

극수비교에 의한 GF(3)의 일반화된 Reed-Muller 계수 생성 방법

The Generation Method to Generalized Reed-Muller Coefficients over GF(3) by means of the Comparison of the Polarity.

李 哲 雨*, 金 興 壽*

(Chol-U Lee* and Heung-Soo Kim*)

요 약

본 논문에서는 다치논리 함수의 GRM(Generalized Reed-Muller)계수 생성 방법에 관하여 제안하였다. 일반적인 GRM계수의 생성 방법은 Reed-Muller(RM) 전개식을 이용하여 극수 $P=0$ 의 RM계수를 구하고 이를 확장하여 모든 GRM계수를 구하는 방법을 사용한다. 본 논문에서 제안한 알고리즘은 모든 극수의 GRM계수를 구하지 않고 극수의 0의 개수를 순차적으로 비교해가며 GRM계수를 구하는 방식이다.

Abstract

This paper presents a method for the generation of GRM coefficients over GF(3) by using a comparison of polarity. In general production method to GRM coefficients over GF(3) is searching for pn different polarity of an n -variable and from these optimal function according to the maximum number of zero coefficients is selected. This paper presents a method for the generation of GRM coefficients by means of compare to the number of zero coefficients without constructing the whole polarity GRM coefficients.

I. 서 론

현재 사용되고 있는 논리회로 시스템은 부울함수를 기초로 한 2진 논리회로로 구성되었으며 지난 수십년간의 빠른 기술의 발달은 칩의 집적도 성능을 비약적으로 향상시켰다. 그러나 회로가 지나치게 집적화되는 것은 칩내 소자간의 배선 및 회로 결선에 의한 신호선 사이의 혼선잡음과 신호지연등의 문제점이 발

생하게 되었다. 이러한 문제점의 해결 방안의 하나로 다치논리를 VLSI 칩속에 적용하여 상호 연결의 수와 연결 길이를 감소하여 집적회로의 처리 능력과 신뢰성을 향상시킬 수 있는 다치논리에 대한 연구가 활발히 진행되고 있다[1-6]. 다치논리는 하나의 신호선에 2진 이상의 신호를 전송하여 시스템을 구성하는 논리회로로 이러한 다치논리 시스템의 장점은 칩밀도 향상, 상호 연결들의 전체 길이와 상호 연결 지연 감소, 처리 속도 향상, 전력 소비 감소등이 있다.

이러한 다치논리에 대한 연구는 현재로서도 상당히 많은 부분이 개발되어야 하며 다치 시스템을 구현

* 仁荷大學校 電子工學科

(Dept. of Electronics Eng., Inha Univ.)

接受日: 1999年7月26日, 修正完了日: 1999年11月22日

하기 위하여는 기존의 2진 논리에서 적용되는 모든 연산을 다치신호로 표현해야하며 다치 하드웨어에 대한 알고리즘의 발전을 지속적으로 행하여야 한다.

본 논문에서는 다치논리 함수에 대한 연구분야의 하나로 함수의 간략화에 대한 논문으로서 Reed-Muller 전개식을 이용하여 일반화된 RM계수를 구하고 최적의 함수를 구성하는 방법에 관하여 논하였다.

RM 전개식을 이용하는 이점은 소자 수와 게이트 상호 연결 수에 있어서 타 함수의 논리회로 실현보다 경제적이며 테스트가 용이하다. RM전개식에 의하여 표현된 함수 영역의 유일한 함수로 존재하는 것은 아니다. 만약 입력변수들에 대하여 입력의 형태를 달리 하면 새로운 형태의 정규화된 함수를 구성한다. 이와 같이 GF(p)상의 n개의 변수에 대하여 p^n 개의 서로 다른 극수(polarity)에 대한 계수를 구하는 과정을 GRM변환이라 하며 여러 가지 방법들이 제안되었다.[7-9]

Besslich는 EX-OR함수를 이용한 GRM계수의 생성 방법을 극수의 변화에 따르는 계수간의 관계를 맵(map_상에서 해석하여 순차적인 GRM계수를 구하는 방법에 관하여 나타내었다.[10] Wu 등도 2치 함수에 대하여 EX-OR함수를 이용한 GRM계수의 생성방법을 맵을 이용하여 구하였다.[11] Green은 2치 함수에 대하여 Kronecker 곱에 의한 전달행렬을 구하여 이를 이용한 RM변환과 GRM계수를 구하였으며 3치와 4치 함수에 대한 GRM 계수 생성방법을 제안하였다.[12]

이러한 방법들은 변수의 개수가 많아짐에 따라 연산과정이 복잡해지는 단점을 갖고 있다. 따라서 본 논문에서는 순차적으로 GRM 계수의 0의 개수를 비교하여 가며 모든 극수의 GRM계수를 구하지 않고 최적의 GRM계수를 구하는 방법을 제안하였다.

