論 文

다수논리 결정자를 이용한 리드뮬러코드의 시스템 설계

正會員 金 映 坤* 正會員 康 昌 彦**

Design of an Encoding-Decoding System using Majority-Logic Decodable Circuits of Reed-Muller Code

Young Gon KIM * *and* Chang Eon KANG ** *Regular Members*

요 약 본 논문은 리드-뮬러 코드의 인코더 및 디코더의 설계에 관하여 체계화 시켰으며, 이러한 코드는 [J/2]개 이하의 에러를 정정할 수 있다. 리드-뮬러 코드의 해석을 이용하여, (15, 11) 리드-뮬러 코드에 대하여 순환 성질을 이용하여 인코더를 설계하고, 다수 논리 결정자 방법에 의해 디코더를 설계하였다. 또, 이러한 코드에 대해 Weight 분포및 Performance를 구하였다. 마이크로 컴퓨터의 CPU와 Memory를 이용하기 위해 8255를 이용한 인터페이스 회로와 인코더및 디코더를 설계, 제작하여 직접 인터페이스 시켜 실험을 하였다. 실험한 결과 한개의 에러를 완벽히 정정하였고, 한개의 정보를 수행하는데 소요되는 시간은 약 70μsec 임을 알 수 있었다. 특히 (15, 11), (15, 5)리드-뮬러 코드는 채널 에러가 10^{-6}~10^{-4}인 곳에 적합함을 보였다.

ABSTRACT Using the Reed-Muller Codes, the encoder and decoder system has been designed and tested in this paper. The error correcting capability of this code is [J/2] or less and the error correcting procedure can be implemented easily by using simple logic circuitry. The encoding and decoding circuits are obtained by the cyclic property and for the (15, 11) Reed-Muller code majority-logic decoding is taken. The performance is measured in error probability and weight distribution. The encoder and decoder system has been designed, implemented and interfaced with the microcomputer by using the 8255 chip. Experimental results show that the system has single error-correcting capability and total execution time for a data is about 70 μsec. When the probability of channel error is 10^{-4}, the system using the (15, 11) Reed-Muller code works very good.

1. 서 론

디지틀 정보 전송 시스템에서 신뢰할 수 있는 정보를 얻기 위해서는 채널에서 발생한 에러를 정정할 수 있는 코드가 필요하다[1], [4]. 이러한 에러 정정 코드에는 많은 종류가 있으나 그 가운데 중요한 부류에 속하는 것으로 순환 코드가 있다. 이 순환 코드의 디코딩 방법에는 여러가지 있으나 다수 논리 결정자 방법은 신속성이 요구되는 정보 전송 시스템에서의 에러 조정에 효과적이고, 다른 디코딩 알고리즘보다 실행이 간단하고 디코딩 지연 시간이 짧은 장점이 있다[5]. 이러한

*, ** 延世大學校工科大學電子工學科
Dept. of Electronic Engineering Yonsei University, Seoul, 120 Korea.
論文番號 : 85-27(接受 : 1985. 5. 20)

디코딩은 1954년 Muller에 의해 다수 에러를 정정 시키는 코드가 개발되었고, Reed에 의해 다수 논리 결정자 알고리즘이 고안되었고, 그후, Kasami, Lin, Peterson에 의해 발달되어 Massey에 의해 일반화 되었다[9], [10], [11], [12].

본 논문은 다수 논리 결정자 코드의 일종인 Reed-Muller 코드에 대해서 순환성을 이용하여 인코더를 설계하고, 다수 결정자 디코딩 방법을 사용하여 [J/2]개 이하의 에러를 정정할 수 있는 디코더를 설계하였다. 이러한 코드에 필요한 기본이론을 해석하였으며, 이러한 이론에 기인하여 (15, 11) Reed-Muller 코드에 대하여 인코더 및 디코더를 실제적으로 설계하였다. 이 (15, 11) 코드의 weight 분포 및 performance를 나타내었고, 마지막으로 실험및 결론을 나타내었다.

2. 본 론

1) 이론해석

유클리드 기하학 코드에서 매개변수 중 $S=1$ 인 경우가 generalized Reed-Muller 코드이므로, 여기서 매개변수를 정의하면 다음과 같다.

코 드 어 : $n=q^m-1$
정 보 어 : $k=1+\binom{m}{1}+\cdots+\binom{m}{r}$
패리티체크비트의 수 : $n-K=1+\binom{m}{1}+\cdots+\binom{m}{m-r-1}$
최 소 거 리 : $d_{min}=q^{m-r}-1$
step의 수 : $L=r+1$
정 정 능 력 : $t=(d_{min}-1)/2$
패리티체크합의 수 : $j=d_{min}-1$

그리고 GRM(generalized Reed Muller) 코드의 중요한 다섯가지의 성질은,

- GRM 코드는 순환성 코드이다.
- r_{th}-order GRM 코드의 설계된 거리 $d=(q-R)q^{(m-Q-1)}$은 확장된 BCH코드의 부코드(sub-code)이다. 여기서 Q, R은 r을 $(q-1)$로 나누었을 때 몫과 나머지이다.
- r_{th}-order GRM코드의 이체(dual)는 $[(q-1)m-r-1]$st-order GRM 코드이다.
- zero$_{th}$-order GRM코드는 반복코드(repetion code)이다.
- $[(q-1)m-2]nd$-order GRM코드는 확장된 한개의 에러 정정 BCH 코드이다.

