AN APPLICATION OF MULTIPLE-VALUED LOGIC TO A DESIGN OF PROGRAMMABLE LOGIC ARRAYS

Tsutomu Sasao

Department of Electronic Engineering
Osaka University, Osaka 565, Japan

Abstract: A three-level programmable logic array (three-level PLA) consists of three main parts, the D array, the AND array, and the OR array, and each of these arrays can be programmed. In this paper, a design method for three-level PLA's is described. Main results obtained are 1) The minimization of the AND array corresponds to the minimization of a multiple-valued input two-valued output logic function; 2) By using the theory of multiple-valued decomposition of two-valued function, the computation time and the memory requirement for the minimization of the AND array can be reduced; and 3) The design of multiple-output function can be done in a similar way by introducing a variable which denotes the outputs.

I. Introduction

In the development of new integrated circuits, high cost as well as excessive lead time have long been recognized as serious problems by both the manufacturers and users of semiconductor devices. One approach to solve these problems that has appears commercially over the years involves customing only the interconnection pattern of standard prediffused array of logic gates. The other approach is now generally known as programmable logic arrays[1]-[5].

In this paper, a design method for three-level PLA's is described. The three-level PLA consists of three main parts, the D array, the AND array, and the OR array as shown in Fig.1.1, and each of these arrays can be programmed. For example, a six-input three-output function can be realized by the three-level PLA shown in Fig.1.2. In the D array, the horizontal lines are distributed OR gates with inputs represented as X's at selected line crossing. In the AND array, the dots are analogous to AND gate inputs where the gate is represented by the vertical line. In the OR array, the X's denote the OR gate inputs where the gate is reprsented by the horizontal line.

Three-level PLA's have several advantages to the conventional two-level PLA's[5].

(1) In order to realize an arbitrary function of n-variables, the array size of $\underline{O}(2^n)$ is sufficient in a three-level PLA realization while $\underline{O}(n2^n)$ is necessary in a conventional two-level PLA realization.

(2) Array size can be further reduced by utilizing the partial symmetry, the decomposability, and the redundancy of the given function.

Major disadvantages of three-level PLA's are as follows:

(3) Three-level PLA's are slower than two-level PLA's.

(4) For small n, three-level PLA's sometimes require larger arrays than two-level PLA's.

In Section II, a design method for three-level PLA's which is obtained in [5] will be discussed. And it will be shown that the minimization of multiple-valued input two-valued output function corresponds to the minimization of the AND array. In Section III, the theory of multiple-valued decomposition of two-valued function will be introduced. And it will be shown that the computation time and memory requirement for the minimization of the AND array can be reduced by using the theory. In Section IV, a design method for multiple-output functions will be discussed.

II. Three-level Programmable Logic Arrays.

In this section, a design method which minimizes the size of a three-level PLA will be considered. As shown in Fig.1.1, the three-level PLA can realize an arbitrary OR-AND-OR circuits. Several works are known about OR-AND-OR circuits minimization[6]-[7], but these methods need too much computation even if the number of input variables is small. To avoid this difficulty, the design of three-level PLA is divided into two parts. The first part is the design of the D array, and the second part is the design of the AND array. It will be shown that in order to minimize the size of the AND array for a given function, it is sufficient to obtain a minimal sum-of-products expression for the corresponding multiple -valued input two-valued output function.

Definition 2.1: A three-level PLA consists of the D array, the AND array, and the OR array as shown in Fig.1.1. The size of n-variable m-output three-level PLA is defined as $C(n)=(2n+W)H+Wm$, where W is the number of columns of the AND array, H is the number of rows of the AND array, and m is the number of rows of the OR array.

Definition 2.2: Let $X=(x_1,x_2,\ldots,x_n)$ be a vaiable in $B^n=\{0,1\}^n$. The set of variables in X is denoted by $\{x_1,x_2,\ldots,x_n\}$ or by $\{X\}$. The number of the variables in $\{X\}$ is denoted by $d(X)$. $(X_1,X_2,\ldots,X_r)$ is said to be a partition of X iff $\{X_1\}\cup\{X_2\}\cup\ldots\cup\{X_r\}=\{X\}$, $\{X_i\}\cap\{X_j\}=\phi$ $(i\neq j)$, and $\{X_i\}\neq\phi$.

Definition 2.3: Let $\underline{a}=(a_1,a_2,\ldots,a_n)$ be a constant in B^n. $X^{\underline{a}}$: $B^n\rightarrow B$ is a function such that $X^{\underline{a}}=0$ if $X\neq\underline{a}$ and $X^{\underline{a}}=1$ if $X=\underline{a}$. Let $S\subseteq B^n$, X^S is defined as $X^S=\bigvee_{\underline{a}_i\in S} X^{\underline{a}_i}$.

