Reed-Muller 전개식에 의한 다치 논리회로의 구성에 관한 연구

성 현 경†

요 약

본 논문에서는 Reed-Muller 전개식에 의한 다치 논리 회로의 구성에 관한 한 가지 방법을 제시하였다. 먼저, Perfect Shuffle 기법과 Kronecker 곱에 의한 다치 논리함수의 입출력 상호연결에 대하여 논하였고, $GF(4)$의 가산회로와 승산회로를 이용하여 다치 Reed-Muller 전개식의 변환행렬과 역변환행렬을 실행하는 기본 셀을 설계하였다. 이 기본 셀들과 Perfect Shuffle과 Kronecker 곱에 의한 입출력 상호연결 방법을 이용하여 다치 Reed-Muller 전개식에 의한 다치 논리 회로를 구현하였다. 제시된 다치 Reed-Muller 전개식의 설계방법은 모듈구조를 기반으로 하여 행렬변환을 이용하므로 동일한 함수에 대하여 타 방법과 비교하여 간단하고 회로의 가산회로와 승산회로를 줄이는데 매우 효과적이다. 제안된 다치 논리회로의 설계방법은 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬동작의 특징을 가진다.

키워드 : 다치논리회로, Kronecker 곱, Perfect Shuffle, 다치 Reed-Muller 전개식

Study on Construction of Multiple-Valued Logic Circuits Based on Reed-Muller Expansions

Hyeon-Kyeong Seong†

ABSTRACT

In this paper, we present a method on the construction of multiple-valued circuits using Reed-Muller Expansions(RME). First, we discussed the input-output interconnection of multiple-valued function using Perfect Shuffle techniques and Kronecker product and designed the basic cells of performing the transform matrix and the reverse transform matrix of multiple-valued RME using addition circuit and multiplication circuit of $GF(4)$. Using these basic cells and the input-output interconnection technique based on Perfect Shuffle and Kronecker product, we implemented the multiple-valued logic circuit based on RME. The proposed design method of multiple-valued RME is simple and very efficient to reduce addition circuits and multiplication circuits as compared with other methods for same function because of using matrix transform based on modular structures. The proposed design method of multiple-valued logic circuits is simple and regular for wire routing and possess the properties of concurrency and modularity of array.

Key Words : Multiple-Valued Logic Circuits, Kronecker Product, Perfect Shuffle, Multiple-Valued Reed-Muller Expansions

1. 서 론

Reed-Muller 전개식(Reed-Muller Expansions; RME)은 신호처리, 오류검출과 특히 오류제어를 위한 집합코드 또는 블록코드와 관련된 부호화 기법과 같은 여러 분야에서 성공적으로 사용되고 있으며, 유한체는 CDMA 시스템에서 오류정정코드로서 사용되고 있다. 오늘날 반도체 기술의 발달로 인하여 칩의 집적밀도가 비약적으로 증가하고, 회로의 복잡도가 날로 높아가고 있다. 그러나 이렇게 대형화된 집적회로에 심각하게 대두되고 있는 단자수 제한문제, 단자간 상호연결 문제, 보다 많은 정보량의 처리문제와 연산속도의 제한성이라는 근본적인 문제에 직면하게 되었으며, 이러한 문제점을 해결하기 위하여 처리속도와 집적회로의 면적 등에서 장점을 갖는 유한체와 RM 전개식에 대한 연구가 활발히 진행되고 있다[1-4].

RM 전개식은 AND 연산과 XOR 연산의 일반화에 따라서 여러 방법으로 다치 함수로 확장할 수 있으며, RM 전개식의 2진 논리함수는 FPGA와 같은 쉽게 이용할 수 있는 하드웨어를 사용하여 간단하고 쉽게 실현할 수 있다. n변수 p치 논리함수는 p^n개의 서로 다른 RM 전개식으로 확장할 수 있다[5, 6].

RM 전개식은 2진 논리함수와 다치 논리함수 모두로 표현되어 왔다. 다치 논리함수는 2진 논리함수에 비하여 동일 정보량을 처리하는데 상호연결의 복잡성을 감소시키며 단위 면적당 높은 함수 기능과 같은 많은 유연성을 제공하며, 더

※이 논문은 2005년도 상지대학교 교내연구비 지원에 의해 연구되었음
†종신회원 : 상지대학교 컴퓨터정보공학부 교수
논문접수 : 2006년 9월 5일, 심사완료 : 2007년 1월 18일

집약적이다. 그러나 최적의 RM 전개식을 찾는데 있어서 일반화된 RM 전개식의 수는 변수의 수가 증가할 때 2진 논리함수의 경우보다 다치 논리함수의 경우에 더 급격하게 증가한다. 그러므로 많은 알고리즘이 다치 논리함수에 대하여 RM 전개식을 계산하기 위해 제시되었다[7].

다치 논리함수의 고속 변환 알고리즘을 제시한 Yang[8]은 Kronecker 곱을 이용하여 Q치 함수의 모듈러 대수 전개식의 행렬 변환 알고리즘이 효과적인 계산 절차를 가지며, 임의의 3치 변환에서 입력 변수를 증가하므로 연산이 감소함을 보였다. Zaitseva 등[9]은 이산 직교 변환 행렬을 이용하여 다치 논리함수를 표현하는 다항식을 논리적으로 합성하는 방법을 제시하였으며, 제시된 다치 논리함수의 다항식 합성 방법은 다치 논리함수의 성질을 조사하는데 이용될 수 있음 보였다. Rahardja 등[10]은 Reed-Muller 전개식의 4치 스위칭 함수의 행렬을 계산하는 새로운 알고리즘을 제시하였으며, 계수 행렬이 순환 정방형 행렬에 의해서 생성됨을 보였다. 제시된 알고리즘은 변수가 적을 때 효과적이나 변수가 증가하면 소자수가 급격히 증가하는 단점이 있다.

Stankovic 등[11]은 RMF(Reed-Muller Fourier) 전개식과 GF(Galois Fields) 표현식에 대한 변환행렬을 보였으며, 변수가 증가하면 RMF 전개식에 의한 다치 논리회로를 구현하는데 더 효과적임을 보였다. Falkowski 등[12]은 GF(5)에서 RM 변환을 계산하는 방법을 제시하였다. 제시된 방법은 일정한 순서로 하나씩 RM 함수의 스펙트럴 계수 벡터를 계산하는 단점이 있다.

이들이 제시한 방법들은 다치 논리회로의 구성에서 승산과 가산의 연산과정이 증가하는 문제점이 있으며, 이러한 문제점을 해결하기 위하여 간단하고 규칙적이며 배열의 모듈성에 의해 병렬로 동작하는 RME에 의한 다치 논리함수의 구성이 요구되었다.

