

PLA의 고밀도설계를 위한 다출력논리함수의 최소화기법

임 재 윤*

Multi-output Boolean Function Minimization Algorithms for High Density PLA Design

Jea-Yun Lim*

ABSTRACT

A new algorithm on minimization for multi-output boolean functions is proposed. Base minterm is selected among the minterms that compose the boolean functions. Product function, candidate product term(CPT) and relation function are generated by cube relation of base minterm. For efficient minimization, the properties of multi-output functions are considered. According to the heuristic conditions of adjacent minterm, CPT is changed to prime implicant by relation function. The proposed algorithm can reduce necessary comparisons of minterm and doesn't generate unnecessary prime implicants, it can get more improved minimized results.

Key words : Minterm, Product function, Candidate product term, Relation function Prime implicant, Base Minterm.

1. 서 론

최근 VLSI의 집적도가 증가하고 설계할 회로가 복잡해 짐에 따라, 설계자동화 시스템에 대한 필요성이 증대되었다. 한편, 소비자들의 다양한 주문에 의한 다품종 소량생산의 필요성이 증대됨에 따라 설계비용의 절감이 절실히 요구되고 있다.¹⁾⁻⁴⁾ 이러한 문제점을 해결하기 위한 방안의 하나로 PLA(Programmable Logic

Array), PLD(Programmable Logic Device) 등과 같은 제품들이 개발되어 사용자가 직접 설계한 논리회로를 용이하게 실현 시킬수 있다.^{5),6)} 이중 PLA는 구조가 간단하고 규칙적인 구조를 갖고 있으므로 설계와 변경 및 테스트가 용이하여 논리회로의 설계에 효과적인 수단으로 그 사용이 날로 증가함에 따라 칩 이용율을 높이기 위하여 논리 최소화 기법이 절실히 요구되고 있다.^{7),8)} 기존의 방법들은 대부분 주항생성 및 주항 선택의 과정을 별도로 수행하고 있어 최종해에 포함되지 않는 주항들까지 모두 생성해야 하므로 주항의 개수가 많아질 경우 메모리

* 제주대학교 통신공학과

Dept. of Telecommunication Eng., Cheju Nat'l Univ.

와 계산시간이 지수함수적으로 증가하여 사용이 불가능하였다. 이러한 문제점을 해결하기 위하여 MINI, ESPRESSO-II등이 개발되었으나 논리함수를 구성하는 최소항을 2진수로 다루고 있어 메모리의 효율성이 떨어지고 입.출력 변수의 증가에 따른 계산시간이 크게 증가하는 단점이 있다. Arevalo등의 방법에서는 기준 최소항의 인접도와 같은 거리에 있는 최소항을 탐색하여 적항을 만들어야 하는 단점이 있다.

본 논문에서는 보다 빠른 시간에 최적에 가까운 해를 구할 수 있는 다출력 논리함수 최소화 기법을 제안하며, 이는 주항생성에 필요한 최소항 간의 비교회수가 최대 입력변수의 개수로 제한되므로 종래의 방법보다 비교회수가 크게 줄며 불필요한 주항을 생성하지 않으므로서 불필요한 메모리 및 수행시간의 감소로서 최적의 해를 구할수 있게 할 수 있다.

II. 기본 정의 및 성질

1. PLA의 일반구조

PLA는 ROM과 같은 규칙적인 배열을 갖는 AND-OR Array로 구성되며 다입력, 다출력 논리함수를 실현할 수 있는 논리소자로서, PLA

의 일반구조를 나타내면 Fig. 1 과 같다. 다출력함수를 최소화함으로서 적항수가 감소되어 전체 PLA의 크기를 줄일 수 있으며, 다출력 함수의 경우 단일출력함수는 가능한 적은 수의 적항을 가져야하고, 이 적항들이 각 단일출력함수 전체에 최대한 많이 공유되어야 한다.

