論 文

32～1～3

EXANOR를 사용한 Three-Level PLA

The Three-Level PLA Design Using EXANOR

趙 東 燮*·李 鍾 元**·黃 熙 隆***

(Dong-Sub Cho · Jong-Won Rhee · Hee-Yeung Hwang)

Abstract

This paper deals with the three-level PLA constructed by EXCLUSIVE-OR, AND, and OR. (abbreviated as EXANOR).

Most PLA circuits have constraints on minimum chip area and minimal input lines. Thus, the reduction of PLA chip area is an important factor in design of logic circuits.

In this paper, newly constructed architecture of PLA is proposed and then, its reduction effect is proved theoretically and some of selected examples are illustrated for designing three-level PLA circuits.

1. Introduction

Programmable logic arrays (PLA's) are known as an approach to implement logic circuits having low production potential [1]-[3]. In this paper, two types of PLA is discussed; two-level PLA and three-level PLA.

First of all, the problem to implement PLA are;

1) the number of inputs
2) the number of outputs
3) the number of product term lines
4) the number of NOT gates
5) logic time delay

In this paper, 3) and 4) are discussed for reduction of PLA size. The three-level PLA using EXANOR is shown in Fig. 1.

In section 2, two types of three-level PLA structures are introduced, in section 3, how the three-level PLA is implemented is introduced, and in section 4, implementation of three-level PLA circuits is introduced. And then in section 5, the PLA sizes of two types of PLA are compared with the conventional PLA.

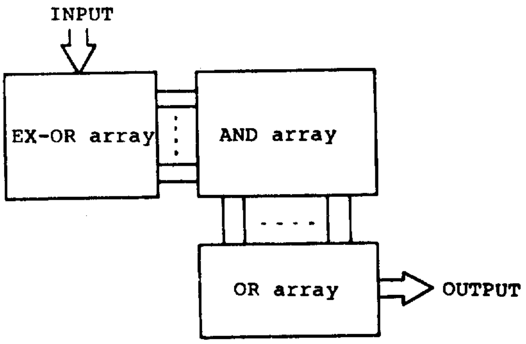

Fig. 1. Three-level PLA using EXANOR

2. Two Types of Three-Level PLA

(EXCLUSIVE OR-AND-OR)

Three-level PLA which is treated in this paper, is divided into the following two types.

TYPE 1: First type of three-level PLA implementation is shown in Fig. 2. and its MOS structure in Fig. 3.

* 正 會 員：서울大 大學院 電子計算機工學科 博士課程
** 正 會 員：서울大 大學院 電子計算機工學科 硬士課程
*** 正 會 員：서울大 工大 電子計算機工學科 副教授·工博
接受日字：1982年 8月 13日

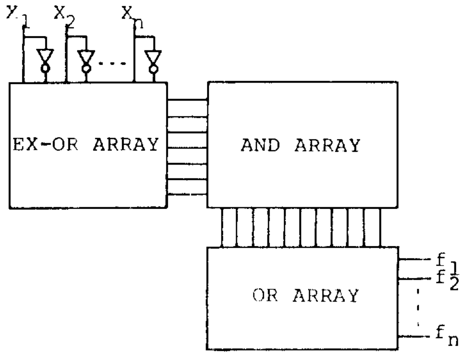

Fig. 2. The number of input; 2n

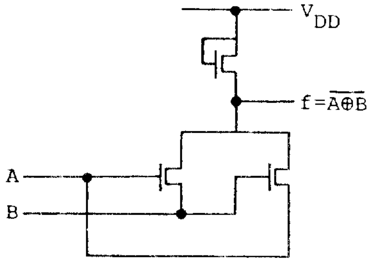

Fig. 3. The strusture of EX-OR array

As shown in Fig. 2, inputs and their inverted inputs are used. So, the number of input lines is 2n. (where n is the number of input variables)

TYPE 2: Second type of three-level PLA implementation is shown in Fig. 4. and its MOS structure is in Fig. 5.
The number of input lines is n+2.

In this case, NOT gates is needless to be used. Instead, X and X is implemented by $X \odot O$ and $X \oplus 1$, respectively. In every row, EX-OR gate which has MOS elements is located to the first part of PLA. So, the number of MOS's required for circuit design can be reduced.

3. How to implement The Three-level PLA

In this section, some definitions and theorems are derived and two methods to implement three-level PLA are introduced.

