
Perfect Shuffle에 의한 5치 논리회로의 구성에 관한 연구

성현경*

Study on Construction of Quinary Logic Circuits Using Perfect Shuffle

Hyeon-Kyeong Seong*

이 논문은 2009년도 상지대학교 교내 연구비 지원에 의해 연구되었음

요 약

본 논문에서는 Perfect Shuffle에 의한 5치 논리 회로의 구성에 관한 한 가지 방법을 제시하였다. 먼저, Perfect Shuffle 기법과 Kronecker 곱에 의한 5치 논리함수의 입출력 상호연결에 대하여 논하였고, $GF(5)$ 의 가산회로와 승산회로를 이용하여 5치 Reed-Muller 전개식의 변환행렬과 역변환행렬을 실행하는 기본 셀을 설계하였다. 이 기본 셀들과 Perfect Shuffle과 Kronecker 곱에 의한 입출력 상호연결 방법을 이용하여 5치 Reed-Muller 전개식에 의한 5치 논리 회로를 구현하였다. 제시된 5치 Reed-Muller 전개식의 설계방법은 모듈구조를 기반으로 하여 행렬변환을 이용하므로 동일한 함수에 대하여 타 방법과 비교하여 간단하고 회로의 가산회로와 승산회로를 줄이는데 매우 효과적이다. 제안된 5치 논리회로의 설계방법은 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬동작의 특징을 가진다.

ABSTRACT

In this paper, we present a method on the construction of quinary logic circuits using Perfect shuffle. First, we discussed the input-output interconnection of quinary logic function using Perfect Shuffle techniques and Kronecker product, and designed the basic cells of performing the transform matrix and the reverse transform matrix of quinary Reed-Muller expansions(QRME) using addition circuit and multiplication circuit of $GF(5)$. Using these basic cells and the input-output interconnection technique based on Perfect Shuffle and Kronecker product, we implemented the quinary logic circuit based on QRME. The proposed design method of QRME is simple and very efficient to reduce addition circuits and multiplication circuits as compared with other methods for same logic function because of using matrix transform based on modular structures. The proposed design method of quinary logic circuits is simple and regular for wire routing and possess the properties of concurrency and modularity of array.

키워드

5치 논리회로, Kronecker 곱, Perfect Shuffle, Reed-Muller 전개식

Key word

(Quinary Logic Circuits, Kronecker Product, Perfect Shuffle, Reed-Muller Expansions)

* 종신회원 : 상지대학교 (교신저자, hkseong@sangji.ac.kr)

접수일자 : 2010. 10. 18

심사완료일자 : 2010. 11. 16

I. 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 높아지고 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 RME에 대한 연구가 활발히 진행되고 있다[1,2].

RME는 AND 연산과 XOR 연산의 일반화에 따라서 여러 방법으로 다치 함수로 확장할 수 있으며, RME의 2진 논리함수는 FPGA와 같은 쉽게 이용할 수 있는 하드웨어를 사용하여 간단하고 쉽게 실현할 수 있다. n 변수 p 치 논리함수는 p^n 개의 서로 다른 RME으로 확장할 수 있다[3,4].

RME는 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 집약적이다. 그러나 최적의 RME를 찾는 데 있어서 일반화된 RME의 수는 변수의 수가 증가할 때 2진 논리함수의 경우보다 다치 논리함수의 경우에 더 급격하게 증가한다. 그러므로 많은 알고리즘이 다치 논리함수에 대하여 RME를 계산하기 위해 제시되었다[5].

다치 논리함수의 고속 변환 알고리즘을 제시한 Yang[6]은 Kronecker 곱을 이용하여 Q 치 함수의 모듈러 대수 전개식의 행렬 변환 알고리즘이 효과적인 계산 절차를 가지며, 임의의 3차 변환에서 입력 변수를 증가하므로 연산이 감소함을 보였다. Zaitseva 등[7]은 이산 직교 변환 행렬을 이용하여 다치 논리함수를 표현하는 다항식을 논리적으로 합성하는 방법을 제시하였으며, 제시된 다치 논리함수의 다항식 합성 방법은 다치 논리

함수의 성질을 조사하는데 이용될 수 있음 보였다. Rahardja 등[8]은 Reed-Muller 전개식의 4차 스위칭 함수의 행렬을 계산하는 새로운 알고리즘을 제시하였으며, 계수 행렬이 순환 정방행렬 행렬에 의해서 생성됨을 보였다. 제시된 알고리즘은 변수가 적을 때 효과적이거나 변수가 증가하면 소자수가 급격히 증가하는 단점이 있다.