본 논문에서는 Davio[13]가 제시한 Perfect Shuffle 기법과 Kronecker 곱을 이용하여 다치 논리함수의 입출력 상호연결 방법에 대하여 논하였으며, 다치 논리함수의 입출력 상호연결 방법을 이용하여 다치 RM 전개식에 의한 다치 논리회로를 구현하였다. 제시된 다치 RM 전개식의 설계방법은 모듈구조를 기반으로 하여 행렬변환을 이용하므로 회로의 가산회로와 승산회로를 줄이는데 매우 효과적임을 보인다.

2. 수학적 배경과 기본 회로

2.1. 수학적 배경

(1) Kronecker 곱의 성질

m개의 행렬 M의 Kronecker 곱은 결합법칙에 의해 다음과 같이 표현할 수 있다[13].

$$Z = M_{m-1} \otimes M_{m-2} \otimes \cdots \otimes M_1 \otimes M_0 = \bigotimes_{k=m-1}^{0} M_k \quad (1)$$

식 (1)에서 Z의 엔트리를 $z(i,j)$라 하고, 행렬 M_k의 엔트리를 $m_k(i_k, j_k)$라 하고, M_k는 (r_k, c_k) 행렬이라 하면 다음과 같다.

$$z(i,j) = \prod_{k=0}^{m-1} m_k(i_k, j_k) \quad (2)$$

여기서 i와 j는 유한체상의 원소들을 갖는 행 벡터열 $[r_{m-1}, \cdots, r_1, r_0]$와 열 벡터열 $[c_{m-1}, \cdots, c_1, c_0]$에 각각 대응하는 입력 벡터열 $[i_{m-1}, \cdots, i_1, i_0]$와 출력 벡터열 $[j_{m-1}, \cdots, j_1, j_0]$를 갖는다. 또한 m개의 동일 원소를 갖는 Kronecker 곱을 M의 m차 Kronecker 멱이라 하고 $M^{[m]}$으로 나타낸다.

(2) Perfect Shuffle 기법의 성질

Shuffle 기법은 순열 (permutation: σ)로서 정의되며 임의의 순열로서 (b_1, b_0)-Shuffle은 인접행렬 S_{b_1, b_0}로 나타내며, 차수 b_0, b_1의 정방행렬이며, 행j와 열 i에 속하는 인접행렬 원소 $S_{b_1, b_0}(j,i)$로 표현된다[13].

인접행렬 S_{b_1, b_0}는 다음과 같다.

$$S_{b_1, b_0}(j,i) = \begin{cases} 1 & \text{if } j = i \cdot \sigma \\ 0 & otherwise \end{cases} \quad (3)$$

인접행렬과 Shuffle 기법사이에서 순열행렬을 나타내면 다음과 같다.

$$I_{b_2} \otimes S_{b_1, b_0} \quad (4)$$

블록벡터 $[b_2, b_1, b_0]$상에서 행하는 순열행렬은 블록벡터 $[b_1, b_0]$내에서 분리되어 행하는 b_2 독립의 (b_1, b_0)-Shuffle로서 표현된다. 여기서 I_{b_2}는 블록 b_2의 단위행렬이다.

식 (4)의 행렬 (j,i) 엔트리를 계산하기 위해서 입력 i가 블록벡터 $[b_2, b_1, b_0]$에 대하여 $[i_2, i_1, i_0]$로 주어진다면 블록벡터 $[b_2, b_0, b_1]$에서 $[i_2, i_0, i_1]$으로 표현된 출력 j에 사상됨을 알 수 있다. 특히, 행렬은 임의의 영역에서 블록벡터 b의 k 최소유효비트 (LSB)를 우측으로 한 위치 순환천이를 행한다.

$$I_{b^{m-k}} \otimes S_{b^k, b} \quad (5)$$

유사한 방법으로 순열행렬은 블록벡터 $[b_2, b_1]$내에서 분리되어 행하는 b_0 독립의 (b_2, b_1)-Shuffle로서 표현된다.

$$S_{b_2, b_1} \otimes I_{b_0} \quad (6)$$

여기서 I_{b_0}는 블록 b_0의 단위행렬이다.

특히, 행렬은 임의의 영역에서 블록벡터 b의 k 최대유효비트 (MSB)를 우측으로 한 위치 순환천이를 행한다.

$$S_{b^{k-1},b} \otimes I_{b^{m-k}} \tag{7}$$

Shuffle 기법의 인수분해는 다음과 같다.

$$\begin{aligned} ① \; S_{b_2,b_1,b_0} &= (S_{b_2,b_1,b_1}) \cdot (S_{b_2,b_1,b_0}) \\ &= (S_{b_2,b_1,b_0}) \cdot (S_{b_2,b_1,b_1}) \end{aligned} \tag{8}$$

$$② \; S_{b_2,b_1,b_0} = (S_{b_2,b_0} \otimes I_{b_1}) \cdot (I_{b_2} \otimes S_{b_1,b_0}) \tag{9}$$

2.2. $GF(4)$상의 기본 게이트 설계

이 절에서는 다치 논리함수의 회로설계에 필요한 $GF(4)$상의 기본 게이트를 T-게이트를 이용하여 설계한다.. 다치 논리회로에서 논리 값은 집합 $R=\{0,1,2,3\}$이다.

(1) $GF(4)$상의 가산게이트

유한체 $GF(p^m)$상에서 $GF(4)$는 $p=2$이고 $m=2$인 경우이다. 그러므로 $GF(4)$상의 원소들은 m차 기약다항식을 구하는 $x^{p^m}-x$의 기약인수에 의해 구할 수 있다[15].

$$\begin{aligned} x^{p^m} - x &= x^4 - x \\ &= x \cdot (x-1) \cdot (x^2+x+1) \end{aligned} \tag{10}$$

식 (10)에서 기약인수인 x^2+x+1은 2차 다항식이며, $x^2+x+1=0$의 한 근을 α라 하면 기약다항식은 식 (11)과 같다.

$$\begin{aligned} F(\alpha) &= \alpha^2 + f_1 \cdot \alpha + f_0 = 0 \\ \alpha^2 &= f_1 \cdot \alpha + f_0 \end{aligned} \tag{11}$$

여기서 $f_1, f_0 \in \{0,1\}$이다. 그러므로 $GF(4)$의 원소는 <표 1>과 같다.