2. 기본정의 및 성질

입력변수가 n 인 논리함수는 최소 1개 최대 2^n 개의 최소항으로 표현되어 진다. 이러한 논리함수는 최소화 과정을 통해 보다 적은 수의 주항으로 각각의 주항 또한 보다 적은 변수로 표현할 수 있다.

입력변수가 n 인 두 최소항 M_i, M_j 의 2진수 표현에서 서로 보수논리값을 갖는 비트의 수를 두 최소항의 거리라 하며 최소항 M 과 거리가 1 인 최소항을 인접최소항이라 한다.

논리함수를 구성하는 최소항 중 임의의 최소항 M 과 거리가 1 인 최소항의 개수를 최소항 M 의 인접도라하고 r 로 표시하며, 이때 r 이 0 인 최소항을 독립최소항이라 한다. 최소항 M 의 인접최소항 중에서 이미 생성된 주항에 의해 커버된 최소항의 개수를 rc 로 표시한다.

논리함수를 구성하는 각 최소항들 사이에 서로 대응하는 n 개의 변수가 서로다른 2개의 논리값

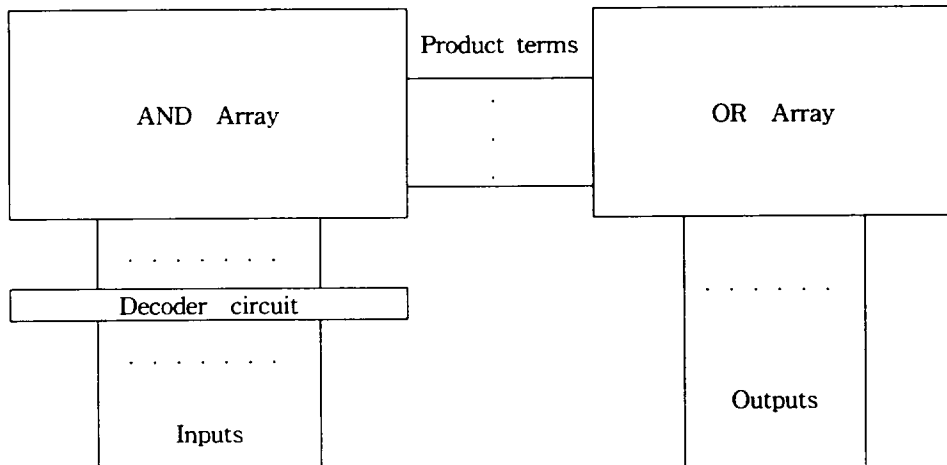


Fig. 1 General structure of PLA

을 가질 때, 이 최소항들은 n 큐브 관계를 가진다고 하며, 논리함수 최소화를 수행할 때, 하나의 기준 최소항을 선택하여 이와 큐브관계를 이루는 최소항을 찾아 n 큐브관계를 갖도록 하여 주항을 선택하며, 이때 주항생성에 필요한 최소항들의 비교횟수를 줄이기 위해 이와 큐브관계를 이룰 수 없는 최소항들을 원 함수로부터 제거시킨다.

무관항을 포함한 f의 최소항을 on-최소항, $f'(1-f)$ 의 최소항을 off-최소항이라 하며, f의 임의의 기준 최소항에 인접된 off-최소항과 기준 최소항으로부터 거리가 n의 최소항이 결합하여 구성된 큐브를 무관큐브(No Relation Cube: NRC)라하고, 무관큐브를 구성하는 최소항을 무관최소항(No Relation Mintrem: NRM)이라 한다. 이러한 무관 최소항들은 기준 최소항과 큐브관계를 갖지 않는 항들로서 n 큐브 형성과는 무관하므로 큐브형성을 위한 최소항의 비교 대상에서 제외시키며, 독립최소항 역시 기준최소항과는 큐브관계를 이룰수 없으므로 비교대상에서 제외시킨다.

함수 f의 기준최소항 B에 대한 무관 최소항 $NRM(B)$ 및 독립최소항을 함수 f로부터 제외한 나머지 최소항들로 구성된 함수를 기준최소항 B의 상관함수라 하고 $fr(B)$ 로 표시한다.