$GF(q^m)$에 대해 $n=q^m-1$의 길이(length)를 가진 r_{th}-order GRM는 순환코드이므로 생성다항식은

$$g(x)=\prod_{\substack{0\le j\le q^m-1\\ 0\le W(j)\le (q-1)(m-r-1)}}(x-\alpha^j) \qquad (2)$$

이고, 여기서 $W(j)$는 정수 j의 $q-ary$ 확장의 디지트의 합이고, α는 $GF(q^m)$에서 primitive element이다.

정보 block을 코드어로 인코딩 하기 위해서는 코드어의 첫 k 디지트는 정보 block과 똑같고, 마지막 $(n-k)$ 디지트를 패리티 체크 디지트라 하며 이러한 형태를 systematic 형태라 한다. 여기서 패리티 체크 디지트는 잡음이 있는 채널에서 데이타 전송시 에러가 발생 했을 때 검출 및 정정 하는데 사용된다. k 디지트의 정보 block의 인코딩은 $V(x)=r(x)+x^{n-K}m(x)$에 의해 수행된다. $r(x)$는 $x^{n-K}m(x)$를 생성 다항식 $g(x)$에 의해 나누어진 나머지이므로, 이것을 생성 다항식에 따른 궤환 연결한 shift register로 구성되는 dividing 회로에 의해 수행된다. 생성 다항식은

$$g(x)=1+g_1x+g_2x^2+\cdots+g_{n-K-1}x^{n-K-1}+x^{n-K} \qquad (3)$$

로 주어지고, 인코딩 회로는 그림1과 같다.

코드어가 채널에 의해 전송될 때, 잡음에 의해 영향을 받는다. Decoder는 수신된 벡터가 코드 벡터인지 아닌지를 검사해야 하므로, 이것은 수신된 벡터의 신드롬 계산에 의해 수행될 수 있다. 신드롬은 수신된 패리티 체크 디지트와 수신된 정보 디지트로 부터 계산된 패리티 체크 디지트의 modulo-2 합에 의해 얻어진다. 즉 신드롬은 수신된 벡터를 생성 다항식으로 나누어서 얻어진 나머지와 같다. 신드롬은

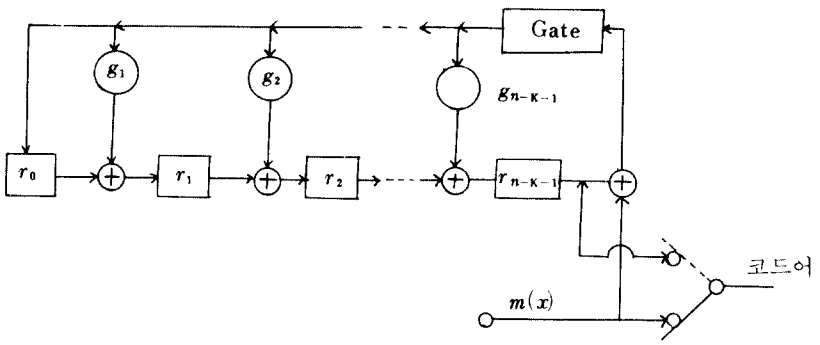


그림 1 $(n-k)$-단 shift register 인코더.
An $(n-k)$-stage shift register encoding circuit.

$$r(x) = P(x)\ g(x) + S(x) \tag{4}$$

이고 만일 신드롬이 0이라면 decoder는 수신된 vector를 전송된 코드 벡터처럼 받아들이고, 신드롬이 0이 아니면 수신된 벡터는 코드 벡터가 아니므로 에러가 검출된다. 신드롬 계산은 전송기에서 인코더와 같은 dividing회로에 의해 수행된다.

Systematic 형태에서 순환코드의 생성행렬은

$$x^{n-K+i} = q_i(x)\ g(x) + r_i(x),\quad i = 0,\ 1, \cdots, k-1 \tag{5}$$

이고, $r_i(x) = r_{i0} + r_{i1}x + r_{i2}x^2 + \cdots + r_{i,n-K-1}x^{n-K-1}$

이다. 그러므로 코드 다항식은

$$V_i(x) = r_i(x) + x^{n-K+i},\quad i = 0,\ 1, \cdots,\ k-1 \tag{6}$$

이다. 생성 행렬은

$$G = \begin{pmatrix} V_0 \\ V_1 \\ \vdots \\ V_{K-1} \end{pmatrix} = \begin{pmatrix} r_{00} & r_{01} \cdots & r_{0,n-K-1} & 1 & & \\ r_{10} & r_{11} \cdots & r_{1,n-K-1} & & 1 & \\ \vdots & \vdots & \vdots & & & \ddots \\ r_{K-1,0} & r_{K-1,1} \cdots & r_{K-1,n-K-1} & & & 1 \end{pmatrix}$$
$$= [P,\ I_K] \tag{7}$$

