論文 85-22-5-4

Cascade 방식을 이용한 순환곱셈코드의 시스템 설계

(Design of A Cascaded Cyclic Product Coding System)

金 信 姈*, 康 昌 彦**

(Sin Ryeong Kim and Chang Eon Kang)

要 約

본 논문에서는 random 및 burst 에러를 동시에 정정할 수 있는 순환 곱셈 코드를 실현하였다.

우선 두 코드의 곱을 실현하는 방법을 제시하였고, (7, 4)순환 Hamming 코드와 (3, 1)순환 코드를 이용하여 실제로 하드웨어를 구현하였다.

시스템은 인코더와 디코더 그리고 인터페이스 회로로 구성하였고 마이크로 컴퓨터를 이용하여 실험을 하였다. 인코더는 각 부 코드의 인코더에 지연 소자만 넣어 실현하였고, 디코더는 가장 간단한 디코딩 방식인 에러 trapping 디코더를 cascade 연결하여 실현하였다.

본 연구의 결과로서 이 순환 곱셈 코드는 디코딩이 쉽고, 4개의 random 에러와 burst 길이 8인 에러를 정정할 수 있으며, 성능은 일반 순환 코드보다 $10^2 \sim 10^3$ 정도 좋음을 알 수 있었다.

Abstract

In this paper, the cyclic product codes which are capable of correcting random erros and burst errors simultaneously have been designed and constructed.

First, the procedure for product of two cyclic codes is shown and then the encoder and decoder system using the (7, 4) cyclic Hamming code and the (3, 1) cyclic code is implemented.

The micro-computer is used for experiment and the system consists of encoder, decoder and interface circuits. The encoder of cyclic product code is implemented by interlacing encoders while the decoder is implemented by cascading decoders that interlace error trapping decoders.

In conclusion, cyclic product codes are easily decodable and are capable of correcting four random errors and eight-burst errors.

Better performance is obtained with low error rate.

I. 序 論

송신된 코드어가 채널을 통하여 수신 될 때, 채널 에러에 의하여 수신된 신호에 random과 burst 형태의 에러가 발생하게 된다. 일반적으로 여러 통신 채널에 있어서 에러는 random이나 burst의 형태로 각각 독립적으로 나타나는 것이 아니라, 이 두가지가 복합적인 모습으로 발생하게 된다. 그러므로 쉽고 효율적인 방법으로 디코딩할 수 있고 random 및 burst 에러를 동시에 정정할 수 있는 코드가 필요하게 되었으며, 그 대표적인 것으로서 곱셈코드가 있다.

이 곱셈 코드는 고차원(multidimension) 코드 중에서 가장 잘 알려진 코드로서 1954년에 Elias에 의하여 처음으로 소개되었고, 곱셈 코드의 최소 거리(minimum

*準會員, **正會員, 延世大學校 電子工學科

(Dept. of Electron. Eng., Yon Sei Univ.)

接受日字 : 1984年 12月 20日

distance)가 각 부코드의 최소 거리의 곱이 됨을 보여 주었다.[4]

1965년에 burton과 weldon은 일반적인 곱셈 코드보다 수행을 더 간단하게 하기 위하여 순환 코드의 특성을 지니는 순환 곱셈 코드(cyclic product code)를 정의하고[5], 곱셈 코드가 순환 코드로 되기 위한 조건 및 생성다항식이 부코드의 함수임을 증명하였다.

1968년에는 Abramson에 의하여 순환 곱셈 코드의 디코딩 방식중에서 가장 실현 가능한 cascade 디코딩 방식이 연구되어졌다.[6]

본 논문에서는 서론에 이어 제Ⅱ장에서 곱셈 코드 및 순환 곱셈 코드의 특성을 소개하였고, 제Ⅲ장에서는 computer simulation을 통하여 (7, 4) Hamming 코드와 (3, 1) 순환 코드를 이용한 곱셈 코드의 에러 정정 능력과 채널 에러에 따른 성능(performance)을 조사하였고 제Ⅳ장에서는 하드웨어를 제작하여 실험을 수행하였으며 끝으로 결론을 내렸다.

Ⅱ. 곱셈 코드 및 순환 곱셈 코드

1. 곱셈코드

여러 통신 채널에 있어서는 강력한 코드를 얻기 위하여 두 개 이상의 코드를 조합하게 되었고, 그 대표적인 코드가 곱셈 코드이다.

