Reed-Muller 전개식에 의한 3치 논리회로의 설계

성 현 경*

Design of Ternary Logic Circuits Based on Reed-Muller Expansions

Hyeon-Kyeong Seong*

요 약

본 논문에서는 Reed-Muller 전개식에 의한 3치 논리 회로를 설계하는 한 가지 방법을 제시하였다. 제시된 3치 논리 회로의 설계 방법은 Reed-Muller 전개식의 계수에 대하여 모든 변수의 차수를 검사하여 RME 모듈(Reed-Muller Expansions module)의 수를 최소화하는 최적의 제어 입력 변수의 순서를 결정한다. 최적의 제어 입력 변수의 순서는 회로 비용 행렬의 계산에 사용되며, 이 회로 비용 행렬의 계산 결과를 이용하여 Reed-Muller 전개식에 의한 RME 모듈의 나무 구조의 3치 논리 회로를 설계한다. 제시된 방법은 최적 제어 입력 변수를 찾는데 유일하게 단위시간 내에 수행되며, 컴퓨터 프로그램이 가능하고, 프로그래밍 수행 시간이 3^n이다.

ABSTRACT

In this paper, we present a design method of the ternary logic circuits based on Reed-Muller expansions. The design method of the presented ternary logic circuits checks the degree of each variable for the coefficients of Reed-Muller Expansions(RME) and determines the order of optimal control input variables that minimize the number of Reed-Muller Expansions modules. The order of optimal control input variables is utilized the computation of circuit cost matrix. The ternary logic circuits of the minimized tree structures to be constructed by RME modules based on Reed-Muller Expansions are realized using the computation results of its circuit cost matrix. This method is only performed under unit time in order to search for the optimal control input variables. Also, this method is able to be programmed by computer and the run time on programming is 3^n.

키워드

Ternary logic circuit, Reed-Muller Expansions, RME module, circuit cost matrix

I. 서 론

오늘날 사용되고 있는 디지털 시스템은 대부분 2진 논리 이론을 기초로 하고 있으며, 반도체 집적기술의 발달로 인하여 칩의 집적도가 비약적으로 증가하고 회로의 복잡도가 날로 높아가고 있다. 이러한 VLSI 시스템에 심각하게 대두되고 있는 단자수의 제한 문제, 단자 간의 상호 연결 문제, 보다 많은 정보량의 처리 문제와 연산 속도의 제한성이라는 근본적인 문제에 직면하게 되었으며, 이러한 문제점을 해결하는 가장 효과적인 방법 중의 한 가지 방법이 다치 논리 회로가 될 것으로 예상된다 [1,2].

다치 논리 회로는 함수적 완전성, 유연성, 회로의 실현을 만족하여야 한다. Reed-Muller 전개식(Reed-Muller

* 상지대학교 컴퓨터정보공학부

접수일자 : 2006. 10. 23

Expansion; RME)은 이러한 성질을 만족하므로, Reed-Muller 전개식을 이용한 다치 논리 회로의 설계에 관한 연구가 활발히 진행되고 있다. 대부분의 많은 다치 논리 회로 합성에 관한 연구는 규칙적인 셀 배열과 간단하고 규칙적인 상호 연결 형식을 갖고 논리 회로를 합성할 수 있는 동일한 기본 빌딩 블록을 사용하여 논리 함수를 실현하고 있으며, 대표적으로 Reed-Muller 전개식 모듈(Reed- Muller Expansions Module; RMEM)를 사용한다[3-6].

기본 모듈인 RME 모듈을 이용하는 다치 논리 회로의 합성은 나무 구조를 가지며, 이러한 다치 논리 회로 합성은 다치 논리 함수의 스위칭 함수에 효과적으로 적용할 수 있다. 나무 구조 형식에서 요구되는 최대 RME 모듈의 수가 n변수 3치 논리회로인 경우 $(3^n-1)/(3-1)$이며, 제어 입력 변수의 순서를 어떻게 선택하는가에 따라서 RME 모듈의 수를 상당히 줄일 수 있으므로 최적의 제어 입력 변수의 순서를 선택하는 것이 매우 중요하다. 그러므로 선택된 제어 입력 변수의 순서에 따라서 다치 논리 회로를 합성하면 Reed-Muller 전개식에 의한 RME 모듈의 간단한 나무 구조를 갖는 다치 논리 회로를 실현할 수 있다[7,8].

여러 가지 다치 논리 회로의 구현 방법이 제안되었으나 이러한 방법들은 다치 논리 회로의 구현 방법에서 약간의 단점들을 가지고 있다. Hong과 Fei[9]는 3치 Reed-Muller 전개식에서 변수의 수가 많을 때 그레이 코드 순서에서 단계적 흐름 그래프의 계산보다 더 적은 계산 비용을 가지나 병렬 계산에서 순환 알고리듬을 사용하므로 변환 행렬이 복잡하고, 기수가 높은 Reed-Muller 전개식에서는 다치 논리 회로 합성 방법이 어렵게 되고 회로가 복잡해진다. 또한 Fei와 Hong[10]는 Reed-Muller 전개식에 대하여 Kronecker 적을 사용하여 계수를 찾는 방법을 제시하였다. 그러나 Kronecker 적을 사용하므로 계산 비용이 높고 다변수일 경우에는 회로가 매우 복잡해지고 회로 비용이 높은 단점이 있다.