본 논문의 구성을 살펴보면 II장에서는 RM전개식, GRM전개식등 수학적배경에 대하여 소개하였고, III장에서는 단변수에 대한 순차적 GRM계수 생성에 대하여 제안하였으며 IV장에서는 극수비교에 의한 계수 생성 알고리즘을 도출하고 예제를 통하여제안한 알고리즘을 적용하였다. V장에서는 결과를 기존의 연구와 비교 및 검토하였으며 마지막으로 VI장에서는 결론을 맺었다.

II. 수학적 배경

2.1 유한체의 성질

부울(Boolean)체의 확장체인 유한체(Galois Field ;GF)는 p를 소수로 p개의 원소로 구성된 체를 형성하며 GF(p)로 표기한다. 예를들어 GF(p)는 {0,1,...,p-1}의 p개의 원소를 갖는 p치의 함수를 나타내며 원소 사이에는 교환법칙, 결합법칙, 분배법칙 등이 성립하고 가산(감산) 및 승산(제산)에 대한 역원이 유일하게 존재한다. 또한 GF(p)상의 모든 연산은 mod-p의 연산으로 이루어진다. p를 소수, m을 양의 정수라 할 때 GF(pm)은 pm개의 원소로 구성된 체를 의미하며 GF(p)의 확장체라 한다.

2.2 RM변환

다치논리 함수에 대한 해석은 입력함수에 대한 출력값을 일반화한 진리표로 나타나는 연산영역에서 함수의 해석과 입력변수를 함수적으로 표현하여 함수 영역에서 해석이 가능하며 두 영역사이의 변환과정을 RM변환이라 한다. 일반적으로 RM변환에 의하여 연산영역의 데이터를 함수영역으로 변환하는 과정은 변환을 위한 전달행렬을 먼저 구하고 연산영역의 데이터를 이용한 행렬연산을 행하여 함수 영역의 데이터를 얻을 수 있다. 역으로 함수영역의 데이터를 연산 영역의 값으로 변환할 때에는 역행렬을 이용하여 구하면 연산영역에서의 값을 얻을 수 있다. 이를 그림으로 표현하면 다음과 같다.

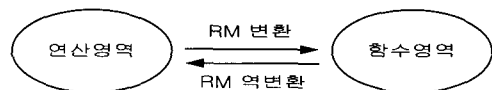


그림 2.1 RM변환에 의한 영역사이의 변환

2.2.1 GF(3)상의 RM변환

연산 영역에서의 함수 $f(x)$ 의 계수 값을 d_i , 함수 영역에서의 계수 값을 c_i 라 할 때 두 영역의 계수

들간의 관계는 식2.1과 같이 표현된다.

$$\begin{aligned} d_0 &= f(0) = c_0 \\ d_1 &= f(1) = c_0 + c_1 + c_2 \\ d_2 &= f(2) = c_0 + 2c_1 + c_2 \end{aligned} \quad (2.1)$$

식2.1을 행렬의 형태로 표현하면 식2.2와 같다.

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \text{ over GF(3)} \quad \text{or} \\ d = S_1 c$$

S_1 의 역변환을 $T(T = S_1^{-1})$ 라 하면 T 는 연산 영역에서 함수 영역으로의 변환을 의미하며 식2.3과 같이 표현된다.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix} \text{ over GF(3)} \quad \text{or} \\ c = T_1 d$$

GF(3)상에서 입력변수가 2개인 경우의 Reed-Muller expansion은 다음과 같이 표현된다.

$$\begin{aligned} f(x_2, x_1) &= c_0 + c_1 x_1^1 + c_2 x_1^2 + c_3 x_2 + c_4 x_2 x_1 \\ &\quad + c_5 x_2 x_1^1 + c_6 x_2^2 + c_7 x_2^2 x_1 + c_8 x_2^2 x_1^2 \end{aligned}$$

2.3 GRM변환

RM변환에 의하여 구성된 함수식은 유일한 함수식이 아니다. 만약 입력변수 x 대신에 $x+1$ 을 입력한다면 또 다른 형태의 정규화된 RM함수식으로 표현된다.

GF(p)상에서의 n개의 입력 변수에 대한 pn개의 서로 다른 입력형태가 만들어지며 이에 대한 RM계수를 구하는 과정을 GRM변환이라 한다. GRM계수를 구하는 목적은 pn개의 서로 다른 정규화된 함수식에 대하여 항의 개수를 최소로 하는 함수를 선정하여 이

를 이용한 함수 구현을 목적으로 한다. 최적의 GRM 함수를 구한다는 것은 결과적으로 0의 개수를 최고로 많이 갖는 입력변수에 대한 입력형식을 찾는 것이다. GRM계수를 구하는 방법은 연산영역의 값을 이용하여 GRM계수를 구하는 방법도 있지만 일반적으로 연산영역으로부터 RM함수를 구하여 이를 확장하여 GRM함수를 구하는 방법을 사용한다.

