Perfect Shuffle에 의한 Reed-Muller 전개식에 관한 다치 논리회로의 설계

성 현 경†

요 약

본 논문에서는 Perfect Shuffle 기법과 Kronecker 곱에 의한 다치 신호처리회로의 입출력 상호연결에 대하여 논하였고, 다치 신호처리회로의 입출력 상호연결 방법을 이용하여 유한체 $GF(p^m)$ 상에서 다치 신호처리가 용이한 다치 Reed-Muller 전개식의 회로설계 방법을 제시하였다. 제시된 다치 신호처리회로의 입출력 상호연결 방법은 모듈구조를 기반으로 하여 행렬변환을 이용하면 회로의 가산게이트와 승산게이트를 줄이는데 매우 효과적임을 보인다. $GF(p^m)$ 상에서 다치 Reed-Muller 전개식에 대한 다치 신호처리회로의 설계는 $GF(3)$ 상의 기본 게이트들을 이용하여 다치 Reed-Muller 전개식의 변환행렬과 역변환행렬을 실행하는 기본 셀을 설계하였고, 다치 신호처리회로의 입출력 상호연결 방법을 이용하여 기본 셀들을 상호연결하여 실현하였다. 제안된 다치 신호처리회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬동작의 특징을 가지므로 VLSI 화에 적합하다.

Design of Multiple-Valued Logic Circuits on Reed-Muller Expansions Using Perfect Shuffle

Hyeon-Kyeong Seong†

ABSTRACT

In this paper, the input-output interconnection method of the multiple-valued signal processing circuit using Perfect Shuffle technique and Kronecker product is discussed. Using this method, the circuit design method of the multiple-valued Reed-Muller Expansions (MRME) which can process the multiple-valued signal easily on finite fields $GF(p^m)$ is presented. The proposed input-output interconnection methods show that the matrix transform is an efficient and the structures are modular. The circuits of multiple-valued signal processing of MRME on $GF(p^m)$ design the basic cells to implement the transform and inverse transform matrix of MRME by using two basic gates on $GF(3)$ and interconnect these cells by the input-output interconnection technique of the multiple-valued signal processing circuits. The proposed multiple-valued signal processing circuits that are simple and regular for wire routing and possess the properties of concurrency and modularity are suitable for VLSI.

키워드 : 다치논리회로(multiple-valued logic circuits), Kronecker 곱(Kronecker product), Perfect Shuffle, 다치 Reed-Muller 전개식 (Multiple-Valued Reed-Muller expansions)

1. 서 론

현재 사용하고 있는 논리 시스템은 대부분이 2진 논리 이론을 기초로 하고 있으며, 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 2진 논리회로를 수행하는 부울 대수의 확장체인 유한체(finite fields ; GF)를 기초로 한 다치 논리이론의 연구가 활발히 진행되었다[1, 2].

다치 논리함수는 2진 논리함수에 비하여 동일 정보량을 처리하는데 상호연결의 복잡성을 감소시키며, 단위 면적당 높은 함수 기능 및 고밀도 실현인 VLSI/ULSI가 가능하다[3].

다치 논리함수의 기초가 되는 유한체는 스위칭이론, 오류정정부호, 디지털 신호처리 및 화상처리, 디지털 통신의 암호화 및 해독화를 요하는 보안통신 등에 많이 응용되고 있다. 특히 유한체는 신호처리와 화상처리 분야에서 특별한 계산을 요하거나 범용 컴퓨터 계산의 고속화를 보조하는 고성능 전

※ 이 논문은 2001년도 상지대학교 교내연구비 지원에 의해 연구되었음.
† 종신회원 : 상지대학교 컴퓨터 · 정보공학부 교수
논문접수 : 2002년 5월 7일, 심사완료 : 2002년 9월 3일

용 컴퓨터의 설계에 주목을 받고 있으며, Reed-Solomon 부호기 및 복호기의 VLSI 설계에 사용되고 있다[4-8].

한편, Pollard[9]는 DFT(Discrete Fourier Transform)와 유사한 변환을 p가 소수인 p^n개의 원소를 갖는 유한체 $GF(p^n)$에서 정의하였다. 이 정의는 유한체 $GF(p^n)$에서 위수 N에 의하는 N점 변환은 유한체 원소 r에 의하여 요구됨을 논하였다. Wang과 Zhu[10]는 유한체 $GF(p^m)$상에서 프리에 변환이 Reed-Solomon과 BCH(Bose-Chaudhuri-Hocquenghem) 코드의 부호와 복호에 요구되며, 유한체상에서 프리에 변환을 계산하기 위한 새로운 알고리듬을 제시하였다.

다치 논리함수의 고속 변환 알고리듬을 제시한 Yang[11]은 Kronecker 곱을 이용하여 Q치 함수의 모듈러 대수 전개식의 행렬 변환 알고리듬이 효과적인 계산 절차를 가지며, 임의의 3치 변환에서 입력 변수를 증가하므로 연산이 감소함을 보였다. Zaitseva 등[12]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. Stankovic 등[13]은 RMF(Reed-Muller Fourier) 전개식과 GF(Galois Fields) 표현식에 대한 변환행렬을 보였으며, 변수가 증가하면 RMF 전개식에 의한 다치 논리회로를 구현하는데 더 효과적임을 보였다. 또한 Stankovic 등[14]은 다양한 스위칭 함수에 대한 다양한 부울 표현식을 피보나치 상호연결망에서 사용되는 함수로 확장하였으며, 비트-레벨과 워드-레벨의 그래픽 표현으로서 대응되는 결정도를 보였다.

