DESIGN OF DOUBLE ERROR CORRECTING CODES BASED ON CELLULAR AUTOMATA

SUNG-JIN CHO, UN-SOOK CHOI AND SEONG-HUN HEO

Abstract. Error correcting codes are essential to digital communication system design required exchange and processing of large volumes of data. In this paper, we give an improved encoding and decoding scheme for cellular automata based double error correcting code.

AMS Mathematics Subject Classification : 68Q80
Key words and phrases : Cellular Automata, error correcting code, characteristic matrix, syndrome, PBCA.

1. Introduction

The theory of error detecting and correcting codes is that branch of engineering and mathematics which deals with the reliable transmission and storage of data. The coding theory has wide applications not only the fields of study such as physics, chemistry, biology, engineering, etc but also the newest industry fields such as a weather or communications satellite, computer, DVD, etc. In many cases there are transmission errors caused by noise on the channel. Thus we assume that the information and check bits are transmitted from a source to a destination and that a few of the bits may become erroneous in transit. Error correcting code has become an essential means of ensuring data integrity in a variety of applications.

The system reliability largely depends on the error-free transfer and error correcting ability of data. For detection of bit errors, parity-checking circuits are usually employed([1], [2]). In 1994, Chowdhury et al.[2] proposed the code which is faster decoding than Hsiao code by using the parity bit generated by Null Boundary Cellular Automata(NBCA). Cho et al.[4] designed a single error correcting code based on Periodic Boundary CA(PBCA). They reduced check
bits more than Chowdhury’s result. In this paper, we will give an improved encoding and decoding scheme for cellular automata based double error correcting code.

2. Cellular automata

CA are mathematical idealizations of physical systems in which space and time are discrete, and each cells can assume either the value 0 or 1. The cells evolve in discrete time steps according to some deterministic rule that depends only on logical neighborhood. In the simplest case, Wolfram([8]) suggested the use of a simple two-state, 3-neighborhood one-dimensional CA(1-D CA) with cells arranged linearly in one dimension. Each cell is essentially comprised of a memory element and a combinatorial logic that generates the next-state of the cell from the present-state of its neighboring cells-left, right and self.

If the next-state generating logic employs only XOR logic, then the CA is called a linear CA([1]). So the global transition function of an $n$-cell linear CA can be represented in terms of an $n \times n$ square matrix referred to as the characteristic matrix of the CA. The rules used in a linear CA are shown in Table 1. A uniform CA, where the rules of all the cells are identical, can be characterized by a single rule. In contrast, a hybrid CA, where the rules of the different cells vary, can be characterized by a rule vector.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>$q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$</td>
</tr>
<tr>
<td>90</td>
<td>$q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$</td>
</tr>
<tr>
<td>102</td>
<td>$q_i(t+1) = q_i(t) \oplus q_{i+1}(t)$</td>
</tr>
<tr>
<td>150</td>
<td>$q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$</td>
</tr>
<tr>
<td>170</td>
<td>$q_i(t+1) = q_{i+1}(t)$</td>
</tr>
<tr>
<td>204</td>
<td>$q_i(t+1) = q_i(t)$</td>
</tr>
<tr>
<td>240</td>
<td>$q_i(t+1) = q_{i-1}(t)$</td>
</tr>
</tbody>
</table>

Table 1. Linear CA Rules

On the basis of the properties of state-transition graphs, CA can be broadly classified into two classes - group CA and nongroup CA. While in the state-transition graph of a group CA all states belong to some disjoint set of cycles, the nongroup CA are characterized by the tree-structured transition behavior of acyclic states in the state-transition graph([5], [6]). If the two end cells of a CA are connected to logic 0-state, it is said to be a null boundary CA(NBCA); on the other hand, if the end cells are connected in an end-around fashion, it is called periodic boundary CA(PBCA)([1]).
3. PBCA-based double error correcting code

3.1. Encoding

First, we propose a very simple scheme for generation of \((n, k, 5)\) double error correcting code using regular structure of CA. The basic idea is to allow the CA to run autonomously for a few cycles with the \(k\) information bits \((D)\) loaded as its seed. By using the characteristic matrix of the CA, the \(k\) check bits \((CB)\) are generated. Then the concatenation of \(k\) information bits \(D\) and \(k\) check bits \(CB\) is a codeword.

