PBCA를 이용한 단일 오류 정정 부호의 설계†

조성진† ,황윤희† , 김한두††, 표용수† ,최언숙† , 허성훈†††
† 부경대학교 수리과학부
††인제대학교 컴퓨터응용과학부
†††부경대학교 정보보호학 협동과정

Design of Single Error Correcting Code Using PBCA†

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

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

요 약

오늘날 많은 양의 데이터가 디지털 논리 회로와 상호 연결에 의해서 다양한 컴퓨터 시스템과 서브시스템 사이에서 전송된다. 데이터가 전송될 때 전자적 잡음, 장치 결함, 시간 오류 등에 의해서 오류가 발생한다. 일반적으로 비트 오류가 발생했을 때 패리티 검사회로가 사용된다. 시스템의 신뢰성과 유용성을 높이기 위해서는 효율적인 패리티 검사회로가 요구된다.

본 논문에서는 PBCA의 특성행렬을 이용하여 거리가 3인 부호를 생성하는 방법과 오류 정정에 사용되는 행렬을 구성하는 방법 및 복호기법을 제시한다.

1.서론

오류정정부호는 메모리 시스템 설계, 디지털 데이터 통신 등에 널리 적용된다. 오류정정부호는 원래의 정보에 검사비트를 부여함으로써 설계되며, 이 검사비트를 이용하여 오류를 정정한다.

본 논문에서는 셀룰라 오토마타의 특성행렬을 이용하여 검사비트를 생성하고 수신된 부호어를 복호하는 방법을 제시한다. 오늘날 많은 양의 데이터가 디지털 논리 회로와 상호 연결에 의해서 다양한 컴퓨터 시스템과 서브시스템 사이에서 전송된다. 데이터가 전송될 때 전자적 잡음, 장치 결함, 시간 오류 등에 의해서 오류가 발생한다. 시스템의 신뢰성은 데이터의 오류 정정 능력에 의존한다. 일반적으로 비트오류가 발생했을 때 패리티 검사회로가 쓰인다[1][2]. 하지만 시스템의 신뢰성과 유용성을 높이기 위해서 보다 효율적인 패리티 검사회로에 대한 연구가 필요하다. 본 논문의 목적은 셀룰라 오토마타(Cellular Automata, 이하 CA)의 간단하고, 규칙적이며, 모듈화한 특성을 이용하여 효율적인 부호화 및 복호화 방법을 고안하는 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 CA의 정의, 특성행렬, rule, CA의 종류에 대하여 살펴보고 3장에서는 PBCA를 이용한 단일 오류 정정부호의 생성 및 복호화에 대한 방법을 제시하고 4장에서 결론을 맺는다.

† 본 연구는 정보통신부 2002 기초기술연구지원사업(IITA:C1-2002-053-03)의 연구비에 의해 연구되었음

2. 셀룰라 오토마타

CA란 동역학계(dynamical system)를 해석하는 한 방법으로 공간과 시간을 이산적으로 다루

는 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. 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. 본 논문에서 다루는 CA는 3-이웃 1-D CA에 국한시킨다.

CA를 설명하기 위해서 다음 기호들이 사용된다.

- i : 일차원으로 배열되어 있는 각 셀들의 위치
- t : 시간 단계
- $q_i(t)$: 시간 t에서 i번째 셀의 상태
- $q_i(t+1)$: 시간 $t+1$ 에서 i번째 셀의 상태

세 개의 이웃을 가지는 CA에 대한 다음상태 전이함수(transition function)는 다음과 같이 나타낸다.

$q_i(t+1) = f[q_{i-1}(t), q_i(t), q_{i+1}(t)]$ 여기서 f는 결합논리를 가지는 국소 전이함수이다. f는 3개의 변수를 가지는 Boolean 함수이므로 2^{2^3}, 즉 256개의 다음 상태 전이함수들이 있으며 이것을 CA의 rule이라고 한다.

rule 90: $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$
rule 150: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$

CA의 셀들의 상태를 0과 1의 두 가지 값으로 다루고 주어진 CA의 다음상태 전이함수를 아래와 같이 표현한다. 여기서 첫 행은 시간 t에서 인접한 세 개의 셀들의 가능한 8가지 상태의 배열이고 다음 행들은 시간 $t+1$에서 i번째 셀의 갱신된 상태이다.

