

Folding 技法을 이용한 高密度 PLA 設計

(A Design Method for High Density PLA by Folding Techniques)

李 在 旻*, 林 寅 七*
(Jae-Min Lee and In-Chil Lim)

要 約

본 논문에서는 Simple Row Folding 기법을 이용한 고밀도 PLA의 설계방법을 제안한다. 서로 다른 입력의 정입력선과 보수입력선을 folding할 수 있게 하고, 두 입력의 정입력선과 보수입력선중 한 입력선만이 서로 folding 가능할 때도 이를 folding함으로써 종래의 방법보다 folding률을 증가시킨다. Folding쌍선택 및 열(column) 배열을 위한 알고리즘에서는 주변회로와의 연결관계에 대한 제한조건을 고려하고, 열순서의 효과적인 배열방법을 제시한다. 제안한 전체 알고리즘을 AT (XENIX) 상에서 C언어로 실현한다.

Abstract

In this paper, a new design method for high density PLA by a simple row folding technique is proposed. The normal input line and its complement line of different two input lines are folded on the same row. Especially the only one input line pair of the different two input lines are folded by this method. Consequently, the results of folding are better than those of conventional methods. An efficient technique of ordering columns is described. Also, constraints about outside circuitry are considered in this algorithm. The proposed algorithm has been implemented on a personal computer by C language.

I. 序 論

최근 VLSI 기술이 발전함에 따라 그 설계기술은 최적의 설계시간과 논리적 정확성이 요구되고 있다. 이러한 요구에 부응하여 Array Logic은 VLSI 설계에 널리 사용되어 왔으며 특히 PLA(Programmable Logic Array)는 규칙적인 구조를 갖고 있어 설계 및 수정이 용이하고 임의의 다출력 논리함수를 바로 실현할 수 있기 때문에 다품종 소량생산을 위한 유용한 수단으로 그 사용이 날로 증가하고 있다. 그러나 PLA로 다출력 논리함수를 바로 실현하는 경우 평균 칩 이용율은 4%

~10%에 불과하며, Random Logic에 비해 약 10%~20% 정도 칩 손실을 가져온다.^{1),2)} 따라서 동일한 논리함수를 PLA로 실현하면 Random Logic에 비해 칩의 크기가 커지고 동작속도도 떨어지게 된다. PLA의 칩 이용율을 향상시키기 위한 방법으로는 논리함수의 최소화, 디코우더를 사용한 설계, 그리고 folding 기법 등이 있다. 이 가운데 folding에 의한 방법은 직접적인 면적축소가 가능한 방법으로서, 최초로 wood에 의하여 제안되었으며,³⁾ Hachtel 등에 의해서 구체적인 알고리즘⁴⁾이 제시되었고, 계속해서 많은 연구가 행해지고 있다.^{5),6)} 그런데 이들 folding 기법^{3),4)}은 AND array에서 한 입력에 대하여 그 정입력선과 보수입력선을 동일 입력선으로 간주하여 folding을 행하였다.

본 논문에서는 PLA의 구조적 특성을 고려하여 AND

* 正會員, 漢陽大學校 電子工學科
(Dept. of Electronic Eng., Hanyang Univ.)
接受日字: 1986年 4月 8日

array의 한 행상에 정입력선과 보수입력선을 별개의 입력으로 구별하여 folding하고, 두 입력의 정입력선과 보수입력선중 어느 한 입력선만이라도 서로 folding 가능하면 이를 folding함으로써 종래의 방법에 비해 folding쌍의 수를 근본적으로 증가시켜 folding율을 향상시킬 수 있는 새로운 고밀도 PLA 설계방법을 제안한다. 또한 주변회로와의 연결관계 및 대규모 PLA folding시 계산복잡도가 매우 커지는 것을 고려하여 folding쌍 선택을 위한 열 배열을 효과적으로 행할 수 있는 휴리스틱 알고리즘을 제시한다. Folding 기법은 그 설계기술에 따라 여러가지 종류로 구별되는데^{1), 2)} 본 논문에서 대상으로 하는 folding 기법은 Simple Row Folding (SRF)이다.