이고, $GH^T = 0$에서 패리티 체크 행렬은

$$H = \begin{pmatrix} 1 & r_{00} & r_{10} & \cdots & r_{K-1,0} \\ & 1 & r_{01} & r_{11} & \cdots & r_{K-1,1} \\ & & \vdots & \vdots & & \vdots \\ & & 1 & r_{0,n-K-1} & r_{1,n-K-1} \cdots & r_{K-1,n-K-1} \end{pmatrix}$$
$$= [I_{n-K},\ P^T] \tag{8}$$

이다. 에러 e에 관련된 신드롬은

$$S = (S_0,\ S_1,\ S_2, \cdots,\ S_{n-K-1})$$
$$= eH^T \tag{9}$$

이고, 여기서 H^T는 H의 전치행렬이다. 신드롬 비트는

$$\begin{aligned} S_0 &= e_0 + r_{00}e_{n-K} + \cdots + r_{K-1,0}e_{n-1} \\ S_1 &= e_1 + r_{01}e_{n-K} + \cdots + r_{K-1,1}e_{n-1} \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ S_{n-K-1} &= e_{n-K-1} + r_{0,n-K-1}e_{n-K} + \cdots + r_{K-1,K-1}e_{n-1} \end{aligned}$$

이고, 여기서 r_{ij}는 0이나 1이 된다.

신드롬 비트의 합은

$$A = a_0S_0 + a_1S_1 + \cdots + a_{n-K-1}S_{n-K-1} \tag{11}$$

이고, A는 에러 비트의 합이므로

$$A = b_0e_0 + b_1e_1 + \cdots + b_{n-1}S_{n-1} \tag{12}$$

이고, 여기서 a_i와 b_i는 0이나 1이 된다. 집합 E 속의 모든 에러 디지트 e_u이 모든 체크합 A_J에 의해 체크되고, 다른 에러 디지트가 하나의 체크합 보다 더 많은 것에 의해 체크되지 않는다면, J패리티 체크합 A_1, A_2, $\cdots A_J$의 집합은 집합 E에 의해 orthogonal 되었다고 한다. e_u에 대해 각 체크합 orthogonal 형태는

$$A_J = \sum_{i,\ l=k} e_k + e_u \tag{13}$$

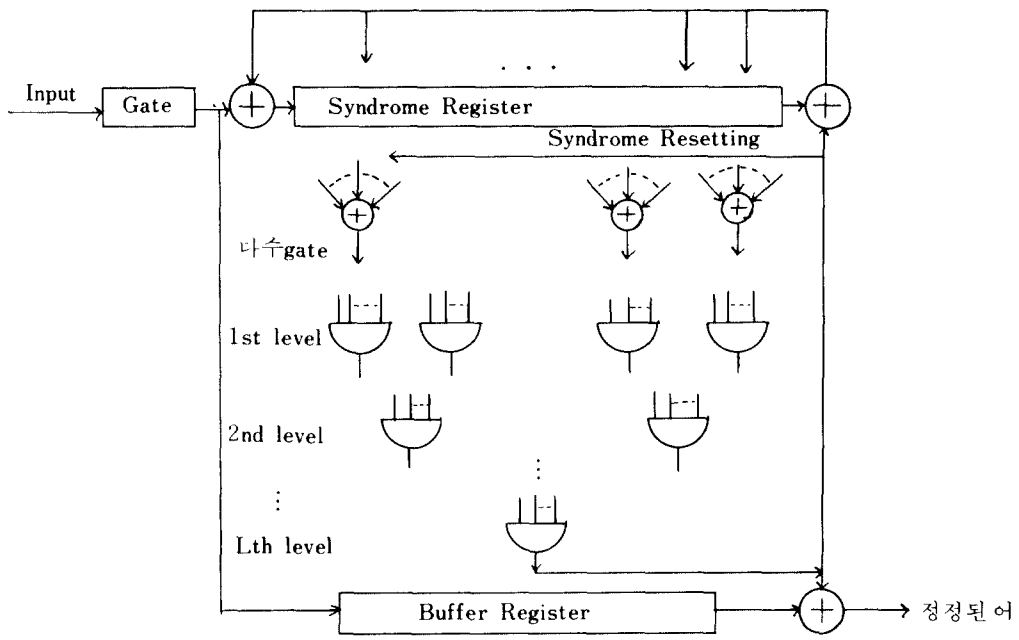


그림 2 L-단계 Orthogonalizable 코드에 대한 다수논리 디코더
A Majority-logic decoder for an L-step orthogonalizable code.