Rahardja 등[15]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. Harking 등[16]은 다치 논리함수에 대한 일반화된 Reed-Muller 전개식을 계산하는 새로운 방법을 제시하였으며, 이 방법은 행렬 승산을 행하지 않고 병렬 버터플라이 알고리즘을 이용하여 계수들을 계산할 수 있음을 보였다. 제시된 병렬 버터플라이 알고리즘은 다치 논리함수에 대한 일반화된 Reed-Muller 전개식을 수행할 수가 없으며, 단지 GF(2)에 대한 버터플라이 알고리즘을 보였다.

이들이 제시한 방법들은 다치 논리회로의 구성에서 승산과 가산의 연산과정이 증가하는 문제점이 있으며, 이러한 문제점을 해결하기 위하여 함수 기능이 높은 다치 논리함수의 계수들을 계산하는 다치 신호처리가 요구되었다.

본 논문에서는 Davio[17]가 제시한 Perfect Shuffle 기법과 Kronecker 곱을 이용하여 다치 신호처리회로의 입출력 상호연결 방법에 대하여 논하였으며, 이 방법은 Harking 등이 제시한 병렬 버터플라이 알고리즘을 이론적으로 증명하였다. 다치 신호처리회로의 입출력 상호연결 방법을 이용하여 유한체 $GF(p^m)$상에서 다치 신호처리가 용이한 Reed-Muller 전개식의 회로 설계 방법을 제시하였다. 제시된 다치 신호처리회로의 입출력 상호연결 방법은 모듈구조를 기반으로 하여 행렬변환을 이용하면 회로의 가산게이트와 승산 게이트를 줄이는데 매우 효과적임을 보인다.

2. 수학적 배경과 기본 게이트

2.1 수학적 배경

2.1.1 유한체의 성질

유한체 $GF(p^m)$은 p가 소수이고 m이 양의 정수인 p^m개의 원소들을 가지며, p^m개의 원소들을 갖는 기초체 $GF(p)$의 확대체이다. 즉 유한체 $GF(p)$는 $\{0, 1, 2, \cdots, p-1\}$의 원소들로 구성된다. $GF(p^m)$에서 모든 산술연산은 그 결과를 $\bmod(p)$ 연산으로 이루어지며, $GF(p^m)$의 0이 아닌 모든 원소들은 원시 원소 α에 의해 생성되며, α는 $GF(p^m)$의 원시기약다항식 $F(\alpha)=0$의 근이다[5-7].

$GF(p^m)$의 원시기약다항식은 다음과 같이 나타낸다.

$$F(\alpha)=\sum_{i=0}^{m-1} f_i \cdot \alpha^i \quad ; \quad \{f_i, \alpha^i\} \in GF(p^m) \tag{1}$$

여기서 f_i와 α^i는 $\{0, 1, 2, \cdots, p-1\}$ 중의 한 원소를 갖는다.

또한, $GF(p^m)$의 0이 아닌 원소들은 α의 멱(power)으로 표현이 가능하며 다음과 같다.

$$\{0, \alpha^1, \alpha^2, \cdots, \alpha^{p^m-2}, \alpha^{p^m-1}=1\} \in GF(p^m) \tag{2}$$

유한체 $GF(p^m)$의 유용한 성질들을 증명 없이 설명하면 다음과 같다.

1) $GF(p^m)$에서 임의의 한 원소 α에 대하여

$$\alpha^{p^m}=\alpha, \ \alpha^{p^m-1}=1 \quad ; \quad \alpha \in GF(p^m) \tag{3}$$

2) $GF(p^m)$에서 임의의 두 원소들 α와 β에 대하여

$$(\alpha+\beta)^{p^m}=\alpha^{p^m}+\beta^{p^m} \quad ; \quad \alpha, \beta \in GF(p^m) \tag{4}$$

3) $GF(p^m)$에서

$$\begin{aligned} \alpha^i \cdot \alpha^j &= \alpha^{(i+j) \bmod (p^m-1)} \\ &= \alpha^{r \bmod (p^m-1)} \quad ; \quad \alpha^i, \alpha^j, \alpha^r \in GF(p^m) \end{aligned} \tag{5}$$

이다.

2.1.2 Kronecker 곱의 성질

Kronecker 곱은 결합법칙에 의해 다음과 같이 표현할 수

있다[17].

$$Z = M_{m-1} \otimes M_{m-2} \otimes \cdots \otimes M_1 \otimes M_0 = \bigotimes_{k=m-1}^{0} M_k \quad (6)$$

식 (6)에서 Z의 엔트리를 $z(i, j)$라 하고, M_k의 엔트리를 $m_k(i_k, j_k)$라 하자. 그리고 M_k는 (r_k, c_k) 행렬이라 하면 다음과 같다.

$$z(i, j) = \prod_{k=0}^{m-1} m_k(i_k, j_k) \quad (7)$$

여기서 i와 j는 유한체 상의 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.

2.1.3 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, 행 j와 열 i에 속하는 인접행렬 원소 $S_{b_1, b_0}(j, i)$로 표현된다[18, 19].

인접행렬 S_{b_1, b_0}는 다음과 같다.

$$S_{b_1, b_0}(j, i) = \begin{cases} 1 & \text{if } j = i \cdot \sigma \\ 0 & otherwise \end{cases} \quad (8)$$

인접행렬과 Shuffle 기법사이에서 순열행렬을 나타내면 다음과 같다.

$$I_{b_2} \otimes S_{b_1, b_0} \quad (9)$$

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 (9)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, 행렬

$$I_{b^{[m-k]}} \otimes S_{b^{[k-1]}, b} \quad (10)$$

은 임의의 영역에서 블록벡터 b의 k 최소유효비트(LSB)를 우측으로 한 위치 순환천이를 행한다.