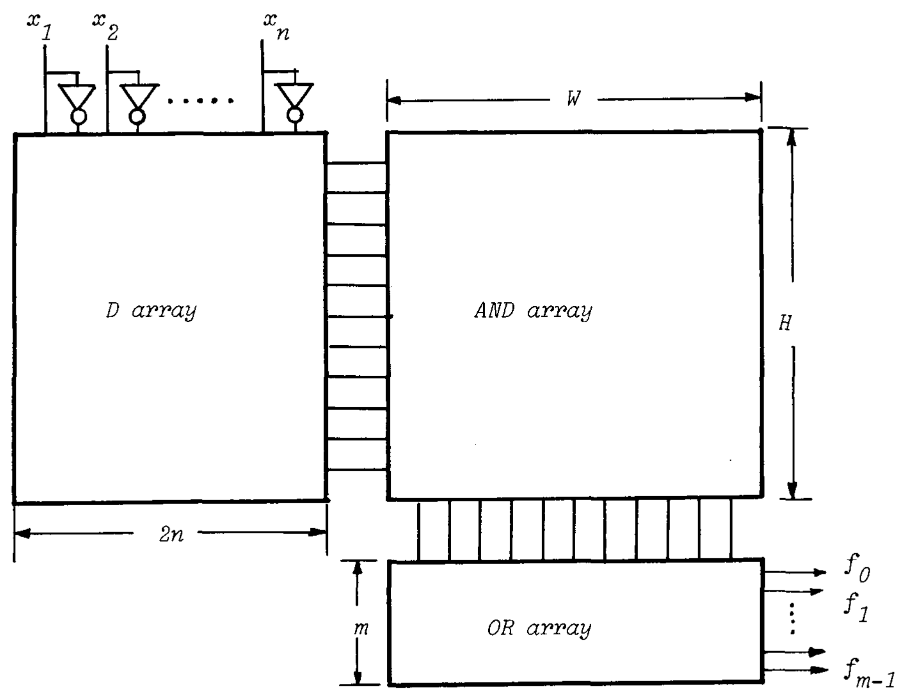

Fig. 1.1 Three-level PLA

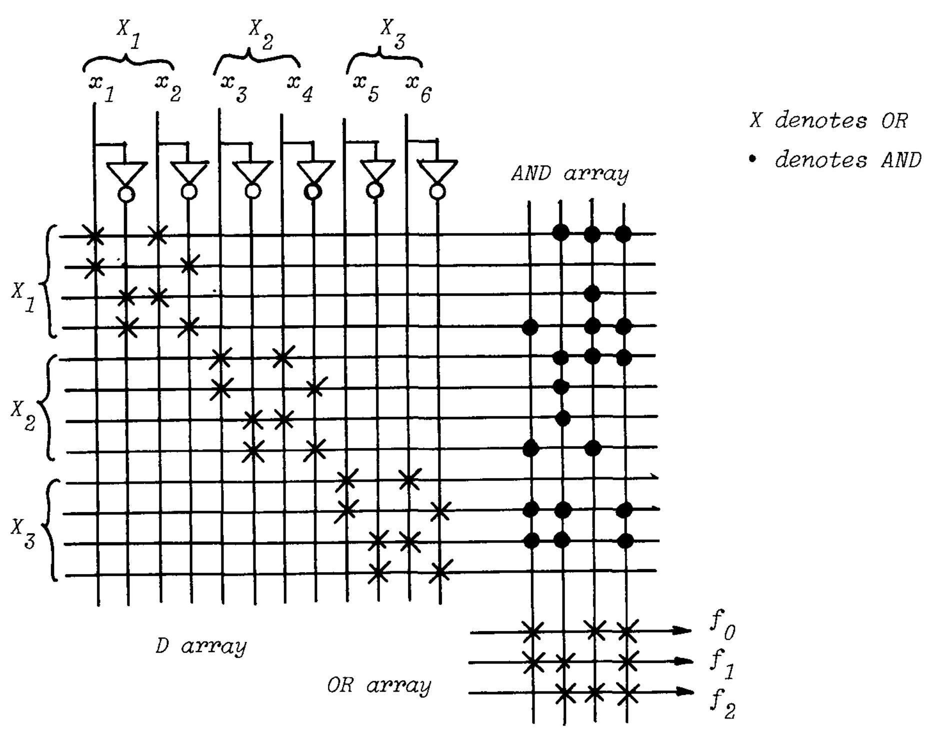

Fig.1.2 An example of three-level PLA

Lemma 2.1: Let $(X_1, X_2, \ldots, X_r)$ be a partition of X. An arbitrary function f(X) of n variable is expressed in the form

$$\bigvee_{(\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_r)} f(\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_r) X_1^{\underline{a}_1} \cdot X_2^{\underline{a}_2} \cdot \ldots \cdot X_r^{\underline{a}_r},$$

where $\underline{a}_i \in B^{n_i}$, $d(X)=n$, and $d(X_i)=n_i$.

Example 2.1: Consider the four-variable function

$f(X)=x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4 \vee \bar{x}_1x_2x_3x_4$.

Let (X_1, X_2) be a partition of $X=(x_1,x_2,x_3,x_4)$ and let $X_1=(x_1,x_2)$, $X_2=(x_3,x_4)$. f(X) can be represented in the form

$f(X)=X_1^{(11)} \cdot X_2^{(00)} \vee X_1^{(11)} \cdot X_2^{(01)} \vee X_1^{(10)} \cdot X_2^{(11)}$
$\vee X_1^{(01)} \cdot X_2^{(11)}$.

Lemma 2.2: Let $d(X)=n$ and $S_1, S_2 \subseteq B^n = I$.