〈표 1〉 $GF(4)$의 원소표

$f_1 \cdot \alpha$	f_0	$F(x)$	Symbol
$0 \cdot \alpha$	0	0	0
$0 \cdot \alpha$	1	1	1
$1 \cdot \alpha$	0	α	2
$1 \cdot \alpha$	1	$\alpha+1$	3

그러므로 <표 1>에서 $GF(4)$의 원소들 $\{0,1,\alpha,\alpha+1\}$에 의한 $GF(4)$의 가산표가 <표 2>와 같다.

〈표 2〉 $GF(4)$의 가산표

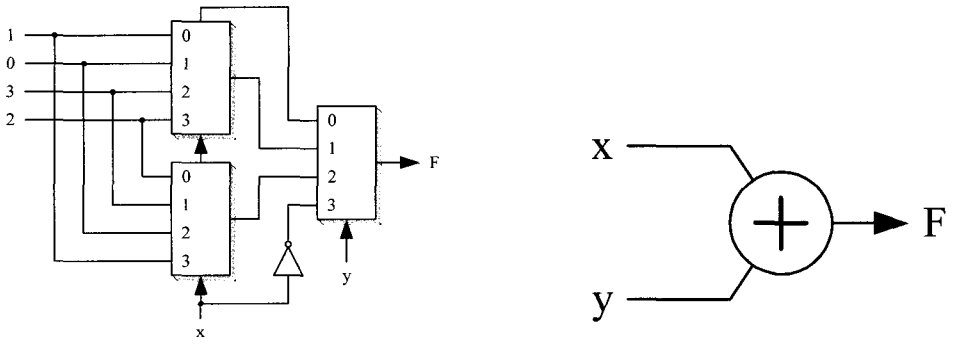
y \ x $\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

T-게이트를 이용하여 $GF(4)$의 가산표인 <표 2>를 실현하는 가산회로가 (그림 1)과 같다. (그림 1)의 (a)는 T-게이트에 의한 $GF(4)$의 가산회로이고, (그림 1)의 (b)는 가산회로의 기호이다.

또한, $GF(4)$의 승산표는 <표 3>과 같다.

〈표 3〉 $GF(4)$의 승산표

y \ x •	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2



(a) (b)

(그림 1) T-게이트에 의한 $GF(4)$의 가산회로
(a) 가산회로 (b) 기호

T-게이트를 이용하여 $GF(4)$의 승산표인 <표 3>을 실현하는 승산회로가 (그림 2)와 같다. (그림 2)의 (a)는 T-게이트에 의한 $GF(4)$의 승산회로이고, (그림 2)의 (b)는 승산회로의 기호이다.

3. 다치 논리 함수의 설계

이 장에서는 2장에서 서술한 수학적 배경을 이용하여 Reed-Muller 전개식에 의한 다치 논리회로의 설계에 대하여 논한다. 먼저, 다치 논리함수의 Perfect Shuffle 기법과 Kronecker 곱과의 상호 관계식을 논하고, 이를 이용하여 Reed-Muller 전개식에 의한 다치 논리회로의 설계를 구현한다.

3.1. 다치 논리함수의 Perfect Shuffle 기법과 Kronecker 곱과의 관계

m개의 행렬 M의 Kronecker 곱을 Z라 하면 다음과 같이 표현된다.

$$Z = M_{n-1} \otimes M_{n-2} \otimes \cdots \otimes M_1 \otimes M_0 = \bigotimes_{k=m-1}^{0} M_k \quad (12)$$

Kronecker 곱은 교환법칙이 존재하지 않으므로 Perfect Shuffle 기법을 이용하여 교환법칙을 성립시킬 수 있다.

[정리 1] 행렬M_k와 행렬N_k가 각각 (r_i, c_i)-행렬과 (r_j, c_j)-행렬이면 다음과 같다.

$$① \left(\bigotimes_{k=m-1}^{0} M_k\right) \otimes \left(\bigotimes_{k=m-1}^{0} N_k\right) = S_{r_j, r_i} \cdot \left[\left(\bigotimes_{k=m-1}^{0} N_k\right) \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right)\right] \cdot S_{c_i, c_j} \quad (13)$$

$$② \bigotimes_{k=m-1}^{0} M_k = \left[S_{r_0, (r_1 \cdots r_m - 1)} \cdot (M_0 \otimes (S_{r_1, (r_2 \cdots r_m - 1)} \cdot (M_1 \otimes \cdots \otimes (S_{r_{m-2}, r_{m-1}} \cdot (M_{m-2} \otimes M_{m-1}) \cdot S_{c_{m-1}, c_{m-2}} \cdots) \cdot S_{(c_m - 1 \cdots c_0), c_1} \cdot S_{(c_m - 1 \cdots c_1), c_0}\right] \quad (14)$$

여기서 S_{r_j, r_i}와 S_{c_i, c_j}는 인접행렬로 표현되는 (r_i, c_i)-Shuffle과 (r_j, c_j)-Shuffle이고, r_i, r_j와 c_i, c_j는 각각 행 벡터와 열 벡터이며, $i, j = \{0, 1, 2, \cdots, m-1\}$이다.

[증명] 행렬의 (x_k, y_k)-엔트리를 계산하기 위하여 $x_k = x_{k_i} \cdot r_j + x_{k_j}$와 $y_k = y_{k_i} \cdot c_j + y_{k_j}$라 하면 좌측항의 (x_k, y_k)-엔트리는 $m_k(x_{k_i}, y_{k_i}) \cdot n_k(x_{k_j}, y_{k_j})$이다. 우측에서 동일한 계산을 수행하기 위하여 $\left(\bigotimes_{k=m-1}^{0} N_k\right) \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right)$을 P라 하고 우측항의 (x_k, y_k)-엔트리를 $r_j \cdot r_i = w$, $c_j \cdot c_i = z$라 하면 식 (18)과 같이 단일항으로 감소된다.

$$\sum_{u=0}^{w-1} \sum_{v=0}^{z-1} S_{r_j, r_i}(x_k, u) \cdot P(u, v) \cdot S_{c_i, c_j}(v, y_k) \quad (15)$$

실제로 u의 단일 값을 U라 하면

$$S_{r_j, r_i}(x_k, U) = 1$$

식 (3)에 의해 u는 $x_k = U \cdot \sigma(r_j, r_i)$로 주어진다. 즉, $U = x_k \cdot \sigma(r_i, r_j) = x_{k_j} \cdot r_i + x_{k_i}$이다.