2.3.1 GF(3)상의 GRM변환

단변수에 대한 RM전개식은 다음과 같다.

$$f(x) = c_0 + c_1 x + c_2 x^2 \quad (2.4)$$

GF(3)상에서 입력변수 x 의 보수에 해당하는 \bar{x} 는 $x+1, x+2$ 를 의미하며 이는 극수 1(P=1)과 2(P=2)를 의미한다.

식(2.4)의 x' 에 $x+k(k=1\text{또는}2)$ 를 대입하여 x' 항을 표현하면

$$[1 \ x' \ x'^2] = [1 \ x \ x^2] \begin{bmatrix} 1 & k & k^2 \\ 0 & 1 & 2k \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

이며 역변환을 행하면 식(2.6)과 같다.

$$[1 \ x \ x^2] = [1 \ x' \ x'^2] \begin{bmatrix} 1 & 2k & k^2 \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

식(2.5)를 식(2.4)에 대입하면 식(2.7)와 같은 결과를 얻을 수 있다.

$$\begin{aligned} f(x) &= c'_0 + c'_1 x + c'_2 x^2 \\ &= [1 \ x' \ x'^2] \begin{bmatrix} 1 & 2k & k^2 \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \end{aligned} \quad (2.7)$$

식(2.7)의 극수 k 에 의해 표현된 행렬에 $k=1$ 과 $k=2$ 를 적용하여 P=0의 RM계수에서 P=1과 P=2에 대한 계수의 변환 값을 표로 나타내면 다음과 같다

표 2.1 GF(3)의 극수에 따른 계수 변환

c_i	$P=0$ ($x'=x$)	$P=1$ ($x'=x+1$)	$P=2$ ($x'=x+2$)
c_0	c_0	$c_0 + 2c_1 + c_2$	$c_0 + c_1 + c_2$
c_1	c_1	$c_1 + c_2$	$c_1 + 2c_2$
c_2	c_2	c_2	c_2

n변수에 대한 극수의 변환은 단변수에 대한 변환 과정을 n변수로 확장 적용하여 구하면 3^n개에 해당하는 극수의 GRM계수를 구할 수 있다.

[정의 1] $c_i^{(k)}$ 은 $P=k$ 변환에 의해 생성된 다항식의 i번째 계수를 의미한다.

[정의 2] $f^k(x)$ 는 $f(x)$ 함수의 $P=k$ 의 GRM 함수 $f^k(x)$ 를 의미한다.

[정의 3] $B^{(k)}$ 는 극수 p=k변환에 의해 생성된 다항식의 계수 벡터를 의미한다.

III. 단변수에 대한 순차적 GRM계수 생성

p차 단변수의 경우 P=0을 제외한 모든 극수의 GRM계수생성은 극수에 따라서 p-1개의 서로 다른 계수의 생성방법이 존재함을 알 수 있다. GF(3)의 경우 P=1과 P=2에 대한 계수의 변환은 표3.1과 같다. 표3.1을 이용하여 단 변수에 대한 각 계수에 대한 생성과정을 살펴보면 다음과 같다

(1) $c_2^{(k)}$ 계수 생성

표 3.1에서와 같이 극수에 따른 $c_2^{(k)}$ 계수의 값은 변화가 없음을 알 수 있다.

(2) $c_1^{(k)}$ 계수 생성

극수에 따른 $c_1^{(k)}$ 계수의 생성은 다음과 같이 2가

지 방법이 존재한다. $c_0^{(k)}$ 의 계수 만을 이용하여 $c_1^{(k)}$ 의 계수를 구하면 표3.1에서와 같이 극수에 따라 서로 다른 계수 생성방법을 사용하여야 한다. 그러나, 다음과 같은 방법을 이용하면 일관성있는 방법으로 $c_1^{(k)}$ 의 계수를 구할 수 있다.