$X^{S_1} \cdot X^{S_2} = X^{S_1 \cap S_2}$, $X^{S_1} \vee X^{S_2} = X^{S_1 \cup S_2}$,
$\overline{X^{S_1}} = X^{I-S_1}$, $X^I=1$, and $X^\phi=0$.

Definition 2.4: X^S is said to be a literal. A product of distinct literals is said to be a term. A sum of terms is said to be a sum-of-products expression. The number of terms in a sum-of-products expression P is denoted by t(P). P is said to be minimal if there is no expression Q such that $t(Q)<t(P)$ and that Q denotes the same function as P. Let E_1 and E_2 be terms. E_2 is subterm of E_1 iff $E_1 \neq E_2$ and $E_1 \leq E_2$. E_1 is said to be a prime implicant of f if $E_1 \leq f$ and if there is no subterm E_2 of E_1 such that $E_2 \leq f$.

Lemma 2.3: Let $(X_1, X_2, \ldots, X_r)$ be a partition of X. An arbitrary function f(X) can be represented in a form

$$f(X)=\bigvee_{(S_1,S_2,\ldots,S_r)} X_1^{S_1} \cdot X_2^{S_2} \cdot \ldots \cdot X_r^{S_r}, \text{ ---- } (2.1)$$

where $S_i \subseteq B^{n_i}$, and $d(X_i)=n_i$. In a three-level PLA, if the D array generates all the maxterms of $\{X_i\}$ for $i=1,2,\ldots,r$, then an arbitrary term which has the form $X_1^{S_1} \cdot X_2^{S_2} \cdot \ldots \cdot X_r^{S_r}$ can be realized in each column of the AND array.

Example 2.2: The function of Example 2.1 can be represented as

$f(X)=X_1^{(11)} \cdot X_2^{\{(00),(01)\}} \vee X_1^{\{(10),(01)\}} \cdot X_2^{(11)}$.

Theorem 2.1: Let $(X_1, X_2, \ldots, X_r)$ be a partition of X, and let the D array generate all the maxterms of $\{X_i\}$ for $i=1,2,\ldots,r$. In order to minimize the size of the AND array for f(X), it is sufficient to obtain a minimal sum-of-products expression of f(X) having the form

$$f(X_1,X_2,\ldots,X_r)=\bigvee_{(S_1,S_2,\ldots,S_r)} X_1^{S_1} \cdot X_2^{S_2} \cdot \ldots \cdot X_r^{S_r}.$$

If P is minimal expression for f(X), then $t(P) \leq 2^{n-\max_i\{n_i\}}$, where $n_i=d(X_i)$.

Proof: By Lemma 2.3, we have the first part. Assume without loss of generality that $n_1=\max_i\{n_i\}$. f(X) can be represented as

$$f(X_1,X_2,\ldots,X_r)=\bigvee_{(S_1,\underline{a}_2,\underline{a}_3,\ldots,\underline{a}_r)} X_1^{S_1} \cdot X_2^{\underline{a}_2} \cdot X_3^{\underline{a}_3} \cdot \ldots \cdot X_r^{\underline{a}_r} \text{ ------(2.2)}$$

The number of terms in (2.2) is at most $\prod_{i=2}^{r} 2^{n_i} = 2^{n-n_1} = 2^{n-\max_i\{n_i\}}$. Q.E.D.

Corollary 2.1: The size of three-level PLA which is sufficient to realize an arbitrary function of n variable is $C(n)=(2n+W)H+W$, where $W=2^{n-\max_i\{n_i\}}$, $H=\sum_{i=1}^{r} 2^{n_i}$ and $\underline{n}=(n_1,n_2,\ldots,n_r)$ is a vector which represents the partition of input variables.

III. Multiple-Valued Decomposition of Two-Valued Logic Function.

In this section, the theory of muliple-valued decomposition of two-valued function will be described. By using this theory, we can reduce the computation time and the memory reqirement for the minimization of the AND arrays.

Definition 3.1: Let $(X_1, X_2, \ldots, X_r)$ be a partition of X, and f(X) be a function such that

$f: B^{n_1} \times B^{n_2} \times \ldots \times B^{n_r} \rightarrow B.$

For $\underline{a}$, $\underline{b} \in B^{n_i}$, define a relation

$\underline{a} \overset{i}{\sim} \underline{b} \iff f(X|\underline{a}\rightarrow X_i)=f(X|\underline{b}\rightarrow X_i)$,

where $f(X|\underline{a}\rightarrow X_i)$ denotes $f(X_1,X_2,\ldots,X_{i-1},\underline{a},X_{i+1},\ldots,X_r)$. Obviously, the relation $\overset{i}{\sim}$ is an equivalence relation. Let $\Pi_i=(L_0^i,L_1^i,\ldots,L_{k_i-1}^i)$ be a partition of B^{n_i} induced by the equivalence relation $\overset{i}{\sim}$. A function $\psi_i: B^{n_i} \rightarrow M_i$; $M_i=\{0,1,\ldots,k_i-1\}$ such that $\psi_i(\underline{a})=j \iff \underline{a} \in L_j^i$ is called a partition function of B^{n_i}.

Example 3.1: Consider a six-variable function

$f(X)=(\bar{x}_1 \vee \bar{x}_2) \cdot (x_3 \oplus x_4) \cdot (\bar{x}_5 \vee \bar{x}_6)$
$\vee (x_1 \vee x_2) \cdot (X_3 \oplus \bar{x}_4) x_5 \vee (x_1 \oplus x_2) \cdot (x_5 \oplus x_6)$.