Stankovic 등[9]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[10]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[11]가 제시한 Perfect Shuffle 기법과 Kronecker 곱을 이용하여 5차 논리함수의 입출력 상호연결 방법에 대하여 논하였으며, 5차 논리함수의 입출력 상호연결 방법을 이용하여 5차 RME에 의한 5차 논리회로를 구현하였다. 제시된 5차 RME의 설계방법은 모듈구조를 기반으로 하여 행렬변환을 이용하므로 회로의 가산회로와 승산회로를 줄이는데 매우 효과적임을 보인다.

II. 수학적 배경과 기본 회로

2.1. 수학적 배경

(1) Kronecker 곱의 성질

m 개의 행렬 M 의 Kronecker 곱은 결합법칙에 의해 다음과 같이 표현할 수 있다[11].

$$\begin{aligned}
 Z &= M_{m-1} \otimes M_{m-2} \otimes \cdots \otimes M_1 \otimes M_0 \\
 &= \bigotimes_{k=m-1}^0 M_k
 \end{aligned} \tag{1}$$

식 (1)에서 Z 의 엔트리를 $z(i, j)$ 라 하고, 행렬 M_k 의 엔트리를 $m_k(i_k, j_k)$ 라 하고, M_k 는 (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 i 와 j 는 유한체상의 원소들을 갖는 행 벡터열 $[r_{m-1}, \dots, r_1, r_0]$ 와 열 벡터열 $[c_{m-1}, \dots, c_1, c_0]$ 에 각각 대응하는 입력 벡터열 $[i_{m-1}, \dots, i_1, i_0]$ 와 출력 벡터열 $[j_{m-1}, \dots, j_1, j_0]$ 를 갖는다. 또한 m 개의 동일 원소를 갖는 Kronecker 곱을 M 의 m 차 Kronecker 먹이라 하고 $M^{[m]}$ 으로 나타낸다.

(2) Perfect Shuffle 기법의 성질

Shuffle기법은 순열(permutation: σ)로서 정의되며 임의의 순열로서 (b_1, b_0) -Shuffle은 인접행렬 S_{b_1, b_0} 로 나타내며, 차수 b_0, b_1 의 정방행렬이며, 행 j 와 열 i 에 속하는 인접행렬 원소 $S_{b_1, b_0}(j, i)$ 로 표현된다[11].

인접행렬 S_{b_1, b_0} 는 다음과 같다.

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

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_2 의 단위행렬이다.

식 (4)의 행렬 (j, i) 엔트리를 계산하기 위해서 입력 i 가 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 j 에 사상됨을 알 수 있다. 특히, 행렬은 임의의 영역에서 블록벡터 b 의 k 최소유효비트 (LSB)를 우측으로 한 위치 순환천이를 행한다.

$$I_{b_2} \otimes S_{b_1, b_0} \quad (5)$$

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_0 의 단위행렬이다.

특히, 행렬은 임의의 영역에서 블록벡터 b 의 k 최대 유효비트 (MSB)를 우측으로 한 위치 순환천이를 행한다.

$$S_{b_2, b_1} \otimes I_{b_0} \quad (7)$$

Shuffle 기법의 인수분해는 다음과 같다.

$$\textcircled{1} S_{b_2, b_1, b_0} = (S_{b_2, b_0, b_1}) \cdot (S_{b_2, b_1, b_0}) \\ = (S_{b_2, b_1, b_0}) \cdot (S_{b_2, b_0, b_1}) \quad (8)$$

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

2.2. GF(5)상의 기본 게이트 설계

이 절에서는 5치 논리회로의 설계에 필요한 GF(5)상의 기본 게이트를 T-게이트를 이용하여 설계한다. 5치 논리회로에서 논리 값은 집합 $R = \{0, 1, 2, 3, 4\}$ 이다.

(1) GF(5)상의 가산 게이트

유한체 $GF(p^m)$ 상에서 $GF(5)$ 는 $p = 5$ 이고 $m = 1$ 인 경우이다. 그러므로 $GF(5)$ 상의 원소들은 $p = 5$ 가 소수이므로 기약다항식에서 f_0 만으로 표시된다. 이때 $f_0 \in \{0, 1, 2, 3, 4\}$ 이며, $GF(5)$ 상의 원소는 표 1.과 같다.

표 1. GF(5)의 원소표
Table 1. Element table of GF(5)

f_0	$F(x)$	Symbol
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4

그러므로 $GF(5)$ 의 가산은 두 원소들을 가산한 후 $\text{mod}(5)$ 를 행하는 다음의 함수에 의해 이루어진다.

$$F(x,y) = (x + y) \text{mod}(5) \quad (10)$$

$GF(5)$ 의 가산표는 식 (10)에 의해 표 2.와 같다.