유사한 방법으로 순열행렬

$$S_{b_2, b_1} \otimes I_{b_0} \quad (11)$$

은 블록벡터 $[b_2, b_1]$내에서 분리되어 행하는 b_0 독립의 (b_2, b_1)-Shuffle로서 표현된다. 여기서 I_{b_0}는 블록 b_0의 단위행렬이다.

특히, 행렬

$$S_{b^{[k-1]}, b} \otimes I_{b^{[m-k]}} \quad (12)$$

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

Shuffle 기법의 인수분해는 다음과 같다.

1) $S_{b_2, b_1, b_0} = (S_{b_2, b_0, b_1}) \cdot (S_{b_2, b_1, b_0}) = (S_{b_2, b_1, b_0}) \cdot (S_{b_2, b_0, b_1})$ (13)

2) $S_{b_2, b_1, b_0} = (S_{b_2, b_0} \otimes I_{b_1}) \cdot (I_{b_2} \otimes S_{b_1, b_0})$ (14)

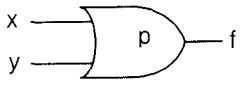
2.2 기본 게이트

이 절에서는 다치 논리함수의 신호처리 회로설계에 필요한 기본 게이트를 기호적으로 설명한다. 다치 신호처리회로에서 논리 값은 집합 $R = \{0, 1, 2, \cdots, p-1\}$이며, 다음과 같은 연산 함수를 표현할 수 있다.

2.2.1 가산게이트

$GF(p)$상에서 p치 2변수 x와 y에 대하여 가산을 수행하는 함수는 다음과 같이 표현할 수 있으며, (그림 1)은 식 (15)를 실현하는 $GF(p)$ 가산게이트를 나타낸다.

$$f = x \oplus y = (x+y) \bmod (p) \quad (15)$$

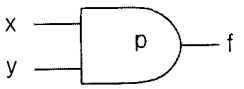


(그림 1) $GF(p)$ 가산게이트

2.2.2 승산게이트

$GF(p)$상에서 p치 2변수 x와 y에 대하여 승산을 수행하는 함수는 다음과 같이 표현할 수 있으며, (그림 2)는 식 (16)를 실현하는 $GF(p)$ 승산게이트를 나타낸다.

$$f = x \cdot y = (x \times y) \bmod (p) \quad (16)$$



(그림 2) $GF(p)$ 승산게이트

3. 다치 논리함수의 신호처리 설계

이 장에서는 2에서 서술한 수학적 배경을 이용하여 다치

논리함수의 신호처리회로 설계에 대하여 논한다. 먼저, 다치 논리함수의 Perfect Shuffle 기법과 Kronecker 곱과의 상호 관계식을 논하고, 이를 이용하여 다치 논리함수의 신호처리 회로 설계를 실현한다.

3.1 다치 논리함수의 Perfect Shuffle 기법과 Kronecker 곱과의 관계

유한체 $GF(p^m)$상에서 p^m개의 원소들을 갖는 Z의 Kronecker 곱은 다음과 같이 표현된다.

$$Z = M_{m-1} \otimes M_{m-2} \otimes \cdots \otimes M_1 \otimes M_0 = \overset{0}{\underset{k=m-1}{\otimes}} M_k \quad (17)$$

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

[정리 1] 유한체 $GF(p^m)$상에서 p^m개의 원소들을 갖는 M_k와 N_k가 각각 (r_i, c_i)-행렬과 (r_j, c_j)-행렬이면 다음과 같다.

$$1)\ \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right) \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} N_k\right) = S_{r_j, r_i} \cdot \left[\left(\overset{0}{\underset{k=m-1}{\otimes}} N_k\right) \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right)\right] \cdot S_{c_i, c_j} \quad (18)$$

$$2)\ \overset{0}{\underset{k=m-1}{\otimes}} M_k = [S_{r_0, (r_1 \cdots r_{m-1})} \cdot (M_0 \otimes (S_{r_1, (r_2 \cdots r_{m-1})} \cdot (M_1 \otimes \cdots \otimes (S_{r_{m-2}, r_{m-1}} \cdot (M_{m-2} \otimes M_{m-1}) \cdot S_{c_{m-1}, c_{m-2} \cdots}) \cdot S_{(c_{m-1} \cdots c_0), c_1} \cdot S_{(c_{m-1} \cdots c_1), c_0}] \quad (19)$$

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 $\left(\overset{0}{\underset{k=m-1}{\otimes}} N_k\right) \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right)$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.

$$\sum_{u=0}^{w-1} \sum_{v=0}^{z-1} S_{r_j, r_i}(x_k, u) \cdot P(u, v) \cdot S_{c_i, c_j}(v, y_k) \quad (20)$$

실제로 u의 단일 값을 U라 하면

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

식 (8)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 (8)에 의해 $V = y_k \cdot \sigma(c_i, c_j) = y_{k_j} \cdot c_i + y_{k_i}$이다. 그러므로 식 (20)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 (18)을 증명할 수 있다. Q.E.D

[정리 1]을 이용하여 블록벡터 p가 2인 두 행렬 M_1과 M_0의 Kronecker 곱의 변형된 연산을 행하는 예가 다음과 같다.