Let (X_1, X_2, X_3) be a partition of X, where $X_1=(x_1,x_2)$, $X_2=(x_3,x_4)$, and $X_3=(x_5,x_6)$. Note that

$f(X|(01)\rightarrow X_1)=f(X|(10)\rightarrow X_1)$,
$f(X|(00)\rightarrow X_2)=f(X|(11)\rightarrow X_2)$, and
$f(X|(10)\rightarrow X_2)=f(X|(01)\rightarrow X_2)$.

The partition functions of B^{n_i} are shown in Table 3.1.

Table 3.1 Partition functions

X_i	$\psi_1(X_1)$	$\psi_2(X_2)$	$\psi_3(X_3)$
00	0	0	0
01	1	1	1
10	1	1	2
11	2	0	3

Lemma 3.1: Let $(X_1,X_2,\ldots,X_r)$ be a partition of X, $d(X_i)=n_i$, and let ψ_i be a partition function of B^{n_i}. There exists a multiple-valued input two-valued output function

$g: M_1 \times M_2 \times \ldots \times M_r \to B$ such that

$f(X_1,X_2,\ldots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\ldots,\psi_r(X_r))$.

Proof: If $\underline{a}_i \in L^i_{b_i}$ $(i=1,2,\ldots,r)$, then let $g(b_1,b_2,\ldots,b_r)=f(\underline{a}_1,\underline{a}_2,\ldots,\underline{a}_r)$. It is easy to show that this function satisfies the condition. Q.E.D.

This lemma is similar to the well known decomposition theorem of Ashenhurst[8]-[11]. But ψ_i is, in general , a multiple-valued function. When $M_i=\{0,1\}$ $(i=1,2,\ldots,r)$, this lemma reduced to the ordinary decomposition theorem.

Example 3.2: Consider the function of Example 3.1. By Lemma 3.1, $f(X)$ can be represented as $f(X_1,X_2,X_3)=g(\psi_1(X_1),\psi_2(X_2),\psi_3(X_3))$, where $g(Y_1,Y_2,Y_3)$ is shown in Table 3.2.

Table 3.2

Y_1	Y_2	Y_3	g
0	0	0	0
0	0	1	0
0	0	2	0
0	0	3	0
0	1	0	1
0	1	1	1
0	1	2	1
0	1	3	0
1	0	0	0
1	0	1	1
1	0	2	1
1	0	3	1
1	1	0	1
1	1	1	1
1	1	2	1
1	1	3	0
2	0	0	0
2	0	1	0
2	0	2	1
2	0	3	1
2	1	0	0
2	1	1	0
2	1	2	0
2	1	3	0

Definition 3.2: Let $M=\{0,1,\ldots,k-1\}$, $t\in M$, and Y^t: $M\to B$ be a function such that $Y^t=0$ if $Y\neq t$ and $Y^t=1$ if $Y=t$. Let $T\subseteq M$, Y^T is a function such that $Y^T = \bigvee_{t\in T} Y^t$

Lemma 3.2: Let T_1, $T_2 \subseteq M=I$.

$Y^{T_1}\cdot Y^{T_2}=Y^{T_1\cap T_2}$, $Y^{T_1}\vee Y^{T_2}=Y^{T_1\cup T_2}$,

$\overline{Y^{T_1}}=Y^{I-T_1}$, $Y^I=1$, and $Y^\phi=0$.

Lemma 3.3: A multiple-valued input two-valued output function

$g: M_1 \times M_2 \times \ldots \times M_r \to B$

can be represented in the form

$$g(Y_1,Y_2,\ldots,Y_r)=\bigvee_{(t_1,t_2,\ldots,t_r)} g(t_1,t_2,\ldots,t_r)\, Y_1^{t_1}\cdot Y_2^{t_2}\cdot\ldots\cdot Y_r^{t_r} \quad \text{-----(3.1)}$$

or in a form

$$g(Y_1,Y_2,\ldots,Y_r)=\bigvee_{(T_1,T_2,\ldots,T_r)} Y_1^{T_1}\cdot Y_2^{T_2}\cdot\ldots\cdot Y_r^{T_r}, \quad \text{-----(3.2)}$$

where $t_i\in T_i$, $T_i\subseteq M_i$, and $M_i=\{0,1,\ldots,k_i-1\}$.

Proof: By Definition 3.2, it is easy to show that (3.1) holds. By Lemma 3.2 and (3.1), we have (3.2). Q.E.D.

Example 3.3: The function g of Example 3.2 can be represented in the form

$$g(Y_1,Y_2,Y_3)=Y_1^0Y_2^1Y_3^0 \vee Y_1^0Y_2^1Y_3^1 \vee Y_1^0Y_2^1Y_3^2 \vee Y_1^1Y_2^0Y_3^1 \vee Y_1^1Y_2^0Y_3^2 \vee Y_1^1Y_2^0Y_3^3 \vee Y_1^1Y_2^1Y_3^0 \vee Y_1^1Y_2^1Y_3^1 \vee Y_1^1Y_2^1Y_3^2 \vee Y_1^2Y_2^0Y_3^2 \vee Y_1^2Y_2^0Y_3^3, \quad \text{-----(3.3)}$$

or in a form

$$g(Y_1,Y_2,Y_3)=Y_1^{\{0,1\}}\cdot Y_2^1\cdot Y_3^{\{0,1,2\}} \vee Y_1^{\{1,2\}}\cdot Y_2^0\cdot Y_3^{\{2,3\}} \vee Y_1^1\cdot Y_3^{\{1,2\}} \quad \text{-----(3.4)}$$

By using positional cube notations to represent terms[12]-[14], (3.3) and (3.4) can be represented as Table 3.3 and Table 3.4, respectively.

