, \cdots, s_{p_{|N|}})$.

The transformation function consists of two parts, that is the number of arrangement of the neighboring cells N and the tri-valued logic operation sequence $W_{(i,j)}$. Where the form of $W_{(i,j)}$ is $(\Psi_{(i,j)_1}, \Psi_{(i,j)_2}, \cdots, \Psi_{(i,j)_{n-1}})$, $n = |N|$, $\Psi_{(i,j)_1}, \cdots, \Psi_{(i,j)_{n-1}} \in M$.

In this paper, the process of all cells transforming from one state to another next one is called one-step. In one-step, N must be the same, while the $f_{(i,j)}$ can be different from each other. It can be understood as that the parameters of $f_{(i,j)}$ is the same while the tri-valued logic operation sequence $(\Psi_{(i,j)_1}, \Psi_{(i,j)_2}, \cdots, \Psi_{(i,j)_{n-1}})$ be different. In the next one-step, the change of N will be considered. This will set the CA be more flexible and controllable.

3.2.1. Arrangement of the neighboring cells N. As the parameters of $f_{(i,j)}$ in the transformation is determined by the arrangement of the neighboring cells N, it is quite necessary to discuss the details about how to index the neighboring cells in the transformation of CA. Generally, the next state of any specific cell can be determined by the states of its neighborhood cells. It is shown as table(2) which presents the index rules of some specific cell between itself and its neighbors. Actually, the neighborhood cells could be much larger than that of table(2).

⋮	9	10	11	12
23	8	1	2	13
22	7	0	3	14
21	6	5	4	15
20	19	18	17	16

TABLE 2 *The index of neighborhood cells*

Table(2) shows that if the next state of the cell (i,j) indexed by 0 will be calculated, what is needed to focus on is the cells neighboring to 0 in the middle of table(2) according to the number of arrangement in the real application. Firstly set the N, say $N(s_0, s_3, s_7)$ means that the next state of the cell is determined by the states of itself, its right and left cells. In one-step, as N of each cell is the same, the next states of the cells can be determined according to the rules in table (2) indicates. In the real case, there is one more thing should be noticed, that is the boundary issue. For to some of the cells to be processed, their neighbors could be beyond the boundary. To solve the issues like this, two methods work. One is simply set 0 to the states of all of its neighbors, the other is try to cycle the boundary as the cell to be a cycling one.

3.2.2. Tri-valued logic operation sequence $W_{(i,j)}$. To determine the transformation function, two things should be done as mentioned above. One is to set the arrangement of the neighboring cells N and the other is to determine the tri-valued logic operation sequence $W_{(i,j)}$. Obviously the N could be set as the above section describes, this section will discuss the method how to determine the operation sequence $W_{(i,j)}$.

There are altogether m×n functions f in the process of construction of TPCA with two m×n dimension matrices. Each of the cell has a unique index $f_{(i,j)}$ marked by (i,j), where $i \in \{0, 1, \cdots, m-1\}$,$j \in \{0, 1, \cdots, n-1\}$. As N is the same in one-step, to customize the $f_(i,j)$ what is needed to do is to determine the tri-valued logic operation sequence $(\Psi_{(i,j)_1}, \Psi_{(i,j)_2}, \cdots, \Psi_{(i,j)_{n-1}})$. This can be implemented by inserting $W_{(i,j)}$ into the vector $(s_{p_1}, s_{p_2}, \cdots, s_{p_{|N|}})$. For example the transformation function $s^{t+1}_{(i,j)} = s^t_{p_0} \Psi^t_{(i,j)_1} s^t_{p_3} \Psi^t_{(i,j)_2} s^t_{p_7}$ means the state of cell (i,j) at time t+1 is determined by itself $\Psi^t_{(i,j)_1}$, its $\Psi^t_{(i,j)_2}$ right-neighbor and its $\Psi^t_{(i,j)_3}$ left-neighbor. The $(s^t_{p_0}, s^t_{p_3}, s^t_{p_7})$ is the state of N , where $s \in S$. $\Psi^t_{(i,j)_1}, \Psi^t_{(i,j)_2} \in M$.

The tri-valued logic operation sequence of all the cells consist of the Θ operation sequence, such as $(\Theta_1, \cdots, \Theta_{n-1})$, $n = |N|$. Θ is a tri-valued logic operation matrix with each element being a tri-valued logic operation.

Equation(2) is 4×4 CA with N to be (s_0, s_3, s_7) and the Θ operation sequence to be (Θ_1, Θ_2) . The numbers in the Equation, such as 17139, are the index of the corresponding tri-valued logic operation. Apply the analysis result discussed above, it can easily obtain the $W_{(i,j)}$ from the (Θ_1, Θ_2) . Say $W_{(0,0)}$ is $(\Psi_{17139}, \Psi_{15693})$, the numbers indexed by (0,0) of the (Θ_1, Θ_2).