표 2. $GF(5)$ 의 가산표
Table 2. Addition table of $GF(5)$

		x				
+		0	1	2	3	4
y	0	0	1	2	3	4
	1	1	2	3	4	0
	2	2	3	4	0	1
	3	3	4	0	1	2
	4	4	0	1	2	3

T-게이트를 이용하여 $GF(5)$ 의 가산표인 표 2를 실현하는 가산회로가 그림 1과 같다. 그림 1의 (a)는 T-게이트에 의한 $GF(5)$ 의 가산회로이고, 그림 1의 (b)는 가산회로의 기호이다.

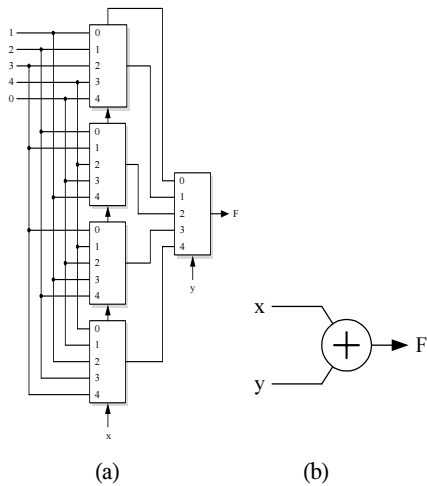


그림 1. T-게이트에 의한 $GF(5)$ 의 가산회로
(a) 가산회로 (b) 기호
Fig. 1. Additive circuit of $GF(5)$ using T-gate
(a) Addition circuit (b) symbol

(2) $GF(5)$ 상의 승산 게이트

$GF(5)$ 의 승산은 두 원소들을 승산한 후 $\text{mod}(5)$ 를 행하는 다음의 함수에 의해 이루어진다.

$$F(x,y) = (x \cdot y) \text{mod}(5) \quad (11)$$

$GF(5)$ 의 승산표는 식 (11)에 의해 표 3.과 같다.

표 3. $GF(5)$ 의 승산표
Table 3. Multiplication table of $GF(5)$

		x				
•		0	1	2	3	4
y	0	0	0	0	0	0
	1	0	1	2	3	4
	2	0	2	4	1	3
	3	0	3	1	4	2
	4	0	4	3	2	1

T-게이트를 이용하여 $GF(5)$ 의 승산표인 표 3을 실현하는 승산회로가 그림 2와 같다. 그림 2의 (a)는 T-게이트에 의한 $GF(5)$ 의 승산회로이고, 그림 2의 (b)는 승산회로의 기호이다.

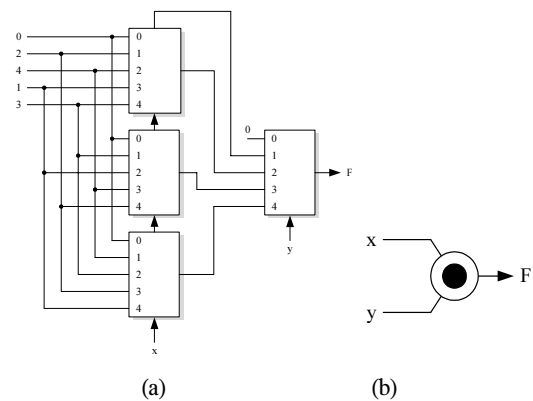


그림 2. T-게이트에 의한 $GF(5)$ 의 승산회로
(a) 승산회로 (b) 기호
Fig. 2 Multiplication circuit of $GF(5)$ using T-gate
(a) Multiplication circuit (b) symbol

III. 5차 논리 함수의 설계

이 장에서는 2장에서 서술한 수학적 배경을 이용하여 Reed-Muller 전개식에 의한 5차 논리회로의 설계에 대하여 논한다. 먼저, 5차 논리함수의 Perfect Shuffle 기법과 Kronecker 곱과의 상호 관계식을 논하고, 이를 이용하여 Reed-Muller 전개식에 의한 5차 논리회로의 설계를 구현한다.

3.1. 다차 논리함수의 Perfect Shuffle 기법과 Kronecker 곱과의 관계

m 개의 행렬 M 의 Kronecker 곱을 Z 라 하면 다음과 같이 표현된다.

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

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

[정리 1] 행렬 M_k 와 행렬 N_k 가 각각 (r_i, c_i) -행렬과 (r_j, c_j) -행렬이면 다음과 같다.