II. 최적 Folding의 원리 및 Folded PLA 구성

1. 최적 Folding의 원리

(정의 1) PLA내 임의의 두 입력 및 출력이 동일 열상에 프로그램된 소자를 갖고 있지 않을 때 이들은 서로 disjoint하다고 한다.

기본 PLA에서

입력변수의 집합을 $X = \{x_1, x_2, \dots, x_i, x_j, \dots, x_n\}$,

출력변수의 집합을 $F = \{f_1, f_2, \dots, f_i, f_j, \dots, f_m\}$,

적항선의 집합을 $P = \{p_1, p_2, \dots, p_i, p_j, \dots, p_n\}$

이라 하면 행 folding에 의해 disjoint 한 (x_i, x_j) 및 (f_i, f_j) 가 동일 행 상에 놓이며 열 folding에 의해 disjoint한 (p_i, p_j) 가 동일 열 상에 놓인다.

최적 folding은 $\{(x_i, x_j) | x_i, x_j \text{는 입력 folding쌍}\} + \{(f_i, f_j) | f_i, f_j \text{는 출력 folding쌍}\}$ 이 최대가 되도록 (x_i, x_j) 및 (f_i, f_j) 를 선택하는 것으로 이것은 NP 완전문제가 알려져 있다.^{1), 2)}

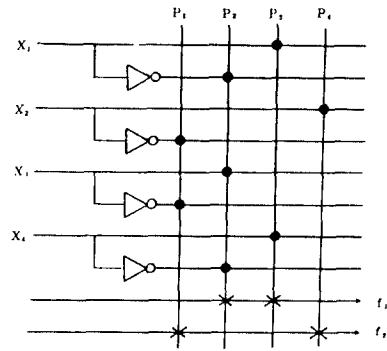
그림 1의 (a)는 기본 PLA를 나타낸 것이고 (b)는 그에 대한 특성 행렬이다.

여기서 적항선은 열에, 입출력선은 행에 배열한다.

그림 1에 나타낸 PLA의 칩 이용율(U_F)은 식(1)에 의하여 구해진다.

$$U_F = \frac{N_{AN} + N_{OR}}{S} (\%) \tag{1}$$

여기서 $S (= N_F \times (2N_i + N_o))$ 는 PLA의 크기, N_F 는 적항선의 수, N_i 는 입력의 수, N_o 는 출력의 수, N_{AN} 은 AND array의 프로그램된 소자수, N_{OR} 은 OR array의 프로그램된 소자수를 나타낸다. 그림 1의 경우 $N_F = 4$, $N_i = 4$, $N_o = 2$, $N_{AN} = 8$, $N_{OR} = 4$ 이므로 $U_F = 30(\%)$ 가 된다. Folding에 의해 동일 수평선 및 수직선 상에 서로 다른 입출력들을 놓음으로써 S를 줄일 수 있고 따라서 U_F 가 향상될 수 있다.



(a) 기본 PLA

	p_1	p_2	p_3	p_4				
x_1	2	0	1	2	AND array	OR array		
x_2	0	2	2	1			0	보수입력선에 프로그램
x_3	0	1	2	2			1	정입력선에 프로그램
x_4	2	0	1	2			2	don't care
f_1	2	1	1	2				
f_2	1	2	2	1				

(b) 특성 matrix

그림 1. 기본 PLA 및 특성 행렬

Fig. 1. Basic PLA and its personality matrix.

2. Folded PLA의 구성

종래의 논문들^{1), 2)}에서는 folding쌍 선정시 임의의 입력 신호선에 대해 그 정입력선과 보수입력선을 동일 신호선으로 취급하였다.

그림 2의 folded PLA 구성 방식중 (a)는 정입력선과 보수입력선을 동일한 row에 folding하는 방식을 나타내고 (b)는 서로 다른 입력의 정입력선과 보수입력선중 한 선만 folding하는 방식을 나타낸다.

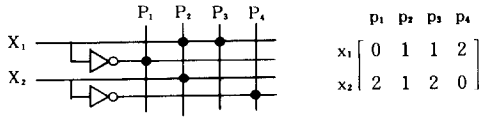
여기서 그림 2의 (a), (b)의 각 열은 열 배열 알고리즘 적용후 최종배열된 상태를 나타낸다.