[예 1] 블록벡터 p가 2인 (2, 2)-행렬 M_1과 (2, 2)-행렬 M_0가 다음과 같다고 하자.

$$M_1 = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ 이고 } M_0 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ 이다.}$$

식 (19)에 의하여 두 행렬의 Kronecker 곱은 다음과 같이 표현할 수 있다.

$$M_1 \otimes M_0 = S_{2,2} \cdot [M_0 \otimes M_1] \cdot S_{2,2} = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} aA & aB & bA & bB \\ aC & aD & bC & bD \\ cA & cB & dA & dB \\ cC & cD & dC & dD \end{bmatrix} \cdot \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix} = \begin{bmatrix} Aa & Ab & Ba & Bb \\ Ac & Ad & Bc & Bd \\ Ca & Cb & Da & Db \\ Cc & Cd & Dc & Dd \end{bmatrix} \quad (21)$$

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] 유한체 $GF(p^m)$상에서 p^m개의 원소들을 갖는 M_k가 (r_i, c_i)-행렬이면 다음과 같다.

$$1)\ \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right) \otimes I_{p_{k_i}} = S_{p_{k_i}, r_i} \cdot \left[I_{p_{k_i}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right)\right] \cdot S_{c_i, p_{k_i}} \quad (22)$$

$$2)\ I_{p_{k_i}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right) \otimes I_{p_{k_i}} = S_{p_{k_i}, p_{k_i}, r_i} \cdot \left[I_{p_{k_i}, p_{k_i}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k\right)\right] \cdot S_{p_{k_i}, c_i, p_{k_i}} \quad (23)$$

3) $I_{p_{k_1}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k \right) \otimes I_{p_{k_1}}$

$$= (I_{p_{k_1}} \otimes S_{p_{k_1}, r_i}) \cdot \left[I_{p_{k_1}, p_{k_1}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k \right) \right] \cdot (I_{p_{k_1}} \otimes S_{c_i, p_{k_1}}) \quad (24)$$

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

[증명] 1)은 식 (16)에서 $\overset{0}{\underset{k=m-1}{\otimes}} N_k$ 대신에 $I_{p_{k_1}}$를 대입하여 구할 수 있다.

2)는 $(p_{k_1} \cdot r_i,\ p_{k_1} \cdot c_i)$-행렬 $I_{p_{k_1}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k \right)$을 식 (19)에 대입하여 구할 수 있다.

3)은 식 (19)에 의해서 다음과 같이 구할 수 있다.

$$I_{p_{k_1}} \otimes \left[\left(\overset{0}{\underset{k=m-1}{\otimes}} M_k \right) \otimes I_{p_{k_1}} \right] = I_{p_{k_1}} \otimes \left(S_{p_{k_1}, r_i} \cdot \left(I_{p_{k_1}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k \right) \cdot S_{c_i, p_{k_1}} \right) \right) \quad (25)$$

이 식을 인수 분해하여 구할 수 있다. Q.E.D

[정리 1]은 Perfect Shuffle 기법을 이용하여 Kronecker 곱의 교환법칙을 성립시키며, [정리 2]는 행렬 $\overset{0}{\underset{k=m-1}{\otimes}} M_k$을 Perfect Shuffle 기법에 의해 다양하게 표현할 수 있음을 나타낸다. [정리 2]에서 $I_{p_{k_1}} \otimes \left(\overset{0}{\underset{k=m-1}{\otimes}} M_k \right) \otimes I_{p_{k_1}}$는 입력과 출력의 연결 방법을 나타내며, 이 변환식은 $\overset{0}{\underset{k=m-1}{\otimes}} M_k$을 수행하는 블록벡터 연산자들의 집합으로 회로설계에 이용된다.

[정리 2]을 이용하여 블록벡터 p가 2인 행렬 M과 단위행렬 I_P의 관계를 연산하는 예가 다음과 같다.

[예 2] 블록벡터 p가 2인 (2, 2)-행렬 M과 단위행렬 I_P가 다음과 같다고 하자.

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{이고 } I_P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{이다.}$$

식 (23)에 의하여 두 행렬의 연산은 다음과 같이 표현할 수 있다.

$$I_P \otimes M \otimes I_P = S_{2,(2\cdot 2)} \cdot [I_{(2\cdot 2)} \otimes M] \cdot S_{(2\cdot 2),2}$$

$$= \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} A&B \\ C&D \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} \\ \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} A&B \\ C&D \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} \\ \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} A&B \\ C&D \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} \\ \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} 0&0 \\ 0&0 \end{bmatrix} & \begin{bmatrix} A&B \\ C&D \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} A&0&B&0&0&0&0&0 \\ 0&A&0&B&0&0&0&0 \\ C&0&D&0&0&0&0&0 \\ 0&C&0&D&0&0&0&0 \\ 0&0&0&0&A&0&B&0 \\ 0&0&0&0&0&A&0&B \\ 0&0&0&0&C&0&D&0 \\ 0&0&0&0&0&C&0&D \end{bmatrix} \quad (26)$$

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

[정리 3] 유한체 $GF(p^m)$상에서 p^m개의 원소들을 갖는 행렬곱으로 나타내기 위하여 $X \in \{0, 1, 2, \cdots, m-1\}$의 순열이라 할 때 (r_k, c_k)-행렬 M_k의 Kronecker 곱의 인수분해는 다음과 같다.

$$\overset{0}{\underset{k=m-1}{\otimes}} M_k = \overset{0}{\underset{j=k\cdot X}{\prod}} (I_{(p^{m-1}\cdots p^k+1)} \otimes M_k \otimes I_{(p^{k-1}\cdots p_0)}) \quad (27)$$

식 (27)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 (27)의 좌측 항에서 M_k가 그 곱의 $k \cdot X$ 위치에 나타나도록 각각의 Kronecker 인수 M_k를 m개의 인수들의 동일한 행렬 곱인 식 (28)로 대체한다.

$$I_{r_k}^{[m-1-k\cdot X]} \cdot M_k \cdot I_{c_k}^{[k\cdot X]} \quad (28)$$

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 3]은 단위행렬에의한 Kronecker 곱의 확장식을 이용하여 Kronecker 곱을 인수분해함으로서 일반 행렬곱으로 표현한다. [정리 3]에 의해 Kronecker 곱을 인수분해하여 행렬곱으로 연산하는 예가 다음과 같다.