① 극수 1의 계수 생성식을 이용

극수 1의 $c_1^{(1)}$ 의 계수를 구하는 식을 이용하면 극수 1과 극수 2의 $c_1^{(1)}, c_1^{(2)}$ 의 계수를 구할 수 있으며 다음과 같은 연산을 행한다.

$$c_1^{(1)} = c_1 + c_2$$

$$c_1^{(2)} = c_1 + 2c_2$$

$$= c_1 + c_2 + c_2 = c_1^{(1)} + c_2$$

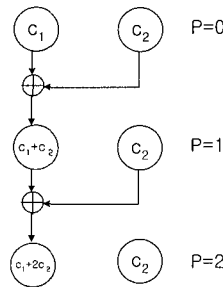


그림 3.1

그림 3.1 P=1의 변환을 이용한 $c_1^{(k)}$ 계수 생성

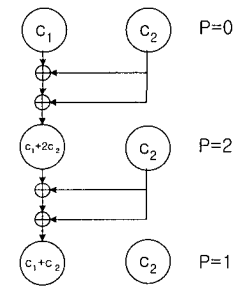


그림 3.2

그림 3.2 P=2의 변환을 이용한 $c_1^{(k)}$ 계수 생성

(3) $c_0^{(k)}$ 계수의 생성

극수 1과 극수 2에 대한 c_0 의 계수를 구하기 위하여 다음과 같은 방법을 사용하여 구하면 최소의 연산자로서 계수의 구성이 가능하다.

① $c_0^{(1)}$ 계수의 생성

계수 $c_0^{(1)}, c_0^{(2)}$ 의 계산은 극수의 순환성을 이용하면 연산자의 개수가 늘어나므로 연산자의 개수를 줄이기 위하여 다음과 같은 방법을 사용한다.

② $c_0^{(2)}$ 계수의 생성

극수에 따른 계수의 생성에서 $c_1^{(k)}$ 의 생성은 극

수 1과 2의 생성 방법이 동일하게 적용되어 계수의 생성이 편리하며 $c_0^{(k)}$ 의 계수 생성과정은 다른 극수의 계수를 참조해야 하는 복잡성이 따르지만 연산자의 계수를 줄일 수 있는 방법으로 생각된다.

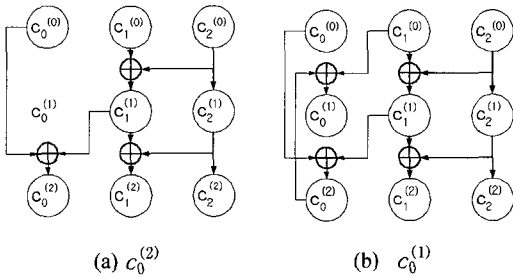


그림 3.3 $c_0^{(1)}$, $c_0^{(2)}$ 계수의 생성

IV. 극수비교에 의한 계수 생성 알고리즘

최적의 GRM함수를 구하는 방법은 모든 극수의 함수를 구하여 함수의 항이 가장 적은, 즉 GRM계수의 0의 개수가 가장 많은, GRM함수를 구하는 방법이다. 본 논문에서 제안된 방법은 부분적인 극수의 GRM계수를 구하여 최적의 GRM함수를 구하는 방법에 대하여 설명한다. 표 2.1에서 단 변수에 대한 계수의 변환과정에서 P=1 변환의 경우 c_2 의 계수생성을 제외한 경우 계수 c_1 의 생성 방법이 가장 적은 연산자를 사용함을 알 수 있다. P=1 변환을 이용한 $c_1^{(1)}$, $c_1^{(2)}$ 의 생성 방법은 그림 3.1을 이용하여 구하며 최소의 연산자의 개수로 구성된 GRM계수를 구할 수 있다. n변수에 대한 생성 방법도 단변수에 대한 방법을 확장하여 구하면 결과를 얻을 수 있다. 그림 3.1 과 그림 3.2의 방법을 이용하여 순차적으로 계수를 생성하여 비교과정을 반복적으로 행하여 GRM계수를 구하는 방법은 n변수인 경우에는 단계적으로 다음과 같다.

[단계 1] c_1, c_2 계수의 생성

단변수에 대한 c_1 과 c_2 의 계수를 구한다. c_2 의 항은 표2.1에 의하여 극수 변환시에 계수의 변환이

없다. c_1 의 경우는 표2.1에 의하여 계수 변환시에 최소의 연산자를 갖는 극수 1의 변환을 사용하면 최소의 연산자로 구할 수 있다. n변수에 대하여는 단 변수에 대한 과정을 n변수로 확장 적용하여 구하면 된다. 모든 변수에 대하여 그림 3.1과 같은 방법으로 c_1, c_2 에 대한 계수 값을 구한다.

[단계 2] 0의 계수 항 비교

[단계 1]에 의해 생성된 계수에 대한 0의 개수를 비교하여 0의 개수의 차이가 n변수에 대하여 3^{n-1} 이상의 차이를 보이는 극수에 대하여는 다음 단계의 극수 생성을 생략한다.

[단계 3] c_0 계수의 생성

c_0 의 계수는 그림 3.3에 나타난 바와 같이 연산자의 계수를 줄이기 위하여 서로 다른 극수의 계수를 참조하여 구한다.

[단계 4] 계수 0의 개수 비교

0의 개수가 최고로 많은 극수에 대하여 최적의 GRM 계수를 선정한다.