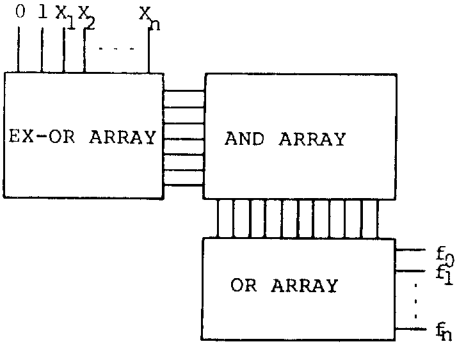

Fig. 4. The number of inputs; n+2

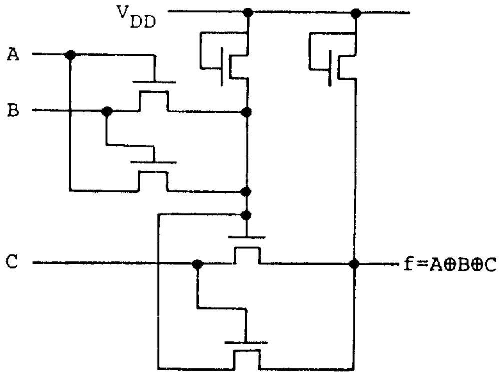

Fig. 5. The structure of Ex-OR array

A. Definitions for implementing three-level PLA

An ordinary function $f(x_1, x_2, \ldots, x_n)$: B^n-> B, B = $\{0, 1\}$, can be represented by a Boolean expression of two-valued variables x_i $(i = 1,2,\ldots, n)$.

Definition 1) An element of a Bolean algebra B is called a constant on B.

Definition 2) A symbol that may represent any one of the elements of B is called a (Boolean) variable on B.

Definition 3) There are $2\,2^n$ different mappings which map from the Cartesian Product set $\{0, 1\}^n$ into $\{0, 1\}$.

Definition 4) A literal is defined to be a constant, variable or complemented variable.

Definition 5) Let $a = (a_1, a_2, \ldots, a_n)$ be a constant in B^n.

$$f(x_1, x_2, \ldots, x_n) = f(a_1, a_2, \ldots, a_n)$$

$$x_1^{a_1}, x_2^{a_2} \dots x_n^{a_n}$$

where $x_i^{a_i} = x_i$ if $a_i = 1$

$\bar{x}_i$ otherwise

is called the canonical sum-of-product form.

B. First method to implement three-level PLA of Boolean function

First of all, two theorems necessary to implement first method are derived as follows;

Theorem 1] Any AND-OR minterms can be converted into EX-OR-AND-OR minterms;

$$f(x_1, x_2, \dots, x_n, +, \cdot) = f(x_1, x_2, \dots, x_n, +, \cdot, \oplus)$$

Proof. Let a switching function be $f(x_1, x_2, \dots, x_n)$, first n variables are divided into two parts such that $f((x_1), (x_2, x_3, \dots, x_n)) = f(p_1, P_2)$
Thus $f(P_1, P_2) = f(r_1, r_2) P_1^{r_1} . P_2^{r_2}$ where (r_1, r_2) is element of P_1 x P_2 and x denotes Cartesian Product.

The term $P_1^{r_1} \cdot P_2^{r_2}$ can be replaced by

$$P_1^{r_1} (P_1^{r_1} \cdot P_2^{r_2} + P_1^{\bar{r}_1} . P_2^{\bar{r}_2}) =$$

$$P_1^{r_1} (P_1 \oplus P_2)^{r_1 \oplus r_2}$$

Applying this rule to overall variables, switching function can be expressed as EX-OR-AND-OR forms without loss of generality.

EXAMPLE 1. If a switching function is $f(x_1, x_2) = x_1 \bar{x}_2$, it can be represented
$f(x_1, x_2) = (1, 0)\ x_1^1 \cdot x_2^0$ by theorem 1).
From here,

$$x_1^1 . x_2^0 = x_1^1 (x_1^1 . x_2^0 + x_1^0 . x_2^1)$$

$$= x_1^1 (x_1 \oplus x_2)^{1 \oplus 0}$$

$$= x_1^1 (x_1 \oplus x_2)^1$$

Theorem 2] Any two minterms can be combined and reduced to one term.

Proof. Let $X = (a_1, a_2, \dots, a_n)$ and $Y = (b_1, b_2, \dots, b_n)$ be the binary representation of two terms of f.
Two terms such that $X_1^{a_1} X_2^{a_2} \dots X_n^{a_n}$

and $X_1^{b_1} X_2^{b_2} \dots X_n^{b_n}$

be supposed to be identical term except for two bit position i, j ($1 \leq i$ or $j \leq n$).
And then two terms are reduced to one term by using EX-OR and EX-NOR operation.

$$X_1^{a_1} X_2^{a_2} \dots X_i^{a_i} \dots X_j^{a_j} \dots X_n^{a_n}$$

$$+ X_1^{b_1} X_2^{b_2} \dots X_i^{b_i} \dots X_j^{b_j} \dots X_n^{b_n}$$

$$= X_1^{a_1} X_2^{a_2} \dots X_n^{a_n} (X_i^{a_i} X_j^{a_j} + X_i^{b_i} X_j^{b_j})$$

$$= X_1^{a_1} X_2^{a_2} \dots X_n^{a_n} (X_i \oplus X_j)^{a_i \oplus a_j}$$

By theorem 1 and 2, it is proved that any Boolean function expressed by sum-of-products can be converted into EX-OR-AND-OR three-level form.