Jankovic 등[11]은 Reed-Muller 전개식에 대한 고정극성 다항식의 표현의 계산에 대한 표사용 기법을 제안하였다. 이 기법은 Reed-Muller 전개식에 대응하는 방법의 일반성을 유도하였고, 관련된 연산의 간단성, 계산절차의 병렬화에 대한 가능성과 Kronecker 적으로 계수의 계산을 확장할 수 있는 방법을 제시하였다. Falkowski 등[12]은 Reed-Muller 전개식에서 고속 선형 독립 변환 방법을 소개하였다. 선형 독립 변환은 3치 Reed-Muller 전개식에서 최소 계산 복잡성을 감소시키며, 양면 논리 게이트에서 쉽게 구현할 수 있다. Al-Rabadi[13]는 순열기반 고속변환을 사용하는 3치 양자회로의 합성방법을 제시하였다. 제시된 방법은 양방향 순열 버터플라이 회로를 사용하며, 유한체상에서 가산과 승산을 수행하는 양자게이트의 양방향성을 사용하는 장점이 있다.

실제로 나무 구조를 갖는 RME 모듈의 3치 논리 회로의 설계 방법은 제어 입력 변수의 최적 할당을 선택하는 것이다. 일반적으로 n변수 함수는 $n!$종류의 할당 방법이 있다. 그러므로 n의 값이 크면 제어 입력 변수의 할당 방법이 급격히 증가한다.

본 연구는 제어 입력 변수의 최적 순서를 할당하는 한 가지 알고리듬을 제시한다. 이 알고리듬은 Reed-Muller 전개식의 계수를 갖는 모든 변수의 차수를 변수 차수표를 사용하여 각 변수의 차수를 조사함으로서 최적의 제어 입력 변수의 순서를 선택한다. 이러한 알고리듬에 의해 선택된 제어 입력 변수의 최적 순서를 Reed-Muller 전개식에 의한 회로 비용 행렬을 이용하여 최적의 나무 구조를 갖는 RME 모듈의 3치 논리 회로를 설계한다.

Ⅱ. 이론적 배경

2.1 기본 RME 모듈

본 절에서는 3치 논리 회로의 구현에 필요한 수학적 배경을 설명한다. 3치 논리 회로에서 논리값이 집합 $R=\{0,1,2\}$이며, 이를 이용한 Reed-Muller 전개식에 의한 RME 모듈 M_f에 대하여 설명한다.

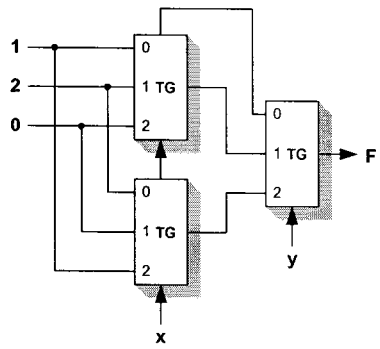
2.1.1 3치 가산회로

3치 2변수 x와 y의 가산은 두 원소를 가산한 후 $mod(3)$을 수행하며, 다음과 같은 함수식을 갖는다. 표 1은 식 (1)을 수행하는 가산표이며, 그림 1(a)는 T-게이트를 이용하여 표 1의 가산표를 실현하는 가산회로이며 그림 1(b)는 가산회로의 기호이다.

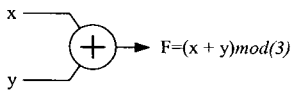
$$F = x \oplus y = (x + y) mod(3) \tag{1}$$

표 1. 2변수 3치 가산표
Table 1. 2 variable ternary addition table.

+	x=0	x=1	x=2
y=0	0	1	2
y=1	1	2	0
y=2	2	0	1



(a)



(b)

그림 1. 2변수 3치 가산회로; (a) 가산회로 (b) 기호
Fig. 1. 2 variable ternary addition circuit; (a) addition circuit (b) symbol

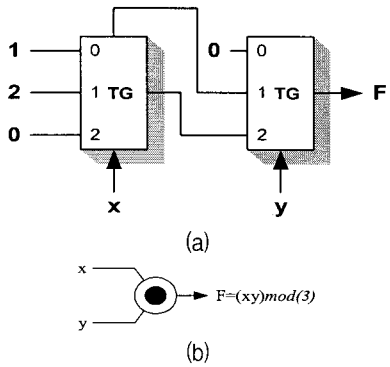
2.1.2 3치 승산회로

3치 2변수 x와 y의 승산은 두 원소를 승산한 후 $mod(3)$을 수행하며, 다음과 같은 함수식을 갖는다. 표 2는 식 (2)을 수행하는 승산표이며, 그림 2(a)는 T-게이트에 의한 표 2의 승산표를 실현하는 승산회로이며 그림 2(b)는 승산회로의 기호이다.

$$F = x \cdot y = (x \cdot y) mod(3) \tag{2}$$

표 2. 2변수 3치 승산표
Table 2. 2 variable ternary multiplication table.

