

# OVAG를 이용한 다치조합논리함수의 설계 기법

윤병희\*, 황종학\*, 심재환\*\*, 박준명\*\*\*, 김홍수\*

\* 인하대학교 전자공학과 \*\*인천전문대학 통신과 \*\*\*충주산업대학교 컴퓨터공학과  
인천광역시 남구 용현동 253 402-751

## A Design Techniques of the Multiple-Valued Combinational Logic Functions Using the Output Value Array Graphs

Byoung-Hee Yoon\*, Jong-Hak Hwang\*, Jai-Hwan Sim\*, Chun-Myoung Park\*\*, Heung-Soo Kim\*

\* Dept. of Electronic Eng. In-Ha Univ.

\*\*Dept. of Communication Eng. Incheon Junior College

\*\*\* Dept. of Computer Eng. Chung-Ju National Univ.

E-mail : bhyoon@nownuri.net, bhyoon00@shinbiro.com

### 요 약

다치결정도(Multiple-valued Decision Diagram : MDD)와 순서화된 다치결정도(Ordered MDD : OMDD)는 다치논리함수의 표현에 폭넓게 사용된다.  $p$ 치  $n$ 변수 인 경우  $p^{(n-1)}$ 으로 증가하는 노드의 수는 ROMDD(Reduced OMDD)를 사용하여 현저하게 감소시킬 수 있다. 그러나 다치와 다변수의 경우에는 더욱 많은 공정을 수반하게 된다. 이러한 단점을 보완하기 위해 Honghai Jiang이 제안한 2치시스템에서의 input implicit/output explicit 관계를 갖는 OVAG(Output Value Array Graph)를 사용하여 다치논리함수를 표현한다. 그리고 MDD 표현이 어려운 상황에서 MOVAG(Multi OVAG)를 사용하여 보다 쉽게 출력값을 배열하는 그래프를 이끌어 낼 수 있다.

본 논문에서는 MOVAG의 구성방법과 회로에서 MOVAG로의 변환에 대한 알고리즘을 제안하였고, 알고리즘에 의한 결과를 MDD와 비교하여 노드수 감소에 따르는 처리속도가 개선됨을 검증하였다.

### I. 서 론

최근 집적회로 기술의 발전으로 회로의 형태가 VLSI로 발전되었다. 그러나 단일 칩상에 방대한 양의 회로를 집적하기 위해서는 칩상의 상호 결선의 복잡성과 외부 단자의 증가를 감수해야만 한다. 이러한 문제점들을 해결하기 위해 단일 결선에서 더 많은 양의 정보를 보낼 수 있는 다치논리이론의 연구가 1970년대 초부터 활발히 진행되고 있고<sup>[1]</sup> 최근에는 그래프 이론<sup>[3]</sup>에 근거하여 다

치논리이론을 해석 하고 회로 설계에 적용하는 연구가 진행중이다. R.E.Bryant<sup>[3]</sup>는 S.B.Akers<sup>[2]</sup>의 2치결정도(Binary DD : BDD)에 기초하여 부울 함수를 방향성비순환그래프(Directed Acyclic Graph : DAG)형태의 데이터 구조로 표현한 후 부울 함수를 간략화하는 방법을 제안하였고, D.M.Miller<sup>[4]</sup>는 R.E.Bryant가 제안한 개념을 확장하여 MDD의 개념을 정립하였다. BDD와 MDD는 지금까지 여러 시도에 의해 ROMDD와 같이 공정을 좀더 효과적으로 할 수 있었으나 많은 응용에도 불구하고 BDD와 MDD는 논리 증명과 같은 응용에 사용하기에는 너무 느린 계산 능력을 갖고 있다. 이러한 단점을 보완하기 위하여 Honghai Jiang<sup>[6]</sup>은 2치시스템에서의 입력과 출력사이의 효과적인 관계를 도식적으로 표현하는 OVAG개념을 제안하였다. MDD가 효과없는 상황 즉, 너무 많은 처리시간을 필요로 하는 상황에서는 다치논리함수의 OVAG 변환이 필요하게 된다.

