

하나의 다수결 논리 디코더를 사용하는
순환 곱 코드에 대한 디코딩

85302

강창언 정연호
연세대학교 공과대학 전자공학과

The decoding for cyclic product codes using
a majority-logic decoder

Chang-Eon Kang Yon-Ho Jeong
Dept. of Electronic Engineering, Yonsei University

ABSTRACT

In this paper, the decoding for the product of (7,3) cyclic code and (3,1) cyclic code was realized with an one-step majority logic decoder. Using the transformation relation between two-dimensional array and one-dimensional array of a code word (or a received word), the parity check sums of the product codes were represented in terms of two-dimensional error bits. The experimental results showed that this decoder was capable of correcting any combination of five random errors and of simultaneously correcting any error burst of length seven or less.

1. 서론

순환 곱 코드는 다른 코드에 비해서 많은 대수적 구조를 지니며 짧은 성분 코드에 의해서 표현된다 는 장점을 지니고 있다. 성분 코드(행 코드와 열 코드)의 생성 다항식을 알게되면, 순환 곱 코드의 생성 다항식을 구할 수 있다. 곱 코드가 순환 적이기 위해서는 첫째 성분 코드는 모두 순환적이 고, 둘째 성분 코드의 겹이는 서로 소 (relatively prime)이어야 하며, 셋째 코드 워드를 채널로 방출하기 전에 코드 워드의 2차원 배열을 1차원 배열로 변환하는 일정한 규칙을 따라야 한다. 순환 곱 코드에 대한 디코더는 보통 두 성분 코드의 각각에 대한 디코더를 종속 접속 (cascade connection) 시킨다. 이와 달리 본 논문에서는 (7,3) 순환 코드와 (3,1) 순환 코드의 곱에 대한 디코더를 하나의 1단계 다수결 논리 디코더에 의해 설계하였다. 성분 코드의 모두가 1단계 다수결 논리 순환 코드이고, 각각의 최소 거리가 d_1, d_2

이면 성분 코드들의 곱은 최소 거리 d_1, d_2 의 1단계 다수결 논리 순환 코드이다. 디코더의 다수결 게이트는 ROM을 사용하며, 디코더를 마이크로 컴퓨터 (Apple-II)에 연결시켜서, 디코더의 동작을 살펴보기도 한다.

2. 선택된 순환 곱 코드

선택된 순환 곱 코드의 성분 코드의 내용은 표 1과 같다.

Table 1. Contents of Component codes

성분 코드 구조	행 코드	열 코드
생성 다항식	$x^4 + x^3 + x^2 + 1$	$x^2 + x + 1$
검사 다항식	$x^3 + x^2 + 1$	$x + 1$
최소 거리	4	3

이후 행 코드에 첨자 1을, 열 코드에 첨자 2를, 순환 곱 코드에 첨자 없이 사용하여 구분하기로 한다. 두 성분 코드의 곱은 (21, 3) 순환 곱 코드이며 최소 거리는 12이다. 순환 곱 코드의 구조는 그림 1과 같이 행 코드의 우측이 정보 비트이고 열 코드의 하측이 정보 비트로 배열되어 있다. 이번에는 곱 코드의 생성 다항식, 생성 행렬, 검사 행렬을 구하기로 한다. 생성 다항식은

$$g(x) = \text{LCM} \{ GCD [g_1(x^{bn_2}), x^{n_1 n_2} - 1], GCD [g_2(x^{an_1}), x^{n_1 n_2} - 1] \} \quad (1)$$

이다. 곱 코드가 순환적이기 위한 조건은

$$a n_1 + b n_2 = 1 \quad (2)$$

이고 $n_1 = 7, n_2 = 3$ 이므로 $a = 1, b = -2$ 이다.

$$\begin{aligned} GCD[g_1(x^{bn^2}), x^{n^1 n^2 - 1}] &= GCD[g_1(x^{-6}), \\ x^{21} + 1] &= x^{12} + x^6 + x^3 + 1 \quad \text{이고, } \therefore \\ GCD[g_2(x^{an^1}), x^{21} + 1] &= x^{14} + x^7 + 1 \end{aligned}$$

이다. 따라서 생성 다항식은

$$g(x) = LCM\{x^{12} + x^6 + x^3 + 1, x^{14} + x^7 + 1\} = x^{18} + x^{17} + x^{16} + x^{14} + x^{11} + x^{10} + x^9 + x^7 + x^4 + x^3 + x^2 + 1 \quad \text{--- (3)}$$

