셀룰라 오토마타를 기반으로 하는 $(n, k, 3)$ 비트 오류정정부호의 설계†

조성진† , 최언숙† , 김한두††, 표용수† , 허성훈†††, 황윤희†

† 부경대학교 수리과학부
††인제대학교 컴퓨터응용과학부
†††부경대학교 정보보호학 협동과정

Design of $(n, k, 3)$ Bit Error Correcting Code Based on Cellular Automata

Sung-Jin Cho† , Un-Sook Choi† , Han-Doo Kim††, Yong-Soo Pyo† ,
Seong-Hun Heo†††, Yoon-Hee Hwang†

† Division of Mathematical Sciences, Pukyong National Univ.
††School of Computer Aided Science, Inje Univ.
†††Dept. of Information Security, Pukyong National Univ.

요 약

물리, 화학, 생물, 공학 등의 학문 분야뿐만 아니라 인공위성과의 통신, 컴퓨터, 컴팩트 디스크 등 첨단산업기술에 까지 광범위하게 응용되고 있는 부호는 입력잘못이나 전파 방해 등 여러 가지 원인으로 오류가 발생할 수 있다. 본 논문에서는 셀룰라 오토마타를 기반으로 하여 검사비트를 생성하고 선형대수 이론을 이용하여 수신된 부호어를 효과적으로 복호할 수 있는 복호기법을 제안한다.

1. 서론

2차 세계대전 이후 통신수단의 급속한 발달에 따라, 정보를 신속하고 정확하게, 그리고 경제적으로 전달하기 위한 연구가 활발해 짐에 따라 부호이론은 수학의 가장 중요한 응용분야의 하나로 각광받고 있다. Shannon[1]이 처음 이 분야의 연구를 개척한 이래, 부호이론은 물리, 화학, 생물, 공학 등의 학문 분야뿐만 아니라 인공위성과의 통신, 컴퓨터, 컴팩트 디스크 등 첨단 산업기술에까지 광범위하게 응용되고 있다. 그런데 데이터의 전송과정에서 입력잘못이나 전파 방해 등 여러 가지 원인으로 자료의 일부가 0과 1이 뒤바뀌는 오류가 발생할 가능성이 존재한다는 것이 문제가 된다. 이러한 가능성은 때로 정보전달에 있어 치명적 오류가 될 수 있다. 이런 문제를 해결하기 위해 제안된 것이 스스로 오류를 발견하고 정정하는 기능을 가진 부호이다. 이러한 부호를 오류정정부호라 한다.

시스템의 신뢰성은 데이터의 오류 정정 능력에 의존한다. 일반적으로 비트오류가 발생했을 때 패리티 검사회로가 쓰인다[2, 3]. 본 논문에서는 특별한 전이규칙을 가지는 셀룰라 오토마타(Cellular Automata, 이하 CA)를 기반으로 하는 $(n, k, 3)$ 비트 오류 정정부호를 설계한다.

먼저 2장에서 CA를 간단히 소개하고 3장에서 특별한 PBCA의 압축된 특성행렬을 이용하여 단일 비트 오류정정부호를 설계하고 4장에서 결론을 맺는다.

2. 셀룰라 오토마타

† 본 연구는 정보통신부 2002 기초기술연구지원사업(IITA:C1-2002-053-03)의 연구비를 지원 받았음.

CA란 동역학계(dynamical system)를 해석하는 한 방법으로 공간과 시간을 이산적으로 다루는 시스템이며, 셀룰라 공간(cellular space)의 기본 단위인 각 셀(cell)이 취할 수 있는 상태를 유한하게 처리하며, 각 셀들의 상태가 국소적인 상호작용에 의해서 동시에 갱신되는 시스템이다. 이러한 CA는 간단하고 가장 간단한 구조를 가지는 1차원 CA (One-Dimensional CA, 이하 1-D CA)에서는 모든 셀들이 선형으로 배열되어 있고 1-D CA 중에서 국소적 상호작용이 세 개의 셀, 즉 자신과 인접한 두 셀에 의해 이루어지는 CA를 3-이웃(3-neighborhood) CA라 한다.