이고, 만일 $i, l \neq k$에서 합 A_j 속의 모든 에러 디지트 e_k가 0이라면 $e_u = A_j$이다. 만일 orthogonalization의 L-step이 에러 디지트에 의해 디코딩이 요구된다면, 이러한 코드를 L-step orthogonalizable 하다고 하고, J 패리티 체크합이 $d_{min}-1$과 같다고 하면 completely orthogonlizable 하다고 한다. 그러므로 GRM 코드도 $J = d_{min}-1$이므로 completely orthogonlizable 하므로, L-step 다수논리 decoder에 의해 수행된다. 다수gate의 추정되는 값은, 만일 입력의 1의 갯수가 전체의 반보다 많으면 출력이 1이고, 그렇지 않은 경우는 0이 된다. L-step decoder 는 그림 2와 같다.

2) (15, 11) 리드-뮬러 코드의 Encoder 및 Decoder 설계

$q = 2$, $m = 4$로 결정 했을 때 식(1)에 의해 코드어는 15비트이고 정보어는 11비트가 되므로, (15, 11) 리드-뮬러 코드는 order가 2이고, $d_{min} = 3$이고, 패리티 체크 합은 2이고, 다수 논리 단계는 3이 된다. 생성 다항식은 식(2)에 의해, 디지트의 합 $W(j)$는 $q=2$, $m=4$, $r=2$에서 $0 < W(j) \leq 1$이고, $0 \leq j \leq 15$이므로 binary 0000로 부터 1111까지에서 $W(j)$가 1인 것은 $j = 1, 2, 4, 8$이 된다. 그러므로 생성다항식은

$$g(x) = (x+\alpha)(x+\alpha^2)(x+\alpha^4)(x+\alpha^8) = 1 + x + x^4 \quad (14)$$

이 된다. 생성 다항식 $g(x)$에 의해 encoder 는 그림 3과 같다.

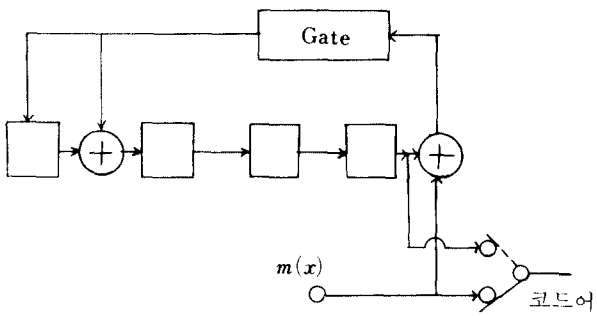


그림 3 (15, 11) 리드-뮬러코드의 인코더
An Encoder for (15, 11) Reed-Muller code.

생성 행렬은 x^{n-K+i}를 $i=0, 1, \cdots, k-1$ 에 대해서 생성 다항식 $g(x)$로 나누었을 때 나머지에 의해, (15, 11) 리드-뮬러 코드는 $n-k=4$ 이므로, x^{4+i}를 생성 다항식 $g(x)$로 나누면, 즉 $x^{4+i}=q_i(x)\,g(x)+r_i(x)$ 이고, 여기서 $i=0, 1, \cdots, 10$이 된다.

가. $i=0$ 일때 $x^4=1\cdot g(x)+(x+1)$

나. $i=1$ 일때 $x^5=x\cdot g(x)+(x^2+x)$

다. $i=2$ 일때 $x^6=x^2\cdot g(x)+(x^3+x^2)$

라. $i=3$ 일때 $x^7=(x^3+1)\cdot g(x)+(x^3+x^2+1)$

마. $i=4$ 일때 $x^8=(x^4+x+1)\cdot g(x)+(x^2+1)$

바. $i=5$ 일때 $x^9=(x^5+x^2+x)\cdot g(x)+(x^3+x)$

사. $i=6$ 일때 $x^{10}=(x^6+x^3+x^2+1)\cdot g(x)+(x^2+x+1)$

아. $i=7$ 일때 $x^{11}=(x^7+x^4+x^3+x)\cdot g(x)+(x^3+x^2+x)$ (15)

자. $i=8$ 일때 $x^{12}=(x^8+x^5+x^4+x^2+1)\cdot g(x)+(x^3+x^2+x+1)$

차. $i=9$ 일때 $x^{13}=(x^9+x^6+x^5+x^3+x+1)\cdot g(x)+(x^3+x^2+1)$

카. $i=10$일때 $x^{14}=(x^{10}+x^7+x^6+x^4+x^2+x+1)\,g(x)+(x^3+1)$

이 되므로 생성 행렬은

$$G=\begin{pmatrix}
1&1&0&0&1&&&&&&&\\
0&1&1&0&&1&&&&&&\\
0&0&1&1&&&1&&&0&&\\
1&1&0&1&&&&1&&&&\\
1&0&1&0&&&&&1&&&\\
0&1&0&1&&&&&&1&&\\
1&1&1&0&&&&&&&1&\\
0&1&1&1&&&&0&&1&&\\
1&1&1&1&&&&&&&1&\\
1&0&1&1&&&&&&&&1\\
1&0&0&1&&&&&&&&&1
\end{pmatrix} \quad (16)$$