이웃 상태	000	100	010	001	110	011	101	111
다음 상태	0	1	0	1	1	1	0	0
(rule 90)								
다음 상태	0	1	1	1	0	0	0	1
(rule 150)								

위의 rule에 대한 논리결합은 다음 식으로 표현할 수 있고 ⊕는 XOR을 나타낸다.

CA는 적용되는 rule의 논리의 종류에 따라 선형 CA(Linear CA), 가산 CA(Additive CA), 비가산 CA(Nonadditive CA)로 분류되는데 각 셀에 적용된 rule이 XOR 논리로만 이루어진 CA는 선형 CA이다. 선형 CA의 상태 전이 함수는 행렬로 표현 될 수 있고 이 행렬을 특성행렬이라 한다. 또한 셀에 적용되는 rule이 XNOR과 XOR 논리로 이루어진 CA를 여원 CA(Complemented CA)라 하고 선형 CA와 여원 CA를 가산 CA라 한다. 셀들의 rule이 AND-OR논리로 이루어진 CA를 비가산 CA라 한다. <표 1>은 가산 CA에서 선형 rule을 보여준다.

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 rule

CA의 각 셀에 모두 같은 rule이 적용된 CA를 uniform CA, 그렇지 않고 2가지 이상의 서로 다른 rule이 적용된 CA를 hybrid CA라 한다. CA의 rule에 의해 변화되는 상태를 나타낸 상태전이 그래프의 형태에 따라 Group CA와 Nongroup CA로 분류할 수 있다. Group CA는 모든 셀들의 상태가 몇 개의 사이클을 이루며 반복되는 CA로 임의의 한 상태에 대한 이전상태가 유일하다. 이와 달리 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. CA에서 가장 왼쪽과 오른쪽의 셀은 2개의 이웃만을 가지므로

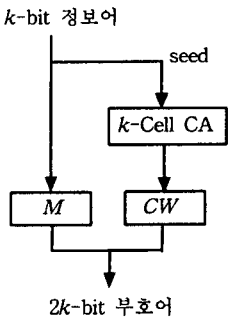
세 번째 이웃의 상태를 결정해 주어야 한다. 이것을 CA의 경계조건이라 하고 일반적으로 다음 세 가지의 경계조건을 이용한다. 제일 왼쪽과 오른쪽의 셀들이 0상태에 연결되어 있는 NBCA(Null Boundary CA), 양끝의 셀들이 서로 연결되어 있는 PBCA(Periodic Boundary CA), 가장 왼쪽(오른쪽) 셀의 다음 상태가 그 자신과 그것의 오른쪽(왼쪽) 이웃, 두 번째 오른쪽(왼쪽) 이웃 셀의 상태에 의존하는 IBCA(Intermediate Boundary CA)이다[3][4].

n개의 셀을 가지는 선형 1-D CA에서는 현재 상태를 다음 상태로 전이시키는 전이함수를 $n \times n$ 행렬로 나타낼 수 있다[5].

3. PBCA를 이용한 단일 오류정정부호

(1) 부호화

CA의 규칙적인 구조를 이용하여 비트 오류정정부호를 생성하는 방법을 알아본다. 기본 개념은 k비트의 정보어 (M)가 seed로 로드(load)되면 CA의 특성행렬을 이용하여 검사어 (CW)를 생성하고 정보비트에 연결되어 부호어를 생성한다. <그림1>은 제안된 방법을 보여준다.



<그림 1> 부호어 생성

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

[정리 2] $k \times k$특성행렬 T의 서로 다른 i $(0 < i < d)$개의 열의 비트별 합(bitwise sum)이 적어도 $(d-i)$개의 1을 포함하면 특성행렬 T를 가지는 k-셀 CA는 거리가 d 인 부호를 생성한다. □

[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 3] rule이 <102, 102, 102, …, 102, 150>인 k-셀 PBCA는 $(2k, k, 3)$부호를 생성한다. □

<예1> rule이 <102, 102, 102, 102, 102, 102, 102, 150>인 PBCA를 이용하여 부호를 생성해 보자.

특성행렬은 $T = \begin{bmatrix} 1&1&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1 \\ 1&0&0&0&0&0&1&1 \end{bmatrix}$ 이다.

정보어가 8비트 (10101010)일 때, 검사어를 구해보면

$$CW = T[M] = \begin{bmatrix} 1&1&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1 \\ 1&0&0&0&0&0&1&1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

(11111110)이 된다.

따라서 부호어는 16 비트 $[M|CW]$= (1010101011111110)이 된다.