[예 3] 블록벡터 $p = 2$이고 $m = 2$인 (2, 2)-행렬 M_k의 인수분해식이 식 (27)에 의하여 다음과 같다.

$$\overset{0}{\underset{k=m-1}{\otimes}} M_k = \overset{0}{\underset{j=k\cdot X}{\prod}} (I_{(p^{m-1}\cdots p^k+1)} \otimes \tilde{M}_k \otimes I_{(p^{k-1}\cdots p_0)}) = M_1 \otimes M_0 = (M_1 \otimes I_{P_0}) \cdot (I_{P_1} \otimes M_0) = (I_{P_1} \otimes M_0) \cdot (M_1 \otimes I_{P_0}) \quad (29)$$

행렬 $M_1 = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$이고 행렬 $M_0 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$이라고 하자.

그러면 인수분해에 의한 행렬 연산은 다음과 같다.

$$\begin{aligned} M_1 \otimes M_0 &= (M_1 \otimes I_{P_0}) \cdot (I_{P_1} \otimes M_0) \\ &= (M_1 \otimes I_2) \cdot (I_2 \otimes M_0) \\ &= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\ &= \begin{bmatrix} A & 0 & B & 0 \\ 0 & A & 0 & B \\ C & 0 & D & 0 \\ 0 & C & 0 & D \end{bmatrix} \cdot \begin{bmatrix} a & b & 0 & 0 \\ c & d & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix} \end{aligned} \quad (30)$$

이러한 인수분해식은 Perfect Shuffle 기법을 이용하여 다치 신호처리함수의 회로합성에서 모듈들의 상호교환과 모듈들의 다단 배열을 나타낼 수 있다.

3.2 다치 신호처리회로의 연결방법

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

$$\overset{0}{\underset{k=m-1}{\otimes}} M_k = \overset{0}{\underset{j=k \cdot X}{\prod}} (I_{(p^m-1 \cdots p^k+1)} \otimes M_k \otimes I_{(p^k-1 \cdots p_0)}) \quad (31)$$

표기를 간단히 하기 위하여 다음과 같이 정의한다.

$$P_i = \prod_{j=0,\, j \neq i}^{m-1} P_j \quad (32)$$

그러므로 식 (31)은 다음과 같다.

$$\overset{0}{\underset{i=m-1}{\otimes}} M_i = \overset{0}{\underset{i=m-1}{\prod}} [S_{P_i, P_i} \cdot (I_{P_i} \otimes M_i)] \quad (33)$$

식 (33)은 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 $(I_{P_i} \otimes M_i)$의 연속적 실행에 의해 식 (33)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 (33)은 다음과 같이 나타낼 수 있다.

$$M^{[m]} = [S_{P^{[m-1]}, P} \cdot (I_{P^{[m-1]}} \otimes M)]^m \quad (34)$$

또한 식 (34)와 동일한 회로는 Kronecker 곱 연산의 회로설계를 얻는 식 (23) 대신에 식 (24)를 식 (31)에 대입하면 다음과 같다.

$$\overset{0}{\underset{i=m-1}{\otimes}} M_i = \overset{0}{\underset{i=m-1}{\prod}} (I_{P^{[m-1-i]}} \otimes M_i \otimes I_{P^{[i]}}) \quad (35)$$

식 (35)의 우측 항에 식 (34)를 대입하면 식 (36)과 같다.

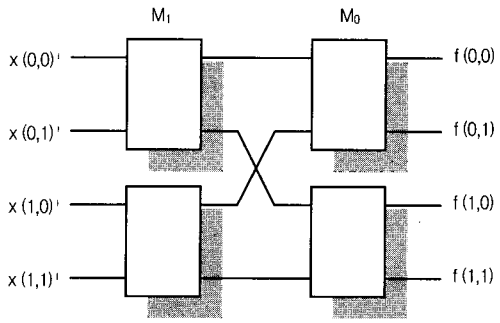
$$\begin{aligned} \overset{0}{\underset{i=m-1}{\otimes}} M_i = \overset{0}{\underset{i=m-1}{\prod}} &(I_{P^{[m-1-i]}} \otimes S_{P^{[i]}, P}) \cdot (I_{P^{[m-1]}} \otimes M_i) \cdot \\ &(I_{P^{[m-1-i]}} \otimes S_{P, P^{[i]}}) \end{aligned} \quad (36)$$

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 4] $p=2$이고 $m=2$인 $GF(2^2)$상의 $F]=[M_i] \cdot x]$ 함수식을 식 (36)에 의하여 연산하면 다음과 같다.

$$\begin{aligned} \overset{0}{\underset{i=m-1}{\otimes}} M_i &= M_1 \otimes M_0 \\ &= \overset{0}{\underset{i=2-1}{\prod}} (I_{2^{[2-1-i]}} \otimes S_{2^{[i]}, 2}) \cdot (I_{2^{[2-1]}} \otimes M_i) \\ &\quad \cdot (I_{2^{[2-1-i]}} \otimes S_{2, 2^{[i]}}) \\ &= S_{2,2} \cdot (I_2 \otimes M_1) \cdot S_{2,2} \cdot (I_2 \otimes M_0) \end{aligned} \quad (37)$$

식 (37)에 의하여 기본 셀을 상호연결하면 (그림 3)과 같다.