III. Folding쌍 선택 및 열 배열

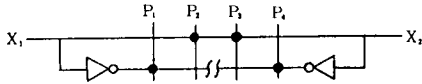
1. Folding Pair Graph(FPG)

PLA의 모든 가능한 입출력쌍에 대한 folding쌍 관계를 라벨 그래프(labeled graph)로 표현하여 이를 FPG(Folding Pair Graph)라 하며 FPG를 구성하는 방법은 다음과 같다. PLA의 모든 입출력 변수들은 FPG의 절점(vertex)이 된다. 또 각 절점 사이에 다음과 같은 folding쌍 관계(ℓ) ~ (n)중 해당 관계기호 ℓ, m, n 을 라벨로 갖는 절선(edge)을 연결한다.

[Folding쌍 관계]

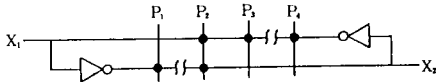


(1) 기본 PLA



	P_1	P_2	P_3	P_4
x_1	0	1	1	2
x_2	2	1	2	0

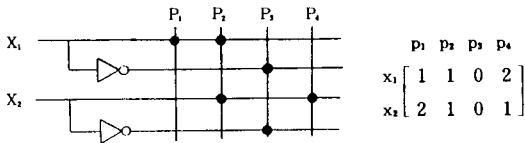
(2) 종래의 방법으로 folding 불가능한 예



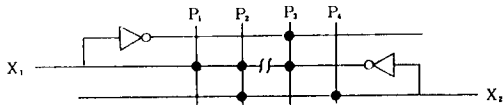
	P_1	P_2	P_3	P_4
x_1	0	1	1	2
x_2	2	1	2	0

(3) 두 입력의 보수선끼리 folding 가능한 예

(a)



(1) 기본 PLA



	P_1	P_2	P_3	P_4
x_1	1	1	0	2
x_2	2	1	0	1

x_1 의 정입력선과 x_2 의 보수
입력선이 disjoint하다.

(2) Disjoint한 하나의 선만을 folding

(b)

그림 2. 새로운 folded PLA 구성

- (a) 정입력선과 보수입력선을 folding
- (b) 서로 disjoint한 한 선만을 folding

Fig. 2. New method for folded PLA.

- (a) Folding of normal input line and complement input line.
- (b) Folding of one disjoint line pair.

(관계 l) 두 입력변수가 동일한 열 상에 공통으로 프로그램된 소자를 전혀 갖고 있지 않는 관계.

(관계 m) 두 입력변수가 동일한 열 상에 그림3과 같이, 특성 행렬내 공통으로 프로그램된 적항선을 갖고 있는 관계.

(관계 n) 두 입력변수가 동일한 열 상에 그림4와 같은, 특성 행렬내 공통으로 프로그램된 적항선을 갖고 있지 않은 관계.

관계 n 은 관계 l, m 의 folding쌍에 대한 열의 순서가 결정된 후 적용한다.

그림 5는 그림 1의 PLA에 대한 FPG를 나타낸 것이다.

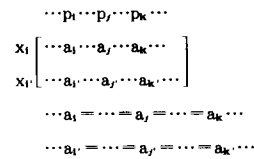


그림 3. 관계 2의 조건

Fig. 3. Condition for relation 2.

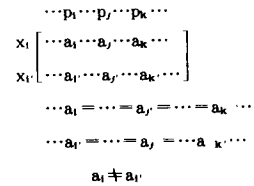


그림 4. 관계 3의 조건

Fig. 4. Condition for relation 3.

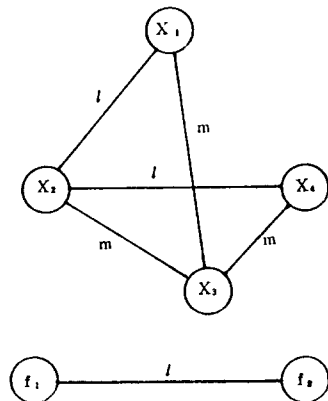


그림 5. 그림 1의 회로에 대한 FPG

Fig. 5. FPG for the circuit in fig 1.

