

# 단일변수 변환 행렬을 이용한 GRM 상수 생성 방법

이철우\*, 김영건\*\*, 박동영\*\*\*, 강성수\*\*\*\*, 김홍수\*

\*인하대학교 전자공학과 \*\*안산전문대학 전산정보처리과

\*\*\*원주전문대학 전자통신과 \*\*\*\*부천대학 전자계산과

## The method to produce GRM coefficient using single transform matrix

Chol U Lee<sup>\*</sup>, Young Gun Kim<sup>\*\*</sup>, Dong Young Park<sup>\*\*\*</sup>

Sung Soo Kang<sup>\*\*\*\*</sup>, Heung-Soo Kim<sup>\*</sup>

\* Dept. of Electronics Eng. In-Ha Univ.

\*\* Dept. of Computer & Information processing Ansan Jr. College

\*\*\* Dept. of Electronics Communication Wonju Jr. College

\*\*\*\* Dept. of Computer Science Puchon Jr. College

E-mail : martin1@shinbiro.com

FAX : 82-32-860-7413

### Abstract

This paper propose the method to produce GRM(Generalized Reed-Muller)expansion. The general method to obtain GRM expansion coefficient for p valued n variable is derivation of single variable transform matrix and expand it n times using Kronecker product. In this case the size of matrix increases depending on the augmentation of variables. In this paper we propose the simple algorithm to produce GRM coefficient using a single variable transform matrix.

### 1. 서론

현재 사용되고 있는 논리회로의 체계는 부울함수를 기초로 한 2진 논리로 구성되어 있다. 그러나, 2진 논리회로가 갖는 회로 복잡성, 단자수 제한 등의 문제점으로 인하여 부울체의 확장인 유한체(Galois Field)를 기초로 한 다치논리에 대한 연구가 활발히 진행중이다.[1-2] Galois 체는  $p$ 를 소수,  $m$ 을 양의 정수라 할 때  $p^m$ 개의 원소로 구성된 체를 말한다. Galois 체에서는 2진 논리에서와 마찬가지로 교환법칙, 결합법칙, 분배법칙이 성립

한다.[7] 다치논리 함수는 일반적으로 입력 값의 조합에 의한 출력 값이 주어지는 진리치표를 일반화한 연산영역과 입력변수를 함수적으로 표현한 함수영역에서 해석이 가능하며 이를 영역사이의 변환은 RM(Reed-Muller)변환에 의해서 이루어진다. RM을 이용하는 이유는 소자수, 결선수 감소로 인한 경제성과 검사가 용이하다는 점에서 많이 사용되고 있다.[3-5] RM상수를 구하는 방법은 변환 행렬을 이용하여 구하는 방법, 결정도를 이용하여 구하는 방법 등 여러 가지가 있다. 그러나, 그 중에서도 변환행렬을 이용해서 구하는 방법이 많이 이용되고 있다. 그러나 변수가 증가할 수록 변환행렬의 차수가 커진다는 단점을 가지고 있다. 그러므로 본 논문에서는 2치 4변수 함수를 예를 들어 본 논문에서 제안한 방법을 적용하여 단일 변수 변환 행렬만을 이용하여 GRM 상수를 구하였다.

### 2. RM 상수와 GRM 상수 생성

#### 2.1 RM 상수의 생성

단일변수 부울 함수식을 단일변수 RM 전개식으로 변환할 때의 계수선택행렬을 나타내기 위해 단일변수 부울 함수식을 이용하면 다음과 같다.

$$f(x_1) = d_0 \bar{x}_1 + d_1 x_1 \quad (1)$$

여기서  $d_0, d_1$  는 0 또는 1의 진리값을 갖는다.