III. 다출력 논리함수의 최소화

1. 기준최소항 선택

논리함수 최소화 과정에서 기준 최소항의 선택은 해의 최적도를 결정하는 중요한 요소의 하나가 되므로 제안한 알고리즘에서는 다음조건에 의해서 기준 최소항을 선택함으로써 생성되는 주항이 가능한 많은 최소항을 커버하도록 하여 주항의 개수를 최소화 시킨다.

[기준최소항의 선택조건]

1. 커버되지 않고 빈도수가 최대인 항
2. 1의 경우가 다수인 경우 인접도가 최소인 항

3. 2의 경우가 다수인 경우 rc가 최대인 항
4. 그외는 임의의 항 선택

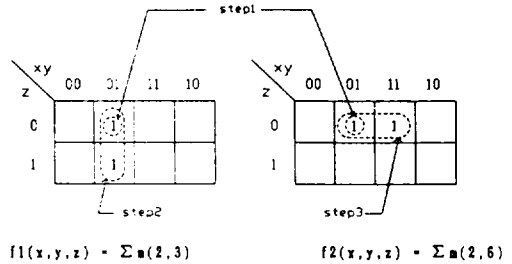


Fig. 2 Base minterm selection with maximum frequency

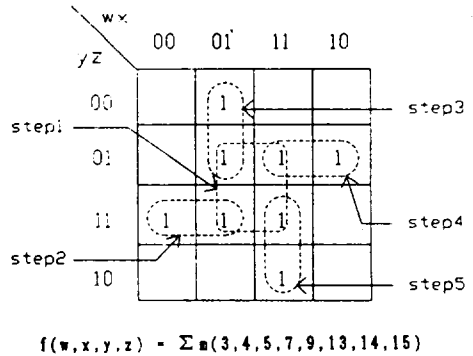


Fig. 3 Base minterm selection with maximum degree of adjacency

2. 후보적항 생성

n 변수 논리함수의 최소항 중에서 선택된 기준최소항 B의 인접도가 r 일때 B는 r 개의 인접최소항 A_i 를 가진다.

$$B = (b_{n-1}, b_{n-2}, \dots, b_k, \dots, b_1, b_0)$$

$$A_i = (a_{n-1}, a_{n-2}, \dots, a_k, \dots, a_1, a_0) \quad 1 \leq i \leq r$$

이러한 기준 최소항과 인접최소항의 차 $|B - A_i|$ 가 P 라면 $\log_2 P$ 번째 비트가 서로 보수의 논리값을 갖게 되며, 그 비트에 해당되는 변수로 소거될 수 있다. 이러한 기준 최소항이 인접최소항 모두와 결합되어 진다고 가정하면 기준최소항의 후보적항이 생성될 수 있으며 이러

한 후보적항 생성함수는 Fig.4.와 같다.

```
CPT_Generate(B,A)
{ CPT(B) = 0;
  for( i = 1; i <= B's Adjacency: i++)
  { K = log2| B - A |;
    CPT(B) |= 1 << K;
  }
}
```

Fig.4 Generation of CPT

3. 적합수

다출력 논리함수에서 기준최소항이 공통으로 포함되는 출력함수들간에 공통으로 포함되는 on-최소항과 무관집합으로 구성되는 항들에 의해 생성된 주항은 출력함수에 공통으로 포함되므로 논리함수의 적합의 수를 최소화 할 수 있다. 이러한 적합수를 구하는 과정을 기술하면 다음과 같다.

- [과정1] 기준최소항이 on-최소항이 되는 함수를 선택한다.
- [과정2] 다출력함수를 이루는 임의의 최소항이 과정 1에서 선택한 함수들에 대해 어느 한 출력이 off최소항이 되면 0 으로 하고, 모든 출력이 on-최소항이면 1 로, 그렇지 않으면 -1로 한다.
- [과정3] 모든최소항에 대해 과정 1,2 반복수행

적합수를 구하는 과정은 각 출력함수를 비트 표시하며 모두 1이되는 조건을 구하여 마스크로 정하고 각 최소항에 대해 비트동작을 한번만 수행하고, 각 상관함수내에서만 수행하게 되므로 계산속도를 빠르게 할 수 있다.