2. Folding쌍 선택

FPG로부터 다음의 조건 ℓ, m, n 에 따라 folding 쌍을 선택한다.

(조건 1) FPG의 절점 가운데 그 차수(degree)가 최소인 것을 선택한다. 차수가 같을 때는 임의로 선택한다. 이것은 최소 차수의 절점은 먼저 folding 함으로써 나머지 절점(선택한 절점보다 차수가 큰 절점)의 folding 가능성을 높이기 위함이다.

(조건 2) 조건 1에서 선택한 절점의 쌍으로서 선택할 절점은 그 차수가 최대인 것을 선택한다. 이것은 folding 가능성이 높은 절점을 선택함으로써 열순서에 모순이 발생할 확률을 감소시킬 수 있는 조건이 된다.

관계 ℓ, m 의 folding쌍에 대한 열의 순서가 결정된 후 관계 n 을 만족하는 folding쌍에 대해 다음 조건 3을 적용한다.

(조건 3) 관계 n 의 folding쌍에 대해 조건 1, 2를 적용한다.

이상의 조건에 따라 그림 5의 FPG로부터 folding 쌍을 선택하면 다음과 같다.

(f_1, f_2)

(x_1, x_2)

(x_4, x_3)

3. 열 배열

FPG로부터 선택한 folding쌍들은 III절의 관계 ℓ, m, n 중 그 어느 한 관계를 만족하게 된다. 각 관계를 갖는 folding쌍들은 그 프로그램된 소자들이 행 상에서 좌우로 분리되어야만 folding 가능하다. 따라서 각 folding 쌍에 대한 관계로부터 열의 배열이 필요하다. 선택된 folding쌍에 대한 열의 순서를 결정하기 위해 다음과 같은 정의를 이용하여 알고리즘을 제시한다.

(정의 2) $C_k(k=1, \dots, n$ 단, n 은 열의 수)는 특성행렬내 각 열, $C_k[]$ 은 folding쌍의 좌우 위치관계에 대한 라벨을 저장하는 array, $C'_k[]$ 은 임의의 C_k 들을 저장하는 array, $C_k - []$ 은 C_k 의 이전 값, 그리고 $C'_k - []$ 은 C'_k 의 이전 값을 나타낸다.

[열 배열 알고리즘]

(단계 1) FPG로부터 folding쌍을 선택한다.

(단계 2) Folding쌍 관계 ℓ, m 에 따라 다음의 조작을 행한다. 즉 관계 ℓ 인 경우, folding쌍을 좌우에 임의로 위치시키고 정입력선과 정입력선을, 보수입력선과 보수입력선을 서로 disjoint하도록 열 배열을 행한다. 이때 앞서 결정된 열 순서와 모순이 발생하면 좌우 위치를 바꾼다. 다시 모순이 발생하면 다른 folding 쌍을 선택한다.

(단계 3) 관계 m 인 경우 정입력선과 보수입력선을 쌍

으로 단계 2와 같은 조작을 행한다.

(단계 4) 모든 folding쌍에 대해 같은 과정을 반복한다.

(단계 5) Folding 가능한 모든 쌍이 고려되었으면 열 배열 알고리즘을 종료한다.

```

/* Algorithm for Column Ordering */
Col_ord()
{
    while (folding pair =  $\emptyset$ ) {
        select a new( $X_i, X_j$ ) or ( $F_i, F_j$ );
        for ( $i = 1; i < n+1; i++$ )
            if (the folding pair relation is 1) {
                #  $C_i < C_j$  //  $C_i$  is left and  $C_j$  is right #/
                 $C_i' < C_j'$  //  $C_i'$  is the complement of  $C_i$  //
                column order setted;
                if (Conflict occurs in column order) {
                     $C_j < C_i$ ;
                     $C_j' < C_i'$ ;
                    if (Conflict occurs in column order)
                        break;
                }
            }
            else if (the folding pair relation is m) {
                 $C_i < C_j'$ ;
                 $C_i' < C_j$ ;
                Column Order Setted;
                if (Conflict occurs in Column order) {
                     $C_j < C_i$ ;
                     $C_j' < C_i'$ ;
                    if (Conflict occurs in column order)
                        break;
                }
            }
    }
    // All column setted #/
    Sorting all columns;
}

*
if (( $C_i < C_j$ ) and ( $C_i' < C_j'$ )) {
     $C_i[j] <--- C_i[j]$ ;
     $C_j[i] <--- C_j[i] + 1$ ;
     $C_i'[i] <--- C_j'$ ;
     $C_j'[j] <--- C_j'[j]$ ;
}
if (( $C_h < C_j$ ) and ( $C_h' < C_j'$ )) {
     $C_h[j] <--- C_h[j]$ ;
     $C_h'[i] <--- C_h'[i], C_i, C_i'[i]$ ;
     $C_j'[i] <--- C_j'[i]$ ;
     $C_i'(C_h[j])[i] <--- C_i'(C_h[j])[i] + 1$ ;
}
// h, i, j < k #/