이 되고, $G=[P, I_K]$에서 $H=[I_{n-K}\ P^T]$에 의해 패리티 체크 행렬은

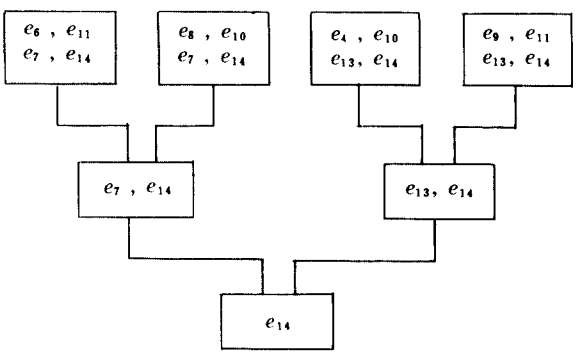
$$H=\begin{pmatrix}
1&0&0&0&1&0&0&1&1&0&1&0&1&1&1\\
0&1&0&0&1&1&0&1&0&1&1&1&1&0&0\\
0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1
\end{pmatrix} \quad (17)$$

이 된다. 에러와 패리티 체크 행렬에 의해 신드롬 비트는 $S=eH^T$에 의해 $S=(S_0, S_1, \cdots S_{14})$ 이고 에러를 $e=(e_0, e_1, \cdots, e_{14})$이라 두면 신드롬 비트는 아래와 같다.

$$\begin{aligned}
S_0&=e_0+e_4+e_7+e_8+e_{10}+e_{12}+e_{13}+e_{14}\\
S_1&=e_1+e_4+e_5+e_7+e_9+e_{10}+e_{11}+e_{12}\\
S_2&=e_2+e_5+e_6+e_8+e_{10}+e_{11}+e_{12}+e_{13}\\
S_3&=e_3+e_6+e_7+e_9+e_{11}+e_{12}+e_{13}+e_{14}
\end{aligned} \quad (18)$$

$J=2$이므로 각 gate의 입력은 2개가 되고, 첫 단계의 입력은 8개이고, 4개에 의해 orthogonal 되고, 두 번째 단계의 입력은 4개이고, 2개에 의해 orthogonal되고, 세 번째 단계의 입력은 2개이고 1개(e_{14})에 의해 orthogonal되게 체크합을 형성해야 한다. (15, 11) 리드-뮬러코드의 체크합의 관계는 아래 표와 같다.

표 1 (15, 11) 리드-뮬러 코드의 체크합의 관계
Relation of check sums for (15, 11) RM code.



위의 표에 의해 신드롬과 에러 비트의 합인 패리티 체크합은

가. $E_1=\{e_6, e_{11}, e_7, e_{14}\}$

$A_{11}=S_3=e_3+e_6+e_7+e_9+e_{11}+e_{12}+e_{13}+e_{14}$

$A_{12}=S_0\oplus S_2=e_0+e_2+e_4+e_5+e_6+e_7+e_{11}+e_{14}$

나. $E_2=\{e_8, e_{10}, e_7, e_{14}\}$

$A_{21}=S_0=e_0+e_4+e_7+e_8+e_{10}+e_{12}+e_{13}+e_{14}$

$A_{22}=S_0\oplus S_3=e_2+e_3+e_5+e_7+e_8+e_9+e_{10}+e_{14}$ (19)

다. $E_3=\{e_4, e_{10}, e_{13}, e_{14}\}$

$A_{31}=S_0=e_0+e_4+e_7+e_8+e_{10}+e_{12}+e_{13}+e_{14}$

$A_{32}=S_1\oplus S_3=e_1+e_3+e_4+e_5+e_6+e_{10}+e_{13}+e_{14}$

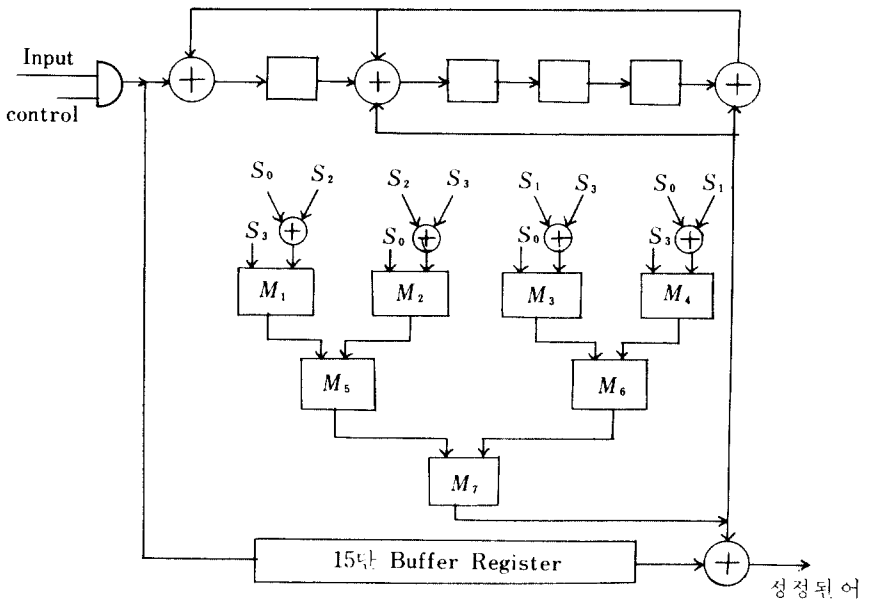


그림 4 (15, 11) 리드-뮬러 코드의 디코더
An Decoder for (15, 11) RM code.