이러한 적합수를 구하는 과정을 기술하면 Fig.5. 와 같다.

예로

$$f_1(x,y,z) = \sum m(0,2,3,4,6) + \sum d(1,7)$$

```
fn_product(B)
{ mask = f(B)[0] & f(B)[1];
  for(each minterm mi)
  { n = f(mi)[0] & mask;
    if(n != mask) fp[mi]=0;
    else
    { if((f(mi)[1] & mask) == mask)
      fp[mi] = 1;
      else
      fp[mi] = -1;
    }
  }
}
```

Fig 5 Product function procedure

$$f_2(x,y,z) = \sum m(0,2,3,4,6) + \sum d(1,7)$$

$$f_3(x,y,z) = \sum m(0,2,3,4,6) + \sum d(1,7)$$

위의 3출력 함수중 최소항 3에 대한 적합수를 구하는 과정으로서, 우선 최소항 3의 출력중 함수값이 1에 해당되는 출력값을 마스크로 하여, 모든 최소항에 부울 논리를 수행해서 적합수를 구하며 Fig. 6. 은 위 함수에 대한 적합수를 구하는 과정을 보인 것으로서 기준최소항 3에 대한 적합수는 $f_p(3) = \sum m(0,3,6) + \sum d(1,7)$ 이다.

	f ₁	f ₂	f ₃	f ₁	f ₃	f _p (3)
0	1	0	1	1	1	1
1	-1	0	1	-1	1	-1
2	1	1	0	1	0	0
3	1	0	1	1	1	1
4	1	-1	0	0	1	0
5	0	0	1	0	1	0
6	1	0	1	1	1	1
7	-1	1	-1	-1	-1	-1

a) initial state b) mask c) fp

Fig. 6 An example for production function

실제 적합수를 구하는 과정은 각 출력함수를 비트표시하며 모두 1이 되는 조건을 구하여 마스크로 정하고 각 최소항에 대해 비트동작을 한번만 수행케 하고, 이를 상관함수 내에서만 구하게 되므로 계산속도를 크게 높일 수 있다.

4. 큐브 확장 과정

선택된 기준최소항으로부터 최대 커버 가능한 큐브쌍을 선택하기 위한 함수로서 가능하면 많은 최소항 및 최대한의 큐브를 선택하여 총 적항의 수 및 리터럴의 수를 최소화 하기 위한 과정으로서, 선정된 기준최소항과 1 큐브관계를 갖는 최소항들을 중심으로 연관 최소항 집합을 구한 후 이 집합 내에서 최대의 정의 최소항 및 최대 큐브관계를 순차적으로 형성해 나가는 과정이다.

[과정1] 기준최소항을 중심으로한 연관 최소항 집합을 선정하여 정렬한후, 기준최소항 B와 1 큐브 관계를 갖는 최소항 중 최대 인접도를 갖는 최소항을 선정하고, 만일 이 최소항이 기존에 선정된 큐브 내에 포함되면 다음 최소항 선정 모든 최소항이 선정되면 과정 5로 간다.

[과정2] 선정된 최소항과 기준최소항과 AMSK를 선정 두 큐브항을 임시큐브 조건항으로 구성한 후, 만일 이 항들이 모두 상관함수 내에 존재하면 이를 큐브조건으로 형성하고 이중 최대인 쌍을 선정하여 MASK에 해당비트를 1로 할당후 이 과정을 재귀적으로 반복수행한다.

[과정3] 이미 형성된 최소항에 의한 큐브 조건쌍이 존재하면, 커버될 1의 출력값이 크고, 최소항의 수가 큰 순으로 새로운 큐브조건상태를 구성한다.

[과정4] 형성된 기준 최소항 및 MASK, 큐브관계를 이루는 최소항의 수 및 출력 1의 개수를 저장한 후 과정 2로 간다.