이고 검사 다항식 $h(x)$ 는

$$h(x) = (x^{21} + 1) / g(x) = x^3 + x^2 + 1 \quad \text{--- (4)}$$

이다. 생성 행렬 G , 검사 행렬 H 는 각각 식

(5), (6)이 된다.

$$G = \left[\begin{array}{cccccc|cccccc|cc} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{array} \right] \quad \text{--- (5)}$$

$$H = \left[\begin{array}{cccccccccccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 \end{array} \right] \quad \text{--- (6)}$$

한편 코드 워드 (혹은 수신된 워드)의 2차원 배열과 1차원 배열의 변환 관계는 표 2와 같다.

Table 2. The transformation between two-dimensional array and one-dimensional array

$$\left[\begin{array}{cccccc} 00 & 01 & 02 & 03 & 04 & 05 & 06 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 20 & 21 & 22 & 23 & 24 & 25 & 26 \end{array} \right] \Leftrightarrow \left[\begin{array}{cccccc} 0 & 15 & 9 & 3 & 18 & 12 & 6 \\ 7 & 1 & 16 & 10 & 4 & 19 & 13 \\ 14 & 8 & 2 & 17 & 11 & 5 & 20 \end{array} \right]$$

(2차원 배열)

(1차원 배열)

표 2에서 곱 코드의 정보 비트는 2차원 배열에서는 26, 25, 24이고 1차원 배열에서는 20, 11, 5이다.

3. 디코더의 설계

에피 패턴은 $e = (e_0, e_1, e_2, \dots, e_{20})$ 이고 신드롬은 1차원 원소이며, 표 2를 이용하여 2차원 원소로 변환하면 식 (7)이 된다.

$$\begin{aligned} s_0 &= e_{00} + e_{04} + e_{15} \\ s_1 &= e_{11} + e_{15} + e_{26} \\ s_2 &= e_{22} + e_{04} + e_{15} + e_{26} \\ s_3 &= e_{03} + e_{04} + e_{26} \\ s_4 &= e_{14} + e_{04} \\ s_5 &= e_{25} + e_{15} \\ s_6 &= e_{06} + e_{26} \\ s_7 &= e_{10} + e_{04} + e_{15} \\ s_8 &= e_{21} + e_{15} + e_{26} \\ s_9 &= e_{02} + e_{04} + e_{15} + e_{26} \quad \text{--- (7)} \\ s_{10} &= e_{13} + e_{04} + e_{26} \\ s_{11} &= e_{24} + e_{04} \\ s_{12} &= e_{05} + e_{15} \\ s_{13} &= e_{16} + e_{26} \\ s_{14} &= e_{20} + e_{04} + e_{15} \\ s_{15} &= e_{01} + e_{15} + e_{26} \\ s_{16} &= e_{12} + e_{04} + e_{15} + e_{26} \\ s_{17} &= e_{23} + e_{04} + e_{26} \\ e_{26} \text{에 대해서 orthogonal 관계가 있는 검사 합} & \text{(check sum)은 식 (8)이다.} \\ A_1 &= e_{21} + e_{25} + e_{26} = s_5 + s_8 \\ A_2 &= e_{23} + e_{24} + e_{26} = s_{11} + s_{17} \\ A_3 &= e_{20} + e_{22} + e_{26} = s_2 + s_{14} \\ A_4 &= e_{11} + e_{15} + e_{26} = s_1 \\ A_5 &= e_{13} + e_{14} + e_{26} = s_4 + s_{10} \quad \text{--- (8)} \\ A_6 &= e_{10} + e_{12} + e_{26} = s_7 + s_{16} \\ A_7 &= e_{01} + e_{05} + e_{26} = s_{12} + s_{15} \\ A_8 &= e_{03} + e_{04} + e_{26} = s_3 \\ A_9 &= e_{00} + e_{02} + e_{26} = s_0 + s_9 \end{aligned}$$

$$\begin{array}{ll} A_{10} = e_{16} + e_{26} & = S_{13} \\ A_{11} + e_{06} + e_{26} & = S_6 \end{array}$$

식 (8)의 검사 합은 디코더의 다수결 게이트의 입력 연결에 사용되고 있다. 이 코드의 1단계 다수결는 티 디코더는 그림 2에서 보여주며, 디코더 구성에 사용된 IC는 표 3과 같다.