•	x=0	x=1	x=2
y=0	0	0	0
y=1	0	1	2
y=2	0	2	1



(a)

(b)

그림 2. 2변수 3치 승산회로; (a) 승산회로 (b) 기호
Fig. 2. 2 variable ternary multiplication circuit; (a) multiplication circuit (b) symbol

2.1.3 단일 변수 3치 Reed-Muller 전개식의 설계

3치 단일 변수 x에 대하여 식 (1)과 식 (2)에 의한 Reed-Muller 전개식을 구현하는 RME 모듈 M_f의 함수는 다음과 같으며, 그림 3(a)는 3치 가산회로와 3치 승산회로를 이용하여 구성한 RME 모듈 M_f의 회로이며, 그림 3(b)는 RME 모듈 M_f의 기호이다. $c_i = \{0,1,2\}$이다.

$$M_f(c_0, c_1, c_2, x) = c_0 \oplus c_1 \cdot x \oplus c_2 \cdot x^2 \tag{3}$$

2.2. N변수 3치 Reed-Muller 전개식의 설계

3치 논리 시스템에서 n변수 3치 함수는 다음과 같이 일반적인 Reed-Muller 전개식으로 표현할 수 있다.

$$F(x_1, x_2, \cdots, x_n) = \sum_{i=0}^{3^n-1} c_i \left[\prod_{j=1}^{n} x_j^{i \cdot j} \right] \tag{4}$$

여기서 $c_i = \{0,1,2\}$이고 $x_j^{i \cdot j}$는 변수 x_j의 $i \cdot j$의 승수(power)이다.

식 (4)를 n변수 3치 Reed-Muller 전개식으로 나타내면 다음과 같다.

$$F(x_1,x_2,\cdots,x_n)=c_0\cdot x_1^0x_2^0\cdots x_n^0\oplus c_1\cdot x_1^0x_2^0\cdots x_n^1\oplus\cdots \oplus c_j\cdot x_1^{e_1}x_2^{e_2}\cdots x_n^{e_n}\oplus c_{3^n-1}\cdot x_1^2x_2^2\cdots x_n^2 \quad (5)$$

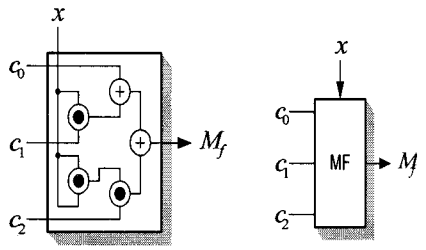


그림 3. Ree-Muller 전개식의 모듈 M_f;
(a) RME M_f의 회로 (b) RME M_f의 기호
Fig. 3. Module M_f of Reed-Muller expansions;
(a) circuit of RME M_f (b) symbol of RME M_f

여기서, $x_i^0=1$, $x_i^1=x_i$, $x_i^2=x_i\cdot x_i$이며, $i=\{0,1,\cdots,n\}$이다. 또한 $c_j\in\{0,1,2\}$이며, $j=\{0.1.2,\cdots,3^n-1\}$이고, $e_i\in\{0,1,2\}$이고, $(j)_{10}=(e_1e_2\cdots e_n)_3$이다.

식 (5)의 n변수 3치 Reed-Muller 전개식은 그림 4에서 보인 것처럼 RME 모듈 M_f를 사용하여 나무 구조 형식으로 실현할 수 있으며, 그림 4에서 요구되는 최대 RME 모듈 M_f의 수가 $(3^n-1)/(3-1)$이다. 그림 4에서 보인 것처럼 n번째 단을 나무의 뿌리로서 정의한다.

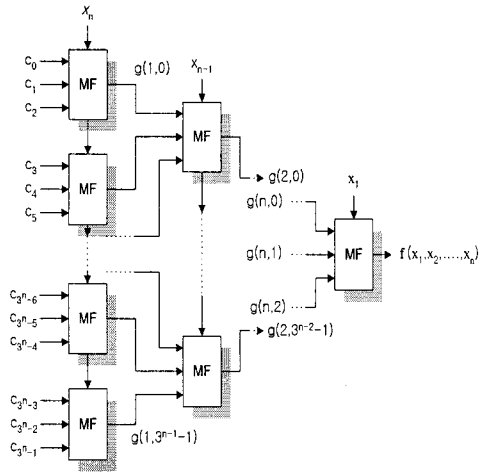


그림 4. RME 모듈 M_f를 사용한 n변수 3치 함수의 실현
Fig. 4. Realization of n variable ternary function using RME module M_f.

Ⅲ. 3치 Reed-Muller 전개식의 회로 비용 행렬

[정의 1] n단으로 구성된 나무 구조의 3치 논리 회로 구현에서 임의의 변수 x_k가 i-번째 중간 함수의 RME 모듈 M_f의 출력 $g_k(i,j)$를 정의하며, $g_k(0,j)=c_j$이다[9]. 여기서 $i=0,1,2,\cdots,n$이고 $j=0,1,2,\cdots,3^n-1$이다.

x_n이 첫 번째 제어 입력 변수라고 가정하자. 그러면 식 (5)는 다음과 같이 표현할 수 있다.

$$F(x_1,x_2,\cdots,x_n)=(c_0\oplus c_1\cdot x_n\oplus c_2\cdot x_n^2)\oplus(c_3\oplus c_4\cdot x_n\oplus c_5\cdot x_n^2)x_{n-1} \oplus\cdots\oplus(c_{3^n-3}\oplus c_{3^n-2}\cdot x_n\oplus c_{3^n-1}\cdot x_n^2)x_1^2x_2^2\cdots x_{n-1}^2 \quad (6)$$

[정의 1]에 따라서 식 (6)은 다음과 같이 표현할 수 있다.

