MULTIPLE-VALUED LOGIC AND THE DESIGN OF PROGRAMMABLE LOGIC ARRAYS WITH DECODERS*

Tsutomu Sasao and Hiroaki Terada

Department of Electronic Engineering
Osaka University, Osaka 565, Japan

Abstract

A programmable logic arrays (PLA's) with decoders consists of three parts; the fixed size decoders, the AND array, and the OR array. Basic problems on the design of PLA's with decoders are considered. Main subjects included are 1) The minimization of the AND array: it corresponds to the minimization of multiple-valued logic function; 2) The sizes of PLA's for various classes of functions which are obtained theoretically; 3) The assignment problem of input variables; 4) The average sizes of PLA's for randomly generated functions which are obtained by computer simulation; and 5) The average number of prime implicants of multiple-valued logic functions which is obtained theoretically.

I. Introduction

The problem of logic minimization - covering switching function - is classical. The use of programmable logic arrays (PLA's) as a solution to an acute problem of LSI fabrication has led to a resurgence of interest in this problem [1]-[7]. In this paper, two types of PLA's will be considered: two level PLA's and PLA's with decoders. The first type of a PLA, the two-level PLA is shown in Fig.1.1. It consists of the AND array and the OR array. For example, the four-input function shown in Table 1.1 can be realized by the two-level PLA shown in Fig.1.2. This PLA corresponds to a two-level AND-OR circuit. The size of this PLA is defined as $C(n)=(2n+m)W$. In order to realize an arbitrary function of n variables, a PLA with the size of $(n+(1/2))2^n$ is necessary. Therefore, PLA's of this type require large arrays for complex functions. The second type of a PLA, the PLA with decoders is shown in Fig.1.3. Each decoder generates all the maxterms of its input variables. For example, the function of Table 1.1 can be realized as shown in Fig.1.4. The size of this PLA is defined as $C(n)=(H+m)W$. The two-level PLA can be considered as a special case of PLA of this type, i.e, the PLA with one-bit decoders.

Table 1.2 shows the sizes of PLA's in order to realize various kind of functions for each type of realization. Table 1.3 shows the average size of PLA's in order to realize randomly generated functions. From these tables, PLA's with two-bit decoders require smaller arrays than two-level PLA's.

* This work is supported in part by the Ministry of Education of Japan under the Grant 375187, (Tsutomu Sasao, 1978). ** The number of the terms in a sum-of-products expression P is denoted by t(P). *** The number of the variables in $\{X_i\}$ is denoted by $d(X_i)$.

In II, basic ideas of the design of PLA's with decoders are considered, and it is shown that the minimization of the AND array corresponds to the minimization of multiple-valued input two-valued output function. In III, sizes of PLA's to realize various classes of functions are derived by using the theory of multiple-valued decomposition. In IV, the assignment problem of input variables is introduced. In V, the average numbers of prime implicants of multiple-valued logic functions are derived. This result is useful for the estimation of the computation time and the memory requirement for the classical minimization of logic functions. In VI, the result of computer simulation is summarized.

II. Programmable Logic Arrays with Decoders

In this section, a design method which minimizes the size of a PLA with decoders will be considered [6]-[7].

Definition 2.1: Let $X=(x_1,x_2,\ldots,x_n)$ be a variable in B, where $B=\{0,1\}$. The set of variables in X is denoted by $\{X\}$. $(X_1,X_2,\ldots,X_r)$ 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 2.2: 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 denotes the function such that $X^S=\bigvee_{\underline{a}_i\in S} X^{\underline{a}_i}$. 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. 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.**

Theorem 2.1: Let $(X_1,X_2,\ldots,X_r)$ be a partition of X. An arbitrary function f(X) can be represented in a form

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

where $S_i\subseteq B^{n_i}$ and $n_i=d(X_i)$.*** In a PLA with decoders, if each decoder 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. If P is a minimal sum-of-