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

$$\begin{aligned} \textcircled{2} \bigotimes_{k=m-1}^0 M_k &= [S_{r_0, (r_1 \cdots r_{m-1})} \cdot (M_0 \otimes (S_{r_1, (r_2 \cdots r_{m-1})} \cdot (M_1 \otimes \cdots \otimes \\ & (S_{r_{m-2}, r_{m-1}} \cdot (M_{m-2} \otimes M_{m-1})) \cdot S_{c_{m-1}, c_{m-2}}) \cdots) \\ & \cdot S_{c_{m-1}, \cdots, c_0, c_1} \cdot S_{c_{m-1}, \cdots, c_0, c_1}] \end{aligned} \quad (14)$$

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_j 와 c_i, c_j 는 각각 행 벡터와 열 벡터이며, $i, j = \{0, 1, 2, \dots, m-1\}$ 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

P 라 하고 우측항의 (x_k, y_k) -엔트리를 $r_j \cdot r_i = w$, $c_j \cdot c_i = z$ 라 하면 식 (15)와 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 u 의 단일 값을 U 라 하면 $S_{r_j, r_i}(x_k, U) = 1$ 식 (3)에 의해 u 는 $x_k = U \cdot \sigma(r_j, r_i)$ 로 주어진다. 즉, $U = x_k \cdot \sigma(r_i, r_j) = x_{k_j} \cdot r_i + x_{k_i}$ 이다. 유사한 방법으로 v 의 단일 값을 V 라 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 위하여 임의 행렬들을 확장할 필요가 있다. 이 행렬의 확장은 단위 행렬과 Kronecker 곱으로 이루어진다.

[정리 2] 행렬 M_k 가 (r_i, c_i) -행렬이면 다음과 같다.

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

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

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

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

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

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

③은 식 (14)에 의해서 다음과 같이 구할 수 있다.

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

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 $I_{p_k} \otimes \left(\bigotimes_{k=m-1}^0 M_k \right) \otimes I_{p_k}$ 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 회로합성에서 승산회로가 증가한다. 이를 해결하기 위한 방법은 Kronecker 곱을 행렬곱으로 변환하여 연산하면 승산과정이 감소한다. 임의의 행렬과 단위행렬의 확장적인 정리 2를 이용하여 Kronecker 곱을 인수 분해함으로써 행렬곱의 연산이 가능하다.

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

$$\bigotimes_{k=m-1}^0 M_k = \prod_{j=k \cdot X}^0 (I_{(p_{m-1}, \dots, p_{j+1})} \otimes M_k \otimes I_{(p_{k-1}, \dots, p_0)}) \quad (20)$$

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_k 가 그 곱의 $k \cdot X$ 위치에 나타나도록 각각의 Kronecker 인수 M_k 를 m 개의 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_k 는 원소 $I_p^{[k]}$ 가 좌측 항에 곱해진 다. 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 분해함으로써 일반 행렬 곱으로 표현한다.

3.2. 다치 논리함수의 상호연결 방법

유한체 $GF(p^m)$ 상에서 p^m 개의 원소들을 갖는 m 변수 p 치인 M_i 의 Kronecker 곱에 대한 인수분해인 정리 3을 다시 쓰면 다음과 같다.

$$\bigotimes_{i=m-1}^0 M_i = \prod_{i=m-1}^0 (I_{(p_{m-1}, \dots, p_{i+1})} \otimes M_i \otimes I_{(p_{i-1}, \dots, p_0)}) \quad (22)$$

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.

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

식 (24)는 블록 벡터 $[p_{m-1}, p_{m-2}, \dots, p_1, p_0]$ 에서 입력 벡터열 $[i_{m-1}, i_{m-2}, \dots, i_1, i_0]$ 가 주어지면 $(I_{P_i} \otimes M_i)$ 의 연속적 실행에 의해 식 (24)의 좌측 항에서 나타나는 Kronecker 곱의 실행을 대체하며, 각 입력 벡터열의 i 번째 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_i 는 블록 벡터가 p 이므로 $S_{p^{[m-1]}, p}$ 에 의한 상호연결 형식이 회로에서 일정하다. 모든 행렬 M_i 가 동일하므로 식 (24)은 다음과 같이 나타낼 수 있다.