[과정5] 기준 최소항과 최적 큐브관계를 갖는 모든 최소항에 해당하는 함수값 중 출력값이 1인 항은 -1로 대체하고, fcnt

를 1 감소시킨 후 큐브를 형성하고 복귀한다.

5. 리던던트항 제거

다출력 논리함수의 최소화시 출력 공통큐브를 생성하는 과정에서 기존의 생성된 큐브와 리던던트한 큐브를 생성할 수 있다. 이때 이를 PLA등으로 구성할 경우 적항선에는 영향을 미치지 않으나 다단 논리회로 구성시 이 큐브항에 의해 다단 논리회로가 복잡해 질 수 있으므로 이러한 리던던트항을 제거함으로서 다단논리회로를 간소화 할 수 있다.

주항 C_a 를 구성하는 각 최소항들이 주항 C_b 를 구성하는 최소항들을 포함할 경우 C_a 는 C_b 를 커버한다고하며, 주항 C_a 가 주항 C_b 를 커버하고 이들의 출력 f_i 들 중 정의 출력치가 서로 공통되면 C_b 는 f_i 에 의해 리던던트항이 된다.

이러한 리던던트항 제거 과정은 새로형성된 주항과 기존에 형성된 주항과의 커버가능 조건을 검사하여 만일 리던던트쌍 C_a, C_b 가 존재하고 C_a 가 C_b 를 커버하며 C_b 의 출력값이 1, C_a 의 출력값이 1인 출력이 존재하면 C_a 의 출력을 0으로 한다. Fig. 7은 이러한 예를 보인 것이다

$$\begin{aligned} C_a &: 0 \ 1 \ -1 \ 1 \ 1 \ 0 \ 1 \\ C_b &: -1 \ -1 \ -1 \ 1 \ 0 \ 1 \ 1 \\ & \text{a) Redundant cube} \end{aligned}$$

$$\begin{aligned} C_a &: 0 \ 1 \ -1 \ 0 \ 1 \ 0 \ 0 \\ C_b &: -1 \ -1 \ -1 \ 1 \ 0 \ 1 \ 1 \\ & \text{a) Simplified cube} \end{aligned}$$