식(1)은 드모르강의 법칙에 의해

$$f(x_1) = \overline{d_0 x_1} \cdot \overline{d_1 x_1} \quad (2)$$

와 같이 변환되며 모듈러-2 연산에서는  $\bar{a} = a \oplus 1$  또는  $a \oplus 1 = \bar{a}$ 이므로 식(2)를 다시 모듈러-2 연산식으로 전개하면 다음과 같다.

$$\begin{aligned} f(x_1) &= (d_0(x_1 \oplus 1) \oplus 1) \cdot (d_1 x_1 \oplus 1) \oplus 1 \\ &= d_0 d_1 x_1 \oplus d_0 d_1 x_1 \oplus d_0 x_1 \oplus d_0 \oplus d_1 x_1 \end{aligned} \quad (3)$$

$$= d_0 \oplus (d_0 \oplus d_1) x_1$$

$$f(x_1) = c_0 \oplus c_1 x_1 \quad \text{over GF}(2) \quad (4)$$

식(3)과 식(4)의 관계를 행렬로 나타내면 다음과 같다.

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \end{bmatrix} \quad \text{over GF}(2) \quad (5)$$

이 때, 단 변수에서 연산영역의 계수  $d_i$ 와 함수 영역의 계수  $c_i$ 와의 관계는 식(5)에서 나타난 것처럼 다음의 행렬 형태로 나타나게 되고 이를 변환 행렬(transform matrix)이라 한다.

$$T_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad T_0 = [1] \quad (6)$$

여기서, n변수에 대한 행렬은 다음과 같다.

$$T_n = \begin{bmatrix} T_{n-1} & 0 \\ T_{n-1} & T_{n-1} \end{bmatrix} \quad (7)$$

식(7)은 변수의 개수가 적을 때에는 변환 행렬을 구하는 것이 용이하지만 변수의 숫자가 증가함에 따라 행렬의 차수가 증가하므로 연산하기에 많은 어려움이 생긴다.

## 2.2 GRM상수의 생성

RM 전개식에 의하여 표현된 함수는 다치논리의 함수영역에서 유일한 것이 아니다. 만약 입력 변수  $x_i$ 를  $\bar{x}_i$ 로 대치한다면 다른 형태의 정규화된

형식으로 표현된다. 이와 같이 n개의 입력변수에 대해  $2^n$ 개의 서로 다른 입력형태가 만들어지며 이에 대한 RM상수를 구하는 것이 GRM 변환이다. GRM 상수 값의 결정은 보통 연산영역에서 극수에 대한 GRM 상수를 구하는 방법이 있지만 극수가 0인 fixed polarity를 구하고, 극수를 확장 시켜 GRM 상수를 구하는 것이 최적화된 극수를 구하는데 더욱 효과적이다.

단일변수인 경우 GRM 형식은 다음과 같다.

$$f(x_1') = c_0 \oplus c_1 x_1' \quad (8)$$

$x_1'$ 은  $x_1$ 의 입력형태가  $x_1$ 이거나 또는  $\bar{x}_1$ 인 2가지 형태 중에 하나를 의미한다.

만약  $x_1' = x_1$ 이고 GRM 함수의 일반식이  $f(x_1) = a_0 \oplus a_1 x_1$ 이라면 식(8)은 다음과 같이 표현된다.

$$f(x_1') = c_0 \oplus c_1 x_1 = a_0 \oplus a_1 x_1 \quad (9)$$

식(9)은 극수가 0이며 변환행렬로 표현하면 다음과 같다.

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} \quad \text{또는 } a = Z_0 c \quad (10)$$

만약  $x_1' = \bar{x}_1$ 라면 모듈러-2 성질에 의하여  $\bar{x}_1 = x_1 \oplus 1$ 로 다음과 같이 표현된다.

$$\begin{aligned} f(x_1') &= c_0 \oplus c_1 \bar{x}_1 = c_0 \oplus c_1 (x_1 \oplus 1) \\ &= a_0 \oplus a_1 x_1 \end{aligned} \quad (11)$$

극수가 1인 함수의 변환 행렬을 표현하면 다음과 같다.

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} \quad \text{또는 } a = Z_1 c \quad (12)$$

단일변수에 대한 GRM변환행렬  $Z_0$  와  $Z_1$ 을 이용하여 n변수에 대한 임의의 극수 i의 GRM 변환행렬에 대한 일반식은 다음과 같으며 변환행렬은  $2^n \times 2^n$ 의 차수를 갖는 행렬이며 단일변수에 대한

kronecker곱에 의하여 생성된다.

$$Z_{n,n-1,n-2,\dots,1} = Z_{\langle i \rangle} = Z_n * Z_{n-1} * \dots * Z_1 \quad (13)$$

여기서  $Z_n, Z_{n-1}, \dots, Z_1$ 은 0 또는 1이다.