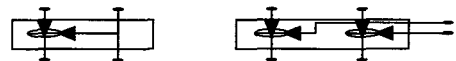
[예제 1] 비교극수를 이용한 GRM계수 생성방법을 적용하여 3차 2변수 P=0의 함수가 다음과 같은 경우 최적의 GRM계수를 구하여라.

$$f(x) = x_2 + x_2^2 + x_1 + 2x_1x_2 + 2x_1x_2^2 + x_1^2 + x_1^2x_2 + x_1^2x_2^2$$

P=1의 변환을 이용하여 c_1, c_2 의 계수를 구하기 위하여 그림 3.1과 같은 순서로 극수의 변환을 행한다.

[단계 1] c_1, c_2 계수의 생성

c_1, c_2 의 계수는 그림 4.1의 가산기를 이용한 기본 블록 BK를 다음과 같이 정의하여 구하면 그림 4.1과 같다



BK1 BK2
그림 4.1 BK1, BK2 블록도

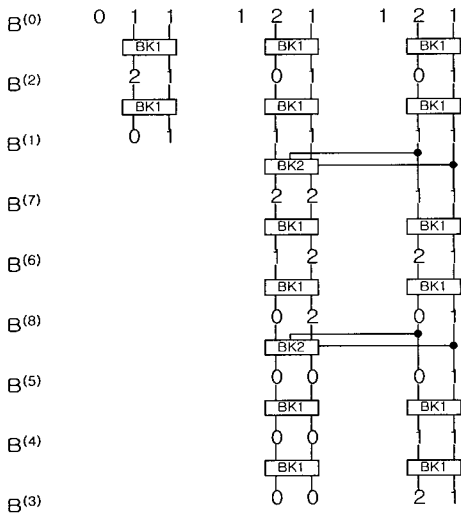


그림 4.2 c_1, c_2 계수 생성

그림 4.2의 B(1) --> B(7)의 변환에서 $c_1^{(7)}, c_2^{(7)}$ 계수 값은 단변수에 대한 계수생성 방법을 2변수로 확장하여 적용하면 다음과 같은 BK3로 구할 수 있다.

$$c_1^{(7)} = c_1^{(1)} + 2c_4^{(1)} + c_7^{(1)}$$

$$c_2^{(7)} = c_2^{(1)} + 2c_5^{(1)} + c_8^{(1)}$$

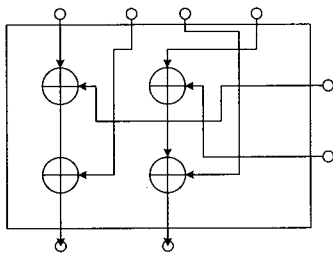


그림 4.3 BK3 블록도

BK3에 의하여 생성된 계수 값은 BK1의 단변수에 대한 c_1, c_2 를 x_2 변수에 적용하여 B(6), B(8)의 계수를 구할 수 있다. B(8) --> B(5)에 대한 변환도 BK3를 적용하여 구하면 그림 4.4와 같다.

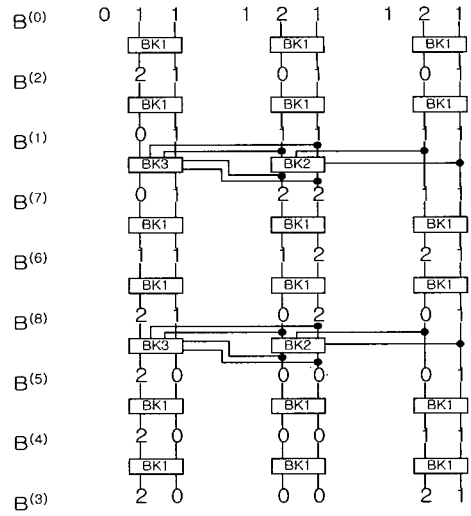


그림 4.4 c_1, c_2 계수 생성

[단계 2] 0의 계수 개수 비교

그림 3.7에 의하여 P=0의 함수를 제외한 모든 극수의 계수 값은 x_2 변수에 대한 c_0 의 계수 값을 구하면 된다. 위의 예제의 경우 최대의 0의 개수를 갖는 극수가 모든 극수에 비하여 3개 이상의 0의 개수에 차이를 나타내면 해당 극수가 최적의 극수이다. 예제의 경우 B(5)가 4개의 0의 개수를 갖지만 모든 극수에 대하여 3개 이상의 차이를 갖지 않으므로 계수 c_0 에 대한 생성 방법을 계속 진행한다.