EXAMPLE 2 Suppose that $X = x_1 x_2 x_3 x_4 \bar{x}_5 x_6$

$$Y = x_1 x_2 \bar{x}_3 x_4 x_5 x_6.$$

$$X + Y = x_1 x_2 x_3 x_4 \bar{x}_5 x_6 + x_1 x_2 \bar{x}_3 x_4 x_5 x_6$$

$$= x_1 x_2 x_4 x_6 (x_3 \bar{x}_5 + \bar{x}_3 x_5)$$

$$= x_1 x_2 x_4 x_6 (x_3 \oplus x_5)$$

by theorem 2), any two-minterms can be reduced one ex-or or ex-nor form.

EXAMPLE 3. In case of realizing three-level PLA from two-level PLA, suppose that a AND array part of two-level PLA is constructed the form shown in Fig. 6. To minimize the AND array part, theorem 2 can be applied to Fig. 6. Product terms (= AND array part of two-level PLA) shown in Fig. 6. can be converted EX-OR-AND terms using first method

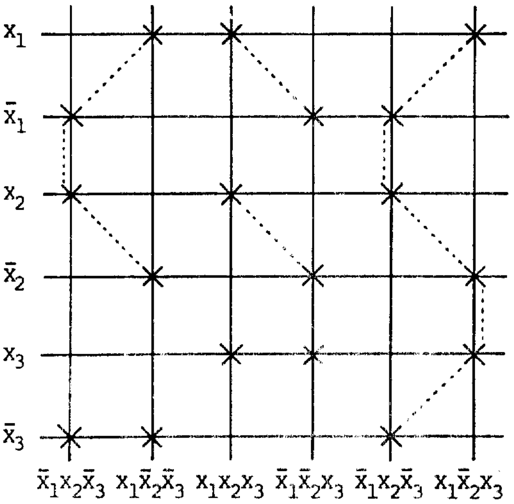

Fig. 6 The AND array of two-evel PLA

1) From 1st and 2nd column; By theorem 2

$$\bar{X}_1 \cdot X_2 \cdot \bar{X}_3 + X_1 \cdot \bar{X}_2 \cdot \bar{X}_3 = (X_1 \oplus X_2)\ \bar{X}_3$$

2) From 3rd and 4th column; By theorem 2

$$X_1\ X_2\ X_3 + \bar{X}_1\ \bar{X}_2\ X_3 = (X_1 \odot X_2)\ X_3$$
$$= (X_1 \oplus \bar{X}_2) X_3 = (\bar{X}_1 \oplus X_2)\ X_3$$

3) From 5th and 6th column; By theorem 2

$$\bar{X}_1\ X_2\ \bar{X}_3 + X_1\ \bar{X}_2\ X_3$$
$$= (X_1 \oplus X_2)\ (X_2 \oplus X_3)$$

Fig. 6. can be converted as Fig. 7.

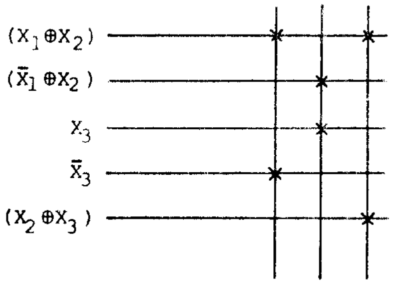

Fig. 7. The EX-OR and AND array of three-level PLA

EXAMPLE 4. In the Fig. 8, 9, 10, and 11, example which is the case of changing binary code into gray code, is shown. (applying method)

	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	1
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Fig. 8. Truth-table for Changing binary code into gray code