### 3. 단일변수를 이용한 GRM 상수의 생성

앞 절에서 나타난바와 같이 GRM 상수를 구하는 데 있어서 변수가 많아지면 행렬의 차수가 커진다는 것을 알 수 있다. 예를 들어, 2치 4변수 함수에서 GRM 상수를 구하려면  $16 \times 16$  행렬이 필요하다. 그러나, 이보다 더욱 간단하게  $2 \times 2$ , 혹은  $4 \times 4$  행렬을 이용하여 GRM 상수를 구하는 방법에 대해 알아본다.

먼저, GRM 상수를 구하는 방법은 변환행렬  $Z_1$ 을 이용한다. 2치 4변수 함수의 k-map은 다음과 같이 나타난다.

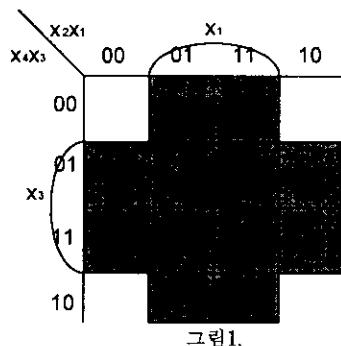


그림1.

단일 변수 변환 행렬을 이용하여 GRM 상수를 구하는 방법은 알고자 하는 극수가 어떤 변수에 해당하는지를 알고 그 변수들의 보수를 취한 후에 해당 변수에만 단일 변수 변환 행렬을 곱한다. 2진논리에서 각 변수가 가질 수 있는 값의 조합은 모두 4가지이다. 그 중 어떤 경우에도 다시 0이 되는 00을 제외하면 3가지이다. 이들과 변환행렬의 연산은 다음과 같다.

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

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

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

예를 들어 2치 4변수 함수에서 극수가  $p=6$ 이라면 k-map은 그림1.에서와 같이 빛금친 부분으로 나타난다.

그림1.과 같이  $p=6$ 이면 4개의 변수 중에  $X_3, X_2$ 가 보수가 취해졌다는 것을 알 수 있다. 그러면 먼저  $X_1$ 에 관해서 0과 1일 때를 구분해서  $2 \times 2$  행렬 연산을 한다. 그리고 그 결과의 행렬에 다시  $X_4$ 에 대하여 다시 행렬연산을 한다. 다음 예제1.에서는  $16 \times 16$ 인 변환행렬을 구하여 연산하는 것 보다 더욱 간단하게 GRM 상수를 구할 수 있다는 것을 예제를 통해서 두 개의 알고리즘을 비교해본다.

#### 예제1.

다음과 같이 함수영역의 함수가 있다. 이 함수의 극수  $p=6$ 을 구하여라.

$$f(x_4, x_3, x_2, x_1) = (1001111101010111)^T$$

I) 기존의 방법을 사용하여 원하는 극수를 구하려면 다음과 같이  $16 \times 16$  행렬을 구해서 계수 행렬과 연결해야 한다.

변환행렬을 구하는 것은 극수를 2진으로 나타내고 해당되는 행렬들을 kronecker곱을 취한다. 이 때, 변환행렬  $Z$ 는  $Z_0 * Z_1 * Z_2 * Z_3$ 의 kronecker곱으로 나타낼 수 있다.

$$Z = Z_0 * Z_1 * Z_2 * Z_3$$

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

$$= \begin{array}{|c|} \hline 1010101000000000 \\ 0101010100000000 \\ 0010001000000000 \\ 0001000100000000 \\ 0000101000000000 \\ 0000010100000000 \\ 0000001000000000 \\ 0000000100000000 \\ 0000000010000000 \\ 0000000001010101 \\ 00000000001010101 \\ 00000000000100010 \\ 00000000000010001 \\ 00000000000001010 \\ 00000000000000101 \\ 00000000000000010 \\ 00000000000000001 \\ \hline \end{array}$$