본 논문에서는 OVAG의 개념을 확장하여 다치논리함수 구현에 적용하였고 회로설계를 통해서 MOVAG를 도출하는 알고리즘을 제안하였으며 다치조합논리함수에 대한 회로를 MOVAG 변환을 통해 MDD와 비교·검토하여 그 처리시간이 단축됨을 검증하였다.

### II. MOVAG의 정의와 기본 성질

본 장에서는 본 논문을 전개하는데 필요한 MOVAG에 대한 수학적 배경을 논의한 후 MOVAG의 기본 성질과 정의를 내린다.

1. 수학적 배경

다치조합논리함수  $F$ 의 입력변수의 수를  $N$ 이라 하면 이들  $N$ 개의 입력변수는 집합  $n$ 으로 나눌 수 있으며 각각의 집합은 입력변수  $m(i)(i=1,2,\dots,n)$ 로 구성된다. 이를 식으로 표현하면 다음 식(2-1)과 같다.

$$N = \sum_{i=1}^n m(i) \quad (2-1)$$

레벨  $i$ 에서의 모든 변수를  $X_i = (x_{i1}, x_{i2}, \dots, x_{im(i)})$ 로 표현하고 이때  $X_i$ 를 레벨  $i$ 의 변수배열이라고 한다.

본 논문에서는  $(x_{i1}, x_{i2}, \dots, x_{im(i)}) = (0, 0, \dots, 0)$ 과  $(x_{i1}, x_{i2}, \dots, x_{im(i)}) = (p-1, p-1, \dots, p-1)$ 를 나타내기 위해  $X_i = 0, X_i = p^{m(i)} - 1$ 로 표현한다.

위 내용으로부터 입력행렬(input matrix)  $X$ 를 표현하면 다음 식(2-2)와 같다.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m(1)} \\ x_{21} & x_{22} & \dots & x_{2m(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm(n)} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \quad (2-2)$$

여기서,  $x_{ij} \in \{0, 1, 2, \dots, p-1\}$  이다.

변수배열  $X_i$ 에 Shannon의 확장식<sup>[6]</sup>을 전개하기 위해 표2-1, 표2-2의 연산자를 정의한다.

표 2-1. 4차 (a) conjunction (b) disjunction

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

(a)  $x \& y = \text{MIN}(x, y)$

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

(b)  $x \cup y = \text{MAX}(x, y)$

표 2-2. 다치 부정(negation) 연산자

$X^0$	$X^1$	$X^2$	.....	$X^{p-1}$
0	1	2	.....	$p-1$
1	2	3	.....	0
2	3	.....	$p-1$	0
.....	.....	.....	.....	.....
$p-1$	0	1	.....	$p-2$

위의 표(2-1)과 표(2-2)로부터 변수배열  $X_i$ 에 대한 다치논리함수의 Shannon 확장식을 정의하면 다음 식(2-3)과 같다.

$$f(X_i) = x_{i1}^0 f_0(x_{i2}, \dots, x_{im(i)}) \\ \vee x_{i2}^1 f_1(x_{i2}, \dots, x_{im(i)}) \vee \dots \quad (2-3) \\ \vee x_{i1}^{p-1} f_{p-1}(x_{i2}, \dots, x_{im(i)})$$

여기서,  $X_i = k-1 (1 \leq k \leq p^{m(i)})$ 이고  $x_{ij}^0, x_{ij}^1, \dots, x_{ij}^{p-1}$ 은 표2-2의 부정연산자에 따른다.

위 식(2-3)으로부터 다음과 같은 성질(2-1)을 얻을 수 있다.

- (성질 2-1)
- (1)  $k$ 에 대하여  $F(X)|_{X_i=k-1} \equiv p-1$ 이면,  $F(X) = p-1$ 이다.
- (2) 이외의 경우에는  $F(X) \equiv F(X)|_{X_i=k-1}$ 이다.

2. MOVAG 기본 성질

본 절에서는 MOVAG의 기본 성질을 다음과 같이 정의한다.