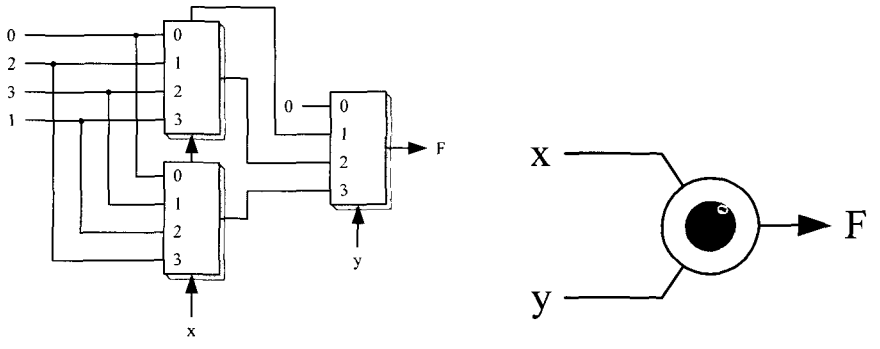
유사한 방법으로 v의 단일 값을 V라 하면

$$S_{c_i, c_j}(V, y_k) = 1$$

식 (3)에 의해 $V = y_k \cdot \sigma(c_i, c_j) = y_{k_j} \cdot c_i + y_{k_i}$이다. 그러므로 식 (15)에서 0이 아닌 항 $P(U, V)$는 $n_k(x_{k_j}, y_{k_j}) \cdot m_k(x_{k_i}, y_{k_i})$와 같다. 유사한 방법으로 식 (14)을 증명할 수 있다.

<Q.E.D>

Kronecker 곱을 일반 행렬곱으로 변환하여 연산하면 승산과정이 감소하므로 Kronecker 곱을 행렬곱으로 계산하기



(a) (b)

(그림 2) T-게이트에 의한 $GF(4)$의 승산회로
(a) 승산회로 (b) 기호

위하여 임의 행렬들을 확장할 필요가 있다. 이 행렬의 확장은 단위 행렬과 Kronecker 곱으로 이루어진다.

[정리 2] 행렬 M_k가 (r_i,c_i)-행렬이면 다음과 같다.

$$① \left(\bigotimes_{k=m-1}^{0} M_k\right) \otimes I_{p_{k_i}} = S_{p_{k_i},r_i} \cdot \left[I_{p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right)\right] \cdot S_{c_i,p_{k_i}} \quad (16)$$

$$② I_{p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right) \otimes I_{p_{k_i}} = S_{p_{k_i},p_{k_i},r_i} \cdot \left[I_{p_{k_i}p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right)\right] \cdot S_{p_{k_i},c_i,p_{k_i}} \quad (17)$$

$$③ I_{p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right) \otimes I_{p_{k_i}} = (I_{p_{k_i}} \otimes S_{p_{k_i},r_i}) \cdot \left[I_{p_{k_i}p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right)\right] \cdot (I_{p_{k_i}} \otimes S_{c_i,p_{k_i}}) \quad (18)$$

여기서 $p_{k_j},p_{k_1},c_i,r_i \in \{0,1,2,\cdots,m-1\}$이다.

[증명] ①은 식 (13)에서 $\bigotimes_{k=m-1}^{0} N_k$ 대신에 $I_{p_{k_i}}$를 대입하여 구할 수 있다.

②는 $(p_{k_1} \cdot r_i\,,\, p_{k_1} \cdot c_i)$-행렬 $I_{p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right)$을 식 (14)에 대입하여 구할 수 있다.

③은 식 (14)에 의해서 다음과 같이 구할 수 있다.

$$I_{p_{k_i}} \otimes \left[\left(\bigotimes_{k=m-1}^{0} M_k\right) \otimes I_{p_{k_i}}\right] = I_{p_{k_i}} \otimes \left(S_{p_{k_i},r_i} \cdot \left(I_{p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right) \cdot S_{c_i,p_{k_i}}\right)\right) \quad (19)$$

이 식을 인수 분해하여 구할 수 있다.

<Q.E.D>

정리 1은 Perfect Shuffle 기법을 이용하여 Kronecker 곱의 교환법칙을 성립시키며, 정리 2는 행렬 $\bigotimes_{k=m-1}^{0} M_k$을 Perfect Shuffle 기법에 의해 다양하게 표현할 수 있음을 나타낸다. 정리 2에서 $I_{p_{k_i}} \otimes \left(\bigotimes_{k=m-1}^{0} M_k\right) \otimes I_{p_{k_i}}$는 입력과 출력의 연결방법을 나타내며, 이 변환식은 $\bigotimes_{k=m-1}^{0} M_k$을 수행하는 블록벡터 연산자들의 집합으로 회로설계에 이용된다.

Kronecker 곱은 연산과정에서 승산수가 증가함으로 회로 합성에서 승산회로가 증가한다. 이를 해결하기 위한 방법은 Kronecker 곱을 행렬곱으로 변환하여 연산하면 승산과정이 감소한다. 임의의 행렬과 단위행렬의 확장식인 정리 2를 이용하여 Kronecker 곱을 인수 분해함으로서 행렬곱의 연산이 가능하다.

[정리 3] $X \in \{0,1,2,\cdots,m-1\}$의 순열이라 할 때 (r_k,c_k)-행렬 M_k의 Kronecker 곱의 인수분해는 다음과 같다.

$$\bigotimes_{k=m-1}^{0} M_k = \prod_{j=k\cdot X}^{0} \left(I_{(p^{m-1}\cdots p^{k+1})} \otimes M_k \otimes I_{(p^{k-1}\cdots p_0)}\right) \quad (20)$$

식 (20)에서 만약 $k\cdot X \geq i\cdot X$이면 $p_i = r_i$이고 $k\cdot X < i\cdot X$이면 $p_i = c_i$이다.

[증명] 식 (20)의 좌측 항에서 M_k가 그 곱의 $k\cdot X$ 위치에 나타나도록 각각의 Kronecker 인수 M_k를 m개의 인수들의 동일한 행렬 곱인 식 (21)로 대체한다.

$$I_{r_k}^{[m-1-k\cdot X]} \cdot M_k \cdot I_{c_k}^{[k\cdot X]} \quad (21)$$

만약 $i>k$이면 M_k는 원소 $I_{p_i}^{[k]}$가 좌측 항에 곱해진다. 분명히 이 원소는 $k\cdot X \geq i\cdot X$이면 I_{r_i}이고, $k\cdot X < i\cdot X$이면 I_{c_i}이다. 유사하게 $i<k$인 경우도 구할 수 있다.

<Q.E.D>

정리 3은 단위행렬에 의한 Kronecker 곱의 확장식을 이용하여 Kronecker 곱을 인수 분해함으로서 일반 행렬곱으로 표현한다.