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

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

$$\bigotimes_{i=m-1}^0 M_i = \prod_{i=m-1}^0 (I_{p^{[m-1-i]}} \otimes M_i \otimes I_{p^{[i]}}) \quad (26)$$

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

$$\bigotimes_{i=m-1}^0 M_i = \prod_{i=m-1}^0 (I_{p^{[m-1-i]}} \otimes S_{p^{[i]}, p} \cdot (I_{p^{[m-1]}} \otimes M_i) \cdot (I_{p^{[m-1-i]}} \otimes S_{P_i, p^{[i]}}) \quad (27)$$

3.3 5차 논리함수의 회로설계

이 절에서는 앞 절에서 논한 Perfect Shuffle 기법과 Kronecker 곱을 이용하여 Reed-Muller 전개식에 대한 5차 논리회로의 설계를 논한다.

(1) 5차 Reed-Muller 전개식의 회로설계

$GF(p^m)$ 상에서 p 차 n 변수 함수는 일반적인 Reed-Muller 전개식으로 식 (28)과 같이 표현할 수 있다.

$$F(x_0, x_1, \dots, x_n) = \sum_{i=0}^{p^m-1} c_i \cdot \left[\prod_{j=0}^n x_j^{i \cdot j} \right] \quad (28)$$

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

식 (28)에서 $p = 5$ 단일변수 Reed-Muller 전개식이 다음과 같다.

$$\begin{aligned} F(x) &= \sum_{i=0}^{5-1} c_i \cdot x^i \\ &= c_0 \oplus c_1 x^1 \oplus c_2 x^2 \oplus c_3 x^3 \oplus c_4 x^4 \end{aligned} \quad (29)$$

여기서 $c_i, x^i \in GF(5)$ 이고, $i = \{0, 1, 2, 3, 4\}$ 이다. 식 (29)에서 5차 단일변수 Reed-Muller 전개식의 계수 c_i 의 변환계수 d_i 를 구하면 식 (30)과 같다.

$$\begin{aligned} F(0) &= d_0 = c_0 \\ F(1) &= d_1 = c_0 \oplus c_1 \oplus c_2 \oplus c_3 \oplus c_4 \\ F(2) &= d_2 = c_0 \oplus 2c_1 \oplus 4c_2 \oplus 3c_3 \oplus c_4 \\ F(3) &= d_3 = c_0 \oplus 3c_1 \oplus 4c_2 \oplus 2c_3 \oplus c_4 \\ F(4) &= d_4 = c_0 \oplus 4c_1 \oplus c_2 \oplus 4c_3 \oplus c_4 \end{aligned} \quad (30)$$

여기서 $c_i, d_i \in GF(5)$ 이다. 변환계수들에 대하여 행렬로 나타내면 식 (31)과 같다.

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 1 \\ 1 & 3 & 4 & 2 & 1 \\ 1 & 4 & 1 & 4 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \quad (31)$$

식 (31)를 간단하게 표현하면 식 (32)와 같다.

$$[d_i] = [M] \cdot [c_i] \quad (32)$$

식 (32)에서 변환행렬 M 은 함수영역을 연산영역으로 변환하며 식 (33)과 같다.

$$[M] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 1 \\ 1 & 3 & 4 & 2 & 1 \\ 1 & 4 & 1 & 4 & 1 \end{bmatrix} \quad (33)$$

식 (33)로부터 Reed-Muller 전개식의 계수 c_i 를 식 (34)와 같이 유도할 수 있다.

$$[c_i] = [M^{-1}] \cdot [d_i] = [T] \cdot [d_i] \quad (34)$$

식 (34)에서 역변환행렬 $[T] = [M^{-1}]$ 이며, 계수 c_i 를 구하면 식 (35)와 같다.

$$\begin{aligned} c_0 &= d_0 \\ c_1 &= 4d_1 \oplus 2d_2 \oplus 3d_3 \oplus d_4 \\ c_2 &= 4d_1 \oplus d_2 \oplus d_3 \oplus 4d_4 \\ c_3 &= 4d_1 \oplus 3d_2 \oplus 2d_3 \oplus d_4 \\ c_4 &= 4d_0 \oplus 4d_1 \oplus 4d_2 \oplus 4d_3 \oplus 4d_4 \end{aligned} \quad (35)$$

식 (35)을 행렬식으로 표현하면 식 (36)과 같다.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 3 & 1 \\ 0 & 4 & 1 & 1 & 4 \\ 0 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \quad (36)$$

식 (36)을 간단하게 표현하면 식 (37)과 같다.

$$[c_i] = [T] \cdot [d_i] \quad (37)$$

식 (37)에서 역변환행렬 T 는 연산영역에서 함수영역으로 변환하며 식 (38)과 같다.