Table 3. The IC's used for the decoder

항 목	I C 번호
mod-2 게이트 (eOR)	74LS 86
신드롬 레지스터	74LS 273
바퍼 레지스터	74LS 125
다수결 게이트	2716 (ROM)
부정 (인버터)	74LS04

4. 실험과 결과 고찰

실험 장치 도형 (그림 3)과 같이 마이크로 컴퓨터 (Apple-II)를 디코더에 연결시켜 디코더의 동작을 실험하였다. 에러 패턴은 보조 기억 장치 (디스크)에 저장하고 디코더의 제어 프로그램, 수신된 워드, 정정된 워드는 각각 Apple-II 의 기억 장소 \$800, \$3000, \$4000 에 저장한다. \$3000 에 있는 수신된 워드는 I/O 를 통해서 디코더로 보내고 디코더에 의해 정정된 워드는 I/O 를 통해서 \$4000 에 저장한다. 디코더를 동작시키는데 필요한 클럭은 Apple-II의 클럭을 사용하였다. 처음 21개 클럭이 경과하면 처음 수신된 비트의 신드롬이 신드롬 레지스터에 만들어지고 동시에 하나의 수신 워드 (21비트)가 바퍼 레지스터에 저장된다. 다수결 게이트의 입력 수는 11개이고 1의 값인 입력 수가 6개 이상이면 게이트의 출력은 1이다.

출력이 1인 상태 수 1024개가 ROM에 기억되어 있다. 게이트의 출력이 1이면 처음 수신된 비트는 에터임을 의미하며 바퍼 레지스터로부터 방출되는 내용을 정정시킨다. 이 후 21개 클럭이 경과하면 하나의 수신된 워드가 정정된다. 이 디코더는 5비트의 산발 에터와 동시에 7비트의 연접 에터의 정정 능력을 가지고 있음이 실험을 통해서 확인되었다.

5. 결론

본 논문에서 $(7, 3)$ 순환 코드와 $(3, 1)$ 순환 코드의 곱에 대한 디코더를 하나의 1단계 다수결는 티 디코더로 구성하였다. 코드 워드 (혹은 수신

된 워드)의 2차원 배열과 1차원 배열의 변환 관계를 사용해서, 순환 곱 코드의 검사 합들이 2차원에서 비트들에 의해 표현되었다.

디코더를 마이크로 컴퓨터 (Apple-II)에 연결 시켜 실험하였는데 이 디코더는 5비트의 산발 에터와 동시에 7비트의 연접 에터를 정정할 수 있음이 확인되었다.

6. 참고 문헌

1. S. Lin, "An Introduction to Error-Correcting Codes", Prentice - Hall, 1970
2. H.O. Burton, E.J. Weldon Jr., "Cyclic Product Codes", IEEE Trans. on Information Theory pp. 433-439 Vol. IT-11 1965
3. S. Lin, E.J. Weldon Jr. "Further Results on Cyclic Product Codes", IEEE Trans. on Information Theory pp.452-459 Vol. IT-16 1970
4. W.W. Peterson, E.J. Weldon Jr. "Error - Correcting Codes", 2nd Edition, The M.I.T Press, 1972

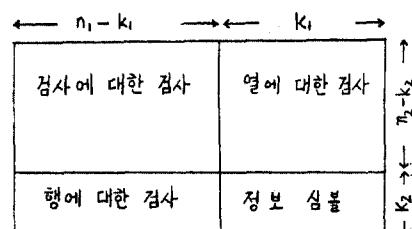


Fig 1. Structure of Cyclic product codes

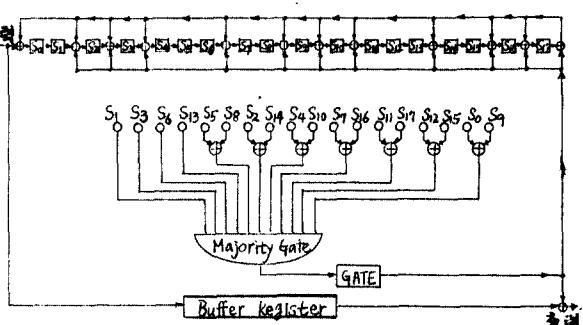


Fig 2. The one-step majority-logic decoder for the product of $(7,3)$ cyclic code and $(3,1)$ cyclic code

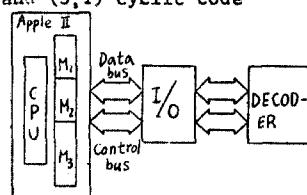


Fig 3. A block diagram of the experimental apparatus