라. $E_4=\{e_9, e_{11}, e_{13}, e_{14}\}$

$A_{41}=S_3 = e_3+e_6+e_7+e_9+e_{11}+e_{12}+e_{13}+e_{14}$

$A_{42}=S_0 \oplus S_1=e_0+e_1+e_5+e_8+e_9+e_{11}+e_{13}+e_{14}$

이 된다. (15, 11) 리드-뮬러 코드는 3단계 orthogonalizable에 대한 다수 논리 decoder 는 그림 4와 같다.

3) Weight 분포 및 Performance

최소 weight 의 갯수는 d_{min}이 되어 코드의 중요한 매개변수가 된다. 리드-뮬러 코드의 최소 weight 의 갯수는

$$N_{2^{h-1}}=\prod_{i=0}^{h-1}\frac{(2^m-2^i)}{(2^h-2^i)} \qquad (20)$$

이고, 여기서 $h=m-r$이 된다. (15, 11) 리드-뮬러 코드의 최소 weight 갯수는 식(20)에 의해 35이다. weight 분포는 코드어 발생에서 계산된 weight 수를 구한 것으로 표 2 에 나타 내었다.

표 2 (15, 11) 리드-뮬러코드의 Weight 분포
Weight distribution for (15, 11) RM code.

Weight 수	Weight 수의 갯수
0	1
1	0
2	0
3	35
4	105
5	168
6	280
7	435
8	435
9	280
10	168
11	105
12	35
13	0
14	0
15	1

표 2에서 d_{min}이 3임을 알 수 있고, weight 분포가 완전 대칭임을 알 수 있다.

채널을 BSC이라 하고, 각 비트의 에러가 서로 독립적으로 발생된다고 하면, 리드-뮬러 코드는 $t=(d_{min}-1)/2$이므로 완전한 코드가 되어, 에러를 정정 못할 확률은

$$P_M = \sum_{m=t+1}^{n} \binom{n}{m} P^m (1-P)^{n-m} \quad (21)$$

이 되고, 여기서 P는 채널의 에러이다. (15, 11) 리드-뮬러 코드는 $d_{min}=3$ 이고, $t=1$ 이므로 완전한 코드가 되어 P_M은

$$P_M = \sum_{m=2}^{15} \binom{15}{m} P^m (1-P)^{15-m} \quad (22)$$

이고, (15, 5) 리드-뮬러 코드는 $d_{min}=7$ 이고, $t=3$ 이므로 완전한 코드가 되어 P_M은

$$P_M = \sum_{m=4}^{15} \binom{15}{m} P^m (1-P)^{15-m} \quad (23)$$

이 된다. 위의 두 식을 P와 P_M의 관계를 나타내면 아래 표와 같다.

표 3 $n=15$ RM코드의 performance
Performance for $n=15$ RM code.

DATA	PM-11	PM-5
.1	.450957	.0555556
.05	.170953	5.46726E-03
.04	.11911	2.44969E-03
.01	9.62978E-03	1.24976E-05
5E-03	2.51377E-03	8.1636E-07
1E-03	1.04094E-04	1.35304E-09
8E-04	6.67357E-05	5.5518E-10
5E-04	2.61365E-05	8.4938E-11
1E-04	1.04909E-06	1.3638E-13
5E-05	2.62386E-07	8.52748E-15
1E-05	1.04991E-08	1.36488E-17
5E-06	2.62489E-09	8.53089E-19
3E-06	9.44977E-10	1.10562E-19
1E-06	1.04999E-10	1.36499E-21
5E-07	2.62499E-11	8.53123E-23
3E-07	9.44999E-12	1.10565E-23
1E-07	1.05E-12	1.365E-25

표 3을 그래프로 나타내면 그림 5와 같다.