By using Corollary 3.2([4]) we can prove the following theorem.

**Theorem 3.1.** A \(k\)-cell CA \(C\) generate a \((n, k, 5)\) code if the characteristic matrix \(T\) of \(C\) satisfies the following two conditions:
(a) Every column of \(T\) contains at least four 1’s.
(b) The sum of two (three, four) columns of \(T\) is not a zero vector.

The following is the algorithm which construct \(T\) with the minimum distance 5 by using Theorem 3.1.

**Algorithm 3.1. Construction-\(T\)**

Step 1. Construct the characteristic matrix \(T_k\) of the given \(k\)-cell PBCA with rule \(<102, 102, \cdots, 102, 150>\).

Step 2. Construct \(T_k^3\).

Step 3. Change the \((1, k)\)-element of \(T_k^3\) with 1. Add the matrix

\[
\begin{pmatrix}
1 & 0 & 1 & 0 & \cdots & 1 & 0 \\
0 & 1 & 0 & 1 & \cdots & 1 & 1
\end{pmatrix}
\]

to the changed matrix. Let us call this matrix \(T’\).

Step 4. Add the rows which don’t have l’s at the same positions up to be the matrix with the minimum rows satisfying Theorem 3.1.

Step 5. Let us denote the matrix obtained in Step 4 by \(T\).

The block diagram of the proposed scheme is shown in Figure 1.
Example 3.2. Consider the generation of 5-distance code for 10 information bits (1010101010). Then we obtain $T_{10}$ and $T_{10}^3$ as the following:

$$T = \begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}$$

and

$$T_{10}^3 = \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

By Step 3 we obtain $T'$ as the following:
By Step 4 we obtain the $9 \times 10$ matrix $T$ as the following:

$$T' = \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
\end{pmatrix}$$

Therefore the check bit \(CB\) = \([T][D]\) is as the following:

$${[CB]} = (001011011)^T$$

Hence the codeword is \(1010101010001011011\).

Table 2 shows the comparison of the number of check bits obtained by using compression rules on the characteristic matrix $T$ of the PBCA generating a $(n, k, 5)$ code with the number of check bits of the code proposed by Chowdhury et al.([2]).

### 3.2 Decoding

We consider the problem of decoding to correct the erroneous bit positions from the received information word. The first step in decoding a code is to compute a syndrome. If the received word is $w(= (M'CB'))$, then the syndrome
of \( w \) can be expressed as \( S(w) = Hw^T = H(M'|CB')^T \) where \( H \) is the parity check matrix formed by concatenating the compressed matrix \( T' \) with the identity matrix \( I_k \), that is, \( H = (T'|I_k) \). Thus we have

\[
S(w) = H(M'|CB')^T = (T'|I_k)(M'|CB')^T = T'M' \oplus I_k CB' = TM' \oplus CB'
\]

If the syndrome \( S(w) \) of a received word \( w \) is 0, then the word is assumed to be error free. If \( S(w) \neq 0 \), then errors are detected.

Let \( M_e = \{e_1, e_2, \ldots, e_k\} \) and \( CB_e = \{e_{k+1}, e_{k+2}, \ldots, e_n\} \) denote the error vectors corresponding to the information and check bits in the received word, respectively. Clearly, the received information word \( M' \) and the received check bits \( CB' \) can be represented as follows:

\[
M' = M \oplus M_e, \quad CB' = CB \oplus CB_e
\]

Now

\[
H(M|CB)^T \oplus H(M_e|CB_e)^T = S
\]

<table>
<thead>
<tr>
<th>Information bits(k)</th>
<th>Check bits(n-k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>20</td>
<td>13</td>
</tr>
<tr>
<td>32</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 2. Comparison of check bits of \((n, k, 5)\) code