3.2. 다치 논리함수의 상호연결 방법

유한체 $GF(p^m)$상에서 p^m개의 원소들을 갖는 m변수 p치인 M_i의 Kronecker 곱에 대한 인수분해인 정리 3을 다시 쓰면 다음과 같다.

$$\bigotimes_{k=m-1}^{0} M_k = \prod_{j=k\cdot X}^{0} \left(I_{(p^{m-1}\cdots p^{k+1})} \otimes M_k \otimes I_{(p^{k-1}\cdots p_0)}\right) \quad (22)$$

표기를 간단히 하기 위하여 다음과 같이 정의한다.

$$P_i = \prod_{j=0,j\neq i}^{m-1} P_j \quad (23)$$

그러므로 식 (22)은 다음과 같다.

$$\bigotimes_{i=m-1}^{0} M_i = \prod_{i=m-1}^{0} \left[S_{P_i,p_i} \cdot \left(I_{P_i} \otimes M_i\right)\right] \quad (24)$$

식 (24)는 블록 벡터 $[p_{m-1},p_{m-2},\cdots,p_1,p_0]$에서 입력 벡터열 $[i_{m-1},i_{m-2},\cdots,i_1,i_0]$가 주어지면 $\left(I_{P_i} \otimes M_i\right)$의 연속적 실행에 의해 식 (24)의 좌측 항에서 나타나는 Kronecker 곱의 실행을 대체하며, 각 입력 벡터열의 i번째 원소에 의해 다른 블록 상에서 동작한다. 이 원소는 차례로 Shuffle인 S_{P_i,p_i}에 의해 생성된 연속적 순환 천이에 의해 단위 위치를 이동한다. 또한 모든 행렬 M_i는 블록 벡터가 p이므로

$S_{P^{m-1},P}$에 의한 상호연결 형식이 회로에서 일정하다. 모든 행렬 M_i가 동일하므로 식 (24)은 다음과 같이 나타낼 수 있다.

$$M^{[m]} = \left[S_{P^{m-1},P} \cdot \left(I_{P^{m-1}} \otimes M \right) \right]^m \tag{25}$$

또한 식 (25)와 동일한 회로는 Kronecker 곱 연산의 회로 설계를 얻는 식 (17) 대신에 식 (18)를 식 (22)에 대입하면 다음과 같다.

$$\bigotimes_{i=m-1}^{0} M_i = \prod_{i=m-1}^{0} \left(I_{P^{m-1-i}} \otimes M_i \otimes I_{P^i} \right) \tag{26}$$

식 (26)의 우측 항에 식 (25)를 대입하면 식 (27)과 같다.

$$\bigotimes_{i=m-1}^{0} M_i = \prod_{i=m-1}^{0} \left(I_{P^{m-1-i}} \otimes S_{P^{i},P} \right) \cdot \left(I_{P^{m-1}} \otimes M_i \right) \cdot \left(I_{P^{m-1-i}} \otimes S_{P,P^{i}} \right) \tag{27}$$

앞에서 논한 Perfect Shuffle 기법과 Kronecker 곱과의 관계를 이용한 유한체 $GF(p^m)$상의 다치 신호처리 입출력 상호연결 방법에 대한 예를 들면 다음과 같다.

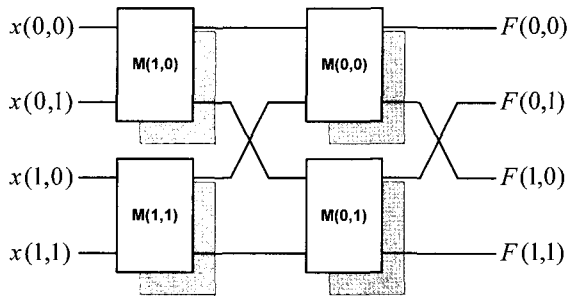
[예] $p=2$이고 $m=2$인 $GF(2^2)$상의 $F=[M_i \cdot x]$ 함수식을 식 (27)에 의하여 연산하면 다음과 같다.

$$\begin{aligned} \bigotimes_{i=m-1}^{0} M_i &= \bigotimes_{i=2-1}^{0} M_i = M_1 \otimes M_0 \\ &= \prod_{i=2-1}^{0} \left(I_{2^{2-1-i}} \otimes S_{2^i,2} \right) \cdot \left(I_{2^{2-1}} \otimes M_i \right) \cdot \left(I_{2^{2-1-i}} \otimes S_{2,2^i} \right) \\ &= S_{2,2} \cdot \left(I_2 \otimes M_1 \right) \cdot S_{2,2} \cdot \left(I_2 \otimes M_0 \right) \end{aligned} \tag{28}$$

식 (28)에 의하여 기본 셀을 상호연결하면 (그림 3)과 같다.

3.3. 다치 논리함수의 회로설계

이 절에서는 앞 절에서 논한 Perfect Shuffle 기법과 Kronecker 곱을 이용하여 다치 Reed-Muller 전개식에 대한 다치 논리 회로의 설계를 논한다.



(그림 3) $GF(2^2)$상에서 입출력 상호연결

(1) 4치 Reed-Muller 전개식의 회로설계

$GF(p^m)$상에서 p치 n변수 함수는 일반적인 Reed-Muller 전개식으로 식 (29)와 같이 표현할 수 있다.

$$F(x_0, x_1, \cdots, x_n) = \sum_{i=0}^{p^m-1} c_i \cdot \left[\prod_{j=0}^{n} x_j^{i \cdot j} \right] \tag{29}$$

여기서 $c_i \in GF(p)$이고 $x_j^{i \cdot j}$는 변수 x_j의 $i \cdot j$의 승수(power)이다.

식 (29)에서 $p=4$ 단일변수 Reed-Muller 전개식이 다음과 같다.

$$\begin{aligned} F(x) &= \sum_{i=0}^{4-1} c_i \cdot x^i \\ &= c_0 \oplus c_1 \cdot x \oplus c_2 \cdot x^2 \oplus c_3 \cdot x^3 \end{aligned} \tag{30}$$

여기서 $c_i, x^i \in GF(4)$이고, $i=\{0,1,2,3\}$이다.

식 (30)에서 4치 단일변수 Reed-Muller 전개식의 계수 c_i의 변환계수 d_i를 구하면 식 (31)과 같다.

$$\begin{aligned} F(0) &= d_0 = c_0 \\ F(1) &= d_1 = c_0 \oplus c_1 \oplus c_2 \oplus c_3 \\ F(2) &= d_2 = c_0 \oplus 2 \cdot c_1 \oplus 3 \cdot c_2 \oplus c_3 \\ F(3) &= d_3 = c_0 \oplus 3 \cdot c_1 \oplus 2 \cdot c_2 \oplus c_3 \end{aligned} \tag{31}$$

여기서 $c_i, d_i \in GF(4)$이다. 변환계수들에 대하여 행렬로 나타내면 식 (32)와 같다.