Table 3.3

Y_1 Y_2 Y_3
012 01 0123
100-01-1000
100-01-0100
100-01-0010
010-10-0100
010-10-0010
010-10-0001
010-01-1000
010-01-0100
010-01-0010
001-10-0010
001-10-0001

Table 3.4

Y_1 Y_2 Y_3
012 01 0123
110-01-1110
011-10-0011
010-11-0110

Lemma 3.4: Let f, ψ_i, and g be functions such that

$$f: B^{n_1} \times B^{n_2} \times \dots \times B^{n_r} \rightarrow B,$$

$$\psi_i: B^{n_i} \rightarrow M_i \quad ; \quad M_i = \{0,1,\dots,k_i-1\},$$

$$g: M_1 \times M_2 \times \dots \times M_r \rightarrow B.$$

and $f(X_1,X_2,\dots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\dots,\psi_r(X_r))$. Let $\psi_i=(L_0^i,L_1^i,\dots,L_{k_i-1}^i)$ be a partition function of B^{n_i} induced by the relation $\overset{i}{\sim}$, and let $\psi_i(\underline{a})=j \iff \underline{a}\in L_j^i$. A literal $Y_i^{T_i}$ of the expression $g(Y_1,Y_2,\dots,Y_r)$ corresponds to a literal

$$X_i^{S_i} \; ; \; S_i = \bigcup_{j\in T_i} L_j^i$$

of the expression $f(X_1,X_2,\dots,X_r)$. And a term $Y_1^{T_1}\cdot Y_2^{T_2}\cdot \dots \cdot Y_r^{T_r}$ of $g(Y)$ corresponds to a term $X_1^{S_1}\cdot X_2^{S_2}\cdot \dots \cdot X_r^{S_r}$ of $f(X)$.

Proof: It is easy to show by Definition 3.1 and Definition 3.2. Q.E.D.

Example 3.4: Consider the function $f(X)$ of Example 3.1 and the function $g(Y)$ of Example 3.3. For the term $Y_1^{\{0,1\}}\cdot Y_2^1 \cdot Y_3^{\{0,1,2\}}$ of $g(Y)$, the corresponding term of $f(X)$ is $X_1^{\{(00),(01),(10)\}}\cdot X_2^{\{(01),(10)\}}\cdot X_3^{\{(00),(01),(10)\}}$.
For the term $Y_1^{\{1,2\}}\cdot Y_2^0 \cdot Y_3^{\{2,3\}}$, the corresponding term of $f(X)$ is $X_1^{\{(01),(10),(11)\}}\cdot X_2^{\{(00),(11)\}}\cdot X_3^{\{(10),(11)\}}$, and for the term $Y_1^1\cdot Y_3^{\{1,2\}}$, the corresponding term of $f(X)$ is $X_1^{\{(01),(10)\}}\cdot X_3^{\{(01),(10)\}}$.

Theorem 3.1: Let two expressions

$$f(X_1,X_2,\dots,X_r)=\bigvee_{(S_1,S_2,\dots,S_r)} X_1^{S_1}\cdot X_2^{S_2}\cdot \dots \cdot X_r^{S_r}$$

and

$$g(Y_1,Y_2,\dots,Y_r)=\bigvee_{(T_1,T_2,\dots,T_r)} Y_1^{T_1}\cdot Y_2^{T_2}\cdot \dots \cdot Y_r^{T_r}$$

satisfy the relation $f(X_1,X_2,\dots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\dots,\psi_r(X_r))$. If P_1 and Q_1 are minimal expressions for $f(X)$ and $g(Y)$, respectively, then

$$t(P_1)=t(Q_1) \le \left(\prod_{i=1}^{r} k_i\right)/\left(\max_i\{k_i\}\right),$$

where $\psi_i: B^{n_i} \rightarrow M_i$; $M_i=\{0,1,\dots,k_i-1\}$, and $d(X_i)=n_i$.

Proof: (1) For P_1, a minimal sum-of-products expression of $f(X)$, consider the expression P_2 which has the form

$$\bigvee_{(G_1,G_2,\dots,G_r)} Y_1^{G_1}\cdot Y_2^{G_2}\cdot \dots \cdot Y_r^{G_r}, \text{ where}$$

$G_i=\{j \mid L_j^i \subseteq A_i\}$ and $A_i=\{\underline{a} \mid \underline{a} \overset{i}{\sim} \underline{b}, \underline{b}\in S_i\}$. Clearly, $t(P_1)=t(P_2)$. It is easy to show that P_2 represents $g(Y)$. For Q_1, a minimal sum-of-products expression of $g(Y)$, consider the expression Q_2 which has the form

$$\bigvee_{(D_1,D_2,\dots,D_r)} X_1^{D_1}\, X_2^{D_2} \dots X_r^{D_r}, \text{ where } D_i = \bigcup_{j\in T_i} L_j^i.$$

Clearly, $t(Q_1)=t(Q_2)$. By Lemma 3.4, Q_2 represents $f(X)$. As P_1 is a minimal expression of $f(X)$, we have $t(P_1)\le t(Q_2)$. As Q_1 is a minimal expression of $g(Y)$, we have $t(Q_1)\le t(P_2)$. Therefore, $t(P_1)=t(Q_1)$.