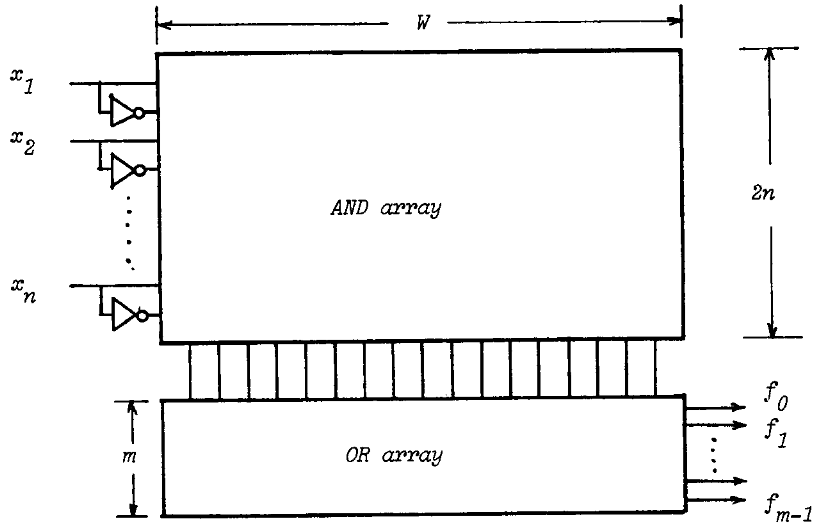

Fig. 1.1 Two-level PLA

Table 1.1 4-variable function

x_1	x_2	x_3	x_4	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

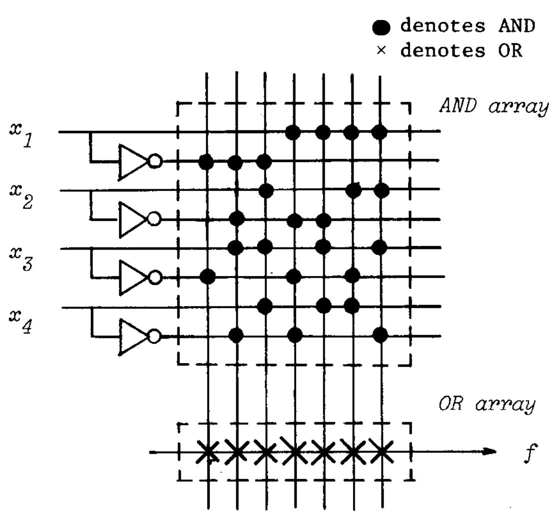

Fig.1.2 Realization of Table 1.1 by a two-level PLA

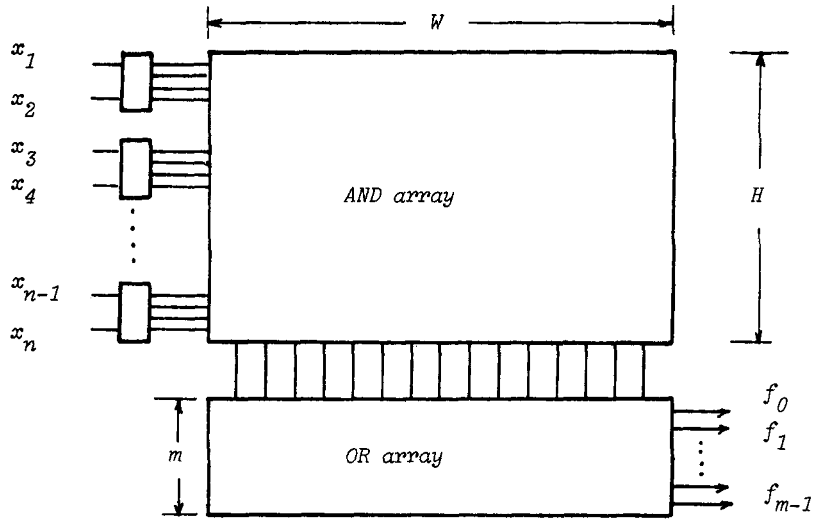

Fig. 1.3(a) PLA with two-bit decoders

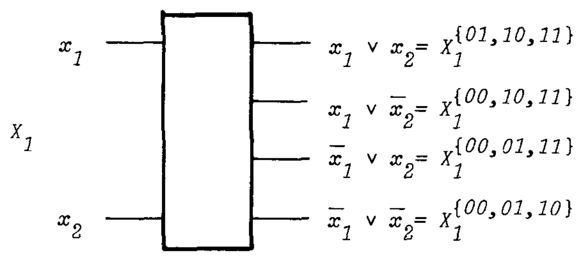

Fig. 1.3 (b) Two-bit decoder

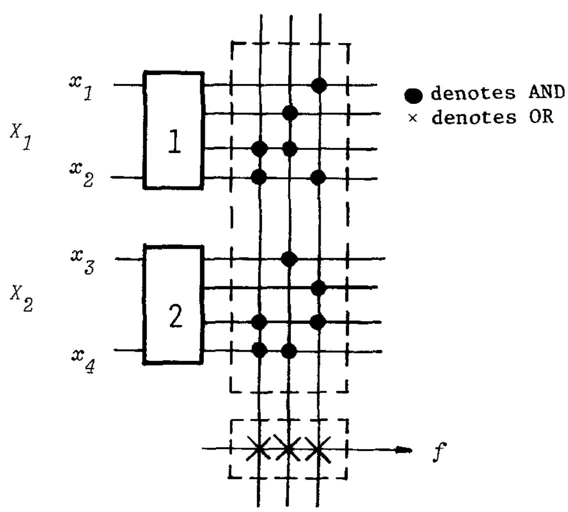

Fig. 1.4 Realization of Table 1.1 by a PLA with two-bit decoders. When the input variables are assigned as $X_1=(x_1,x_2)$, $X_2=(x_3,x_4)$.

Table 1.2 Sizes of the PLA's to realize n-variable functions (worst case).

	Two-level PLA	PLA with two-bit decoders		
n	Parity function	Arbitrary function	Symmetric function	Parity function
6	416	208	117	52
8	2,176	1,088	459	136
10	10,752	5,376	1,701	336
12	51,200	25,600	6,075	800
14	237,568	118,784	21,141	1,856
16	1,081,344	540,672	72,171	4,224
n	$(n+\frac{1}{2})2^n$	$\frac{1}{2}(n+\frac{1}{2})2^n$	$\frac{2}{3}(n+\frac{1}{2})\sqrt{3}^n$	$(n+\frac{1}{2})\sqrt{2}^n$

Table 1.3 Average size of PLA's in order to realize randomly generated functions of n variables.*

	Two-level PLA					PLA with two-bit decoders				
	d					d				
n	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
6	75.4	109.2	148.2	170.3	185.9	63.7	88.4	117.0	128.7	132.6
8	321.3	537.2	661.3	759.9	770.1	280.5	464.1	535.5	578.0	564.4
10	1545.6	2478.0	3009.3	3234.0	3423.0	1383.0	2068.5	2349.9	2446.5	2520.0

"n" denotes the number of the external input variables. "d" denotes the percentage of input combinations which are mapped to 1.

* The entries of 40% and 50% of 10-variable function denote the average of 5 near minimal solutions; the other entries are the average of 10 minimal solutions.