(정의 2-1)

MOVAG는 방향성그래프이고 노드들의 집합을  $\delta$ 라 하면, 레벨값배열은  $0, 1, 2, \dots, p-1, *$ 로 구성된다. 만일 레벨값배열의 요소가 노드  $V_k$ 의 child라면  $\text{child}(V_k) \in \delta$ 이다. 그러므로 종단노드가 아닌 노드는 적어도 하나의  $*$ 를 포함하고 종단노드는  $0, 1, 2, \dots, p-1$ 의 값을 갖는다.

(정의 2-2)

OVAG는  $V = (v_1, v_2, \dots, v_{3m})$ 으로 표현되는 루트노드를 가지며 그에 대한 레벨변수배열이  $XV = (x_1, x_2, \dots, x_m)$ 이고 이때 다치함수의 출력값은 아래의 식(2-4)와 같다.

$$f_v = x_1^{p-1} x_2^{p-1} \dots x_m^{p-1} f_{v_1 \vee x_1^p} \vee x_2^{p-1} \dots x_m^{p-2} f_{v_2 \vee \dots} \\ \vee x_1^0 x_2^0 \dots x_m^1 f_{v(p-1)^m} \vee x_1^0 x_2^0 \dots x_m^m f_{v^m} \quad (2-4)$$

여기서,  $x_i^0, x_i^1, \dots, x_i^{p-1} (i=1, 2, \dots, m)$ 는 표2-2의 부정연산자에 따른다.

위 식(2-4)에서  $Xv = k-1 (k=1, 2, \dots, p^m)$ 이고 다음과 같은 성질(2-2)을 갖는다.

(성질 2-2)

- (1) 만일  $v_k = p-1$ 이면,  $f_v = f_{v_k} = p-1$ 이다.
- (2) 만일  $v_k = *$ 이면,  $f_v = f_{v_k} = \text{child}(v_k)$ 이다.

(정의 2-3)

OVAG의 같은 레벨에서 두 개의 노드  $V'$ 와  $V''$ 의 각각의 요소가 같다면 동일한 노드(배열)라고 하고 동일한 요소의 조건은 다음과 같은 성질(2-3)을 갖는다.

(성질 2-3)

$v_k' \in V', v_k'' \in V''$ 의 조건은 1:1 동일함을 의미하며

(1)  $v_k' = v_k$  이면  $v_k = *$  또는  $v_k' = * *$ 이다.

(2)  $f_{v_k} = f_{v_k'}$  이면  $v_k = *$  또는  $v_k' = * *$ 이다

(정의 2-4)

OVAG의 크기는 주어진 입력행렬에 의해 영향을 받으므로 입력행렬의 순서에 따라 OVAG 노드들의 최대수를 다음 표 2-3과 같이 나타낼 수 있다.

표 2-3. 입력행렬 순서에 따른 OVAG 노드의 최대수

레벨	최대 노드수
1	@N <sub>1</sub> 1
2	@N <sub>2</sub> =MIN( p <sup>p-1</sup> , @N <sub>1</sub> * p <sup>m-1</sup> )
3	@N <sub>3</sub> =MIN( p <sup>p-1</sup> , @N <sub>2</sub> * p <sup>m-1</sup> )
⋮	⋮
n	@N <sub>n</sub> =MIN( p <sup>p-1</sup> , @N <sub>n-1</sub> * p <sup>m-1</sup> )

### III. MOVAG 구성

MOVAG는 입력행렬에 따라 MOVAG의 크기가 결정되므로 입력변수를 최적의 순서로 배열하는 것이 주요 관점이다. 본 장에서는 입력행렬선정 알고리즘을 제안하고 MDD와 MOVAG의 관계를 논의한다. 그리고 다치 OVAG 구성에 필요한 알고리즘을 제안하고 이를 바탕으로 MOVAG 구성하는 방법을 논의한다.