C_1은 (n_1, k_1)코드이고 C_2는 (n_2, k_2) 코드라 가정하자. 이 때 길이 n의 선형 코드 C가 $n=n_1 \cdot n_2$의 조건을 만족한다면 이 코드는 n_1과 n_2의 행렬형으로 표현될 수 있다.

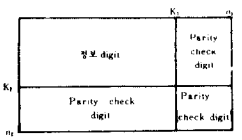


그림 1. 코드 배열
Fig. 1. Code array.

이와 같이 두 코드 C_1과 C_2의 곱셈 코드 C는 각 행이 C_1의 코드 단어이고, 각 열이 C_2의 코드 단어로서 구성된 코드이다.

이 곱셈 코드의 에러 정정 능력은 다음과 같다. 우선 random 에러 정정 능력에 대하여 알아보면, C_1과 C_2코드의 최소 거리가 d_1과 d_2라면 곱셈 코드의 최소 거리는 각각의 부코드의 최소 거리를 곱한 것과 같다. 즉 $d=d_1 \cdot d_2$ 이다. 그러므로 곱셈 코드의 random 에러 정정 능력 t는 다음과 같다.

$$t=\frac{d-1}{2}=\frac{d_1 d_2-1}{2} \qquad (1)$$

곱셈 코드의 burst 에러 정정 능력은 B_1과 B_2를 C_1과 C_2코드의 burst 에러 정정 능력이라 할 때, 코드 array가 열로 수신된다면 $n_1 B_2$보다 작은 burst 에러가 수신 신호에 나타나게 된다.

이 때 C_2코드가 B_2의 burst에러 정정 능력을 가지므로 행으로 디코딩하면 burst 에러가 정정된다. 그역도 성립하므로 곱셈 코드의 burst 에러정정 능력 B는

$$B \geq \max\{n_1 B_2,\ n_2 B_1\} \qquad (2)$$

이고, 위의 결과로서 곱셈 코드는 t개의 random 에러와 B개 이하의 burst에러 정정 능력을 가진 코드임을 알 수 있다.

2. 순환 곱셈 코드

코딩에서는 대개 순환 코드의 실현이 간단하므로 이러한 순환성질을 이용한 코드가 순환 곱셈 코드이다.

순환 곱셈 코드는 부코드인 C_1과 C_2코드가 순환 코드이고 n_1, n_2가 $a \cdot n_1 + b \cdot n_2 = 1$인 조건을 만족하는 코드이며, 코드 행렬은 식(3)과 같다.

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0(n1-1)} \\ a_{10} & a_{11} & \cdots & a_{1(n1-1)} \\ \vdots & & & \vdots \\ a_{(n2-1)0} & & \cdots & a_{(n2-1)(n1-1)} \end{bmatrix} \qquad (3)$$

이 때 코드 벡터 b_i는 Chinese remainder정리에 의하여 나타낼 수 있는데 이 정리를 이용하여 코드 벡터 b_i를 구해보면, $[a_{ij}]$가 순환 곱셈 코드의 한 코드 단어 $i=0, 1, \cdots, n_1 n_2-1$에 대하여 i의 modulo n_1을 i_1으로 i의 module n_2를 i_2라 할 때, $n=n_1 \cdot n_2$인 n차의 코드 벡터 $[b_i]$는 $a_{i_2 i_1}=b_i$에 의해 나타난다.

생성다항식 $g(X)$는 $g_1(X)$가 (n_1, k_1)코드의 생성다항식이고 $g_2(X)$가 (n_2, k_2)코드의 생성다항식이라면

$$g(X)=\mathrm{GCD}[X^{n_1 n_2}+1,\ g_1(X^{bn_2}) \cdot g_2(X^{an_1})] \qquad (4)$$

이고, 식(4)에서 GCD는 최대 공약수를 나타내고, 순환 곱셈 코드가 인터레이스 형태로 구성될 수 있음을 알 수 있다.

이 순환 곱셈 코드는 곱셈 코드의 한 종류이므로, random에러 정정 능력 t는 최소 거리 d가 $d_1 \cdot d_2$ 이므로

$$t=\frac{d-1}{2}=\frac{d_1 \cdot d_2-1}{2} \qquad (5)$$

이고 일반 곱셈 코드의 에러 정정 능력과 같다.