(그림 3) $GF(2^2)$상에서 기본 셀의 상호연결

3.3 다치 신호처리회로 설계

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

3.3.1 3치 Reed-Muller 전개식의 신호처리회로 설계

$GF(p^m)$상에서 p치 n변수 함수는 일반적인 Reed-Muller 전개식으로 식 (38)과 같이 표현할 수 있다.

$$F(x_1, x_2, \cdots, x_n) = \sum_{i=0}^{p^m-1} c_i \cdot \left[\prod_{j=1}^{n} x_j^{i \cdot j} \right] \quad (38)$$

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

식 (38)에서 $p=3$인 단일변수 Reed-Muller 전개식이 다음과 같다.

$$F(x) = \sum_{i=0}^{3-1} c_i \cdot x^i$$
$$= c_0 \oplus c_1 \cdot x \oplus c_2 \cdot x^2 \quad (39)$$

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

식 (39)에서 3치 단일변수 Reed-Muller 계수 c_i의 변환계수 d_i를 구하면 식 (40)과 같다.

$$d_0 = c_0$$
$$d_1 = c_0 \oplus c_1 \oplus c_2$$
$$d_2 = c_0 \oplus 2 \cdot c_1 \oplus c_2 \quad (40)$$

여기서 $c_i, d_i \in GF(3)$이다. 변환계수들에 대하여 행렬로 나타내면 식 (41)과 같다.

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \quad (41)$$

식 (41)을 간단하게 표현하면 식 (42)와 같다.

$$[d_i] = [M] \cdot [c_i] \quad (42)$$

식 (42)에서 변환행렬 M은 함수영역을 연산영역으로 변환하며 식 (43)과 같다.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \quad (43)$$

식 (42)로부터 Reed-Muller 전개식의 계수 c_i를 식 (44)와 같이 유도할 수 있다.

$$[c_i] = [M^{-1}] \cdot [d_i] = [T] \cdot [d_i] \quad (44)$$

식 (44)에서 역변환행렬 $T = M^{-1}$이 식 (45)와 같다.

$$c_0 = d_0$$
$$c_1 = 2 \cdot d_1 \oplus d_2$$
$$c_2 = 2 \cdot d_0 \oplus 2 \cdot d_1 \oplus 2 \cdot d_2 \quad (45)$$

식 (45)를 행렬식으로 표현하면 식 (46)과 같다.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix} \quad (46)$$

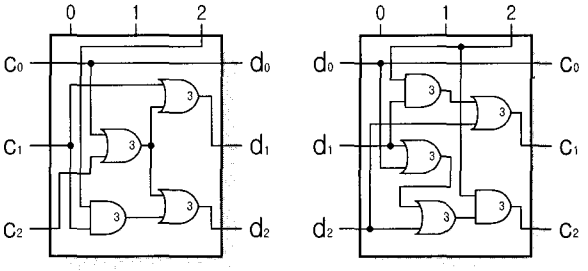
식 (46)을 간단하게 표현하면 식 (47)과 같다.

$$[c_i] = [T] \cdot [d_i] \quad (47)$$

식 (47)에서 역변환행렬 T는 연산영역에서 함수영역으로 변환하며 식 (48)과 같다.

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \quad (48)$$

(그림 1)의 $GF(3)$의 가산게이트와 (그림 2)의 $GF(3)$상의 승산게이트를 사용하여 3치 Reed-Muller 전개식의 변환행렬 M과 역변환행렬 T에 의한 식 (41)과 식 (46)을 실현하는 기본 셀을 구성하면 (그림 4)와 같다. (그림 4) (a)는 3치 Reed-Muller 전개식의 계수 c_i를 이용하여 변환계수 d_i를 구하는 변환행렬 M을 실현하는 기본 셀이고, (그림 4) (b)는 역변환행렬 T를 실현하는 기본 셀이다.



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

3.3.2 3치 2변수 Reed-Muller 전개식

3치 2변수 Reed-Muller 전개식은 식 (38)에 의해서 다음과 같이 나타낼 수 있다.

$$F(x_2, x_1) = c_0 \oplus c_1 \cdot x_1 \oplus c_2 \cdot x_1^2 \oplus c_3 \cdot x_2 \oplus c_4 \cdot x_2 x_1$$
$$\oplus c_5 \cdot x_2 x_1^2 \oplus c_6 \cdot x_2^2 \oplus c_7 \cdot x_2^2 x_1 \oplus c_8 \cdot x_2^2 x_1^2 \quad (49)$$

여기서 $c_i, x_1^i, x_2^i \in GF(3)$이다.

식 (49)에서 3치 2변수 Reed-Muller 전개식의 변환행렬 M_i는 식 (43)의 M을 Kronecker 곱하여 식 (50)과 같이 구할 수 있다.

$$\bigotimes_{i=0}^{2-1} M_i = \begin{bmatrix} M^{[i]} & 0 & 0 \\ M^{[i]} & M^{[i]} & M^{[i]} \\ M^{[i]} & 2 \cdot M^{[i]} & M^{[i]} \end{bmatrix} \quad (50)$$

또한 역변환행렬 T_i는 식 (48)의 T를 Kronecker 곱하여 구하며 다음과 같다.