[단계 2] 0의 계수 개수 비교

그림 3.7에 의하여 P=0의 함수를 제외한 모든 극수의 계수 값은 x_2 변수에 대한 c_0 의 계수 값을 구하면 된다. 위의 예제의 경우 최대의 0의 개수를 갖는 극수가 모든 극수에 비하여 3개 이상의 0의 개수에 차이를 나타내면 해당 극수가 최적의 극수이다. 예제의 경우 B(5)가 4개의 0의 개수를 갖지만 모든 극수에 대하여 3개 이상의 차이를 갖지 않으므로 계수 c_0 에 대한 생성 방법을 계속 진행한다.

[단계 3] c_0 계수 생성

B(1), B(2) 를 기준극수로 사용하기 위하여 그림

4.5과 같은 BK4 를 사용하여 B(1), B(2) 에 대한 계수를 구하면 그림 4.7과 같다.

$$c_0^{(2)} = c_0^{(0)} + 2c_1^{(0)} + c_2^{(0)}, \quad c_3^{(2)} = c_3^{(0)} + 2c_4^{(0)} + c_5^{(0)}$$

$$c_6^{(2)} = c_6^{(0)} + 2c_7^{(0)} + c_8^{(0)}, \quad c_0^{(1)} = c_0^{(2)} + 2c_1^{(2)} + c_2^{(2)}$$

$$c_3^{(1)} = c_3^{(2)} + 2c_4^{(2)} + c_5^{(2)}, \quad c_6^{(1)} = c_6^{(2)} + 2c_7^{(2)} + c_8^{(2)}$$

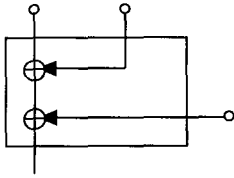


그림 4.5 BK5 블록도

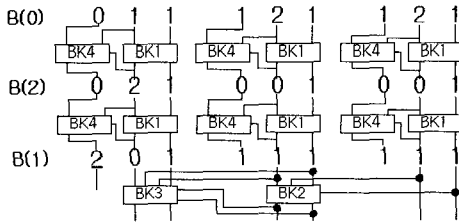


그림 4.6 B(1), B(2) 계수 생성

c_0, c_3, c_6 은 임의의 x_2 변수의 항에 대해 c_0, c_1, c_2 를 의미하며 이들에 대하여 $P=1$ 의 변환을 이용하여 $B(0) \rightarrow B(6) \rightarrow B(3)$ 의 변환을 행하며 BK1 블록을 사용하여 구하면 다음과 같다.

$$c_3^{(6)} = c_3^{(0)} + c_6^{(0)}, \quad c_6^{(6)} = c_6^{(0)}$$

$$c_3^{(3)} = c_3^{(6)} + c_6^{(6)}, \quad c_6^{(3)} = c_6^{(6)}$$

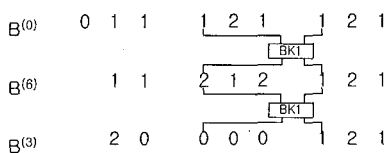


그림 4.7 계수 생성

그림 4.8의 x_2 변수에 대한 계수 생성 방법을 $B(1) \rightarrow B(7) \rightarrow B(4)$ 와 $B(2) \rightarrow B(8) \rightarrow B(5)$ 에 적용하여 $c_3^{(7)}, c_6^{(7)}, c_3^{(4)}, c_6^{(4)}$ 와 $c_3^{(8)}, c_6^{(8)}, c_3^{(5)}, c_6^{(5)}$ 계수들을 구할 수 있으며 그림 4.8와 같다. 그림 4.8에서 0의 개수를 가장 많이 포함하고 있는 함수는 $P=5$ 의 변환에 의하여 이루어진 $B(5)$ 계수이며 6개의 0의 개수를 포함한다. 다음의 값은 $B(2), B(3), B(4), B(8)$ 로서 4개의 0의 개수를 포함한다. 그러므로 아직 구하지 않은 c_0 의 계수를 구하지 않고도 $P=5$ 의 변환이 최적의 변환임을 알 수 있으며, 아래와 같은 성질을 이용하여 비교극수 $B(2)$ 로부터 $c_0^{(5)}$ 를 구한다.

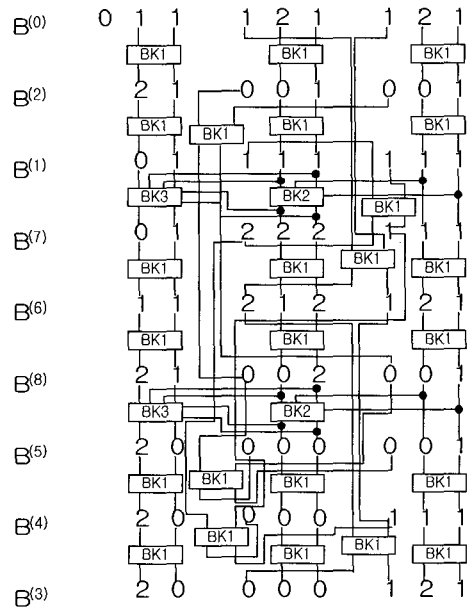


그림 4.8 계수 생성

$B(2) \rightarrow B(5) \rightarrow B(8)$ 의 변환 과정을 보면 아래와 같은 성질이 있다.