(2) 복호화

수신된 부호어에서 오류가 있는 비트를 정정하기 위한 복호방법에 대해서 알아본다. 수신된 부호어를 복호하기 위해 먼저 신드롬(syndrome)을 계산한다.

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

$S(w)=H\begin{bmatrix}M'\\CW'\end{bmatrix}=[T|M_k]\begin{bmatrix}M'\\CW'\end{bmatrix}=$

$[T][M']\oplus[I_k][CW']=[T][M']\oplus[CW']$

로 표현될 수 있다. $S(w)=0$이면 오류가 없음을 나타내고 $S(w)\neq 0$이면 수신된 부호어에 오류가 없음을 나타낸다.

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

$$[H]\begin{bmatrix}M\\CW\end{bmatrix}\oplus[H]\begin{bmatrix}M_e\\CW_e\end{bmatrix}=[S]$$

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

$[H]\begin{bmatrix}M_e\\CW_e\end{bmatrix}=[S]$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 검사 행렬 $[H_{(n-k)\times n}]$를 정방행렬 $[T_{aug}]_{n\times n}$로 변환하기 위해서 k개의 추가 행을 $[H_{(n-k)\times n}]$에 덧붙인다. 이때 $[T_{aug}]_{n\times n}$가 역행렬이 존재하도록 group CA가 되도록 구성한다. 즉, $det[T_{aug}]=1$ 이다.

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

$$[T_{aug}][E]=\begin{bmatrix}[H]\\\cdots\\[\text{추가행}]\end{bmatrix}_{n\times n}\begin{bmatrix}M_e\\\cdots\\CW_e\end{bmatrix}_{n\times 1}=\begin{bmatrix}S\\\cdots\\S_{aug}\end{bmatrix}_{n\times 1}$$

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

<예2> 예1에서 생성한 부호어 (10101010 11111110)이 전송 도중 세 번째 비트에 하나의 오류가 발생했다고 하자. 즉, $C'=(M'|CW')$ $=(1000101011111110)$이 수신되었다고 하자.

복호화 방법에서 쓰이는 T_{aug}는 다음과 같다.

$$T_{aug}=\begin{bmatrix}11000000\ 10000000\\01100000\ 01000000\\00110000\ 00100000\\00011000\ 00010000\\00001100\ 00001000\\00000110\ 00000100\\00000011\ 00000010\\10000011\ 00000001\\00000000\ 10000000\\00000000\ 01000000\\00000000\ 00100000\\00000000\ 00010000\\00000000\ 00001000\\00000000\ 00000100\\00000000\ 00000010\\00000000\ 00000001\end{bmatrix}$$

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

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

$S=T[M']\oplus CW'$

$=1000101010\oplus 11111110=01100000$ 이

므로, S_{aug}는 00000000이 된다.

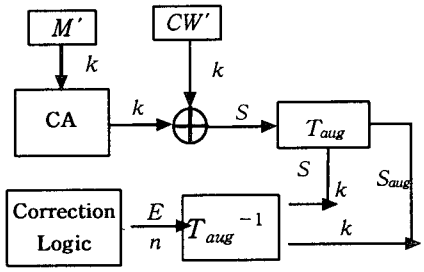
오류벡터는 $[E]=[T_{aug}]^{-1}\begin{bmatrix}S\\S_{aug}\end{bmatrix}$

$$=\begin{bmatrix}00000011\ 00000011\\10000011\ 10000011\\11000011\ 11000011\\11100011\ 11100011\\11110011\ 11110011\\11111011\ 11111011\\11111111\ 11111111\\00000000\ 10000000\\00000000\ 01000000\\00000000\ 01000000\\00000000\ 00100000\\00000000\ 00010000\\00000000\ 00001000\\00000000\ 00000100\\00000000\ 00000010\\00000000\ 00000001\end{bmatrix}\begin{bmatrix}0\\1\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}=\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$$ 에 의해서

$E = 0010000000000000$ 임을 알 수 있다.

□

다음 그림은 복호화 방법을 보여준다.



<그림 2> 복호화

4. 결론

본 논문에서는 rule이 $< 120, 120, \cdots 120, 150 >$ 인 PBCA의 특성행렬을 이용한 단일 오류 정정부호를 설계하였다. CA의 간단하고, 규칙적이며, 모듈화한 특성을 이용한 이러한 방법은 기존의 VLSI 기술보다 효율적으로 실행될 수 있다는 장점이 있다.

참고문헌

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

[2] 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.

[3] 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.

[4] 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, July, 2002, pp. 980-986.

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