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 3 & 1 \\ 0 & 4 & 1 & 1 & 4 \\ 0 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \quad (38)$$

그림 1의 $GF(5)$ 의 가산회로와 그림 2의 $GF(5)$ 의 승산회로를 사용하여 5차 Reed-Muller 전개식의 변환행렬 M 과 역변환행렬 T 에 의한 식 (31)과 식 (36)을 실현하는

기본 셀을 구성하면 그림 3과 같다. 그림 3의 (a)는 5차 Reed-Muller 전개식의 계수 c_i 를 이용하여 변환계수 d_i 를 구하는 변환행렬 M 을 실현하는 기본 셀이고, 그림 3의 (b)는 역변환행렬 T 를 실현하는 기본 셀이다.

(2) 5차 2변수 Reed-Muller 전개식의 회로설계

5차 2변수 Reed-Muller 전개식은 식 (28)에 의해서 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 F(x_0, x_1) = & c_0 \oplus c_1 x^1 \oplus c_2 x^2 \oplus c_3 x^3 \oplus c_4 x^4 \oplus c_5 x_1 \\
 & \oplus c_6 x_1 x_0 \oplus c_7 x_1 x_0^2 \oplus c_8 x_1 x_0^3 \oplus c_9 x_1 x_0^4 \oplus c_{10} x_1^2 \\
 & \oplus c_{11} x_1^2 x_0 \oplus c_{12} x_1^2 x_0^2 \oplus c_{13} x_1^2 x_0^3 \oplus c_{14} x_1^2 x_0^4 \oplus c_{15} x_1^3 \\
 & \oplus c_{16} x_1^3 x_0 \oplus c_{17} x_1^3 x_0^2
 \end{aligned} \tag{39}$$

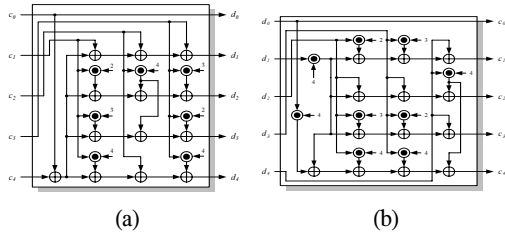


그림 3. 기본 셀 (a) 변환행렬 M 의 회로
(b) 역변환행렬 T 의 회로

Fig. 3. The basic cell (a) circuit of transform matrix M
(b) circuit of reverse transform matrix T

여기서 $c_i, x_1^i, x_0^i \in GF(5)$ 이다.

식 (39)에서 5차 2변수 Reed-Muller 전개식의 변환행렬 M_i 는 식 (33)의 M 을 Kronecker 곱하여 식 (40)과 같이 구할 수 있다.

$$\bigotimes_{i=0}^{2-1} M_i = \begin{bmatrix} M^{[i]} & 0 & 0 & 0 & 0 \\ M^{[i]} & M^{[i]} & M^{[i]} & M^{[i]} & M^{[i]} \\ M^{[i]} & 2M^{[i]} & 4M^{[i]} & 3M^{[i]} & M^{[i]} \\ M^{[i]} & 3M^{[i]} & 4M^{[i]} & 2M^{[i]} & M^{[i]} \\ M^{[i]} & 4M^{[i]} & M^{[i]} & 4M^{[i]} & M^{[i]} \end{bmatrix} \tag{40}$$

또한 역변환행렬 T_i 는 식 (38)의 T 를 Kronecker 곱하여 구하며 다음과 같다.

$$\bigotimes_{i=0}^{2-1} T_i = \begin{bmatrix} T^{[i]} & 0 & 0 & 0 & 0 \\ 0 & 4T^{[i]} & 2T^{[i]} & 3T^{[i]} & T^{[i]} \\ 0 & 4T^{[i]} & T^{[i]} & T^{[i]} & 4T^{[i]} \\ 0 & 4T^{[i]} & 3T^{[i]} & 2T^{[i]} & T^{[i]} \\ 4T^{[i]} & 4T^{[i]} & 4T^{[i]} & 4T^{[i]} & 4T^{[i]} \end{bmatrix} \tag{41}$$

식 (40)에서 2변수인 경우이므로 $M = M_1 \otimes M_0$ 이고, 식 (41)은 $T = T_1 \otimes T_0$ 이다. 식 (40)과 식 (41)을 앞 절에서 논한 Perfect Shuffle 기법과 Kronecker 곱에 의한 식 (27)을 이용하여 변환행렬 M 을 연산하면 식 (42)와 같다.

$$\begin{aligned}
 M &= \bigotimes_{i=2-1}^0 M_i = M_1 \otimes M_0 \\
 &= \prod_{i=2-1}^0 [(Z_5^{2-1-i} \otimes S_{5,5}^{[i]}) \cdot (Z_5^{2-1-i} \otimes M) \cdot (Z_5^{2-1-i} \otimes S_{5,5}^{[i]})] \\
 &= S_{5,5} \cdot (I_5 \otimes M) \cdot S_{5,5} \cdot (I_5 \otimes M_0)
 \end{aligned} \tag{42}$$

식 (42)에 의하여 실현한 회로가 그림 4와 같다. 그림 4의 각 기본 셀의 내부회로는 $GF(5)$ 상의 가산회로와 승산회로에 의해 실현된 그림 3의 (a)의 기본 셀의 회로와 같다.