C_1코드가 코드 길이 n_1과 random에러 정정 능력 t_1 가지고 burst에러 정정 능력 B_1을 가진다고 하자. 그리고 C_2코드는 n_2, t_2, B_2를 가진다고 하면 순환 곱셈 코드의 burst에러 정정 능력은 다음과 같다.

$$B \geq n_1 t_2 + B_1$$
$$B \geq n_2 t_1 + B_2 \qquad (6)$$

이와 같은 순환 곱셈 코드를 디코딩하는 방식은 여러가지가 있지만[3,6]본 논문에서는 cascade 디코딩 방식을 이용한다.

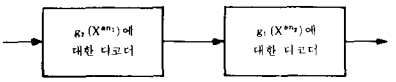


그림 2. Cascade 디코더
Fig. 2. Cascade decoder.

이 cascade 디코더는 각 부 코드의 디코더를 cascade 연결한 것으로서 실현 가능하고 성능이 좋다. 하지만 이 디코딩 방식에는 다음과 같은 permanent에러 패턴이 존재한다.

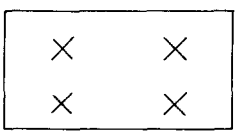


그림 3. Permanent 에러 패턴
Fig. 3. Permanent error pattern.

이 permanent 에러 패턴에 대한 상한선을 구해보면 일반적인 디코더는 t개 이하의 에러 정정 능력을 가지는 반면에, cascade 디코더는 t개 이상의 에러도 어느 정도 정정할 수 있으므로 성능이 좋음을 알 수 있다.

Ⅲ. (21, 4)순환 곱셈 코드의 설계 및 computer simulation

1. (21, 4)순환 곱셈 코드의 설계

이 코드는 C_1코드로 (7, 4)hamming 코드를 사용하고, C_2코드로서 (3, 1)순환 코드를 사용하여 설계 할 수가 있고 코드 길이 n=21, 정보량 k=4 이며 최소거리 d=9임을 알 수 있다. 생성 다항식 g(X)는 (n-k)차 다항식이고 식(4)를 이용하여 구할 수 있다. C_1, C_2코드의 생성 다항식은

$$g_1(X) = 1+X+X^3$$
$$g_2(X) = 1+X+X^2 \qquad (7)$$

이고, $g_1(X^{-6})$와 $g_2(X^7)$을 곱하여 $(X^{21}+1)$과의 최대 공약수를 구하면 된다.

$$g(X) = 1+X+X^3+X^7+X^8+X^{10}+X^{14}+X^{15}+X^{17} \qquad (8)$$

위의 생성다항식을 이용하여 다음과 같은 인코더와 디코더를 설계한다.

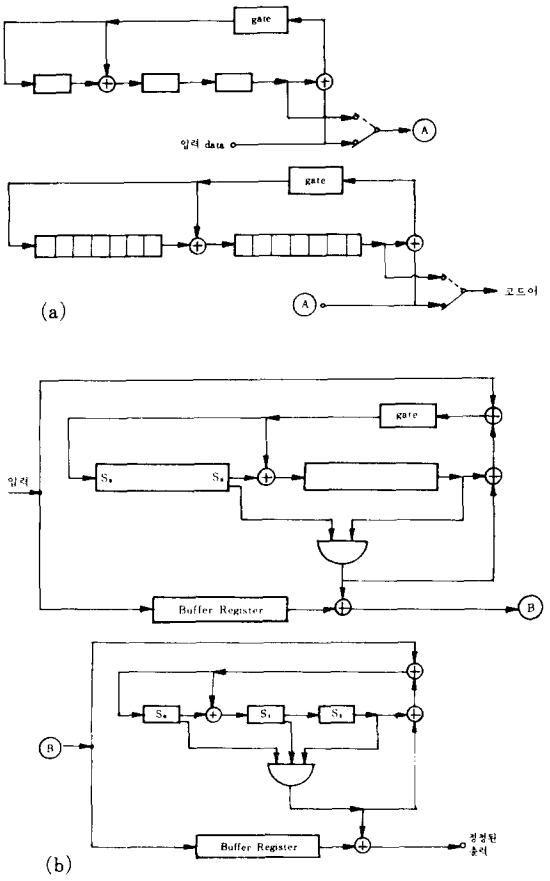


그림 4. (21, 4)순환곱셈 코드의 시스템
Fig. 4. (21, 4) Cyclic Product Coding System.

이러한 시스템을 사용하였을 때, random에러 정정 능력은 d가 9이므로 4개이고 burst 에러 정정 능력은 인터레이스된 형태이므로 길이 8 이하임을 알 수 있다.