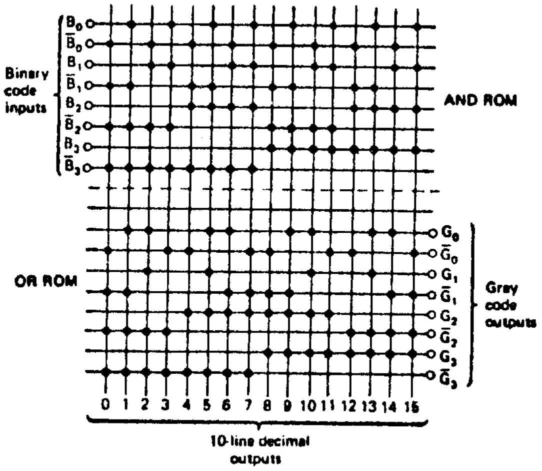

Fig. 9. The canonical sum-of-products for changing binary code into gray code

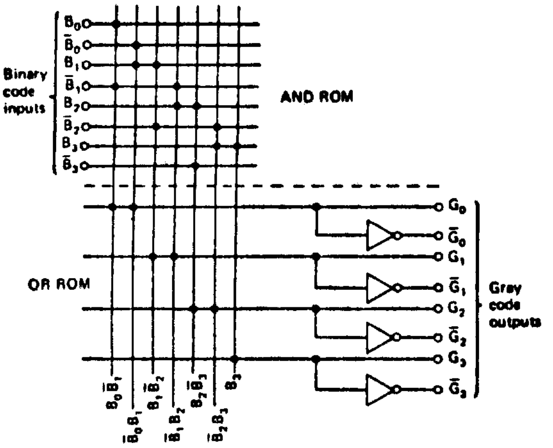

Fig. 10. The standard sum-of-products for two-level PLA

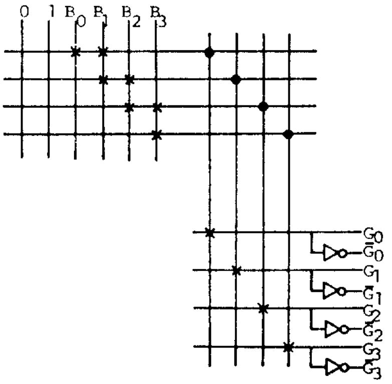

Fig. 11. The Binary-to-Gray code converter using EXANOR PLA

Fig. 8. is for truth table, fig. 9. for canonical sum-of-products form, fig. 10. for standard sum-of-products form, and fig. 11. for three-level PLA.

C. Second method to implement three-level PLA of Boolean Function

In this case, MASK [9] method, which is computer-aided method, is used. Using MASK method, two MASK tables can be made. One is made using given variables, and another is using new varables which consists of EX-OR forms. From two MASK tables, a new MASK table, which is called modified MASK table, is made and PI's are selected with optimal cost in that table.

Theorems necessary to implement second method are derived as follows;

Theroem 3] Expressions of switching functions for three-level PLA by the linked variables;

$$f(x_1, x_2, \ldots, x_n) = \Sigma f(a_1, a_2, \ldots, a_n)\, x_1^{a_1} (x_1 + x_2)^{a_1 \oplus a_2} \ldots (x_{n-2} + x_{n-1})^{a_{n-2} \oplus a_{n-1}} (x_{n-1} + x_n)^{a_{n-1} \oplus a_n}$$

Proof. Applying theorem 1] to the each two variables, theorem 3] can be proved.

Theorem 4] Expressions of switching functions for three-level PLA by the partitioned variables;

$$f(x_1, x_2, \ldots, x_n) = \Sigma f(a_1, a_2, \ldots, a_n)\, x_1^{a_1} x_3^{a_3} \ldots x_{n-1}^{a_{n-1}} (x_1 \oplus x_2)^{a_1 \oplus a_2} (x_3 \oplus x_4)^{a_3 \oplus a_4} \ldots (x_{n-1} \oplus x_n)^{a_{n-1} \odot a_n}$$

Proof. Applying theorem 1] to the partitioned variables.

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \sum f(a_1, a_2, \ldots, a_n)\, x_1^{a_1} x_1^{a_2} \ldots x_n^{a_n} \\ &= \sum f(a_1, a_2, \ldots, a_n)\, x_1^{a_1} (x_1 \oplus x_2)^{a_1 \oplus a_2} x_3^{a_3} (x_3 \oplus x_4)^{a_3 \oplus a_4} \ldots x_{n-1}^{a_{n-1}} (x_{n-1} + x_n)^{a_{n-1} + a_n} \\ &= \sum f(a_1, a_2, \ldots, a_n)\, x_1^{a_1} x_3^{a_3} \ldots x_{n-1}^{a_{n-1}} \end{aligned}$$

$$(x_1 \oplus x_2)^{a_1 \oplus a_2} \cdot (x_3 \oplus x_4)^{a_3 \oplus a_4} \ldots$$

$$(x_{n-1} \oplus x_n)^{a_{n-1} \oplus a_n}$$

EXAMPLE 5. Of theorem 4], the term $x_1\bar{x}_2\bar{x}_3x_4$, can be expressed as EX-OR-AND

form $f(x_1, x_2, x_3, x_4) = x_1\bar{x}_2\bar{x}_3x_4$

$$= x_1(x_1 \oplus x_2)\bar{x}_3(x_3 \oplus x_4)$$

$$= x_1\bar{x}_3(X_1 \oplus x_2)(x_3 \oplus x_4)$$

Theorem 5] Terms such as x_1, $(x_1 \oplus x_2)$, . . ., $(x_{n-1} \oplus x_n)$ can be substituted for newly generated variables; s_1, s_2, . . ., s_n. So, the original truth table is converted to the new truth table of n variables.