#### 1. 입력행렬선정 알고리즘

**[단계1]** 다치함수의 입력변수의 수만큼 각각의 입력행렬 레벨에 입력변수를 할당한다. 이때 i레벨에서는 변수의 수에 따라 노드수가 p<sup>m</sup>으로 확장된다.

**[단계2]** i레벨의 입력행렬에서 레벨에 삽입될 입력변수를 고려한다.

**[단계3]** 고려된 입력변수에 해당하는 레벨에서 다치조합논리함수의 출력값이 최대값을 가질 수 있도록 입력변수를 선택한다.

**[단계4]** 나머지 레벨에 대해서도 단계3을 적용한다.

#### 2. MDD와 그의 OVAG의 관계

본 절에서는 MDD와 그의 OVAG 사이의 관계를 논의하기 위해 간단한 예를 제시한다. p지 2입력변수를 갖는 다치조합논리함수식 F(X)=x<sub>1</sub>x<sub>2</sub>에 대한 MDD는 그림 3-1과 같고 그에 대한 OVAG를 도출하면 그림 3-2와 같다. 여기서 II장의 식(2-2)에 의해 다음과 같은 입력행렬을 선정한다.

예) 입력행렬 :  $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

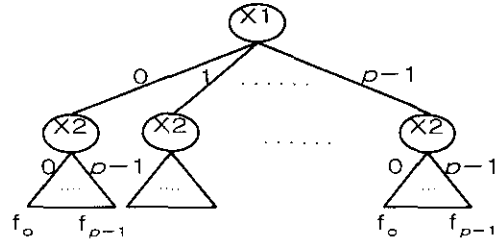


그림 3-1. MDD

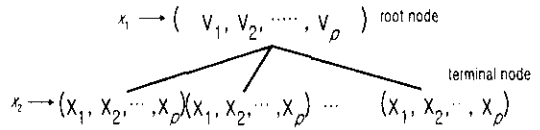


그림 3-2. F(X)=x<sub>1</sub>x<sub>2</sub>에 대한 OVAG

그림 3-2에서 루트 노드의 요소는 0,1,...,p-1,\* 이고 터미널노드의 요소는 0,1,...,p-1이다. 만약, 루트노드의 요소가 모두 \*으로 이루어졌다면 MOVAG의 크기는 최대가 될 것이다.

#### 3. MOVAG 도출 알고리즘

1절과 2절의 내용을 토대로 최종 MOVAG 도출 알고리즘을 제안하면 다음과 같다.

**[단계1]** 주어진 다치조합논리함수의 입력변수로부터 최적의 입력행렬을 선정한다.

**[단계2]** 선정된 입력행렬의 i레벨변수에서 노드를 p<sup>m</sup>으로 확장한다.

**[단계3]** 입력행렬을 레벨별로 나누어서 각각의 레벨에 해당하는 MOVAG의 노드를 단계2를 토대로 도시한다. 이때 주어진 다치조합논리함수식에서 첫 번째 레벨의 입력변수만을 고려하여 출력값이 나오면 0,1,2,...,p-1를 도출한다.

**[단계4]** 이외의 경우에는 성질 2-2와 같이 \*로 표시하여 다음 레벨의 입력변수와 상관하여 비교한다.

**[단계5]** 마지막 레벨에서는 상위 레벨변수배열에 의해 축약된 함수식을 고려하여 다치조합논리함수 출력값을 도출한다.

**[단계6]** 최종 MOVAG를 구성한다.

위의 다치 MOVAG 도출 알고리즘을 흐름도로 나타내면 다음의 그림 3-2와 같다.

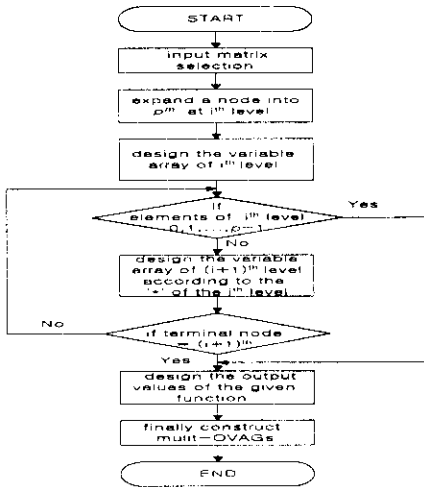


그림 3-2. MOVAG 도출 알고리즘의 흐름도

#### IV. 적용예 및 비교·검토

본 장에서는 앞의 III장에서 제안한 MOVAG를 적용하여 MDD와 비교하고 그 결과를 논의한다.