Table 5.1 Average Number of Prime Implicants

p=2 (Two-valued function) $G_2(n,u)$ $u=\frac{2^n}{100}\times d$

	d									
n	10%	20%	30%	40%	50%	60%	70%	80%	90%	$G_2(n)$
6	5	9	15	19	24	28	31	31	25	24
8	20	42	65	90	118	145	168	181	157	118
10	83	181	294	421	585	757	940	1109	1100	585
12	342	790	1311	1988	2902	3909	5265	6677	7663	2902
14	1418	3428	5849	9380	14225	19934	28993	39348	51714	14225

p=4 (Four-valued function) $G_4(n,u)$ $u=\frac{2^n}{100}\times d$

	d									
n	10%	20%	30%	40%	50%	60%	70%	80%	90%	$G_4(n)$
3	5	9	14	19	24	28	31	31	24	24
4	19	42	68	99	136	176	218	249	223	136
5	83	191	330	515	757	1067	1464	1931	2210	758
6	353	871	1580	2636	4093	6232	9355	13991	20050	4095
7	1521	3915	7508	13219	21561	35153	56857	95966	165243	21565

"n" denotes the number of variables. "d" denotes the percentage of input combinations which are mapped to 1.

products expression having a form (2.1), then

$t(P)\leq 2^{n-\max_i\{n_i\}}$, where $n_i=d(X_i)$.

Example 2.1:

(i) In the case of the two-level PLA.

Let $X_1=(x_1)$, $X_2=(x_2)$, $X_3=(x_3)$, and $X_4=(x_4)$ be a (trivial) partition of $X=(x_1,x_2,x_3,x_4)$. The function of Table 1.1 can be represented as follows.*

$$f(X_1,X_2,X_3,X_4)=X_1^0X_2^0X_3^0X_4^0 \vee X_1^0X_2^0X_3^0X_4^1 \vee X_1^0X_2^0X_3^1X_4^0 \vee X_1^0X_2^1X_3^0X_4^0 \vee X_1^0X_2^1X_3^0X_4^1 \vee X_1^0X_2^1X_3^1X_4^1 \vee X_1^1X_2^0X_3^0X_4^0 \vee X_1^1X_2^0X_3^1X_4^1 \vee X_1^1X_2^1X_3^0X_4^1 \vee X_1^1X_2^1X_3^1X_4^0 .$$

This expression can be simplified as follows.

$$f(X_1,X_2,X_3,X_4)=X_1^0X_2^{\{0,1\}}\cdot X_3^0X_4^{\{0,1\}} \vee X_1^0X_2^0X_3^1X_4^0 \vee X_1^0X_2^1X_3^1X_4^1 \vee X_1^1X_2^0X_3^0X_4^0 \vee X_1^1X_2^0X_3^1X_4^1 \vee X_1^1X_2^1X_3^0X_4^1 \vee X_1^1X_2^1X_3^1X_4^0. \qquad \text{------(2.2)}$$

Fig.1.2 is a two-level PLA realization of this function. Each column of the AND array corresponds to each term of (2.2).

(ii) In the case of the PLA with two-bit decoders.

Let $X=(X_1,X_2)$, $X_1=(x_1,x_2)$, and $X_2=(x_3,x_4)$ be a partition of $X=(x_1,x_2,x_3,x_4)$. The function of Table 1.1 can be represented as

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

This expression can be simplified as follows.**

$$f(X_1,X_2)=X_1^{\{00,01\}}\cdot X_2^{\{00,01\}} \vee X_1^{\{00,11\}}\cdot X_2^{\{01,10\}} \vee X_1^{\{01,10\}}\cdot X_2^{\{00,11\}} \qquad \text{------(2.3)}$$

Fig.1.4 is a PLA with two-bit decoders realization of this function. Each column of the AND array corresponds to each term of (2.3).

(End of the example).

By Theorem 2.1, 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 (2.1). In the case of PLA with two-bit decoders, the expression denotes the four-valued logic function $\{00,01,10,11\}^r \to \{0,1\}$, where $n=2r$.

Theorem 2.2: Let W_1 and W_2 be the widths of the two-level PLA and the PLA with decoders in order to realize a function, respectively. Then $W_1 \geq W_2$.

In fact, PLA's with two-bit decoders require smaller arrays than two-level PLA's. The result of computer simulation in VI shows this fact.

* $X_1^0\cdot X_2^0\cdot X_3^0\cdot X_4^0$ is sometimes denoted by $X_1^0X_2^0X_3^0X_4^0$.

$X_i^{\{0\}}$ is sometimes denoted by X_i^0

** $X_i^{\{(00),(11)\}}$ is sometimes denoted by $X_i^{\{00,11\}}$.

III. Sizes of PLA's in order to Realize Various Classes of Functions.

In this section, sizes of PLA's with two-bit decoders in order to realize various classes of functions will be considered.

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

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

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

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

where $f(X|\underline{a}\to X_i)$ 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 $\psi_i: B^{n_i} \to M_i$; $M_i=\{0,1,\ldots,k_i-1\}$ such that $\psi_i(\underline{a})=j \iff \underline{a} \in L_j^i$ is called a partition function of B^{n_i}.

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 3.1: 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 function

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

$$f(X_1,X_2,\ldots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\ldots,\psi_r(X_r)). \qquad \text{-------(3.1)}$$

The function g can be represented in a form

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

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

Theorem 3.1: Let two expressions (2.1) and (3.2) satisfy the relation (3.1). If P and Q are minimal expressions for f(X) and g(Y), respectively, then

$$t(P)=t(Q) \leq \left(\prod_{i=1}^{r} k_i\right)/(\max_i\{k_i\}) . \qquad \text{-------(3.3)}$$

If there exists $\underline{a} \in B^{n_i}$ such that $f(X|\underline{a}\to X_i)\equiv 0$, then k_i can be reduced by one.

Lemma 3.2: There exists a symmetric function of n variables which requires 3^{r-1} terms in a PLA with two-bit decoders realization, where $n=2r$.