The following two examples are taken for application of the second method.

EXAMPLE 6. Suppose that any Karnaugh map of Boolean function is given in Fig. 12. a). It can be converted using second method. From Fig. 12. a), if two-level PLA is used, there are the following 16 prime implicants;

$x_1x_2x_3x_4x_5x_6$, $x_1x_1x_3x_4x_5x_6$,

$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5\bar{x}_6$, $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4x_5x_6$, $\bar{x}_1\bar{x}_2\bar{x}_3x_4\bar{x}_5x_6$,

$\bar{x}_1x_2x_3\bar{x}_4\bar{x}_5\bar{x}_6$, $\bar{x}_1x_2x_3\bar{x}_4x_5x_6$, $\bar{x}_1x_2x_3x_4\bar{x}_5x_6$,

$x_1\bar{x}_2x_3\bar{x}_4\bar{x}_5\bar{x}_6$, $x_1\bar{x}_2x_3x_4\bar{x}_5x_6$, $x_1\bar{x}_2x_3x_4x_5\bar{x}_6$,

$x_1x_2\bar{x}_3\bar{x}_4\bar{x}_5\bar{x}_6$, $x_1x_2\bar{x}_3\bar{x}_4x_5x_6$, $x_1x_2\bar{x}_3x_4\bar{x}_5x_6$,

$\bar{x}_1\bar{x}_2\bar{x}_3x_4x_5\bar{x}_6$,

$\bar{x}_1x_2x_3x_4x_5\bar{x}_6$,

$x_1\bar{x}_2x_3\bar{x}_4x_5x_6$,

$x_1x_2\bar{x}_3x_4x_5\bar{x}_6$,

And by the theorem 3],

$$s_1s_2s_3s_4s_5s_6 = x_1\ x_2\ (x_2 \oplus x_3)x_4\ x_5\ (x_5 \oplus x_6)$$

Then new varables such as s_1,s_2,s_3, s_4, are assigned to each position and new truth table is generated as shown in Fig. 12.b).

From Fig. 12.b), the following 4 prime implicants are obtained;

$\bar{s}_1\bar{s}_3\bar{s}_4\bar{s}_6$, $\bar{s}_1\ \bar{s}_3s_4s_6$, $s_1s_3\bar{s}_4\bar{s}_6$, $s_1s_3s_4s_6$

$x_1x_2x_3$ \ $x_4x_5x_6$	000	001	011	010	110	111	101	100
000	1		1		1		1	
001								
011	1		1		1		1	
010								
110	1		1		1		1	
111								
101	1		1		1		1	
100								

a) for variables xi

$s_1s_2s_3$ \ $s_4s_5s_6$	000	001	011	010	110	111	101	100
000	1			1		1	1	
001								
011								
010	1			1		1	1	
110								
111	1			1		1	1	
101	1			1		1	1	
100								

b) for variables si

Fig. 12. The Karnaugh maps for example 4

EXAMPLE 7. Suppose that any Karnaugh map of Boolean function is given in Fig. 13. a). It can be converted using second method. From Fig. 13. a), there are 15 prime implicant;

$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5$, $\bar{x}_1\bar{x}_2\bar{x}_4x_6$, $\bar{x}_1\bar{x}_2\bar{x}_3x_4x_5\bar{x}_6$,

$\bar{x}_1x_2x_4x_6$, $x_2x_3\bar{x}_4\bar{x}_5\bar{x}_6$, $\bar{x}_1x_2x_3x_4x_5$,

$x_1\bar{x}_2x_4x_6$, $x_1x_3\bar{x}_4\bar{x}_5\bar{x}_6$ $x_1\bar{x}_2x_3x_4x_5$,

$x_1x_2\bar{x}_4\bar{x}_5$, $x_1x_2\bar{x}_3\bar{x}_4x_6$, $x_1x_2\bar{x}_3x_4x_5\bar{x}_6$

$\bar{x}_3x_4\bar{x}_5x_6$, $\bar{x}_2x_3\bar{x}_4x_5x_6$, $\bar{x}_1x_3\bar{x}_4x_5x_6$

From Fig. 13. b), following 11 prime implicants are obtained;