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \tag{32}$$

식 (32)를 간단하게 표현하면 식 (33)와 같다.

$$[d_i] = [M] \cdot [c_i] \tag{33}$$

식 (33)에서 변환행렬 M은 함수영역을 연산영역으로 변환하며 식 (34)와 같다.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 1 \end{bmatrix} \tag{34}$$

식 (33)로부터 Reed-Muller 전개식의 계수 c_i를 식 (35)와 같이 유도할 수 있다.

$$[c_i]=[M^{-1}]\cdot[d_i]=[T]\cdot[d_i] \tag{35}$$

식 (35)에서 역변환행렬 $T=M^{-1}$이 식 (36)과 같다.

$$\begin{aligned} c_0 &= d_0 \\ c_1 &= d_1 \oplus 3\cdot d_2 \oplus 2\cdot d_3 \\ c_2 &= d_1 \oplus 2\cdot d_2 \oplus 3\cdot d_3 \\ c_3 &= d_0 \oplus d_1 \oplus d_2 \oplus d_3 \end{aligned} \tag{36}$$

식 (36)을 행렬식으로 표현하면 식 (37)과 같다.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{37}$$

식 (37)을 간단하게 표현하면 식 (38)과 같다.

$$[c_i]=[T]\cdot[d_i] \tag{38}$$

식 (38)에서 역변환행렬 T는 연산영역에서 함수영역으로 변환하며 식 (39)과 같다.

$$T=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{39}$$

(그림 1)의 $GF(4)$의 가산회로와 (그림 2)의 $GF(4)$의 승산회로를 사용하여 4치 Reed-Muller 전개식의 변환행렬 M과 역변환행렬 T에 의한 식 (32)과 식 (37)을 실현하는 기본 셀을 구성하면 (그림 4)와 같다. (그림 4)의 (a)는 4치 Reed-Muller 전개식의 계수 c_i를 이용하여 변환계수 d_i를 구하는 변환행렬 M을 실현하는 기본 셀이고, (그림 4)의 (b)는 역변환행렬 T를 실현하는 기본 셀이다.

(2) 4치 2변수 Reed-Muller 전개식의 회로설계

4치 2변수 Reed-Muller 전개식은 식 (29)에 의해서 다음과 같이 나타낼 수 있다.

$$\begin{aligned} F(x_1,x_0) &= c_0 \oplus c_1\cdot x_0 \oplus c_2\cdot x_0^2 \oplus c_3\cdot x_0^3 \oplus c_4\cdot x_1 \oplus c_5\cdot x_1x_0 \\ &\oplus c_6\cdot x_1x_0^2 \oplus c_7\cdot x_1x_0^3 \oplus c_8\cdot x_1^2 \oplus c_9\cdot x_1^2x_0 \oplus c_{10}\cdot x_1^2x_0^2 \\ &\oplus c_{11}\cdot x_1^2x_0^3 \oplus c_{12}\cdot x_1^3 \oplus c_{13}\cdot x_1^3x_0 \oplus c_{14}\cdot x_1^3x_0^2 \oplus c_{15}\cdot x_1^3x_0^3 \end{aligned} \tag{40}$$

여기서 $c_i, x_1^i, x_0^i \in GF(4)$이다.

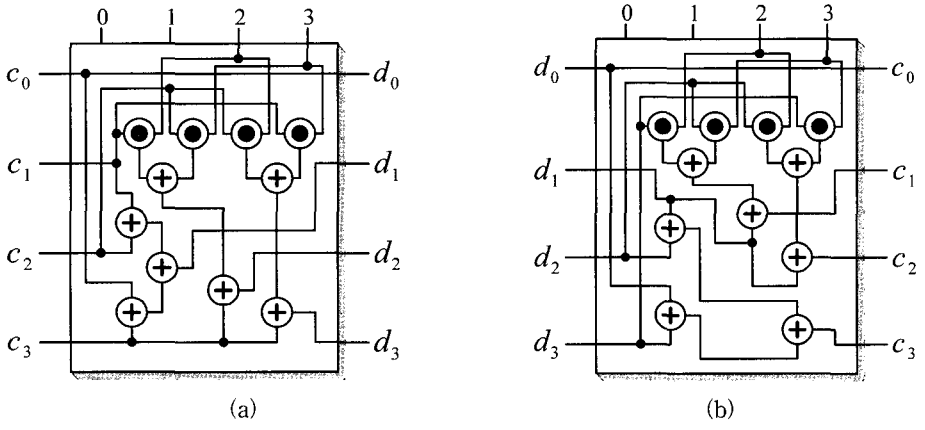
식 (40)에서 4치 2변수 Reed-Muller 전개식의 변환행렬 M_i는 식 (34)의 M을 Kronecker 곱하여 식 (41)과 같이 구할 수 있다.

$$\bigotimes_{i=0}^{2-1} M_i = \begin{bmatrix} M^{[i]} & 0 & 0 & 0 \\ M^{[i]} & M^{[i]} & M^{[i]} & M^{[i]} \\ M^{[i]} & 2\cdot M^{[i]} & 3\cdot M^{[i]} & M^{[i]} \\ M^{[i]} & 3\cdot M^{[i]} & 2\cdot M^{[i]} & M^{[i]} \end{bmatrix} \tag{41}$$

또한 역변환행렬 T_i는 식 (39)의 T를 Kronecker 곱하여 구하며 다음과 같다.

$$\bigotimes_{i=0}^{2-1} T_i = \begin{bmatrix} T^{[i]} & 0 & 0 & 0 \\ 0 & T^{[i]} & 3\cdot T^{[i]} & 2\cdot T^{[i]} \\ 0 & T^{[i]} & 2\cdot T^{[i]} & 3\cdot T^{[i]} \\ T^{[i]} & T^{[i]} & T^{[i]} & T^{[i]} \end{bmatrix} \tag{42}$$

식 (41)에서 2변수인 경우이므로 $M=M_1\otimes M_0$이고, 식 (42)은 $T=T_1\otimes T_0$이다. 식 (41)과 식 (42)을 앞 절에서 논한 Perfect Shuffle 기법과 Kronecker 곱에 의한 식 (27)을 이용하여 변환행렬 M을 연산하면 식 (43)과 같다.