위의 표와 그래프에서 본 것과 같이 (15, 11) 리드-뮬러 코드는 채널 에러가 10^{-6}인 곳에 적합

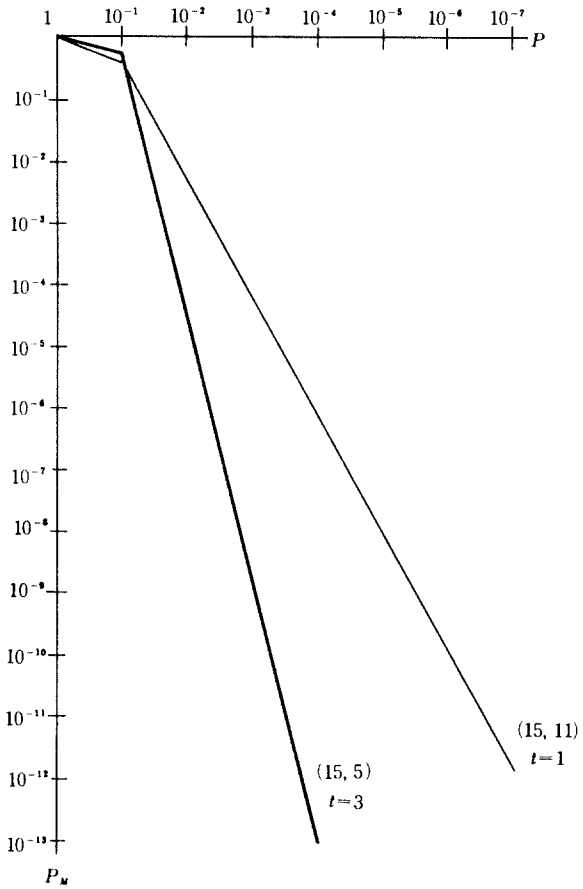


그림 5 $n=15$ 리드-뮬러 코드의 performance의 그래프
The graph of performance for $n=15$ RM code.

하고, (15, 5) 리드-뮬러 코드는 10^{-4}인 곳에 적합함을 알 수 있다.

4) 실험 및 결과고찰

Apple-Ⅱ의 CPU와 Memory를 이용하기 위해 8255를 이용한 Interface회로와 Encoder 및

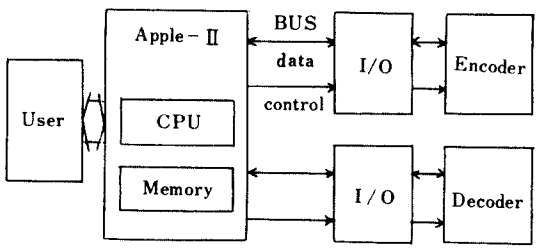


그림 6 실험계통도
Experimental Modeling.

Decoder를 설계, 제작하여 직접 인터페이스 시켜 실험을 하였다. 전체적인 실험 계통도는 그림 6과 같다.

실험 방법을 순서도로 나타내면 아래그림 7과 같다.

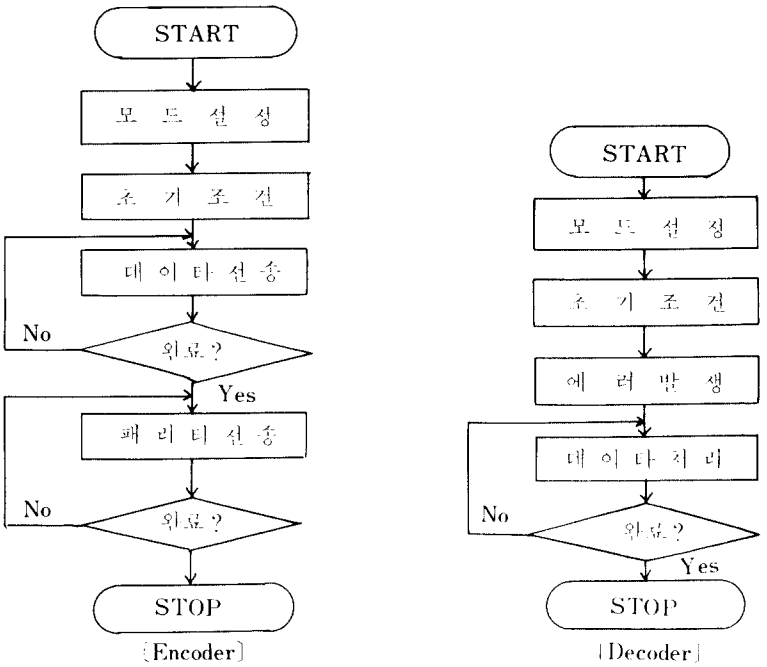


그림 7 실험의 순서도
Experimental flow chart.

그리고, 본 실험에 사용된 timing diagram은 그림 8과 같다.

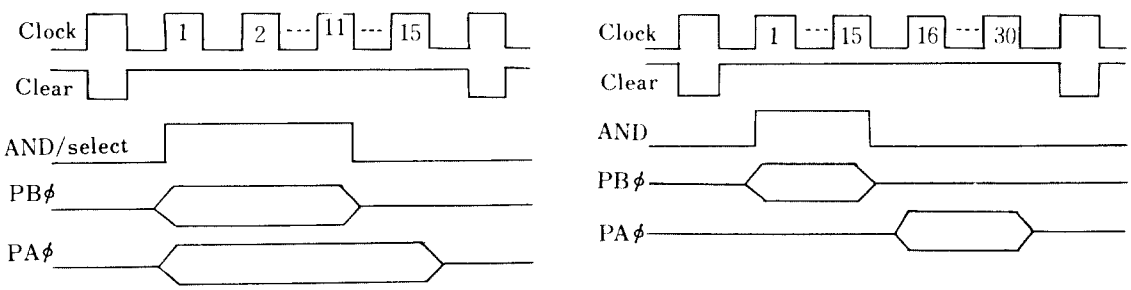


그림 8 실험에 사용된 timing diagram
Experimental timing diagram.