(Proof) Let f(X) be a symmetric function. f(X) can be written as

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

where $X_i=(x_{2i-1},x_{2i})$, $\psi_i(00)=0$, $\psi_i(01)=\psi_i(10)=1$, and $\psi_i(11)=2$. Let g(Y) be a function such that

$$g(Y_1,Y_2,\ldots,Y_r)=\begin{cases}1 & \text{if } Y_1+Y_2+\ldots+Y_r=0 \pmod 3.\\ 0 & \text{otherwise.}\end{cases}$$

g(Y) can be written as

$$g(Y_1,Y_2,\ldots,Y_r)=\bigvee_{\substack{t_1+t_2+\ldots+t_r=3k\\ k=0,1,2,3,\ldots}} Y_1^{t_1}\cdot Y_2^{t_2}\cdot\ldots\cdot Y_r^{t_r} \quad t_i\in\{0,1,2\}. \qquad \text{-------(3.4)}$$

Each term of a expression for g(Y) is minterm, because if the expression has a term having a form

$$Y_1^{t_1}\cdot Y_2^{t_2}\cdot\ldots\cdot Y_i^{S_i}\cdot\ldots\cdot Y_r^{t_r}, \quad |S_i|\geq 2,$$

then, it cannot satisfy the condition for g(Y). Therefore, the minimal expression which represents g(Y) has the form (3.4). For arbitrary $t_2,t_3,\ldots,t_r\in\{0,1,2\}$, there exists $t_1\in\{0,1,2\}$ such that $t_1+t_2+\ldots+t_r=3k$. So the number of terms of (3.4) is 3^{r-1}. Hence, the number of terms which is necessary to represent f(X) is 3^{r-1}. Q.E.D.

Lemma 3.3: There exists an n-variable function which requires 2^{n-2} terms in a PLA with two-bit decoders realization, if the assignment of the variables to the decoders is fixed, where n=2r.

(Proof) f(X) can be written as $f(X_1,X_2,\ldots,X_r)=g(\psi_1(X_1),\psi_2(X_2),\ldots,\psi_r(X_r))$, where $\psi_i(00)=0$, $\psi_i(01)=1$, $\psi_i(10)=2$, and $\psi_i(11)=3$. Let g(Y) be a function such that

$$g(Y_1,Y_2,\ldots,Y_r)=\begin{cases}1 & \text{if } Y_1+Y_2+\ldots+Y_r=0 \pmod 4.\\ 0 & \text{otherwise.}\end{cases}$$

g(Y) can be written as

$$g(Y_1,Y_2,\ldots,Y_r)=\bigvee_{\substack{t_1+t_2+\ldots+t_r=4k\\ k=0,1,\ldots}} Y_1^{t_1}\cdot Y_2^{t_2}\cdot\ldots\cdot Y_r^{t_r} \quad t_i\in\{0,1,2,3\} \qquad \text{--------(3.5)}$$

Similar to Lemma 3.2, we can show that (3.5) is the minimal expression and the number of the terms in (3.5) is $4^{r-1}=2^{n-2}$. Therefore, the number of terms which is necessary to represent f is 2^{n-2}. Q.E.D.

Theorem 3.2: In order to realize an n-variable function in a PLA with two-bit decoders, the following sizes are necessary and sufficient, when the assignment of the variables are fixed to the decoders, where n=2r.

1) For an arbitrary function: $\frac{1}{2}(n+\frac{1}{2})2^n$
2) For a symmetric function: $\frac{2}{3}(n+\frac{1}{2})\sqrt{3}^{\,n}$
3) For a parity function : $(n+\frac{1}{2})\sqrt{2}^{\,n}$

(Proof) The size of a PLA with two-bit decoders is defined as $C(n)=(2n+1)W$.

1) Sufficiency: By Theorem 2.1. Necessity: By Lemma 3.3.
2) Sufficiency. By the definition of the symmetric function, there is a partition function ψ_i such that $\psi_i(00)=0, \psi_i(01)=\psi_i(10)=1$, and $\psi_i(11)=2$ for each i. By Theorem 3.1, we have $W \leq (\prod_{i=1}^{r} 3)/3 = 3^{r-1}$. Necessity: By Lemma 3.2.
3) Sufficiency: By the definition of the parity function. There is a partition function ψ_i such that $\psi_i(00)=\psi_i(11)=0$ and $\psi_i(10)=\psi_i(01)=1$, for each i. By Theorem 3.1, $W \leq (\prod_{i=1}^{r} 2)/2 = 2^{r-1}$. Necessity: Similar to Lemma 3.2. Q.E.D.

The sizes of PLA's with two-bit decoders in order to realize the various classes of functions are shown in Table 1.2.

IV. The Assignment Problem of Input Variables

In this section, the assignment problem of input variables is introduced [20]-[23]. In the case of PLA's with two-bit decoders, we often have an option of the assignment of the input variables to the decoders. We explain this by using the following example.

Example 4.1: Let us realize the function of Table 1.1 by using a PLA with two-bit decoders. Assume that $X=(X_1,X_2)$ is a partition of the input variables X. There exists three possible way of assignment of four input variables to two decoders.

1) When the input variables are assigned as $X_1=(x_1,x_2)$ and $X_2=(x_3,x_4)$.
   As shown in Example 2.1 and Fig.1.4, three columns are necessary to realize $f(X_1,X_2)$:
   $$f(X_1,X_2)=X_1^{\{00,01\}}\cdot X_2^{\{00,01\}} \vee X_1^{\{00,11\}}\cdot X_2^{\{01,10\}} \vee X_1^{\{01,10\}}\cdot X_2^{\{00,11\}}.$$
2) When the input variables are assigned as $X_1=(x_1,x_3)$ and $X_2=(x_2,x_4)$. (See Fig. 4.1)
   $f(X_1,X_2)$ can be written as
   $$f(X_1,X_2)=X_1^{\{00,01,10\}}\cdot X_2^{\{00,11\}} \vee X_1^{\{00,11\}}\cdot X_2^{\{01,10\}}.$$
3) When the input variables are assigned as $X_1=(x_1,x_4)$ and $X_2=(x_2,x_3)$. (See Fig. 4.2)
   $f(X_1,X_2)$ can be written as
   $$f(X_1,X_2)=X_1^{\{00,11\}}\cdot X_2^{\{01,10\}} \vee X_1^{\{01,10\}}\cdot X_2^{\{00,11\}} \vee X_1^{\{00,01\}}\cdot X_2^{\{00,10\}}.$$

Therefore, when the input variables are assigned as shown in Fig.4.1, the array is the minimum.