다음 예에 주어진 3치조합논리함수 ROTDD로 구현하고 제안한 알고리즘에 적용하여 다치OVAG를 구현하면 다음과 같다.

예) 3치 조합논리함수  $F=x+y+z$

입력행렬 :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

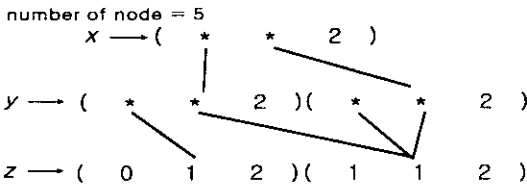


그림 4-1.  $F(x,y,z)=x+y+z$ 의 MOVAG

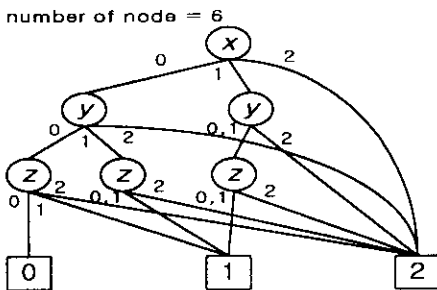


그림 4-2.  $F(x,y,z)=x+y+z$ 의 OTDD

위 예의 결과 OTDD의 노드수는 6개, MOVAG의 노드수는 5개로 노드가 감소된 것을 볼 수 있다. 특히, 다변수함수에서는 노드수 감소 효과는 더욱 향상된다.

#### VI. 결론

본 논문에서는 기존의 다치논리시스템에 도입되고 있는 결정도의 단점을 보완한 새로운 형태의 그래프를 제안하였다. 다치조합논리함수에서 MDD의 구현이 어려운 상황, 즉, 다변수함수에서 노드수가 무한히 증가하는 단점을 보완하기 위하여 MOVAG를 적용하였다. MDD와 MOVAG를 비교하여 MOVAG의 노드수 감소율이 높다는 사실을 검증하였다.

따라서, MOVAG의 구현은 VLSI 설계를 위해 MDD를 사용하는 것보다는 다변수함수에서 MOVAG를 사용하는 것이 처리시간의 감소와 비용을 줄일 수 있다. 그러나 현재는 MOVAG의 적용 범위가 한정되어 있으므로 일반화를 통해 모든 회로의 구현에 적용할 수 있는 MOVAG를 연구하는 것이 필요하다.

#### 참고문헌

- [1] T. Hanyu, M. Kameyama, T. Higuchi, "Prospects of Multiple-Valued VLSI Processors," IEICE Trans. Electron, vol.E76-C, No.3, pp.383-392, March 1993
- [2] S. B. Aker, "Binary Decision Diagrams," IEEE Trans. Comput. vol.C-27, no.6, pp.509-516, Jun. 1978
- [3] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulations," IEEE Trans. Comput. vol.C-35, no.8, pp.677-691, Aug. 1986
- [4] D. M. Miller, "Multiple-Valued Logic Design Tools," IEEE Proc. of Symposium on Multiple-Valued Logic, Sacramto, California, pp.2-11, May. 1993
- [5] Zvi Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill Book Company, Inc. 1978
- [6] Honghai Jiang, Jay C. Majithia, "Suggestion for a New Representation for Binary Function," IEEE Trans. Comput. vol.45, no.12, pp.1445-1449, December. 1996
- [7] Masahiro FUJITA, "Evaluation and Improvements of Boolean Comparison Method Based on Binary Decision Diagrams." IEEE Proc. CAD. pp.2-5, September, 1998
- [8] 윤병희 외4명 "A Representation Technique of Ternary-Valued Combinational Logic Function based on the OVAGs" KITE Proc. Fall Con. pp.711-714, November, 1997