```

그림 1에 열 배열 알고리즘을 적용하면 그림 6과 같다.

C_k	p_1	p_2	p_3	p_4
$C_k[]$	2	1	0	2
$C'_k[]$	0	1, 2, 4	1, 2, 3, 4	0

그림 6. 그림 1의 회로에 대한 열배열 결과
Fig. 6. Column ordering of circuit in fig. 1.

IV. 제한조건을 고려한 SRF 알고리즘

PLA folding 시 반드시 고려해야할 것은 주변회로와의 연결관계이다. 주변회로와의 연결관계를 고려하지 않은 알고리즘으로 실제 회로를 구성하면 연결이 복잡해지고 칩 이용률도 떨어진다. 본 논문에서는 Simple Row Folding(SRF)시 주변회로와의 연결관계에 대한 제한조건으로서, 서로 다른 입력 및 출력 가운데 서로 인접해야 하거나 좌우 어느 한 쪽에 반드시 놓여야 하는 조건 등을 고려한다. 이러한 제한조건은 알고

리즘 구성시 folding쌍 선정 및 선정순서를 정하는 과정에서 고려되어 진다. 제한조건을 고려한 SRF 전체 알고리즘은 다음과 같다.

[SRF 알고리즘]

(단계 1) Folding할 PLA의 특성 행렬을 입력한다.

(단계 2) FPG를 작성한다. 이때 제한조건을 고려하여 같은 방향에 위치할 입출력들은 folding쌍에서 제외한다.

(단계 3) 관계 ℓ, m 의 라벨을 가진 절점에 대해서 III 절의 조건1, 2를 적용하여 folding쌍을 선택한다. 이때 앞서 선택된 입출력과 인접해야 하는 입출력은 우선적으로 선택한다. 최소 차수의 절점을 왼쪽에 최고 차수의 절점을 오른쪽에 위치시킨다.

(단계 4) 선택된 folding쌍에 대해 열을 배열한다. 이때 앞서 생성된 열의 배열순서와 모순이 발생하면 folding쌍의 좌우 위치를 교환한다. 다시 모순이 발생하면 단계 3으로 가서 다음 folding쌍을 선택한다.

(단계 5) 관계 ℓ, m 의 모든 folding쌍 선정이 끝나면 열 순서의 라벨을 오름차순으로 배열함으로써 적항선의 순서를 결정한다.

(단계 6) 조건 3을 적용하여 관계3의 folding쌍을 구한다.

그림 7은 SRF 알고리즘의 흐름도를 나타낸 것이다. 이상의 SRF알고리즘을 임의의 PLA에 적용한 예를 그림 8에 나타낸다.

V. 알고리즘 실현 및 검토

제안한 SRF 알고리즘을 AT (XENIX) 상에서 C언어로 프로그램하였다. 임의의 PLA에 대한 특성 행렬과 주변회로와의 연결관계에 대한 입출력 위치의 제한조건을 입력데이터로 주면 모든 folding쌍 및 최종 배열된 열의 순서를 출력으로 얻을 수 있게 하였다.

표 1은 임의의 PLA에 제안한 SRF 알고리즘을 적용한 결과를 나타낸 것이다. 실제 사용되고 있는 PLA의 평균 프로그램율은 약 10% 정도이나 본 논문에서는 이 보다 높은 20%~30%의 프로그램율을 갖는 PLA를 예로 택하였다.