(End of the example).

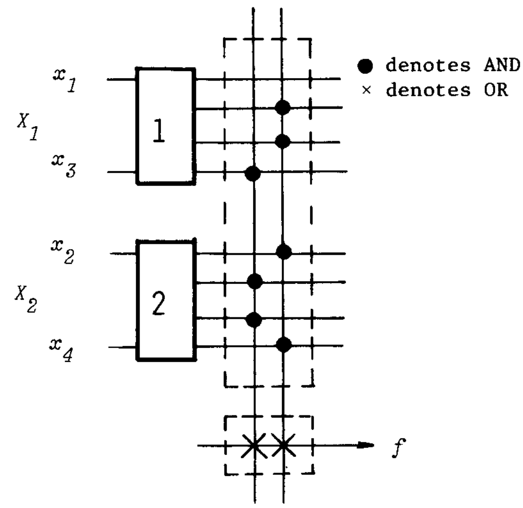

Fig. 4.1 Realization of Table 1.1

When the input variables are assigned as $X_1=(x_1,x_3)$, $X_2=(x_2,x_4)$.

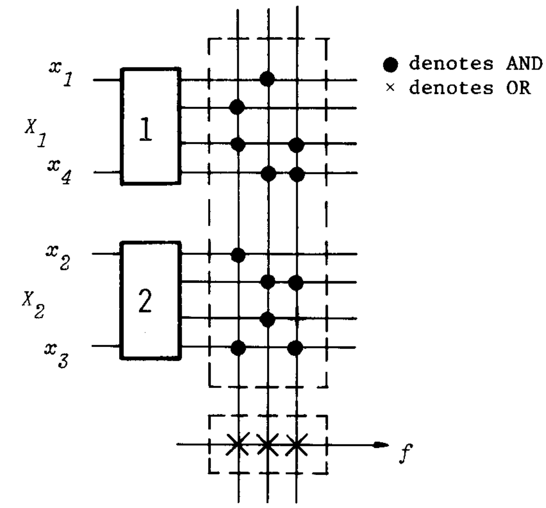

Fig. 4.2 Realization of Table 1.1

When the input variables are assigned as $X_1=(x_1,x_4)$, $X_2=(x_2,x_3)$.

Table 6.1 Average number of terms in order to realize 8-variable functions.

Type of PLA		density : d (%) 5%	10%	15%	20%	30%	40%
PLA with two-bit decoders	When the assignment is optimal	8.4	14.7	19.9	24.2	28.4	30.2
	Average [1]	10.10	17.27	22.71	27.80	31.77	33.58
	When the assignment is worst	10.8	19.2	25.5	31.1	35.3	37.0
	Standard deviation [2]	0.629	0.984	1.23	1.41	1.42	1.41
Two-level PLA		10.8	19.2	26.5	33.1	39.5	44.4

* Each entry is the average of 10 randomly generated functions.

1. The average of 105 assignments.

2. $\sigma_n^2 = \frac{1}{n}\sum_{i=1}^{n}(t_i - \bar{t})$, $\bar{t} = \frac{1}{n}\sum_{i=1}^{n} t_i$.

The problem to find an assignment of the input variables to the decoders which minimize the size of the PLA is called "Assignment Problem of Input Variables". In order to find the optimal assginment, the concept of the previous section is sometimes useful. We explain this by using the following example.

Example 4.2: Consider the function of Table 1.1. Let (X_1,X_2) be a partition of X.

1) When the input variables are assigned as $X_1=(x_1,x_2)$ and $X_2=(x_3,x_4)$. By definition 3.1,

$f(X|(00)\to X_1)=X_2^{\{00,01,10\}}$. $f(X|(00)\to X_2)=X_1^{\{00,01,10\}}$

$f(X|(01)\to X_1)=X_2^{\{00,01,11\}}$. $f(X|(01)\to X_2)=X_1^{\{00,01,11\}}$

$f(X|(10)\to X_1)=X_2^{\{00,11\}}$. $f(X|(10)\to X_2)=X_1^{\{00,11\}}$

$f(X|(11)\to X_1)=X_2^{\{01,10\}}$. $f(X|(11)\to X_2)=X_1^{\{01,10\}}$

We have $k_1=k_2=4$.

2) When the input variables are assigned as $X_1=(x_1,x_3)$ and $X_2=(x_2,x_4)$.

$f(X|(00)\to X_1)=X_2^{\{00,01,10,11\}}$ $f(X|(00)\to X_2)=X_1^{\{00,01,10\}}$

$f(X|(01)\to X_1)=X_2^{\{00,11\}}$. $f(X|(01)\to X_2)=X_1^{\{00,11\}}$

$f(X|(10)\to X_1)=X_2^{\{00,11\}}$. $f(X|(10)\to X_2)=X_1^{\{00,11\}}$

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

We have $k_1=3$ and $k_2=2$.

3) When the input variables are assigned as $X_1=(x_1,x_4)$ and $X_2=(x_2,x_3)$.