CA는 적용되는 규칙이 XOR 논리로만 이루어진 CA를 선형 CA(Linear CA)이라 한다[4]. 이러한 선형 CA는 상태를 전이시키는 함수를 행렬로 표현할 수 있고 이 행렬을 CA의 특성행렬이라 한다.

<표 1>은 선형 CA에서 사용되는 규칙이다.

rule 60	$q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$
rule 90	$q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$
rule 102	$q_i(t+1) = q_i(t) \oplus q_{i+1}(t)$
rule 150	$q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$
rule 170	$q_i(t+1) = q_{i+1}(t)$
rule 204	$q_i(t+1) = q_i(t)$
rule 240	$q_i(t+1) = q_{i-1}(t)$

<표 1> 선형 규칙

CA의 각 셀에 모두 같은 규칙이 적용된 CA를 uniform CA, 그렇지 않고 2가지 이상의 서로 다른 규칙이 적용된 CA를 hybrid CA라 한다. CA의 규칙에 의해 변화되는 상태를 나타낸 상태전이 그래프의 형태에 따라 Group CA와 Nongroup CA로 분류할 수 있는데, Group CA는 모든 셀들의 상태가 몇 개의 사이클을 이루며 반복되는 CA로 임의의 한 상태에 대한 이전상태가 유일한 반면 Nongroup CA는 상태전이 그래프가 트리 구조를 이루고 있으며 상태전이 함수에 의해 얻어질 수 있는 상태인 도달 가능한 상태와 상태전이 함수에 의해 나타날 수 없는 도달 불가능한 상태로 나누어진다[5, 6].

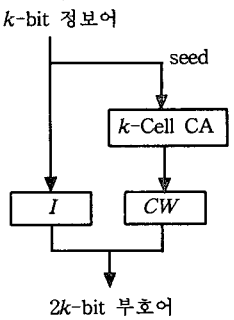
3-이웃 CA에서 가장 왼쪽과 오른쪽의 셀은 2개의 이웃만을 가지므로 세 번째 이웃의 상태를 결정해 주어야 한다. 이것을 CA의 경계조건이라 하고 일반적으로 다음 세 가지의 경계조건을 이용한다. 제일 왼쪽과 오른쪽의 셀들이 0상태에 연결되어 있는 NBCA(Null Boundary CA), 양끝의 셀들이 서로 연결되어 있는 PBCA(Periodic Boundary CA), 가장 왼쪽(오른쪽) 셀의 다음 상태가 그 자신과 그것의 오른쪽(왼쪽) 이웃, 두 번째 오른쪽(왼쪽) 이웃 셀의 상태에 의존하는 IBCA(Intermediate Boundary CA)이다[1]. 본 논문에서는 3-이웃 1-D PBCA를 기반으로 하여 단일 오류를 정정할 수 있는 부호를 설계한다. n개의 셀로 이루어진 3-이웃 1-D PBCA의 특성행렬 T의 각 성분을 t_{ij}라 하면 t_{ij}는 아래와 같다.

$$t_{ij} = \begin{cases} s_i & , \ i=j \\ u_i & , \ j=i+1 \\ r_i & , \ j=i-1 \\ r_1 & , \ i=1, j=n \\ u_1 & , \ i=n, j=1 \end{cases}$$

3. 셀룰라 오토마타를 기반으로 하는 단일 비트 오류정정부호

(1) 부호화

CA의 규칙적인 구조를 이용하여 $(2k, k, 3)$ 단일 비트 오류정정부호를 생성하는 가장 간단한 방법을 먼저 알아본다.



<그림 1> 부호어 생성

기본 개념은 k비트의 정보어(I)가 seed로 로드(load)되면 CA의 특성행렬을 이용하여 검사어(CW)

를 생성하고 정보비트에 연결되어 부호어를 생성한다. <그림1>은 제안된 방법을 보여준다.