위의 행렬에 원래의 계수 행렬을 곱하면 원하는 극수  $p=6$ 의 GRM 상수를 다음과 같이 얻을 수 있다.

$$f(x_4, \bar{x}_3, \bar{x}_2, x_1) = (1110001110111010)^T$$

II) 단일변수 변환행렬을 이용하면 먼저  $X_2$ 가 0일 때와 1일 때를 구분하여 단일변수 변환행렬  $T$ 와 연산한다. 극수가 0인 계수의 행렬은 다음과 같다.

1	1	1	0
0	0	1	1
1	0	1	1
0	0	1	0

p=0일 때 GRM 상수

1	1	0	1
0	0	1	1
1	0	1	1
1	0	0	1

p=2일 때 GRM 상수

다음 과정은  $X_3$ 가 0일 때와 1일 때를 구분해서 변환행렬  $T$ 와 연산한다.

1	0	1	0
1	1	1	1
0	1	1	1
0	1	1	0

p=2일 때 GRM 상수

1	1	1	0
0	0	1	1
1	0	1	1
0	0	1	0

p=6일 때 GRM 상수

이와 같이, 복잡한 행렬 연산을 거치지 않고 단순하게 직감적으로 할 수 있는 행렬 연산으로 극수를 구할 수 있다. 간단하게 표현해서 구하고자 하는 극수에 해당되는 변수들이 0과 1일 때를 구분하여 2개씩 짝을 맞춘 후에 단일 변수 변환 행렬과 연산하면 된다. 이처럼 두 변수가 가능한 조합의 수는 00, 01, 11, 10이고 이 중 00을 제외한 나머지들의 행렬 연산 값은 각각 11, 01, 10이므로 직감적으로 원하는 극수를 구할 수 있다.

#### 4. 결론

본 논문은 단일변수 변환행렬을 이용하여 GRM 상수를 구하는 방법을 제시하였다. 변수의 개수에 따라  $2^n X 2^n$ 로 차수가 증가하는 행렬을 이용하여 GRM 상수를 구하는 기존의 방법보다 단일 변수

변환 행렬을 사용하면 극수를 구하는 것이 단순한  $2X2$  행렬의 연산이므로 더욱 간단해짐을 예제를 통해 증명했다. 추후의 연구과제로는 단일 변수 변환 행렬을 2-치에는 물론이고 3-치 혹은  $p$ -치로 확장시키는 것이 있다.

#### 참고문헌

- [1] W.Besslich, "Efficient computer method for ExOR logic design," IEE Proceedings, vol 130, Pt. E, No. 6, pp.203-206, November 1983
- [2] D.H.Green, I.S.Taylor, "Multiple-valued switching circuit design by means of generalized Reed-Muller expansions," Digital Processes, 2, pp.63-81, 1976
- [3] Qinhua Hong, Benchu Fei, Haomin Wu, Markek A. Perkowski, Nan Zhuang, "Fast Synthesis for Ternary Reed-Muller Expansion," IEEE Proc. of Symposium on Multiple-Valued Logic, Sacramento, California, pp.14-16, May 1993
- [4] X.Wu, X.Chen, S.L.Hurst, "Mapping of Reed-Muller coefficients and the minimisation of exclusive OR-switching functions," IEE PROC., Vol. 129, Pt. E, No. 1, January 1982
- [5] D.H.Green, I.S.Taylor, "Modular representation of multiple-valued logic systems," IEE Proceedings, vol 121, pp. 409-418, 1973
- [6] X.Chen, X.Wu, "THE SYNTHESIS OF TERNARY FUNCTIONS UNDER FIXED POLARITIES AND TERNARY I<sup>2</sup>L CIRCUITS," IEEE Proc. of Symposium on Multiple-Valued Logic, Kyoto, Japan, pp.424-429, May 1983
- [7] David Green, Modern Logic Design, Addison-Wesley Publishing company, Inc.1986
- [8] George Epstein, Multiple-valued logic design an introduction, Institute of Physics Publishing Ltd 1993