$$c_0^{(5)} = c_0^{(2)} + c_3^{(2)} + c_6^{(2)}$$

$$c_0^{(8)} = c_0^{(5)} + c_3^{(5)} + c_6^{(5)}$$

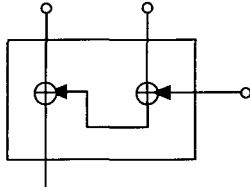


그림 4.9 BK5의 블록도

이 성질은 B(1)→B(4)→B(7) 와 B(0)→B(3)→B(6)에도 그대로 적용된다.

그림 4.9과 같은 BK5를 그림 4.8에 적용하면 그림 4.10과 같이 P=5 경우의 GRM 계수가 얻어진다.

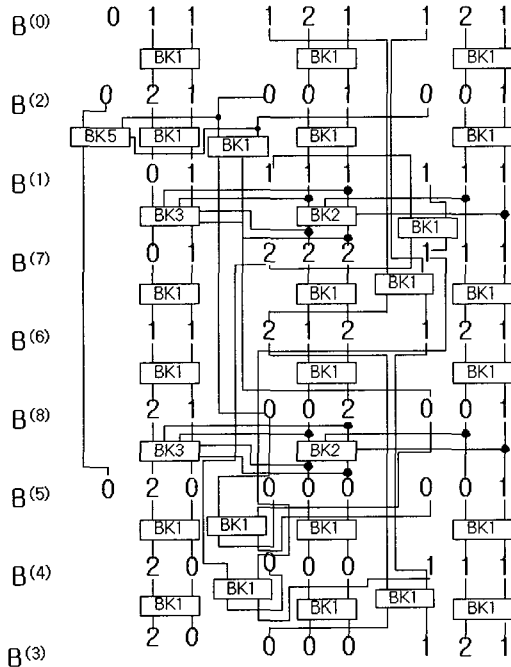


그림 4.10 P=5 GRM 계수 생성

V. 비교 및 검토

본 장에서는 본 논문에서 제안한 극수비교를 이용한 GRM계수의 생성방법과 기 발표된 알고리즘에 대하여 극수 생성 방법과 최적의 GRM계수를 찾는 방법을 중심으로 대표적인 병렬형의 출력을 갖는

Hong 논문과 직렬형의 출력을 갖는 Green의 논문을 중심으로 비교한다.

① 최적의 GRM계수의 선정 방법

기 발표된 논문의 최적의 GRM계수를 생성하는 방법은 모든 극수를 순차적으로 구한 뒤, 모든 극수의 GRM계수에 대하여 0의 개수를 최대로 하는 극수를 선정하여 최적의 함수를 구성하였다. 그러나 본 논문에서는 모든 극수의 GRM계수를 구성하지 않고서도 최적의 GRM계수 생성시에 부분적인 GRM계수의 0의 개수를 비교하여 비교 우위에 의한 극수만을 연산하는 방법으로 최적의 GRM계수를 구하는 알고리즘을 제안하였다. 이와 같은 방법을 사용하면 소프트웨어적인 방법으로 GRM계수 생성시에 모든 극수의 계수를 생성하지 않고서도 최적의 함수를 생성할 수 있으므로 연산의 효율성을 높일 수 있다.

② 연산자의 개수

극수비교를 이용한 방법과 기 발표된 논문과의 연산자의 개수에 대한 비교를 행하면 다음과 같다.

Hong의 병렬형의 방법[12]은 Green의 직렬형[11]의 방법에 비하여 상대적으로 연산자의 개수가 적게 든다. 본 논문에서 제안한 방법도 연산자의 개수에 있어서는 Green의 방법보다 적게 소요됨을 알 수 있다. 또한 계수의 생성과정에 대하여 살펴보면 Hong의 방법은 다른 극수의 계수를 참조해야 하는 복잡성이 따르지만 제안한 방법은 순차적인 계수 생성 구조로서 이전 극수에 대한 계수만을 참조 함으로서 회로의 구현이 용이하다.

표 5.1에서는 Green, Hong과 본 논문에서 제시한 방법에 대하여 연산자의 개수를 표로 나타내었다.

본 논문에서 제시한 연산자의 개수는 모든 극수의 계수들을 다 계산할 경우를 나타낸 것이다.

만약, 극수들을 비교하는 과정에서 일부 극수의 계수 생성이 생략된다면 연산자의 개수는 본 논문에서 제시한 것보다 줄어들 것이다.