$$F(x_1,x_2,\cdots,x_n)=g_n(1,0)\oplus g_n(1,1)x_{n-1}\oplus g_n(1,2)x_{n-1}^2 \oplus\cdots\oplus g_n(1,3^{n-1}-1)x_1^2x_2^2\cdots x_{n-1}^2 \quad (7)$$

식 (6)과 식 (7)을 비교하여 다음과 같이 식 (8)을 얻을 수 있다.

$$\begin{aligned} g_n(1,0)&=c_0\oplus c_1\cdot x_n\oplus c_2\cdot x_n^2\\ g_n(1,1)&=c_3\oplus c_4\cdot x_n\oplus c_5\cdot x_n^2\\ &\vdots\qquad\qquad\vdots\\ g_n(1,3^{n-1}-1)&=c_{3^n-3}\oplus c_{3^n-2}\cdot x_n\oplus c_{3^n-1}\cdot x_n^2 \end{aligned} \quad (8)$$

식 (8)에 의해 Reed-Muller 전개식의 계수들에 대한 회로 비용 행렬을 다음과 같이 표현할 수 있다.

$$\left[g_n(1,0)\ g_n(1,1)\ \cdots\ g_n1,3^{n-1}-1)\right] =\left[1\ x_n\ x_n^2\right]\begin{bmatrix} c_0 & c_3 & \cdots & c_{3^n-3}\\ c_1 & c_4 & \cdots & c_{3^n-2}\\ c_2 & c_5 & \cdots & c_{3^n-1}\end{bmatrix} \quad (9)$$

[정의 2] 임의의 변수 x_m이 q-번째 제어 입력 변수이면, $\left[g_m(q,0)\ g_m(q,1)\ g_m(q,3^{n-1}-1)\right]$를 갖는 중간 함수 $\left[G_m(q-1)\right]$를 정의하며, 이것은 회로 비용 행렬 $\left[G_m(q-1)\right]$로부터 구하여진다.

[정의 2]에 의해 만약 임의의 변수 x_i가 첫 번째 제어 입력 변수이면, Reed-Muller 전개식의 계수의 중간 함수를 표현할 수 있고, 식 (10)과 같은 회로 비용 행렬을 얻을 수 있다.

$$[G_i(1)]=[g_i(1,0)\ g_i(1,1)\ \cdots\ g_i(1,3^{n-1}-1)]$$
$$=[1\ x_i\ x_i^2]\begin{bmatrix} g(0,L^{(0)}) & g(0,L^{(1)}) & \cdots & g(0,L^{(k)}) & \cdots & g(0,L^{(3^{n-1}-1)}) \\ g(0,M^{(0)}) & g(0,M^{(1)}) & \cdots & g(0,M^{(k)}) & \cdots & g(0,M^{(3^{n-1}-1)}) \\ g(0,N^{(0)}) & g(0,N^{(1)}) & \cdots & g(0,N^{(k)}) & \cdots & g(0,N^{(3^{n-1}-1)}) \end{bmatrix}$$
$$=[1\ x_i\ x_i^2][G_i(0)] \qquad (10)$$

여기서,

$$(k)_{10}=(e_1\ e_2\ e_3\ \cdots\ e_{i-1}\ e_{i+1}\ \cdots\ e_n)_3$$
$$(L^{(k)})_{10}=(e_1\ e_2\ e_3\ \cdots\ e_{i-1}\ 0\ e_{i+1}\ \cdots\ e_n)_3$$
$$(M^{(k)})_{10}=(e_1\ e_2\ e_3\ \cdots\ e_{i-1}\ 1\ e_{i+1}\ \cdots\ e_n)_3$$
$$(N^{(k)})_{10}=(e_1\ e_2\ e_3\ \cdots\ e_{i-1}\ 2\ e_{i+1}\ \cdots\ e_n)_3$$

만약 임의의 변수 x_j가 두 번째 제어 입력 변수이면, 식 (11)과 같이 회로 비용 행렬을 얻을 수 있다.

$$[G_j(2)]=[g_j(2,0)\ g_j(2,1)\ \cdots\ g_j(2,3^{n-2}-1)]$$
$$=[1\ x_j\ x_j^2]\begin{bmatrix} g(1,L^{(0)}) & g(1,L^{(1)}) & \cdots & g(1,L^{(k)}) & \cdots & g(1,L^{(3^{n-1}-1)}) \\ g(1,M^{(0)}) & g(1,M^{(1)}) & \cdots & g(1,M^{(k)}) & \cdots & g(1,M^{(3^{n-1}-1)}) \\ g(1,N^{(0)}) & g(1,N^{(1)}) & \cdots & g(1,N^{(k)}) & \cdots & g(1,N^{(3^{n-1}-1)}) \end{bmatrix}$$
$$=[1\ x_j\ x_j^2][G_j(1)] \qquad (11)$$

여기서,

$$(k)_{10}=(e_1\ e_2\ \cdots\ e_{j-1}\ e_{j+1}\ \cdots\ e_{i-1}\ e_{i+1}\ \cdots\ e_n)_3$$
$$(L^{(k)})_{10}=(e_1\ \cdots\ e_{j-1}\ 0\ e_{j+1}\ \cdots\ e_{i-1}\ e_{i+1}\ \cdots\ e_n)_3$$
$$(M^{(k)})_{10}=(e_1\ \cdots\ e_{j-1}\ 1\ e_{j+1}\ \cdots\ e_{i-1}\ e_{i+1}\ \cdots\ e_n)_3$$
$$(N^{(k)})_{10}=(e_1\ \cdots\ e_{j-1}\ 2\ e_{j+1}\ \cdots\ e_{i-1}\ e_{i+1}\ \cdots\ e_n)_3$$