$f(X|(00)\to X_1)=X_2^{\{00,01,10\}}$. $f(X|(00)\to X_2)=X_1^{\{00,01,10\}}$

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

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

$f(X|(11)\to X_1)=X_2^{\{01,10\}}$. $f(X|(11)\to X_2)=X_1^{\{01,10\}}$

We have $k_1=k_2=4$.

Let P_i be a minimal sum-of-products expression of $f(X_1,X_2)$ for each assignment, where i=1,2,3. By Theorem 3.1, we have

1) For the assginment $X_1=(x_1,x_2)$, $X_2=(x_3,x_4)$: $t(P_1) \le 4$.
2) For the assignment $X_1=(x_1,x_3)$, $X_2=(x_2,x_4)$: $t(P_2) \le 2$.
3) For the assignment $X_1=(x_1,x_4)$, $X_2=(x_2,x_3)$; $t(P_3) \le 4$.

Because the assignment $X_1=(x_1,x_3)$, $X_2=(x_2,x_4)$ has the minimum upper bound, it is the first candidate of the optimal assignment.

(End of the example).

V. Number of Prime Implicants

In order to minimize the size of the PLA, it is sufficient to obtain a minimal sum-of-products expression. Classical method, which first obtains the set of all the prime implicants and then obtains the minimal covering of it, can be extend to manipulate multiple-valued variables [8]-[9]. The number of prime implicants shows the inherent complexity of the classical algorithm.

5.1 Maximal Number of Prime Implicants.

Lemma 5.1: Let $\mu(n,p)$ be the maximal number of prime implicants of the p-valued n-variable function $\{0,1,\dots,p-1\}^n \to \{0,1\}$. When n=tm and $t=2^p-1$, the following relation holds.

$(n!)/(m!)^t \le \mu(n,p) \le t^n$.

(Proof) First, consider a term $X_1^{S_1} \cdot X_2^{S_2} \cdots X_n^{S_n}$. The number of distinct subsets $S_i \subseteq \{0,1,\dots,p-1\}$ is t, therefore we have $\mu(n,p) \le t^n$. Next consider a term

$(X_1^{S_1}X_2^{S_1}\cdots X_m^{S_1})(X_{m+1}^{S_2}X_{m+2}^{S_2}\cdots X_{2m}^{S_2})\cdots(X_{n-m}^{S_t}X_{n-m+1}^{S_t}\cdots X_n^{S_t})$,

where $S_1,S_2,\dots,S_t$ are distinct sets. The number of different terms, when the variables are permutated, is $(n!)/(m!)^t$. Each term is not contained by other terms and is maximal. So all the terms are prime. Hence $(n!)/(m!)^t \le \mu(n,p)$. Q.E.D.

Corollary 5.1: There exist positive constants K_1, K_2, and K_3 such that

$$K_1(2^p-1)^n/\ n^{(2^{p-1}-1)} \le \mu(n,p),$$
$$K_2(3^n/n) \le \mu(n,2),$$
$$K_3(15^n/n^7) \le \mu(n,4).$$

(Proof) By using Stirling's formula $n! = \sqrt{2\pi}\ n^{n+\frac{1}{2}}\ e^{-n}$, we have

$(n!)/(m!)^t \simeq (2\pi)^{-(t-1)/2} \cdot (t^{(t/2)}) \cdot t^n \cdot n^{(1-t)/2}$,

where $t=2^p-1$. Q.E.D.

Corollary 5.1 shows that there is an n varialbe function of two-valued variables which has $\underline{O}(3^n/n)$ prime implicants* [12],[18],[19]. In the case of four-valued variables, there exists a function which has $\underline{O}(15^n/n^7)$ prime implicants. These functions are pathological ones and for the most functions, the numbers of prime implicants are much smaller.

---

* A function f(n) is said to be O(g(n)) iff there exists a positive constants K such that $|f(n)/g(n)| \to K$, when $n \to \infty$.

5.2 Average Number of Prime Implicants

Definition 5.1: Let f be a function such that $f: \{0,1,\ldots,p-1\}^n \to \{0,1\}$. Let $U=f^{-1}(1)$. $u=|U|$ is said to be the weight of f.

Theorem 5.1: The average number of prime implicants of p-valued n-variable function of weight u is given by

$$G_p(n,u)=\frac{1}{F^{(u)}}\sum_{\underline{k}} C^{(\underline{k})}\cdot\sum_{t=0}^{\eta(\underline{k})}(-1)^t\cdot\sum_{\underline{t}}\lambda(\underline{k},\underline{t})\binom{w-w(\underline{k},\underline{t})}{u-w(\underline{k},\underline{t})},$$

where $\underline{k}=(k_1,k_2,\ldots,k_p)$ is a partition of n,

$$C^{(\underline{k})}=(n!)\cdot\prod_{i=1}^{p}\frac{1}{k_i!}\binom{p}{i}^{k_i},\quad \eta(\underline{k})=\sum_{i=1}^{p-1}a_i,\quad a_i=k_i(p-i),$$

$$\lambda(\underline{k},\underline{t})=\prod_{i=1}^{p-1}\binom{a_i}{t_i},\quad w(\underline{k},\underline{t})=w(\underline{k})\cdot(1+\sum_{i=1}^{p-1}\frac{t_i}{i}),$$

$$w(\underline{k})=\prod_{i=1}^{p}(i)^{k_i},\quad w=p^n,\quad F^{(u)}=\binom{p^n}{u},\quad \underline{t}=(t_1,t_2,\ldots,t_{p-1})$$

is a partition of t, and $t_i \le a_i$.

(Proof) See [21].

Theorem 5.2: The average number of prime implicants of p-valued n-variable function is given by

$$G_p(n)=\sum_{\underline{k}} C^{(\underline{k})}\cdot 2^{-w(\underline{k})}\prod_{i=1}^{p-1}(1-2^{-w(\underline{k})/i})^{a_i}$$

(Proof) See [21].