실험을 통하여 본 논문에서 제안한 새로운 설계방법에 의해 종래방법으로 folding 불가능한 입출력들이 folding 가능하게 됨으로써 folding율이 향상됨을 확인하였다.

VI. 結 論

본 논문에서는 새로운 folding기법을 이용한 고밀도 PLA의 설계방법을 제안하였다.

종래의 방법들은 실제 PLA의 구조적 특성을 충분히 고려하지 않고, 정입력선과 보수입력선을 동일한 입력선으로 취급하여 folding함으로써 folding율이 그리 높지 못한 반면, 본 논문에서 제안한 방법은 정입력선과 보수입력선을 구별하여 서로 다른 입력의 정입력선과 보수입력선이 folding쌍을 이루게 하였으며 특히 두 입력선의 정입력선과 보수입력선 가운데 한 선만이 folding 가능할 때 이를 folding함으로써 folding율을 향상시킬 수 있었다. 또한 FPG로부터 선택한 folding 쌍들에 대한 열의 순서를 효과적으로 결정하는 방법을 제시하였으며, 입출력의 위치에 대한 제한조건을 고려하여 줌으로써 실제 PLA에 적용할 때 유용한 결과를 얻을 수 있게 하였다. 제안한 전체 알고리즘을 AT (XENIX) 상에서 C언어로 프로그램하여 임의의 PLA에 대한 특성 행렬과 주변회로와의 연결관계에 대한 제한조건을 입력데이터로 주어 모든 folding쌍 및 열의 최종 배열순서를 얻을 수 있게 하였다. 제안한 folding 기법을 이용하여 PLA를 설계하는 경우 종래의 방법에 비해 시간 및 비용절감은 물론 칩 이용율을 더욱 개선시킬 수 있을 것으로 기대된다.

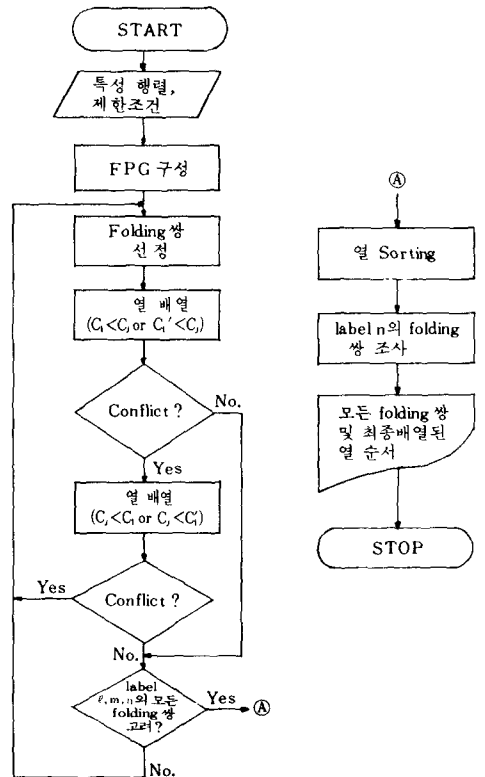
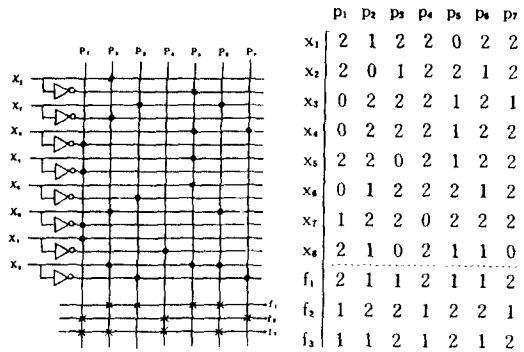
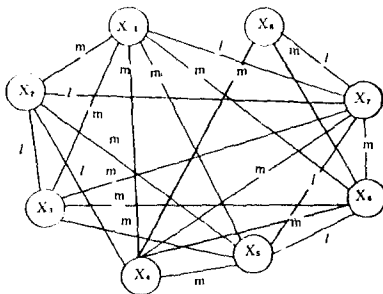


그림 7. SRF알고리즘의 흐름도
Fig. 7. Flowchart of SRF algorithm.