유사하게 만약 임의의 변수 x_w가 i-번째 제어 입력 변수이면 식 (12)와 같이 일반적인 회로 비용 행렬을 얻을 수 있다.

$$[G_w(i)]=[g_w(i,0)\ g_w(i,1)\ \cdots\ g_w(i,3^{n-i}-1)]$$
$$=[1\ x_w\ x_w^2]\begin{bmatrix} g(i-1,L^{(0)}) & g(i-1,L^{(1)}) & \cdots & g(i-1,L^{(k)}) & \cdots & g(i-1,L^{(3^{n-i}-1)}) \\ g(i-1,M^{(0)}) & g(i-1,M^{(1)}) & \cdots & g(i-1,M^{(k)}) & \cdots & g(i-1,M^{(3^{n-i}-1)}) \\ g(i-1,N^{(0)}) & g(i-1,N^{(1)}) & \cdots & g(i-1,N^{(k)}) & \cdots & g(i-1,N^{(3^{n-i}-1)}) \end{bmatrix}$$
$$=[1\ x_w\ x_w^2][G_w(i-1)] \qquad (12)$$

Ⅳ. Reed-Muller 전개식에 의한 3치 논리회로의 설계

4.1 제어 입력변수 선택 알고리즘

RME 모듈 M_f를 사용하는 n변수 3치 논리 회로의 구현은 나무 구조를 가지며, 이러한 3치 논리 회로의 구현은 효과적으로 스위칭 함수에 적용할 수 있다. 이러한 나무 구조의 최적 3치 논리 회로의 구현에 대하여 제어 입력 변수의 순서를 선택하는 것이 매우 중요하며, 선택된 최적 제어 입력 변수의 순서는 최적의 3치 논리 회로의 구현에 대하여 간단한 나무 구조를 가진다. 그러므로 n변수 3치 Reed-Muller 전개식에 대하여 선택된 최적 제어 입력 변수의 순서가 나무 구조를 갖는 3치 논리 회로의 구현에 할당되기 때문에 RME 모듈 M_f를 사용하는 3치 논리 회로 구현에서 회로 비용이 감소될 수 있다. 회로 비용을 감소하기 위한 최적 제어 입력 변수의 순서를 선택하는 알고리듬이 다음과 같다.

[단계 1] Reed-Muller 전개식의 계수들을 갖는 모든 변수의 차수에 관한 변수 차수표(variable degree table; VDT)를 만든다. 예를 들면, 만약 2변수 3치 논리 함수이면 이 함수의 최대 계수의 수가 9이고, 만약 3변수 3치 논리 함수이면, 이 함수의 최대 계수의 수가 27이다.

[단계 2] Reed-Muller 전개식의 모든 계수에 대하여 각 변수의 차수를 검사하여 변수 차수표에 해당되는 차수의 열을 "√" 기호로 표시한다. 만약 2변수이면 각 계수 열에 2개의 "√" 기호를 가지며, 3변수이면 각 계수 열에 3개의 "√" 기호를 갖는다.

[단계 3] 변수 차수표에 표시된 "√" 기호의 수를 각 변수의 차수에 대하여 계산하고, 변수 차수표의 아래에 계산된 수치를 나타낸다. 각 변수의 차수에 대하여 계산된 수의 합이 동일하다.

[단계 4] 변수 차수표에서 차수가 0인 변수(x_i^0)들에서 "√" 기호가 가장 많은 열의 변수를 첫 번째 제어 입력 변수로 선택할 수 있으며, 표시된 수에 따라서 제어 입력 변수가 결정될 수 있다.

[단계 5] 표시된 기호의 수가 모든 변수에 대하여 동일하다면, 차수 1인 변수(x_i^1)에서 가장 많이 표시된 열의 변수를 첫 번째 제어 입력 변수로 할당할 수 있다.

[단계 6] 다음의 제어 입력 변수는 선택된 변수를 제외한 모든 변수에 대하여 [단계 4]와 [단계 5]를 반복하

여 제어 입력 변수의 순서를 결정한다.

[단계 7] 선택된 제어 입력 변수의 순서에 따라서 식 (12)의 회로 비용 행렬을 이용하여 RME 모듈 M_f에 의해 회로를 구현한다.

4.2 Reed-Muller 전개식의 3치 논리회로 설계

위에서 제시한 Reed-Muller 전개식에 대하여 최적 제어 입력 변수의 순서를 선택하는 알고리듬을 사용하여 RME 모듈 M_f에 의한 3치 논리 회로의 설계방법을 예를 들면 다음과 같다.