Table 5.1 shows the values of $G_p(n,u)$ and $G_p(n)$ for p=2 and p=4. In the case of p=2, these results coincide with the results of [13] and [14]. We can obtain the following results form the numerical calculation.

1). $G_p(n,u)$ increases monotonously as u increases from 0, and at some u it takes the maximum value, then it decreases monotonously as u increases to 2^n. For example, $G_2(14,u)$ takes its maximum when $u=2^{14}\times 0.92$, and $G_4(7,u)$ takes its maximum when $u=2^{14}\times 0.96$.
2). $G_4(n/2,u)\ge G_2(n,u)$, when n and u are sufficiently large.
3). $G_p(n,\ p^n/2) \simeq G_p(n)$.

VI. Statistical Results

6.1 Average size of PLA's with two bit decoders

In order to estimate the size of two-level PLA's and PLA's with two-bit decoders, both expressions of two-valued variables and four-valued variables are minimized for numerous randomly generated functions. The program developed consists of two parts: the first part generates all the prime implicants of multiple-valued variables; the second part detects essential terms, row and column dominance relations, and obtains a minimal covering. Table 1.3 shows the average size of PLA's up to 10-variables. $d=(u/2^n)\times 100$ denotes the percentage of minterms which are mapped to one. For $0\le d\le 50$, the larger d, the more complex the function. When n=10 and d=50, PLA's with two bit decoders are about 26% smaller than two-level PLA's.

6.2 Average Number of Prime Implicants .

Fig.6.1 shows the average number of prime implicants, number of terms in a minimal solution, and number of E-terms for randomly generated functions of 10 variables. E-term includes the essential terms and the terms which are selected by row and column dominance relations. The number of prime implicants agree with the theoretical results shown in Table 5.1 within errors of a few percents. The fraction of E-terms rapidly decreases in the case of four valued variables as d increases to 50. In the case of four-valued variables, the number of prime implicants is larger but the number of E-terms are smaller than that of two-valued variables.

6.3 The Effect of the Assignment of Input Variables

In order to investigate the dependance on the way of assignment of input variables, ten randomly generated functions of 8 variables were generated for each density. Then, 105 expressions which corresponds to all possible ways of assignments were minimized: for each function, there are 105 different ways of assignment of 8 input variables to the 4 two-input decoders. Table 6.1 shows the statistical data of this exhaustive investigation. When the density is 40%, optimally assigned PLA's are about 10% smaller than non-optimally assigned PLA's. Table 6.1 also shows the average widths of two-level PLA's for the same functions. In the case of d=40%, optimally assigned PLA's with two-bit decoders are about 32% smaller than two-level PLA's.

VII. Conclusion and Comments

1). PLA's with two-bit decoders require smaller arrays than two-level PLA's. In the case of n=8 and d=40%, the former are 24% smaller than the latter.
2). The size of the arrays of PLA's with two-bit decoders can be reduced by optimizing the assignment of input variables. In the case of n=8 and d=40%, optimally assigned PLA's are 10% smaller than non-optimally assigned PLA's, and are 32% smaller than two-level PLA's. We have recently developed eight different heuristic algorithms which find optimal or near optimal assignments quickly [22],[23].
3). Minimization of the PLA's with two-bit decoders can be done by minimizing the expressions of four-valued logic functions.
4). The number of the prime implicants of four-valued function is greater than that of the corresponding two-valued function.
5). Classical method is statistically unsuitable to solve large minimization problem. We are now developing a heuristic program which finds a near minimal solution quickly.

Acknowledgement

The authors are grateful to Mr. K.Ishikawa for programming the algorithms. One of the authors (Tsutomu Sasao) wish to thank Prof. H. Ozaki and Prof. K. Kinoshita for their encouragement.

o denotes the number of prime implicants
Δ denotes the number of terms in a minimal solution*
× denotes the number of E-terms