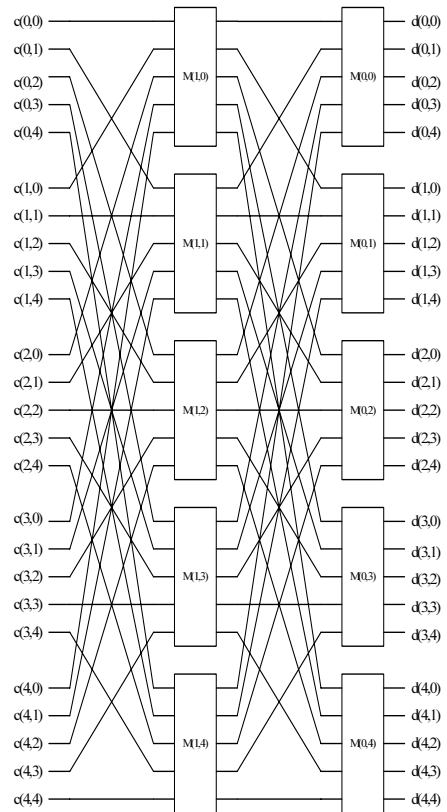


그림 4. 5차 2변수 RM 전개식의 변환회로 실현
Fig. 4. Realization of transform circuit on 2-variable 5-valued RM expansions

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

$$\begin{aligned}
 T &= \bigotimes_{i=2-1}^0 T_i = T_1 \otimes T_0 \\
 &= \prod_{i=2-1}^0 [(I_5^{2-1-i} \otimes S_{5,5}^{(i)}) \cdot (I_5^{2-1-i} \otimes T_i) \cdot (I_5^{2-1-i} \otimes S_{5,5}^{(i)})] \\
 &= (I_5 \otimes T_0) \cdot S_{5,5} \cdot (I_5 \otimes T_1) \cdot S_{5,5} \quad (43)
 \end{aligned}$$

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

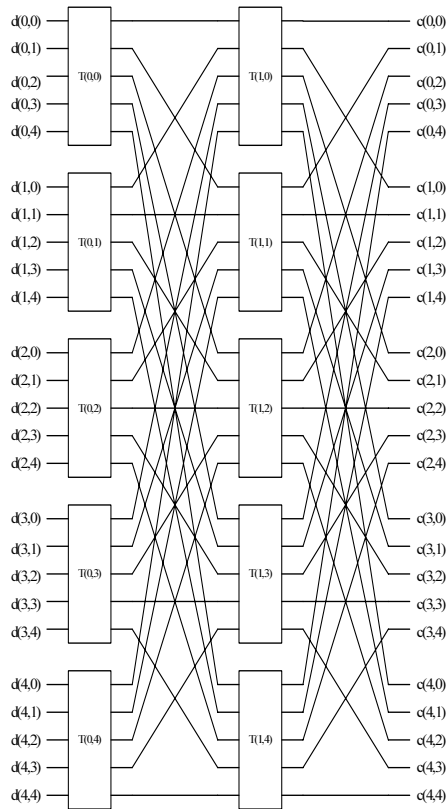


그림 5. 2변수 5치 RM 전개식의 역변환회로의 실현
Fig. 5. Realization of reverse-transform circuit on 2-variable 5-valued RM expansions

IV. 비교 및 검토

이 장에서는 제시한 5치 논리회로를 타 연구의 회로와 비교하였으며, 비교표가 표 4와 같으며, 표 5는 $p=5$ 이고 $m=2$ 인 경우 가산회로와 승산회로를 비교한 표이다. 표 5에서와 같이 제시한 5치 Reed-Muller 전개식에 의한 5치 논리회로는 Yang[6]의 고속 알고리즘에 의한 행렬변환 방법과 승산회로 및 가산회로의 수는 동일하나 Yang의 방법은 레지스터를 포함하고 있어 변수가 증가 할수록 레지스터가 많이 사용된다. Rahardja[10]의 방법은 승산회로가 본 논문에서 제시한 방법보다 승산회로는 약 3배로 증가하고, 가산회로는 약 2.3배로 증가한다. Rahardja의 방법은 변수가 증가하면 회로의 소자수가 급격히 증가하는 단점이 있다. Stankovic[9]의 방법은 GF인 경우 승산회로는 약 3배로 증가하며, 가산회로는 동일한 소자수를 보이며, RMF는 제시한 방법과 동일한 결과를 보인다. 역변환행렬인 경우는 본 논문의 경우 가산회로가 약간 더 필요하다.