$\bar{s}_1\bar{s}_3s_4s_6$, $\bar{s}_1\bar{s}_2\bar{s}_4\bar{s}_5s_6$, $\bar{s}_1s_2s_4s_5\bar{s}_6$,

$s_1\bar{s}_2s_4s_5\bar{s}_6$, $s_1s_3\bar{s}_4\bar{s}_6$, $s_1s_2\bar{s}_4\bar{s}_5$,

$s_1s_3s_4s_6$, $\bar{s}_1\bar{s}_3\bar{s}_4\bar{s}_6$, $\bar{s}_1\bar{s}_2\bar{s}_4s_5\bar{s}_6$,

$\bar{s}_2\bar{s}_3s_4\bar{s}_5s_6$, $\bar{s}_1s_2s_4\bar{s}_5s_6$

$x_1x_2x_3$ \ $x_4x_5x_6$	000	001	011	010	110	111	101	100
000	1	1	1		1		1	
001		1	1					
011	1		1		1	1	1	
010						1	1	
110	1	1	1		1		1	
111	1	1						
101	1		1		1	1	1	
100						1	1	

a) for variables xi

$s_1s_2s_3$ \ $s_4s_5s_6$	000	001	011	010	110	111	101	100
000	1	1		1		1	1	
001		1		1				
011					1		1	
010	1			1		1	1	
110	1	1						
111	1	1		1		1	1	
101	1			1	1	1	1	
100					1		1	

b) for variables si

Fig. 13. The Karnaugh maps for example 5

4. Implementation of three-level PLA circuits

In section 3, two design methods are discussed for three-level PLA. The first method which use the direct conversion approach to generate the EXANOR PLA is simple but tedious. And the second method which treats the switching function more genral in all the design phase, is more applicable to practical implementations. Thus, in this section three-level PLA circuit design is realized based on the second method.

Before discussing the procedure, the terminologies are defined;

Definition 6) The prime implicant generated from original switching function, whose variables are x_i (i = 1,2,...,n), is defined as XPI_i, where i designates the order of genrating PI's.

Definition 7) The prime implicant generated from new switching function, whose variables are determined by the partitioned variables s_i (i = 1,2,..., n), is defined as SPI_i, where i denotes the order of generating PI's.

Definition 8) The table which consists of XPI_i and SPI_i is called the mixed PI table.

Next, procedure for implementation of three-level PLA circuit is as follows;

Procedure:

1) Generate XPI_i whose variables are x_i (i = 1,2,..., n) using MASK method.
2) Generate SPI_i whose variables are s_i (i = 1,2,...,n) using MASK method.
3) Make the mixed PI table.
4) From the mixed PI table, count the number of the checks in each row and let it be the original cost.
5) Select the PI which has the maximum value of cost. If PI is more than one, then PI in SPI has priority.
6) Arrange the new cost, and if all the minterms are covered, go to step 7). Otherwise, go to step 5).
7) Stop.

Example 8. In example 6, choose the PI's using the procedure.
By the procedure, mixed PI table is made. And from the mixed PI table, 4 PI's are selected as follows;

$\bar{s}_1\bar{s}_3\bar{s}_4\bar{s}_6$,　$\bar{s}_1\bar{s}_3 s_4 s_6$,　$s_1 s_3\bar{s}_4\bar{s}_6$,
$s_1 s_3 s_4 s_6$

Cost: the number of minterms covered by the prime implicant.

Minterm XPI or SPI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Cost	Selected PI's
XPI_1	x																1	
XPI_2		x															1	
XPI_3			x														1	
XPI_4				x													1	
XPI_5					x												1	
XPI_6						x											1	
XPI_7							x										1	
XPI_8								x									1	
XPI_9									x								1	
XPI_{10}										x							1	
XPI_{11}											x						1	
XPI_{12}												x					1	
XPI_{13}													x				1	
XPI_{14}														x			1	
XPI_{15}															x		1	
XPI_{16}																x	1	
SPI_1	x	x			x	x											4	+
SPI_2			x	x			x	x									4	+
SPI_3									x	x			x	x			4	+
SPI_4											x	x			x	x	4	+

Fig. 14. The mixed P1 table for example 8

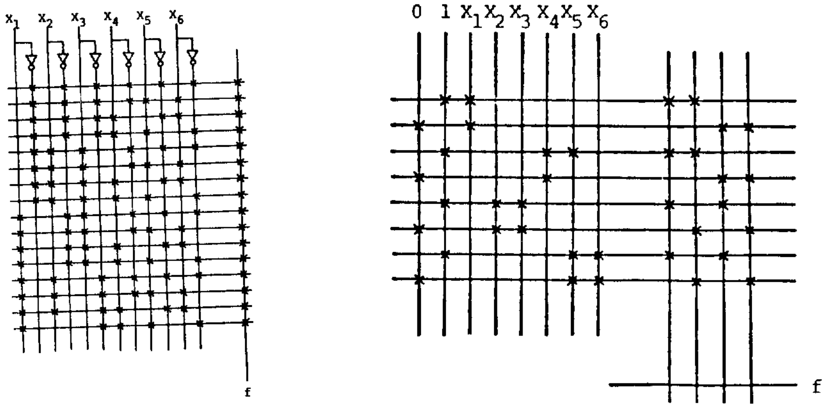
x_1 x_2 x_3 x_4 x_5 x_6
f
0 1 x_1 x_2 x_3 x_4 x_5 x_6
f