(2) Assume without loss of generality that $\max_i\{k_i\}=k_1$. $g(Y)$ can be represented in the form

$$g(Y_1,Y_2,\dots,Y_r)=\bigvee_{(T_1,t_2,\dots,t_r)} Y_1^{T_1}\cdot Y_2^{t_2}\cdot Y_3^{t_3}\cdot \dots \cdot Y_r^{t_r}, \qquad \text{------(3.5)}$$

where $T_1\subseteq M_1$ and $t_i\in M_i$ $(i=2,3,\dots,r)$. The number of terms in (3.5) is at most

$$\prod_{i=2}^{r} k_i = \left(\prod_{i=1}^{r} k_i\right)/\left(\max_i\{k_i\}\right).$$

Hence, we have the theorem. Q.E.D.

Theorem 3.2: Let $(X_1,X_2,\dots,X_r)$ be a partition of X, and the D array generates all the maxterms of X_i for $i=1,2,\dots,r$. In order to minimize the size of the AND array for

$$f(X_1,X_2,\dots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\dots,\psi_r(X_r)),$$

it is sufficient to obtain a minimal sum-of-products expression of $g(Y)$ having the form

$$g(Y_1,Y_2,\dots,Y_r)=\bigvee_{(T_1,T_2,\dots,T_r)} Y_1^{T_1}\cdot Y_2^{T_2}\cdot \dots \cdot Y_r^{T_r}.$$

Proof: By Theorem 3.1. Q.E.D.

Example 3.5: The expression (3.4) of Example 3.3 is a minimal sum-of-products expression for $g(Y)$. Therefore, the corresponding minimal expression for $f(X)$ is given by $f(X_1,X_2,X_3)=$

$$X_1^{\{(00),(01),(10)\}}\cdot X_2^{\{(01),(10)\}}\cdot X_3^{\{(00),(01),(10)\}} \vee$$

$$X_1^{\{(01),(10),(11)\}}\cdot X_2^{\{(00),(11)\}}\cdot X_3^{\{(10),(11)\}} \vee$$

$$X_1^{\{(01),(10)\}}\cdot X_3^{\{(01),(10)\}}.$$

Table 3.5 shows the positional cube notation for $f(X)$.

Table 3.5 Positional cubes for f

X_1				X_2				X_3			
00	01	10	11	00	01	10	11	00	01	10	11
1	1	1	0	0	1	1	0	1	1	1	0
0	1	1	1	1	0	0	1	0	0	1	1
0	1	1	0	1	1	1	1	0	1	1	0

Corollary 3.1: Let f(X) be a function such that

$$f(X_1,X_2,\ldots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\ldots,\psi_r(X_r)).$$

The size of three-level PLA which is sufficient to realize f(X) is given by $C(n)=(2n+W)H+W$, where $W=(\prod_{i=1}^{r} k_i)/(\max_i\{k_i\})$, $H=\sum_{i=1}^{r} 2^{n_i}$, and $\psi_i: B^{n_i}\rightarrow M_i$; $M_i=\{0,1,\ldots,k_i-1\}$.

As shown in the examples of this section, it is clear that obtaining a minimal sum-of-products expression for g(Y) requires less computation time and less memories than obtaining that for f(X). The multiple-valued decomposition of two-valued logic function can be done in a similar way to ordinary two-valued decomposition.

IV. Synthesis of Multiple-Output Functions.

In the case of multiple-output functions, simultaneous minimization often produces better solution than individual minimization[15],[16]. In this section, we will show that the minimization of AND array for a multiple-output function as well as a single-output function can be done by a minimization of a multiple-valued input two-valued output function.

Theorem 4.1: In order to minimize the AND array for m output functions

$$f_j: B^{n_1}\times B^{n_2}\times \ldots \times B^{n_r}\rightarrow B \quad (j=0,1,\ldots,m-1),$$

it is sufficient to obtain a minimal sum-of-products expression for the function

$$F: B^{n_1}\times B^{n_2}\times \ldots \times B^{n_r}\times M\rightarrow B$$

having a form

$$F(X_1,X_2,\ldots,X_r,Z)=\bigvee_{(S_1,S_2,\ldots,S_r,R)} X_1^{S_1}\cdot X_2^{S_2}\cdot\ldots X_r^{S_r}\cdot Z^R,$$

where $F(X_1,X_2,\ldots,X_r,j)=f_j(X_1,X_2,\ldots,X_r)$, $S_i\subseteq B^{n_i}$, $R\subseteq M$, and $M=\{0,1,\ldots,m-1\}$.