The transformation operation of the state of CA can be defined as a number of Θ logic operations between matrices with operation sequence $A_1\Theta_1A_2\Theta_2 \cdots A_{n-1}\Theta_{n-1}$ A_n. $n = N$. $(\Theta_1, cdots, \Theta_{n-1})$can be obtained as what is mentioned above, and $A_1, A_2,\cdots, A_n$ is obtained by the dislocation superposition method which is discussed in detail hereinafter. The symbol of (L_d, S, N, f) is a vector in CA. It is the base to construct the TPCA and can be set as section 3.1 describes.

$$N = (s_0, s_5, s_7)$$

$$\begin{aligned} \Theta_1 &= \begin{pmatrix} 17139 & 9328 & 19540 & 15693 \\ 10578 & 17347 & 12985 & 18781 \\ 18063 & 10786 & 12241 & 18123 \\ 15792 & 10057 & 11503 & 19599 \end{pmatrix} \\ \Theta_2 &= \begin{pmatrix} 15693 & 19540 & 9328 & 17139 \\ 12985 & 17347 & 10578 & 18781 \\ 10786 & 18063 & 18123 & 12241 \\ 15492 & 19599 & 11503 & 10057 \end{pmatrix} \end{aligned} \tag{2}$$

3.3. Dislocation Superposition. Dislocation superposition is a novel method to compute the transformation operation of CA in parallel. The first step is to store the state of each cell of the CA to a matrix A, and set $a_{(i,j)}$ be the value of the cell of the ith row and jth column.

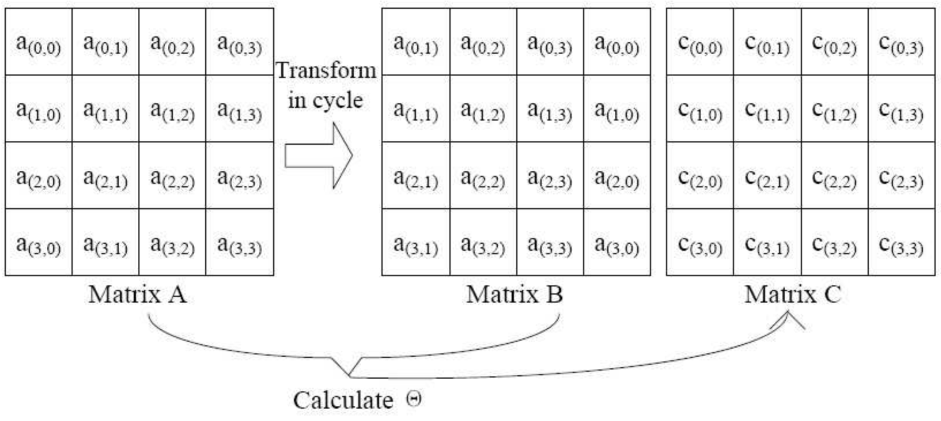


FIGURE 1. *The method of superposition in dislocation*

In this paper, the cycle boundary method is selected to deal with the boundary issue. So, as it is showed in Figure 1, matrix A is transformed to the left in cycle and become the matrix B. The next step is to compute $A\Theta B$. Matrix C is the result of each cell of matrix A calculated by its right neighbor cell. The element of the tri-valued logic operation matrix Θ is determined by the method mentioned above, say the Θ_1 of the equation(2). The following step is to transfrom A in cycle to the right and acquire matrix D, then do the operation $C\Theta D$ .Where the tri-valued logic operation matrix is also determined beforehand, such as the Θ_2 in the equation(2). The result is each cell of matrix A operated with its right neighbor cell and its left neighbor one. According to this rule, it can implement the whole operation with two steps, first try to compute the (L_d, S, N, f) , then change the $(N, W_{(i,j)})$ and do the operation again.

In order to validate the method proposed in the paper, an experiment is discussed in section 4.

4. Experiment and Results

The TPCA operation is an iteration process, and each iteration include a whole calculation process of $A\Theta B$. The following section discusses the implementation of the operation of $A\Theta B$.