3변수 3치 Reed-Muller 전개식인 다음과 같은 3치 논리 함수 $F(x_1,x_2,x_3)$를 RME 모듈 M_f을 사용하여 3치 논리 회로를 설계한다.

$$F(x_1,x_2,x_3) = 2\cdot x_3 \oplus x_3^2 \oplus x_1 \oplus x_1x_3 \oplus x_1x_3^2 \oplus 2\cdot x_1x_2x_3 \oplus 2\cdot x_1x_2x_3^2 \oplus x_1x_2^2x_3 \oplus x_1x_2^2x_3^2 \oplus x_1^2x_3 \oplus x_1^2x_2x_3 \oplus 2\cdot x_1^2x_2^2x_3 \quad (13)$$

식 (13)의 3변수 3치 Reed-Muller 전개식의 계수 c_j의 벡터열을 식 (14)와 같이 나타낼 수 있다.

$$[C] = [021\ 000\ 000\ 111\ 022\ 011\ 010\ 010\ 020]^T \quad (14)$$

위에서 논한 최적 제어 입력 변수의 순서를 선택하는 알고리듬을 사용하여 RME 모듈 M_f에 의한 3치 논리 회로를 설계하는 절차가 다음과 같다.

[단계 1] 식 (13)으로 부터 3변수 3치 Reed- Muller 전개식의 첫 번째 항 $2\cdot x_3$에서 변수 x_1과 x_2의 차수가 각각 0이고, 변수 x_3의 차수가 1이므로 변수 차수표에서 변수 x_1의 0차수 항, 변수 x_2의 0차수 항, 변수 x_3의 1차수 항에 기호 "√"를 표시한다.

[단계 2] 식 (13)의 두 번째 항인 x_3^2에서 변수 x_1과 x_2의 차수가 역시 각각 0이고, 변수 x_3^2의 차수가 2이다. 그러므로 변수 차수표에서 변수 x_1의 0차수 항, x_2의 0차수 항, 변수 x_1의 2차수 항에 기호 "√"를 표시한다.

[단계 3] Reed-Muller 전개식의 모든 계수들에 대하여 반복적으로 이러한 절차를 수행한다. 이러한 과정의 결과가 표 3에서 보인다.

표 3에서 변수 x_1의 0차수 항에서 표시된 "√" 기호의 수가 2이고 1차수 항에서 표시된 "√" 기호의 수가 7이고, 2차수 항에서 표시된 "√" 기호의 수가 3이다. 그리고 변수 x_2에 대하여 0차수 항에서 표시된 "√" 기호의 수가 6이고, 1차수 항에서 표시된 "√" 기호의 수가 3이고, 2차수 항에서 표시된 "√" 기호의 수가 3이다. 변수 x_3에 대하여 0차수, 1차수, 2차수 항에 표시된 "√" 기호의 수가 각각 1, 7, 4이다. 그러므로 첫 번째 제어 입력 변수가 x_2이고, 두 번째 제어 입력 변수가 x_1이고, 세 번째 제어 입력 변수가 x_3이다.

선택된 제어 입력 변수의 순서 (x_2,x_1,x_3)에 의해서 회로 비용 행렬을 사용하여 RME 모듈 M_f의 입력 계수를 결정할 수 있다. 식 (14)의 계수 벡터열에서 회로 비용 행렬 $[G_1(0)]$, $[G_2(0)]$, $[G_3(0)]$을 다음과 같이 구할 수 있다.

$$[G_1(0)] = \begin{bmatrix} 0&2&1&0&0&0&0&0&0 \\ 1&1&1&0&2&2&0&1&1 \\ 0&1&0&0&1&1&0&2&0 \end{bmatrix} \quad (15)$$

$$[G_2(0)] = \begin{bmatrix} 0&2&1&1&1&1&0&1&0 \\ 0&0&0&0&2&2&0&1&0 \\ 0&0&0&0&1&1&0&2&0 \end{bmatrix} \quad (16)$$

$$[G_3(0)] = \begin{bmatrix} 0&0&0&1&0&0&0&0&0 \\ 2&0&0&1&2&1&1&1&2 \\ 1&0&0&1&2&1&0&0&0 \end{bmatrix} \quad (17)$$

표 3. 식 (13)에 의해 체크된 변수 차수표
Table 3. Variable degree table checked by Eq. (13).

c_i	x_1			x_2			x_3		
	0	1	2	0	1	2	0	1	2
2	√			√				√	
1	√			√					√
1		√		√			√	C_i	
1		√		√				√	
1		√		√					√
2		√			√			√	
2		√			√				√
1		√				√		√	
1		√				√			√
1			√	√				√	
1			√		√			√	
2			√			√		√	
*	2	7	3	6	3	3	1	7	4
	12			12			12		

표 3에서 선택된 첫 번째 제어 입력 변수가 x_2이므로 변수 x_2에 대하여 RME 모듈 M_f의 입력 계수를 다음과 같이 결정할 수 있다.

$$[G_2(1)]=[1\ x_2\ x_2^2][G_2(0)]$$

$$=[1\ x_2\ x_2^2]\begin{bmatrix}0&2&1&1&1&1&0&1&0\\0&0&0&0&2&2&0&1&0\\0&0&0&0&1&1&0&2&0\end{bmatrix}$$

$$=[0\ 2\ 1\ 1\ M_f(1,2,1,x_2)\ M_f(1,2,1,x_2)\ 0\ M_f(1,1,2,x_2)\ 0] \quad (18)$$

식 (18)에서 $[G_1(1)]$을 구하면 다음과 같다.

$$[G_1(1)]=\begin{bmatrix}0&2&1\\1&M_f(1,2,1,x_2)&M_f(1,2,1,x_2)\\0&M_f(1,1,2,x_2)&0\end{bmatrix} \quad (19)$$

표 3에서 두 번째 제어 입력 변수가 x_1이므로 x_1에 대하여 RME 모듈 M_f의 입력 계수를 다음과 같이 구할 수 있다.

$$[G_1(2)]=[1\ x_1\ x_1^2][G_1(1)]$$

$$=[1\ x_1\ x_1^2]\begin{bmatrix}0&2&1\\1&M_f(1,2,1,x_2)&M_f(1,2,1,x_2)\\0&M_f(1,1,2,x_2)&0\end{bmatrix}$$

$$=[x_1\ M_f(2,M_f(1,2,1,x_2),M_f(1,1,2,x_2),x_1)\ M_f(1,M_f(1,2,1,x_2),0,x_1] \quad (20)$$

그러므로 RME 모듈 M_f을 사용하여 식 (20)에 대한 3변수 3치 논리 회로를 설계하면 그림 5와 같은 회로를 설계할 수 있다.