2. Computer Simulation

순환 곱셈 코드의 인코더를 통하여 전송된 (21, 4) 코드 단어는 체계적인 형태로 채널을 통과하게 되고, 채널에서의 에러로 인하여 수신된 코드어에는 에러가 발생하게 된다. 이 수신된 코드어는 디코더를 통과함으로써 정정된 단어를 얻을 수 있다.

본 논문에서는 TRS-80을 이용하여 simulation을 행하였고, 그 결과로 4개의 random 에러를 거의 완전히 정정함을 알 수 있었다.

그리고 시스템을 평가하는 기준인 성능은 송신된 신호에 대하여 수신단에서 에러를 정정 못할 확률 PM을 구함으로써 얻을 수 있었다. 여기서 채널은 BSC (binary symmetric channel)이라 가정하고, 에러 확

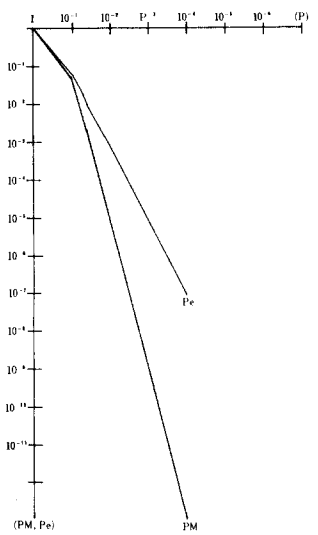
률을 P라 할 때 P는 통계적으로 독립이라고 가정한다. 첫번째 디코더에서 에러를 정정 못할 확률을 Pe라고할 때 시스템의 성능은 다음의 표 1과 같으며 PM은 식(9)에서 표시된다.

$$PM \leq k_1 \sum_{i=t+1}^{n_2} \binom{n_2}{i} Pe^i (1-Pe)^{n_2-i} \qquad (9)$$

표 1. 성 능
Table 1. Performance.

P	Pe	PM
• 1	• 068	• 053
• 05	• 019	4.54E-03
• 04	• 013	1.97E-03
• 01	8.75E-04	9.19E-06
5 E-03	2.21E-04	5.91E-07
1 E-03	8.97E-06	9.66E-10
8 E-04	5.74E-06	3.96E-10

(단, $E-x=10^{-x}$)



위의 표 1에서 이 코드는 10^{-3}의 채널 에러 확률을 가지는 채널에서 에러가 거의 없는 것으로 간주될 수 있음을 알 수 있었다.

Ⅳ. 실험 및 결과 고찰

본 논문에서는 실험 결과를 직접 모니터에서 볼 수 있도록 마이크로 computer와 접속하여 실험을 하였다. 이 마이크로 computer는 APPLE-Ⅱ를 이용하였고, CPU와 memory를 computer에서 이용하기 위하여 8255로 인터페이스 회로를 제작하여 인코더와 디코더

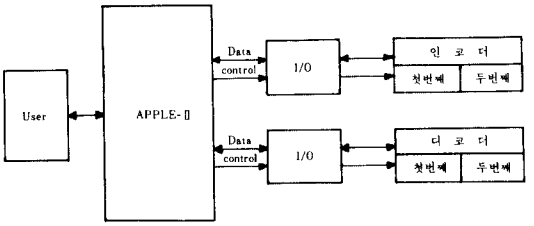


그림 5. 실험에 대한 계통도
Fig. 5. Experimental Modeling.

에 연결시켰다.

실험 방법은 프로그램 즉, 소프트웨어로 조절하는 방법을 택하였는데, 8255의 A port를 출력으로 하고 B port를 입력으로 하였으며 C port를 조절 신호로 사용하여 전체 시스템을 동작시켰다.

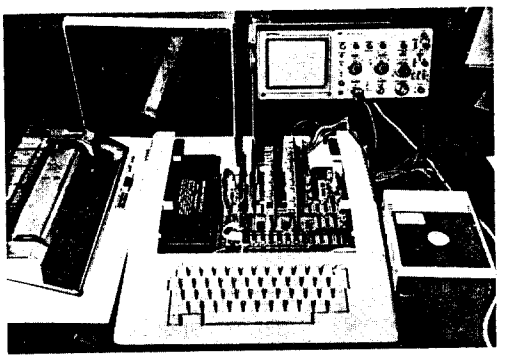


사진 1. 실험 시스템
Photo 1. Experimental system.