[정리 1] 선형(n,k)부호 C의 최소무게가 d이기 위한 필요충분조건은 C의 패리티 검사행렬 H에서 서로 다른 $d-1$개 이하의 열벡터가 독립이다. □

[따름정리 1] 특성행렬 T가 다음 두 조건을 만족한다면, CA는 $(2k,k,3)$부호를 생성한다:
(a) T의 모든 열벡터는 적어도 두 개의 1을 포함한다.
(b) T의 서로 다른 두 개의 열벡터는 적어도 하나의 위치에서 다르다. □

[정리 2] 규칙이 <102, 102, 102, …, 102, 150>인 PBCA는 $(2k,k,3)$부호를 생성하고 이 PBCA의 특성행렬은 T는 다음과 같다.

$$T=\begin{pmatrix}1100\cdots000\\0110\cdots000\\\cdots\\0000\cdots011\\1000\cdots011\end{pmatrix}$$

□

그러나 이러한 방법은 전송되는 정보의 양만큼의 검사비트를 생성해야한다. 본 논문에서는 주어진 정보에 대하여 검사비트를 효과적으로 줄일 수 있는 방법을 제안한다. 고안된 방법은 $(2k,k,3)$부호에서 생성된 i번째 j번째 검사비트 C_i와 C_j를 $C_i \oplus C_j$로 대체함으로써 검사비트의 수를 줄여간다.

<예1> 정보어가 8비트 (10101010)일 때, 부호어를 구해보자. 먼저 (16,8,3) 부호를 생성하는데 사용되는 PBCA의 특성행렬 T_8은 다음과 같다.

$$T_8=\begin{pmatrix}11000000\\01100000\\00110000\\00011000\\00001100\\00000110\\00000011\\10000011\end{pmatrix}$$

또한 i_w와 c_w를 w번째의 정보비트와 검사비트라 하면 다음과 같은 방정식을 만족한다.

$$\begin{aligned}
c_7 &= i_7 \oplus i_6\\
c_6 &= i_6 \oplus i_5\\
c_5 &= i_5 \oplus i_4\\
c_4 &= i_4 \oplus i_3\\
c_3 &= i_3 \oplus i_2\\
c_2 &= i_2 \oplus i_1\\
c_1 &= i_1 \oplus i_0\\
c_0 &= i_7 \oplus i_1 \oplus i_0
\end{aligned}$$

위 방정식에서 정보비트 위치가 모두 다른 두 방정식을 더하여 만들어진 검사비트로 더한 두 검사비트를 대체함으로써 검사비트를 줄인다. c_{ij}를 c_i와 c_j를 더한 방정식이라 하면 위 검사비트를 구하는 방정식은 아래와 같이 구하여도 따름정리 1을 만족한다.

$$\begin{aligned}
c_{74} &= i_7 \oplus i_6 \oplus i_4 \oplus i_3\\
c_{62} &= i_6 \oplus i_5 \oplus i_2 \oplus i_1\\
c_{51} &= i_5 \oplus i_4 \oplus i_1 \oplus i_0\\
c_3 &= i_3 \oplus i_2\\
c_0 &= i_7 \oplus i_1 \oplus i_0
\end{aligned}$$

이는 검사비트가 8비트에서 5비트로 줄면서 단일오류를 정정할 수 있는 (13, 8, 3)부호가 된가. 그러므로 8×8 특성행렬 T에서 5×8로 압축된 행렬 T'을 얻을 수 있다.

$$T'=\begin{pmatrix}11001100\\01100110\\00110011\\00001100\\10000011\end{pmatrix}$$

따라서 PBCA의 특성행렬의 압축으로부터 얻어지는 검사비트는 $CW=T'[I]=(00010)$ 이다. 따라서 전송되는 부호어는 $[I|CW]=(1010101000010)$로 13비트이다. □

<표 2>는 $(2k, k, 3)$ 부호를 생성하는[1] PBCA의 특성행렬 T(정리 2)를 압축하여 생성하는 검사비트수를 나타낸다.

정보비트(k)	9	10	11	12	13	14	15	16
검사비트(n-k)	6	6	6	6	6	6	7	7
정보비트(k)	17	18	19	20	21	22	23	24
검사비트(n-k)	7	7	7	8	8	8	8	9
정보비트(k)	25	26	27	28	29	30	31	32
검사비트(n-k)	9	9	9	9	9	9	10	10

<표 3> (n, k, 3)부호

(2) 복호화

수신된 부호어에서 오류가 발생한 비트를 정정하기 위한 복호법에 대해서 알아본다. 복호하는 첫 번째 단계는 신드롬을 계산하는 것이다.