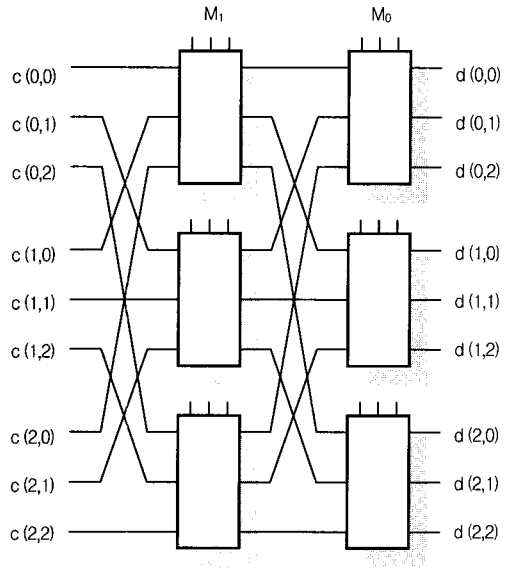
$$\bigotimes_{i=0}^{2-1} T_i = \begin{bmatrix} T^{[i]} & 0 & 0 \\ 0 & 2 \cdot T^{[i]} & T^{[i]} \\ 2 \cdot T^{[i]} & 2 \cdot T^{[i]} & 2 \cdot T^{[i]} \end{bmatrix} \quad (51)$$

식 (50)에서 2변수인 경우이므로 $M = M_1 \otimes M_0$이고, 식 (51)은 $T = T_1 \otimes T_0$이다. 식 (50)과 식 (51)을 앞 절에서 논한 Per

fect Shuffle 기법과 Kronecker 곱에 의한 식 (36)을 이용하여 변환행렬 M을 연산하면 식 (52)와 같다.

$$\begin{aligned} M &= \bigotimes_{i=2-1}^{0} M_i = M_1 \otimes M_0 \\ &= \prod_{i=2-1}^{0} [(I_{3^{[2-1-i]}} \otimes S_{3^{[i]},3}) \cdot (I_{3^{[2-1]}} \otimes M_i) \cdot \\ &\quad (I_{3^{[2-1-i]}} \otimes S_{3,3^{[i]}})] \\ &= S_{3,3} \cdot (I_3 \otimes M_1) \cdot S_{3,3} \cdot (I_3 \otimes M_0) \end{aligned} \quad (52)$$

식 (52)에 의하여 실현한 회로가 (그림 5)와 같다. (그림 5)의 각 기본 셀의 내부회로는 $GF(3)$상의 가산게이트와 승산게이트에 의해 실현된 (그림 3) (a)의 기본 셀의 회로와 같다.

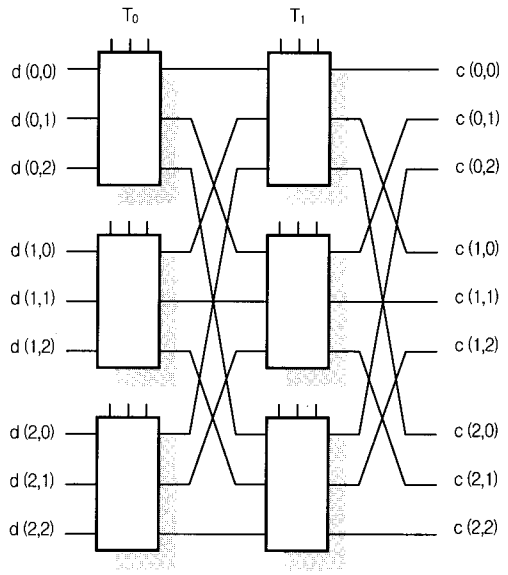


(그림 5) 3치 2변수 RM 전개식의 변환회로 실현

유사한 방법으로 Perfect Shuffle 기법과 Kronecker 곱에 의한 식 (36)을 이용하여 역변환행렬 T를 연산하면 다음과 같다.

$$\begin{aligned} T &= \bigotimes_{i=2-1}^{0} T_i = T_1 \otimes T_0 \\ &= \prod_{i=2-1}^{0} [(I_{3^{[2-1-i]}} \otimes S_{3^{[i]},3}) \cdot (I_{3^{[2-1]}} \otimes T_i) \\ &\quad \cdot (I_{3^{[2-1-i]}} \otimes S_{3,3^{[i]}})] \\ &= S_{3,3} \cdot (I_3 \otimes T_1) \cdot S_{3,3} \cdot (I_3 \otimes T_0) \end{aligned} \quad (53)$$

식 (53)에 의하여 실현한 회로가 (그림 6)과 같다. (그림 6)의 각 기본 셀의 내부회로는 $GF(3)$상의 가산게이트와 승산게이트에 의해 실현된 (그림 3) (a)의 기본 셀의 회로와 같다.



(그림 6) 2변수 3치 RM 전개식의 역변환회로의 실현

〈표 1〉 다치 Reed-Muller 전개식의 비교표

구 분	직접계산	Yang[11]	Rahardja[15]	Stankovic[13]		본 논문	
				GF	RMF	변환행렬	역변환행렬
승산게이트	$p^m \cdot p^m$	$2m \cdot p^{m-1}$	$\frac{2}{9}(9^m - p^m)$	$m \cdot p^m$	$2m \cdot p^{m-1}$	$m \cdot p^{m-1}$	$2m \cdot p^{m-1}$
가산게이트	$p^m(p^m-1)$	$3m \cdot p^{m-1}$	$\frac{3}{9}(9^m - p^m)$	$3m \cdot p^{m-1}$	$3m \cdot p^{m-1}$	$3m \cdot p^{m-1}$	$3m \cdot p^{m-1}$
레지스터	$p^m(p^m+2)$	$3m$	-	-	-	-	-

〈표 2〉 $p=3$이고 $m=2$인 경우의 다치 Reed-Muller 전개식의 비교표

구 분	직접계산	Yang[11]	Rahardja[15]	Stankovic[13]		본 논문	
				GF	RMF	변환행렬	역변환행렬
승산게이트	81	12	16	18	12	6	12
가산게이트	72	18	24	18	18	18	18
레지스터	99	9	-	-	-	-	-

4. 비교 및 검토

이 장에서는 제시한 다치 신호처리회로를 타 연구의 회로와 비교하였으며, 비교표가 <표 1>과 같으며, <표 2>는 $p=3$이고 $m=2$인 경우 가산게이트와 승산게이트를 비교한 표이다. 다치 Reed-Muller 전개식의 신호처리회로의 비교표인 <표 2>에서와 같이 제시한 다치 Reed-Muller 전개식의 신호처리회로는 Yang[11]의 고속 알고리듬에 의한 행렬변환 방법보다 변환행렬의 경우 승산게이트의 수가 2배로 줄어들며 가산게이트의 수는 동일하다. 역변환행렬의 경우 가산게이트와 승산게이트의 수가 동일하다. 이 결과는 가산게이트는 타 연구와 동일한 결과를 보이며, 승산게이트는 약 2배로 감소한다.