Fig. 14-1. Circuit for example 6 and example 8

Example 9. In example 7, choose the PI's using the procedure. By the procedure, mixed PI table is made. And from the mixed PI table, 8 PI's are selected as follows: $x_1x_2x_4x_5$, $\bar{x}_1x_2x_4x_5$, $x_1\bar{x}_2x_4x_6$, $\bar{S}_1\bar{S}_3S_4S_6$, $S_1S_3\bar{S}_4\bar{S}_6$, $S_1S_2\bar{S}_4\bar{S}_5$, $S_1S_3S_4S_6$, $\bar{S}_1\bar{S}_3\bar{S}_4\bar{S}_6$,

Minterm XPI or SPI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	Cost	Selected PI's
XPI_1	X	X																											2	
XPI_2		X	X			X	X																						4	+
XPI_3				X																									1	
XPI_4								X	X			X		X															4	+
XPI_5										X																	X		1	
XPI_6													X	X															2	
XPI_7															X	X			X		X								4	+
XPI_8																	X										X		2	
XPI_9																				X	X								2	
XPI_{10}																						X	X				X	X	4	
XPI_{11}																							X	X					2	
XPI_{12}																										X			1	
XPI_{13}			X				X								X										X				4	
XPI_{14}						X												X											2	
XPI_{15}						X					X																		2	

Minterm XPI or SPI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	Cost	Selected PI's
SPI_1				X*	X*							X*	X*																4	
SPI_2		X*				X*																							2	
SPI_3									X*					X*															2	
SPI_4																X*					X*								2	
SPI_5																	X*	X*				X*		X*					4	
SPI_6																						X*	X*				X*	X*	4	
SPI_7																			X*	X*					X*	X*			4	
SPI_8	X*		X*							X*	X*																		4	
SPI_9			X*				X*																						2	
SPI_{10}				X*											X*														2	
SPI_{11}									X*			X*																	2	

Fig. 15. The mixed P1 table for example 9

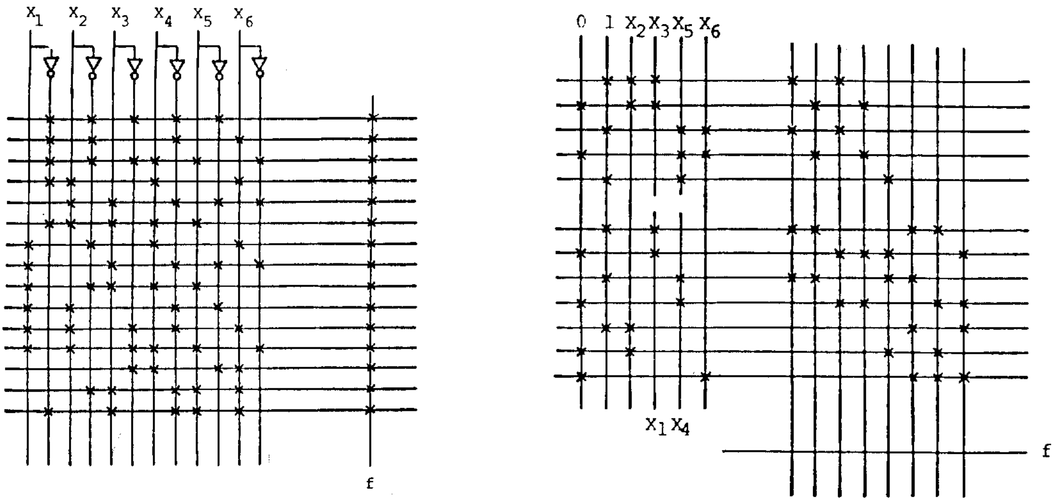

Fig. 15-1. Circuit for example 7 and example 9

5. Size Comprison with Two-Level PLA and Three-Level PLA

In this section, comparison with two-level PLA size and three-level PLA size is discussed. In the case of physical design, MOS array size plays an important role in estimating overall chip size.