수신된 부호어가 $w(=(I'|CW'))$라면 신드롬은 $S(w)=Hw^T=H\begin{bmatrix} I' \\ CW' \end{bmatrix}$이다. 여기서, H는 압축행렬 T'와 단위행렬 I_k를 연결하여 형성된 패리티 검사행렬이다. 즉, $H=[T'|I_k]$이다. 따라서

$$S(w)=H\begin{bmatrix} I' \\ CW' \end{bmatrix}=[T'|I_k]\begin{bmatrix} I' \\ CW' \end{bmatrix}$$
$$=T'[I']\oplus I_k[CW']=T[I']\oplus[CW']$$

이다. 신드롬의 값이 0어면 오류가 없음을 나타내고 0이 아니면 수신된 부호어에 오류가 존재한다는 것을 나타낸다.

수신된 부호어에서 $I_e=\{e_1, e_2, \cdots e_k\}$과 $CW_e=\{e_{k+1}, e_{k+2}, \cdots e_n\}$를 각각 정보비트와 검사비트에 대응되는 오류 벡터라고 하자. 그러면, 수신된 정보어는 $I'=I\oplus I_e$이고, 수신된 검사어는 $CW'=CW\oplus CW_e$ 이다.

$$[H]\begin{bmatrix} I \\ CW \end{bmatrix}\oplus[H]\begin{bmatrix} I_e \\ CW_e \end{bmatrix}=[S]$$

여기서 $[H]\begin{bmatrix} I \\ CW \end{bmatrix}=0$이므로,

$$[H]\begin{bmatrix} I_e \\ CW_e \end{bmatrix}=[S]$$

이 된다. 즉, $[H][E]=[S]$ 에서 오류벡터 E는 $[E]=[H]^{-1}[S]$이 된다. 하지만 $[H]^{-1}$이 존재하려면 $[H]$는 $n\times n$정방행렬이어야 한다. 하지만, 여기서는 $[H]$가 정방행렬이 아니므로 패리티 검사행렬 $[H_{(n-k)\times n}]$를 정방행렬 $[T_{aug}]_{n\times n}$로 변환하기 위해서 k개의 추가 행을 $[H_{(n-k)\times n}]$에 덧붙인다. 이때 $[T_{aug}]_{n\times n}$는 역행렬이 존재하도록 구성해야 한다.

$$[T_{aug}]=\begin{bmatrix} [H]_{(n-k)\times n} \\ \cdots \\ [\text{추가행}]_{k\times n} \end{bmatrix}_{n\times n}$$

[정리 3] 블록행렬 A를 $A=\begin{bmatrix} T & I \\ O & I \end{bmatrix}$라 하면 A는 역행렬이 존재하고 $A^{-1}=\begin{bmatrix} O & I \\ I & T \end{bmatrix}$이다. □

정리 3에 의하여 압축행렬 T'에 대한 T_{aug}를 다음과 같이 정의한다.

$$T_{aug}=\begin{bmatrix} T' & I \\ O & I \end{bmatrix}$$

따라서 오류벡터 E는 아래와 같다.

$$[T_{aug}][E]=\begin{bmatrix} T' & I \\ O & I \end{bmatrix}\begin{bmatrix} I_e \\ CW_e \end{bmatrix}_{n\times 1}$$
$$=\begin{bmatrix} S \\ S_{aug} \end{bmatrix}_{n\times 1}$$

$$[E]=[T_{aug}]^{-1}\begin{bmatrix} S \\ S_{aug} \end{bmatrix}$$

<예2> <예1>에서의 부호어(1010101000010)이 전송도중 두 번째 비트에 하나의 오류가 발생했다고 하자. 즉, $C'=(I'|CW')=(1110101000010)$ 이 수신되었다고 하자.