Since \( H(M|CB)^T = 0 \), \( H(M_e|CB_e)^T = S \). To convert the parity check matrix \( H_{(n-k)\times n} \) into an \( n \times n \) nonsingular square matrix \( T_{aug_{n\times n}} \) additional rows are augmented to the \( H_{(n-k)\times n} \).

\[
T_{aug} = \begin{pmatrix} H_{(n-k)\times n} \\ (Added rows)_{k\times n} \end{pmatrix}
\]

The following theorem gives an idea for the construction of the augmented matrix for the given compressed matrix.
Theorem 3.3. Let

\[ A = \begin{pmatrix} T & I \\ I & 0 \end{pmatrix}. \]

Then \( A \) is invertible and

\[ A^{-1} = \begin{pmatrix} 0 & I \\ I & T \end{pmatrix} \]

Proof. It is straightforward. \( \square \)

The augmented matrix \( T_{\text{aug}} \) for the compressed matrix \( T' \) is defined by

\[ T_{\text{aug}} = \begin{pmatrix} T' & I \\ I & 0 \end{pmatrix} \]

Since \( T_{\text{aug}} \) is invertible by Theorem 3.3 and \( T_{\text{aug}}E = (S|S_{\text{aug}})^T \), the error vector \( E \) is represented as \( E = T_{\text{aug}}^{-1}(S|S_{\text{aug}})^T \).

The proposed decoding algorithm is as follows:

**Algorithm for error correcting**

1. **Step 1.** Compute the syndrome \( S \) for the received information word.
2. **Step 2.** Compute \( T_{\text{aug}} \) and \( T_{\text{aug}}^{-1} \).
3. **Step 3.** Find \( S_{\text{aug}} \) corresponding to \( S \) by using PLA.
4. **Step 4.** Find the error vector \( E \) such that \( E = T_{\text{aug}}^{-1}(S|S_{\text{aug}})^T \).
5. **Step 5.** Find \( C \) such that \( C = C' \oplus E \) and output the resulting code.

**Example 3.4.** For a codeword \((1010101010001011011)\), suppose that there are two errors in positions 4 and 13. That is, the received word is \( C' = (M'|CB') = (1011101010000011011) \). Then

\[ T_{\text{aug}}^{-1} = \begin{pmatrix} I_k & T_{(n-k)\times k} \\ 0_{k \times (n-k)} & I_{n-k} \end{pmatrix} \]

Since \( S = T'D' \oplus CB' = (110100001)^T \) and \( S_{\text{aug}} = (00010000)^T \), \( E = (000100000001000000)^T \).

Hence \( C = (1010101010001011011) \).

A block diagram of the decoding scheme is shown in Figure 2.
4. Conclusion

In this paper, we designed a new \((n, k, 5)\) double error correcting code using a matrix obtained by compressing the characteristic matrix of the PBCA with the special transition rule. The proposed code efficiently reduced the number of check bits. Moreover, algebraic technique is used to give the analysis of \(T_{\text{aug}}\). Thus, the proposed scheme can be implemented with fast encoding-decoding hardware and software. The proposed algorithm is possible to correct double errors by the easy and fast analysis of syndrome.

References


Sung-Jin Cho received his Ph.D at Korea University. Since 1988 he has been at the Pukyong National University. His research interests focus on Cellular Automata, Cryptography and Coding Theory.

Division of Mathematical Sciences, Pukyong National University, Busan 608-737, Korea
e-mail: sjcho@pknu.ac.kr

Un-Sook Choi received her Ph.D at Pukyong National University. Since 2006 she has been at the Tongmyoung University. Her research interests focus on Cellular Automata, Cryptography and Coding Theory.

Department of Multimedia Engineering, Tongmyoung University, Busan 608-737, Korea
e-mail: choies@pknu.ac.kr

Seong-Hun Heo received his master’s degree at Inje University. Since 2005 he has been at Gimhae College. His research interests focus on Cellular Automata, Information Security and Coding Theory.

Department of Computer and Information Science, Gimhae College, Gimhae 621-908, Korea
e-mail: security@gimhae.ac.kr