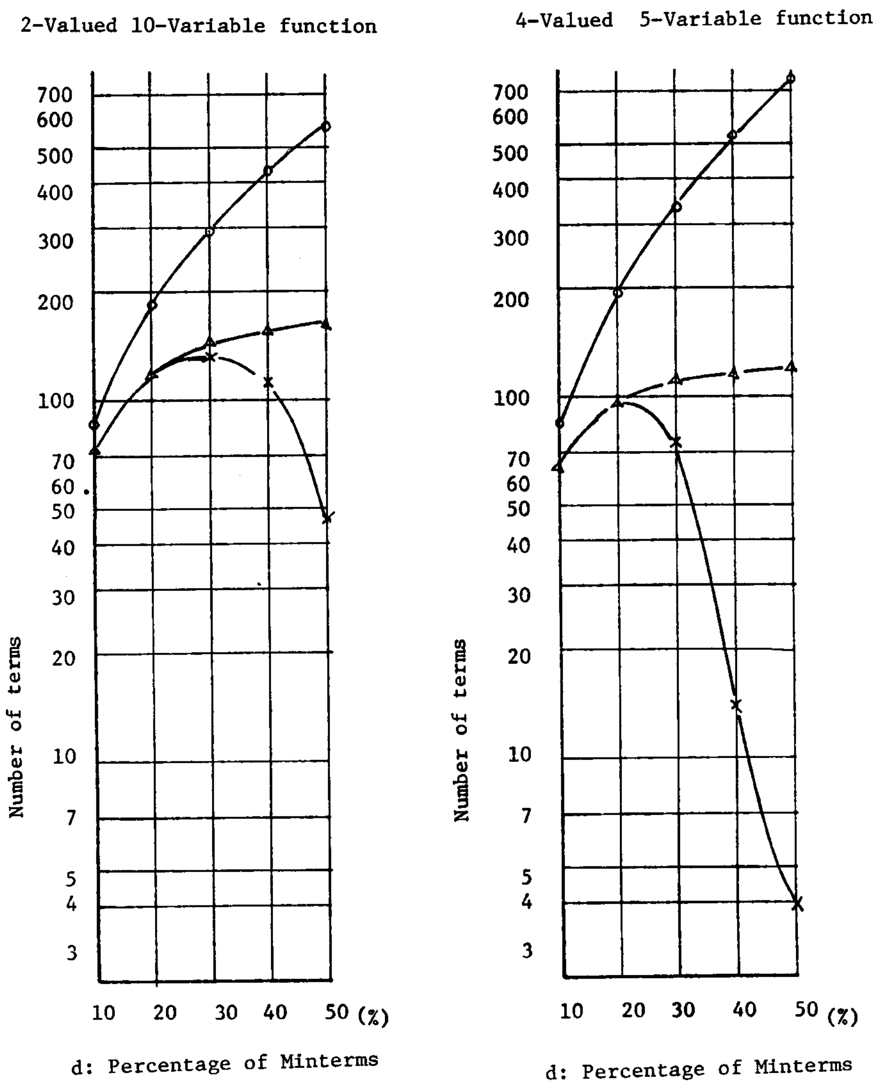

Fig. 6.1. Average numbers of prime implicants, terms in a minimal solution and the E-terms.

* The entries of 40% and 50% denote the average of 5 near minimal solution. The other entries denote the average of 10 minimal solution.

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] R.A.Wood,"High-speed dynamic programmable logic array chip, " IBM J.Res. Develop., Vol.19, pp.379-383, July 1975.
[5] W.N.Carr and J.P.Mize, MOS/LSI Design and Applications, Texas Instruments Electronics Series, MacGraw-Hill, New York, 1972.
[6] T.Sasao, "Basic considerations on the realization of programmable logic arrays (1) and (2)," (in Japanese), The Institute of Electronics and Communication Engineers of Japan, EC 77-12 and EC 77-13, June 1977.
[7] T.Sasao, "An application of multiple-valued logic to a design of programmable locic arrays," The Eighth International Sumposium on Multiple-Valued Logic, May 1978.
[8] S.Y.Su and P.T.Cheung, "Cubical notation for computer aided processing of multiple-valued switching functions," IEEE Trans. Comput., Vol.C-21, pp.995-1003, Sept. 1972.
[9] W.R.Smith III, "Minimization of multivalued functions," in Computer Science and Multiple-Valued Logic, North-Holland, 1977.
[10] M.Breuer, Design Automation of Digital Systems, Printice-Hall, Englewood Cliffs, U.S.A., 1972.
[11] M.Davio, J.P.Deschamps, and A.Thayse, Discrete and Switching Functions, George Publishing Co. and McGraw-Hill, New York, 1978.
[12] R.E.Miller, Switching Theory, Vol.I: Combinational Circuits, Jhon Wiley & Sons. Inc., New Yorkd, 1965.
[13] F.Mileto and G.Putzolu, "Average values of quantities appearing in Boolean function minimization," IEEE Trans. Electron.Comput., Vol.EC-13, pp.87-92, April 1964.
[14] F.Mileto ana G.Putzolu, "Average values of quantities appearing in multiple-output boolean minimization," IEEE Trans. on Electron. Comput. Vol.EC-14, pp.542-552, August 1965.
[15] A.Cobham, R.Fridshal, and J.H.North, "A statistical study of the minimization of boolean functions using integer linear programming," IBM Res. Rept. RC-756, June 1962.
[16] E.Morreale and M.Mennucci, "Computer experience on partitioned list algorithms," IEEE Trans. Comput., Vol.C-19, pp.1099-1105, Nov. 1970.
[17] L.Gilli, A.Laurentini, and A.R.Meo, "Statistical properties of switching function," IEEE Comput. Group Repository, R-70-207, 1970.
[18] B.Dunham and R.Fridshal, "The problem of simplifying logical expressions," Journal of Symbolic Logic, Vol.24, pp.17-19, 1959.
[19] V.V.Glagolev, "Some bounds for disjunctive normal forms of the algebra of logic,"Problemi Kibernetiki, Vol.19, pp.74-93, 1970. (Transl: Systems Theory Research, Consultants Bureau, New York).
[20] T.Sasao,"Basic considerations on the realization of programmable logic arrays (3)" (in Japanese), The Institute of Electronics and Communication Engineers of Japan, EC 77-74, March 1978.
[21] T.Sasao and H.Terada,"Basic considerations on the realization of programmable logic arrays (4) ---On the number of prime implicants of multiple-valued logic functions" (in Japanese), ibid, EC 78-65, Feb. 1979.
[22] K.Ishikawa,T.Sasao and H.Terada, "Basic considerations on the realization of programmable logic arrays (5)---On the assignment problem of input variables for PLA's with decoders" (in Japanese), ibid., EC 78-66, Feb. 1979.
[23] T.Sasao, K.Ishikawa, and H.Terada, "A heuristic method for the assignment of input variables to the PLA's with decoders" (in Japanese), National Convention of the Institute of Electronics and Communication Engineers of Japan, No.1176, March 1979.
[24] T.Sasao and H.Terada, "On the number of prime implicants of multiple-valued logic functions" (in Japanese),ibid., No.1175, March 1979.
[25] J.P.Roth, "Programmed logic array optimization, " IEEE Trans. Comput., Vol.C-27, pp.174-176, Feb. 1978.
[26] Z.Arevalo and J.G.Bredeson, "A method to simplify a boolean function into a near minimal sum-of-products for prgrammable logic arrays," IEEE Trans. Comput., Vol. C-27, pp.1028-1039, Nov. 1978.