Fig 7 Redundant cube simplification process

V. 다출력 논리함수 최적화 알고리즘

이상에서 제안된 다출력 논리함수의 최소화

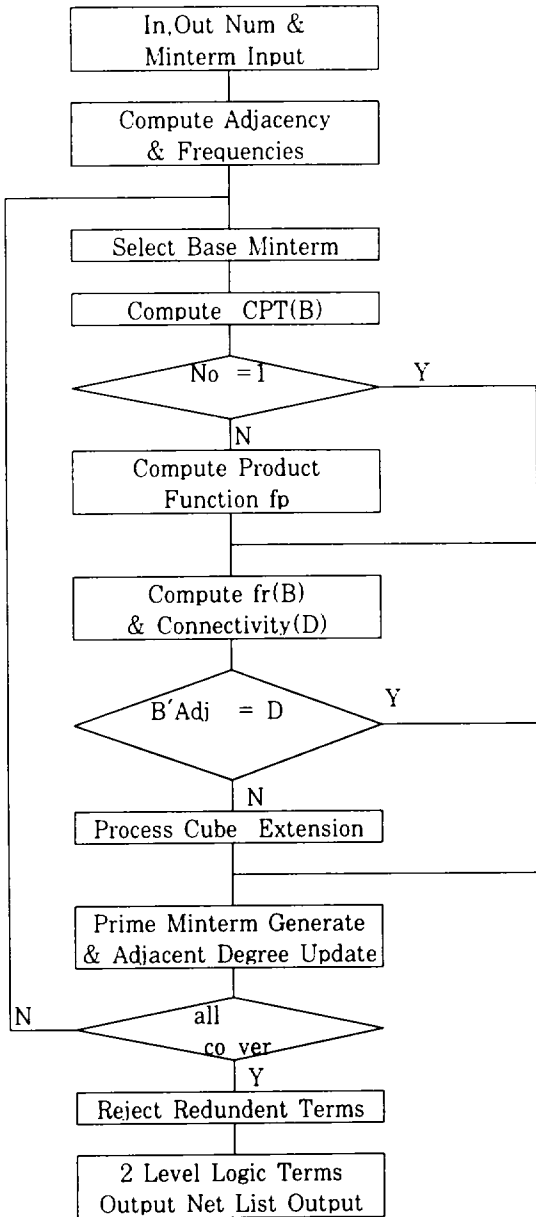


Fig. 8 Flowchart of logic minimization for multi-output boolean functions

알고리즘을 기술하면, 첫째 모든 함수에 대한 인접도 및 빈도수를 계산하여 인접도가 1 이 아닌 최소항은 주항으로 생성하여 출력시키고 빈

도수를 갱신한다. 둘째, 모든 최소항 중에서 기준 최소항 선택조건에 따라 기준최소항 B를 선택한 후 선택된 기준최소항 B의 적함수 $f_p(B)$, 상관함수 $f_r(B)$ 및 결합도를 계산한다. 셋째, 기준최소항 B를 중심으로 적함수 $f_p(B)$ 가 0 이 아닌 최소항들에 대해 큐브확장과정을 수행하여 상관함수의 개수일때까지 큐브확장과정을 재귀적으로 수행한다. 다음으로 최대큐브에 해당하는 주항을 생성하고 해당 출력, 빈도수 및 인접도를 갱신하며 모든항이 커버될 때까지 이 과정을 반복 수행한다. 마지막으로 다단 논리 최소화화를 위해 리던던트항 제거과정을 거친 후 최소화 결과 및 신호선 리스트를 출력하고 종료한다. 이과정에 대한 흐름도는 Fig. 8과 같다.

VI. 실행 비교검토

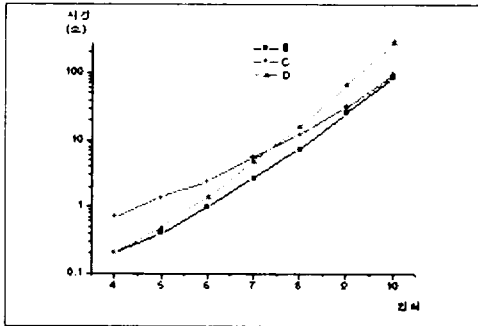
이상에서 제안된 알고리즘을 IBM PC상에서 Visual C++로 프로그래밍하여 여러 가지 예에 대해 다출력 함수의 논리 최소화를 수행하여 그 결과를 기존의 방법들과 비교 검토하여 그 수행 결과를 Fig. 9에 나타내었다. 그 결과 입력수가 증가함에 따라 종래의 방법과 본 논문에서 제안된 방법에 의한 수행 시간의 차이가 커지고 있으며, 적함수에는 거의 차이가 없음을 확인하여 본알고리즘의 유용성을 보였다.

VII. 결론

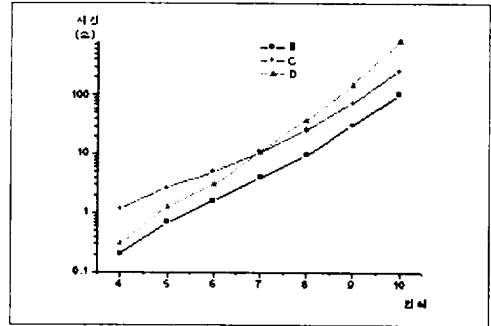
본 논문에서는 PLA의 고밀도 설계를 위한 다출력논리함수의 최소화기법을 제안하였다.

종래의 방법들은 최소항들을 2진수로 처리함으로써 알고리즘 수행에 필요한 메모리가 증가하고 함수를 구성하는 최소항들로부터 주항을 생성하기 위한 최소항들의 비교과정이 복잡하여 입력변수와 출력 개수의 증가에 따른 계산시간이 지수함수적으로 증가하였다.