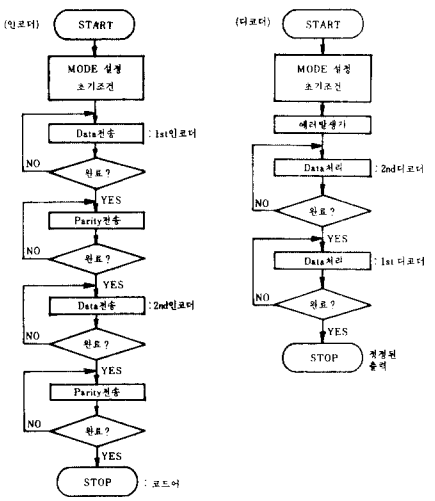


그림 6. 실험의 순서도
Fig. 6. Flow-chart of experiment.

실험에 대한 순서도는 다음의 그림 6과 같고 시스템은 사진 1에 나타나 있다.

이와 같은 방식을 이용하여 실험한 결과, 4개의 random 에러와 길이 8의 burst 에러를 넣었을 때 permanent에러 패턴이 아니면 완전히 정정함을 알았다.

그리고 한 data를 수행하는데 걸리는 시간은 플립플랍(f/f)과 같은 지연 소자가 여러개 있으므로 인코딩 시에는 340μs, 디코딩 시에는 1.1ms가 걸렸으나, 고차원의 코드를 1차원적으로 수행하는 것이 가능하였다.

V. 結 論

본 논문에서는 random 및 burst 에러를 동시에 정정할 수 있는 순환 곱셈 코드의 인코더와 디코더를 설계하고 제작하여 random과 burst 에러 정정 능력에 대하여 알아보았다.

실제로 (7,4)순환 hamming 코드와 (3,1)순환 코드를 인터레이스하여 간단한 순환 곱셈 코드를 제작하였고, 마이크로 computer와 인터페이스하여 실험을 수행해 본 결과로서 4개의 random 에러와 길이 8의 burst 에러를 정정할 수 있음을 알았다. 시스템에 대한 성능은 채널 에러가 10^{-3}일 때 9.66×10^{-10}이 되어 에러가 없는 것으로 간주될 수 있으므로, $10^{-5}\sim10^{-6}$ 정도일 때 10^{-10}의 성능을 갖는 일반 코드보다 좋음을 알 수 있었다.

한개의 data를 수행하는 데 걸리는 시간은 약 1.44ms가 걸렸고, 순환 곱셈 코드의 인코더와 디코더는 부코드의 인코더와 디코더에 지연 소자만 더 첨가하여 실현할 수 있으므로 간단함을 알 수 있었다.

순환 곱셈 코드는 주로 random과 burst 에러가 동시 발생하는 곳에 효과적으로 이용될 수 있으므로 교환 전화국과 같은 곳에 사용될 수 있다.

앞으로는 순환 곱셈 코드의 특성 중에서 코드율을 높이는 방향으로 연구가 계속되어져야 할 것이다.

References

[1] Lin, S., *An Introduction to Error Correcting Codes,* Prentice-Hall, 1970.

[2] Peterson, W.W., and E.J. Weldon, Jr., *Error Correcting Codes,* M.I.T. Press, 1972.

[3] Berlekamp, E.R., *Algebraic Coding* Theory, McGraw-Hill, 1968.

[4] Elias, P., "Error free coding," *IRE Trans. on Information Theory,* PGIT- 4, pp. 29-37, 1954.

[5] Burton, H.O., and E.J. Weldong, Jr., "Cyclic product codes," *IEEE Trans. on Information Theory,* IT-11, pp. 433- 440, 1965.

[6] Abramson, N.M., "Cascade decoding of cyclic product codes," *IEEE Trans. on* 398-402, 1968.

[7] Blahut, R.E., *Theory and Practice of Error Control Codes,* Addison-Wesley, 1983.

[8] Lee, Y.L., and M.C. Cheng, "Cyclic mappings of product codes," *IEEE Trans. on Information Theory,* IT-21, pp. 233-235, 1975.

[9] Goethals, J.M., "Factorization of cyclic codes," *IEEE Trans. on Information Theory,* IT-13, pp. 242-246, 1967.

[10] Fraileigh, J.B., *A First Course in Abstract Algebra,* Addison-Wesley, 1967.