(a) 기본 PLA 및 특성 행렬



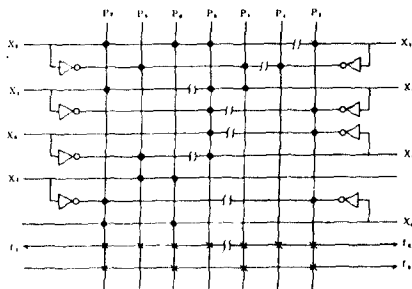
(b) FPG

(f_1, f_2)
 (x_8, x_7)
 (x_3, x_1)
 (x_4, x_5)
 (x_7, x_6)

C_k	p_1	p_2	p_3	p_4	p_5	p_6	p_7
$C_k(\quad)$	2	0	0	2	1	0	1
$C_k^*(\quad)$	0	1,4	1,4	0	1,4	1,4	0
		5,7	5,7		5,7	7	

(c) folding 쌍

(d) 열 배열



(e) Folded PLA

그림 8. SRF 알고리즘에 의한 folded PLA 구성 예
 Fig. 8. Example of folded PLA by SRF algorithm.

표 1. 알고리즘의 적용 예

Table 1. Example of algorithm application.

sample PLAs				proposed	algorithm
N_1	N_0	N_p	U_r (%)	folding pairs	reduced area (%)
6	3	7	20	4 (1)*	40
8	4	8	22	5 (1)	40
10	6	8	20	6 (0)	43
12	4	9	24	7 (1)	39
16	8	14	20	10(2)	40
20	10	16	26	11(1)	36

* () → 서로 다른 입력의 정입력선과 보수입력선 중 한 선만이 folding된 쌍의 수.

參 考 文 獻

- [1] K. Inagaki, "Design of high density programmable logic arrays," *Trans. IECE Japan* vol. J63-D no.9, pp. 739-746, Sept. 1980.
- [2] R.A. Wood, "A high density programmable logic array chip," *IEEE Trans. Computers*, vol. C-28, pp. 602-609, Sept. 1979.
- [3] S. Muroga, "VLSI system design," *Wiley* 1982.
- [4] G.D. Hachtel, A.R. Newton, and A.L. Sangiovanni-Vincentelli, "An algorithm for optimal PLA folding," *IEEE Trans. Computer Aided Design of Integrated Circuits and Syst.*, vol. CAD-1, pp. 63-77, April 1982.
- [5] G.D. Micheli, A.L. Sangiovanni-Vincentelli, "PLEASURE: A Computer program for simple/multiple constrained/unconstrained folding of Programmable Logic Arrays," *Memorandum* no. UCB/ERL M82/57, Aug. 9, 1982.
- [6] W. Grass, "A Depth-First Branch-and-Bound Algorithm For Optimal PLA Folding," in *proc. 19th Design Automation Conf.*, pp. 133-140, 1982.
- [7] J.R. Egan, C.L. Liu, "Bipartite folding and partitioning of a PLA," *IEEE Trans. Computer Aided Design*, vol. CAD-3, no. 3, July 1984.
- [8] David C. Yeh and Timothy N. Trick, *Suboptimal and Optimal Algorithms for PLA Folding*, IEEE proceedings of ISCAS, pp.1467-1470, 1985.

- [9] Q. Yu, O. Wing, "*Interval-Graph-Based PLA Folding*," IEEE proceedings of ISCAS, pp. 1463-1466, 1985.
- [10] D. Makarenko, J. Tartar, "*An efficient algorithm for the optimal folding of PLAS*," IEEE ICCAD, pp. 57-60, Oct. 1985.
- [11] M. Luby, U. Vazirani, and A. Sangiovanni-Vincentelli, "*Some Theoretical Results on the Optimal PLA Folding Problem*," in Proc. Int. Circ. and Comp. Conf., pp. 165-170, Oct. 1982.
- [12] S.Y. Hwang, R.W. Dutton and T. Blank, "*A Best-First Search Algorithm for Optimal PLA Folding*," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 433-442, July 1986.
- [13] Y.S. Kuo, C. Chen and T.C. Hu, "*A Heuristic Algorithm for PLA Block Folding*," in 22nd Des. Autom. Conf. pp. 744-747, June 1985.
-