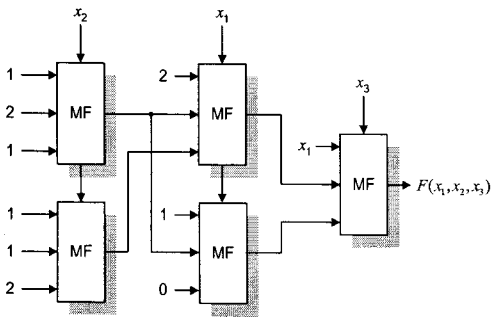


그림 5. RME 모듈 M_f을 사용한 3변수 3치 논리 회로 구현

Fig. 5. Realization of 3 variable ternary logic circuit using RME module M_f.

그림 5에서 3변수 3치 Reed-Muller 전개식을 회로 설계하면 RME 모듈 M_f가 5개 필요하며, 제어변수는 x_2, x_1, x_3의 순서로 구성된다.

V. 비교 및 검토

이 장에는 제시한 Reed-Muller 전개식에 의한 3변수 3치 논리회로를 타 연구의 회로와 비교하였다. 표 4는 3변수 3치 Reed-Muller 전개식에서 6가지 제어 입력 변수의 순서 선택에 대하여 3치 논리 회로를 설계하는데 필요한 RME 모듈 M_f의 수를 나타낸다. 표 4에서 보인 것처럼 제어 입력 변수의 순서를 정하는 방법에 따라 RME 모듈 M_f의 수가 다르다. 만약 제어 입력변수의 순서를 (x_1,x_3,x_2)로 선택하면 9개의 M_f가 필요한 반면에, 제어 입력 변수의 순서로 (x_2,x_1,x_3)를 선택하거나 제어 입력 변수를 (x_2,x_3,x_1)의 순서로 선택하면 5개의 M_f가 필요함을 알 수 있다. 그러므로 최적의 제어 입력 변수의 순서로 (x_2,x_3,x_1)를 선택하면 RME 모듈 M_f이 가장 적음을 알 수 있다.

표 4. 3변수 3치 Reed-Muller 전개식에 대한 제어 입력변수의 선택

Table 4. Selection of control input variable for 3 variable ternary RME.

구 분	1st단	2nd단	3rd단	모듈M_f수
제어 입력 변수 (3!)	x_1	x_2	x_3	8
	x_1	x_3	x_2	9
	x_2	x_1	x_3	5
	x_2	x_3	x_1	5
	x_3	x_1	x_2	9
	x_3	x_2	x_1	8

표 5는 3변수 3치 Reed-Muller 전개식을 설계하는데 필요한 RME 모듈 M_f, 가산회로(A_n), 승산회로(M_n)의 수를 다른 연구 방법과 비교하였다. 최적의 입력 제어 변수의 순서를 선택하는데 Hong 등[9]의 방법은 회로 비용 행렬을 이용하여 각 변수에 대하여 회로 비용을 계산하며, 최소의 회로 비용을 갖는 변수를 첫 번째 제어 입력

변수로 선택하고, 다음에 나머지 변수를 가지고 다시 회로 비용 행렬을 이용하여 제어 입력 변수를 선택한다. Falkowski 등[12]은 Reed-Muller 전개식에서 Kronecker 적에 의해 선형 독립 변환을 수행하여 함수의 계수를 계산하며, 선형 독립 변환은 3치 Reed-Muller 전개식에서 가산 및 승산 연산의 복잡성을 감소시킨다. Jankovic 등[11]은 Reed-Muller 전개식에 대한 고정극성 다항식의 표현의 계산에 대한 표사용 기법을 사용하며, 이 기법은 연산의 간단성, 계산절차의 병렬화에 대한 가능성과 Kronecker 적으로 계수의 계산을 확장할 수 있다. 표 5에서 보인 것처럼 RME 모듈 M_f의 수가 5이고, 가산회로(A_n)가 10개, 승산회로(M_n)가 15개 소요되며, 이는 타 연구에 비하여 설계 면에서 우수함을 알 수 있다. 또한, Hong 등의 방법은 전체 변수에 대한 회로 비용의 탐색 단위 시간 $(3^n)(n+2)(n-1)/2$이 소요되며, Falkowski 등과 Jankovic 등의 방법은 Kronecker 적을 사용하기 때문에 제어 변수 선택에 대한 회로 비용의 탐색 시간이 필요 없다. 그러나 본 논문은 변수 차수표(VDT)를 검사하는 시간만이 필요하므로 이것을 탐색 단위 시간이라 한다면 한번의 탐색 단위 시간이 소요하게 되며, 컴퓨터 프로그래밍에 의한 제어변수를 탐색할 경우 (3^n)의 수행 시간이 소요되므로 탐색 시간이 빠른 장점을 갖는다.