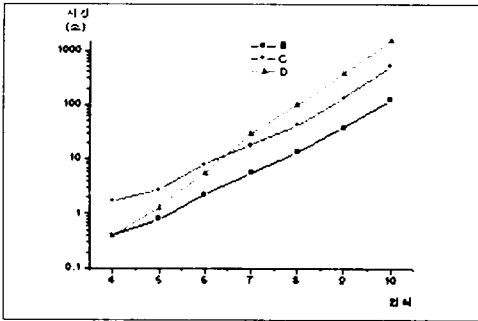
본 알고리즘의 경우 최소항들을 Bitwise로



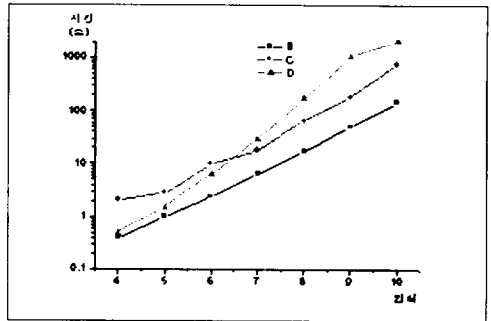
a) outputs = 3



b) outputs = 5



c) outputs = 7



d) outputs = 9

B : This paper C: ESPRESSO-II D: Aleva10

Fig. 9 Comparison of execution time with convention methods

처리하여 주항을 선택한 후 환원과정에서 필요한 최소항들의 비교횟수를 크게 줄여 불필요한 주항을 생성하지 않았으며, 다출력함수의 특성을 고려하여 기준최소항이 공통으로 포함되는 곱함수에 대해서만 최소화를 수행하였다. 본 알고리즘을 각종 예에 적용하여 수행한 결과 기존의 방법인 ESPRESSO-II에 비해서는 2배, Aleva10 방법에 비해서는 40%의 가까운 수행시간의 감소를 보여 본알고리즘의 유용성을 확인하였다. 추후 연구과제로는 본알고리즘을 디코더를 단 PLA의 최소화 확대 적용하고, 이를 다단논리 최소화 방법에 확대 적용하는 것이다.

참고문헌

- 1) Varghese, J., Butts, M. and Batcheller, J., 1993, " An Efficient Logic Emulator System," IEEE Trans. on VLSI Systems, Vol. 1, No.2, pp 171-174.
- 2) Dagenais, M.R., Agarwal, V.K., and Rubin, N.C., 1986, "McBOOLE : A New Procedure for Extract Logic Minimization", IEEE Trans. on CAD., vol. CAD-5 No.1, pp. 229-238.
- 3) Brayton, R.K., Hachtel, G.D., McMullen, C.T., and Sangiovanni Vincentelli, A.L.

- 1984, "ESPRESSO-II: A new logic minimizer for Programmable Logic Arrays", Proc. of IEEE Custom Integrated Circuits Conf., pp 370-376.
- 4) Brayton, R., Detjens, E. et al, 1986, "Multiple-level logic optimization system", in Proc. IEEE Int. Conf. on CAD, pp. 356-359.
- 5) Wolf, W., Keutzer, K. and Akella, J., 1988, "A Kernel-Finding State Assignment Algorithm for Multi-level Logic", IEEE Proc. 25th DA. Conf, pp 433-438.
- 6) Arevalo, Z., and Bredeson, J.G., 1978, "A Method to Simplify a Boolean Function into a Near Minimal Sum-of-Products for Programmable Logic Arrays", IEEE Trans. Computer, Vol. C-27, No.11, pp. 1028-1039.
- 7) Brayton, R., Rudell, R., Sangiovanni Vincentelli, A.L., and Wang, A., 1987, "MISS : a multiple-level logic optimization system", IEEE Trans. on CAD CAD-6, pp. 1062-1081.
- 8) Stornetta, T. and Brewer, F., 1996, "Implementation of an Efficient Parallel BDD Package," 33rd DAC proceeding, pp. 635-640