그러므로 다치 신호처리의 연산에서는 본 논문이 소자수면에서 다소 우수하며, 정보량의 처리면에서도 우수하다. 제시한 다치 신호처리 회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬 동작의 특징을 가진다.

5. 결 론

본 논문에서는 Perfect Shuffle 기법과 Kronecker 곱에 의한 다치 신호처리회로의 입력과 출력의 상호연결 방법에 대하여 논하였고, 다치 신호처리회로의 입력과 출력의 상호연결 방법을 이용하여 $GF(p^m)$상의 다치 Reed-Muller 전개식의 신호처리회로와 설계방법을 제시하였다. 제시된 다치 신호처리회로의 입력과 출력 상호연결 방법은 행렬변환이 효과적이고 모듈구조를 갖는다.

유한체 $GF(3^2)$상의 다치 Reed-Muller 전개식의 변환행렬과 역변환행렬의 회로설계에서 변환행렬의 회로는 Stankovic 등이 제시한 Reed-Muller 전개식의 변환 알고리듬과 연산게이트 수가 동일하며, Yang과 Rahardja 등 이 제시한 알고리듬보다 약간 우수하다.

본 논문에서 제시한 다치 신호처리회로는 회선경로 선택의 규칙성, 간단성, 배열의 모듈성과 병렬 동작의 특징을 가지므로 VLSI/ULSI 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] B. Benjauthrit and I. S. Reed, "Galois Switching Functions and Their Application," *IEEE Trans. Comput.*, Vol.C-25, No.1, pp.78-86, Jan., 1976.

[3] J. T. Butler and A. S. Wojcik, "Guest Editors' Comments," *IEEE Trans. Comput.*, Vol.C-30, No.9, pp.617-618, Sept., 1981.

[4] H. T. Kung, "Why Systolic Architectures?," *IEEE Computer*, Vol.15, pp.37-46, Jan., 1982.

[5] H. M. Shao, T. K. Truoung, L. J. Deutsch, J. H. Yaeh and I. S. Reed, "A VLSI Design of a Pipelining Reed-Solomon Decoder," *IEEE Trans. Comput.*, Vol.C-34, No.5, pp.393-403, May, 1985.

[6] H. Y. Seong and H. S. Kim, "A Construction of Cellular Array Multiplier over $GF(2^m)$," *KITE*, Vol.26, No.4, pp.81-87, April, 1989.

[7] I. S. Hsu, T. K. Truong, L. T. Deutsch and I. S. Reed, "A Comparison of VLSI Architecture of Finite Field Multiplier Using Dual, Normal, or Standard Bases," *IEEE Trans. Comput.*, Vol.C-37, No.6, pp.735-739, June, 1988.

[8] B. B. Zhou, "A New Bit-Serial Systolic Multiplier over $GF(2^m)$," *IEEE Trans. Comput.*, Vol.C-37, No.6, pp.749-751, June, 1988.

[9] J. M. Pollard, "The Fast Fourier Transform in a Finite Field," *Math. Comput.*, Vol.25, No.114, pp.365-374, April, 1971.

[10] Y. Wang and X. Zhu, "A Fast Algorithm for the Fourier Transform over Finite Field and Its VLSI Implementation," *IEEE J. Select.* Area Commu., Vol.6, No.3, pp.573-577, April, 1988.

[11] 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.

[12] 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.

[13] R. S. Stankovic and C. Moraga, "Reed-Muller-Fourier Versus Galois Field Representations of Four-Valued Logic Functions," *Proc. of 28th International Symposium on Multiple-Valued Logic*, Fukuoka, Japan, pp.186-191, May, 1998.

[14] R. S. Stankovic and J. Astola, "Bit-Level and Word-Level Polynomial Expressions for Functions in Fibonacci Interconnection Topologies," *Proc. of 31st International Symposium on Multiple-Valued Logic*, Warsaw, Poland, pp.305-310, May, 2001.

[15] 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.

[16] B. Harking and C. Moraga, "Efficient Derivation of Reed-Muller Expansions in Multiple-Valued Logic Systems," *Proc. of 22nd International Symposium on Multiple-Valued Logic*, pp.436-441, May, 1992.

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

[18] T. Y. Feng, "A Survey of Interconnection Networks," *IEEE Computer*, Vol.14, No.10, pp.12-27, Dec., 1981.

[19] C. L. Wu and T. U. Feng, "On a Class of Multistage Interconnection Networks," *IEEE Trans. Comput.*, Vol.C-29, No.8, pp.694-702, Aug., 1980.

성 현 경

e-mail : hkseong@mail.sangji.ac.kr

1982년 인하대학교 전자공학과 졸업(공학사)

1984년 인하대학교 대학원 전자공학과 졸업 (공학석사)

1991년 인하대학교 대학원 전자공학과 졸업 (공학박사)

1989년~1991년 부천전문대학 전자계산과 조교수

1991년~현재 상지대학교 컴퓨터·정보공학부 부교수

관심분야 : Multiple-Valued Logic Design, Computer Architecture & VLSI 설계, Information & Cryptography Theory, Digital Signal Processing 등