표 5. 3변수 3치 Reed-Muller 전개식에 대한 비교
Table 5. Comparison for 3 variable ternary Reed-Muller expansions.

구 분	직접 계산	Hong & Fei[9]	Falkowski & Fu[12]	Jankovic 등[11]	본 논문
모듈(M_f)	13	8	9	9	5
가산회로 (A_n)	26	16	18	18	10
승산회로 (M_n)	39	24	27	27	15

VI. 결 론

본 논문에서는 Reed-Muller 전개식에 의한 RME 모듈 M_f를 사용하여 3치 논리 회로의 최적 합성 알고리듬을 제시하였다. 제시된 3치 논리 회로의 최적 합성 알고리듬은 Reed-Muller 전개식의 계수에 대하여 모든 변수의 각 차수를 변수 차수표에 표시하고, 각 변수의 0차수 열이 가장 많이 표시된 변수를 탐색함으로서 최적의 제어 입력 변수의 순서를 결정한다. 또한 제시된 알고리듬에 의해 결정된 최적의 제어 입력 변수의 순서를 Reed-Muller 전개식에 의한 RME 모듈 M_f을 이용하여 간단한 나무 구조 모양의 3치 논리 회로를 구현하였다. 따라서 기존에 제시된 3치 논리 회로의 구현 방법보다 향상된 3치 논리 회로의 실현을 보였다.

본 연구는 비교된 타 연구들보다 모듈수 및 연산회로가 다소 간략화 됨을 보였다. 회로 비용 행렬을 이용하여 한 변수에 대한 최소 회로 비용을 계산하는 시간을 탐색 단위 시간으로 가정하면 Hong 등이 제시한 방법의 탐색 시간은 $(n+2)(n-1)/2$의 단위 시간이 소요되는 반면에 본 논문은 변수 차수표(VDT)를 검사하는 시간만이 필요하므로 이것을 탐색 단위 시간이라 한다면 유일하게 한번의 탐색 단위 시간이 소요하게 된다. 또한 컴퓨터 프로그래밍으로 탐색할 경우 Hong 등의 방법은 회로 비용 행렬을 계산하기 때문에 $\{(3^n)(n+2)(n-1)/2\}$ 의 수행 시간이 소요되며, 본 논문의 경우 3^n의 수행 시간이 소요되므로 탐색 시간이 빠른 장점을 갖는다.

참고문헌

[1] M. Kameyama, "Toward the Age of Beyond Binary Electronics and Systems," *IEEE Proc. 20th ISMVL*, pp.162-166, May 1990

[2] D. Etiemble, "On the Performance of Multi-Valued Integrated Circuits: Past, Present and Future," *IEICE Trans. Electron.*, vol. E76-C, No. 3, pp.364-371, Mar 1993.

[3] Z. Zilic and Z. G. Vranesic, "New Interpolation Algorithms for Multiple-Valued Reed-Muller Forms," *IEEE Proc. 26th ISMVL*, pp.16-23, May 1996.

[4] H. Wu, M. A. Perkowski, X. Zeng and N. Zhuang, "Generalized Partially-Mixed Polarity Reed-Muller Expansion and its Fast Computation," *IEEE Trans. Comput.*, vol. 45, No. 9, pp.1084-1088, Sept. 1996.

[5] B. Harking and C. Moraga, "Efficient Derivation of Reed-Muller Expansions in Multiple-Valued Logic

Systems," *IEEE Proc. 22nd ISMVL*, pp.436-441, May 1992.

[6] T. Higuchi and M. Kameyama, "Synthesis of Optimal T-Gate Networks in Multiple- Valued Logic," *IEEE Proc. 9th ISMVL*, pp.190-195, May 1979.

[7] V. H. Tokmen and S. L. Hurst, "A Consideration of Universal Logic Modules for Ternary Synthesis Based upon Reed- Muller Coefficient," *IEEE Proc. 9th ISMVL*, pp.248-256, May 1979.

[8] Z. Hu, X. Wu,"The Logic Synthesis Using Ternary Universal Logic Module U_{hs}," *IEEE Proc. 17th ISMVL*, pp.250-259, May 1987.

[9] Q. Hong and B. Fei, "Fast Synthesis for Ternary Reed-Muller Expansion," *IEEE Proc. 23rd ISMVL*, pp.14-16, May 1993.

[10] B. Fei and Q. Hong, "Calculation of Ternary Mixed Polarity Function Vector," *IEEE Proc. 23rd ISMVL*, pp.236-238, May 1993

[11] D. Jankovic, R. Stankovic and R. Drechsler, "Efficient Calculation of Fixed-Polarity Polynomial Expressions for Multiple-Valued Logic Functions," *IEEE Proc. 32nd ISMVL*, pp.76-82, May 2002.

[12] B. J. Falkowski and C. Fu, "Polynomial Expansions over GF(3) on Fastest Transformation," *IEEE Proc. 33rd ISMVL*, pp.40-45, May 2003.

[13] A. N. Al-Rabadi, "Reversible Fast Permutation Transforms for Quantum Circuit Synthesis," *IEEE Proc. 34th ISMVL*, pp.81-86, May 2004.

저자소개

성 현 경(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, Information & Coding Theory, Cryptography Theory & Security, RFID 설계 및 응용 등