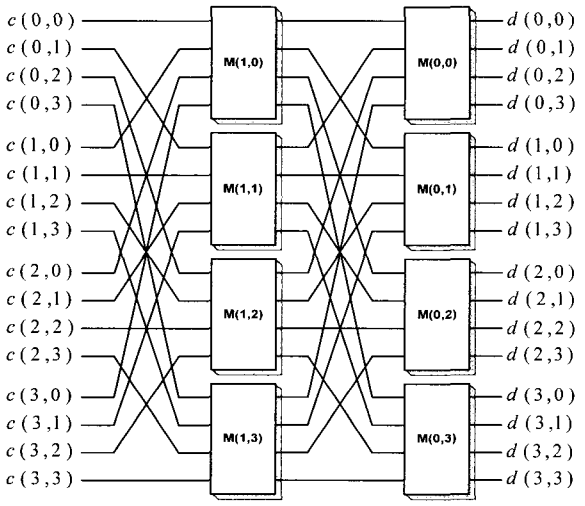
(그림 4) 기본 셀
(a) 변환행렬 M의 회로 (b) 역변환행렬 T의 회로

$$M=\bigotimes_{i=2-1}^{0} M_i = M_1 \otimes M_0$$
$$= \prod_{i=2-1}^{0} [(I_{4^{2-1-i}} \otimes S_{4^i,4}) \cdot (I_{4^i} \otimes M_i) \cdot (I_{4^{2-1-i}} \otimes S_{4,4^i})]$$
$$= S_{4,4} \cdot (I_4 \otimes M_1) \cdot S_{4,4} \cdot (I_4 \otimes M_0) \quad (43)$$

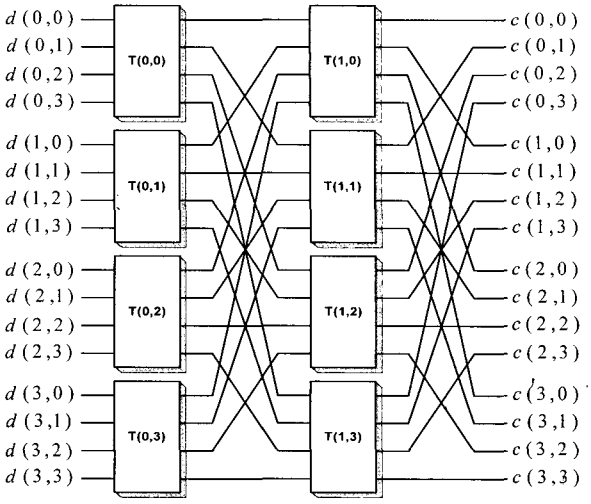
식 (43)에 의하여 실현한 회로가 (그림 5)와 같다. (그림 5)의 각 기본 셀의 내부회로는 $GF(4)$상의 가산회로와 승산회로에 의해 실현된 (그림 3)의 (a)의 기본 셀의 회로와 같다.

유사한 방법으로 Perfect Shuffle 기법과 Kronecker 곱에 의한 식 (27)을 이용하여 역변환행렬 T를 연산하면 다음과 같다.

$$T=\bigotimes_{i=0}^{2-1} T_i = T_0 \otimes T_1$$
$$= \prod_{i=0}^{2-1} [(I_{4^{2-1-i}} \otimes S_{4^i,4}) \cdot (I_{4^{2-1-i}} \otimes T_i) \cdot (I_{4^i} \otimes S_{4,4^{2-1-i}})]$$
$$= (I_4 \otimes T_0) \cdot S_{4,4} \cdot (I_4 \otimes T_1) \cdot S_{4,4} \quad (44)$$



(그림 5) 4치 2변수 RM 전개식의 변환회로 실현



(그림 6) 2변수 4치 RM 전개식의 역변환회로의 실현

식 (44)에 의하여 실현한 회로가 (그림 6)과 같다. (그림 6)의 각 기본 셀의 내부회로는 $GF(4)$상의 가산회로와 승산회로에 의해 실현된 (그림 3)의 (a)의 기본 셀의 회로와 같다.

4. 비교 및 검토

이 장에서는 제시한 다치 논리회로를 타 연구의 회로와 비교하였으며, 비교표가 <표 4>와 같으며, <표 5>는 $p=4$이고 $m=2$인 경우 가산회로와 승산회로를 비교한 표이다. <표 5>에서와 같이 제시한 다치 Reed-Muller 전개식에 의한 다치 논리회로는 Yang[8]의 고속 알고리즘에 의한 행렬 변환 방법과 승산회로 및 가산회로의 수는 동일하나 Yang의 방법은 레지스터를 포함하고 있어 변수가 증가 할수록 레지스터가 많이 사용된다. Rahardja[12]의 방법은 승산회로가 본 논문에서 제시한 방법보다 승산회로는 3배로 증가하고, 가산회로도 약 3배로 증가한다. Rahardja의 방법은 변수가 증가하면 회로의 소자수가 급격히 증가하는 단점이 있

〈표 4〉 다치 Reed-Muller 전개식의 다치 논리회로의 비교표

구 분	직접계산	Yang[8]	Rahardja[10]	Stankovic[11]		본 논문	
				GF	RMF	변환행렬	역변환행렬
승산회로	$p^m \cdot p^m z$	$4m \cdot p^{m-1}$	$\frac{3}{16}(16m \cdot p^m)$	$2m \cdot p^m$	$4m \cdot p^{m-1}$	$4m \cdot p^{m-1}$	$4m \cdot p^{m-1}$
가산회로	$p^m(p^m-1))$	$7m \cdot p^{m-1}$	$\frac{5}{16}(16m \cdot p^m)$	$7m \cdot p^{m-1}$	$7m \cdot p^{m-1}$	$7m \cdot p^{m-1}$	$7m \cdot p^{m-1}$
레지스터	$p^m(p^m+2)$	p^m	-	-	-	-	-

〈표 5〉 $p=4$이고 $m=2$인 다치 Reed-Muller 전개식의 비교표

구 분	직접계산	Yang[8]	Rahardja[10]	Stankovic[11]		본 논문	
				GF	RMF	변환행렬	역변환행렬
승산회로	256	32	96	64	32	32	32
가산회로	240	56	160	56	56	56	56
레지스터	288	16	-	-		-	-

다. Stankovic[11]의 방법은 GF인 경우 승산회로는 2배로 증가하며, 가산회로는 동일한 소자수를 보이며, RMF는 제시한 방법과 동일한 결과를 보인다.