Proof: For expressions of f_j $(j=0,1,\ldots,m-1)$:

$$f_j(X_1,X_2,\ldots,X_r)=\bigvee_{(S_1,S_2,\ldots,S_r)} X_1^{S_1}\cdot X_2^{S_2}\cdot\ldots\cdot X_r^{S_r},$$

consider a expression for F shown above. By definition,

$$X_1^{S_1}\cdot X_2^{S_2}\cdot\ldots\cdot X_r^{S_r}\le f_j \iff X_1^{S_1}\cdot X_2^{S_2}\cdot\ldots\cdot X_r^{S_r}\cdot Z\le F \text{ and } j\in R.$$

It is easy to see that the number of terms in F is equal to the number of columns of the AND array. Hence the theorem. Q.E.D.

Example 4.1: Consider the six-variable three-output function shown in Table 4.1, where the function is represented as a set of positional cubes. Let the partition of X be (X_1,X_2,X_3), where $X_1=(x_1,x_2)$, $X_2=(x_3,x_4)$, and $X_3=(x_5,x_6)$. Let Z be a variable which denotes the outputs. Consider the function

$$F(X_1,X_2,X_3,Z): B^2\times B^2\times B^2\times\{0,1,2\}\rightarrow B.$$

F has the following properties:

$F(X_1,(01),X_3,Z)=F(X_1,(10),X_3,Z)$;

$F(X_1,X_2,(00),Z)=F(X_1,X_2,(11),Z)$; and

$F(X_1,X_2,(10),Z)=F(X_1,X_2,(01),Z)$.

So F can be decomposed as

$$F(X_1,X_2,X_3,Z)=G(\psi_1(X_1),\psi_2(X_2),\psi_3(X_3),Z),$$

where ψ_i are shown in Table 4.2. G is a function such that $G: \{0,1,2,3\}\times\{0,1,2\}\times\{0,1\}\times\{0,1,2\}\rightarrow\{0,1\}$. By Theorem 4.1, in order to minimize the size of the AND array for the muliple-output function, it is sufficient to obtain a minimal sum-of-products expression for G. The terms of G are shown in Table 4.3. It is easy to see that this is a minimal sum-of-products expression for G. The minimal sum-of-products expression for F which corresponds to G is shown in Table 4.4. The three-level PLA which realizes the given functions is shown in Fig.1.2.

Table 4.1 Six-variable three-output function

Input						Output		
x_1	x_2	x_3	x_4	x_5	x_6	f_0	f_1	f_2
01	01	01	01	01	01			
11-	10-	11-	10-	10-	10	1	1	0
11-	10-	11-	10-	01-	01	1	1	0
11-	10-	10-	11-	10-	10	1	1	0
11-	10-	10-	11-	01-	01	1	1	0
10-	11-	11-	10-	10-	10	1	1	0
10-	11-	11-	10-	01-	01	1	1	0
10-	11-	10-	11-	10-	10	1	1	0
10-	11-	10-	11-	01-	01	1	1	0
01-	11-	01-	01-	10-	10	0	1	1
01-	11-	01-	01-	01-	01	0	1	1
11-	01-	01-	01-	10-	10	0	1	1
11-	01-	01-	01-	01-	01	0	1	1
10-	01-	10-	01-	11-	11	1	0	1
10-	01-	01-	10-	11-	11	1	0	1
01-	10-	01-	11-	10-	10	1	1	1
01-	10-	01-	11-	01-	01	1	1	1
01-	10-	11-	01-	10-	10	1	1	1
01-	10-	11-	01-	01-	01	1	1	1
10-	01-	01-	11-	10-	10	1	1	1
10-	01-	01-	11-	01-	01	1	1	1
10-	01-	11-	01-	10-	10	1	1	1
10-	01-	11-	01-	01-	01	1	1	1

Table 4.2 Partition functions

X_i	$\psi_1(X_1)$	$\psi_2(X_2)$	$\psi_3(X_3)$
00	0	0	0
01	1	1	1
10	2	1	1
11	3	2	0

Table 4.3 Minimal expression for G.

Y_1	Y_2	Y_3	Z
0123	012	01	012
1110	110	10	110
0111	001	10	011
0100	010	11	101
0110	011	10	111

Table 4.4 Minimal expression for F

X_1					X_2					X_3					Z		
00	01	10	11		00	01	10	11		00	01	10	11		0	1	2
1	1	1	0	-	1	1	1	0	-	1	0	0	1	-	1	1	0
0	1	1	1	-	0	0	0	1	-	1	0	0	1	-	0	1	1
0	1	0	0	-	0	1	1	0	-	1	1	1	1	-	1	0	1
0	1	1	0	-	0	1	1	1	-	1	0	0	1	-	1	1	1

Corollary 4.1: Let $F(X_1,X_2,\ldots,X_r,Z)$ be a function which represents a multiple-output function f_j $(j=0,1,\ldots,m-1)$, and can be represented as $F(X_1,X_2,\ldots,X_r,Z)=G(\psi_1(X_1),\psi_2(X_2),\ldots,\psi_r(X_r),Z)$. The size of a three-level PLA which is sufficient to realize the function is given by

$$C(n)=(2n+W)H+mW, \text{ where}$$

$$W=\left(\prod_{i=1}^{r+1} k_i\right)/(\max_i \{k_i\}),\quad H=\sum_{i=1}^{r} 2^{n_i},\quad \psi_i: B^{n_i} \to M_i;$$

$M_i=\{0,1,\ldots,k_{i-1}\}$, and $k_{r+1}=m$.