VI. 결 론

표 5.1 연산자 개수 비교

n	연산자	Green	Hong	본 논문
1	가산	6	4	6
	승산	2	0	0
2	가산	72	40	60
	승산	24	0	0
3	가산	702	316	504
	승산	234	0	0
n	가산	$3^n \cdot (3^n - 1)$	$7^n - 3^n$	$8^n - 2^n$
	승산	가산/3	0	0

본 논문은 GF(3)상의 다치논리함수의 GRM계수 생성방법에 관한 방법을 제안하였다. 일반적인 GRM 계수의 생성은 전달행렬을 이용하여 모든 극수들에 대한 GRM계수를 구하고 0의 개수를 비교하여 최적의 GRM함수를 구하는 방법을 사용한다. 본 논문에서 제안한 방법은 기 발표된 논문과는 다르게 모든 GRM계수를 구하는 것이 아니라 부분적인 GRM계수를 순차적으로 생성해가며 0의 개수를 비교하여 최적의 GRM계수를 구하는 방법으로 모든 극수의 GRM계수를 구할 필요가 없으므로 계수생성을 위한 연산의 개수에 있어서 기 발표된 논문에 비교하여 적게 사용할 수 있으며 연산의 방법도 기 발표된 논문에 비하여 어렵지 않다. 제안된 방법은 하드웨어적인 GRM계수 생성보다 소프트웨어적인 계수 생성 방법에 더 적합하며 GF(p)상에서 제안된 방법을 적용할 때도 단변수에 대한 계수 생성과정을 확대 적용할 수 있다는 장점이 있다.

참 고 문 헌

[1] K. C. Smith, "Multiple-valued logic : a tutorial and appreciation," *IEEE Trans. Comput.*, pp. 17-27, Apr. 1988.
 [2] M. Kameyama and T. Higuchi, "Multiple-Valued Information Processing System and its realization," *Trans. IEICE*, vol. J72-A, no. 2, pp. 198-207, Feb.

1989.

[3] M. Kameyama, "Toward the age of beyond-binary electronics and systems," *Proc. IEEE 20th Int. Symposium on Multiple Valued Logic*, pp. 162-166, May. 1990.
 [4] K. C. Smith and P. G. Gulak, "Prospects for Multiple-Valued Integrated Circuits "special issue on Multiple-Valued Integrated circuits," *IEICE Trans. Electron.*, Vol E76-C., no. 3, Mar. 1993.
 [5] P. Dong, "The Optimization of GMC over GF(p)," *IEEE Proc. of International Symposium on Multiple Valued Logic*, Kyoto, Japan, pp. 342-347, May. 1983.
 [6] B. Harking and C. Moraga, "Efficient Derivation of Reed-Muller Expansions in Multiple-Valued Logic System," *IEEE Proc. of International Symposium on Multiple Valued Logic*, Sendai, Japan, pp. 436-441, May. 1992.
 [7] W. Bessilich, "Efficient computer method for EX-OR logic design," *IEE. Proc.*, Part E, vol. 130, no. 6, pp. 203-206, Nov. 1983.
 [8] X. Wu, X. Chen, "Mapping of Reed-Nuller coefficients and the minimization of exclusive OR-switching functions," *IEE. Proc.*, Part E, vol. 129, no. 1, pp. 15-20, Jan. 1982.
 [9] X. Chen and X. Wu, "The Synthesis of ternary Functions under Fixed polarities and Ternary I2L Circuits," *IEEE Proc. of International Symposium on Multiple Valued Logic*, Kyoto, Japan, pp. 424-429, May. 1983.
 [10] Davio Green, *Modern Logic Design*, Addison Wesley publishing co., 1986.
 [11] D. H. Green, "Ternary Reed-Muller switching functions with fixed and mixed polarity," *Int. J. Electronics*, vol. 67, no. 5, pp. 761-775 Nov. 1989.
 [12] Q. Hong, B. C. Fei, H. M. Wu, M. A. Perkowski, and N. Zhaung, "Fast Synthesis for Ternary Reed-Muller Expansion," *IEEE Proc. of Int. Sym. on Multiple Valued Logic*, Sacramento, California,

USA, pp. 14-16, May 1993.

- [13] T. Sasao, "Calculation of Reed-Muller-Fourier Coefficients of Multiple-Valued Functions through Multiple-Place Decision," *IEEE Proc. of Int. Sym. on Multiple Valued Logic*, Boston, Massachusetts, USA, pp. 82-88, May. 1994.

저 자 소 개



李 哲 雨 (會 員 申 請 中)

1998년 인하대학교 전자공학과 공
학사, 1998년~현재 인하대학교 석
사학위과정 입학, 1999년 11월 현
재 인하대학교 석사학위과정 재학
중.

金 興 壽 (正 會 員)

전지전자학회 논문지 第 1卷 第 1號 參照