그러므로 다치 Reed-Muller 전개식에 의한 다치 논리회로의 연산에서는 본 논문이 소자수면에서 다소 우수하며, 정보량의 처리에서도 우수하다. 제시한 Reed-Muller 전개식에 의한 다치 논리회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬 동작의 특징을 가진다.

5. 결 론

본 논문에서는 다치 Reed-Muller 전개식에 의한 다치 논리회로의 구성에 관한 한 가지 방법을 제시하였다. 먼저, Perfect Shuffle 기법과 Kronecker 곱에 의한 다치 논리함수의 입력과 출력의 상호연결 방법에 대하여 논하였고, $GF(4)$의 가산회로와 승산회로를 이용하여 다치 Reed-Muller 전개식의 변환행렬과 역변환행렬을 실행하는 기본 셀을 설계하였다. 이 기본 셀들과 Perfect Shuffle과 Kronecker 곱에 의한 입출력 상호연결 방법을 이용하여 다치 Reed-Muller 전개식에 의한 다치 논리회로를 구현하였다. 제시된 다치 Reed-Muller 전개식의 설계방법은 모듈구조를 기반으로 하여 행렬 변환하므로 동일한 함수에 대하여 타 방법과 비교하여 간단하고 회로의 가산회로와 승산회로를 줄이는데 매우 효과적이다.

본 논문에서 제시한 유한체 $GF(4^2)$상의 다치 Reed-Muller 전개식에 의한 변환행렬과 역변환행렬의 회로설계는 Yang의 방법과 연산회로의 수가 동일하나 Yang의 방법은 변수가 증가하면 레지스터가 증가한다. 또한 본 논문의 결과는 Rahardja 등의 방법보다 연산회로의 수가 약 3배로 감소하며, Stankovic 등의 방법보다는 승산회로의 수가 약 2배로 감소하며, 가산회로의 수는 동일하므로 이들이 제시한 알고리즘보다 약간 우수하다.

본 논문에서 제시한 다치 Reed-Muller 전개식에 의한 다치 논리회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬 동작의 특징을 가지므로 집적회로 실현에 적합하다. 또한 체계화된 다치 논리함수의 변환행렬 회로를 이용하여 신호처리와 화상처리 분야에서 특별한 계산을 요하거나 범용 컴퓨터의 고속화를 보조하는 전용 컴퓨터 및 인공지능에 이용되는 신경회로망 컴퓨터의 설계에 적용 가능하다.

참 고 문 헌

[1] S. L. Hurst, "Multiple-Valued Logic - Its Status and Future," *IEEE Trans. Comput.*, Vol. C-30, No. 9, pp.619-634, Sept. 1981.

[2] J. S. Lee and L. E. Miller, *CDMA Systems Engineering Handbook*, Artech House, Boston, 1998.

[3] D. Jankovic and Claudio Moraga, "Optimization of GF(4) Expressions Using the Extended Dual Polarity Property," *Proc. of 33rd International Symposium on Multiple-Valued Logic*, Tokyo, Japan, pp.50-55, May 2003.

[4] B. J. Falkowski and Cheng Fu, "Polynomial Expansions over GF(3) based on Fastest Transformation," *Proc. of 33rd International Symposium on Multiple-Valued Logic*, Tokyo, Japan, pp.40-45, May 2003.

[5] D. Jankovic, R. S. Stankovic and R. Drechsler, "Efficient Calculation of Fixed-Polarity Polynomial Expressions for Multiple-Valued Logic Functions," *Proc. of 32nd International Symposium on Multiple-Valued Logic*, Boston, Massachusetts, USA, pp.76-82, May 2002.

[6] B. J. Falkowski, C. C. Lozano and S. Rahardja, "Spectra Generation for Fixed-Polarity Reed-Muller Transform over GF(5)," *Proc. of 34th International Symposium on Multiple-Valued Logic*, Toronto, Canada, pp.177-1183, May 2004.

[7] B. J. Falkowski, C. C. Lozano and S. Rahardja, "Fast Optimization of Fixed-Polarity Reed-Muller Expansions over GF(5)," *Proc. of 34th International Symposium on Multiple-Valued Logic*, Toronto, Canada, pp.162-167, May 2004.

[8] F. Yang, "Fast Synthesis of Q-valued Functions Based on Modulo Algebra Expansions" *Proc. of 16th International Symposium on Multiple-Valued Logic.*, Virginia, USA, pp.36-41, May 1986.

[9] E. N. Zaitseva, T. G. Kalganova, and E. G. Kochergov, "Logical not Polynomial Forms to represent Multiple-Valued Functions," *Proc. of 26th International Symposium on Multiple-Valued Logic*, Santiago de Compostela, Spain, pp.302-307, May 1996

[10] S. Rahardja and B. J. Falkowski, "A New Algorithm to Compute Quarternary Reed-Muller Expansions," *Proc. of 30th International Symposium on Multiple-Valued Logic*, Portland, Oregon, pp.153-158, May 2000

[11] R. S. Stankovic, C. Moraga and J. Astola, "Derivatives for Multiple-Valued Functions Induced by Galois Field and Reed-Muller-Fourier Expressions," *Proc. of 34th International Symposium on Multiple-Valued Logic,* Toronto, Canada, pp.184-189, May 2004.

[12] B. J. Falkowski, C. C. Lozano and S. Rahardja, "Calculation of best fixed polarity Reed-Muller transform over GF(5)," *IEICE Electronics Express,* Vol 1, No.5, pp.92-97, June, 2004.

[13] M. Davio, "Kronecker Products and Shuffle Algebra," *IEEE Trans. Comput.*, Vol. C-30, No. 2, pp.116-125, Feb. 1981.

[14] A. N. Al-Rabadi, "Quantum Circuit Synthesis Using Classes of GF(3) Reversilbe Fast Spectral Transforms," *Proc. of 34th International Symposium on Multiple-Valued Logic*, Toronto, Canada, pp.87-93, May 2004.

[15] R. Lidl, H. Niederreiter, and P. M. Cohn, *Finite Fields*, Addison-Wesley Publishing Co., MA, USA, 1983.

성 현 경

e-mail : hkseong@sangji.ac.kr

1982년 인하대학교 전자공학과(공학사).

1984년 인하대학교 대학원 전자공학과(공학석사)

1991년 인하대학교 대학원 전자공학과(공학박사).

2005년~2006년 미국 Naval Postgraduate School 교환교수

1991년~현재 상지대학교 컴퓨터정보공학부 교수

관심분야 : Multiple-Valued Logic Design, Information & Coding Theory, Cryptography Theory & Security, RFID 설계 및 응용 등