V. Concluding Remarks.

In this paper, it is shown that the minimization of the AND array corresponds to the minimization of a multiple-valued input two-valued output logic function. The minimization of the multiple-valued input two-valued output function can be done in a similar way to that of Quine-McCluskey's, when the number of input variables is small. However, when the number of input variables is large, it is quite difficult to obtain a minimal expression. Because it is known that the number of prime implicants of multiple-valued input function is much larger than that of two-valued input function. Furthermore, Quine-McCluskey's method contains the problem of minimal covering which is known to be NP-complete[21]. Therefore, it will be practical to use the heuristic approach [2],[25], when the number of input variables is large.

Acknowledgement

The author would like to thank Prof. H.Ozaki, Prof. H. Terada, and Prof. K.Kinoshita for their help and encouragement.

References

[1] H.Fleisher and L.I.Maissel,"An introduction to array logic", IBM J.Res. Develop. vol.19, pp.98-109, March 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] D.L.Greer,"An associative logic matrix," IEEE J.Solid-State Circuits, vol.SC-11, pp.679-691, Oct. 1976.

[4] W.N.Carr and J.P.Mize, MOS/LSI Design and Application, Texas Instruments Electronics Series, McGraw-Hill, New York, 1972.

[5] T.Sasao,"Basic considerations on the realization of programmable logic arrays (1), (2), and (3)", (in Japanese), The Institute of Electronics and Communication Engineers of Japan, EC 77-12, EC 77-13, and EC 77-74, July 1977 and March 1978.

[6] E.L.Lawler,"An approach to multi level Boolean minimization," J.ACM vol.11, pp.283-295, July 1964.

[7] R.A.Meo, "On the minimal third order expressions of a Boolean function," Proc. 3rd Ann. Sympo. Switching Circuit and Logical Design, PP.6-24, Sept. 1962.

[8] R.L.Ashenhurst,"The decomposition of switching functions," Proc. of an Intl. Sympo. on Theory of Switching, April 2-5, 1957, Ann. Computation Lab. Harverd University, vol.29, pp.74-116, 1959.

[9] H.A.Curtis, A New Approach to the Design of Switching Circuits, D.Van Nostrand Co., Princeton, N.J., 1962.

[10] D.M.Miller and J.C.Muzio,"Two-place decomposition and the synthesis of many valued switching circuits," Conf. Record of the 1976 Sympo. on Multiple-Valued Logic, pp. 164-168.

[11] K.M.Waliuzzaman and Z.G.Vranesic, "On the decomposition of multivalued switching functions," Computer J., vol.13, No.4, pp.359-362, Nov. 1970.

[12] S.Y.H.Su and P.T.Cheung, "Computer minimization of multi-valued switching functions," IEEE Trans. Comput., vol. C-21, pp.995-1003, Sept. 1972.

[13] S.Y.H.Su and P.T.Cheung, "Cubical notation for computer-aided processing of multiple-valued switching functions," Proc. of 1976 Inter Sympo. on Multiple-Valued Logic, pp.24-29, May 1976.

[14] P.L.Tison, "An algebra for logic systems----Switching circuits application," IEEE Trans. Comput. vol. C-20, pp.339-351, March 1971.

[15] T.C.Bartee, "Computer design of multiple-output networks," IRE Trans. Electron. Comput., vol. EC-10, pp.21-30, March 1961.

[16] Y.H.Su and D.L.Dietmeyer, "Computer reduction of two-level multiple-output switching circuits," IEEE Trans. Comput. vol.C-18, pp.56-63, Jan. 1969.

[17] D.L.Dietmeyer and P.R.Schneider, "Identification of symmetry, redundancy, and equivalence class of Boolean functions," IEEE Trans. Electron Comput. vol.EC-16, pp.804-817, Dec. 1967.

[18] S.C.Lee, "Vector Boolean algebra and calculus," IEEE Trans. Comput. vol. C-25, pp.865-874, Sept. 1976.

[19] J.Dussault, G.Metze and M.Krieger, "A multi-valued switching algebra with Boolean properties, " Proc. of 1976 Inter. Sympo. on Multiple-Valued Logic, pp.68-73, May 1976.

[20] P.T.Cheung, "Identification of different functional properties of multiple valued Switching functions," Proc. of 1976 Inter. Sympo. on Multiple-Valued Logic, pp.74-78, May 1976.

[21] R.M.Karp,"Reducibility among combinatorial problems," Complexity of Computer Computations, R.E.Miller and J.W. Thatcher. eds. Plenum. Press, New York, 1972.

[22] Y.Kambayashi,"Optimum logic design using memory -type arrays," Proceedings of the International Symposium on Uniformly Structured Automata and Logic, Tokyo, Aug. 1975.

[23] J.P.Deschmps and A.Thayse, "Representation of discrete functions," Proc. of 1975 Inter. Symposium on Multiple-Valued Logic, pp.99-111, May 1975.

[24] A.Thayse and J.P.Deschamps, "Logic properties of unate discrete and switching functions," IEEE Trans. Comput. vol. C-26, pp.1202-1212, Dec. 1977.

[25] J.P.Roth, "Programmed logic array optimization," IEEE Trans. Comput. vol. C-27, pp.174-176, February 1978.