본 논문에서, 8255 인터페이스와 Encoder 및 Decoder를 설계, 제작하여 마이크로 컴퓨터를 사용하여 실험함으로써 한 개의 에러를 완벽히 정정함을 알 수 있었다. 하나의 데이터를 수행하는데 걸리는 시간은 하나의 clear에서 다음clear 사이의 시간을 오실로스코우프로 측정해 본 결과 70 μsec가 소요됨을 알 수 있었다.

3. 결 론

본 논문에서 리드-뮬러 코드의 인코더 및 디코더 설계에 관하여 체계화 시켰으며, 실제로 (15,

11) 리드-뮬러 코드의 Encoder 및Decoder 를 설계, 제작하여서 마이크로컴퓨터를 이용하여 실험한 결과, 한 개의 에러를 완벽히 정정함을 알아 보았다. 또 (15, 11) 리드-뮬러 코드에 대하여 weight분포및 performance를 구함으로써, weight 분포가 완전 대칭임과 (15, 11) 리드-뮬러 코드는 $P=10^{-6}$, (15, 5) 리드-뮬러는 $P=10^{-4}$에 적합함을 알 수 있었다. 실험결과 하나의 데이타를 수행하는데 걸리는 시간은 약 70 μsec가 되어 복호지연 시간이 짧음을 알 수 있었다. 리드-뮬러 코드는 순환 성질을 갖고있어 Encoder및 신드롬계산 회로를 포함한 Decoder를 쉽게 구성할 수 있다. 리드-뮬러 코드는 에러 발생에 대한 [J/2]개 이하의 에러를 완벽히 정정할 수 있는 능력을 갖고 있으므로 데이터 전송, 위성통신, memory device, ARQ(Automatic Repeat Request)등에 많이 이용될 수 있다.

參 考 文 獻

(1) Lin, S., "An Introduction to error correcting Codes," Prentice-Hall, 1970.

(2) Peterson, W.W., Error Correcting Codes, The M.I.T. press, 1972

(3) Blahut, R.E., Theory and Practice of Error Control codes, Addison-Wesley, 1983.

(4) Proakis, J.G., Digital communications, McGraw-Hill, 1983.

(5) Massey, J.L., Threshold Decoding, The M.I.T. press, 1963.

(6) Chen, C.L., "Note on Majority-Logic Decoding of Finite Geometry codes," IEEE Trans. on Information Theory, vol. IT-18, pp. 539~541, 1972.

(7) Gore, W.C., "The Equivalence of L-step orthogonalization and a Reed Decoding Procedure," IEEE Trans. on Information Theory, vol. IT-15, pp. 184~186, 1969.

(8) Kasami, T., and N. Tokura, "On the weight structure of Reed-Muller codes," IEEE Trans. on Information Theory, vol. IT-16, pp. 752~759, 1970.

(9) Goethals, J.M., and P. Delsarte, "on a class of Majority-Logic Decodable cyclic codes," IEEE Trans. on Information Theory vol. IT-14, pp. 182~188, 1968.

(10) Kasami, T., S. Lin, and W. W. peterson, "New Generalizations of the Reed-Muller codes," IEEE Trans. on Information Theory, vol. IT-14, pp, 189~198, 1968.

(11) Reed, I.S., "A class of Multiple-Error-Correcting codes and the Decoding scheme," I.R.E. Trans. Information Theory, PGIT-4, pp, 38~49, 1954.

(12) Muller, D.E., "Application of Boolean Algebra to switching circuit Design and to Error Detection," I.R.E. Trans. Electron. computer, EG-3, pp. 6~12, 1954.

(13) Rudolph, L.D., "A class of Majority Logic Decodable codes." IEEE Trans. Information Theory, IT-13, pp. 305~307, 1967.

金 映 坤(Young Gon KIM) 正會員
1958年 6月16日生
1983年 2月：慶北大學校電子工學科卒業
1985年 2月：延世大學校大學院電子工學科卒業 (工學碩士)
1985年 6月：韓國電氣通信公社(研究員)

康 昌 彦(Chang Eon KANG) 正會員
1938年 8月26日生
1960年：延世大學校電氣工學科(工學士)
1965年：延世大學校大學院電氣工學科(工學碩士)
1969年：美國미시간주립대학교大學院電氣工學科(工學碩士)
1973年：美國미시간주립대학교大學院電氣工學科(工學博士)
1967年~1973年：美國미시간주립대학교工業研究所先任研究員
1973年~1981年：美國노던일리노이대학교電氣工學科助教授, 副教授
1982年~現在：延世大學校電子工學科 教授, 本學會研究調査委員會 委員長