그러므로 5치 Reed-Muller 전개식에 의한 5치 논리회로의 연산에서는 본 논문이 소자수면에서 다소 우수하며, 정보량의 처리에서도 우수하다. 제시한 Reed-Muller 전개식에 의한 5치 논리회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬 동작의 특징을 가진다.

V. 결론

본 논문에서는 5치 Reed-Muller 전개식에 의한 5치 논리회로의 구성에 관한 한 가지 방법을 제시하였다. 먼저, Perfect Shuffle 기법과 Kronecker 곱에 의한 다치 논리함수의 입력과 출력의 상호연결 방법에 대하여 논하였고, $GF(5)$ 의 가산회로와 승산회로를 이용하여 5치 Reed-Muller 전개식의 변환행렬과 역변환행렬을 실행하는 기본 셀을 설계하였다. 이 기본 셀들과 Perfect Shuffle과 Kronecker 곱에 의한 입출력 상호연결 방법을 이용하여 5치 Reed-Muller 전개식에 의한 5치 논리회로를 구현하였다. 제시된 5치 Reed-Muller 전개식의 설계방법은 모듈구조를 기반으로 하여 행렬 변

표 4. Reed-Muller 전개식의 5치 논리회로의 비교표
Table 4. Comparison of 5-valued logic circuit on Reed-Muller expansions

구 분	직접계산	Yang[6]	Rahardja[8]	Stankovic[9]		본 논문	
				GF	RMF	변환행렬	역변환행렬
승산회로	$p^m \cdot p^m$	$7m \cdot p^{m-1}$	$\frac{4}{25}(25m \cdot p^m)$	$4m \cdot p^m$	$7m \cdot p^{m-1}$	$7m \cdot p^{m-1}$	$9m \cdot p^{m-1}$
가산회로	$p^m(p^m - 1)$	$13m \cdot p^{m-1}$	$\frac{6}{25}(25m \cdot p^m)$	$13m \cdot p^{m-1}$	$13m \cdot p^{m-1}$	$13m \cdot p^{m-1}$	$13m \cdot p^{m-1}$
레지스터	$p^m(p^m + 2)$	p^m	-	-	-	-	-

표 5. $p=5$ 이고 $m=2$ 인 5치 논리회로의 비교표
Table 5. Comparison of 5-valued logic circuit on $p=5$ and $m=2$

구 분	직접계산	Yang[6]	Rahardja[8]	Stankovic[9]		본 논문	
				GF	RMF	변환행렬	역변환행렬
승산회로	625	175	500	500	175	175	225
가산회로	600	325	750	325	325	325	325
레지스터	675	25	-	-	-	-	-

환하므로 동일한 함수에 대하여 타 방법과 비교하여 간단하고 회로의 가산회로와 승산회로를 줄이는데 매우 효과적이다.

본 논문에서 제시한 5치 Reed-Muller 전개식에 의한 5치 논리회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬 동작의 특징을 가지므로 집적회로 실현에 적합하다. 또한 체계화된 5치 논리함수의 변환행렬 회로를 이용하여 신호처리와 화상처리 분야에서 특별한 계산을 요하거나 범용 컴퓨터의 고속화를 보조하는 전용 컴퓨터 및 인공지능에 이용되는 신경회로망 컴퓨터의 설계에 적용 가능하다.

참고문헌

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

[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] 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
- [8] 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
- [9] 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.
- [10] 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.
- [11] M. Davio, "Kronecker Products and Shuffle Algebra," *IEEE Trans. Comput.*, Vol. C-30, No. 2, pp.116-125, Feb. 1981.
- [12] 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.



성현경(Hyeon-Kyeong Seong)

1982년 인하대학교 전자공학과
공학사

1984년 인하대학교 대학원
전자공학과 공학석사

1991년 인하대학교 대학원 전자공학과 공학박사

2005년 ~ 2006년 미국 Naval Postgraduate School
방문교수

1991년 ~ 현재 상지대학교 컴퓨터정보공학부 교수

※ 관심분야 : Multiple-Valued Logic Design, Computer
Architecture Design, Information & Coding Theory,
Cryptography Theory & Security, RFID/WSN 설계 및
응용 등