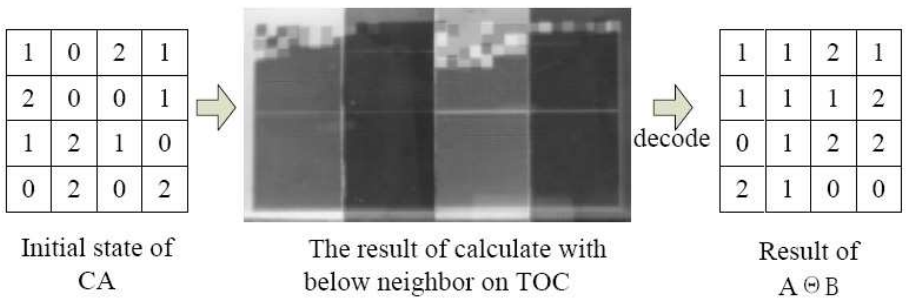
1	0	2	1
2	0	0	1
1	2	1	0
0	2	0	2

TABLE 3. *The state of one CA:*$d = 2; S = 0, 1, 2$

Step 1: Set the states of all cells of the CA as Table(3) shows
Step 2: Customize the (L_d, S, N, f) as equation(2) shows
Step 3: Compute Θ as Figure 1 and Figure 2 shows.
Step 4: If the iteration is finished, go to step 5, otherwise goto step 3.
Step 5: If all the computing is over , go to the Step 6, otherwise go to Step 2 to do the next computing.
Step 6: Output the result of the CA.

The result is

$$\begin{pmatrix} 2 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 1 & 0 & 0 & 2 \\ 1 & 2 & 0 & 1 \end{pmatrix}$$



1	0	2	1
2	0	0	1
1	2	1	0
0	2	0	2

1	1	2	1
1	1	1	2
0	1	2	2
2	1	0	0

FIGURE 2. *The process of $A\Theta B$ on TOC*

5. Analysis of the System

Implemented based on TOC (Ternary Optical Computer) the Tri-valued Programmable Cellular Automata has many advantages over other automata. Besides its high programmability, the parallelism of computing and the tri-valued logic implementation, its time complexity of the TPCA system is not related to the scale of the TPCA. It is only relate to the N , it is $O(|N|)$. This is an interesting result. For it can be easily used in much more complicated system without increase the complexity of the computing effort. Furthermore, the transformation rules of every cell can be customized. These characteristics can much improve the computing speed and increase the complexity of the TPCA which guarantee it is suitable to be used to in more complicated applications.

References

[1] Stephen Wolfram. Cryptography with Cellular Automata.Advances in Cryptology:Crypto '85 Proceedings, LNCS, 1986, 218, 429-432

[2] Peter Dirk Hortensius, Robert D McLeod, Howard C Card. Cellular Automata-Based Signature Analysis for Built-In Self-Test. IEEE Transactions on Computers, 1990, 39, 1273-1283

[3] Yi Jin, Huacan He, Yangtian LU. Ternary Optical Computer Principle [J].Science in China (Series F), 2003, 46(2):145-150

[4] Yi Jin, Yun Fu Shen, Junjie Peng, Gguangtai Ding. Principles and Construction of MSD adder in Ternary Optical Computer, Science in China (Series F),2010, 53 (11): 2159-2168

[5] Yi Jin, Huacan He, Yangtian LU. Ternary Optical Computer Architecture [J].Physical Script, 2005, T118, 98-101

[6] Yi Jin, Huacan He, Lirong Ai. Lane of parallel through carry in ternary optical adder [J]. Science in China (Series F), 2005, 48(1):107-116.

[7] Jiulong Bao, Yi Jin, Chao Cai. An Experiment for Ternary Optical Computer Hundred-Bit Encoder Computer Technology and Development, 2007, 17(2):19-22

[8] Weigang Huang, Yi Jin, Lirong Ai. Design and Implementation of the 100-Bit Coder for Ternary Optical Computers. Computer Engineering & Science, 2006, 28(4):139-142

[9] Yi Jin. Management Strategy of Data Bits in Ternary Optical Computer. Journal of Shanghai University(Natural Science Edition)2007, 13(5)519-523

[10] Junyong Yan, Yi Jin, Kaizhong Zuo. Decrease-radix design priciple for carrying/borrowing free multi-valued and application in ternary optical computer, Scinence in China Series [F].Information Sciences, 2008, 51(10):1415-1426

[11] Stephen Wolfram. Statistical Mechanics of Cellular Automata. Reviews of Modern Physics, 1983, 55, 601-644

[12] Stephen Wolfram. Theory and Applications of Cellular Automata. World Scientific ,1986

School of Computer Engineering and Science & High Performance Computing Center, Shanghai University, Shanghai 200072 P.R.China
E-mail: `jjie.peng@shu.edu.cn` and `yijin@shu.edu.cn`

School of Computer Engineering and Science, Shanghai University, Shanghai 200072 P.R.China
E-mail: `cabbage812@163.com`