복호화에서 쓰이는 T_{aug}는 다음과 같다.

$$T_{aug}=\begin{bmatrix} T' & I \\ O & I \end{bmatrix}=\begin{bmatrix} 11001100\ 10000 \\ 01100110\ 01000 \\ 00110011\ 00100 \\ 00001100\ 00010 \\ 10000011\ 00001 \\ 10000000\ 00000 \\ 01000000\ 00000 \\ 00100000\ 00000 \\ 00010000\ 00000 \\ 00001000\ 00000 \\ 00000100\ 00000 \\ 00000010\ 00000 \\ 00000001\ 00000 \end{bmatrix}$$

오류벡터는 $[E]=[T_{aug}]^{-1}\begin{bmatrix} S \\ S_{aug} \end{bmatrix}$이다

여기서 $\begin{bmatrix} S \\ S_{aug} \end{bmatrix}$는 T_{aug}에서 열벡터를 나타낸다.

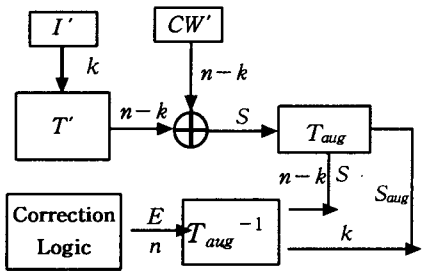
$S=T'[I']\oplus CW'=11010\oplus 00010=11000$이므로, S_{aug}는 01000000이 된다. 수신된 부호어 C'에 오류벡터를 더함으로 오류가 정정된 수신 부호어 C를 얻는다. 즉, $C=C'\oplus E$ 이다.

오류벡터는 $[E]=[T_{aug}]^{-1}\begin{bmatrix} S \\ S_{aug} \end{bmatrix}$

$$= \begin{bmatrix} 11001100\ 10000 \\ 01100110\ 01000 \\ 00110011\ 00100 \\ 00001100\ 00010 \\ 10000011\ 00001 \\ 10000000\ 00000 \\ 01000000\ 00000 \\ 00100000\ 00000 \\ 00010000\ 00000 \\ 00001000\ 00000 \\ 00000100\ 00000 \\ 00000010\ 00000 \\ 00000001\ 00000 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$ 에 의해서

$E = 0100000000000$ 임을 알 수 있다. □

다음 그림은 복호화 방법을 보여준다.



<그림 2> 복호화

4. 결론 및 향후 연구 방향

본 논문에서는 특별한 PBCA의 특성행렬을 압축하여 검사비트를 효과적으로 줄였으며 선형대수의 블록 행렬의 기본적인 이론을 이용하여 주어진 압축행렬 T'에 대한 행렬 T_{aug}을 간단하게 구성함으로써 $(n, k, 3)$ 단일 비트 오류정정부호를 설계하였다. CA의 간단하고, 규칙적이며, 모듈화한 특성을 이용한 이러한 방법은 기존의 VLSI 기술보다 효율적으로 실행될 수 있다는 장점이 있다. 향후 연구계획은 비트오류뿐 아니라 바이트 오류에 대하여 정정이 가능한 PBCA를 기반으로 하는 오류정정부호를 설계하는 것이다.

[참고문헌]

[1] C. E. Shannon, "A mathematical theory of communication", Bell Syst. Tech. J. Vol. 27, 1948, pp. 379-423, 623-656.

[2] P. P. Chaudhuri, et. al., "Additive cellular automata theory and applications", Vol. 1, IEEE Computer Society Press, California, USA, 1997.

[3] D.R. Chowdhury, S. Basu, I.S. Gupta, P.P. Chaudhuri, "Design of CAECC - Cellular Automata Based Error Correcting Code", IEEE Trans. Computers, Vol. 43, June 1994, pp. 759-764.

[4] S.J. Cho, U.S. Choi and H.D. Kim, "Linear nongroup one-dimensional cellular automata characterization on GF(2)", J. Korea Multimedia Soc., Vol. 4, No. 1, 2001, pp. 91-95.

[5] S.J. Cho, H.D. Kim and U.S. Choi, "Behavior of Complemented CA Whose Complement Vector is Acyclic in a Linear TPMACA", Mathematical and Computer Modelling, 36, 2002, pp. 980-986.

[6] S.J. Cho, H.D. Kim and U.S. Choi, "Analysis of complemented CA derived from a linear TPMACA" , Computers & Mathematics with Applications, Vol. 45, 2003, pp. 689-698