In Fig. 16. and Fig. 17, their sizes S(n) are evaluated as follows;

Two-level PLA: $S_2(n) = (2n + m)W$

Three-level PLA: $S_3(n) = (n + 2 + W)\,H + Wm$

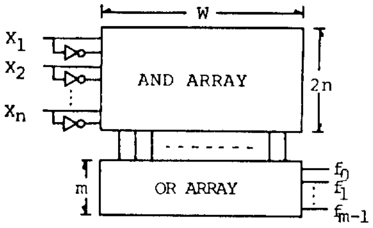

Fig. 16. The dimension of two -levelPLA

Without loss of generality, output function is regarded as simple output switching function.

Theorem 6] In two-level PLA, W is 2^{n-1}, and in three-level PLA, W is 2^{n-2}. The W sizes of two-level and three-level PLA are 2^{n-1} and 2^{n-2}, respectively.

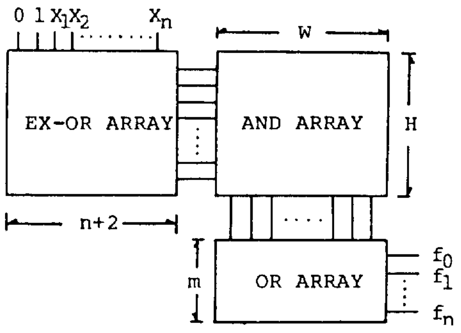

Fig. 17. The dimension of three-level PLA

Proof. In two-level PLA, if the number of input is n, then in worst case, W is 2^n· But the number of columns are $2^n/2$ because of duality of switching function.

And by theorem 2], the number of columns of two-level PLA is divided by 2. So W of three-level PLA leads to $2^{n-1}/2$.

By theorem 6], sizes of two types of PLA are evaluated as follows;

Size of two-level PLA: $S_2(n) = (2n+1)2^{n-1}$

Size of three-level PLA:

$$S_3(n) = (n+2+2^{n-2})\,2n+2^{n-2}$$
$$= (2n+1)2^{n-2}+2n^2+4n$$

And let the ratio of reduction R, be

$$\frac{S_3(n)}{S_2(n)}$$

Ratio of reduction is

$$R = 1 - \frac{(2n+1)2^{n-2}+2n^2+4n}{(2n+1)2^{n-1}} \doteqdot 0.5$$

So, using three-level PLA, a half of size is reduced.

6. Conclusion

In this paper the new design methodology of PLA is considered and evaluated by using PLA chip size. Some of problems encountered in design of PLA are the reduction of input lines and product terms. To reduce the product terms and input lines, the architecture of three-level PLA is proposed in this paper.

This architecture is composed of the three-level PLA such as EX-OR, AND and OR. By theorem 1] - 3], all the switching functions are expressed as EX-OR, AND and OR. And there are two types of expressing the given switching functions. Some examples are introduced for the clarity of theorems, finally, ratio of size reduction is considered and its reduction ratio results in a half time as much as the conventional two-level PLA size.

References

[1] H. Fleisher and L.I. Maissel; "An introduction to array logic", IBM J. Res. Develop., vol. 19, pp. 98-109, Mar. 1975.

[2] S.J. Hong, R.G. Cain, and D.L. Ostapko; "MINI; A heuristic approach for logic minimization", IBM J. Res. Develop. vol. 18, pp. 443-458, Sept. 1974.

[3] R.A. Wood; "High-speed dynamic programmable logic array chip," IBM J. Res. Develop., vol. 19, pp. 379-383, July 1975.

[4] TSUTOMU SASAO; "Multiple-valued decomposition of generalized boolean functions and the complexity of programmable logic arrays," IEEE trans. com., vol. c-30, pp. 635-643, Sept. 1981.

[5] YAHITO KAMBAYASHI; "Logic design of programmable logic arrays," IEEE trans. com., vol. c-28, pp. 609-617, Sept. 1979.

[6] G. Pomper and J.R. Armstrong; "Representation of multivalued functions using the direct cover method," IEEE trans. com. vol c-30, pp. 674-679, Sept. 1981.

[7] Sahni and Horowitz; "Fundamentals of computer algorithms" Computer Science Press, Inc.

[8] THADDEUS KOBYLARZ and ATEF AL-NAJJAR; "An examination of the cost function for programmable Logic Arrays," IEEE Trans. Com. vol. c-28, No. 8, pp. 586-590, Aug. 1979.

[9] Cho Dong Sub and Hwang Hee Yeung; "On the minimization of the switching function by the MASK method," KIEE, vol. 28, No. 11, pp. 801-808, Nov. 1979.

[10] Hwang Hee Yeung; "A new approach to the minimization of switching functions by the simple table method, "KIEE, vol. 28, No. 6, pp. 61-77, June 1979